S29NS512P, S29NS256P, S29NS128P

512/256/128 Mb (32/16/8 M x 16 bit)
1.8 V MirrorBit Flash Memory

Features

- Single 1.8V read/program/erase (1.70–1.95V)
- 90 nm MirrorBit Technology
- Multiplexed Data and Address for reduced I/O count
- Simultaneous Read/Write operation
- Full/Half drive output slew rate control
- 32-word Write Buffer
- Sixteen-bank architecture consisting of 64/32/16 MB for NS512/256/128P, respectively
- Four 32 kB sectors at the top of memory array (NS256/128P)
- 512 128 kB sectors (NS512P), 255/127 128 kB sectors (NS256/128P)
- Programmable linear (8/16/32) with or without wrap around and continuous burst read modes
- Secured Silicon Sector region consisting of 128 words each for factory and customer
- 20-year data retention (typical)
- Cycling Endurance: 100,000 cycles per sector (typical)
- RDY output indicates data available to system
- Command set compatible with JEDEC (42.4) standard
- Hardware (WP#) protection of highest two sectors
- Top Boot sector configuration (NS256/128P)
- Handshaking by monitoring RDY
- Offered Packages
  - NS512P: 64-ball FBGA (8 mm x 9.2 mm)
  - NS256P/NS128P: 44-ball FBGA (6.2 mm x 7.7 mm)
- Low V_{CC} write inhibit
- Persistent and Password methods of Advanced Sector Protection
- Write operation status bits indicate program and erase operation completion
- Suspend and Resume commands for Program and Erase operations
- Unlock Bypass program command to reduce programming time
- Synchronous or Asynchronous program operation, independent of burst control register settings
- V_{PP} input pin to reduce factory programming time
- Support for Common Flash Interface (CFI)

Performance Characteristics

Read Access Times	
Speed Option (MHz)	83 MHz
Max. Synch. Burst Access, ns (t_{BACC})	9.0 ns
Max. Asynch. Access Time, ns (t_{ACC})	80 ns
Max OE# Access Time, ns (t_{OE})	7.0 ns

Current Consumption (typical values)	
Continuous Burst Read @ 83 MHz	42 mA
Simultaneous Operation 83 MHz	60 mA
Program	30 mA
Standby Mode	20 µA

Typical Program & Erase Times	
Single Word Programming	40 µs
Effective Write Buffer Programming (V_{CC}) Per Word	9.4 µs
Effective Write Buffer Programming (V_{PP}) Per Word	6 µs
Sector Erase (16 Kword Sector)	450 ms
Sector Erase (64 Kword Sector)	900 ms

Contents

1. General Description

The Spansion S29NS512/256/128P are MirrorBit Flash products fabricated on 90 nm process technology. These burst mode Flash devices are capable of performing simultaneous read and write operations with zero latency on two separate banks using multiplexed data and address pins. These products can operate up to 83 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.Ordering Information

2. Ordering Information

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

S29NS 512 P xx BJ W 00 0

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

Model Number
00 = Standard

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

Package Type & Material Set
BJ = Very Thin Fine-Pitch BGA,Lead (Pb)-free LF35 Package

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

Process Technology
P = 90 nm MirrorBit Technology

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

Product Family
S29NS =1.8 Volt-Only Simultaneous Read/Write, Burst Mode Multiplexed Flash Memory

Valid Combinations					Package Type
Base Ordering Part Number	**Speed Option**	**Package Type, Material, & Temperature Range**	**Packing Type**	**Model Number**	
S29NS512P	0P, 0S	BJW (Lead (Pb)-free, LF35)	0, 3 (1)	00	8.0 mm x 9.2 mm, 64-ball
S29NS256P					6.2 mm x 7.7 mm, 44-ball
S29NS128P					

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.*

Valid Combinations

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

3. Input/Output Descriptions and Logic Symbol

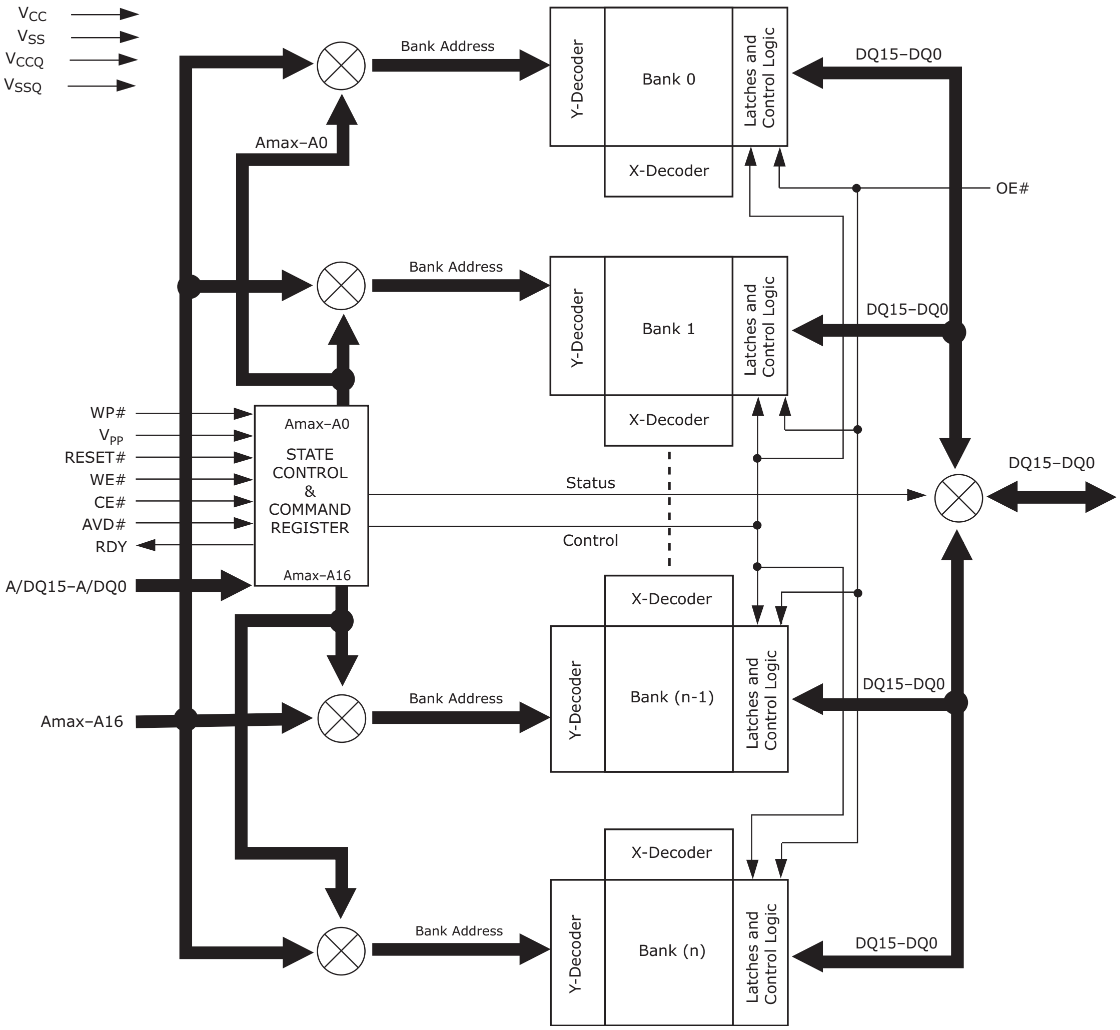
Table 3.1 identifies the input and output package connections provided on the device.

Table 3.1 Input/Output Descriptions

Symbol	Type	Description
A24 – A16	Input	Address inputs, S29NS512P.
A23 – A16	Input	Address inputs, S29NS256P.
A22 – A16	Input	Address inputs, S29NS128P.
A/DQ15 – A/DQ0	I/O	Multiplexed Address/Data input/output.
CE#	Input	Chip Enable. Asynchronous relative to CLK for the Burst mode.
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	Not Connected	No device internal signal is connected to the package connector nor is there any future plan to use the connector for a signal. The connection may safely be used for routing space for a signal on a Printed Circuit Board (PCB).
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.
WP#	Input	Write Protect. At V_{IL}, disables program and erase functions in the four top sectors. Should be at V_{IH} for all other conditions.
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. No device internal signal is currently connected to the package connector but there is potential future use for the connector for a signal. It is recommended to not use RFU connectors for PCB routing channels so that the PCB may take advantage of future enhanced features in compatible footprint devices.
DNU	Do Not Use	A device internal signal may be connected to the package connector. The connection may be used by Spansion for test or other purposes and is not intended for connection to any host system signal. Any DNU signal related function will be inactive when the signal is at V_{IL}. The signal has an internal pull-down resistor and may be left unconnected in the host system or may be tied to V_{SS}. Do not use these connections for PCB signal routing channels. Do not connect any host system signal to these connections.

4. Block Diagrams

Figure 4.1 Simultaneous Operation Circuit



Notes

1. *Amax = A24 for NS512P, A23 for NS256P, A22 for NS128P.*
2. *Bank (n) = 15 for NS512P/ NS256P/ NS128P.*

5. Physical Dimensions/Connection Diagrams

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

5.1 Related Documents

The following documents contain information relating to the S29NS-P devices. Click on the title or go to www.spansion.com, or request a copy from your sales office.

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

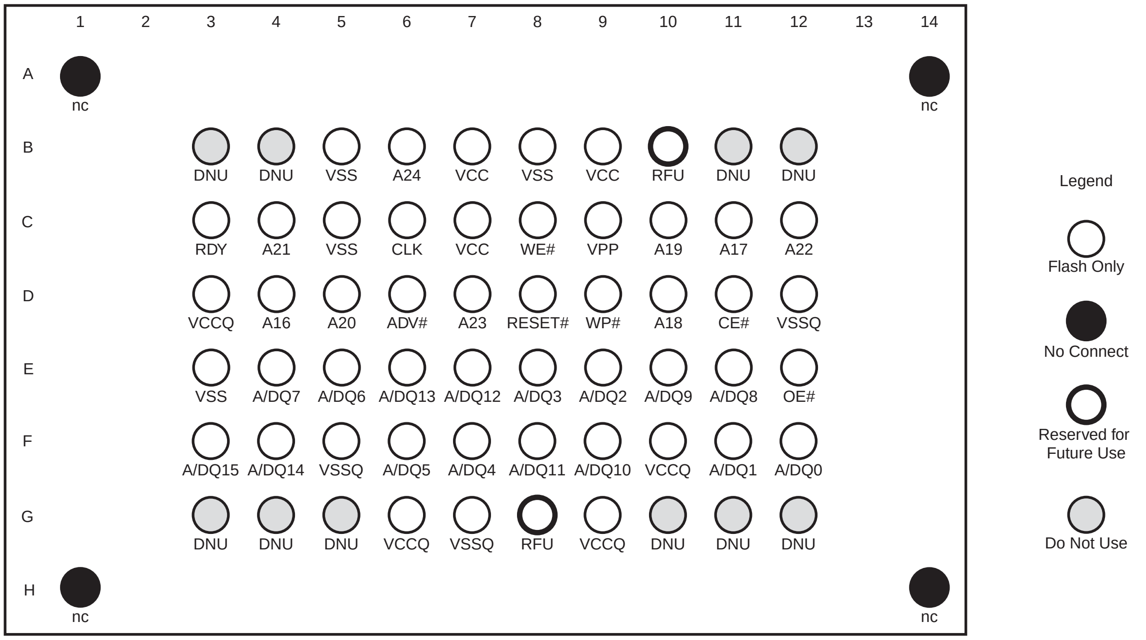
5.2 Special Handling Instructions for FBGA Package

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

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

5.2.1 64-Ball Fine-Pitch Grid Array, S29NS512P

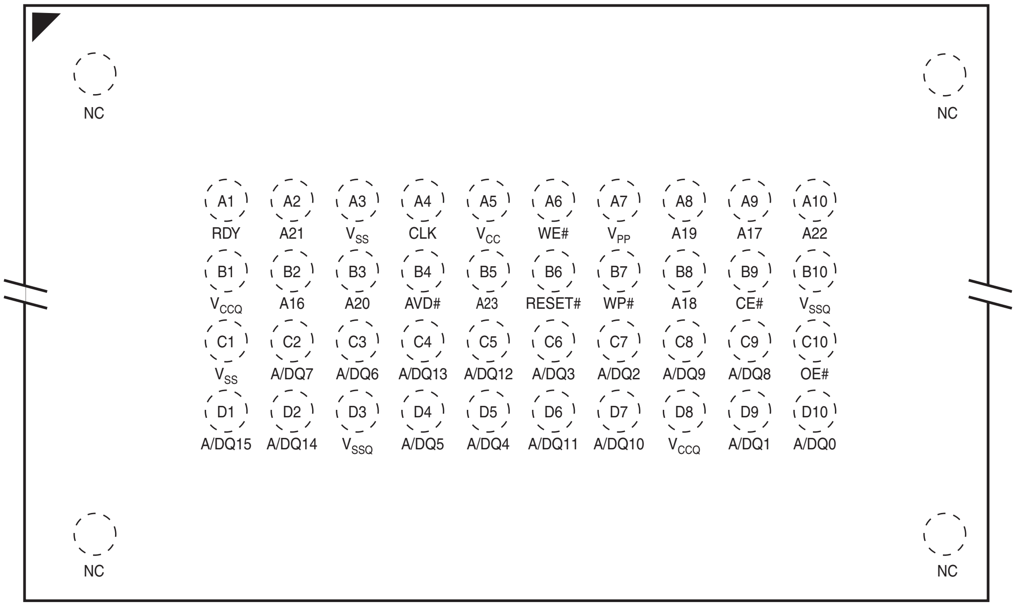
Figure 5.1 64-Ball Very Thin Fine-Pitch Ball Grid Array, S29NS512P Top View, Balls Facing Down



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	nc													nc
B			DNU	DNU	VSS	A24	VCC	VSS	VCC	RFU	DNU	DNU		
C			RDY	A21	VSS	CLK	VCC	WE#	VPP	A19	A17	A22		
D			VCCQ	A16	A20	ADV#	A23	RESET#	WP#	A18	CE#	VSSQ		
E			VSS	A/DQ7	A/DQ6	A/DQ13	A/DQ12	A/DQ3	A/DQ2	A/DQ9	A/DQ8	OE#		
F			A/DQ15	A/DQ14	VSSQ	A/DQ5	A/DQ4	A/DQ11	A/DQ10	VCCQ	A/DQ1	A/DQ0		
G			DNU	DNU	DNU	VCCQ	VSSQ	RFU	VCCQ	DNU	DNU	DNU		
H	nc													nc

5.2.2 44-Ball Very Thin Fine-Pitch Ball Grid Array, S29NS256P

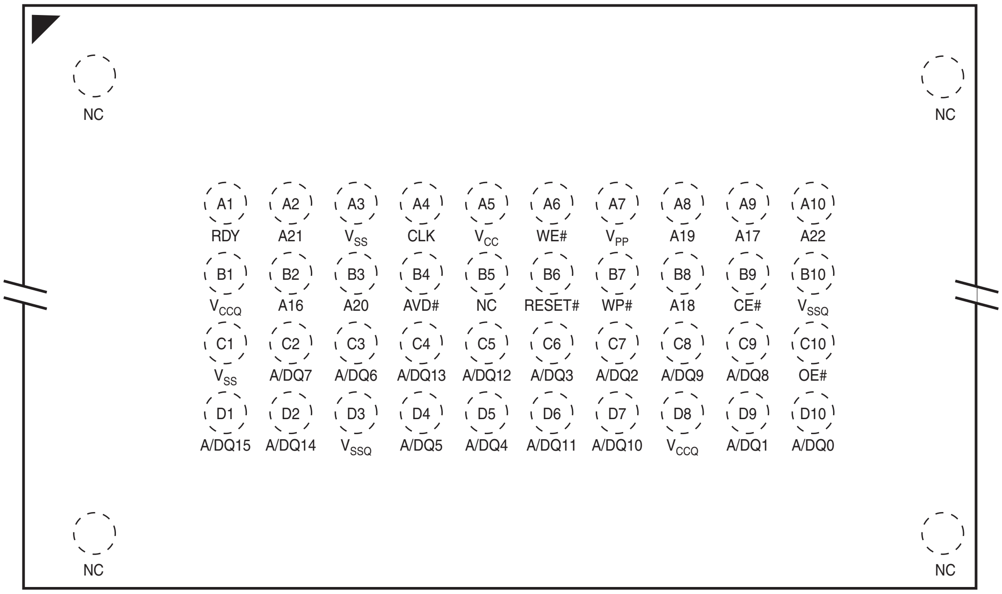
Figure 5.2 44-Ball Very Thin Fine-Pitch Ball Grid Array, S29NS256P Top View, Balls Facing Down



A1 RDY	A2 A21	A3 V_{SS}	A4 CLK	A5 V_{CC}	A6 WE#	A7 V_{PP}	A8 A19	A9 A17	A10 A22
B1 V_{CCQ}	B2 A16	B3 A20	B4 AVD#	B5 A23	B6 RESET#	B7 WP#	B8 A18	B9 CE#	B10 V_{SSQ}
C1 V_{SS}	C2 A/DQ7	C3 A/DQ6	C4 A/DQ13	C5 A/DQ12	C6 A/DQ3	C7 A/DQ2	C8 A/DQ9	C9 A/DQ8	C10 OE#
D1 A/DQ15	D2 A/DQ14	D3 V_{SSQ}	D4 A/DQ5	D5 A/DQ4	D6 A/DQ11	D7 A/DQ10	D8 V_{CCQ}	D9 A/DQ1	D10 A/DQ0

5.2.3 44-Ball Very Thin Fine-Pitch Ball Grid Array, S29NS128P

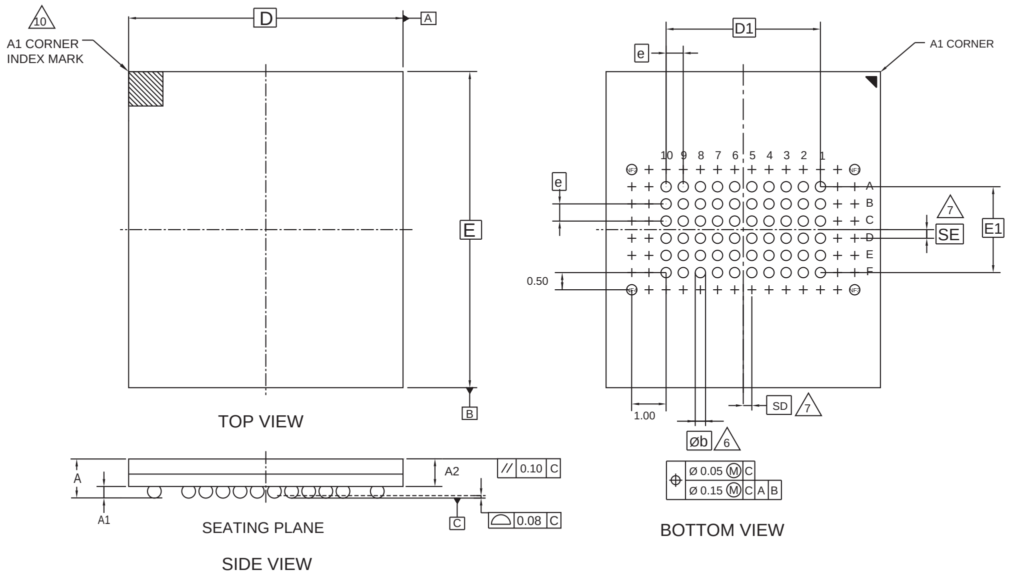
Figure 5.3 44-Ball Very Thin Fine-Pitch Ball Grid Array, S29NS128P Top View, Balls Facing Down



A1 RDY	A2 A21	A3 V_{SS}	A4 CLK	A5 V_{CC}	A6 WE#	A7 V_{PP}	A8 A19	A9 A17	A10 A22
B1 V_{CCQ}	B2 A16	B3 A20	B4 AVD#	B5 NC	B6 RESET#	B7 WP#	B8 A18	B9 CE#	B10 V_{SSQ}
C1 V_{SS}	C2 A/DQ7	C3 A/DQ6	C4 A/DQ13	C5 A/DQ12	C6 A/DQ3	C7 A/DQ2	C8 A/DQ9	C9 A/DQ8	C10 OE#
D1 A/DQ15	D2 A/DQ14	D3 V_{SSQ}	D4 A/DQ5	D5 A/DQ4	D6 A/DQ11	D7 A/DQ10	D8 V_{CCQ}	D9 A/DQ1	D10 A/DQ0

5.2.4 VDD064—64-Ball Very Thin Fine-Pitch Ball Grid Array

Figure 5.4 VDD064—64-Ball Very Thin Fine-Pitch Ball Grid Array, S29NS512P



PACKAGE	VDD 064			
JEDEC	N/A			
	8.00 mm x 9.20 mm NOM PACKAGE			
SYMBOL	MIN	NOM	MAX	NOTE
A	0.86	---	1.00	OVERALL THICKNESS
A1	0.20	---	---	BALL HEIGHT
A2	0.66	0.71	0.76	BODY THICKNESS
D	7.90	8.00	8.10	BODY SIZE
E	9.10	9.20	9.30	BODY SIZE
D1		4.50		BALL FOOTPRINT
E1		2.50		BALL FOOTPRINT
MD		10		ROW MATRIX SIZE D DIRECTION
ME		6		ROW MATRIX SIZE E DIRECTION
N		64		TOTAL BALL COUNT
Ø b	0.25	0.30	0.35	BALL DIAMETER
e		0.50		BALL PITCH
SD / SE		0.25		SOLDER BALL PLACEMENT
				DEPOPULATED SOLDER BALLS

NOTES:

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

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

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

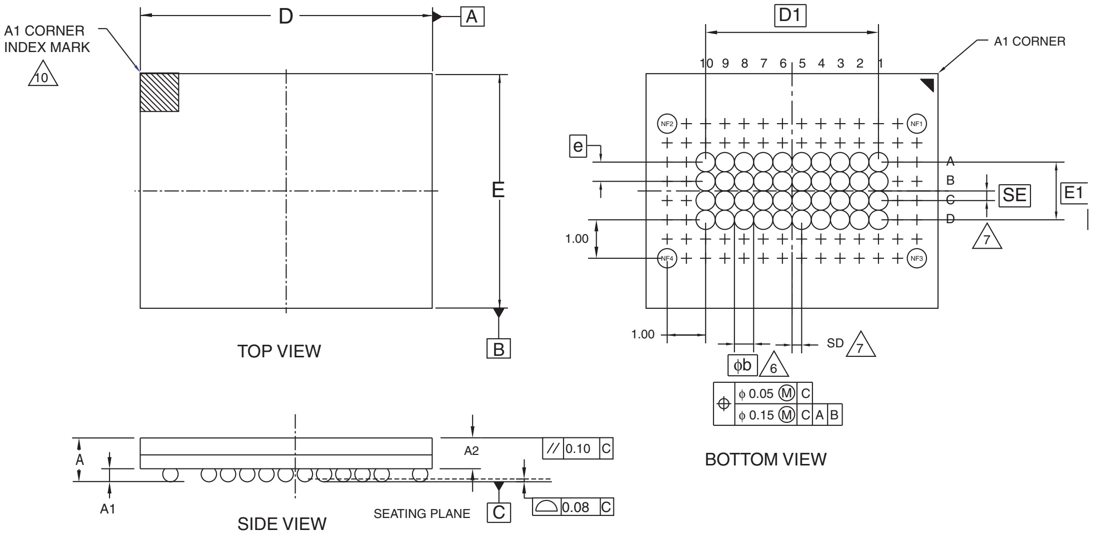
   WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW PARALLEL TO THE D OR E DIMENSION, RESPECTIVELY, SD OR SE = 0.000.

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

3533 \ 16-038.27 \ 12.13.05

5.2.5 VDE44-44-Ball Very Thin Fine-Pitch Ball Grid Array, 7.7 mm x 6.2 mm

Figure 5.5 VDE044—44-Ball Very Thin Fine-Pitch Ball Grid Array, S29NS128/256P



PACKAGE	VDE 044			
JEDEC	N/A			
	7.70 mm x 6.20 mm NOM PACKAGE			
SYMBOL	MIN	NOM	MAX	NOTE
A	0.86	---	1.00	OVERALL THICKNESS
A1	0.20	---	---	BALL HEIGHT
A2	0.66	0.71	0.76	BODY THICKNESS
D	7.6	7.7	7.8	BODY SIZE
E	6.1	6.2	6.3	BODY SIZE
D1		4.50		BALL FOOTPRINT
E1		1.50		BALL FOOTPRINT
MD		10		ROW MATRIX SIZE D DIRECTION
ME		4		ROW MATRIX SIZE E DIRECTION
N		44		TOTAL BALL COUNT
φb	0.25	0.30	0.35	BALL DIAMETER
e		0.50 BSC.		BALL PITCH
SD / SE		0.25 BSC.		SOLDER BALL PLACEMENT
?				DEPOPULATED SOLDER BALLS

NOTES:

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

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

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

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

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

3308.2 \ 16-038.9L

6. Product Overview

The S29NS-P family consists of 512, 256, and 128 Mb, 1.8 volts-only, simultaneous read/write burst mode, multiplexed Flash device optimized for today's wireless designs that demand a large storage array, rich functionality, and low power consumption.

These devices are organized in 32, 16, or 8 Mwords of 16 bits each and are capable of continuous, synchronous (burst) read or linear read (8-word, 16-word, or 32-word aligned group) with or without wrap around. These flash devices multiplex the data and addresses for reduced I/O count. These products also offer single word programming or a 32-word buffer for programming with program/erase and suspend functionality. Additional features include:

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

6.1 Memory Map

The S29NS512/256/128P devices consist of 16 banks organized as shown in Tables 6.1 – 6.3.

Table 6.1 S29NS512P Sector and Memory Address Map (Sheet 1 of 8)

Bank	Sector	Sector Size	Address Range
Bank 0	SA0	64 Kwords	000000h–00FFFFh
	SA1	64 Kwords	010000h–01FFFFh
	SA2	64 Kwords	020000h–02FFFFh
	SA3	64 Kwords	030000h–03FFFFh
	SA4	64 Kwords	040000h–04FFFFh
	SA5	64 Kwords	050000h–05FFFFh
	SA6	64 Kwords	060000h–06FFFFh
	SA7	64 Kwords	070000h–07FFFFh
	SA8	64 Kwords	080000h–08FFFFh
	SA9	64 Kwords	090000h–09FFFFh
	SA10	64 Kwords	0A0000h–0AFFFFh
	SA11	64 Kwords	0B0000h–0BFFFFh
	SA12	64 Kwords	0C0000h–0CFFFFh
	SA13	64 Kwords	0D0000h–0DFFFFh
	SA14	64 Kwords	0E0000h–0EFFFFh
	SA15	64 Kwords	0F0000h–0FFFFFh
	SA16	64 Kwords	100000h–10FFFFh
	SA17	64 Kwords	110000h–11FFFFh
	SA18	64 Kwords	120000h–12FFFFh
	SA19	64 Kwords	130000h–13FFFFh
	SA20	64 Kwords	140000h–14FFFFh
	SA21	64 Kwords	150000h–15FFFFh
	SA22	64 Kwords	160000h–16FFFFh
	SA23	64 Kwords	170000h–17FFFFh
	SA24	64 Kwords	180000h–18FFFFh
	SA25	64 Kwords	190000h–19FFFFh
	SA26	64 Kwords	1A0000h–1AFFFFh
	SA27	64 Kwords	1B0000h–1BFFFFh
	SA28	64 Kwords	1C0000h–1CFFFFh
	SA29	64 Kwords	1D0000h–1DFFFFh
	SA30	64 Kwords	1E0000h–1EFFFFh
	SA31	64 Kwords	1F0000h–1FFFFFh
Bank 1	SA32	64 Kwords	200000h–20FFFFh
	SA33	64 Kwords	210000h–21FFFFh
	SA34	64 Kwords	220000h–22FFFFh
	SA35	64 Kwords	230000h–23FFFFh
	SA36	64 Kwords	240000h–24FFFFh
	SA37	64 Kwords	250000h–25FFFFh
	SA38	64 Kwords	260000h–26FFFFh
	SA39	64 Kwords	270000h–27FFFFh
	SA40	64 Kwords	280000h–28FFFFh
	SA41	64 Kwords	290000h–29FFFFh
	SA42	64 Kwords	2A0000h–2AFFFFh
	SA43	64 Kwords	2B0000h–2BFFFFh
	SA44	64 Kwords	2C0000h–2CFFFFh
	SA45	64 Kwords	2D0000h–2DFFFFh
	SA46	64 Kwords	2E0000h–2EFFFFh
	SA47	64 Kwords	2F0000h–2FFFFFh
	SA48	64 Kwords	300000h–30FFFFh
	SA49	64 Kwords	310000h–31FFFFh
	SA50	64 Kwords	320000h–32FFFFh
	SA51	64 Kwords	330000h–33FFFFh
	SA52	64 Kwords	340000h–34FFFFh
	SA53	64 Kwords	350000h–35FFFFh
	SA54	64 Kwords	360000h–36FFFFh
	SA55	64 Kwords	370000h–37FFFFh
	SA56	64 Kwords	380000h–38FFFFh
	SA57	64 Kwords	390000h–39FFFFh
	SA58	64 Kwords	3A0000h–3AFFFFh
	SA59	64 Kwords	3B0000h–3BFFFFh
	SA60	64 Kwords	3C0000h–3CFFFFh
	SA61	64 Kwords	3D0000h–3DFFFFh
	SA62	64 Kwords	3E0000h–3EFFFFh
	SA63	64 Kwords	3F0000h–3FFFFFh

Table 6.1 S29NS512P Sector and Memory Address Map (Sheet 2 of 8)

Bank	Sector	Sector Size	Address Range
Bank 2	SA64	64 Kwords	400000h–40FFFFh
	SA65	64 Kwords	410000h–41FFFFh
	SA66	64 Kwords	420000h–42FFFFh
	SA67	64 Kwords	430000h–43FFFFh
	SA68	64 Kwords	440000h–44FFFFh
	SA69	64 Kwords	450000h–45FFFFh
	SA70	64 Kwords	460000h–46FFFFh
	SA71	64 Kwords	470000h–47FFFFh
	SA72	64 Kwords	480000h–48FFFFh
	SA73	64 Kwords	490000h–49FFFFh
	SA74	64 Kwords	4A0000h–4AFFFFh
	SA75	64 Kwords	4B0000h–4BFFFFh
	SA76	64 Kwords	4C0000h–4CFFFFh
	SA77	64 Kwords	4D0000h–4DFFFFh
	SA78	64 Kwords	4E0000h–4EFFFFh
	SA79	64 Kwords	4F0000h–4FFFFFh
	SA80	64 Kwords	500000h–50FFFFh
	SA81	64 Kwords	510000h–51FFFFh
	SA82	64 Kwords	520000h–52FFFFh
	SA83	64 Kwords	530000h–53FFFFh
	SA84	64 Kwords	540000h–54FFFFh
	SA85	64 Kwords	550000h–55FFFFh
	SA86	64 Kwords	560000h–56FFFFh
	SA87	64 Kwords	570000h–57FFFFh
	SA88	64 Kwords	580000h–58FFFFh
	SA89	64 Kwords	590000h–59FFFFh
	SA90	64 Kwords	5A0000h–5AFFFFh
	SA91	64 Kwords	5B0000h–5BFFFFh
	SA92	64 Kwords	5C0000h–5CFFFFh
	SA93	64 Kwords	5D0000h–5DFFFFh
	SA94	64 Kwords	5E0000h–5EFFFFh
	SA95	64 Kwords	5F0000h–5FFFFFh
Bank 3	SA96	64 K words	600000h–60FFFFh
	SA97	64 K words	610000h–61FFFFh
	SA98	64 K words	620000h–62FFFFh
	SA99	64 K words	630000h–63FFFFh
	SA100	64 K words	640000h–64FFFFh
	SA101	64 K words	650000h–65FFFFh
	SA102	64 K words	660000h–66FFFFh
	SA103	64 K words	670000h–67FFFFh
	SA104	64 K words	680000h–68FFFFh
	SA105	64 K words	690000h–69FFFFh
	SA106	64 K words	6A0000h–6AFFFFh
	SA107	64 K words	6B0000h–6BFFFFh
	SA108	64 K words	6C0000h–6CFFFFh
	SA109	64 K words	6D0000h–6DFFFFh
	SA110	64 K words	6E0000h–6EFFFFh
	SA111	64 K words	6F0000h–6FFFFFh
	SA112	64 K words	700000h–70FFFFh
	SA113	64 K words	710000h–71FFFFh
	SA114	64 K words	720000h–72FFFFh
	SA115	64 K words	730000h–73FFFFh
	SA116	64 K words	740000h–74FFFFh
	SA117	64 K words	750000h–75FFFFh
	SA118	64 K words	760000h–76FFFFh
	SA119	64 K words	770000h–77FFFFh
	SA120	64 K words	780000h–78FFFFh
	SA121	64 K words	790000h–79FFFFh
	SA122	64 K words	7A0000h–7AFFFFh
	SA123	64 K words	7B0000h–7BFFFFh
	SA124	64 K words	7C0000h–7CFFFFh
	SA125	64 K words	7D0000h–7DFFFFh
	SA126	64 K words	7E0000h–7EFFFFh
	SA127	64 K words	7F0000h–7FFFFFh

Table 6.1 S29NS512P Sector and Memory Address Map (Sheet 3 of 8)

Bank	Sector	Sector Size	Address Range
Bank 4	SA128	64 Kwords	800000h–80FFFFh
	SA129	64 Kwords	810000h–81FFFFh
	SA130	64 Kwords	820000h–82FFFFh
	SA131	64 Kwords	830000h–83FFFFh
	SA132	64 Kwords	840000h–84FFFFh
	SA133	64 Kwords	850000h–85FFFFh
	SA134	64 Kwords	860000h–86FFFFh
	SA135	64 Kwords	870000h–87FFFFh
	SA136	64 Kwords	880000h–88FFFFh
	SA137	64 Kwords	890000h–89FFFFh
	SA138	64 Kwords	8A0000h–8AFFFFh
	SA139	64 Kwords	8B0000h–8BFFFFh
	SA140	64 Kwords	8C0000h–8CFFFFh
	SA141	64 Kwords	8D0000h–8DFFFFh
	SA142	64 Kwords	8E0000h–8EFFFFh
	SA143	64 Kwords	8F0000h–8FFFFFh
	SA144	64 Kwords	900000h–90FFFFh
	SA145	64 Kwords	910000h–91FFFFh
	SA146	64 Kwords	920000h–92FFFFh
	SA147	64 Kwords	930000h–93FFFFh
	SA148	64 Kwords	940000h–94FFFFh
	SA149	64 Kwords	950000h–95FFFFh
	SA150	64 Kwords	960000h–96FFFFh
	SA151	64 Kwords	970000h–97FFFFh
	SA152	64 Kwords	980000h–98FFFFh
	SA153	64 Kwords	990000h–99FFFFh
	SA154	64 Kwords	9A0000h–9AFFFFh
	SA155	64 Kwords	9B0000h–9BFFFFh
	SA156	64 Kwords	9C0000h–9CFFFFh
	SA157	64 Kwords	9D0000h–9DFFFFh
	SA158	64 Kwords	9E0000h–9EFFFFh
	SA159	64 Kwords	9F0000h–9FFFFFh
Bank 5	SA160	64 Kwords	A00000h–A0FFFFh
	SA161	64 Kwords	A10000h–A1FFFFh
	SA162	64 Kwords	A20000h–A2FFFFh
	SA163	64 Kwords	A30000h–A3FFFFh
	SA164	64 Kwords	A40000h–A4FFFFh
	SA165	64 Kwords	A50000h–A5FFFFh
	SA166	64 Kwords	A60000h–A6FFFFh
	SA167	64 Kwords	A70000h–A7FFFFh
	SA168	64 Kwords	A80000h–A8FFFFh
	SA169	64 Kwords	A90000h–A9FFFFh
	SA170	64 Kwords	AA0000h–AAFFFFh
	SA171	64 Kwords	AB0000h–ABFFFFh
	SA172	64 Kwords	AC0000h–ACFFFFh
	SA173	64 Kwords	AD0000h–ADFFFFh
	SA174	64 Kwords	AE0000h–AEFFFFh
	SA175	64 Kwords	AF0000h–AFFFFFh
	SA176	64 Kwords	B00000h–B0FFFFh
	SA177	64 Kwords	B10000h–B1FFFFh
	SA178	64 Kwords	B20000h–B2FFFFh
	SA179	64 Kwords	B30000h–B3FFFFh
	SA180	64 Kwords	B40000h–B4FFFFh
	SA181	64 Kwords	B50000h–B5FFFFh
	SA182	64 Kwords	B60000h–B6FFFFh
	SA183	64 Kwords	B70000h–B7FFFFh
	SA184	64 Kwords	B80000h–B8FFFFh
	SA185	64 Kwords	B90000h–B9FFFFh
	SA186	64 Kwords	BA0000h–BAFFFFh
	SA187	64 Kwords	BB0000h–BBFFFFh
	SA188	64 Kwords	BC0000h–BCFFFFh
	SA189	64 Kwords	BD0000h–BDFFFFh
	SA190	64 Kwords	BE0000h–BEFFFFh
	SA191	64 Kwords	BF0000h–BFFFFFh

Table 6.1 S29NS512P Sector and Memory Address Map (Sheet 4 of 8)

Bank	Sector	Sector Size	Address Range
Bank 6	SA192	64 Kwords	C00000h–C0FFFFh
	SA193	64 Kwords	C10000h–C1FFFFh
	SA194	64 Kwords	C20000h–C2FFFFh
	SA195	64 Kwords	C30000h–C3FFFFh
	SA196	64 Kwords	C40000h–C4FFFFh
	SA197	64 Kwords	C50000h–C5FFFFh
	SA198	64 Kwords	C60000h–C6FFFFh
	SA199	64 Kwords	C70000h–C7FFFFh
	SA200	64 Kwords	C80000h–C8FFFFh
	SA201	64 Kwords	C90000h–C9FFFFh
	SA202	64 Kwords	CA0000h–CAFFFFh
	SA203	64 Kwords	CB0000h–CBFFFFh
	SA204	64 Kwords	CC0000h–CCFFFFh
	SA205	64 Kwords	CD0000h–CDFFFFh
	SA206	64 Kwords	CE0000h–CEFFFFh
	SA207	64 Kwords	CF0000h–CFFFFFh
	SA208	64 Kwords	D00000h–D0FFFFh
	SA209	64 Kwords	D10000h–D1FFFFh
	SA210	64 Kwords	D20000h–D2FFFFh
	SA211	64 Kwords	D30000h–D3FFFFh
	SA212	64 Kwords	D40000h–D4FFFFh
	SA213	64 Kwords	D50000h–D5FFFFh
	SA214	64 Kwords	D60000h–D6FFFFh
	SA215	64 Kwords	D70000h–D7FFFFh
	SA216	64 Kwords	D80000h–D8FFFFh
	SA217	64 Kwords	D90000h–D9FFFFh
	SA218	64 Kwords	DA0000h–DAFFFFh
	SA219	64 Kwords	DB0000h–DBFFFFh
	SA220	64 Kwords	DC0000h–DCFFFFh
	SA221	64 Kwords	DD0000h–DDFFFFh
	SA222	64 Kwords	DE0000h–DEFFFFh
	SA223	64 Kwords	DF0000h–DFFFFFh
Bank 7	SA224	64 K words	E00000h–E0FFFFh
	SA225	64 K words	E10000h–E1FFFFh
	SA226	64 K words	E20000h–E2FFFFh
	SA227	64 K words	E30000h–E3FFFFh
	SA228	64 K words	E40000h–E4FFFFh
	SA229	64 K words	E50000h–E5FFFFh
	SA230	64 K words	E60000h–E6FFFFh
	SA231	64 K words	E70000h–E7FFFFh
	SA232	64 K words	E80000h–E8FFFFh
	SA233	64 K words	E90000h–E9FFFFh
	SA234	64 K words	EA0000h–EAFFFFh
	SA235	64 K words	EB0000h–EBFFFFh
	SA236	64 K words	EC0000h–ECFFFFh
	SA237	64 K words	ED0000h–EDFFFFh
	SA238	64 K words	EE0000h–EEFFFFh
	SA239	64 K words	EF0000h–EFFFFFh
	SA240	64 K words	F00000h–F0FFFFh
	SA241	64 K words	F10000h–F1FFFFh
	SA242	64 K words	F20000h–F2FFFFh
	SA243	64 K words	F30000h–F3FFFFh
	SA244	64 K words	F40000h–F4FFFFh
	SA245	64 K words	F50000h–F5FFFFh
	SA246	64 K words	F60000h–F6FFFFh
	SA247	64 K words	F70000h–F7FFFFh
	SA248	64 K words	F80000h–F8FFFFh
	SA249	64 K words	F90000h–F9FFFFh
	SA250	64 K words	FA0000h–FAFFFFh
	SA251	64 K words	FB0000h–FBFFFFh
	SA252	64 K words	FC0000h–FCFFFFh
	SA253	64 K words	FD0000h–FDFFFFh
	SA254	64 K words	FE0000h–FEFFFFh
	SA255	64 K words	FF0000h–FFFFFFh

Table 6.1 S29NS512P Sector and Memory Address Map (Sheet 5 of 8)

Bank	Sector	Sector Size	Address Range
Bank 8	SA256	64 Kwords	1000000h-100FFFFh
	SA257	64 Kwords	1010000h-101FFFFh
	SA258	64 Kwords	1020000h-102FFFFh
	SA259	64 Kwords	1030000h-103FFFFh
	SA260	64 Kwords	1040000h-104FFFFh
	SA261	64 Kwords	1050000h-105FFFFh
	SA262	64 Kwords	1060000h-106FFFFh
	SA263	64 Kwords	1070000h-107FFFFh
	SA264	64 Kwords	1030000h-108FFFFh
	SA265	64 Kwords	1090000h-109FFFFh
	SA266	64 Kwords	10A0000h-10AFFFFh
	SA267	64 Kwords	10B0000h-10BFFFFh
	SA268	64 Kwords	10C0000h-10CFFFFh
	SA269	64 Kwords	10D0000h-10DFFFFh
	SA270	64 Kwords	10E0000h-10EFFFFh
	SA271	64 Kwords	10F0000h-10FFFFFh
	SA272	64 Kwords	1100000h-110FFFFh
	SA273	64 Kwords	1110000h-111FFFFh
	SA274	64 Kwords	1120000h-112FFFFh
	SA275	64 Kwords	1130000h-113FFFFh
	SA276	64 Kwords	1140000h-114FFFFh
	SA277	64 Kwords	1150000h-115FFFFh
	SA278	64 Kwords	1160000h-116FFFFh
	SA279	64 Kwords	1170000h-117FFFFh
	SA280	64 Kwords	1180000h-118FFFFh
	SA281	64 Kwords	1190000h-119FFFFh
	SA282	64 Kwords	11A0000h-11AFFFFh
	SA283	64 Kwords	11B0000h-11BFFFFh
	SA284	64 Kwords	11C0000h-11CFFFFh
	SA285	64 Kwords	11D0000h-11DFFFFh
	SA286	64 Kwords	11E0000h-11EFFFFh
	SA287	64 Kwords	11F0000h-11FFFFFh

Bank	Sector	Sector Size	Address Range
Bank 9	SA288	64 Kwords	1200000h-120FFFFh
	SA289	64 Kwords	1210000h-121FFFFh
	SA290	64 Kwords	1220000h-122FFFFh
	SA291	64 Kwords	1230000h-123FFFFh
	SA292	64 Kwords	1240000h-124FFFFh
	SA293	64 Kwords	1250000h-125FFFFh
	SA294	64 Kwords	1260000h-126FFFFh
	SA295	64 Kwords	1270000h-127FFFFh
	SA296	64 Kwords	1230000h-128FFFFh
	SA297	64 Kwords	1290000h-129FFFFh
	SA298	64 Kwords	12A0000h-12AFFFFh
	SA299	64 Kwords	12B0000h-12BFFFFh
	SA300	64 Kwords	12C0000h-12CFFFFh
	SA301	64 Kwords	12D0000h-12DFFFFh
	SA302	64 Kwords	12E0000h-12EFFFFh
	SA303	64 Kwords	12F0000h-12FFFFFh
	SA304	64 Kwords	1300000h-130FFFFh
	SA305	64 Kwords	1310000h-131FFFFh
	SA306	64 Kwords	1320000h-132FFFFh
	SA307	64 Kwords	1330000h-133FFFFh
	SA308	64 Kwords	1340000h-134FFFFh
	SA309	64 Kwords	1350000h-135FFFFh
	SA310	64 Kwords	1360000h-136FFFFh
	SA311	64 Kwords	1370000h-137FFFFh
	SA312	64 Kwords	1380000h-138FFFFh
	SA313	64 Kwords	1390000h-139FFFFh
	SA314	64 Kwords	13A0000h-13AFFFFh
	SA315	64 Kwords	13B0000h-13BFFFFh
	SA316	64 Kwords	13C0000h-13CFFFFh
	SA317	64 Kwords	13D0000h-13DFFFFh
	SA318	64 Kwords	13E0000h-13EFFFFh
	SA319	64 Kwords	13F0000h-13FFFFFh

Table 6.1 S29NS512P Sector and Memory Address Map (Sheet 6 of 8)

Bank	Sector	Sector Size	Address Range
Bank 10	SA320	64 Kwords	1400000h-140FFFFh
	SA321	64 Kwords	1410000h-141FFFFh
	SA322	64 Kwords	1420000h-142FFFFh
	SA323	64 Kwords	1430000h-143FFFFh
	SA324	64 Kwords	1440000h-144FFFFh
	SA325	64 Kwords	1450000h-145FFFFh
	SA326	64 Kwords	1460000h-146FFFFh
	SA327	64 Kwords	1470000h-147FFFFh
	SA328	64 Kwords	1430000h-148FFFFh
	SA329	64 Kwords	1490000h-149FFFFh
	SA330	64 Kwords	14A0000h-14AFFFFh
	SA331	64 Kwords	14B0000h-14BFFFFh
	SA332	64 Kwords	14C0000h-14CFFFFh
	SA333	64 Kwords	14D0000h-14DFFFFh
	SA334	64 Kwords	14E0000h-14EFFFFh
	SA335	64 Kwords	14F0000h-14FFFFFh
	SA336	64 Kwords	1500000h-150FFFFh
	SA337	64 Kwords	1510000h-151FFFFh
	SA338	64 Kwords	1520000h-152FFFFh
	SA339	64 Kwords	1530000h-153FFFFh
	SA340	64 Kwords	1540000h-154FFFFh
	SA341	64 Kwords	1550000h-155FFFFh
	SA342	64 Kwords	1560000h-156FFFFh
	SA343	64 Kwords	1570000h-157FFFFh
	SA344	64 Kwords	1580000h-158FFFFh
	SA345	64 Kwords	1590000h-159FFFFh
	SA346	64 Kwords	15A0000h-15AFFFFh
	SA347	64 Kwords	15B0000h-15BFFFFh
	SA348	64 Kwords	15C0000h-15CFFFFh
	SA349	64 Kwords	15D0000h-15DFFFFh
	SA350	64 Kwords	15E0000h-15EFFFFh
	SA351	64 Kwords	15F0000h-15FFFFFh

Bank	Sector	Sector Size	Address Range
Bank 11	SA352	64 K words	1600000h-160FFFFh
	SA353	64 K words	1610000h-161FFFFh
	SA354	64 K words	1620000h-162FFFFh
	SA355	64 K words	1630000h-163FFFFh
	SA356	64 K words	1640000h-164FFFFh
	SA357	64 K words	1650000h-165FFFFh
	SA358	64 K words	1660000h-166FFFFh
	SA359	64 K words	1670000h-167FFFFh
	SA360	64 K words	1630000h-168FFFFh
	SA361	64 K words	1690000h-169FFFFh
	SA362	64 K words	16A0000h-16AFFFFh
	SA363	64 K words	16B0000h-16BFFFFh
	SA364	64 K words	16C0000h-16CFFFFh
	SA365	64 K words	16D0000h-16DFFFFh
	SA366	64 K words	16E0000h-16EFFFFh
	SA367	64 K words	16F0000h-16FFFFFh
	SA368	64 K words	1700000h-170FFFFh
	SA369	64 K words	1710000h-171FFFFh
	SA370	64 K words	1720000h-172FFFFh
	SA371	64 K words	1730000h-173FFFFh
	SA372	64 K words	1740000h-174FFFFh
	SA373	64 K words	1750000h-175FFFFh
	SA374	64 K words	1760000h-176FFFFh
	SA375	64 K words	1770000h-177FFFFh
	SA376	64 K words	1780000h-178FFFFh
	SA377	64 K words	1790000h-179FFFFh
	SA378	64 K words	17A0000h-17AFFFFh
	SA379	64 K words	17B0000h-17BFFFFh
	SA380	64 K words	15C0000h-17CFFFFh
	SA381	64 K words	17D0000h-17DFFFFh
	SA382	64 K words	17E0000h-17EFFFFh
	SA383	64 K words	17F0000h-17FFFFFh

Table 6.1 S29NS512P Sector and Memory Address Map (Sheet 7 of 8)

Bank	Sector	Sector Size	Address Range
Bank 12	SA384	64 Kwords	1800000h-180FFFFh
	SA385	64 Kwords	1810000h-181FFFFh
	SA386	64 Kwords	1820000h-182FFFFh
	SA387	64 Kwords	1830000h-183FFFFh
	SA388	64 Kwords	1840000h-184FFFFh
	SA389	64 Kwords	1850000h-185FFFFh
	SA390	64 Kwords	1860000h-186FFFFh
	SA391	64 Kwords	1870000h-187FFFFh
	SA392	64 Kwords	1830000h-188FFFFh
	SA393	64 Kwords	1890000h-189FFFFh
	SA394	64 Kwords	18A0000h-18AFFFFh
	SA395	64 Kwords	18B0000h-18BFFFFh
	SA396	64 Kwords	18C0000h-18CFFFFh
	SA397	64 Kwords	18D0000h-18DFFFFh
	SA398	64 Kwords	18E0000h-18EFFFFh
	SA399	64 Kwords	18F0000h-18FFFFFh
	SA400	64 Kwords	1900000h-190FFFFh
	SA401	64 Kwords	1910000h-191FFFFh
	SA402	64 Kwords	1920000h-192FFFFh
	SA403	64 Kwords	1930000h-193FFFFh
	SA404	64 Kwords	1940000h-194FFFFh
	SA405	64 Kwords	1950000h-195FFFFh
	SA406	64 Kwords	1960000h-196FFFFh
	SA407	64 Kwords	1970000h-197FFFFh
	SA408	64 Kwords	1980000h-198FFFFh
	SA409	64 Kwords	1990000h-199FFFFh
	SA410	64 Kwords	19A0000h-19AFFFFh
	SA411	64 Kwords	19B0000h-19BFFFFh
	SA412	64 Kwords	19C0000h-19CFFFFh
	SA413	64 Kwords	19D0000h-19DFFFFh
	SA414	64 Kwords	19E0000h-19EFFFFh
	SA415	64 Kwords	19F0000h-19FFFFFh

Bank	Sector	Sector Size	Address Range
Bank 13	SA416	64 Kwords	1A00000h-1A0FFFFh
	SA417	64 Kwords	1A10000h-1A1FFFFh
	SA418	64 Kwords	1A20000h-1A2FFFFh
	SA419	64 Kwords	1A30000h-1A3FFFFh
	SA420	64 Kwords	1A40000h-1A4FFFFh
	SA421	64 Kwords	1A50000h-1A5FFFFh
	SA422	64 Kwords	1A60000h-1A6FFFFh
	SA423	64 Kwords	1A70000h-1A7FFFFh
	SA424	64 Kwords	1A30000h-1A8FFFFh
	SA425	64 Kwords	1A90000h-1A9FFFFh
	SA426	64 Kwords	1AA0000h-1AAFFFFh
	SA427	64 Kwords	1AB0000h-1ABFFFFh
	SA428	64 Kwords	1AC0000h-1ACFFFFh
	SA429	64 Kwords	1AD0000h-1ADFFFFh
	SA430	64 Kwords	1AE0000h-1AEFFFFh
	SA431	64 Kwords	1AF0000h-1AFFFFFh
	SA432	64 Kwords	1B00000h-1B0FFFFh
	SA433	64 Kwords	1B10000h-1B1FFFFh
	SA434	64 Kwords	1B20000h-1B2FFFFh
	SA435	64 Kwords	1B30000h-1B3FFFFh
	SA436	64 Kwords	1B40000h-1B4FFFFh
	SA437	64 Kwords	1B50000h-1B5FFFFh
	SA438	64 Kwords	1B60000h-1B6FFFFh
	SA439	64 Kwords	1B70000h-1B7FFFFh
	SA440	64 Kwords	1B80000h-1B8FFFFh
	SA441	64 Kwords	1B90000h-1B9FFFFh
	SA442	64 Kwords	1BA0000h-1BAFFFFh
	SA443	64 Kwords	1BB0000h-1BBFFFFh
	SA444	64 Kwords	1BC0000h-1BCFFFFh
	SA445	64 Kwords	1BD0000h-1BDFFFFh
	SA446	64 Kwords	1BE0000h-1BEFFFFh
	SA447	64 Kwords	1BF0000h-1BFFFFFh

Table 6.1 S29NS512P Sector and Memory Address Map (Sheet 8 of 8)

Bank	Sector	Sector Size	Address Range
Bank 14	SA448	64 Kwords	1C00000h-1C0FFFFh
	SA449	64 Kwords	1C10000h-1C1FFFFh
	SA450	64 Kwords	1C20000h-1C2FFFFh
	SA451	64 Kwords	1C30000h-1C3FFFFh
	SA452	64 Kwords	1C40000h-1C4FFFFh
	SA453	64 Kwords	1C50000h-1C5FFFFh
	SA454	64 Kwords	1C60000h-1C6FFFFh
	SA455	64 Kwords	1C70000h-1C7FFFFh
	SA456	64 Kwords	1C30000h-1C8FFFFh
	SA457	64 Kwords	1C90000h-1C9FFFFh
	SA458	64 Kwords	1CA0000h-1CAFFFFh
	SA459	64 Kwords	1CB0000h-1CBFFFFh
	SA460	64 Kwords	1CC0000h-1CCFFFFh
	SA461	64 Kwords	1CD0000h-1CDFFFFh
	SA462	64 Kwords	1CE0000h-1CEFFFFh
	SA463	64 Kwords	1CF0000h-1CFFFFFh
	SA464	64 Kwords	1D00000h-1D0FFFFh
	SA465	64 Kwords	1D10000h-1D1FFFFh
	SA466	64 Kwords	1D20000h-1D2FFFFh
	SA467	64 Kwords	1D30000h-1D3FFFFh
	SA468	64 Kwords	1D40000h-1D4FFFFh
	SA469	64 Kwords	1D50000h-1D5FFFFh
	SA470	64 Kwords	1D60000h-1D6FFFFh
	SA471	64 Kwords	1D70000h-1D7FFFFh
	SA472	64 Kwords	1D80000h-1D8FFFFh
	SA473	64 Kwords	1D90000h-1D9FFFFh
	SA474	64 Kwords	1DA0000h-1DAFFFFh
	SA475	64 Kwords	1DB0000h-1DBFFFFh
	SA476	64 Kwords	1DC0000h-1DCFFFFh
	SA477	64 Kwords	1DD0000h-1DDFFFFh
	SA478	64 Kwords	1DE0000h-1DEFFFFh
	SA479	64 Kwords	1DF0000h-1DFFFFFh

Bank	Sector	Sector Size	Address Range
Bank 15	SA480	64 K words	1E00000h-1E0FFFFh
	SA481	64 K words	1E10000h-1E1FFFFh
	SA482	64 K words	1E20000h-1E2FFFFh
	SA483	64 K words	1E30000h-1E3FFFFh
	SA484	64 K words	1E40000h-1E4FFFFh
	SA485	64 K words	1E50000h-1E5FFFFh
	SA486	64 K words	1E60000h-1E6FFFFh
	SA487	64 K words	1E70000h-1E7FFFFh
	SA488	64 K words	1E30000h-1E8FFFFh
	SA489	64 K words	1E90000h-1E9FFFFh
	SA490	64 K words	1EA0000h-1EAFFFFh
	SA491	64 K words	1EB0000h-1EBFFFFh
	SA492	64 K words	1EC0000h-1ECFFFFh
	SA493	64 K words	1ED0000h-1EDFFFFh
	SA494	64 K words	1EE0000h-1EEFFFFh
	SA495	64 K words	1EF0000h-1EFFFFFh
	SA496	64 K words	1F00000h-1F0FFFFh
	SA497	64 K words	1F10000h-1F1FFFFh
	SA498	64 K words	1F20000h-1F2FFFFh
	SA499	64 K words	1F30000h-1F3FFFFh
	SA500	64 K words	1F40000h-1F4FFFFh
	SA501	64 K words	1F50000h-1F5FFFFh
	SA502	64 K words	1F60000h-1F6FFFFh
	SA503	64 K words	1F70000h-1F7FFFFh
	SA504	64 K words	1F80000h-1F8FFFFh
	SA505	64 K words	1F90000h-1F9FFFFh
	SA506	64 K words	1FA0000h-1FAFFFFh
	SA507	64 K words	1FB0000h-1FBFFFFh
	SA508	64 K words	1FC0000h-1FCFFFFh
	SA509	64 K words	1FD0000h-1FDFFFFh
	SA510	64 K words	1FE0000h-1FEFFFFh
	SA511	64 K words	1FF0000h-1FFFFFFh

Table 6.2 S29NS256P Sector and Memory Address Map (Sheet 1 of 4)

Bank	Sector	Sector Size	Address Range
Bank 0	SA0	64 Kwords	000000h–00FFFFh
	SA1	64 Kwords	010000h–01FFFFh
	SA2	64 Kwords	020000h–02FFFFh
	SA3	64 Kwords	030000h–03FFFFh
	SA4	64 Kwords	040000h–04FFFFh
	SA5	64 Kwords	050000h–05FFFFh
	SA6	64 Kwords	060000h–06FFFFh
	SA7	64 Kwords	070000h–07FFFFh
	SA8	64 Kwords	080000h–08FFFFh
	SA9	64 Kwords	090000h–09FFFFh
	SA10	64 Kwords	0A0000h–0AFFFFh
	SA11	64 Kwords	0B0000h–0BFFFFh
	SA12	64 Kwords	0C0000h–0CFFFFh
	SA13	64 Kwords	0D0000h–0DFFFFh
	SA14	64 Kwords	0E0000h–0EFFFFh
	SA15	64 Kwords	0F0000h–0FFFFFh
Bank 1	SA16	64 Kwords	100000h–10FFFFh
	SA17	64 Kwords	110000h–11FFFFh
	SA18	64 Kwords	120000h–12FFFFh
	SA19	64 Kwords	130000h–13FFFFh
	SA20	64 Kwords	140000h–14FFFFh
	SA21	64 Kwords	150000h–15FFFFh
	SA22	64 Kwords	160000h–16FFFFh
	SA23	64 Kwords	170000h–17FFFFh
	SA24	64 Kwords	180000h–18FFFFh
	SA25	64 Kwords	190000h–19FFFFh
	SA26	64 Kwords	1A0000h–1AFFFFh
	SA27	64 Kwords	1B0000h–1BFFFFh
	SA28	64 Kwords	1C0000h–1CFFFFh
	SA29	64 Kwords	1D0000h–1DFFFFh
	SA30	64 Kwords	1E0000h–1EFFFFh
	SA31	64 Kwords	1F0000h–1FFFFFh
Bank 2	SA32	64 Kwords	200000h–20FFFFh
	SA33	64 Kwords	210000h–21FFFFh
	SA34	64 Kwords	220000h–22FFFFh
	SA35	64 Kwords	230000h–23FFFFh
	SA36	64 Kwords	240000h–24FFFFh
	SA37	64 Kwords	250000h–25FFFFh
	SA38	64 Kwords	260000h–26FFFFh
	SA39	64 Kwords	270000h–27FFFFh
	SA40	64 Kwords	280000h–28FFFFh
	SA41	64 Kwords	290000h–29FFFFh
	SA42	64 Kwords	2A0000h–2AFFFFh
	SA43	64 Kwords	2B0000h–2BFFFFh
	SA44	64 Kwords	2C0000h–2CFFFFh
	SA45	64 Kwords	2D0000h–2DFFFFh
	SA46	64 Kwords	2E0000h–2EFFFFh
	SA47	64 Kwords	2F0000h–2FFFFFh
Bank 3	SA48	64 Kwords	300000h–30FFFFh
	SA49	64 Kwords	310000h–31FFFFh
	SA50	64 Kwords	320000h–32FFFFh
	SA51	64 Kwords	330000h–33FFFFh
	SA52	64 Kwords	340000h–34FFFFh
	SA53	64 Kwords	350000h–35FFFFh
	SA54	64 Kwords	360000h–36FFFFh
	SA55	64 Kwords	370000h–37FFFFh
	SA56	64 Kwords	380000h–38FFFFh
	SA57	64 Kwords	390000h–39FFFFh
	SA58	64 Kwords	3A0000h–3AFFFFh
	SA59	64 Kwords	3B0000h–3BFFFFh
	SA60	64 Kwords	3C0000h–3CFFFFh
	SA61	64 Kwords	3D0000h–3DFFFFh
	SA62	64 Kwords	3E0000h–3EFFFFh
	SA63	64 Kwords	3F0000h–3FFFFFh

Table 6.2 S29NS256P Sector and Memory Address Map (Sheet 2 of 4)

Bank	Sector	Sector Size	Address Range
Bank 4	SA64	64 Kwords	400000h–40FFFFh
	SA65	64 Kwords	410000h–41FFFFh
	SA66	64 Kwords	420000h–42FFFFh
	SA67	64 Kwords	430000h–43FFFFh
	SA68	64 Kwords	440000h–44FFFFh
	SA69	64 Kwords	450000h–45FFFFh
	SA70	64 Kwords	460000h–46FFFFh
	SA71	64 Kwords	470000h–47FFFFh
	SA72	64 Kwords	480000h–48FFFFh
	SA73	64 Kwords	490000h–49FFFFh
	SA74	64 Kwords	4A0000h–4AFFFFh
	SA75	64 Kwords	4B0000h–4BFFFFh
	SA76	64 Kwords	4C0000h–4CFFFFh
	SA77	64 Kwords	4D0000h–4DFFFFh
	SA78	64 Kwords	4E0000h–4EFFFFh
	SA79	64 Kwords	4F0000h–4FFFFFh
Bank 5	SA80	64 Kwords	500000h–50FFFFh
	SA81	64 Kwords	510000h–51FFFFh
	SA82	64 Kwords	520000h–52FFFFh
	SA83	64 Kwords	530000h–53FFFFh
	SA84	64 Kwords	540000h–54FFFFh
	SA85	64 Kwords	550000h–55FFFFh
	SA86	64 Kwords	560000h–56FFFFh
	SA87	64 Kwords	570000h–57FFFFh
	SA88	64 Kwords	580000h–58FFFFh
	SA89	64 Kwords	590000h–59FFFFh
	SA90	64 Kwords	5A0000h–5AFFFFh
	SA91	64 Kwords	5B0000h–5BFFFFh
	SA92	64 Kwords	5C0000h–5CFFFFh
	SA93	64 Kwords	5D0000h–5DFFFFh
	SA94	64 Kwords	5E0000h–5EFFFFh
	SA95	64 Kwords	5F0000h–5FFFFFh
Bank 6	SA96	64 K words	600000h–60FFFFh
	SA97	64 K words	610000h–61FFFFh
	SA98	64 K words	620000h–62FFFFh
	SA99	64 K words	630000h–63FFFFh
	SA100	64 K words	640000h–64FFFFh
	SA101	64 K words	650000h–65FFFFh
	SA102	64 K words	660000h–66FFFFh
	SA103	64 K words	670000h–67FFFFh
	SA104	64 K words	680000h–68FFFFh
	SA105	64 K words	690000h–69FFFFh
	SA106	64 K words	6A0000h–6AFFFFh
	SA107	64 K words	6B0000h–6BFFFFh
	SA108	64 K words	6C0000h–6CFFFFh
	SA109	64 K words	6D0000h–6DFFFFh
	SA110	64 K words	6E0000h–6EFFFFh
	SA111	64 K words	6F0000h–6FFFFFh
Bank 7	SA112	64 K words	700000h–70FFFFh
	SA113	64 K words	710000h–71FFFFh
	SA114	64 K words	720000h–72FFFFh
	SA115	64 K words	730000h–73FFFFh
	SA116	64 K words	740000h–74FFFFh
	SA117	64 K words	750000h–75FFFFh
	SA118	64 K words	760000h–76FFFFh
	SA119	64 K words	770000h–77FFFFh
	SA120	64 K words	780000h–78FFFFh
	SA121	64 K words	790000h–79FFFFh
	SA122	64 K words	7A0000h–7AFFFFh
	SA123	64 K words	7B0000h–7BFFFFh
	SA124	64 K words	7C0000h–7CFFFFh
	SA125	64 K words	7D0000h–7DFFFFh
	SA126	64 K words	7E0000h–7EFFFFh
	SA127	64 K words	7F0000h–7FFFFFh

Table 6.2 S29NS256P Sector and Memory Address Map (Sheet 3 of 4)

Bank	Sector	Sector Size	Address Range
Bank 8	SA128	64 Kwords	800000h–80FFFFh
	SA129	64 Kwords	810000h–81FFFFh
	SA130	64 Kwords	820000h–82FFFFh
	SA131	64 Kwords	830000h–83FFFFh
	SA132	64 Kwords	840000h–84FFFFh
	SA133	64 Kwords	850000h–85FFFFh
	SA134	64 Kwords	860000h–86FFFFh
	SA135	64 Kwords	870000h–87FFFFh
	SA136	64 Kwords	880000h–88FFFFh
	SA137	64 Kwords	890000h–89FFFFh
	SA138	64 Kwords	8A0000h–8AFFFFh
	SA139	64 Kwords	8B0000h–8BFFFFh
	SA140	64 Kwords	8C0000h–8CFFFFh
	SA141	64 Kwords	8D0000h–8DFFFFh
	SA142	64 Kwords	8E0000h–8EFFFFh
	SA143	64 Kwords	8F0000h–8FFFFFh
Bank 9	SA144	64 Kwords	900000h–90FFFFh
	SA145	64 Kwords	910000h–91FFFFh
	SA146	64 Kwords	920000h–92FFFFh
	SA147	64 Kwords	930000h–93FFFFh
	SA148	64 Kwords	940000h–94FFFFh
	SA149	64 Kwords	950000h–95FFFFh
	SA150	64 Kwords	960000h–96FFFFh
	SA151	64 Kwords	970000h–97FFFFh
	SA152	64 Kwords	980000h–98FFFFh
	SA153	64 Kwords	990000h–99FFFFh
	SA154	64 Kwords	9A0000h–9AFFFFh
	SA155	64 Kwords	9B0000h–9BFFFFh
	SA156	64 Kwords	9C0000h–9CFFFFh
	SA157	64 Kwords	9D0000h–9DFFFFh
	SA158	64 Kwords	9E0000h–9EFFFFh
	SA159	64 Kwords	9F0000h–9FFFFFh
Bank 10	SA160	64 Kwords	A00000h–A0FFFFh
	SA161	64 Kwords	A10000h–A1FFFFh
	SA162	64 Kwords	A20000h–A2FFFFh
	SA163	64 Kwords	A30000h–A3FFFFh
	SA164	64 Kwords	A40000h–A4FFFFh
	SA165	64 Kwords	A50000h–A5FFFFh
	SA166	64 Kwords	A60000h–A6FFFFh
	SA167	64 Kwords	A70000h–A7FFFFh
	SA168	64 Kwords	A80000h–A8FFFFh
	SA169	64 Kwords	A90000h–A9FFFFh
	SA170	64 Kwords	AA0000h–AAFFFFh
	SA171	64 Kwords	AB0000h–ABFFFFh
	SA172	64 Kwords	AC0000h–ACFFFFh
	SA173	64 Kwords	AD0000h–ADFFFFh
	SA174	64 Kwords	AE0000h–AEFFFFh
	SA175	64 Kwords	AF0000h–AFFFFFh
Bank 11	SA176	64 Kwords	B00000h–B0FFFFh
	SA177	64 Kwords	B10000h–B1FFFFh
	SA178	64 Kwords	B20000h–B2FFFFh
	SA179	64 Kwords	B30000h–B3FFFFh
	SA180	64 Kwords	B40000h–B4FFFFh
	SA181	64 Kwords	B50000h–B5FFFFh
	SA182	64 Kwords	B60000h–B6FFFFh
	SA183	64 Kwords	B70000h–B7FFFFh
	SA184	64 Kwords	B80000h–B8FFFFh
	SA185	64 Kwords	B90000h–B9FFFFh
	SA186	64 Kwords	BA0000h–BAFFFFh
	SA187	64 Kwords	BB0000h–BBFFFFh
	SA188	64 Kwords	BC0000h–BCFFFFh
	SA189	64 Kwords	BD0000h–BDFFFFh
	SA190	64 Kwords	BE0000h–BEFFFFh
	SA191	64 Kwords	BF0000h–BFFFFFh

Table 6.2 S29NS256P Sector and Memory Address Map (Sheet 4 of 4)

Bank	Sector	Sector Size	Address Range
Bank 12	SA192	64 Kwords	C00000h–C0FFFFh
	SA193	64 Kwords	C10000h–C1FFFFh
	SA194	64 Kwords	C20000h–C2FFFFh
	SA195	64 Kwords	C30000h–C3FFFFh
	SA196	64 Kwords	C40000h–C4FFFFh
	SA197	64 Kwords	C50000h–C5FFFFh
	SA198	64 Kwords	C60000h–C6FFFFh
	SA199	64 Kwords	C70000h–C7FFFFh
	SA200	64 Kwords	C80000h–C8FFFFh
	SA201	64 Kwords	C90000h–C9FFFFh
	SA202	64 Kwords	CA0000h–CAFFFFh
	SA203	64 Kwords	CB0000h–CBFFFFh
	SA204	64 Kwords	CC0000h–CCFFFFh
	SA205	64 Kwords	CD0000h–CDFFFFh
	SA206	64 Kwords	CE0000h–CEFFFFh
	SA207	64 Kwords	CF0000h–CFFFFFh
Bank 13	SA208	64 Kwords	D00000h–D0FFFFh
	SA209	64 Kwords	D10000h–D1FFFFh
	SA210	64 Kwords	D20000h–D2FFFFh
	SA211	64 Kwords	D30000h–D3FFFFh
	SA212	64 Kwords	D40000h–D4FFFFh
	SA213	64 Kwords	D50000h–D5FFFFh
	SA214	64 Kwords	D60000h–D6FFFFh
	SA215	64 Kwords	D70000h–D7FFFFh
	SA216	64 Kwords	D80000h–D8FFFFh
	SA217	64 Kwords	D90000h–D9FFFFh
	SA218	64 Kwords	DA0000h–DAFFFFh
	SA219	64 Kwords	DB0000h–DBFFFFh
	SA220	64 Kwords	DC0000h–DCFFFFh
	SA221	64 Kwords	DD0000h–DDFFFFh
	SA222	64 Kwords	DE0000h–DEFFFFh
	SA223	64 Kwords	DF0000h–DFFFFFh
Bank 14	SA224	64 K words	E00000h–E0FFFFh
	SA225	64 K words	E10000h–E1FFFFh
	SA226	64 K words	E20000h–E2FFFFh
	SA227	64 K words	E30000h–E3FFFFh
	SA228	64 K words	E40000h–E4FFFFh
	SA229	64 K words	E50000h–E5FFFFh
	SA230	64 K words	E60000h–E6FFFFh
	SA231	64 K words	E70000h–E7FFFFh
	SA232	64 K words	E80000h–E8FFFFh
	SA233	64 K words	E90000h–E9FFFFh
	SA234	64 K words	EA0000h–EAFFFFh
	SA235	64 K words	EB0000h–EBFFFFh
	SA236	64 K words	EC0000h–ECFFFFh
	SA237	64 K words	ED0000h–EDFFFFh
	SA238	64 K words	EE0000h–EEFFFFh
	SA239	64 K words	EF0000h–EFFFFFh
Bank 15	SA240	64 K words	F00000h–F0FFFFh
	SA241	64 K words	F10000h–F1FFFFh
	SA242	64 K words	F20000h–F2FFFFh
	SA243	64 K words	F30000h–F3FFFFh
	SA244	64 K words	F40000h–F4FFFFh
	SA245	64 K words	F50000h–F5FFFFh
	SA246	64 K words	F60000h–F6FFFFh
	SA247	64 K words	F70000h–F7FFFFh
	SA248	64 K words	F80000h–F8FFFFh
	SA249	64 K words	F90000h–F9FFFFh
	SA250	64 K words	FA0000h–FAFFFFh
	SA251	64 K words	FB0000h–FBFFFFh
	SA252	64 K words	FC0000h–FCFFFFh
	SA253	64 K words	FD0000h–FDFFFFh
	SA254	64 K words	FE0000h–FEFFFFh
	SA255	16 K words	FF0000h–FF3FFFh
	SA256	16 K words	FF4000h–FF7FFFh
	SA257	16 K words	FF8000h–FFBFFFh
	SA258	16 K words	FFC000h–FFFFFFh

Table 6.3 S29NS128P Sector & Memory Address Map (Sheet 1 of 2)

Bank	Sector	Sector Size	Address Range
Bank 0	SA0	64 Kwords	000000h–00FFFFh
	SA1	64 Kwords	010000h–01FFFFh
	SA2	64 Kwords	020000h–02FFFFh
	SA3	64 Kwords	030000h–03FFFFh
	SA4	64 Kwords	040000h–04FFFFh
	SA5	64 Kwords	050000h–05FFFFh
	SA6	64 Kwords	060000h–06FFFFh
	SA7	64 Kwords	070000h–07FFFFh
Bank 1	SA8	64 Kwords	080000h–08FFFFh
	SA9	64 Kwords	090000h–09FFFFh
	SA10	64 Kwords	0A0000h–0AFFFFh
	SA11	64 Kwords	0B0000h–0BFFFFh
	SA12	64 Kwords	0C0000h–0CFFFFh
	SA13	64 Kwords	0D0000h–0DFFFFh
	SA14	64 Kwords	0E0000h–0EFFFFh
	SA15	64 Kwords	0F0000h–0FFFFFh
Bank 2	SA16	64 Kwords	100000h–10FFFFh
	SA17	64 Kwords	110000h–11FFFFh
	SA18	64 Kwords	120000h–12FFFFh
	SA19	64 Kwords	130000h–13FFFFh
	SA20	64 Kwords	140000h–14FFFFh
	SA21	64 Kwords	150000h–15FFFFh
	SA22	64 Kwords	160000h–16FFFFh
	SA23	64 Kwords	170000h–17FFFFh
Bank 3	SA24	64 Kwords	180000h–18FFFFh
	SA25	64 Kwords	190000h–19FFFFh
	SA26	64 Kwords	1A0000h–1AFFFFh
	SA27	64 Kwords	1B0000h–1BFFFFh
	SA28	64 Kwords	1C0000h–1CFFFFh
	SA29	64 Kwords	1D0000h–1DFFFFh
	SA30	64 Kwords	1E0000h–1EFFFFh
	SA31	64 Kwords	1F0000h–1FFFFFh
Bank 8	SA64	64 Kwords	400000h–40FFFFh
	SA65	64 Kwords	410000h–41FFFFh
	SA66	64 Kwords	420000h–42FFFFh
	SA67	64 Kwords	430000h–43FFFFh
	SA68	64 Kwords	440000h–44FFFFh
	SA69	64 Kwords	450000h–45FFFFh
	SA70	64 Kwords	460000h–46FFFFh
	SA71	64 Kwords	470000h–47FFFFh
Bank 4	SA32	64 Kwords	200000h–20FFFFh
	SA33	64 Kwords	210000h–21FFFFh
	SA34	64 Kwords	220000h–22FFFFh
	SA35	64 Kwords	230000h–23FFFFh
	SA36	64 Kwords	240000h–24FFFFh
	SA37	64 Kwords	250000h–25FFFFh
	SA38	64 Kwords	260000h–26FFFFh
	SA39	64 Kwords	270000h–27FFFFh
Bank 5	SA40	64 Kwords	280000h–28FFFFh
	SA41	64 Kwords	290000h–29FFFFh
	SA42	64 Kwords	2A0000h–2AFFFFh
	SA43	64 Kwords	2B0000h–2BFFFFh
	SA44	64 Kwords	2C0000h–2CFFFFh
	SA45	64 Kwords	2D0000h–2DFFFFh
	SA46	64 Kwords	2E0000h–2EFFFFh
	SA47	64 Kwords	2F0000h–2FFFFFh
Bank 6	SA48	64 Kwords	300000h–30FFFFh
	SA49	64 Kwords	310000h–31FFFFh
	SA50	64 Kwords	320000h–32FFFFh
	SA51	64 Kwords	330000h–33FFFFh
	SA52	64 Kwords	340000h–34FFFFh
	SA53	64 Kwords	350000h–35FFFFh
	SA54	64 Kwords	360000h–36FFFFh
	SA55	64 Kwords	370000h–37FFFFh
Bank 7	SA56	64 Kwords	380000h–38FFFFh
	SA57	64 Kwords	390000h–39FFFFh
	SA58	64 Kwords	3A0000h–3AFFFFh
	SA59	64 Kwords	3B0000h–3BFFFFh
	SA60	64 Kwords	3C0000h–3CFFFFh
	SA61	64 Kwords	3D0000h–3DFFFFh
	SA62	64 Kwords	3E0000h–3EFFFFh
	SA63	64 Kwords	3F0000h–3FFFFFh
Bank 12	SA96	64 K words	600000h–60FFFFh
	SA97	64 K words	610000h–61FFFFh
	SA98	64 K words	620000h–62FFFFh
	SA99	64 K words	630000h–63FFFFh
	SA100	64 K words	640000h–64FFFFh
	SA101	64 K words	650000h–65FFFFh
	SA102	64 K words	660000h–66FFFFh
	SA103	64 K words	670000h–67FFFFh

Table 6.3 S29NS128P Sector & Memory Address Map (Sheet 2 of 2)

Bank	Sector	Sector Size	Address Range
Bank 9	SA72	64 Kwords	480000h–48FFFFh
	SA73	64 Kwords	490000h–49FFFFh
	SA74	64 Kwords	4A0000h–4AFFFFh
	SA75	64 Kwords	4B0000h–4BFFFFh
	SA76	64 Kwords	4C0000h–4CFFFFh
	SA77	64 Kwords	4D0000h–4DFFFFh
	SA78	64 Kwords	4E0000h–4EFFFFh
	SA79	64 Kwords	4F0000h–4FFFFFh
Bank 10	SA80	64 Kwords	500000h–50FFFFh
	SA81	64 Kwords	510000h–51FFFFh
	SA82	64 Kwords	520000h–52FFFFh
	SA83	64 Kwords	530000h–53FFFFh
	SA84	64 Kwords	540000h–54FFFFh
	SA85	64 Kwords	550000h–55FFFFh
	SA86	64 Kwords	560000h–56FFFFh
	SA87	64 Kwords	570000h–57FFFFh
Bank 11	SA88	64 Kwords	580000h–58FFFFh
	SA89	64 Kwords	590000h–59FFFFh
	SA90	64 Kwords	5A0000h–5AFFFFh
	SA91	64 Kwords	5B0000h–5BFFFFh
	SA92	64 Kwords	5C0000h–5CFFFFh
	SA93	64 Kwords	5D0000h–5DFFFFh
	SA94	64 Kwords	5E0000h–5EFFFFh
	SA95	64 Kwords	5F0000h–5FFFFFh
Bank 13	SA104	64 K words	680000h–68FFFFh
	SA105	64 K words	690000h–69FFFFh
	SA106	64 K words	6A0000h–6AFFFFh
	SA107	64 K words	6B0000h–6BFFFFh
	SA108	64 K words	6C0000h–6CFFFFh
	SA109	64 K words	6D0000h–6DFFFFh
	SA110	64 K words	6E0000h–6EFFFFh
	SA111	64 K words	6F0000h–6FFFFFh
Bank 14	SA112	64 K words	700000h–70FFFFh
	SA113	64 K words	710000h–71FFFFh
	SA114	64 K words	720000h–72FFFFh
	SA115	64 K words	730000h–73FFFFh
	SA116	64 K words	740000h–74FFFFh
	SA117	64 K words	750000h–75FFFFh
	SA118	64 K words	760000h–76FFFFh
	SA119	64 K words	770000h–77FFFFh
Bank 15	SA120	64 K words	780000h–78FFFFh
	SA121	64 K words	790000h–79FFFFh
	SA122	64 K words	7A0000h–7AFFFFh
	SA123	64 K words	7B0000h–7BFFFFh
	SA124	64 K words	7C0000h–7CFFFFh
	SA125	64 K words	7D0000h–7DFFFFh
	SA126	64 K words	7E0000h–7EFFFFh
	SA127	16 K words	7F0000h–7F3FFFh
	SA128	16 K words	7F4000h–7F7FFFh
	SA129	16 K words	7F8000h–7FBFFFh
	SA130	16 K words	7FC000h–7FFFFFh

7. Device Operations

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

Operations are initiated by writing specific commands or a sequence with specific address and data patterns into the command registers (see Tables 12.1 and 12.2). The command register itself does not occupy any addressable memory location; rather, it is composed of latches that store the commands, along with the address and data information needed to execute the command. The contents of the register serve as input to the internal state machine and the state machine outputs dictate the function of the device. Writing incorrect address and data values or writing them in an improper sequence may place the device in an unknown state, in which case the system must write the reset command to return the device to the reading array data mode.

7.1 Device Operation Table

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

Table 7.1 Device Operations

Operation	CE#	OE#	WE#	CLK	AVD#	Amax–A16	A/DQ15–A/DQ0	RDY	RESET#
Asynchronous Read – Addresses Latched	L	L	H	X	↑	Addr In	I/O	H	H
Asynchronous Write	L	H	↑	X	↑	Addr In	I/O	H	H
Standby (CE#)	H	X	X	X	X	X	High-Z	High-Z	H
Hardware Reset	X	X	X	X	X	X	High-Z	High-Z	↓
Burst Read Operations									
Latch Starting Burst Address by CLK	L	H	H	↑	L	Addr In	Addr In	X	H
Advance Burst read to next address	L	L	H	↑	H	X	I/O	H	H
Terminate current Burst read cycle	H	X	H	X	X	X	High-Z	High-Z	H
Terminate current Burst read cycle via RESET#	X	X	H	X	X	X	High-Z	High-Z	L
Terminate current Burst read cycle and start new Burst read cycle	L	X	H	↑	↓	Addr In	Addr In	X	H

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

Notes
1. *Address is latched on the rising edge of clock.*
2. *CLK must stay low or high after CE# goes low when device in Asynchronous Read mode.*

7.2 Asynchronous Read

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

To read data from the memory array, the system must first assert a valid address while driving AVD# and CE# to V_{IL}. WE# must remain at V_{IH}. The rising edge of AVD# latches the address. The OE# signal must be driven to V_{IL}, once AVD# has been driven to V_{IH}.

The data is output on A/DQ15 – A/DQ0 pins after the access time (t_{OE}) has elapsed from the falling edge of OE#.

7.3 Synchronous (Burst) Read Operation

The device is capable of continuous sequential burst operation and linear burst operation of a preset length. When the device first powers up, it is enabled for Asynchronous read and can be automatically enabled for burst mode and the address is latched on the first rising edge of CLK input, while AVD# is held low for one clock cycle.

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

At startup the system writes the *Set Configuration Register* command sequence to optimize the system performance.

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

Note that the device has a fixed internal address boundary that occurs every 128 words. No boundary crossing latency is required when the device operates with wait states set from 2 to 9.

7.3.1 Latency Tables for Variable Wait State

Tables 7.2 – 7.9 show the latency for variable wait state in a normal Burst operation.

Table 7.2 Address Latency for 9 Wait States

Word	Initial Wait									
0	9 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1		D1	D2	D3	D4	D5	D6	D7	**1 ws**	D8
2		D2	D3	D4	D5	D6	D7	**1 ws**	**1 ws**	D8
3		D3	D4	D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	D8
4		D4	D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8
5		D5	D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8
6		D6	D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8
7		D7	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	**1 ws**	D8

Table 7.3 Address Latency for 8 Wait States

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

Table 7.4 Address Latency for 7 Wait States

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

Table 7.5 Address Latency for 6 Wait States

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

Table 7.6 Address Latency for 5 Wait States

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

Table 7.7 Address Latency for 4 Wait States

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

Table 7.8 Address Latency for 3 Wait States

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

Table 7.9 Address Latency for 2 Wait States

Word	Initial Wait									
0	2 ws	D0	D1	D2	D3	D4	D5	D6	D7	D8
1		D1	D2	D3	D4	D5	D6	D7	D8	D9
2		D2	D3	D4	D5	D6	D7	D8	D9	D10
3		D3	D4	D5	D6	D7	D8	D9	D10	D11
4		D4	D5	D6	D7	D8	D9	D10	D11	D12
5		D5	D6	D7	D8	D9	D10	D11	D12	D13
6		D6	D7	D8	D9	D10	D11	D12	D13	D14
7		D7	D8	D9	D10	D11	D12	D13	D14	D15

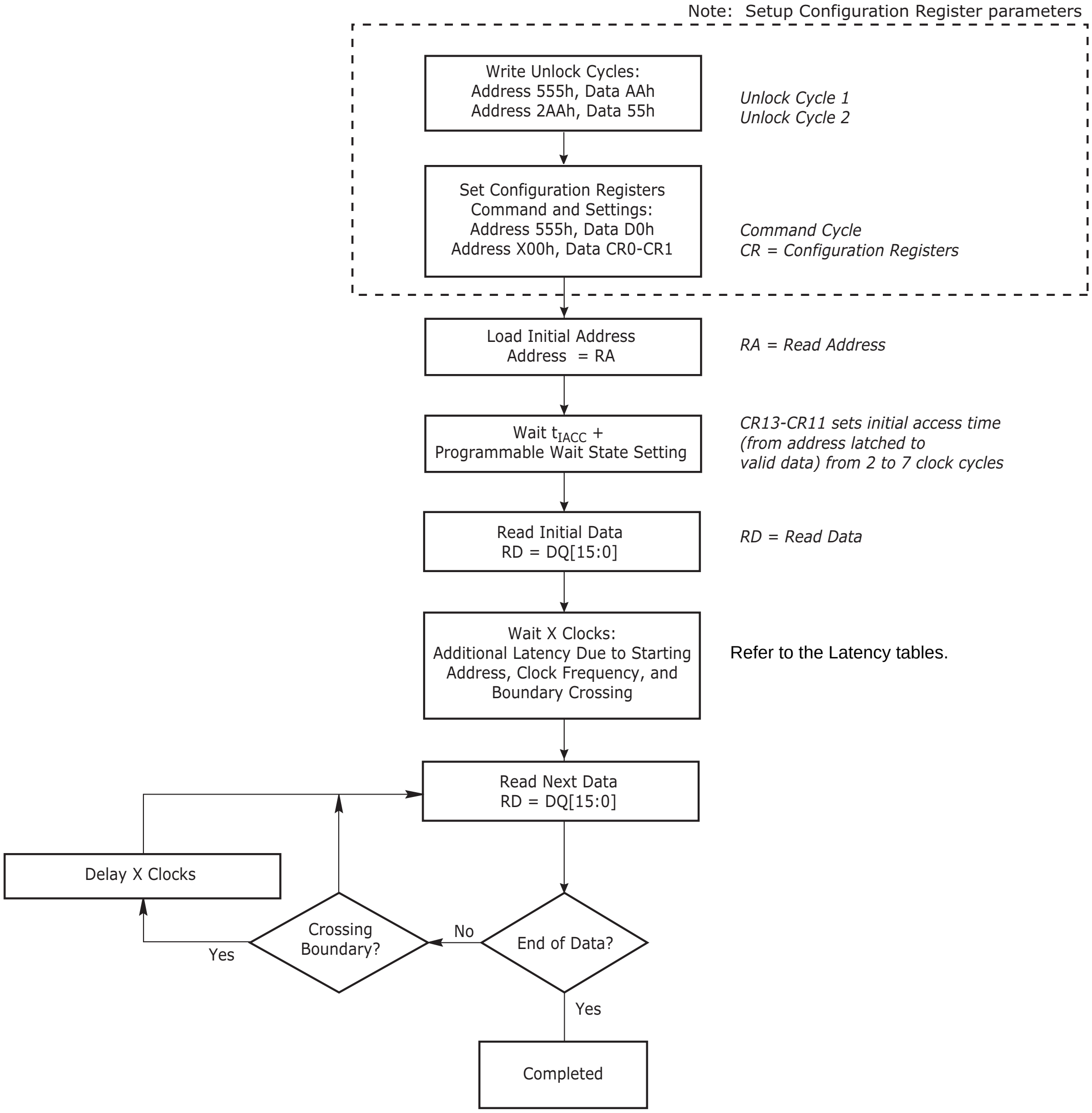


Figure 7.1 Synchronous Read Flow Chart

7.3.2 Continuous Burst Read Mode

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

If the address being read crosses a 128-word line boundary within the same bank, but not into a program or erase suspended sector, as mentioned above, additional latency cycles are required as reflected by the configuration register table (Table 7.11) and Tables 7.2 – 7.9.

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

7.3.3 8-Word, 16-Word, and 32-Word Linear Burst Read with Wrap Around

In a linear burst read operation, a fixed number of words (8, 16, or 32 words) are read from consecutive addresses that are determined by the group within which the starting address falls. The groups are sized according to the number of words read in a single burst sequence for a given mode (see Table 7.10).

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

In a similar fashion, the 16-word and 32-word Linear Wrap modes begin their burst sequence on the starting address provided to the device, then wrap back to the first address in the selected address group.

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

Table 7.10 Burst Address Groups

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

7.3.4 8-Word, 16-Word, and 32-Word Linear Burst without Wrap Around

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

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

7.3.5 Configuration Registers

This device uses two 16-bit configuration registers to set various operational parameters. Upon power-up or hardware reset, the device is capable of the asynchronous read mode and synchronous read, and the configuration register settings are in their default state. The host system should determine the proper settings for the entire configuration register, and then execute the Set Configuration Register command sequence before attempting burst operations. The Configuration Register can also be read using a command sequence (see Table 12.1). The following list describes the register settings.

Table 7.11 Configuration Register

CR Bit	Function	Settings (Binary)
CR0.15	Reserved (Not used)	0 = Reserved (Default) 1 = Reserved
CR0.14	Reserved (Not used)	0 = Reserved (Default) 1 = Reserved

Table 7.11 Configuration Register

CR Bit	Function	Settings (Binary)					
CR1.0	Programmable Wait State (Note 1)	0000	=	initial data is valid on the	2nd	rising CLK edge	AVD# transition to V_{IH}
		0001			3rd		
CR0.13		0010			4th		
		0011			5th		
CR0.12		0100			6th		
		0101			7th		
CR0.11		0110	=	Reserved			
		0111					
		1000	=	initial data is valid on the	8th	rising CLK edge	AVD# transition to V_{IH}
		1001			9th		
		⋮					
		1101	=	initial data is valid on the	13th	rising CLK edge	AVD# transition to V_{IH} (Default)
		1110	=	Reserved			
		1111					
CR0.10	RDY Polarity	0 = RDY signal is active low 1 = RDY signal is active high (Default)					
CR0.9	Reserved (Not used)	0 = Reserved 1 = Reserved (Default)					
CR0.8	RDY	0 = RDY active one clock cycle before data 1 = RDY active with data (Default)					
CR0.7	Reserved (Not used)	0 = Reserved 1 = Reserved (Default)					
CR0.6	Reserved (Not used)	0 = Reserved 1 = Reserved (Default)					
CR0.5	Reserved (Not used)	0 = Reserved (Default) 1 = Reserved					
CR1.4	Output Drive Strength	0 = Full Drive= Current Driver Strength (Default) 1 = Half Drive					
CR0.4	RDY Function	0 = RDY (Default) 1 = Reserved					
CR0.3	Burst Wrap Around	0 = No Wrap Around Burst 1 = Wrap Around Burst (Default)					
CR0.2	Burst Length	000 = Continuous (Default)					
		010 = 8-Word Linear Burst					
CR0.1		011 = 16-Word Linear Burst					
CR0.0		100 = 32-Word Linear Burst					
		(All other bit settings are reserved)					

Notes

1. *The addresses are latched by rising edge of CLK.*
2. *CR1.0 to CR1.3 and CR1.5 to CR1.15 = 1 (Default).*
3. *A software reset command is required after read command.*
4. *CR0.3 is ignored if in continuous read mode (no wrap around).*

7.4 Autoselect

The Autoselect is used for manufacturer ID, Device identification, and sector protection information. This mode is primarily intended for programming equipment to automatically match a device with its corresponding programming algorithm. The Autoselect codes can also be accessed in the system. When verifying sector protection, the sector address must appear on the appropriate highest order address bits (see Table 7.12). The remaining address bits are don't care. The most significant four bits of the address during the third write cycle select the bank from which the Autoselect codes are read by the host. All other banks can be accessed normally for data read without exiting the Autoselect mode.

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

See Table 12.1 for command sequence details.

Table 7.12 Autoselect Addresses

Description	Address	Read Data
Manufacturer ID Byte 00	(BA) + 00h	0001h
Device ID, Byte 01	(BA) + 01h	307Eh (NS512P) 317Eh (NS256P) 327Eh (NS128P)
Sector Lock/Unlock Byte 02	(SA) + 02h	0001h = Locked, 0000h = Unlocked
Indicator Bits Byte 07	(BA) + 07h	DQ15 – DQ8 = reserved DQ7 – Factory Lock Bit; 1 = Locked, 0 = Not Locked DQ6 – Customer Lock Bit; 1 = Locked, 0 = Not Locked DQ5 – Handshake Bit; 1 = Reserved, 0 = Standard Handshake DQ4 and DQ3 – WP# Protection Boot Code; 01 = WP# Protects Top Boot Sectors, DQ2 – DQ0 = reserved
Device ID, Byte 0E	(BA) + 0Eh	303Fh (NS512P) 3141h (NS256P) 3243h (NS128P)
Device ID, Byte 0F	(BA) + 0Fh	3000h (NS512P) 3100h (NS256P) 3200h (NS128P)

Software Functions and Sample Code

Table 7.13 Autoselect Entry

(LLD Function = lld_AutoselectEntryCmd)

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

Table 7.14 Autoselect Exit

(LLD Function = lld_AutoselectExitCmd)

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

Notes

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

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

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

UINT16 manuf_id;

/* Auto Select Entry */

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


/* multiple reads can be performed after entry */

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

/*   Autoselect exit */

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

7.5 Program/Erase Operations

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

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

All addresses are latched on the rising edge of AVD# or falling edge of WE#, and all data is latched on the first rising edge of WE#.

Note the following:

- When the Embedded Program/Erase algorithm is complete, the device returns to the read mode.
- The system can determine the status of the Program/Erase operation. Refer to the Write Operation Status section for further information.
- While *1* can be programmed to *0*, a *0* cannot be programmed to a *1*. Any such attempt is ignored as only an erase operation can covert a *0* to a *1*.
- Any commands written to the device during the Embedded Program/Erase Algorithm are ignored except the Program/Erase Suspend command.
- Secured Silicon Sector, Autoselect, and CFI functions are unavailable when a program operation is in progress.
- A hardware reset or power removal immediately terminates the Program/Erase operation and the Program/Erase command sequence should be reinitiated once the device has returned to the read mode, to ensure data integrity.
- Programming is allowed in any sequence and across sector boundaries only for single word programming operation. See Section 7.5.2, *Write Buffer Programming* on page 37 when using the write buffer.

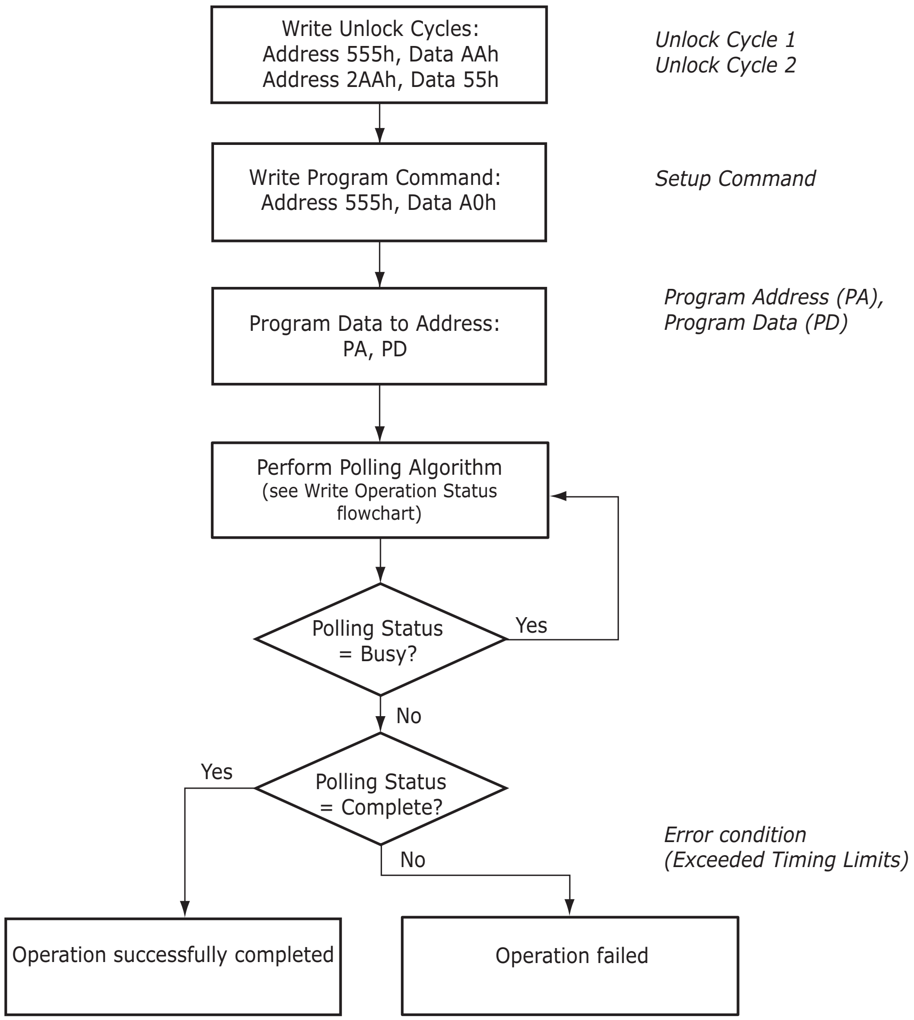
Note: The system may also lock or unlock any sector while the erase operation is suspended.

7.5.1 Single Word Programming

Single word programming mode is the simplest method of programming. In this mode, four Flash command write cycles are used to program an individual Flash address. While the single word programming method is supported by all Spansion devices, in general it is not recommended for devices that support Write Buffer Programming. See Table 12.1 for the required bus cycles and Figure 7.2 for the flowchart.

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

Figure 7.2 Single Word Program



Software Functions and Sample Code

Table 7.15 Single Word Program

(LLD Function = lld_ProgramCmd)

Cycle	Operation	Byte Address	Word Address	Data
Unlock Cycle 1	Write	Base + AAAh	Base + 555h	00AAh
Unlock Cycle 2	Write	Base + 554h	Base + 2AAh	0055h
Program Setup	Write	Base + AAAh	Base + 555h	00A0h
Program	Write	Word Address	Word Address	Data Word

Note
Base = Base Address.

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

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

7.5.2 Write Buffer Programming

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

The system then writes the starting address/data combination. This starting address is the first address/data pair to be programmed, and selects the *write-buffer-page* address. All subsequent address/data pairs must be in sequential order.

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

After writing the Starting Address/Data pair, the system then writes the remaining address/data pairs into the write buffer. Write buffer locations must be loaded in sequential order starting with the lowest address in the page. Note that if the number of address/data pairs do not match the word count, the program buffer to flash command is ignored.

Note that if a Write Buffer address location is loaded multiple times, the *address/data pair* counter decrements for every data load operation. Also, the last data loaded at a location before the *Program Buffer to Flash* confirm command is programmed into the device. It is the software's responsibility to comprehend ramifications of loading a write-buffer location more than once. The counter decrements for each data load operation, NOT for each unique write-buffer-address location.

Once the specified number of write buffer locations have been loaded, the system must then write the *Program Buffer to Flash* command at the Sector Address. Any other address/data write combinations abort the Write Buffer Programming operation. The device then *goes busy*. The Data Bar polling techniques should be used while monitoring the last address location loaded into the write buffer. This eliminates the need to store an address in memory because the system can load the last address location, issue the program confirm command at the last loaded address location, and then data bar poll at that same address. DQ7, DQ6, DQ5, DQ2, and DQ1 should be monitored to determine the device status during Write Buffer Programming.

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

The Write Buffer Programming Sequence is ABORTED in the following ways:

- Load a value that is greater than the buffer size during the *Number of Locations to Program* step (DQ7 is not valid in this condition).
- Write to an address in a sector different than the one specified during the *Write-Buffer-Load* command.
- Write an Address/Data pair to a different write-buffer-page than the one selected by the *Starting Address* during the *write buffer data loading* stage of the operation.
- Write data other than the *Confirm Command* after the specified number of *data load* cycles.

Software Functions and Sample Code

Table 7.16 Write Buffer Program

(LLD Functions Used = lld_WriteToBufferCmd, lld_ProgramBufferToFlashCmd)

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

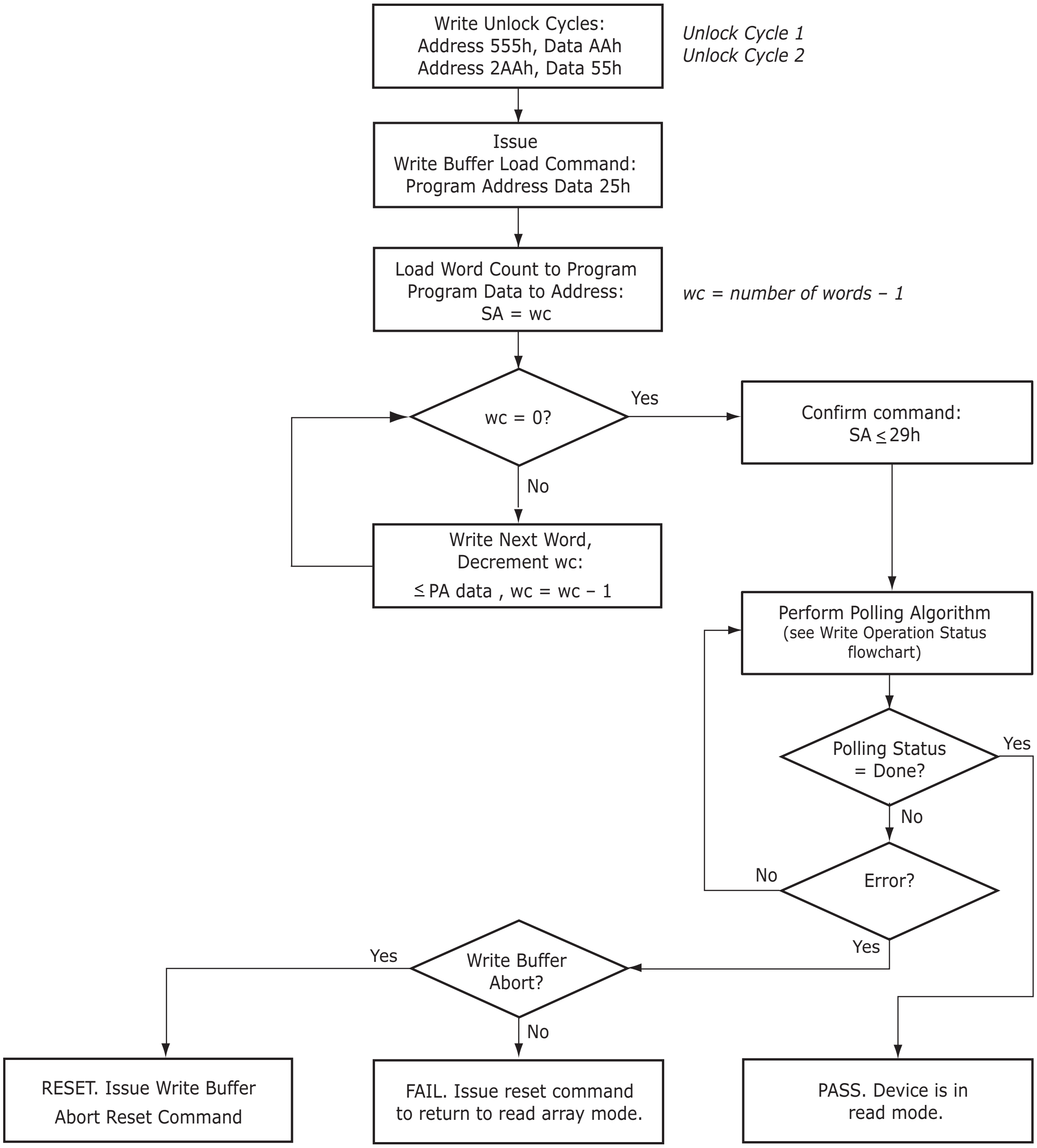
Notes

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

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

```
/* Example: Write Buffer Programming Command */
/* NOTES: Write buffer programming limited to 16 words. */
/* All addresses to be written to the flash in */
/* one operation must be within the same write buffer. */
/* A write buffer begins at addresses evenly divisible */
/* by 0x20.
UINT16 i; */
UINT16 *src = source_of_data; /* address of source data */
UINT16 *dst = destination_of_data; /* flash destination address */
UINT16 wc = words_to_program -1; /* word count (minus 1) */
*( (UINT16 *)base_addr + 0x555 ) = 0x00AA; /* write unlock cycle 1 */
*( (UINT16 *)base_addr + 0x2AA ) = 0x0055; /* write unlock cycle 2 */
*( (UINT16 *)dst ) = 0x0025; /* write write buffer load command */
*( (UINT16 *)dst ) = wc; /* write word count (minus 1) */
for (i=0;i<=wc;i++)
{
*dst++ = *src++; /* ALL dst MUST BE in same Write Buffer */
}
*( (UINT16 *)sector_address ) = 0x0029; /* write confirm command */
/* poll for completion */
/* Example: Write Buffer Abort Reset */
*( (UINT16 *)base_addr + 0x555 ) = 0x00AA; /* write unlock cycle 1 */
*( (UINT16 *)base_addr + 0x2AA ) = 0x0055; /* write unlock cycle 2 */
*( (UINT16 *)base_addr + 0x555 ) = 0x00F0; /* write buffer abort reset */
```

Figure 7.3 Write Buffer Programming Operation



7.5.3 Sector Erase

The sector erase function erases one or more sectors in the memory array. (See Table 12.1 and Figure 7.4.) The device does not require the system to preprogram prior to erase. The Embedded Erase algorithm automatically programs and verifies the entire memory for an all zero data pattern prior to electrical erase. After a successful sector erase, all locations within the erased sector contain FFFFh. The system is not required to provide any controls or timings during these operations.

After the command sequence is written, a sector erase time-out of no less than t_{SEA} occurs. During the time-out period, additional sector addresses and sector erase commands may be written. Loading the sector erase buffer may be done in any sequence, and the number of sectors may be from one sector to all sectors. The time between these additional cycles must be less than t_{SEA}. Any sector erase address and command following the exceeded time-out (t_{SEA}) may or may not be accepted. Any command other than Sector Erase or Erase Suspend during the time-out period resets that bank to the read mode. The system can monitor DQ3 to determine if the sector erase timer has timed out (See the section, *DQ3: Sector Erase Timeout State Indicator*.) The time-out begins from the rising edge of the final WE# pulse in the command sequence.

When the Embedded Erase algorithm is complete, the bank returns to reading array data and addresses are no longer latched. Note that while the Embedded Erase operation is in progress, the system can read data from the non-erasing banks. The system can determine the status of the erase operation by reading DQ7 or DQ6/DQ2 in the erasing bank. Refer to Section 7.5.9, *Write Operation Status* on page 46 for information on these status bits.

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

Figure 7.4 illustrates the algorithm for the erase operation. Refer to Section 7.5, *Program/Erase Operations* on page 35 for parameters and timing diagrams.

Software Functions and Sample Code

Table 7.17 Sector Erase

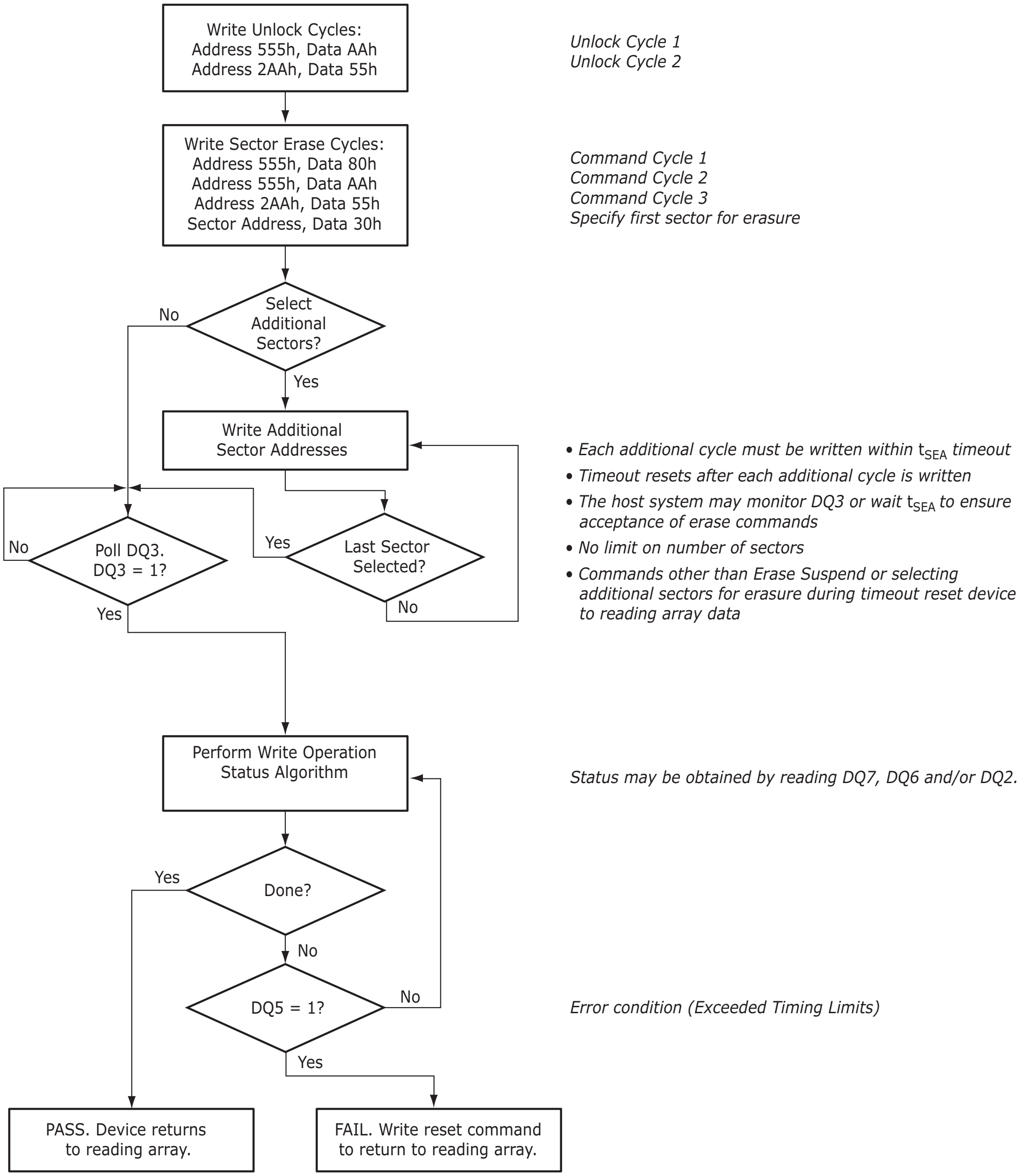
(LLD Function = lld_SectorEraseCmd)

Cycle	Description	Operation	Byte Address	Word Address	Data
1	Unlock	Write	Base + AAAh	Base + 555h	00AAh
2	Unlock	Write	Base + 554h	Base + 2AAh	0055h
3	Setup Command	Write	Base + AAAh	Base + 555h	0080h
4	Unlock	Write	Base + AAAh	Base + 555h	00AAh
5	Unlock	Write	Base + 554h	Base + 2AAh	0055h
6	Sector Erase Command	Write	Sector Address	Sector Address	0030h
Unlimited additional sectors may be selected for erase; command(s) must be written within t_{SEA}.					

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

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

Figure 7.4 Sector Erase Operation



Unlock Cycle 1
Unlock Cycle 2

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

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

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

Error condition (Exceeded Timing Limits)

Notes

1. *See Table 12.1 for erase command sequence.*
2. *See DQ3: Sector Erase Timeout State Indicator for information on the sector erase timeout.*

7.5.4 Chip Erase Command Sequence

Chip erase is a six-bus cycle operation as indicated by Table 12.1. These commands invoke the Embedded Erase algorithm, which does not require the system to preprogram prior to erase. The Embedded Erase algorithm automatically preprograms and verifies the entire memory for an all zero data pattern prior to electrical erase. After a successful chip erase, all locations of the chip contain FFFFh. The system is not required to provide any controls or timings during these operations. Table 12.1 shows the address and data requirements for the chip erase command sequence.

When the Embedded Erase algorithm is complete, that bank returns to the read mode and addresses are no longer latched. The system can determine the status of the erase operation by using DQ7 or DQ6/DQ2. Refer to Section 7.5.9, *Write Operation Status* on page 46 for information on these status bits.

Any commands written during the chip erase operation are ignored. However, note that a hardware reset immediately terminates the erase operation. If that occurs, the chip erase command sequence should be reinitiated once that bank has returned to reading array data, to ensure data integrity.

Software Functions and Sample Code

Table 7.18 Chip Erase

(LLD Function = lld_ChipEraseCmd)

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

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

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

7.5.5 Erase Suspend/Erase Resume Commands

When the Erase Suspend command is written during the sector erase time-out, the device immediately terminates the time-out period and suspends the erase operation. The Erase Suspend command allows the system to interrupt a sector erase operation and then read data from, or program data to, any sector not selected for erasure. The bank address is required when writing this command. This command is valid only during the sector erase operation, including the minimum t_{SEA} time-out period during the sector erase command sequence. The Erase Suspend command is ignored if written during the chip erase operation.

When the Erase Suspend command is written after the t_{SEA} time-out period has expired and during the sector erase operation, the device requires a maximum of t_{ESL} (erase suspend latency) to suspend the erase operation.

After the erase operation has been suspended, the bank enters the erase-suspend-read mode. The system can read data from or program data to any sector not selected for erasure. (The device *erase suspends* all sectors selected for erasure.) Reading at any address within erase-suspended sectors produces status information on DQ7-DQ0. The system can use DQ7, or DQ6, and DQ2 together, to determine if a sector is actively erasing or is erase-suspended. Refer to Table 7.27 for information on these status bits.

After an erase-suspended program operation is complete, the bank returns to the erase-suspend-read mode. The system can determine the status of the program operation using the DQ7 or DQ6 status bits, just as in the standard program operation.

In the erase-suspend-read mode, the system can also issue the Autoselect command sequence. Refer to Section 7.5.2, *Write Buffer Programming* on page 37 and Section 7.4, *Autoselect* on page 33 for details.

To resume the sector erase operation, the system must write the Erase Resume command. The bank address of the erase-suspended bank is required when writing this command. Further writes of the Resume command are ignored. Another Erase Suspend command can be written after the chip has resumed erasing.

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

Software Functions and Sample Code

Table 7.19 Erase Suspend

(LLD Function = lld_EraseSuspendCmd)

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

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

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

Table 7.20 Erase Resume

(LLD Function = lld_EraseResumeCmd)

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

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

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

7.5.6 Program Suspend/Program Resume Commands

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

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

The system may also write the Autoselect command sequence when the device is in Program Suspend mode. The device allows reading Autoselect codes in the suspended sectors, since the codes are not stored in the memory array. When the device exits the Autoselect mode, the device reverts to Program Suspend mode, and is ready for another valid operation. See Section 7.4, *Autoselect* on page 33 for more information.

After the Program Resume command is written, the device reverts to programming. The system can determine the status of the program operation using the DQ7 or DQ6 status bits, just as in the standard program operation. See Section 7.5.9, *Write Operation Status* on page 46 for more information.

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

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

Software Functions and Sample Code

Table 7.21 Program Suspend

(LLD Function = lld_ProgramSuspendCmd)

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

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

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

Table 7.22 Program Resume

(LLD Function = lld_ProgramResumeCmd)

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

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

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

7.5.7 Accelerated Program/Sector Erase

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

The accelerated chip program and erase functions must not be used more than 100 times per sector. In addition, accelerated chip program and erase should be performed at room temperature
(30°C ±10°C).

If the system asserts V_{HH} on this input, the device automatically enters the accelerated mode and uses the higher voltage on the input to reduce the time required for program and erase operations. The system can then use the Write Buffer Load command sequence provided by the Unlock Bypass mode. Note that if a *Write-to-Buffer-Abort Reset* is required while in Unlock Bypass mode, the full 3-cycle RESET command sequence must be used to reset the device. Removing V_{HH} from the V_{PP} input, upon completion of the embedded program or erase operation, returns the device to normal operation.

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

7.5.8 Unlock Bypass

The unlock bypass feature allows the system to primarily program to a bank faster than using the standard program command sequence. The unlock bypass command sequence is initiated by first writing two unlock cycles. This is followed by a third write cycle containing the unlock bypass command, 20h. The device then enters the unlock bypass mode. A two-cycle unlock bypass program command sequence is all that is required to program in this mode. The first cycle in this sequence contains the unlock bypass program command, A0h; the second cycle contains the program address and data. Additional data is programmed in the same manner. This mode dispenses with the initial two unlock cycles required in the standard program command sequence, resulting in faster total programming time. The host system may also initiate the chip erase and sector erase sequences in the unlock bypass mode. The erase command sequences are four cycles in length instead of six cycles. Table 12.1 shows the requirements for the unlock bypass command sequences.

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

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

Refer to the *Erase/Program Timing* section for parameters, and Figures 11.12 and 11.13 for timing diagrams

Software Functions and Sample Code

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

Table 7.23 Unlock Bypass Entry

(LLD Function = lld_UnlockBypassEntryCmd)

Cycle	Description	Operation	Byte Address	Word Address	Data
1	Unlock	Write	Base + AAAh	Base + 555h	00AAh
2	Unlock	Write	Base + 554h	Base + 2AAh	0055h
3	Entry Command	Write	Base + AAAh	Base + 555h	0020h

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

Table 7.24 Unlock Bypass Program

(LLD Function = lld_UnlockBypassProgramCmd)

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

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

Table 7.25 Unlock Bypass Reset

(LLD Function = lld_UnlockBypassResetCmd)

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

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

7.5.9 Write Operation Status

The device provides several bits to determine the status of a program or erase operation. The following subsections describe the function of DQ1, DQ2, DQ3, DQ5, DQ6, and DQ7.

DQ7: Data# Polling

The Data# Polling bit, DQ7, indicates to the host system whether an Embedded Program or Erase algorithm is in progress or completed, or whether a bank is in Erase Suspend. Data# Polling is valid after the rising edge of the final WE# pulse in the command sequence. Note that the Data# Polling is valid only for the last word being programmed in the write-buffer-page during Write Buffer Programming. Reading Data# Polling status on any word other than the last word to be programmed in the write-buffer-page returns false status information. Similarly, attempting to program *1* over a *0* does not return valid Date# information.

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

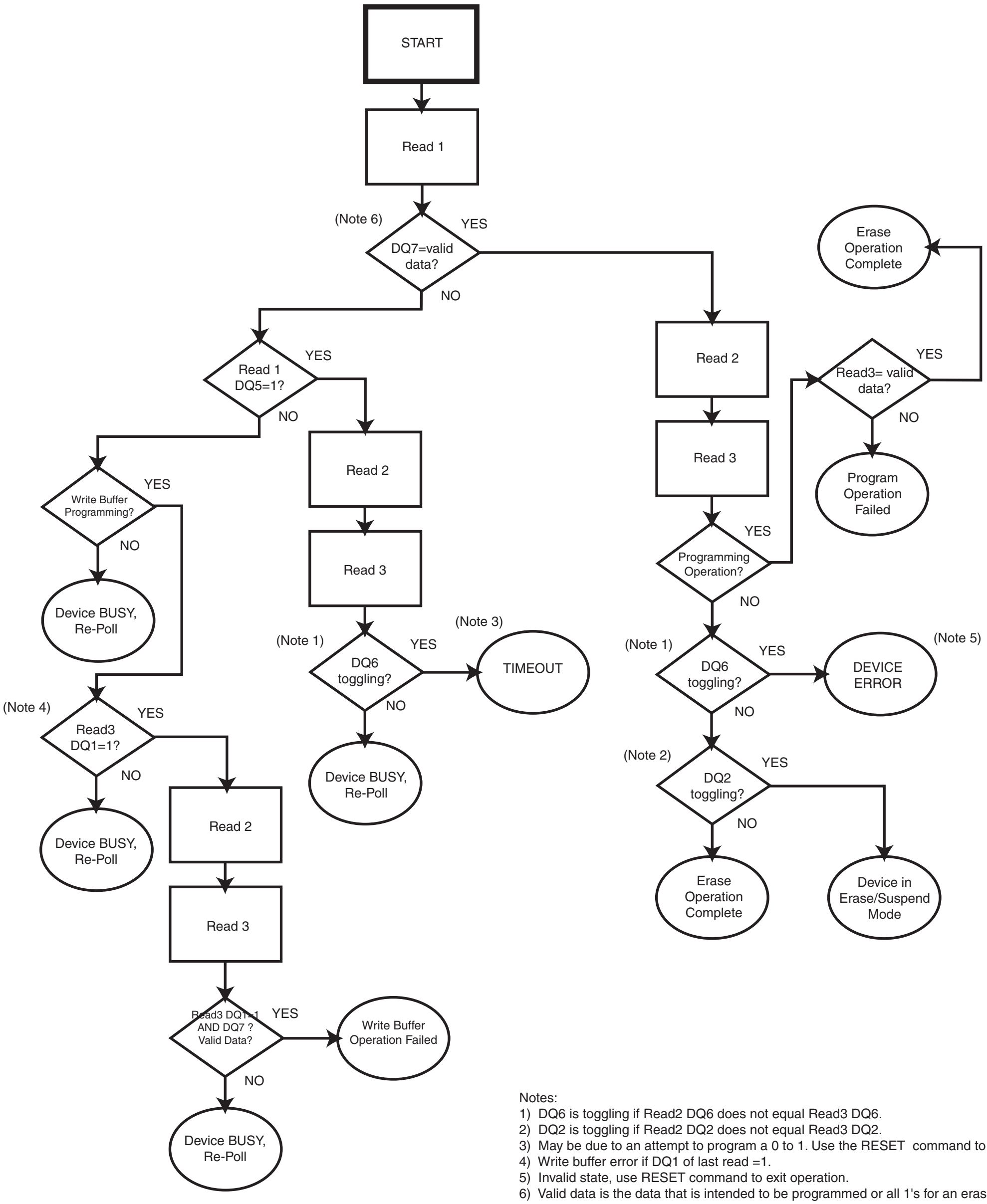
During the Embedded Erase Algorithm, Data# polling produces a *0* on DQ7. When the Embedded Erase algorithm is complete, or if the bank enters the Erase Suspend mode, Data# Polling produces a *1* on DQ7. The system must provide an address within any of the sectors selected for erasure to read valid status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

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

Just prior to the completion of an Embedded Program or Erase operation, DQ7 may change asynchronously with DQ6-DQ1 while Output Enable (OE#) is asserted low. That is, the device may change from providing status information to valid data on DQ7. Even if the device has completed the program or erase operation and DQ7 has valid data, the data outputs on DQ6-DQ1 may be still invalid. Valid data on DQ7-D01 appears on successive read cycles.

See the following for more information: Table 7.27, *Write Operation Status*, shows the outputs for Data# Polling on DQ7. Table 7.5, *Write Operation Status Flowchart*, shows the Data# Polling algorithm; and Figure 11.15, *Data# Polling Timings (During Embedded Algorithm)*, shows the Data# Polling timing diagram.

Figure 7.5 Write Operation Status Flowchart



Notes:
1) DQ6 is toggling if Read2 DQ6 does not equal Read3 DQ6.
2) DQ2 is toggling if Read2 DQ2 does not equal Read3 DQ2.
3) May be due to an attempt to program a 0 to 1. Use the RESET command to exit operation.
4) Write buffer error if DQ1 of last read =1.
5) Invalid state, use RESET command to exit operation.
6) Valid data is the data that is intended to be programmed or all 1's for an erase operation.
7) Data polling algorithm valid for all operations except advanced sector protection.
8) It can fail if one tries to program DQ7 from '0' to '1'

7.5.9.1 DQ6: Toggle Bit I

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

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

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

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

If a program address falls within a protected sector, DQ6 toggles for approximately t_{PSP} after the program command sequence is written, then returns to reading array data.

DQ6 also toggles during the erase-suspend-program mode, and stops toggling once the Embedded Program Algorithm is complete.

See the following for additional information: Figure 7.5, *Write Operation Status Flowchart*; Figure 11.16, *Toggle Bit Timings (During Embedded Algorithm)*, and Tables 7.26 and 7.27.

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

7.5.9.2 DQ2: Toggle Bit II

The *Toggle Bit II* on DQ2, when used with DQ6, indicates whether a particular sector is actively erasing (that is, the Embedded Erase algorithm is in progress), or whether that sector is erase-suspended. Toggle Bit II is valid after the rising edge of the final WE# pulse in the command sequence. DQ2 toggles when the system reads at addresses within those sectors that have been selected for erasure. But DQ2 cannot distinguish whether the sector is actively erasing or is erase-suspended. DQ6, by comparison, indicates whether the device is actively erasing, or is in Erase Suspend, but cannot distinguish which sectors are selected for erasure. Thus, both status bits are required for sector and mode information. Refer to Table 7.26 to compare outputs for DQ2 and DQ6. See the following for additional information: Figure 7.5, the *DQ6: Toggle Bit I* section, and Figures 11.15 – 11.18.

Table 7.26 DQ6 and DQ2 Indications

If device is	and the system reads	then DQ6	and DQ2
programming,	any address at the bank being programmed	toggles,	does not toggle.
actively erasing,	at an address within a sector selected for erasure,	toggles,	also toggles.
	at an address within sectors *not* selected for erasure,	toggles,	does not toggle.
erase suspended,	at an address within a sector selected for erasure,	does not toggle,	toggles.
	at an address within sectors *not* selected for erasure,	returns array data,	returns array data. The system can read from any sector not selected for erasure.
programming in erase suspend	any address at the bank being programmed	toggles,	is not applicable.

7.5.9.3 Reading Toggle Bits DQ6/DQ2

Whenever the system initially begins reading toggle bit status, it must read DQ7 – DQ0 at least twice in a row to determine whether a toggle bit is toggling. Typically, the system notes and stores the value of the toggle bit after the first read. After the second read, the system compares the new value of the toggle bit with the first. If the toggle bit is not toggling, the device has completed the program or erases operation. The system can read array data on DQ7 – DQ0 on the following read cycle. However, if after the initial two read cycles, the system determines that the toggle bit is still toggling, the system also should note whether the value of DQ5 is high (see the section on DQ5). If it is, the system should then determine again whether the toggle bit is toggling, since the toggle bit may have stopped toggling just as DQ5 went high. If the toggle bit is no longer toggling, the device has successfully completed the program or erases operation. If it is still toggling, the device did not complete the operation successfully, and the system must write the reset command to return to reading array data. The remaining scenario is that the system initially determines that the toggle bit is toggling and DQ5 has not gone high. The system may continue to monitor the toggle bit and DQ5 through successive read cycles, determining the status as described in the previous paragraph. Alternatively, it may choose to perform other system tasks. In this case, the system must start at the beginning of the algorithm when it returns to determine the status of the operation. Refer to Figure 7.5 for more details.

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

7.5.9.4 DQ5: Exceeded Timing Limits

DQ5 indicates whether the program or erase time has exceeded a specified internal pulse count limit. Under these conditions DQ5 produces a *1*, indicating that the program or erase cycle was not successfully completed. The device may output a *1* on DQ5 if the system tries to program a *1* to a location that was previously programmed to *0*. Only an erase operation can change a *0* back to a *1*. Under this condition, the device halts the operation, and when the timing limit has been exceeded, DQ5 produces a *1*. Under both these conditions, the system must write the reset command to return to the read mode (or to the erase-suspend-read mode if a bank was previously in the erase-suspend-program mode).

7.5.9.5 DQ3: Sector Erase Timeout State Indicator

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

After the sector erase command is written, the system should read the status of DQ7 (Data# Polling) or DQ6 (Toggle Bit I) to ensure that the device has accepted the command sequence, and then read DQ3. If DQ3 is *1*, the Embedded Erase algorithm has begun; all further commands (except Erase Suspend) are ignored until the erase operation is complete. If DQ3 is *0*, the device accepts additional sector erase commands. To ensure the command has been accepted, the system software should check the status of DQ3 prior to and following each sub-sequent sector erase command. If DQ3 is high on the second status check, the last command might not have been accepted. Table 7.27 shows the status of DQ3 relative to the other status bits.

7.5.9.6 DQ1: Write to Buffer Abort

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

Table 7.27 Write Operation Status

Status			DQ7 (2)	DQ6	DQ5 (1)	DQ3	DQ2 (2)	DQ1 (4)
Standard Mode	Embedded Program Algorithm		DQ7#	Toggle	0	N/A	No toggle	0
	Embedded Erase Algorithm		0	Toggle	0	1	Toggle	N/A
Program Suspend Mode (3)	Reading within Program Suspended Sector		INVALID (Not Allowed)	INVALID (Not Allowed)	INVALID (Not Allowed)	INVALID (Not Allowed)	INVALID (Not Allowed)	INVALID (Not Allowed)
	Reading within Non-Program Suspended Sector		Data	Data	Data	Data	Data	Data
Erase Suspend Mode	Erase-Suspend-Read	Erase Suspended Sector	1	No toggle	0	N/A	Toggle	N/A
		Non-Erase Suspended Sector	Data	Data	Data	Data	Data	Data
	Erase-Suspend-Program		DQ7#	Toggle	0	N/A	N/A	N/A
Write to Buffer (5)	BUSY State		DQ7#	Toggle	0	N/A	N/A	0
	Exceeded Timing Limits		DQ7#	Toggle	1	N/A	N/A	0
	ABORT State		DQ7#	Toggle	0	N/A	N/A	1

Notes

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

7.6 Simultaneous Read/Write

The simultaneous read/write feature allows the host system to read data from one bank of memory while programming or erasing another bank of memory. An erase operation may also be suspended to read from or program another location within the same bank (except the sector being erased). Figure 11.21, *Back-to-Back Read/Write Cycle Timings*, shows how read and write cycles may be initiated for simultaneous operation with zero latency. Refer to the *DC Characteristics* table for read-while-program and read-while-erase current specification.

7.7 Writing Commands/Command Sequences

When the device is in Asynchronous read, only Asynchronous write operations are allowed. During an asynchronous write operation, the system must drive CE# and WE# to V_{IL} and OE# to V_{IH} when providing an address, command, and data. Addresses are latched on the rising edge of AVD#, while data is latched on the rising edge of WE#. An erase operation can erase one sector, multiple sectors, or the entire device. Table 6.1 – Table 6.3 indicate the address space that each sector occupies. The device address space is divided into sixteen banks: for NS512P, all 16 banks contain 64-Kword sectors while for NS256P and NS128P, Banks 0 through 14 contain only 64 Kword sectors, Bank 15 contains 16-Kword boot sectors in addition to 64 Kword sectors. A *bank address* is the set of address bits required to uniquely select a bank. Similarly, a *sector address* is the address bits required to uniquely select a sector. I_{CC2} in the *DC Characteristics* section represents the active current specification for the write mode. *AC Characteristics-Synchronous* and *AC Characteristics-Asynchronous* contain timing specification tables and timing diagrams for write operations.

7.8 Handshaking

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

7.9 Hardware Reset

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

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

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

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

See Figures 11.5 and 11.11 for timing diagrams.

7.10 Software Reset

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

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

Software Functions and Sample Code

Table 7.28 Reset

(LLD Function = lld_ResetCmd)

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

Note
Base = Base Address.

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

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

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

- This command resets the banks to the read and address bits are ignored.
- Reset commands are ignored once erasure has begun until the operation is complete.
- Once programming begins, the device ignores reset commands until the operation is complete
- The reset command may be written between the cycles in a program command sequence before programming begins (prior to the third cycle). This resets the bank to which the system was writing to the read mode.
- If the program command sequence is written to a bank that is in the Erase Suspend mode, writing the reset command returns that bank to the erase-suspend-read mode.
- The reset command may be also written during an Autoselect command sequence.
- If a bank has entered the Autoselect mode while in the Erase Suspend mode, writing the reset command returns that bank to the erase-suspend-read mode.
- If DQ1 goes high during a Write Buffer Programming operation, the system must write the *Write to Buffer Abort Reset* command sequence to RESET the device to reading array data. The standard RESET command does not work during this condition.
- To exit the unlock bypass mode, the system must issue a two-cycle unlock bypass reset command sequence [see command table for details].

7.11 Programmable Output Slew Rate Control

This feature allows the user to change the output slew rate during a read operation by setting the configuration register bit CR1.4. It allows 2 programmable slew rates. This feature is for users who do not want to run the part at its maximum speed and could live with a slower output slew rate thereby reducing noise variations at the output.

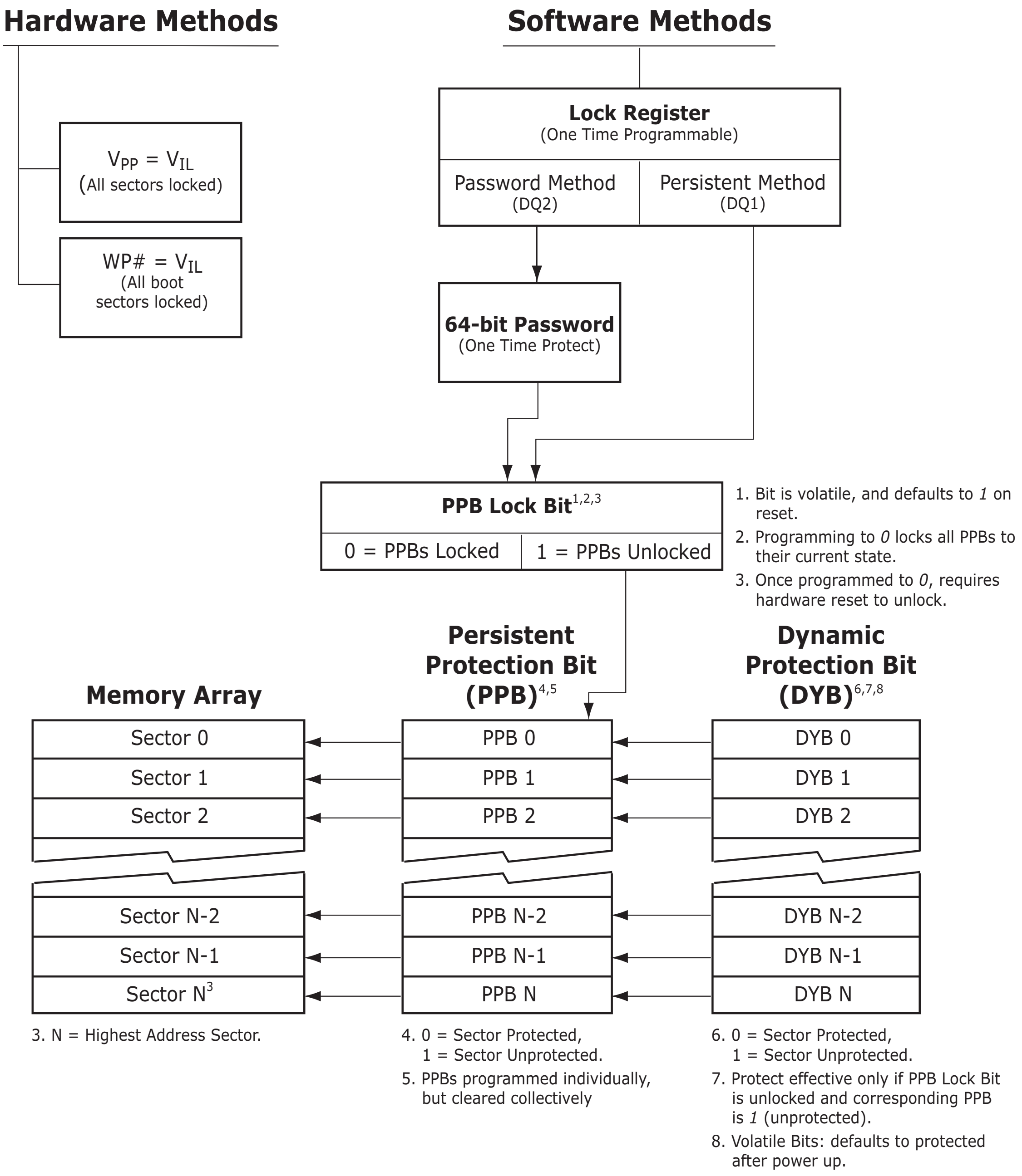
Table 7.29 Programmable Output Slew Rate

Mode	Description	I_{OL} and I_{OH}
1	Full Drive (Default)	100 µA
2	Half Drive	50 µA

8. Advanced Sector Protection/Unprotection

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

Figure 8.1 Advanced Sector Protection/Unprotection



1. Bit is volatile, and defaults to *1* on reset.
2. Programming to *0* locks all PPBs to their current state.
3. Once programmed to *0*, requires hardware reset to unlock.

3. N = Highest Address Sector.

4. 0 = Sector Protected,
   1 = Sector Unprotected.
5. PPBs programmed individually, but cleared collectively

6. 0 = Sector Protected,
   1 = Sector Unprotected.
7. Protect effective only if PPB Lock Bit is unlocked and corresponding PPB is *1* (unprotected).
8. Volatile Bits: defaults to protected after power up.

8.1 Lock Register

The Lock Register consists of 5 bits. The Secured Silicon Sector Protection Bit is DQ0, Persistent Protection Mode Lock Bit is DQ1, Password Protection Mode Lock Bit is DQ2, Persistent Sector Protection OTP bit is DQ3 and Volatile Sector Protection Boot bit is DQ4. If DQ0 is *0*, it means that the Customer Secured Silicon area is locked and if DQ0 is *1*, it means that it is unlocked. When DQ2 is set to *1* and DQ1 is set to *0*, the device can only be used in the Persistent Protection Mode. When the device is set to Password Protection Mode, DQ1 is required to be set to *1* and DQ2 is required to be set to *0*. DQ3 is programmed in the Spansion factory. When the device is programmed to disable all PPB erase command, DQ3 outputs a *0*, when the lock register bits are read. Similarly, if the device is programmed to enable all PPB erase command, DQ3 outputs a *1* when the lock register bits are read. Likewise the DQ4 bit is also programmed in the Spansion Factory. DQ4 is the bit which indicates whether Volatile Sector Protection Bit (DYB) is protected or not after boot up. When the device is programmed to set all Volatile Sector Protection Bit protected after power up, DQ4 outputs a *0* when the lock register bits are read. Similarly, when the device is programmed to set all Volatile Sector Protection Bit unprotected after power up, DQ4 outputs a *1*. Each of these bits in the lock register are non-volatile. DQ15 – DQ5 are reserved and are *1*s.

Lock Register

DQ15-5	DQ4	DQ3	DQ2	DQ1	DQ0
*1*s	DYB Lock Boot Bit 0 = DYB bits power up protected (Default) 1 = DYB bits power up unprotected	PPB One Time Programmable Bit 0 = All PPB Erase Command disabled 1 = All PPB Erase Command enabled	Password Protection Mode Lock Bit	Persistent Protection Mode Lock Bit	Secured Silicon Sector Protection Bit

For programming lock register bits refer to Table 12.2.

Notes

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

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

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

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

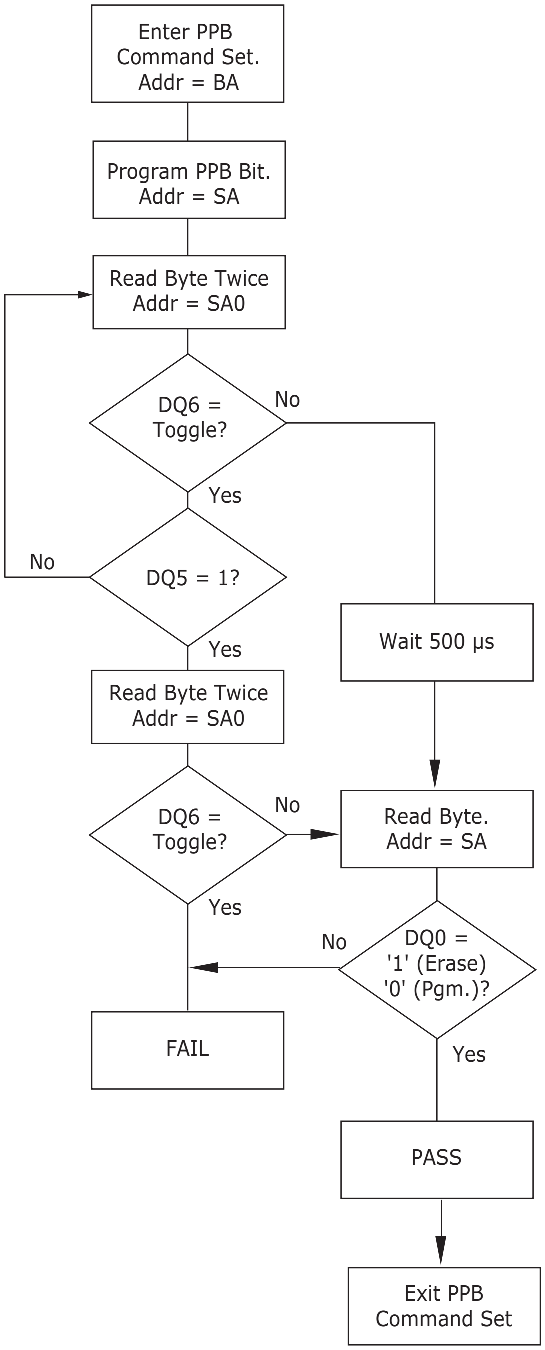
8.2 Persistent Protection Bits

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

Notes

1. Each PPB is individually programmed and all are erased in parallel.
2. While programming PPB for a sector, array data can be read from any other bank, except Bank 0 (used for Data# Polling) and the bank in which sector PPB is being programmed.
3. Entry command disables reads and writes for the bank selected.
4. Reads within that bank return the PPB status for that sector.
5. Reads from other banks are allowed while writes are not allowed.
6. All Reads must be performed using the Asynchronous mode.
7. The specific sector address (Amax – A14) are written at the same time as the program command.
8. If the PPB Lock Bit is set, the PPB Program or erase command does not execute and time out without programming or erasing the PPB.
9. There are no means for individually erasing a specific PPB and no specific sector address is required for this operation.
10. PPB exit command must be issued after the execution which resets the device to read mode and re-enables reads and writes for Bank 0
11. The programming state of the PPB for a given sector can be verified by writing a PPB Status Read Command to the device as described by the flow chart shown in Figure 8.2.
12. During PPB program/erase data polling can be done synchronously.
13. If customers attempt to program or erase a protected sector, the device ignores the command and returns to read mode.

Figure 8.2 PPB Program/Erase Algorithm



8.3 Dynamic Protection Bits

Dynamic Protection Bits are volatile and unique for each sector and can be individually modified. DYBs only control the protection scheme for unprotected sectors that have their PPBs cleared (erased to *1*). By issuing the DYB Set or Clear command sequences, the DYBs are set (programmed to *0*) or cleared (erased to *1*), thus placing each sector in the protected or unprotected state respectively. This feature allows software to easily protect sectors against inadvertent changes yet does not prevent the easy removal of protection when changes are needed.

Notes

1. The DYBs can be set (programmed to *0*) or cleared (erased to *1*) as often as needed.
2. When the parts are first shipped, the DYBs are set and programmed to 0 upon power up or reset.
3. The default state of DYB is protected after power up and all sectors can be modified depending on the status of PPB bit for that sector, (erased to *1*). Then the sectors can be modified depending upon the PPB state of that sector (see Table 8.1).
4. It is possible to have sectors that are persistently locked with sectors that are left in the dynamic state.
5. The DYB Set or Clear commands for the dynamic sectors signify protected or unprotectedstate of the sectors respectively. However, if there is a need to change the status of the persistently locked sectors, a few more steps are required. First, the PPB Lock Bit must be cleared by either putting the device through a power cycle, or hardware reset. The PPBs can then be changed to reflect the desired settings. Setting the PPB Lock Bit once again locks the PPBs, and the device operates normally again.
6. To achieve the best protection, it is recommended to execute the PPB Lock Bit Set command early in the boot code and protect the boot code by holding WP# = V_{IL}.
7. Data polling is not available for DYB program/erase.
8. DYB read data can be done synchronously.
9. If customers attempt to program or erase a protected sector, the device ignores the command and returns to read mode.

8.4 Persistent Protection Bit Lock Bit

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

Notes

1. If the password mode is chosen, then the password must be programmed and verified before setting the corresponding lock register bit.
2. No software command sequence unlocks this bit unless the device is in the password protection mode; only a hardware reset or a power up clears this bit.
3. The PPB Lock Bit must be set (programmed to *0*) only after all PPBs are configured to the desired settings.

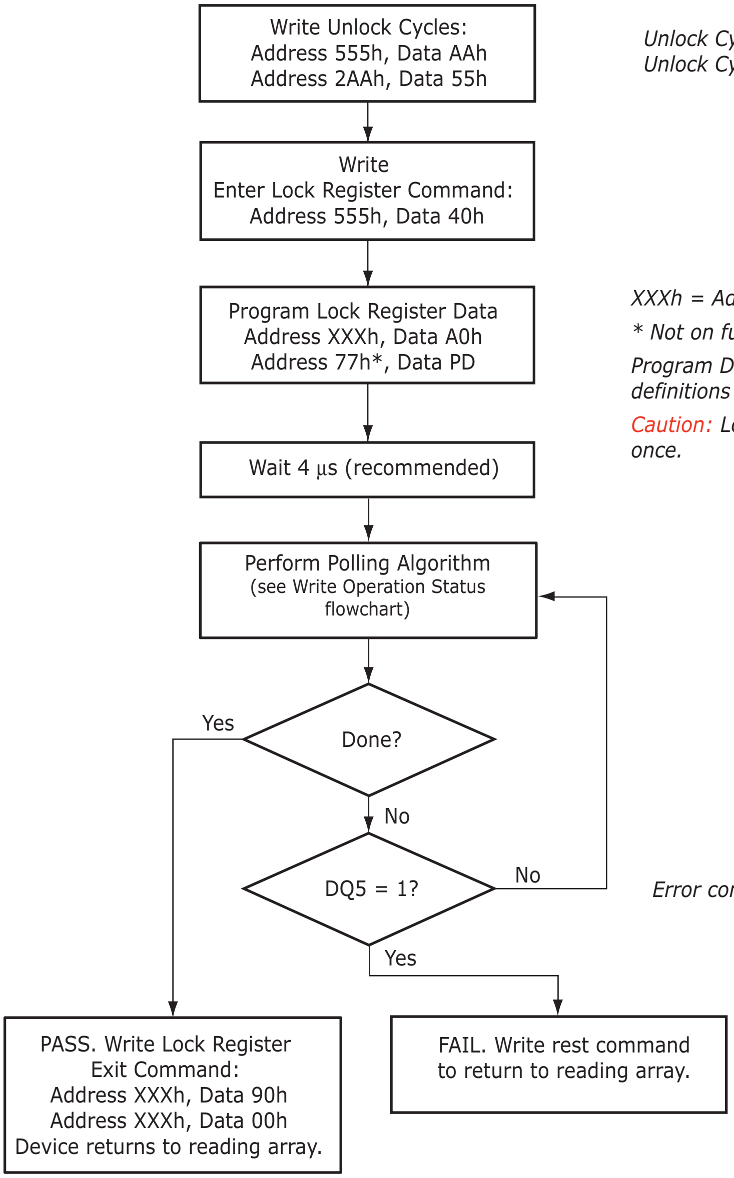
8.5 Password Protection Method

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

Notes

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

Figure 8.3 Lock Register Program Algorithm



Unlock Cycle 1
Unlock Cycle 2

XXXh = Address don't care

** Not on future devices*

Program Data (PD): See text for Lock Register definitions

Caution: Lock register can only be progammed once.

Error condition (Exceeded Timing Limits)

8.6 Advanced Sector Protection Software Examples

Table 8.1 Sector Protection Schemes

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

Table 8.1 contains all possible combinations of the DYB, PPB, and PPB Lock Bit relating to the status of the sector.

8.7 Hardware Data Protection Methods

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

- When WP# is at V_{IL}, the highest two sectors are locked (device specific).
- When V_{PP} is at V_{IL}, all sectors are locked.

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

8.7.1 WP# Method

The Write Protect feature provides a hardware method of protecting the highest two sectors (NS256P and NS128P). This function is provided by the WP# pin and overrides the previously discussed Sector Protection/Unprotection method.

If the system asserts V_{IL} on the WP# pin, the device disables program and erase functions in the highest two sectors (NS256P and NS128P) as well as Secured Silicon Area.

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

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

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

8.7.2 V_{PP} Method

This method is similar to above, except it protects all sectors (including the Secured Silicon Area). Once V_{PP} input is set to V_{IL}, all program and erase functions are disabled and hence all sectors are protected.

8.7.3 Low V_{CC} Write Inhibit

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

The command register and all internal program/erase circuits are disabled, and the device resets to reading array data. Subsequent writes are ignored until V_{CC} is greater than V_{LKO}. The system must provide the proper signals to the control inputs to prevent unintentional writes when V_{CC} is greater than V_{LKO}.

8.7.4 Write Pulse *Glitch Protection*

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

8.7.5 Power-Up Write Inhibit

If WE# = CE# = RESET# = V_{IL} and OE# = V_{IH} during power up, the device does not accept commands on the rising edge of WE#. The internal state machine is automatically reset to the read mode on power up.

9. Power Conservation Modes

9.1 Standby Mode

When the system is not reading or writing to the device, it can place the device in the standby mode. In this mode, current consumption is greatly reduced, and the outputs are placed in the high impedance state, independent of the OE# input. The device enters the CMOS standby mode when the CE# and RESET# inputs are both held at V_{CC} ± 0.2 V. The device requires standard access time (t_{CE}) for read access, before it is ready to read data. If the device is deselected during erasure or programming, the device draws active current until the operation is completed. I_{CC3} in the *DC Characteristics* section represents the standby current specification

9.2 Automatic Sleep Mode

The automatic sleep mode minimizes Flash device energy consumption while in asynchronous mode. the device automatically enables this mode when addresses remain stable for t_{ACC} + 20 ns. The automatic sleep mode is independent of the CE#, WE#, and OE# control signals. Standard address access timings provide new data when addresses are changed. While in sleep mode, output data is latched and always available to the system. While in synchronous mode, the automatic sleep mode is disabled. Note that a new burst operation is required to provide new data. I_{CC6} in the *DC Characteristics* section represents the automatic sleep mode current specification.

9.3 Hardware RESET# Input Operation

The RESET# input provides a hardware method of resetting the device to reading array data. When RESET# is driven low for at least a period of t_{RP}, the device immediately terminates any operation in progress, tristates all outputs, resets the configuration register, and ignores all read/write commands for the duration of the RESET# pulse. The device also resets the internal state machine to reading array data. The operation that was interrupted should be reinitiated once the device is ready to accept another command sequence to ensure data integrity.

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

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

9.4 Output Disable (OE#)

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

10. Secured Silicon Sector Flash Memory Region

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

Please note the following general conditions:

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

Table 10.1 Secured Silicon SectorSecure Sector Addresses

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

10.1 Factory Secured Silicon Sector

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

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

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

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

10.2 Customer Secured Silicon Sector

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

Please note the following:

- Once the Customer Secured Silicon Sector area is protected, the Customer Indicator Bit is permanently set to *1*.

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

10.3 Secured Silicon Sector Entry and Exit Command Sequences

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

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

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

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

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

Software Functions and Sample Code

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

Table 10.2 Secured Silicon Sector Entry (LLD Function = lld_SecSiSectorEntryCmd)

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

Note
Base = Base Address.

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

Table 10.3 Secured Silicon Sector Program (LLD Function = lld_ProgramCmd)

Cycle	Operation	Byte Address	Word Address	Data
Unlock Cycle 1	Write	Base + AAAh	Base + 555h	00AAh
Unlock Cycle 2	Write	Base + 554h	Base + 2AAh	0055h
Program Setup	Write	Base + AAAh	Base + 555h	00A0h
Program	Write	Word Address	Word Address	Data Word

Note
Base = Base Address.

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

Table 10.4 Secured Silicon Sector Exit (LLD Function = lld_SecSiSectorExitCmd)

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

Note

Base = Base Address.

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

11. Electrical Specifications

11.1 Absolute Maximum Ratings

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

Notes

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

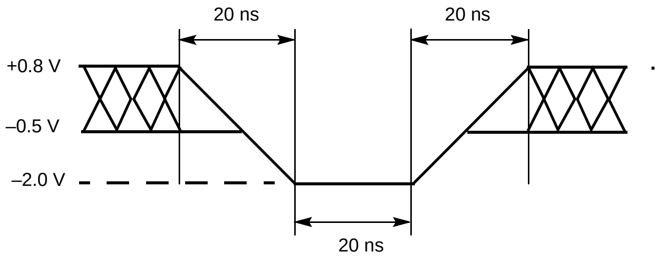
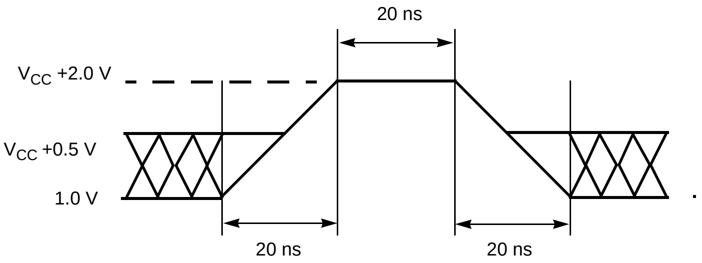


Figure 11.2 Maximum Positive Overshoot Waveform



11.2 Operating Ranges

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

Note

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

11.3 DC Characteristics

11.3.1 CMOS Compatible

Table 11.1 DC Characteristics—CMOS Compatible

Parameter	Description	Test Conditions (1)		Min	Typ	Max	Unit
I_{LI}	Input Load Current	$V_{IN} = V_{SS}$ to V_{CC}, $V_{CC} = V_{CC}$max				±1	µA
I_{LO}	Output Leakage Current	$V_{OUT} = V_{SS}$ to V_{CC}, $V_{CC} = V_{CC}$max				±1	µA
I_{CCB}	V_{CC} Active burst Read Current	CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = 8	83 Mhz		26	36	mA
			66 Mhz		24	33	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = 16	83 Mhz		26	38	mA
			66 Mhz		24	35	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = 32	83 Mhz		28	40	mA
			66 Mhz		26	37	mA
		CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}, burst length = Continuous	83 Mhz		30	42	mA
			66 Mhz		28	39	mA
I_{CC1}	V_{CC} Active Asynchronous Read Current (2)	CE# = V_{IL}, OE# = V_{IH}, WE# = V_{IH}	10 MHz		40	80	mA
			5 MHz		20	40	mA
			1 MHz		10	20	mA
I_{CC2}	V_{CC} Active Write Current (3)	CE# = V_{IL}, OE# = V_{IH}, $V_{PP} = VI_H$	V_{PP}		1	5	µA
			V_{CC}		<20	<40	mA
I_{CC3}	V_{CC} Standby Current (4)	CE# = RESET# = V_{CC} ± 0.2 V	V_{PP}		1	5	µA
			V_{CC}		20	70	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)	CE# = V_{IL}, OE# = V_{IH}, $V_{PP} = V_{IH}$, (7)			50	60	mA
I_{CC6}	V_{CC} Sleep Current	CE# = V_{IL}, OE# = V_{IH}			5	40	µA
I_{PPW}	Accelerated Program Current (5)	CE# = V_{IL}, OE# = $V_{IH,}$ V_{PP} = 9.5 V	V_{PP}		<7	<10	mA
			V_{CC}		<15	<20	mA
V_{IL}	Input Low Voltage			–0.2		0.4	V
V_{IH}	Input High Voltage			V_{CC} – 0.4		V_{CC} + 0.4	
V_{OL}	Output Low Voltage	I_{OL} = 100 µA, $V_{CC} = V_{CC\ min} = V_{CC}$				0.1	V
V_{OH}	Output High Voltage	I_{OH} = –100 µA, $V_{CC} = V_{CC\ min}$		V_{CC} – 0.1			V
V_{HH}	Voltage for Accelerated Program			8.5		9.5	V
V_{LKO}	Low V_{CC} Lock-out Voltage					1.4	V

Notes

1. *Maximum I_{CC} specifications are tested with $V_{CC} = V_{CC}$max.*
2. *The I_{CC} current listed is typically less than 2 mA/MHz, with OE# at V_{IH}.*
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. *$V_{CCQ} = V_{CC}$ during all ICC measurements.*
7. *Clock frequency 66 Mhz and in Continuous Mode.*
8. *For I_{CC6}, when $V_{IH} = V_{IO}$, $V_{IL} = V_{SS}$.*

11.4 Capacitance

Table 11.2 Capacitance

Symbol	Description	Test Condition		Minimum	Typical	Maximum	Unit
C_{IN}	Input Capacitance	$V_{IN} = 0$	Die	0.30	0.40	0.50	pF
			Package	0.75	1.00	1.25	pF
C_{OUT}	Output Capacitance	$V_{OUT} = 0$	Die	0.60	0.80	1.00	pF
			Package	0.90	1.20	1.50	pF
C_{IN2}	Control Pin Capacitance	$V_{IN} = 0$	Die	0.30	0.40	0.50	pF
			Package	1.05	1.40	1.75	pF

Notes

Sampled, not 100% tested

Values are specified as follows: Min = Nominal -25%, Typ = Nominal, Max = Nominal + 25%

Total capacitance can be calculated as a sum of die and package values

11.5 Test Conditions

Figure 11.3 Test Setup

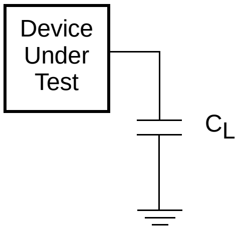
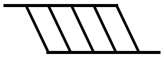
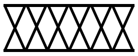
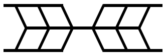


Table 11.3 Test Specifications

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

11.6 Key to Switching Waveforms

Waveform	Inputs	Outputs
	Steady	
	Changing from H to L	
	Changing from L to H	

Waveform	Inputs	Outputs
	Don't Care, Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High Impedance State (High-Z)

11.7 Switching Waveforms

Figure 11.4 Input Waveforms and Measurement Levels

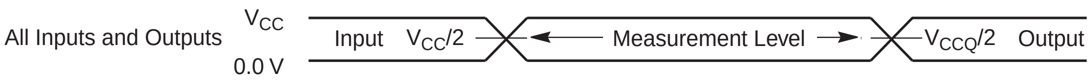
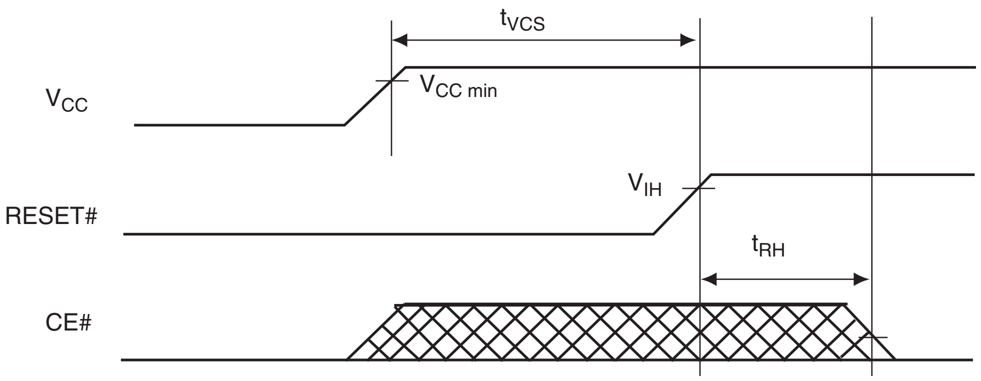


Table 11.4 V_{CC} Power-Up with No Ramp Rate Restriction

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

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

Figure 11.5 V_{CC} Power-Up Diagram



11.8 CLK Characterization

Table 11.5 CLK Characterization

Parameter	Description		66 MHz	83 MHz	Unit
f_{CLK}	CLK Frequency	Max	66	83	MHz
		Min	60 KHz in 8 word Burst, 120 KHz in 16 word Burst, 250 KHz in 32 word Burst, 1 MHz in Continuous Mode		
t_{CLK}	CLK Period	Min	15.1	12.5	ns
t_{CL}/t_{CH}	CLK Low/High Time	Min	0.40 t_{CLK}		ns
		Max	0.60 t_{CLK}		
t_{CR}	CLK Rise Time	Max	3.0	2.5	ns
t_{CF}	CLK Fall Time				

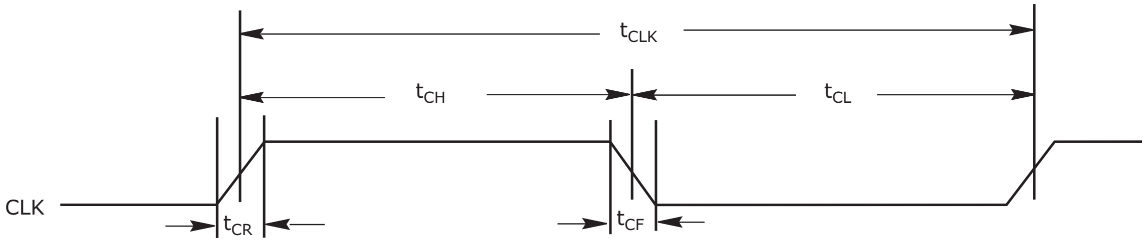


Figure 11.6 CLK Characterization

11.9 AC Characteristics

11.9.1 Synchronous/Burst Read

Table 11.6 Synchronous/Burst Read

Parameter		Description		66 MHz	83 MHz	Unit
JEDEC	Standard					
	t_{IACC}	Synchronous Access Time	Max	80(2)		ns
	t_{BACC}	Burst Access Time Valid Clock to Output Delay	Max	11.2	9	ns
	t_{ACS}	Address Setup Time to CLK (1)	Min	4		ns
	t_{ACH}	Address Hold Time from CLK (1)	Min	6	5	ns
	t_{BDH}	Data Hold Time	Min	3	3	ns
	t_{RDY}	Chip Enable to RDY Active	Max	10		ns
	t_{OE}	Output Enable to RDY Low	Max	9	9	ns
	t_{CEZ}	Chip Enable to High-Z	Max	10	10	ns
	t_{OEZ}	Output Enable to High-Z	Max	10	10	ns
	t_{CES}	CE# Setup Time to CLK	Min	4		ns
	t_{RACC}	Ready Access Time from CLK	Max	11.2	9	ns
	t_{CAS}	CE# Setup Time to AVD#	Min	0		ns
	t_{AVDS}	AVD# Low to CLK Setup Time	Min	5		ns
	t_{AVDH}	AVD# Hold Time from CLK	Min	3		ns
	t_{AVD0}	AVD# High to OE# Low	Min	0		ns
	t_{AVD}	AVD# Pulse	Min	6		ns

Notes

1. Addresses are latched on the rising edge of CLK

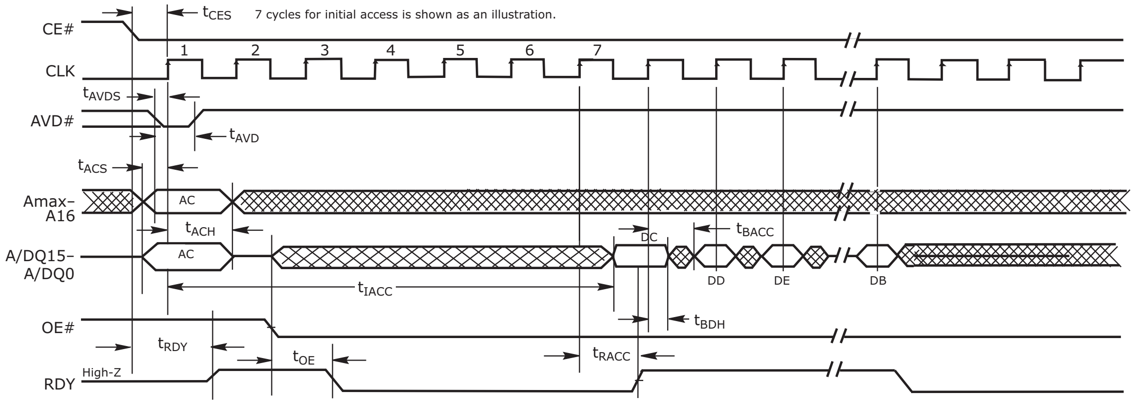
2. Synchronous Access Time is calculated using the formula (#of WS – 1)(clock period) + (t_{BACC} or Clock to Out)*

3. Not 100% tested for t_{CEZ}, t_{OEZ}.

Table 11.7 Synchronous Wait State Requirements

Max Frequency	Wait State Requirement
Frequency ≤ 14 MHz	2
14 < Frequency ≤ 27MHz	3
27 MHz < Frequency ≤ 40 MHz	4
40 MHz < Frequency ≤ 54 MHz	5
54 MHz < Frequency ≤ 66 MHz	6
66 MHz < Frequency ≤ 83 MHz	8

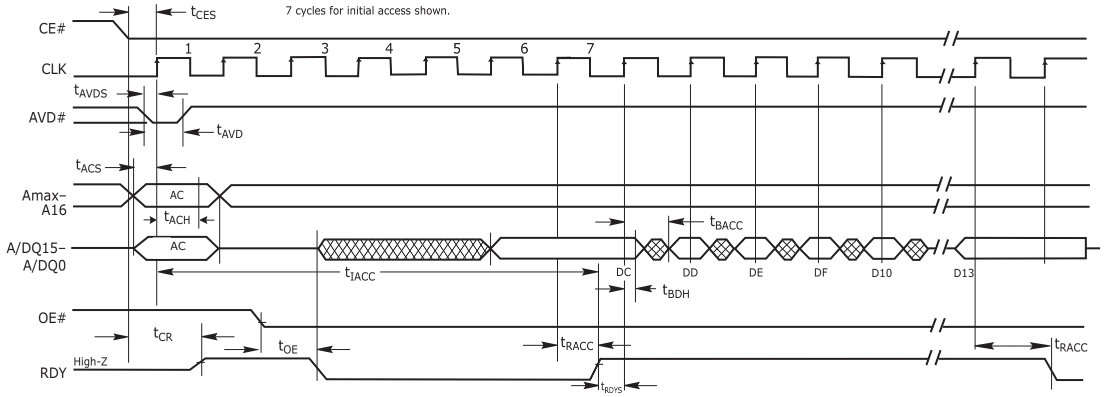
Figure 11.7 8-Word Linear Synchronous Single Data Rate Burst with Wrap Around



Notes

1. *Figure shows for illustration the total number of wait states set to seven cycles.*
2. *The device is configured synchronous single data rate mode and RDY active with data.*
3. *CE# (High) drives the RDY to High-Z while OE# (High) drives the A/DQ15 – A/DQ0 pins to High-Z.*

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



Notes

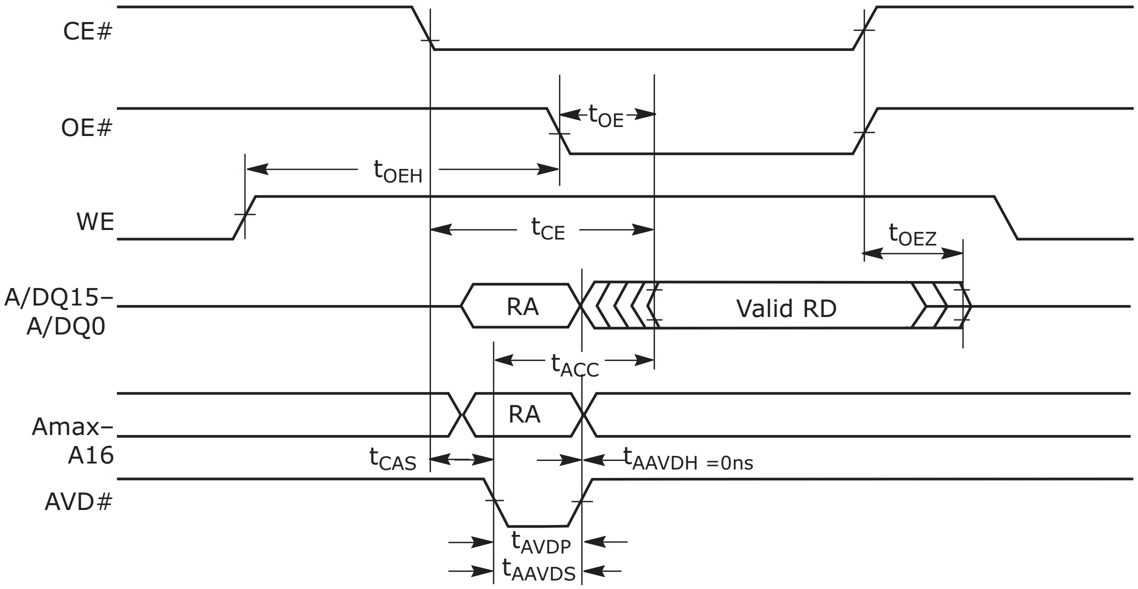
1. *Figure shows for illustration the total number of wait states set to seven cycles.*
2. *The device is configured synchronous single data rate mode and RDY active with data.*
3. *CE# (High) drives the RDY to High-Z while OE# (High) drives the A/DQ15 – A/DQ0 pins to High-Z.*

11.9.2 Asynchronous Mode Read

Table 11.8 Asynchronous Mode Read

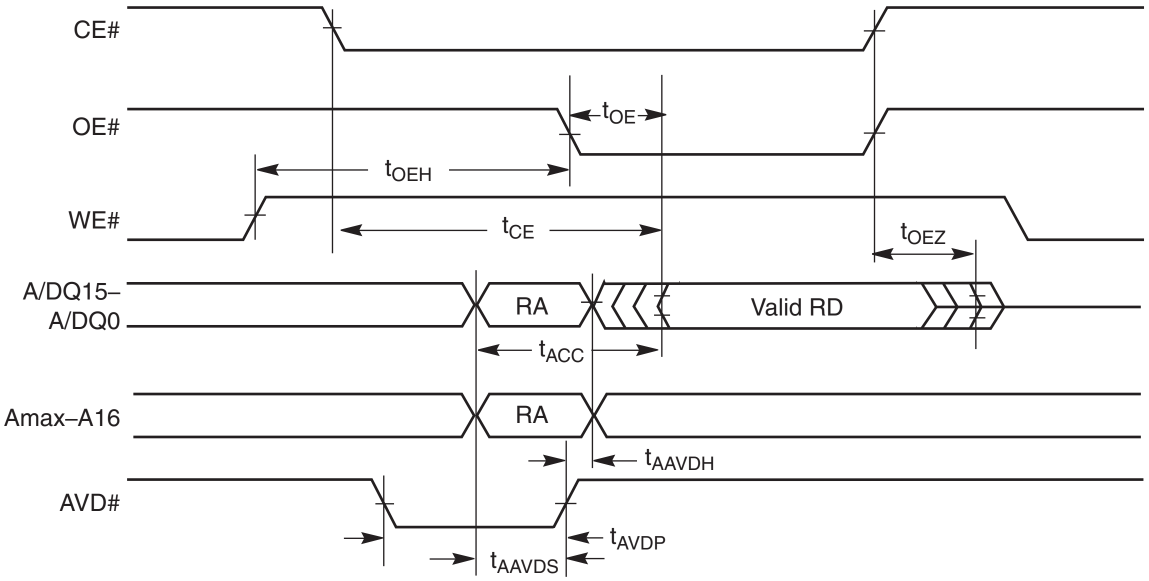
Parameter		Description			66 MHz	83 MHz	Unit
JEDEC	Standard						
	t_{CE}	Access Time from CE# Low		Typ	83		ns
	t_{ACC}	Asynchronous Access Time		Max	80		ns
	t_{AVDP}	AVD# Low Time		Min	7.5		ns
	t_{AAVDS}	Address Setup Time to Rising Edge of AVD#		Min	5		ns
	t_{AAVDH}	Address Hold Time from Rising Edge of AVD#		Min	3.5		ns
	t_{OE}	Output Enable to Output Valid		Max	9	9	ns
	t_{OEH}	Output Enable Hold Time	Read	Min	0		ns
			Toggle and Data# Polling	Min	10	10	ns
	t_{OEZ}	Output Enable to High-Z		Max	10	10	ns
	t_{CAS}	CE# Setup Time to AVD#		Min	0		ns

Figure 11.9 Asynchronous Mode Read with Latched Addresses



Note
RA = Read Address, RD = Read Data.

Figure 11.10 Asynchronous Mode Read



Note

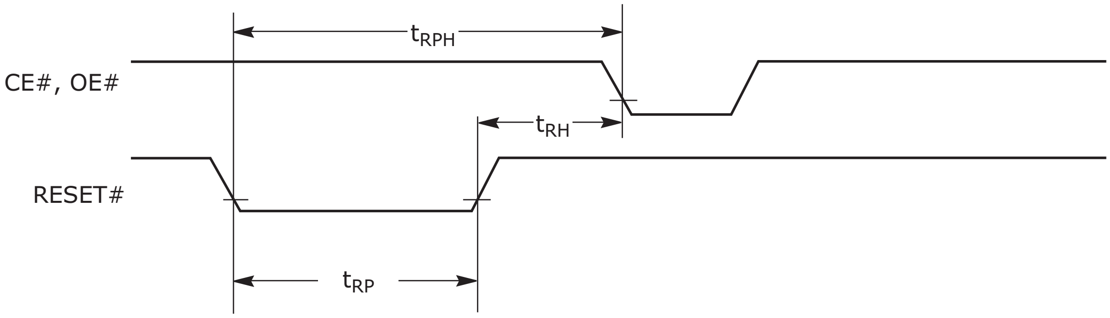
RA = Read Address, RD = Read Data.

11.9.3 Hardware Reset (RESET#)

Table 11.9 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	µs

Figure 11.11 Reset Timings



11.9.4 Erase/Program Timing

Table 11.10 Erase/Program Timing

Parameter		Description		66 MHz	83 MHz	Unit
JEDEC	Standard					
t_{AVAV}	t_{WC}	Write Cycle Time (1)	Min	60		ns

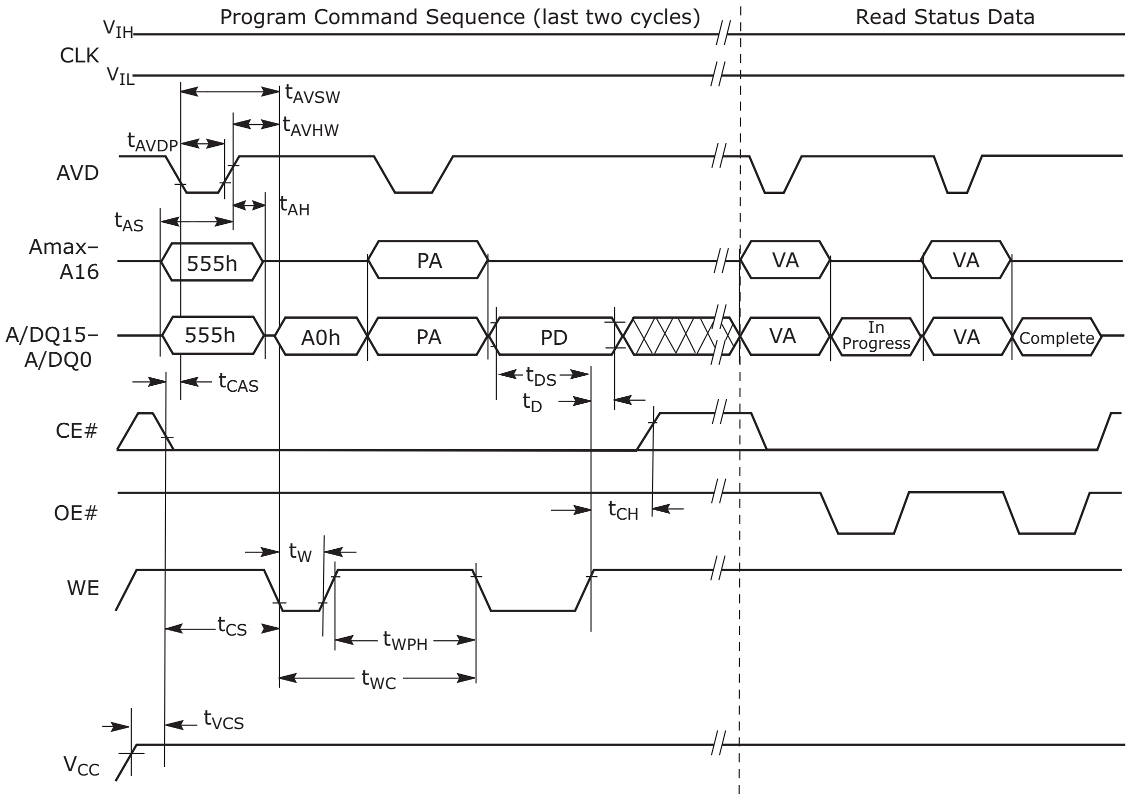
Table 11.10 Erase/Program Timing

Parameter		Description			66 MHz	83 MHz	Unit
JEDEC	Standard						
t_{AVWL}	t_{AS}	Address Setup Time (2)	Synchronous	Min	4		ns
			Asynchronous		4		ns
t_{WLAX}	t_{AH}	Address Hold Time (2)	Synchronous	Min	3.5		
			Asynchronous		3.5		
	t_{AVDP}	AVD# Low Time		Min	6		ns
t_{DVWH}	t_{DS}	Data Setup Time		Min	20		ns
t_{WHDX}	t_{DH}	Data Hold Time		Min	0		ns
t_{GHWL}	t_{GHWL}	Read Recovery Time Before Write		Min	0		ns
	t_{CAS}	CE# Setup Time to AVD#		Min	0		ns
t_{WHEH}	t_{CH}	CE# Hold Time		Min	0		ns
t_{WLWH}	t_{WP}	Write Pulse Width		Min	25		ns
t_{WHWL}	t_{WPH}	Write Pulse Width High		Min	20		ns
	$t_{SR/W}$	Latency Between Read and Write Operations		Min	0		ns
	t_{VID}	V_{PP} Rise and Fall Time		Min	500		ns
	t_{VIDS}	V_{PP} Setup Time (During Accelerated Programming)		Min	1		µs
t_{ELWL}	t_{CS}	CE# Setup Time to WE#		Min	4		ns
	t_{AVSW}	AVD# Setup Time to WE#		Min	6		ns
	t_{AVHW}	AVD# Hold Time to WE#		Min	4		ns
	t_{SEA}	Sector Erase Accept Time out		Min	50		µs
	t_{ESL}	Erase Suspend Latency		Min	20		µs
	t_{PSL}	Program Suspend Latency		Min	20		µs
	t_{ASP}	Toggle Time During Erase within a Protected Sector		Typ	280		µs
	t_{PSP}	Toggle Time During Programming Within a Protected Sector		Typ	1		µs
	t_{ERS}	Erase Resume to Erase Suspend		Min	30		µs
	t_{PRS}	Program Resume to Program Suspend		Min	30		µs

Notes

1. *Not 100% tested.*
2. *In asynchronous operation timing, addresses are latched on the rising edge of AVD#.*
3. *See Section 11.10, Erase and Programming Performance on page 80 for more information. Does not include the preprogramming time.*

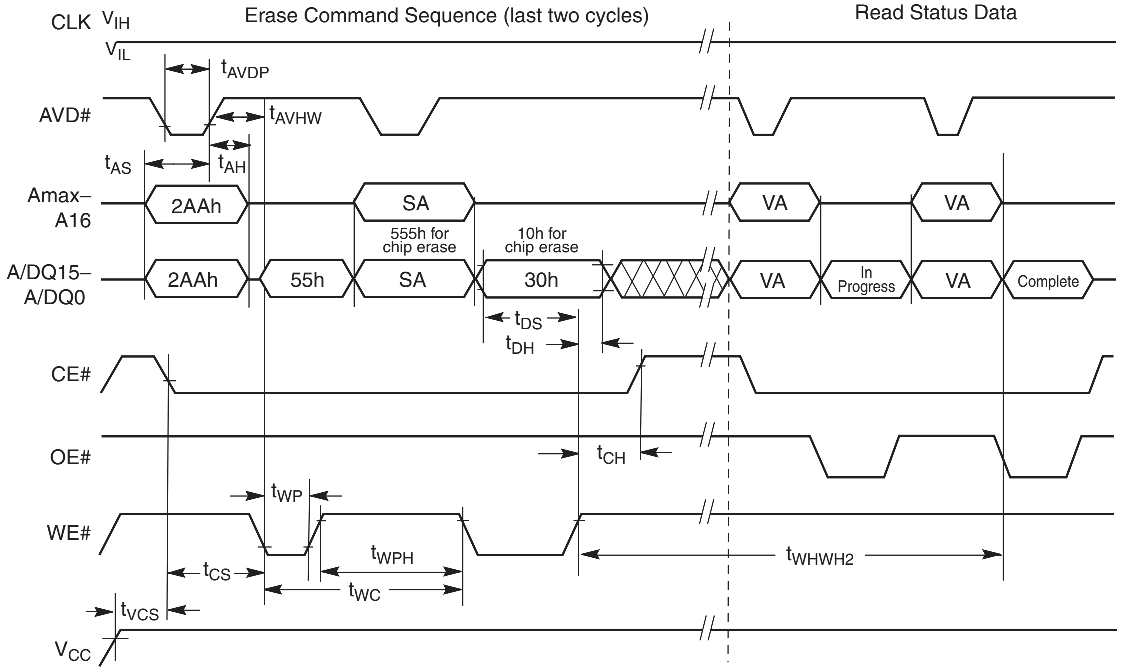
Figure 11.12 Asynchronous Program Operation Timings



Notes

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

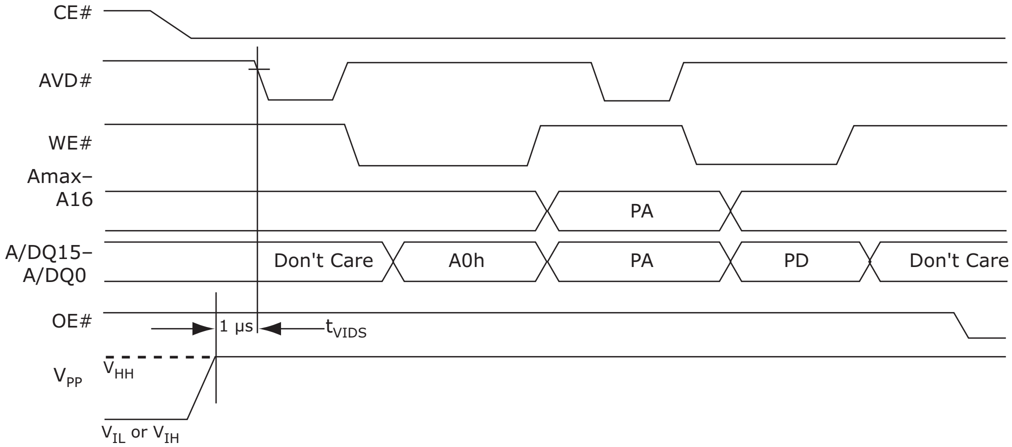
Figure 11.13 Chip/Sector Erase Command Sequence



Note

SA is the sector address for Sector Erase.

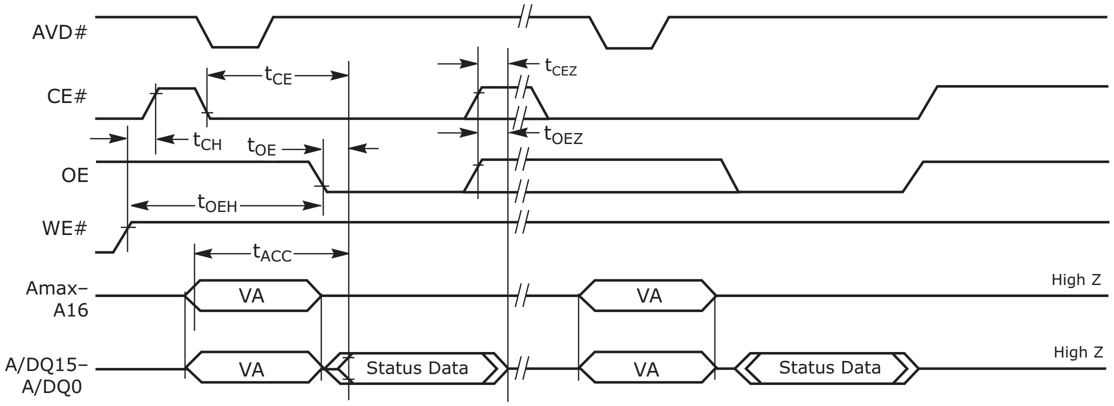
Figure 11.14 Accelerated Unlock Bypass Programming Timing



Note

Use setup and hold times from conventional program operation.

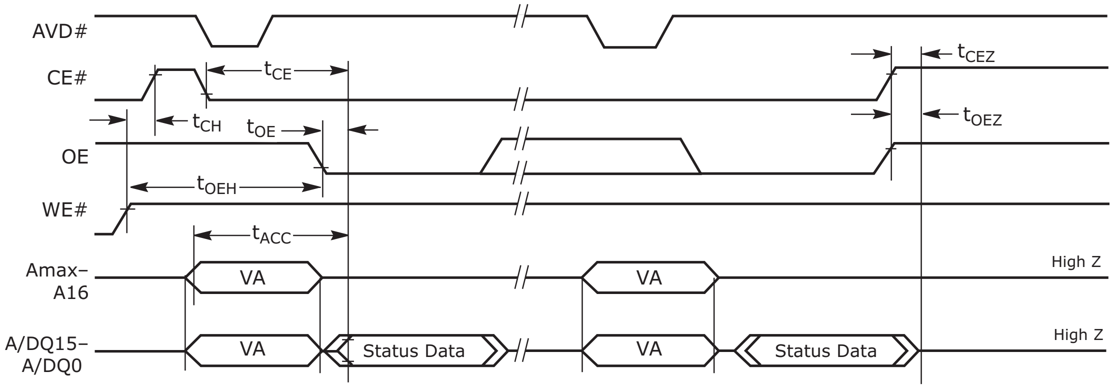
Figure 11.15 Data# Polling Timings (During Embedded Algorithm)



Notes

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

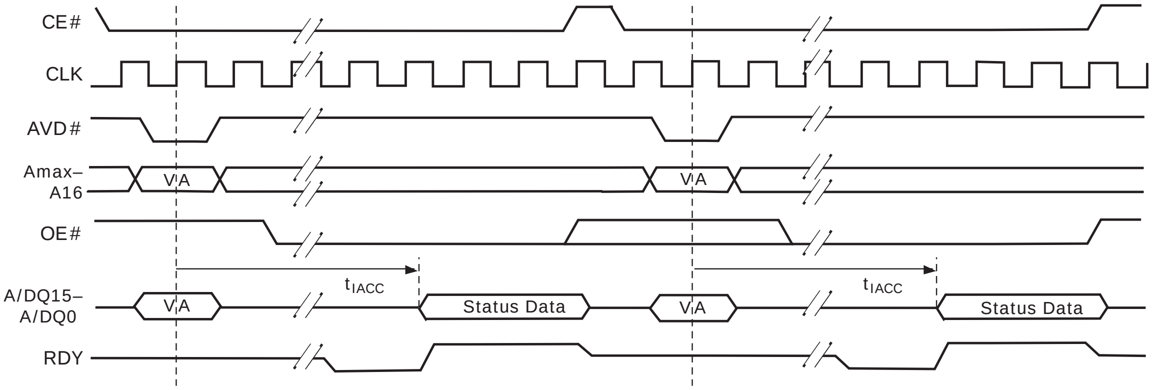
Figure 11.16 Toggle Bit Timings (During Embedded Algorithm)



Notes

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

Figure 11.17 Synchronous Data Polling Timings/Toggle Bit Timings



Notes

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

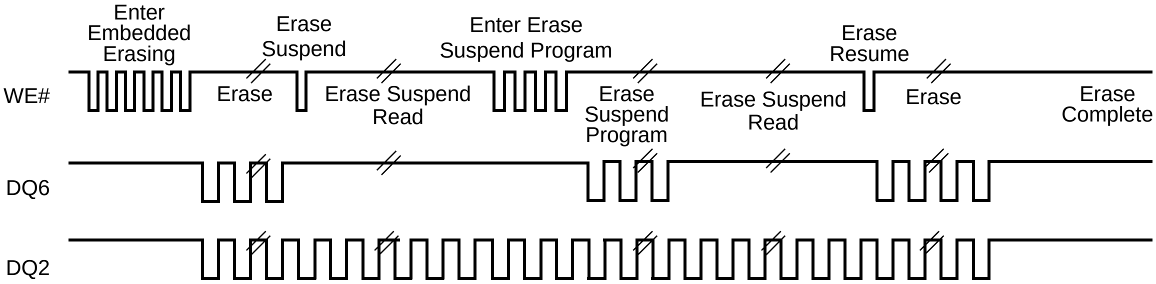
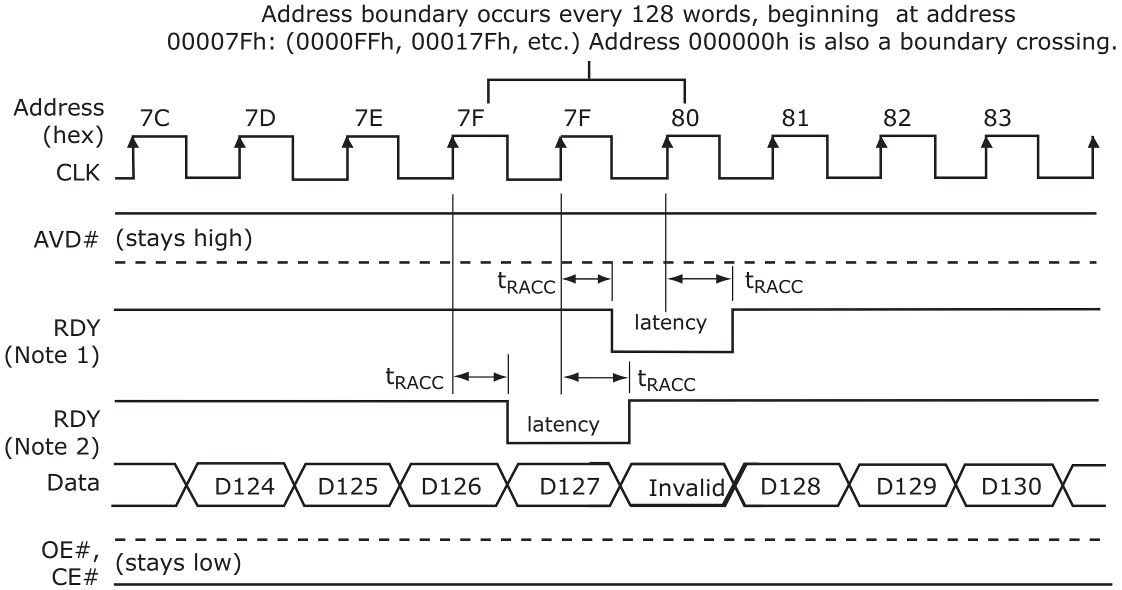


Figure 11.18 DQ2 vs. DQ6

Note

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

Figure 11.19 Latency with Boundary Crossing



Notes

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

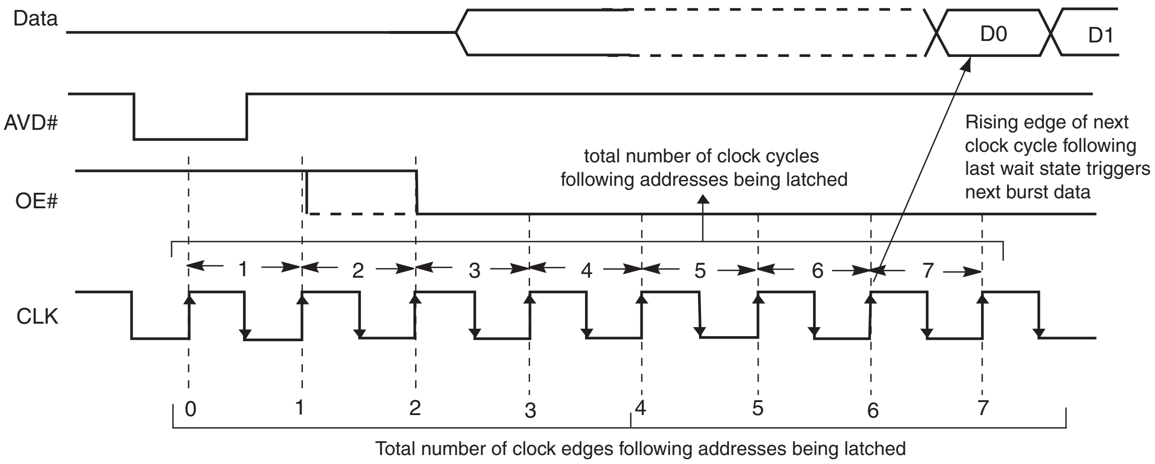


Figure 11.20 Wait State Configuration Register Setup

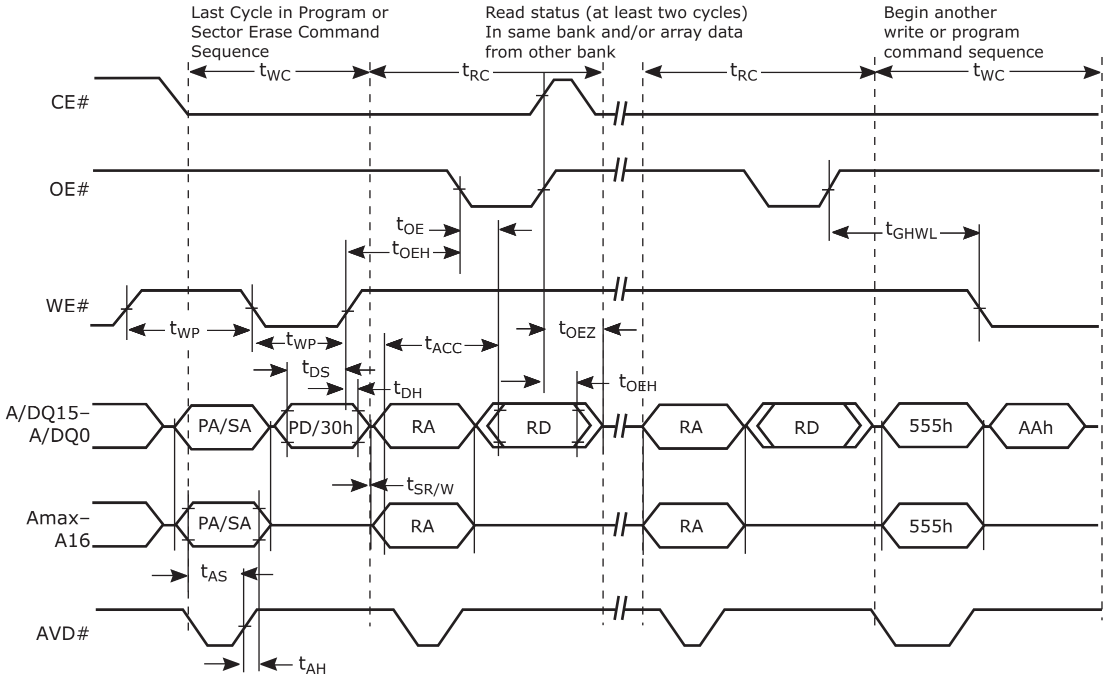
Table 11.11 Example of Programmable Wait States

CR1.0	Programmable Wait State (See Note)	0000	=	initial data is valid on the	2nd	rising CLK edge	after addresses are latched
CR0.13		0001			3rd		
CR0.12		0010			4th		
CR0.11		0011			5th		
		0100			6th		
		0101			7th		
		0110	=	Reserved			
		0111					
		1000	=	initial data is valid on the	8th	rising CLK edge	after addresses are latched
		1001			9th		
		⋮					
		1101	=	initial data is valid on the	13th	rising CLK edge	AVD# transition to V_{IH} (Default)
		⋮					
		1110	=	Reserved			
		1111					

Note

The addresses are latched by rising edge of CLK.

Figure 11.21 Back-to-Back Read/Write Cycle Timings



Note

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

11.10 Erase and Programming Performance

Table 11.12 Erase and Programming Performance

Parameter			Typ (1)	Max (2)	Unit	Comments
Sector Erase Time	64 Kword	V_{CC}	0.8	3.5	s	Excludes 00h programming prior to erasure (3)
	16 Kword	V_{CC}	0.15	2.0		
	64 Kword	V_{PP}	0.8	3.5		
	16 Kword	V_{PP}	0.15	2.0		
Sector Erase Time	64 Kword	V_{CC}	0.90	5.00		Includes 00h programming prior to erasure (3)
	16 Kword	V_{CC}	0.45	1.85		
	64 Kword	V_{PP}	0.70	3.75		
	16 Kword	V_{PP}	0.35	1.40		
Chip Erase Time		V_{CC}	77 (NS128P) 154 (NS256P) 306 (NS512P)	154 (NS128P) 308 (NS256P) 612 (NS512P)	s	
Word Programming Time		V_{CC}	40	400	µs	Excludes system level overhead (4)
		V_{PP}	24	240		
Effective Word Programming Time utilizing Program Write Buffer		V_{CC}	9.4	94	µs	
		V_{PP}	6	60		
Total 32-Word Buffer Programming Time		V_{CC}	300	3000		
		V_{PP}	192	1920		
Chip Programming Time (using 32 word buffer)		V_{CC}	78.6 (NS128P) 157.3 (NS256P) 314.6 (NS512P)	157.3 (NS128P) 314.6 (NS256P) 629.2 (NS512P)	s	Excludes system level overhead (4)
		V_{PP}	51 (NS128P) 101 (NS256P) 202 (NS512P)	102 (NS128P) 202 (NS256P) 404 (NS512P)		
Erase Suspend/Erase Resume		Min		20	µs	
Program Suspend/Program Resume		Min		20	µs	

Notes

1. *Typical program and erase times assume the following conditions: 25°C, 1.8 V V_{CC}, 10,000 cycles using checkerboard patterns.*
2. *Under worst case conditions of 90°C, V_{CC} = 1.70 V, 100,000 cycles.*
3. *In the pre-programming step of the Embedded Erase algorithm, all words are programmed to 00h before erasure.*
4. *System-level overhead is the time required to execute the two- or four-bus-cycle sequence for the program command. See Table 12.1, Memory Array Commands on page 81 and Table 12.2, Sector Protection Commands on page 83for further information on command definitions.*

12. Appendix

This section contains information relating to software control or interfacing with the Flash device. For additional information and assistance regarding software, see www.spansion.com.

Table 12.1 Memory Array Commands

Command Sequence (Notes)		Cycles	Bus Cycles (1- 6)											
			First		Second		Third		Fourth		Fifth		Sixth	
			Addr	Data (19)	Addr	Data (19)	Addr	Data (19)	Addr	Data (19)	Addr	Data (19)	Addr	Data (19)
Asynchronous Read (7)		1	RA	RD										
Reset (8)		1	XXX	F0										
Autoselect (9)	Manufacturer ID	4	555	AA	2AA	55	(BA) 555	90	(BA) X00	0001				
	Device ID (10)	6	555	AA	2AA	55	(BA) 555	90	(BA) X01	3x7E	(BA)X 0E	(10)	(BA) X0F	(10)
	Indicator Bits	4	555	AA	2AA	55	(BA) 555	90	(BA) X07	(12)				
	Sector Unlock/Lock Verify (11)	4	555	AA	2AA	55	(SA) 555	90	(SA) X02	0000/ 0001				
	Revision ID	4	555	AA	2AA	55	(BA) 555	90	(BA) X03					
Single Word Program		4	555	AA	2AA	55	555	A0	PA	Data				
Write to Buffer (17)		6	555	AA	2AA	55	SA	25	SA	WC	PA	PD	WBL	PD
Write Buffer to Flash		1	SA	29										
Write to Buffer Abort Reset (10)		3	555	AA	2AA	55	555	F0						
Chip Erase		6	555	AA	2AA	55	555	80	555	AA	2AA	55	555	10
Sector Erase		6	555	AA	2AA	55	555	80	555	AA	2AA	55	SA	30
Program/Erase Suspend (15)		1	BA	B0										
Program/Erase Resume (16)		1	BA	30										
Set Configuration Register (21, 22, 24)		5	555	AA	2AA	55	555	D0	X00	CR0	X01	CR1		
Read Configuration Register		4	555	AA	2AA	55	555	C6	X0 (0 or 1)	CR (0 or 1)				
CFI Query (17)		1	(BA) 55	98										
Unlock Bypass Mode (23)	Unlock Bypass Entry (18)	3	555	AA	2AA	55	555	20						
	Unlock Bypass Program (13), (14)	2	XX	A0	PA	PD								
	Unlock Bypass Sector Erase (13), (14)	2	XX	80	SA	30								
	Unlock Bypass Erase (13), (14)	2	XX	80	XXX	10								
	Unlock Bypass CFI (13), (14)	1	XX	98										
	Unlock Bypass Reset	2	XX	90	XXX	00								

Legend

X = Don’t care.

RA = Read Address.

RD = Read Data.

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

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

SA = Sector Address. NS128P = A22 – A14; NS256P = A23 – A14.

BA = Bank Address. NS128P = A22 – A20, and A19; NS064P = A21, A20 – A18; NS256P = A23 – A20.CR = Configuration Register data bits D15 – D0.

WBL = Write Buffer Location. Address must be within the same write buffer page as PA.

WC = Word Count. Number of write buffer locations to load minus 1.

Notes

1. *See Table 7.1 for description of bus operations.*
2. *All values are in hexadecimal.*
3. *Except for the following, all bus cycles are write cycle: read cycle, fourth through sixth cycles of the Autoselect commands, fourth cycle of the configuration register verify and password verify commands, and any cycle reading at RD(0) and RD(1).*
4. *Data bits DQ15 – DQ8 are don't care in command sequences, except for RD, PD, WD, PWD, and PWD3 – PWD0.*
5. *Unless otherwise noted, address bits Amax – A14 are don't cares.*
6. *Writing incorrect address and data values or writing them in the improper sequence may place the device in an unknown state. The system must write the reset command to return the device to reading array data.*
7. *No unlock or command cycles required when bank is reading array data.*
8. *The Reset command is required to return to reading array data (or to the erase-suspend-read mode if previously in Erase Suspend) when a bank is in the autoselect mode, or if DQ5 goes high (while the bank is providing status information) or performing sector lock/unlock.*
9. *The fourth cycle of the autoselect address is a read cycle. The system must provide the bank address.*
10. *(BA) + 0Eh ----> For NS128 = 43h, NS256 = 41h, NS512 = 3Fh (BA) + 0Fh ----> For NS128/256/512 = 00h*
11. *The data is 0000h for an unlocked sector and 0001h for a locked sector*
12. *See Table 7.12, Autoselect Addresses on page 33.*
13. *The Unlock Bypass command sequence is required prior to this command sequence.*
14. *The Unlock Bypass Reset command is required to return to reading array data when the bank is in the unlock bypass mode.*
15. *The system may read and program in non-erasing sectors, or enter the autoselect mode, when in the Erase Suspend mode. The Program/ Erase Suspend command is valid only during a program/ erase operation, and requires the bank address.*
16. *The Program/Erase Resume command is valid only during the Program/Erase Suspend mode, and requires the bank address.*
17. *The total number of cycles in the command sequence is determined by the number of words written to the write buffer. The maximum number of cycles in the command sequence is 37.*
18. *Write Buffer Programming can be initiated after Unlock Bypass Entry.*
19. *Data is always output at the rising edge of clock.*
20. *Do not enter wrong address or data cycles.*
21. *Do not use 0x30 for CR data (otherwise in the erase suspend --> CR read or set sequence, the device will go into erase resume instead of CR read or set).*
22. *Software reset is needed after CR read (otherwise the device is still in CR read mode).*
23. *When device is in Unlock Bypass mode, do not enter another command before Unlock Bypass reset command is issued).*
24. *Configuration Registers can not be programmed out of order. CR0 must be programmed prior to CR01 otherwise the configuration registers retain their previous settings.*

Table 12.2 Sector Protection Commands (Sheet 1 of 2)

Command Sequence (Notes)		Cycles	Bus Cycles (Notes 1 - 6)													
			First		Second		Third		Fourth		Fifth		Sixth		Seventh	
			Addr	Data (10)	Addr	Data (10)	Addr	Data (10)	Addr	Data (10)	Addr	Data (10)	Addr	Data (10)	Addr	Data (10)
Secured Silicon	Entry (5)	3	555	AA	2AA	55	555	88								
	Program	4	555	AA	2AA	55	555	A0	PA	PD						
	Read	1	00	data												
	Exit (7)	4	555	AA	2AA	55	555	90	XX	00						
Lock Register	Register Command Set Entry (5)	3	555	AA	2AA	55	555	40								
	Register Bits Program (6)	2	XX	A0	00	data										
	Register Bits Read	1	00	data												
	Register Command Set Exit (7)	2	XX	90	XX	00										
Password	Protection Command Set Entry	3	555	AA	2AA	55	555	60								
	Program (9)	2	XX	A0	00/01/02/03	PWD0/1/2/3/										
	Read Password (10)	4	00	PWD0	01	PWD1	02	PWD2	03	PWD3						
	Unlock (9)	7	00	25	00	03	00	PWD0	01	PWD1	02	PWD2	03	PWD3	00	29
	Protection Command Set Exit	2	XX	90	XX	00										
PPB	Non-Volatile Sector Protection Command Set Entry (5)	3	555	AA	2AA	55	(BA) 555	C0								
	Program	2	XX	A0	(BA) SA	00										
	All Erase (8)	2	XX	80	SA0	30										
	Status Read	1	(BA) SA	RD(0)												
	Non-Volatile Sector Protection Command Set Exit (7)	2	XX	90	XX	00										
PPB Lock Bit	Global Volatile Sector Protection Freeze Command Set Entry (5)	3	555	AA	2AA	55	555	50								
	Set	2	XX	A0	XX	00										
	Status Read	1	XX	RD(0)												
	Global Volatile Sector Protection Freeze Command Set Exit (7)	2	XX	90	XX	00										
DYB	Volatile Sector Protection Command Set Entry (5)	3	555	AA	2AA	55	(BA) 555	E0								
	Set	2	XX	A0	(BA) SA	00										
	Clear	2	XX	A0	(BA) SA	01										
	Status Read	1	(BA) SA	RD(0)												
	Volatile Sector Protection Command Set Exit (7)	2	XX	90	XX	00										

Table 12.2 Sector Protection Commands (Sheet 2 of 2)

Command Sequence (Notes)		Cycles	Bus Cycles (Notes 1 - 6)													
			First		Second		Third		Fourth		Fifth		Sixth		Seventh	
			Addr	Data (10)	Addr	Data (10)	Addr	Data (10)	Addr	Data (10)	Addr	Data (10)	Addr	Data (10)	Addr	Data (10)
Accelerated	Program	2	555	A0	PA	Data										
	Sector Erase	2	555	80	SA	30										
	Asynchronous Read	1	RA	RD												
	Write to Buffer	4	SA	25	SA	WC	PA	PD	WBL	PD						
	Program Buffer to Flash	1	SA	29												

Legend

X = Don't care

RA = Read Address.

RD = Read Data.

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

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

SA = Sector Address. NS128P = A22 – A14, NS256P = A23 – A14.

BA = Bank Address. NS128P = A22 – A20, and A19; NS256P = A23 – A20.

CR = Configuration Register data bits D15 – D0.

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

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

PWD = Password Data.

RD(0) = DQ0 protection indicator bit. If protected, DQ0 = 0, if unprotected, DQ0 = 1.

WBL = Write Buffer Location. Address must be within the same write buffer page as PA.

WC = Word Count. Number of write buffer locations to load minus 1.

Notes

1. *See Table 7.1 for description of bus operations.*
2. *All values are in hexadecimal.*
3. *Except for the following, all bus cycles are write cycle: read cycle, fourth through sixth cycles of the Autoselect commands, fourth cycle of the configuration register verify and password verify commands, and any cycle reading at RD(0) and RD(1).*
4. *Data bits DQ15 – DQ8 are don't care in command sequences, except for RD, PD, WD, PWD, and PWD3 – PWD0.*
5. *Unless otherwise noted, address bits Amax – A14 are don't cares.*
6. *Writing incorrect address and data values or writing them in the improper sequence may place the device in an unknown state. The system must write the reset command to return the device to reading array data.*
7. *No unlock or command cycles required when bank is reading array data.*
8. *The data is 0000h for an unlocked sector and 0001h for a locked sector.*
9. *The Exit command must be issued to reset the device into read mode, otherwise the device hangs.*
10. *Data is always output at the rising edge of clock.*

12.1 Common Flash Memory Interface

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

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

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

```
/* Example: CFI Entry command */
  *( (UINT16 *)bank_addr + 0x0055 = 0x0098;   /* write CFI entry command         */

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

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

Table 12.3 CFI Query Identification String

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

Table 12.4 System Interface String

Addresses	Data	Description
1Bh	0017h	V_{CC} Min. (write/erase) D7 – D4: volt, D3 – D0: 100 millivolt
1Ch	0019h	V_{CC} Max. (write/erase) D7 – D4: volt, D3 – D0: 100 millivolt
1Dh	0000h	V_{PP} Min. voltage (00h = no V_{PP} pin present)
1Eh	0000h	V_{PP} Max. voltage (00h = no V_{PP} pin present)
1Fh	0005h	Typical Program Time per single word write 2^N µs (for example, 30 µs)
20h	0009h	Typical Program Time using buffer 2^N µs (for example, 300us) (00h = not supported)
21h	000Ah	Typical time for sector erase 2^N ms
22h	0000h	Typical time for full chip erase 2^N ms (00h = not supported)
23h	0003h	Max. Program Time per single word [2^N times typical value]
24h	0002h	Max. Program Time using buffer [2^N times typical value]
25h	0002h	Max. time for sector erase [2^N times typical value]
26h	0000h	Max. time for full chip erase [2^N times typical value] (00h = not supported)

Table 12.5 Device Geometry Definition (Sheet 1 of 2)

Addresses	Data	Description
27h	0018h (NS128P) 0019h (NS256P) 001Ah (NS512P)	Device Size = 2^N byte
28h 29h	0001h 0000h	Flash Device Interface 0h=x8; 1h=x16; 2h=x8/x16; 3h=x32 [lower byte] [upper byte] (00h = not supported)
2Ah 2Bh	0006h 0000h	Max. number of bytes in multi-byte buffer write = 2^N [lower byte] [upper byte] (00h = not supported)
2Ch	0002h (NS128P) 0002h (NS256P) 0001h (NS512P)	Number of Erase Block Regions within device 01h = Uniform Sector; 02h = Boot + Uniform; 03h = Boot + Uniform + Boot
2Dh	007Eh (NS128P) 00FEh (NS256P) 01FFh (NS512P)	Erase Block Region 1 Information (Large Sector Section) [lower byte] – Number of sectors. 00h=1 sector; 01h=2 sectors... 03h=4 sectors [upper byte] [lower byte] – Equation =>(n = Density in Bytes of any 1 sector/256)h [upper byte]
2Eh	0000h	
2Fh	0000h	
30h	0002h	

Table 12.5 Device Geometry Definition (Sheet 2 of 2)

Addresses	Data	Description
31h	0003h (NS128P) 0003h (NS256P) 0000h (NS512P)	Erase Block Region 2 Information (Small Sector Section) [lower byte] – Number of sectors. [upper byte] [lower byte] – Equation =>(n = Density in Bytes of any 1 sector/256)h [upper byte]
32h	0000h	
33h	0080h (NS128P) 0080h (NS256P) 0000h (NS512P)	
34h	0000h	
35h 36h 37h 38h	0000h 0000h 0000h 0000h	Erase Block Region 3 Information [lower byte] – Number of sectors. 00h=1 sector; 01h=2 sectors... 03h=4 sectors [upper byte] [lower byte] – Equation =>(n = Density in Bytes of any 1 sector/256)h [upper byte]
39h 3Ah 3Bh 3Ch	0000h 0000h 0000h 0000h	Erase Block Region 4 Information

Table 12.6 Primary Vendor-Specific Extended Query (Sheet 1 of 2)

Addresses	Data	Description
40h 41h 42h	0050h 0052h 0049h	Query-unique ASCII string *PRI*
43h	0031h	Major CFI version number, ASCII
44h	0034h	Minor CFI version number, ASCII
45h	0014h	Address Sensitive Unlock (Bits 1 – 0) 00b = Required, 01b = Not Required Silicon Technology (Bits 5 – 2) 0011b = 130 nm; 0100b = 110 nm; 0101b = 90 nm 001010b = 000Ah
46h	0002h	Erase Suspend 0 = Not Supported, 1 = To Read Only, 2 = To Read & Write
47h	0001h	Sector Protection per Group 0 = Not Supported, X = Number of sectors in per group
48h	0000h	Sector Temporary Unprotect 00 = Not Supported, 01 = Supported
49h	0008h	Sector Protect/Unprotect scheme 08h = Advanced Sector Protection; 07h = New Sector Protection Scheme
4Ah	0078h (NS128P) 00F0h (NS256P) 01E0h (NS512P)	Simultaneous Operation Number of Sectors in all banks except bank0
4Bh	0001h	Burst Mode Type 00 = Not Supported, 01 = Supported
4Ch	0000h	Not supported
4Dh	0085h	V_{PP} (Acceleration) Supply Minimum 00h = Not Supported, D7 – D4: Volt, D3 – D0: 100 mV
4Eh	0095h	V_{PP} (Acceleration) Supply Maximum 00h = Not Supported, D7 – D4: Volt, D3 – D0: 100 mV
4Fh	0003h (NS128P) 0003h (NS256P) 0005h (NS512P)	Write Protect Function 00h = No Boot, 01h = Dual Boot, 02h = Bottom Boot, 03h = Top Boot, 04h = Uniform Bottom, 05h = Uniform Top, 06h = All Sectors
50h	0001h	Program Suspend. 00h = not supported
51h	0001h	Unlock Bypass 00 = Not Supported, 01=Supported
52h	0008h	Secured Silicon Sector (Customer OTP Area) Size 2^N bytes

Table 12.6 Primary Vendor-Specific Extended Query (Sheet 2 of 2)

Addresses	Data	Description
53h	0014h	Hardware Reset Low Time-out during an embedded algorithm to read mode Maximum 2^N ns (for example, 10 µs => n=14)
54h	0014h	Hardware Reset Low Time-out not during an embedded algorithm to read mode Maximum 2^N ns (for example, 10 µs => n=14)
55h	0005h	Erase Suspend Time-out Maximum 2^N µs
56h	0005h	Program Suspend Time-out Maximum 2^N µs
57h	0010h	Bank Organization: X = Number of banks
58h	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 0 Region Information. X = Number of sectors in bank
59h	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 1 Region Information. X = Number of sectors in bank
5Ah	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 2 Region Information. X = Number of sectors in bank
5Bh	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 3 Region Information. X = Number of sectors in bank
5Ch	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 4 Region Information. X = Number of sectors in bank
5Dh	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 5 Region Information. X = Number of sectors in bank
5Eh	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 6 Region Information. X = Number of sectors in bank
5Fh	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 7 Region Information. X = Number of sectors in bank
60h	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 8 Region Information. X = Number of sectors in bank
61h	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 9 Region Information. X = Number of sectors in bank
62h	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 10 Region Information. X = Number of sectors in bank
63h	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 11 Region Information. X = Number of sectors in bank
64h	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 12 Region Information. X = Number of sectors in bank
65h	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 13 Region Information. X = Number of sectors in bank
66h	0008h (NS128P) 0010h (NS256P) 0020h (NS512P)	Bank 14 Region Information. X = Number of sectors in bank
67h	000Bh (NS128P) 0013h (NS256P) 0020h (NS512P)	Bank 15 Region Information. X = Number of sectors in bank

13. Revision History

Spansion Publication Number: 002-01103

Section	Description
Revision A (June 29, 2006)	
	Initial release
Revision A1 (February 20, 2007)	
Global	The t_{AVDS} specification is changed from 4 ns to 5 ns The wait state for 83 MHz is changed to 8 I_{CC3}(Max) is changed to 70 µA and I_{CC6}(Max) is changed to 40 µA V_{IL} (Min) is changed to -0.2 V t_{OE} (Max) in both Asynchronous & Synchronous modes is changed to 9 ns across all frequencies t_{CEZ} (Max) is changed to 10 ns across all frequencies t_{OEZ} (Max) in both Asynchronous & Synchronous modes is changed to 10 ns across all frequencies t_{ACH}(Min) is changed to 6 ns (66 MHz) and 5 ns (83 MHz and 108 MHz) t_{RDY}(Max) is changed to 10 ns t_{RACC}(Max) is changed to 7.6 ns for 108 MHz t_{OEH}(Min) in Asynchronous mode is changed to 10 ns for 108 MHz Erase and Programing Performance table is updated t_{CE} in Asynchronous mode is changed to 83ns
Revision A2 (June 6, 2007)	
Timing Diagrams	Revised Fig 10.13 Chip/Sector Erase Command Sequence to include t_{AVHW} parameter
Revision A3 (June 14, 2007)	
AC Characteristics	Revised t_{BACC} @ 108 MHz to 7.0 ns instead of 7.6 ns
Revision A4 (December 13, 2007)	
Global	Removed 108 MHz speed offering and corresponding details such as OPN, Valid combination, Product Selector Guide and specifications
Revision A5 (February 13, 2008)	
Capacitance	Added Section 10.4 for product capacitance
Revision A6 (March 19, 2008)	
AC Characteristics	Revised Figure 10.9 to correct the starting edge of t_{AAVDS}
Revision A7 (September 22, 2009)	
Performance Characteristics	Revised Typical Program & Erase Times values
Revision A8 (September 8, 2011)	
Input/Output Descriptions	Updated table: NC, DNU, RFU descriptions
Special Handling Instructions for FBGA Package	Updated figure 64-Ball Fine-Pitch Grid Array, S29NS512P: Revised ball labels to be consistent with Input/Output descriptions

Document History Page

Document Title: S29NS512P, S29NS256P, S29NS128P 512/256/128 Mb (32/16/8 M x 16 bit), 1.8 V MirrorBit Flash Memory
Document Number: 002-01103

Rev.	ECN No.	Orig. of Change	Submission Date	Description of Change
**	–	WIOB	06/29/2006 to 09/08/2011	Spansion revision A: Initial release Spansion revision A1: The tAVDS specification is changed from 4 ns to 5 ns The wait state for 83 MHz is changed to 8 ICC3(Max) is changed to 70 µA and ICC6(Max) is changed to 40 µA VIL (Min) is changed to -0.2 V tOE (Max) in both Asynchronous & Synchronous modes is changed to 9 ns across all frequencies tCEZ (Max) is changed to 10 ns across all frequencies tOEZ (Max) in both Asynchronous & Synchronous modes is changed to 10 ns across all frequencies tACH(Min) is changed to 6 ns (66 MHz) and 5 ns (83 MHz and 108 MHz) tRDY(Max) is changed to 10 ns tRACC(Max) is changed to 7.6 ns for 108 MHz tOEH(Min) in Asynchronous mode is changed to 10 ns for 108 MHz Erase and Programing Performance table is updated tCE in Asynchronous mode is changed to 83ns Spansion revision A2: Revised Fig 10.13 Chip/Sector Erase Command Sequence to include tAVHW parameter Spansion revision A3: Revised tBACC @ 108 MHz to 7.0 ns instead of 7.6 ns Spansion revision A4: Removed 108 MHz speed offering and corresponding details such as OPN, Valid combination, Product Selector Guide and specifications Spansion revision A5: Added Section 10.4 for product capacitance Spansion revision A6: Revised Figure 10.9 to correct the starting edge of tAAVDS Spansion revison A7: Revised Typical Program & Erase Times values Spansion revision A8: Updated table: NC, DNU, RFU descriptions Updated figure 64-Ball Fine-Pitch Grid Array, S29NS512P: Revised ball labels to be consistent with Input/Output descriptions
*A	4959017	WIOB	10/13/2015	Updated to Cypress template.

Sales, Solutions, and Legal Information

Worldwide Sales and Design Support

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

Products

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

PSoC® Solutions

psoc.cypress.com/solutions

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

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

© Cypress Semiconductor Corporation, 2006-2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.

Cypress®, Spansion®, MirrorBit®, MirrorBit® Eclipse™, ORNAND™, HyperBus™, HyperFlash™, and combinations thereof, are trademarks and registered trademarks of Cypress Semiconductor Corp. All products and company names mentioned in this document may be the trademarks of their respective holders.