Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
采购部

S29GL01GT/S29GL512T

1 Gbit (128 Mbyte)/512 Mbit (64 Mbyte), GL-T MirrorBit® Eclipse™ Flash

General Description

The Cypress® S29GL01GT/512T are MirrorBit® Eclipse™ flash products fabricated on 45 nm process technology. These devices offer a fast page access time as fast as 15 ns, with a corresponding random access time as fast as 100 ns. They feature a Write Buffer that allows a maximum of 256 words/512 bytes to be programmed in one operation, resulting in faster effective programming time than standard programming algorithms. This makes these devices ideal for today's embedded applications that require higher density, better performance, and lower power consumption.

Distinctive Characteristics

- 45 nm MirrorBit Eclipse Technology
- Single supply (V_{CC}) for read / program / erase (2.7 V to 3.6 V)
- Versatile I/O feature
  - Wide I/O voltage range (V_{IO}): 1.65 V to V_{CC}
- x8/x16 data bus
- Asynchronous 32-byte Page read
- 512-byte Programming Buffer
  - Programming in Page multiples, up to a maximum of 512 bytes
- Single word and multiple program on same word options
- Automatic Error Checking and Correction (ECC) — internal hardware ECC with single bit error correction
- Sector Erase
  - Uniform 128-kbyte sectors
- Suspend and Resume commands for Program and Erase operations
- Status Register, Data Polling, and Ready/Busy pin methods to determine device status
- Advanced Sector Protection (ASP)
  - Volatile and non-volatile protection methods for each sector
- Separate 2048-byte One Time Program (OTP) array
  - Four lockable regions (SSR0 - SSR3)
  - SSR0 is Factory Locked
  - SSR3 is Password Read Protect
- Common Flash Interface (CFI) parameter table
- Temperature Range / Grade:
  - Industrial (–40 °C to +85 °C)
  - Industrial Plus (–40 °C to +105 °C)
  - Extended (–40 °C to +125 °C)
  - Automotive, AEC-Q100 Grade 3 (–40 °C to +85 °C)
  - Automotive, AEC-Q100 Grade 2 (–40 °C to +105 °C)
- 100,000 Program / Erase Cycles
- 20-year data retention
- Packaging Options
  - 56-pin TSOP
  - 64-ball LAA Fortified BGA, 13 mm × 11 mm
  - 64-ball LAE Fortified BGA, 9 mm × 9 mm
  - 56-ball VBU Fortified BGA, 9 mm × 7 mm

Performance Summary

Performance Summary for Operating Temperature Range of –40 °C to +85 °C

Maximum Read Access Times					
Density	**Voltage Range**	**Random Access Time (t_{ACC})**	**Page Access Time (t_{PACC})**	**CE# Access Time (t_{CE})**	**OE# Access Time (t_{OE})**
512 Mb	Full $V_{CC} = V_{IO}$	100	15	100	25
	Versatile I/O V_{IO}	110	25	110	35
1 Gb	Full $V_{CC} = V_{IO}$	100	15	100	25
	Versatile I/O V_{IO}	110	25	110	35

Performance Summary Operating Temperature Range of –40 °C to +105 °C

Maximum Read Access Times					
Density	**Voltage Range**	**Random Access Time (t_{ACC})**	**Page Access Time (t_{PACC})**	**CE# Access Time (t_{CE})**	**OE# Access Time (t_{OE})**
512 Mb	Full $V_{CC} = V_{IO}$	110	15	110	25
	Versatile I/O V_{IO}	120	25	120	35
1 Gb	Full $V_{CC} = V_{IO}$	110	15	110	25
	Versatile I/O V_{IO}	120	25	120	35

Performance Summary Operating Temperature Range of –40 °C to +125 °C

Maximum Read Access Times					
Density	**Voltage Range**	**Random Access Time (t_{ACC})**	**Page Access Time (t_{PACC})**	**CE# Access Time (t_{CE})**	**OE# Access Time (t_{OE})**
512 Mb	Full $V_{CC} = V_{IO}$	120	15	120	25
	Versatile I/O V_{IO}	130	25	130	35
1 Gb	Full $V_{CC} = V_{IO}$	120	15	120	25
	Versatile I/O V_{IO}	130	25	130	35

Typical Program and Erase Rates

Operation	–40 °C to +85 °C	–40 °C to +105 °C	–40 °C to +125 °C
Buffer Programming (512 bytes)	1.14 MB/s	1.14 MB/s	1.14 MB/s
Sector Erase (128 kbytes)	245 kB/s	245 kB/s	245 kB/s

Maximum Current Consumption

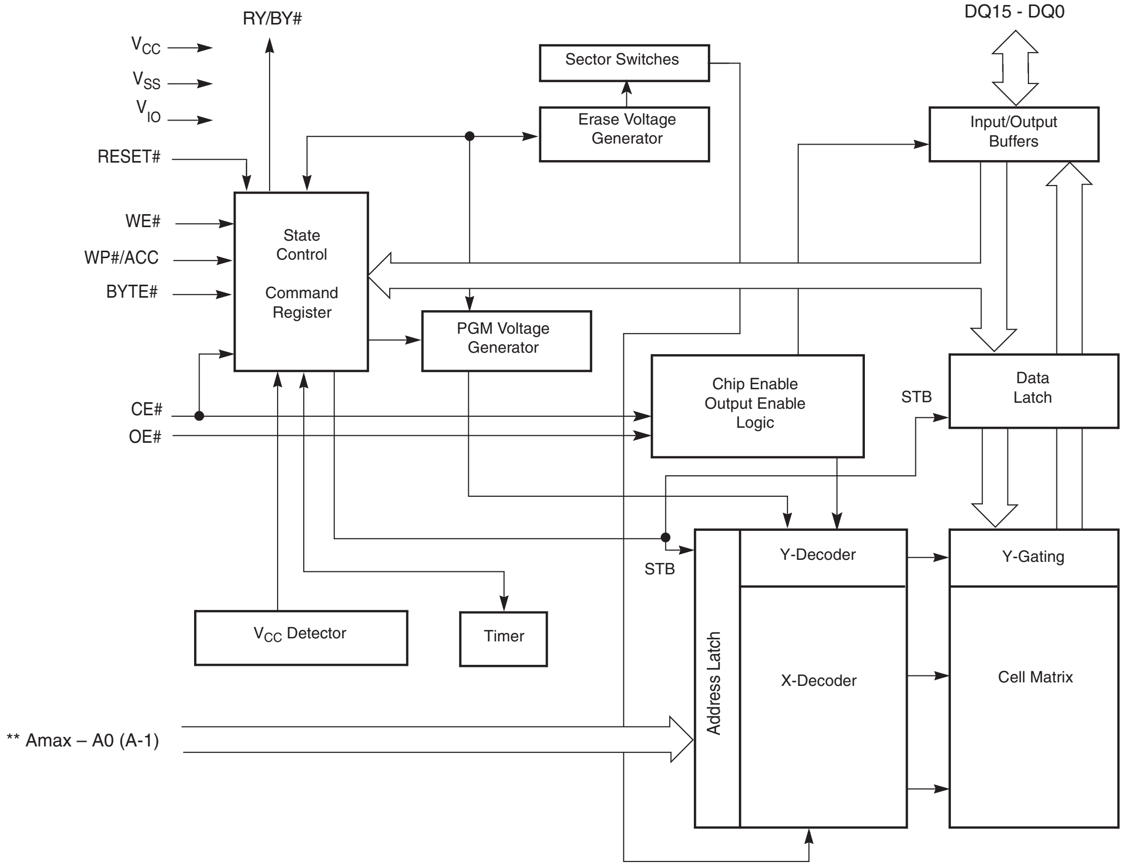
Operation	–40 °C to +85 °C	–40 °C to +105 °C	–40 °C to +125 °C
Active Read at 5 MHz, 30 pF	60 mA	60 mA	60 mA
Program	100 mA	100 mA	100 mA
Erase	100 mA	100 mA	100 mA
Standby	100 µA	200 µA	215 µA

Contents

1. Product Overview

The GL-T family consists of 512-Mbit to 1-Gbit, 3.0 V core, Versatile I/O, non-volatile, flash memory devices. These devices have an 8-bit (byte) / 16-bit (word) wide data bus and use only byte / word boundary addresses. All read accesses provide 8/16 bits of data on each bus transfer cycle. All writes take 8/16 bits of data from each bus transfer cycle.

Figure 1.1 Block Diagram



Note:
*** Amax GL01GT = A25, Amax GL512T = A24.*

The GL-T family combines the best features of eXecute In Place (XIP) and Data Storage flash memories. This family has the fast random access of XIP flash along with the high density and fast program speed of Data Storage flash.

Read access to any random location takes 100 ns to 120 ns depending on device density and I/O power supply voltage. Each random (initial) access reads an entire 32-byte aligned group of data called a Page. Other words within the same Page may be read by changing only the low order 4 bits of word address. Each access within the same Page takes 15 ns to 25 ns. This is called Page Mode read. Changing any of the higher word address bits will select a different Page and begin a new initial access. All read accesses are asynchronous.

Table 1.1 S29GL-T Address Map

	x16		x8	
Type	**Count**	**Addresses**	**Count**	**Addresses**
Address within Page	16	A3–A0	32	A3–A1
Address within Write Buffer	256	A7–A0	512	A7–A1
Page	4096 per Sector	A15–A4	4096 per Sector	A15–A4
Write-Buffer-Line	256 per Sector	A15–A8	256 per Sector	A15–A8
Sector	1024 (1 Gb) 512 (512 Mb)	Amax–A16	1024 (1 Gb) 512 (512 Mb)	Amax–A16

The device control logic is subdivided into two parallel operating sections, the Host Interface Controller (HIC) and the Embedded Algorithm Controller (EAC). HIC monitors signal levels on the device inputs and drives outputs as needed to complete read and write data transfers with the host system. HIC delivers data from the currently entered address space on read transfers; places write transfer address and data information into the EAC command memory; notifies the EAC of power transition, hardware reset, and write transfers. The EAC looks in the command memory, after a write transfer, for legal command sequences and performs the related Embedded Algorithms.

Changing the non-volatile data in the memory array requires a complex sequence of operations that are called Embedded Algorithms (EA). The algorithms are managed entirely by the device internal EAC. The main algorithms perform programming and erase of the main array data. The host system writes command codes to the flash device address space. The EAC receives the commands, performs all the necessary steps to complete the command, and provides status information during the progress of an EA.

The erased state of each memory bit is a logic 1. Programming changes a logic 1 (High) to a logic 0 (Low). Only an Erase operation is able to change a 0 to a 1. An erase operation must be performed on an entire 128-kbyte aligned and length group of data call a Sector. When shipped from Cypress all Sectors are erased.

Programming is done via a 512-byte Write Buffer. In x16 it is possible to write from 1 to 256 words, anywhere within the Write Buffer before starting a programming operation. Within the flash memory array, each 512-byte aligned group of 512 bytes is called a Line. In x8 it is possible to write from 1 to 256 bytes, anywhere within the Write Buffer before starting a program operation. A programming operation transfers volatile data from the Write Buffer to a non-volatile memory array Line. The operation is called Write Buffer Programming.

As the device transfers each 32-byte aligned page of data that was loaded into the Write buffer to the 512-byte Flash array line, internal logic programs an ECC Code for the Page into a portion of the memory array not visible to the host system software. The internal logic checks the ECC information during the initial access of every array read operation. If needed, the ECC information corrects a one bit error during the initial access time.

The Write Buffer is filled with 1’s after reset or the completion of any operation using the Write Buffer. Any locations not written to a 0 by a Write to Buffer command are by default still filled with 1’s. Any 1’s in the Write Buffer do not affect data in the memory array during a programming operation.

As each Page of data that was loaded into the Write Buffer is transferred to a memory array Line.

Sectors may be individually protected from program and erase operations by the Advanced Sector Protection (ASP) feature set. ASP provides several, hardware and software controlled, volatile and non-volatile, methods to select which sectors are protected from program and erase operations.

Software Interface

2. Address Space Overlays

There are several separate address spaces that may appear within the address range of the flash memory device. One address space is visible (entered) at any given time.

- Flash Memory Array: the main non-volatile memory array used for storage of data that may be randomly accessed by asynchronous read operations.
- ID/CFI: a memory array used for Cypress factory programmed device characteristics information. This area contains the Device Identification (ID) and Common Flash Interface (CFI) information tables.
- Secure Silicon Region (SSR): a One Time Programmable (OTP) non-volatile memory array used for Cypress factory programmed permanent data, and customer programmable permanent data.
- Lock Register: an OTP non-volatile word used to configure the ASP features and lock the SSR.
- Persistent Protection Bits (PPB): a non-volatile flash memory array with one bit for each Sector. When programmed, each bit protects the related Sector from erasure and programming.
- PPB Lock: a volatile register bit used to enable or disable programming and erasure of the PPB bits.
- Array Password: an OTP non-volatile array used to store a 64-bit password used to enable changing the state of the PPB Lock Bit when using Password Mode sector protection.
- SSR3 Password: an OTP non-volatile array used to store a 64-bit password used to enable reading the SSR3.
- Dynamic Protection Bits (DYB): a volatile array with one bit for each Sector. When set, each bit protects the related Sector from erasure and programming.
- Status Register: a volatile register used to display Embedded Algorithm status.
- Data Polling Status: a volatile register used as an alternate, legacy software compatible, way to display Embedded Algorithm status.
- ECC Status: provides the status of any error detection or correction action taken when reading the selected Page.

The main Flash Memory Array is the primary and default address space but, it may be overlaid by one other address space, at any one time. Each alternate address space is called an Address Space Overlay (ASO).

Each ASO replaces (overlays) the entire flash device address range. Any address range not defined by a particular ASO address map, is reserved for future use. All read accesses outside of an ASO address map returns non-valid (undefined) data. The locations will display actively driven data but the meaning of whatever 1’s or 0’s appear are not defined.

There are four device operating modes that determine what appears in the flash device address space at any given time:

- Read Mode
- Data Polling Mode
- Status Register (SR) Mode
- Address Space Overlay (ASO) Mode

In Read Mode the entire Flash Memory Array may be directly read by the host system memory controller. The memory device Embedded Algorithm Controller (EAC), puts the device in Read mode during Power-on, after a Hardware Reset, after a Command Reset, or after an Embedded Algorithm (EA) is suspended. Read accesses and command writes are accepted in read mode. A subset of commands are accepted in read mode when an EA is suspended.

While in any mode, the Status Register read command may be issued to cause the Status Register ASO to appear at every word address in the device address space. In this Status Register ASO Mode, the device interface waits for a read access and, any write access is ignored. The next read access to the device accesses the content of the status register, exits the Status Register ASO, and returns to the previous (calling) mode in which the Status Register read command was received.

In EA mode the EAC is performing an Embedded Algorithm, such as programming or erasing a non-volatile memory array. While in EA mode, none of the main Flash Memory Array is readable because the entire flash device address space is replaced by the Data Polling Status ASO. Data Polling Status will appear at every word location in the device address space.

While in EA mode, only a Program / Erase suspend command or the Status Register Read command will be accepted. All other commands are ignored. Thus, no other ASO may be entered from the EA mode.

When an Embedded Algorithm is suspended, the Data Polling ASO is visible until the device has suspended the EA. When the EA is suspended the Data Polling ASO is exited and Flash Array data is available. The Data Polling ASO is reentered when the suspended EA is resumed, until the EA is again suspended or finished. When an Embedded Algorithm is completed, the Data Polling ASO is exited and the device goes to the previous (calling) mode (from which the Embedded Algorithm was started).

In ASO mode, one of the remaining overlay address spaces is entered (overlaid on the main Flash Array address map). Only one ASO may be entered at any one time. Commands to the device affect the currently entered ASO. Only certain commands are valid for each ASO. These are listed in the Table 7.1 on page 48, in each ASO related section of the table.

The following ASOs have non-volatile data that may be programmed to change 1’s to 0’s:

- Secure Silicon Region
- Lock Register
- Persistent Protection Bits (PPB)
- Password
- Only the PPB ASO has non-volatile data that may be erased to change 0’s to 1’s

When a program or erase command is issued while one of the non-volatile ASOs is entered, the EA operates on the ASO. The ASO is not readable while the EA is active. When the EA is completed the ASO remains entered and is again readable. Suspend and Resume commands are ignored during an EA operating on any of these ASOs.

2.1 Flash Memory Array

The S29GL-T family has uniform sector architecture with a sector size of 128 kB. The following tables show the sector architecture of the different devices.

Table 2.1 S29GL01GT Sector and Memory Address Map

Sector Size (kbyte)	Sector Count	Sector Range	Address Range (16-Bit)	Address Range (8-Bit)	Notes
128	1024	SA0	0000000h-000FFFFh	0000000h-001FFFFh	Sector Starting Address
		:	:	:	–
		SA1023	3FF0000h-3FFFFFFh	7FF0000h-7FFFFFFh	Sector Ending Address

Table 2.2 S29GL512T Sector and Memory Address Map

Sector Size (kbyte)	Sector Count	Sector Range	Address Range (16-Bit)	Address Range (8-Bit)	Notes
128	512	SA0	0000000h-000FFFFh	0000000h-001FFFFh	Sector Starting Address
		:	:	:	–
		SA511	1FF0000h-1FFFFFFh	3FF0000h-3FFFFFFh	Sector Ending Address

Note: These tables have been condensed to show sector related information for an entire device on a single page Sectors and their address ranges that are not explicitly listed (such as SA1-SA510 on the GL512T) have sectors starting and ending addresses that form the same pattern as all other sectors of that size. For example, all 128 kB sectors have the pattern XXX0000h-XXXFFFFh in x16 and XXX0000h-XXX1FFFF in x8.

2.2 Device ID and CFI (ID-CFI) ASO

There are two traditional methods for systems to identify the type of flash memory installed in the system. One has traditionally been called Autoselect and is now referred to as Device Identification (ID). The other method is called Common Flash Interface (CFI).

For ID, a command is used to enable an address space overlay where up to 16 word locations can be read to get JEDEC manufacturer identification (ID), device ID, and some configuration and protection status information from the flash memory. The system can use the manufacturer and device IDs to select the appropriate driver software to use with the flash device.

CFI also uses a command to enable an address space overlay where an extendable table of standard information about how the flash memory is organized and operates can be read. With this method the driver software does not have to be written with the specifics of each possible memory device in mind. Instead the driver software is written in a more general way to handle many different devices but adjusts the driver behavior based on the information in the CFI table.

Traditionally these two address spaces have used separate commands and were separate overlays. However, the mapping of these two address spaces are non-overlapping and so can be combined in to a single address space and appear together in a single overlay. Either of the traditional commands used to access (enter) the Autoselect (ID) or CFI overlay will cause the now combined ID-CFI address map to appear.

The ID-CFI address map appears overlays the entire Flash Array.

The ID-CFI address map starts at location 0 of the selected sector. Locations above the maximum defined address of the ID-CFI ASO to the maximum address of the selected sector have undefined data. The ID-CFI enter commands use the same address and data values used on previous generation memories to access the JEDEC Manufacturer ID (Autoselect) and Common Flash Interface (CFI) information, respectively.

Table 2.3 ID-CFI Address Map Overview

Word Address	Byte Address	Description	Read / Write
(SA) + 0000h to 000Fh	(SA) + 0000h to 001Fh	Device ID (traditional Autoselect values)	Read Only
(SA) + 0010h to 0079h	(SA) + 0020h to 00F2h	CFI data structure	Read Only
(SA) + 0080h to FFFFh	(SA) + 00F3h to 1FFFFh	Undefined	Read Only

For the complete address map see Table 7.3 on page 55.

2.2.1 Device ID

The Joint Electron Device Engineering Council (JEDEC) standard JEP106T defines the manufacturer ID for a compliant memory. Common industry usage defined a method and format for reading the manufacturer ID and a device specific ID from a memory device. The manufacturer and device ID information is primarily intended for programming equipment to automatically match a device with the corresponding programming algorithm. Cypress has added additional fields within this 32-byte address space.

The original industry format was structured to work with any memory data bus width e. g. x8, x16, x32. The ID code values are traditionally byte wide but are located at bus width address boundaries such that incrementing the device address inputs will read successive byte, word, or double word locations with the ID codes always located in the least significant byte location of the data bus. Because the device data bus is word wide each code byte is located in the lower half of each word location. The original industry format made the high order byte always 0. Cypress has modified the format to use both bytes in some words of the address space. For the detail description of the Device ID address map see Table 7.3 on page 55.

2.2.2 Common Flash Memory Interface

The JEDEC Common Flash Interface (CFI) specification (JESD68.01) defines a standardized data structure that may be read from a flash memory device, which allows vendor-specified software algorithms to be used for entire families of devices. The data structure contains information for system configuration such as various electrical and timing parameters, and special functions supported by the device. Software support can then be device-independent, Device ID-independent, and forward-and-backward-compatible for entire Flash device families.

The system can read CFI information at the addresses within the selected sector as shown in Device ID and Common Flash Interface (ID-CFI) ASO Map on page 55.

Like the Device ID information, CFI information is structured to work with any memory data bus width e. g. x8, x16, x32. The code values are always byte wide but are located at data bus width address boundaries such that incrementing the device address reads successive byte, word, or double word locations with the codes always located in the least significant byte location of the data bus. Because the data bus is word wide each code byte is located in the lower half of each word location and the high order byte is always 0.

For further information, please refer to the *CFI Specification, Version 1.4* (or later), and the *JEDEC publications JEP137-A and JESD68.01*. Please contact JEDEC (www.jedec.org) for their standards and the CFI Specification may be found at the Cypress Web site (www.cypress.com/spansionappnotes at the time of this document's publication) or by contacting a local Cypress sales office listed on the web site.

2.3 Status Register ASO

The Status Register ASO contains a single word of registered volatile status for Embedded Algorithms. When the Status Register read command is issued, the current status is captured (by the rising edge of WE#) into the register and the ASO is entered. The Status Register content appears on all word locations. The first read access exits the Status Register ASO (with the rising edge of CE# or OE#) and returns to the address space map in use when the Status Register read command was issued. Write commands will not exit the Status Register ASO state.

2.4 Data Polling Status ASO

The Data Polling Status ASO contains a single word of volatile memory indicating the progress of an EA. The Data Polling Status ASO is entered immediately following the last write cycle of any command sequence that initiates an EA. Commands that initiate an EA are:

- Word Program
- Program Buffer to Flash
- Chip Erase
- Sector Erase
- Erase Resume / Program Resume
- Program Resume Enhanced Method
- Blank Check
- Lock Register Program
- Password Program
- PPB Program
- All PPB Erase
- Evaluate Erase Status

The Data Polling Status word appears at all word locations in the device address space. When an EA is completed the Data Polling Status ASO is exited and the device address space returns to the address map mode where the EA was started.

2.5 Secure Silicon Region ASO

The Secure Silicon Region (SSR) provides an extra memory area that can be programmed once and permanently protected from further changes, i. e., it is a One Time Program (OTP) area. The SSR is
2048 bytes in length. It consists of 512 bytes for Factory Locked Secure Silicon Region (SSR0), 1024 bytes for Customer Locked Secure Silicon Regions (SSR1 and SSR2), and 512 bytes for Customer Locked Secure Silicon Region with Read password (SSR3).

SSR0 is shipped locked, preventing further programming. SSR1 and SSR2 are OTP with each having separate lock bits and once locked no further programming is allowed for that region. SSR3 is an OTP and requires a SSR3 password to read or program that region. Once SSR3 is locked no further programming is allowed for that region.

The sector address supplied during the Secure Silicon Entry command selects the Flash Memory Array sector that is overlaid by the Secure Silicon Region address map. The SSR is overlaid starting at location 0 in the selected sector. Use of the sector 0 address is recommended for future compatibility. While the SSR ASO is entered the content of all other sectors is memory core data for read operations. Program is not allowed outside of ASO.

Table 2.4 Secure Silicon Region

Word Address Range	Byte Address Range	Content	Region	Size
(SA) + 0000h to 00FFh	(SA) + 0000h to 01FFh	Factory Locked Secure Silicon Region	SSR0	512 bytes
(SA) + 0100h to 01FFh	(SA) + 0200h to 03FFh	Customer Locked Secure Silicon Region	SSR1	512 bytes
(SA) + 0200h to 02FFh	(SA) + 0400h to 05FFh	Customer Locked Secure Silicon Region	SSR2	512 bytes
(SA) + 0300h to 03FFh	(SA) + 0600h to 07FFh	Customer Locked Secure Silicon Region with Read Password	SSR3	512 bytes
(SA) + 0400h to FFFFh	(SA) + 0800h to 1FFFFh	Undefined	n/a	126 kbytes

2.6 Sector Protection Control

2.6.1 Lock Register ASO

The Lock register ASO contains a single word of OTP memory. When the ASO is entered the Lock Register appears at all word locations in the device address space. However, it is recommended to read or program the Lock Register only at location 0 of the device address space for future compatibility.

2.6.2 Persistent Protection Bits (PPB) ASO

The PPB ASO contains one bit of a Flash Memory Array for each Sector in the device. When the PPB ASO is entered, the PPB bit for a sector appears in the Least Significant Bit (LSB) of each address in the sector. Reading any address in a sector displays data where the LSB indicates the non-volatile protection status for that sector. However, it is recommended to read or program the PPB only at address 0 of the sector for future compatibility. If the bit is 0 the sector is protected against programming and erase operations. If the bit is 1 the sector is not protected by the PPB. The sector may be protected by other features of ASP.

2.6.3 PPB LOCK ASO

The PPB Lock ASO contains a single bit of volatile memory. The bit controls whether the bits in the PPB ASO may be programmed or erased. If the bit is 0 the PPB ASO is protected against programming and erase operations. If the bit is 1 the PPB ASO is not protected. When the PPB Lock ASO is entered the PPB Lock bit appears in the least significant bit (LSB) of each address in the device address space. However, it is recommended to read or program the PPB Lock only at address 0 of the device for future compatibility.

2.6.4 Password ASO

The Password ASO contains four words of OTP memory. When the ASO is entered the Password appears starting at address 0 in the device address space. All locations above the fourth word are undefined.

2.6.5 Dynamic Protection Bits (DYB) ASO

The DYB ASO contains one bit of a volatile memory array for each Sector in the device. When the DYB ASO is entered, the DYB bit for a sector appears in the least significant bit (LSB) of each address in the sector. Reading any address in a sector displays data where the LSB indicates the non-volatile protection status for that sector. However, it is recommended to read, set, or clear the DYB only at address 0 of the sector for future compatibility. If the bit is 0 the sector is protected against programming and erase operations. If the bit is 1 the sector is not protected by the DYB. The sector may be protected by other features of ASP.

2.7 ECC Status ASO

The system can access the ECC Status ASO by issuing the ECC Status entry command sequence during Read Mode. The ECC Status ASO provides the enabled or disabled status of the ECC function or if the ECC function corrected a single-bit Error when reading the selected Page. Section 5.3, Automatic ECC on page 21 describes the ECC function in more detail.

The ECC Status ASO allows the following activities:

- Read ECC Status for the selected Page.
- ASO Exit.

2.7.1 ECC Status

The contents of the ECC Status ASO indicate, for the selected ECC Page, whether the ECC logic has corrected an error in the ECC Page eight bit ECC code, in the ECC page of 32-bytes of data, or that ECC is disabled for that ECC unit. The address specified in the ECC Status Read Command, provided in Table 7.1 on page 48 and Table 7.2 on page 51, selects the ECC Page.

Table 2.5 ECC Status Word – Upper Byte

Bit	15	14	13	12	11	10	9	8
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
Value	X	X	X	X	X	X	X	X

Table 2.6 ECC Status Word – Lower Byte

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	ECC Enabled on 16-Word Page	Single Bit Error Corrected ECC Bits	Single Bit Error Corrected Data Bits	RFU
Value	X	X	X	X	0=ECC Enabled 1=ECC Disabled	0=No Error Corrected 1=Single Bit Error Corrected	0=No Error Corrected 1=Single Bit Error Corrected	X

3. Data Protection

The device offers several features to prevent malicious or accidental modification of any sector via hardware means.

3.1 Device Protection Methods

3.1.1 Power-Up Write Inhibit

RESET#, CE#, WE#, and, OE# are ignored during Power-On Reset (POR). During POR, the device can not be selected, will not accept commands on the rising edge of WE#, and does not drive outputs. The Host Interface Controller (HIC) and Embedded Algorithm Controller (EAC) are reset to their standby states, ready for reading array data, during POR. CE# or OE# must go to V_{IH} before the end of POR (t_{VCS}).

At the end of POR the device conditions are:

- all internal configuration information is loaded,
- the device is in read mode,
- the Status Register is at default value,
- all bits in the DYB ASO are set to un-protect all sectors,
- the Write Buffer is loaded with all 1's,
- the EAC is in the standby state.

3.1.2 Low V_{CC} Write Inhibit

When V_{CC} is less than V_{LKO}, the HIC does not accept any write cycles and the EAC resets. This protects data during V_{CC} power-up and power-down. The system must provide the proper signals to the control pins to prevent unintentional writes when V_{CC} is greater than V_{LKO}.

3.2 Command Protection

Embedded Algorithms are initiated by writing command sequences into the EAC command memory. The command memory array is not readable by the host system and has no ASO. Each host interface write is a command or part of a command sequence to the device. The EAC examines the address and data in each write transfer to determine if the write is part of a legal command sequence. When a legal command sequence is complete the EAC will initiate the appropriate EA.

Writing incorrect address or data values, or writing them in an improper sequence, will generally result in the EAC returning to its Standby state. However, such an improper command sequence may place the device in an unknown state, in which case the system must write the reset command, or possibly provide a hardware reset by driving the RESET# signal Low, to return the EAC to its Standby state, ready for random read.

The address provided in each write may contain a bit pattern used to help identify the write as a command to the device. The upper portion of the address may also select the sector address on which the command operation is to be performed. The Sector Address (SA) includes Amax through A16 flash address bits (system byte address signals Amax through A16). A command bit pattern is located in A10 to A0 flash address bits (system byte address signals A11 through A1).

The data in each write may be: a bit pattern used to help identify the write as a command, a code that identifies the command operation to be performed, or supply information needed to perform the operation. See Table 7.1 on page 48 for a listing of all commands accepted by the device.

3.3 Secure Silicon Region (OTP)

See Section 2.5, Secure Silicon Region ASO on page 9 for a description of the secure silicon region. See Section 5.4.9.3, Secure Silicon Region ASO on page 33 for a description of the allowed commands.

3.4 Sector Protection Methods

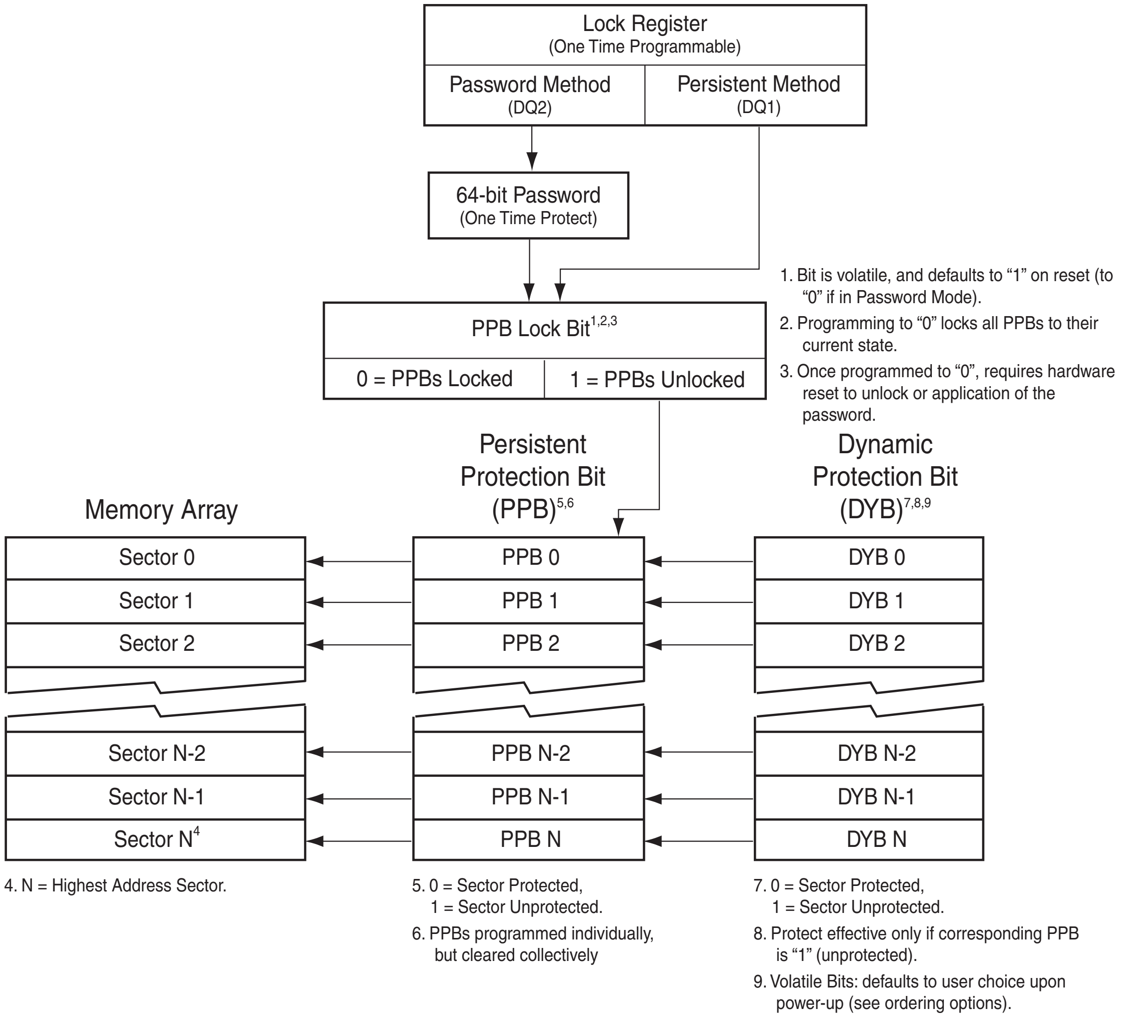
3.4.1 Write Protect Signal

If WP# = V_{IL}, the lowest or highest address sector is protected from program or erase operations independent of any other ASP configuration. Whether it is the lowest or highest sector depends on the device ordering option (model) selected. If WP# = V_{IH}, the lowest or highest address sector is not protected by the WP# signal but it may be protected by other aspects of ASP configuration. WP# has an internal pull-up; when unconnected, WP# is at V_{IH}. WP# should not change between V_{IL} and V_{IH} during any embedded operation.

3.4.2 ASP

Advanced Sector Protection (ASP) is a set of independent hardware and software methods used to disable or enable programming or erase operations, individually, in any or all sectors. This section describes the various methods of protecting data stored in the memory array. An overview of these methods is shown in Figure 3.1.

Figure 3.1 Advanced Sector Protection Overview



1. Bit is volatile, and defaults to “1” on reset (to “0” if in Password Mode).
2. Programming to “0” locks all PPBs to their current state.
3. Once programmed to “0”, requires hardware reset to unlock or application of the password.
4. N = Highest Address Sector.
5. 0 = Sector Protected,
   1 = Sector Unprotected.
6. PPBs programmed individually, but cleared collectively
7. 0 = Sector Protected,
   1 = Sector Unprotected.
8. Protect effective only if corresponding PPB is “1” (unprotected).
9. Volatile Bits: defaults to user choice upon power-up (see ordering options).

Every main flash array sector has a non-volatile (PPB) and a volatile (DYB) protection bit associated with it. When either bit is 0, the sector is protected from program and erase operations.

The PPB bits are protected from program and erase when the PPB Lock bit is 0. There are two methods for managing the state of the PPB Lock bit, Persistent Protection and Password Protection.

The Persistent Protection method sets the PPB Lock to 1 during POR or Hardware Reset so that the PPB bits are unprotected by a device reset. There is a command to clear the PPB Lock bit to 0 to protect the PPB bits. There is no command in the Persistent Protection method to set the PPB Lock bit therefore the PPB Lock bit will remain at 0 until the next power-off or hardware reset. The Persistent Protection method allows boot code the option of changing sector protection by programming or erasing the PPB, then protecting the PPB from further change for the remainder of normal system operation by clearing the PPB Lock bit. This is sometimes called Boot-code controlled sector protection.

The Password method clears the PPB Lock bit to 0 during POR or Hardware Reset to protect the PPB. A 64-bit password may be permanently programmed and hidden for the password method. A command can be used to provide a password for comparison with the hidden password. If the password matches the PPB Lock bit is set to 1 to unprotect the PPB. A command can be used to clear the PPB Lock bit to 0.

The selection of the PPB Lock management method is made by programming OTP bits in the Lock Register so as to permanently select the method used.

The Lock Register also contains OTP bits, for protecting the SSR.

The PPB bits are erased so that all main flash array sectors are unprotected when shipped from Cypress. The Secured Silicon Region can be factory protected or left unprotected depending on the ordering option (model) ordered.

3.4.3 PPB Lock

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

The PPB Lock command is used to clear the bit to 0. The PPB Lock Bit must be cleared to 0 only after all the PPBs are configured to the desired settings.

In Persistent Protection mode, the PPB Lock is set to 1 during POR or a hardware reset. When cleared, no software command sequence can set the PPB Lock, only another hardware reset or power-up can set the PPB Lock bit.

In the Password Protection mode, the PPB Lock is cleared to 0 during POR or a hardware reset. The PPB Lock can only set to 1 by the Password Unlock command sequence. The PPB Lock can be cleared by the PPB Lock Bit Clear command.

3.4.4 Persistent Protection Bits (PPB)

The Persistent Protection Bits (PPB) are located in a separate nonvolatile flash array. One of the PPB bits is assigned to each sector. When a PPB is 0 its related sector is protected from program and erase operations. The PPB are programmed individually but must be erased as a group, similar to the way individual words may be programmed in the main array but an entire sector must be erased at the same time. Preprogramming and verification prior to erasure are handled by the EAC.

Programming a PPB bit requires the typical word programming time. During a PPB bit programming operation or PPB bit erasing, Data polling Status DQ6 Toggle Bit I will toggle until the operation is complete. Erasing all the PPBs requires typical sector erase time.

If the PPB Lock is 0, the PPB Program or erase commands do not execute and time-out without programming or erasing the PPB.

The protection state of a PPB for a given sector can be verified by executing a PPB Status Read command when entered in the PPB ASO.

3.4.5 Dynamic Protection Bits (DYB)

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

The DYB can be set to 0 or cleared to 1 as often as needed.

3.4.6 Sector Protection States Summary

Each sector can be in one of the following protection states:

- Unlocked – The sector is unprotected and protection can be changed by a simple command. The protection state defaults to unprotected after a power cycle or hardware reset.
- Dynamically Locked – A sector is protected and protection can be changed by a simple command. The protection state is not saved across a power cycle or hardware reset.
- Persistently Locked – A sector is protected and protection can only be changed if the PPB Lock Bit is set to 1. The protection state is non-volatile and saved across a power cycle or hardware reset. Changing the protection state requires programming or erase of the PPB bits.

Table 3.1 Sector Protection States

Protection Bit Values			Sector State
PPB Lock	PPB	DYB	
1	1	1	Unprotected – PPB and DYB are changeable
1	1	0	Protected – PPB and DYB are changeable
1	0	1	Protected – PPB and DYB are changeable
1	0	0	Protected – PPB and DYB are changeable
0	1	1	Unprotected – PPB not changeable, DYB is changeable
0	1	0	Protected – PPB not changeable, DYB is changeable
0	0	1	Protected – PPB not changeable, DYB is changeable
0	0	0	Protected – PPB not changeable, DYB is changeable

3.4.7 Lock Register

The Lock Register holds the non-volatile OTP bits for controlling protection of the SSR and determining the PPB Lock bit management method (protection mode).

Table 3.2 Lock Register

Bit	Default Value	Name
15-12	1	Reserved
11	1	SSR Region 3 Password Protection Mode Lock Bit
10	1	SSR Region 3 (Customer) Lock Bit
9	1	SSR Region 2 (Customer) Lock Bit
8	0	Reserved
7	1	Reserved
6	1	SSR Region 1 (Customer) Lock Bit
5	1	Reserved
4	1	Reserved
3	1	Reserved
2	1	Password Protection Mode Lock Bit
1	1	Persistent Protection Mode Lock Bit
0	0	SSR Region 0 (Factory) Lock Bit

The Secure Silicon Region (SSR) protection bits must be used with caution, as once locked, there is no procedure available for unlocking the protected portion of the Secure Silicon Region and none of the bits in the protected Secure Silicon Region memory space can be modified in any way. Once the Secure Silicon Region area is protected, any further attempts to program in the area will fail with status indicating the area being programmed is protected. The Region 0 Indicator Bit is located in the Lock Register at bit location 0, Region 1 in bit location 6, Region 2 in bit location 9, and Region 3 in bit location 10.

As shipped from the factory, all devices default to the Persistent Protection method, with all sectors unprotected, when power is applied. The device programmer or host system can then choose which sector protection method to use. Programming either of the following two, one-time programmable, non-volatile bits, locks the part permanently in that mode:

- Persistent Protection Mode Lock Bit (DQ1)

Password Protection Mode Lock Bit (DQ2) If both lock bits are selected to be programmed at the same time, the operation will abort. 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.

If the password mode is to be chosen, the password must be programmed prior to setting the corresponding lock register bit. Setting the Password Protection Mode Lock Bit (DQ2) will disable the ability to program or read the password.

The programming time of the Lock Register is the same as the typical word programming time. During a Lock Register programming EA, Data polling Status DQ6 Toggle Bit I will toggle until the programming has completed. The system can also determine the status of the lock register programming by reading the Status Register. See Status Register on page 36 for information on these status bits.

The user is not required to program DQ2 or DQ1, and DQ6 or DQ0 bits at the same time. This allows the user to lock the SSR before or after choosing the device protection scheme. When programming the Lock Bits, the Reserved Bits must be 1 (masked).

3.4.8 Persistent Protection Mode

The Persistent Protection method sets the PPB Lock to 1 during POR or Hardware Reset so that the PPB bits are unprotected by a device reset. There is a command to clear the PPB Lock bit to 0 to protect the PPB. There is no command in the Persistent Protection method to set the PPB Lock bit to 1 therefore the PPB Lock bit will remain at 0 until the next power-off or hardware reset.

3.4.9 Password Protection Mode

3.4.9.1 PPB Password Protection Mode

PPB Password Protection Mode allows an even higher level of security than the Persistent Sector Protection Mode, by requiring a 64-bit password for setting the PPB Lock. In addition to this password requirement, after power up and reset, the PPB Lock is cleared to 0 to ensure protection at power-up. Successful execution of the Password Unlock command by entering the entire password sets the PPB Lock to 1, allowing for sector PPB modifications.

Password Protection Notes:

- The Password Program Command is only capable of programming 0’s.
- The password is all 1’s when shipped from Cypress. It is located in its own memory space and is accessible through the use of the Password Program and Password Read commands.
- All 64-bit password combinations are valid as a password.
- Once the Password is programmed and verified, the Password Mode Locking Bit must be set in order to prevent reading or modification of the password.
- The Password Mode Lock Bit, once programmed, prevents reading the 64-bit password on the data bus and further password programming. All further read commands to the password region are disabled (data is read as 1’s). There is no means to verify what the password is after the Password Protection Mode Lock Bit is programmed. Password verification is only allowed before selecting the Password Protection mode. Any program operation will fail and will report the results as a normal program failure on a locked sector.
- The Password Mode Lock Bit is not erasable.

- The exact password must be entered in order for the unlocking function to occur.
  - The addresses can be loaded in any order but all 4 words are required for a successful match to occur.
  - The Sector Addresses (Amax–A16) and Word Line Addresses (A15–A8) are compared to 'zero' while the password address/ data are loaded. If the Sector Address or Word Line Address don't match then the error will be reported at the end of that write cycle. The status register will return to the ready state with the Program Status Bit set to 1 and Write Buffer Abort Status Bit set to 1 indicating a failed programming operation. The data polling status will remain active, with DQ7 set to the complement of the DQ7 bit in the last word of the password unlock command, and DQ6 toggling. RY/BY# will remain low.
  - The specific address and data are compared after the Program Buffer To Flash command has been given. If they don't match to the internal set value than the status register will return to the ready state with the Program Status Bit set to 1 indicating a failed programming operation. The data polling status will remain active, with DQ7 set to the complement of the DQ7 bit in the last word of the password unlock command, and DQ6 toggling. RY/BY# will remain low. In this error case due to incorrect password, the device requires a wait time of t_{PPB} and a software reset command to clear the error prior to the Password ASO Exit command to properly exit the Password ASO. Failure to do so will cause the device to remain in the Password ASO.
- The device requires t_{PPB} for setting the PPB Lock after the valid 64-bit password is given to the device.This makes it take an unreasonably long time (58 million years) for a hacker to run through all the 64-bit combinations in an attempt to correctly match a password. The EA status checking methods may be used to determine when the EAC is ready to accept a new password command.
- If the password is lost after setting the Password Mode Lock Bit, there is no way to clear the PPB Lock.

4. Read Operations

4.1 Asynchronous Read

Each read access may be made to any location in the memory (random access). Each random access is self-timed with the same latency from CE# or address to valid data (t_{ACC} or t_{CE}).

4.2 Page Mode Read

Each random read accesses an entire 32-byte Page in parallel. Subsequent reads within the same Page have faster read access speed. The Page is selected by the higher address bits (Amax-A4), while the specific word of that Page is selected by the least significant address bits A3-A0 (A3-A-1 in x8 mode). The higher address bits are kept constant and only A3-A0 (A3-A-1 in x8 mode) changed to select a different word in the same Page. This is an asynchronous access with data appearing on DQ15-DQ0 (DQ7-DQ0 in x8 mode) when CE# remains Low, OE# remains Low, and the asynchronous Page access time (t_{PACC}) is satisfied. If CE# goes High and returns Low for a subsequent access, a random read access is performed and time is required (t_{ACC} or t_{CE}).

5. Embedded Operations

5.1 Embedded Algorithm Controller (EAC)

The EAC takes commands from the host system for programming and erasing the flash memory array and performs all the complex operations needed to change the non-volatile memory state. This frees the host system from any need to manage the program and erase processes.

There are four EAC operation categories:

- Standby (Read Mode)
- Address Space Switching
- Embedded Algorithms (EA)
- Advanced Sector Protection (ASP) Management

5.1.1 EAC Standby

In the standby mode current consumption is greatly reduced. The EAC enters its standby mode when no command is being processed and no Embedded Algorithm is in progress. If the device is deselected
(CE# = High) during an Embedded Algorithm, the device still draws active current until the operation is completed (I_{CC3}). I_{CC4} in DC Characteristics on page 68 represents the standby current specification when both the Host Interface and EAC are in their Standby state.

5.1.2 Address Space Switching

Writing specific address and data sequences (command sequences) switch the memory device address space from the main flash array to one of the Address Space Overlays (ASO).

Embedded Algorithms operate on the information visible in the currently active (entered) ASO. The system continues to have access to the ASO until the system issues an ASO Exit command, performs a Hardware RESET, or until power is removed from the device. An ASO Exit Command switches from an ASO back to the main flash array address space. The commands accepted when a particular ASO is entered are listed between the ASO enter and exit commands in the command definitions table. See Command Summary on page 48 for address and data requirements for all command sequences.

5.1.3 Embedded Algorithms (EA)

Changing the non-volatile data in the memory array requires a complex sequence of operations that are called Embedded Algorithms (EA). The algorithms are managed entirely by the device internal Embedded Algorithm Controller (EAC). The main algorithms perform programming and erasing of the main array data and the ASO's. The host system writes command codes to the flash device address space. The EAC receives the commands, performs all the necessary steps to complete the command, and provides status information during the progress of an EA.

5.2 Program and Erase Summary

Flash data bits are erased in parallel in a large group called a sector. The Erase operation places each data bit in the sector in the logical 1 state (High). Flash data bits may be individually programmed from the erased 1 state to the programmed logical 0 (low) state. A data bit of 0 cannot be programmed back to a 1. A succeeding read shows that the data is still 0. Only erase operations can convert a 0 to a 1. Programming the same word location more than once with different 0 bits will result in the logical AND of the previous data and the new data being programmed. The duration of program and erase operations is shown in Embedded Algorithm Performance Table on page 44.

Program and erase operations may be suspended.

- An erase operation may be suspended to allow either programming or reading of another sector (not in the erase sector). No other erase operation can be started during an erase suspend.
- A program operation may be suspended to allow reading of another location (not in the Line being programmed).
- No other program or erase operation may be started during a suspended program operation – program or erase commands will be ignored during a suspended program operation.
- After an intervening program operation or read access is complete the suspended erase or program operation may be resumed. The resume can happen at any time after the suspend, assuming the device is not in the process of executing another command.
- Program and Erase operations may be interrupted as often as necessary but in order for a program or erase operation to progress to completion there must be some periods of time between resume and the next suspend commands greater than or equal to t_{PRS} or t_{ERS} in Embedded Algorithm Performance Table on page 44.
- When an Embedded Algorithm (EA) is complete, the EAC returns to the operation state and address space from which the EA was started (Erase Suspend, EAC Standby, ...).

The system can determine the status of a program or erase operation by reading the Status Register or using Data Polling Status. Refer to Status Register on page 36 for information on these status bits. Refer to Data Polling Status on page 37 for more information.

Any commands written to the device during the Embedded Program Algorithm are ignored except the Program Suspend (x51h), Status Read command (x70h), and Erase Suspend/Program Suspend command (xB0h).

Any commands written to the device during the Embedded Erase Algorithm are ignored except Status Read (x70h) and Erase Suspend/Program Suspend command (xB0h).

A hardware reset immediately terminates any in progress program / erase operation and returns to read mode after t_{RPH} time. The terminated operation should be reinitiated once the device has returned to the idle state, to ensure data integrity.

For performance and reliability reasons reading and programming is internally done on full 32-byte Pages. I_{CC3} in DC Characteristics on page 68 represents the active current specification for a write (Embedded Algorithm) operation.

5.2.1 Program Granularity

The S29GL-T supports two methods of programming, Word or Write Buffer Programming. Each Page can be programmed by either method. Pages programmed by different methods may be mixed within a Line for the Industrial Temperature version (-40°C to +85°C). For the Industrial Plus version (-40°C to +105°C) and Extended version (-40°C to +125°C) the device will only support one programming operation on each 32-byte page between erase operations and Single Word Programming command is not supported.

Word programming examines the data word supplied by the command and programs 0's in the addressed memory array word to match the 0's in the command data word.

Write Buffer Programming examines the write buffer and programs 0's in the addressed memory array Pages to match the 0's in the write buffer. The write buffer does not need to be completely filled with data. It is allowed to program as little as a single bit, several bits, a single word, a few words, a Page, multiple Pages, or the entire buffer as one programming operation. Use of the write buffer method reduces host system overhead in writing program commands and reduces memory device internal overhead in programming operations to make Write Buffer Programming more efficient and thus faster than programming individual words with the Word Programming command.

5.2.2 Incremental Programming

The same word location may be programmed more than once, by either the Word or Write Buffer Programming methods, to incrementally change 1's to 0's. Note that more than one programming operation on the same Page will disable ECC for that Page.

5.3 Automatic ECC

5.3.1 ECC Overview

The Automatic ECC feature works transparently with normal program, erase, and read operations. As the device transfers each Page of data from the Write Buffer to the memory array, internal ECC logic programs ECC Code for the Page into a portion of the memory array that is not visible to the host system. The device evaluates the Page data and the ECC Code during each initial Page access. If needed, the internal ECC logic will correct a one bit error during the initial access.

Programming more than once to a particular Page will disable the ECC function for that Page. The ECC function will remain disabled for that Page until the next time the host system erases the Sector containing that Page. The host system may read data stored in that Page following multiple programming operations; however, ECC is disabled and an error in that Page will not be detected or corrected.

5.3.2 Program and Erase Summary

For performance and reliability reasons, reading and programming operations are performed on full 32-byte Pages in parallel. The device provides ECC on each Page by adding an ECC Code to each Page when first programmed. The ECC Code is automatic and transparent to the host system.

5.3.3 ECC Implementation

Each 32-byte Page in the main flash array, as well as each 32-byte OTP region, features an associated ECC Code. Internal ECC logic is able to detect and correct any single bit error found in a Page, or the associated ECC Code, during a read access.

The first Write Buffer program operation applied to a Page programs the ECC Code for that Page. Subsequent programming operations, that occur more than once, on a particular Page disable the ECC function for that Page. This allows bit or word programming; however, note that multiple programming operations to the same Page will disable the ECC function on the Page where incremental programming occurs. An erase of the Sector containing a Page with ECC disabled will re-enable the ECC function for that Page.

The ECC function is automatic and transparent to the user. The transparency of the Automatic ECC function enhances data integrity for typical programming operations that write data once to each Page. The ECC function also facilitates software compatibility to previous generations of GL Family products by allowing single word programming and bit walking where the same Page or word is programmed more than once. When a Page has Automatic ECC disabled, the ECC function will not detect or correct an error on a data read from that Page.

5.3.4 Word Programming

Word programming programs a single word anywhere in the main Flash Memory Array. Programming multiple words in the same 32-byte Page disables Automatic ECC protection on that Page. A sector erase of the sector containing that Page will re-enable Automatic ECC following multiple word programming operations on that Page.

5.3.5 Write Buffer Programming

Each Write Buffer Program operation allows for programming of 1 bit up to 512 bytes. A 32-byte Page is the smallest program granularity that features Automatic ECC protection. Programming the same Page more than once will disable the Automatic ECC function on that Page. Cypress recommends that a Write Buffer programming operation program multiple Pages in an operation and write each Page only once. This keeps the Automatic ECC protection enabled on each Page. For the very best performance, program in full Lines of 512 bytes aligned on 512-byte boundaries.

5.4 Command Set

5.4.1 Program Methods

5.4.1.1 Word Programming

Word programming is used to program a single word anywhere in the main Flash Memory Array.

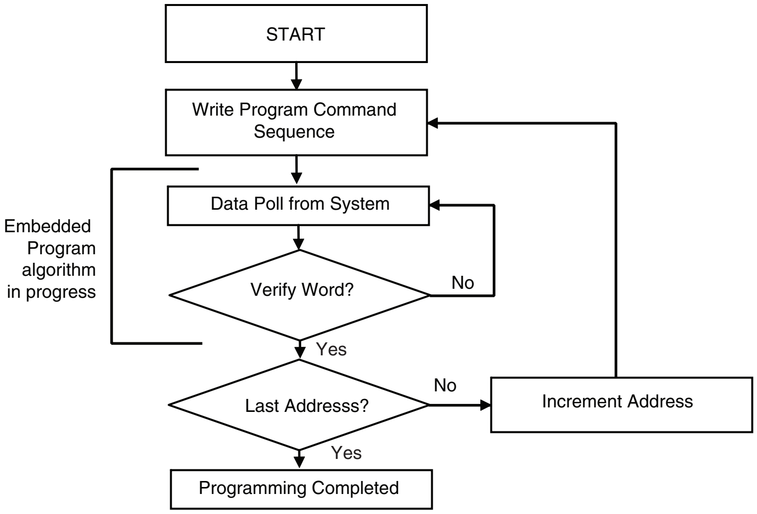
The Word Programming command is a four-write-cycle sequence. The program command sequence is initiated by writing two unlock write cycles, followed by the program set up command. The program address and data are written next, which in turn initiate the Embedded Word Program algorithm. The system is not required to provide further controls or timing. The device automatically generates the program pulses and verifies the programmed cell margin internally. When the Embedded Word Program algorithm is complete, the EAC then returns to its standby mode.

The system can determine the status of the program operation by using Data Polling Status, reading the Status Register, or monitoring the RY/BY# output. See Status Register on page 36 for information on these status bits. See Data Polling Status on page 37 for information on these status bits. See Figure 5.1 on page 22 for a diagram of the word programming operation.

Any commands other than Program Suspend written to the device during the Embedded Program algorithm are ignored. Note that a hardware reset (RESET# = V_{IL}) immediately terminates the programming operation and returns the device to read mode after t_{RPH} time. To ensure data integrity, the Program command sequence should be reinitiated once the device has completed the hardware reset operation.

A modified version of the Word Programming command, without unlock write cycles, is used for programming when entered into the Lock Register, Password, and PPB ASOs or the Unlock Bypass mode. The same command is used to change volatile bits when entered in to the PPB Lock, and DYB ASOs. See Table 7.1 on page 48 for program command sequences.

Figure 5.1 Word Program Operation



5.4.1.2 Write Buffer Programming

A write buffer is used to program data within a 512-byte address range aligned on a 512-byte boundary (Line). Thus, a full Write Buffer Programming operation must be aligned on a Line boundary. Programming operations of less than a full 512 bytes may start on any word boundary but may not cross a Line boundary. At the start of a Write Buffer programming operation all bit locations in the buffer are all 1's (FFFFh words) thus any locations not loaded will retain the existing data. See Product Overview on page 4 for information on address map.

Write Buffer Programming allows up to 512 bytes to be programmed in one operation. It is possible to program from 1 bit up to 512 bytes in each Write Buffer Programming operation. It is recommended that a multiple of Pages be written and each Page written only once. For the very best performance, programming should be done in full Lines of 512 bytes aligned on 512-byte boundaries.

Write Buffer Programming is supported only in the main flash array or the SSR ASO.

The Write Buffer Programming operation is initiated by first writing two unlock cycles. This is followed by a third write cycle of the Write to Buffer command with the Sector Address (SA), in which programming is to occur. Next, the system writes the number of word locations minus 1. This tells the device how many write buffer addresses are loaded with data and therefore when to expect the Program Buffer to flash confirm command. The Sector Address must match in the Write to Buffer command and the Write Word Count command. The Sector to be programmed must be unlocked (unprotected).

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-Line address. The Sector address must match the Write to Buffer Sector Address or the operation will abort and goes to the Abort state. All subsequent address / data pairs must be in sequential order. All write buffer addresses must be within the same Line. If the system attempts to load data outside this range, the operation will abort and go to the Abort state.

The counter decrements for each data load operation. Note that while counting down the data writes, every write is considered to be data being loaded into the write buffer. No commands are possible during the write buffer loading period. The only way to stop loading the write buffer is to write with an address that is outside the Line of the programming operation. This invalid address will immediately abort the Write to Buffer command.

Once the specified number of write buffer locations has been loaded, the system must then write the Program Buffer to Flash command at the Sector Address. The device then goes busy. The Embedded Program algorithm automatically programs and verifies the data for the correct data pattern. The system is not required to provide any controls or timings during these operations. If an incorrect number of write buffer locations have been loaded the operation will abort and goes to the Abort state. The abort occurs when anything other than the Program Buffer to Flash is written when that command is expected at the end of the word count.

The write-buffer embedded programming operation can be suspended using the Program Suspend command. When the Embedded Program algorithm is complete, the EAC then returns to the EAC standby or Erase Suspend standby state where the programming operation was started.

The system can determine the status of the program operation by using Data Polling Status, reading the Status Register, or monitoring the RY/BY# output. See Status Register on page 36 for information on these status bits. See Data Polling Status on page 37 for information on these status bits. See Figure 5.2 on page 24 for a diagram of the programming operation.

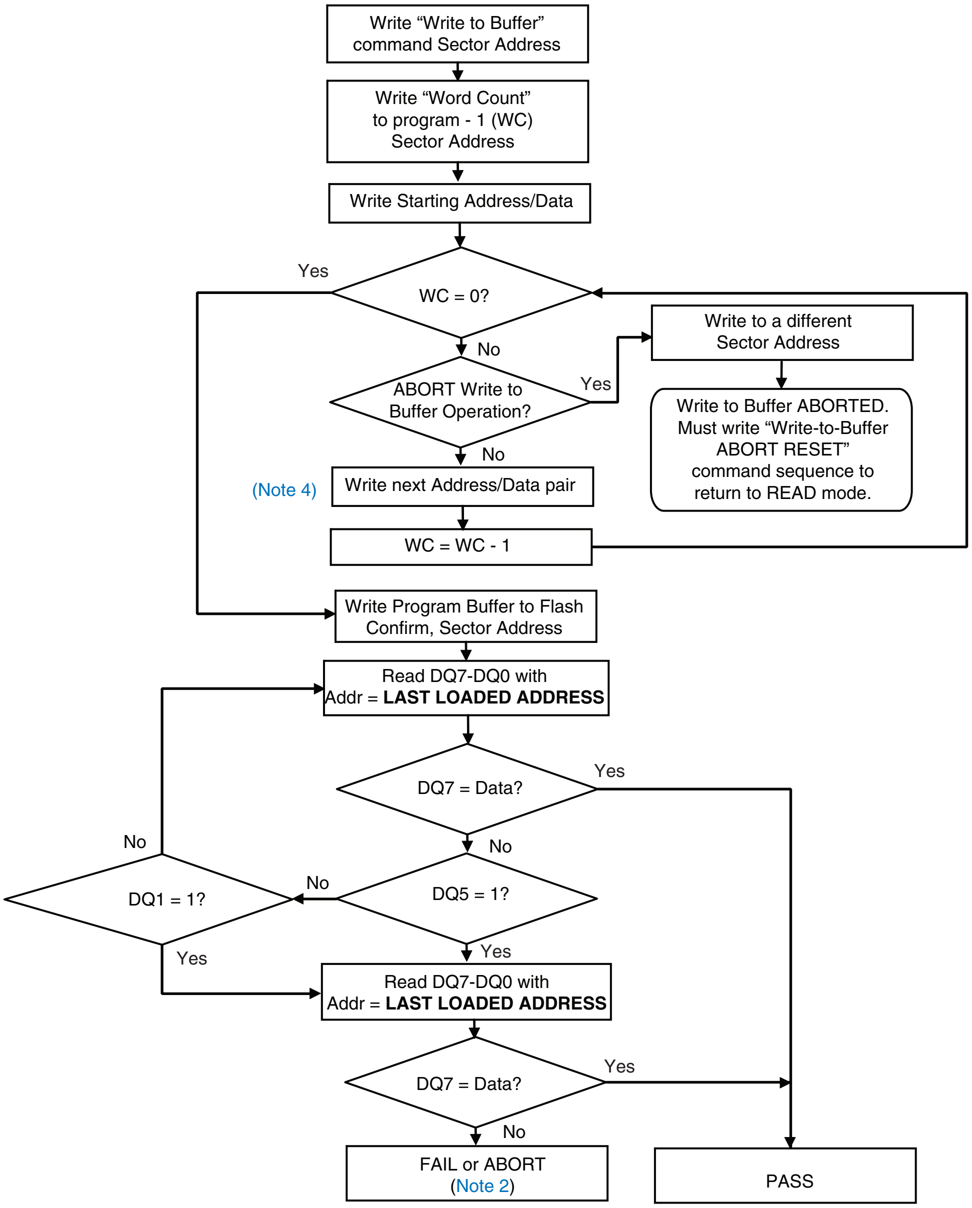
The Write Buffer Programming Sequence will be aborted under the following conditions:

- Load a Word Count value greater than the buffer size (255).
- Write an address that is outside the Line provided in the Write to Buffer command.
- The Program Buffer to Flash command is not issued after the Write Word Count number of data words is loaded.

When any of the conditions that cause an abort of write buffer command occur the abort will happen immediately after the offending condition, and will indicate a Program Fail in the Status Register at bit location 4 (PSB = 1) due to Write Buffer Abort bit location 3 (WBASB = 1). The next successful program operation will clear the failure status or a Clear Status Register may be issued to clear the PSB status bit.

The Write Buffer Programming Sequence can be stopped by the following: Hardware Reset or Power cycle. However, these using either of these methods may leave the area being programmed in an intermediate state with invalid or unstable data values. In this case the same area will need to be reprogrammed with the same data or erased to ensure data values are properly programmed or erased.

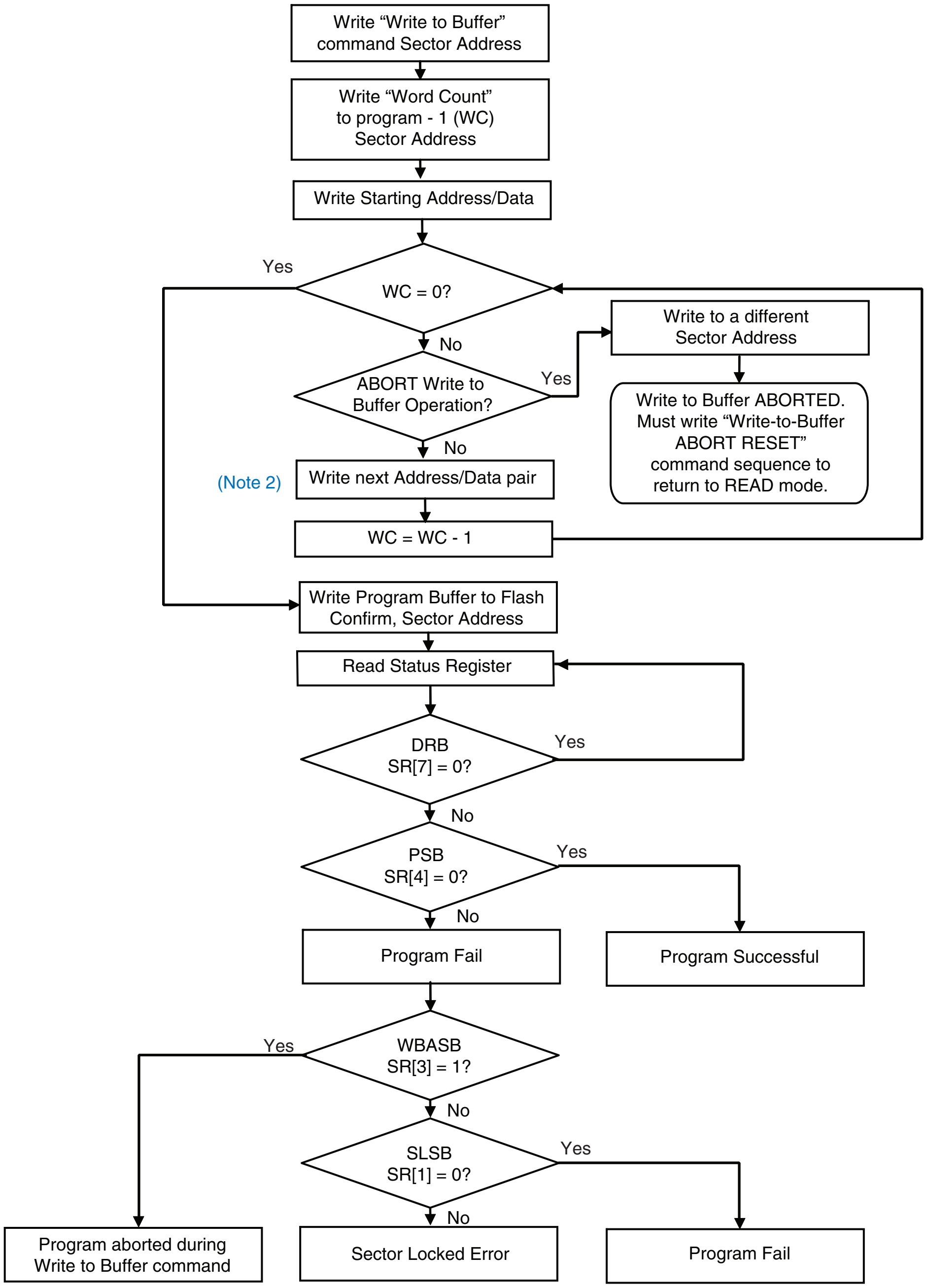
Figure 5.2 Write Buffer Programming Operation with Data Polling Status


Write "Write to Buffer" command Sector Address
Write "Word Count" to program - 1 (WC) Sector Address
Write Starting Address/Data
WC = 0?
Yes
No
ABORT Write to Buffer Operation?
Yes
Write to a different Sector Address
Write to Buffer ABORTED. Must write "Write-to-Buffer ABORT RESET" command sequence to return to READ mode.
No
(Note 4)
Write next Address/Data pair
WC = WC - 1
Write Program Buffer to Flash Confirm, Sector Address
Read DQ7-DQ0 with Addr = LAST LOADED ADDRESS
DQ7 = Data?
Yes
No
DQ1 = 1?
No
No
DQ5 = 1?
Yes
Yes
Read DQ7-DQ0 with Addr = LAST LOADED ADDRESS
DQ7 = Data?
Yes
No
FAIL or ABORT (Note 2)
PASS

Notes:

1. *DQ7 should be rechecked even if DQ5 = 1 because DQ7 may change simultaneously with DQ5.*
2. *If this flowchart location was reached because DQ5 = 1, then the device FAILED. If this flowchart location was reached because DQ1 = 1, then the Write Buffer operation was ABORTED. In either case the proper RESET command must be written to the device to return the device to READ mode. Write-Buffer-Programming-Abort-Rest if DQ1 = 1, either Software RESET or Write-Buffer-Programming-Abort-Reset if DQ5 = 1.*
3. *See* Table 7.1, Command Definitions x16 *on page 48* for the command sequence as required for Write Buffer Programming.
4. *When Sector Address is specified, any address in the selected sector is acceptable. However, when loading Write-Buffer address locations with data, all addresses MUST fall within the selected Write-Buffer Page.*

Figure 5.3 Write Buffer Programming Operation with Status Register



Write "Write to Buffer" command Sector Address
Write "Word Count" to program - 1 (WC) Sector Address
Write Starting Address/Data
Yes
WC = 0?
No
ABORT Write to Buffer Operation?
Yes
Write to a different Sector Address
Write to Buffer ABORTED. Must write "Write-to-Buffer ABORT RESET" command sequence to return to READ mode.
No
(Note 2)
Write next Address/Data pair
WC = WC - 1
Write Program Buffer to Flash Confirm, Sector Address
Read Status Register
DRB SR[7] = 0?
Yes
No
PSB SR[4] = 0?
Yes
No
Program Fail
Program Successful
WBASB SR[3] = 1?
Yes
No
SLSB SR[1] = 0?
Yes
No
Program aborted during Write to Buffer command
Sector Locked Error
Program Fail

Notes:

1. *See* Table 7.1, Command Definitions x16 *on page 48* for the command sequence as required for Write Buffer Programming.
2. *When Sector Address is specified, any address in the selected sector is acceptable. However, when loading Write-Buffer address locations with data, all addresses MUST fall within the selected Write-Buffer Page.*