AN98586

Migration from GL-N and GL-P to GL-S Flash

Authors: Doug Kearns, Gernot Hoyler

Associated Part Family:S29GL-N/P/S

AN98586 outlines the differences and potential issues when migrating from the Cypress GL-N and GL-P to the GL-S flash family (128 Mb and larger; except GL064S). It discusses affected device features, timing parameters as well as packages.

Contents

1 Introduction 1
2 GL-N, GL-P, and GL-S Flash Feature Comparison 1
3 Feature Difference Discussion 3
3.1 Density 3
3.2 Sector Architecture 3
3.3 Data Bus Width 3
3.4 Read Page Buffer Size 3
3.5 Write Buffer Size 3
3.6 High Voltage Accelerated Programming 3
3.7 Autoselect Register Access 4
3.8 Device ID 4
3.9 Unlock Bypass 4
3.10 Multi-Sector Erase 5
3.11 Secure Silicon OTP Area 5
3.12 Individual Sector Write Protection 5
3.13 Data Polling 5
3.14 Status Register 6
3.15 Blank Check 6
3.16 CFI Register 7
3.17 Lock Register Differences 8
4 Power On and Warm Reset Timing 8
5 DC and AC Parameter Differences 10
6 Packaging 11
7 Appendix — Status Register Read Source Code Example 16
Document History Page 19
Worldwide Sales and Design Support 20

1 Introduction

Cypress continues to extend the MirrorBit® GL family of 3V NOR flash with the introduction of the GL-S parallel NOR family based on 65 nm process technology. Cypress developed the GL-S flash family with migration in mind and most existing applications developed for GL-N and GL-P flash can also use GL-S flash. This document outlines the product differences that require attention in order to migrate to GL-S flash (except GL064S).

Note: This document discusses the use of GL-S family members from 128 Mbit to 1 Gbit density (S29GLxxxS) and the 2 Gbit density provided in a Dual Die Package (DDP) using a pair of 1 Gbit devices (S70GL02GS). This document does not address the unique feature set of the discrete 2 Gbit (S29GL02GS) device for gaming applications; nor does it cover the S29GL064GS.

2 GL-N, GL-P, and GL-S Flash Feature Comparison

Table 1 provides an overview of the features of the GL-N, GL-P, and GL-S flash families. The 65 nm MirrorBit Eclipse GL-S flash provides significantly improved program and erase performance for high density 3V parallel NOR applications, while maintaining basic hardware and software compatibility to allow use on existing designs which use GL-N and/or GL-P flash. The feature differences are discussed in more detail in Section 3.

One key difference between the legacy GL-N and GL-P families and the GL-S family is the underlying device architecture; the former uses a custom, hard-wired state machine while the latter uses a general purpose microprocessor core running factory-programmed firmware. The microprocessor architecture provides many benefits in terms of feature set, user configurability, factory configurability, and reduced factory development and test time. However it also means more time is required to initialize the flash device.

The differences in Power-On-Reset (POR) timing are discussed in Section 4, *Power On and Warm Reset Timing* on page 8. The differences in DC and AC specifications are detailed in Section 5, *DC and AC Parameter Differences* on page 10.

The differences in footprint and packaging are discussed in Section 6, *Packaging* on page 11.

Table 1. GL Family Feature Comparison

	S29GL-N 110 nm	S29GL-P 90 nm	S29GL-S 65 nm	Migration Issues	See Section
Density					
128 Mbit	√	√	√	No	3.1 Density
256 Mbit	√	√	√	No	
512 Mbit	√	√	√	No	
1024 Mbit	–	√	√	No	
2048 Mbit	–	√ (multi-die)	√ (multi-die)	No	
Sector Architecture					
Uniform 128 kB	√	√	√	No	3.2 Sector Architecture
Access					
x16 Data Bus Width	√	√	√	No	3.3 Data Bus Width
x8 Data Bus Width	√	√	–	Yes	
Asynchronous	√	√	√	No	
Read Page Mode	√	√	√	No	
Read Page Buffer Size	16 Byte	16 Byte	32 Byte	No	3.4 Read Page Buffer Size
Write Buffer Size	32 Byte	64 Byte	512 Byte	No	3.5 Write Buffer Size
Security					
Secure Silicon OTP Area	256 Byte	256 Byte	2 x 512 Byte	Maybe	3.11 Secure Silicon OTP Area
Individual Sector Write Protection	√	√	√	No	3.12 Individual Sector Write Protection
Other					
12V Accelerated Programming	√	√	–	Maybe	3.6 High Voltage Accelerated Programming
High Voltage Autoselect Access	√	√	–	Maybe	3.7 Autoselect Register Access
Device ID	√	√	√	No	3.8 Device ID
Unlock Bypass Command	√	√	–	Maybe	3.9 Unlock Bypass
Multi-Sector Erase	√	√	–	Maybe	3.10 Multi-Sector Erase
Status via Data Polling	√	√	√	No	3.13 Data Polling
Status via Status Register	–	–	√	No	3.14 Status Register, 7 Appendix — Status Register Read Source Code Example
Blank Check	–	–	√	No	3.15 Blank Check
CFI Version	1.3	1.3	1.5	Maybe	3.16 CFI Register
Lock Register	LIke GL-P	Like GL-N	Updated	Maybe	3.17 Lock Register Differences
POR and Warm Reset Timing	Like GL-P	Like GL-N	Updated	Maybe	4 Power On and Warm Reset Timing
DC and AC Parameters	Like GL-P	Like GL-N	Updated	Maybe	5 DC and AC Parameter Differences
Packaging and Ordering Options					
64-ball BGA 10 x 13 mm (FAA064)	√	√	–	Yes	6 Packaging
64-ball BGA 11 x 13 mm (LAA064)	√	√	√	No	
64-ball BGA 9 x 9 mm (LAE064)	–	–	√	No	
SnPb Solder Option (RoHS 5/6)	√	√	√	No	

3 Feature Difference Discussion

3.1 Density

The S29GL-S flash is available in monolithic 128, 256, 512, and 1024 Mbit densities. The multi-die S70GL-S flash is available in 2048 Mbit density only.

3.2 Sector Architecture

The GL-S flash has a 128 kB uniform sector architecture, identical to that in the GL-N and GL-P flash.

3.3 Data Bus Width

The GL-S flash supports only a x16 Data Bus Width, e.g. DQ[15:0]. The BYTE# input on the x8/x16 GL-N and GL-P flash devices, pin 53 on 56-TSOP and ball F7 on 64-BGA, is labeled “Reserved for Future Use” (RFU) on the GL-S and that input is not connected internally. Migration to GL-S flash is only possible on existing GL-N and GL-P designs that have the BYTE# input pulled > V_{IH} to force operation in x16 Data Bus Width.

3.4 Read Page Buffer Size

The GL-S flash has a 32 byte (16 word) read page buffer, which is double the length of the legacy GL-N and GL-P flash to facilitate larger processor cache line fill operations. No software modifications are required to operate with 16 byte maximum read page transfers supported by the GL-N and GL-P flash.

Software can be modified to take advantage of the larger GL-S read page buffer by querying the CFI page mode register at CFI word offset 4Ch and configuring software to perform additional page mode read cycles.

3.5 Write Buffer Size

The GL-S flash has a 512 byte (256 word) write buffer, sixteen times the size of the GL-N flash and eight times the size of the GL-P. The larger write buffer facilitates faster programming throughput and better data alignment with most file systems. No software modifications are required to operate with a 32 byte or 64 byte maximum write buffer fill supported by the GL-N and GL-P flash, respectively.

Software can be modified to take advantage of the larger GL-S write buffer by querying the CFI programming buffer size register at CFI word offset 2Ah and configuring software to perform large write buffer fills. It is recommended that GL-S buffer writes be performed on multiples of 32-byte pages to maximize data integrity; e.g. load the write buffer with one to sixteen 32-byte pages of data that are programmed into the array in a single operation.

3.6 High Voltage Accelerated Programming

The GL-S flash does not support accelerated programming, unlike the GL-N and GL-P flash which support high voltage accelerated programming when V_{HH} (nominally 12V) is applied to the WP#/ACC input.

The GL-S flash programming throughput is dramatically faster than the high voltage accelerated programming throughput of the GL-N and GL-P flash which removes the need for this legacy feature.

The GL-S does not support input levels greater than 4.0V on any input. If an existing design is enabled to support high voltage application for accelerated programming of GL-N or GL-P flash, it must be modified so that it does not apply voltages greater than V_{IO} to GL-S flash.

3.7 Autoselect Register Access

The Autoselect Register provides manufacturer and device identification through special identifier codes. The GL-S flash only supports Autoselect Register access via the Autoselect command (now called the ID Entry command), unlike the GL-N and GL-P flash, which support Autoselect Register access via the Autoselect command as well as a high voltage method which requires V_{ID} (nominally 12V) applied to Address input A9.

The GL-S does not support input levels greater than 4.0V on any input. If an existing design is enabled to support Autoselect Register access via the high voltage method, it must be modified so that it does not apply voltages greater than V_{IO} to GL-S flash.

In the GL-N and GL-P flash, the Autoselect Register is overlaid onto Sector Address zero (SA00). In the GL-S flash, the Autoselect Register is overlaid onto whichever sector is selected with the Autoselect Entry command. SW modification is not required to access the GL-S flash Autoselect Register in existing GL-N and GL-P designs.

Sector Lock Status can be determined by accessing Autoselect Register word offset 02h within the desired sector. In GL-N and GL-P flash, the lock protection status of multiple sectors can be determined after entering Autoselect mode at the flash base address. On GL-S flash, only the protection status of the sector selected in the Autoselect entry command can be determined. To determine the lock protection status of a different sector, Autoselect mode has to be exited and reentered using the desired sector in the Autoselect Entry command. See Lock Register Differences on page 8.

3.8 Device ID

The 128 Mbit, 256 Mbit, 512 Mbit, and 1024 Mbit GL-S flash have the same Device ID register values as the corresponding density GL-N and GL-P flash. The S70GL02GS flash has the same Device ID as the S70GL02GP flash.

Table 2. GL Flash Device IDs

Description	Address	Read Data
Device ID word 1	(SA) + 0001h	227Eh
Device ID word 2	(SA) + 000Eh	2248h = 2 Gbit 2228h = 1 Gbit 2223h = 512 Mbit 2222h = 256 Mbit 2221h = 128 Mbit
Device ID word 3	(SA) + 000Fh	2201h

Existing software which supports the GL-N or GL-P flash that uses Device ID to set up software command support does not require modification to enable compatibility with the GL-S flash. Use of specific CFI Register queries should be employed to take advantage of new GL-S superset features, such as larger read page buffers and write buffers. The CFI Process Generation bits at CFI word offset 45h provide for in-system determination of the unique GL family, e.g. GL-N: 0010h, GL-P: 0014h, GL-S: 001Ch.

Device ID can only be accessed via software Autoselect Register commands on the GL-S flash and not the optional high voltage method supported on the GL-N and GL-P flash. See Autoselect Register Access on page 3.

3.9 Unlock Bypass

The GL-S flash does not support Unlock Bypass mode programming, unlike the existing the GL-N and GL-P flash.

Unlock Bypass mode programming is a legacy feature used to decrease the command cycle overhead by 50% when performing programming with the single byte/word Program command only. Applications using high density GL devices rely on multi-word write buffer programming to maximize programming throughput. Write buffer programming has an inherently low effective command overhead and supports single byte/word programming. If an existing design is enabled to support Unlock Bypass programming, it must be modified so that it does not use Unlock Bypass commands with GL-S flash.

3.10 Multi-Sector Erase

The GL-S flash does not support multi-sector erase or the Chip Erase command, unlike the GL-N and GL-P flash.

Multi-sector erase is a legacy feature of the Sector Erase command that allows queueing of multiple sector erase operations within one Sector Erase command string to minimize command overhead. This is a rarely used function so removal of this feature should not prevent migration to GL-S flash on most existing designs. Those applications that do support multi-sector erase require modification to send a separate erase command for each sector erase operation.

On GL-N and GL-P flash, the DQ3 bit does not transition to logic 1 until ~50 microseconds after the sixth sector erase command cycle to allow entry of additional sector address and erase command pairs; since the GL-S does not support the multi-sector erase feature, the ~50 microsecond delay is not required.

3.11 Secure Silicon OTP Area

The S29GL-S flash devices have 1024 bytes of one time programmable (OTP) memory. This Secure Silicon Region (SSR) is divided into two areas, the lower 512B region, SSR Region 0, is Factory modifiable and the upper 512B region, SSR Region 1, is Customer modifiable. SSR0 can be ordered Factory pre-programmed. SSR0 is Factory locked and cannot be Customer modified. The Secure Silicon Region can only be accessed after writing the Secure Silicon Entry command and is mapped into the lower 1 kB of the sector selected during the entry command. SSR Region 0 is overlaid onto word offset 0x0000 to 0x00FF of the selected sector. SSR Region 1 is overlaid onto word offset 0x0100 to 0x01FF of the selected sector. Memory outside of the 1 kB Secure Silicon Region has undefined data when in Secure Silicon access mode.

The GL-N and GL-P flash have 256B of OTP in the Secure Silicon Sector region. This region can be ordered Factory preprogrammed and locked, or it can be programmed and locked by the user. This region can only be accessed in Secure Silicon Access mode and is overlaid onto word 0x0000 to 0x007F of SA0.

The Secure Silicon Lock Register bit usage is different for GL-S. The GL-S Lock Register DQ0 is Factory programmed to 0 to indicate SSR Region 0 is Factory locked. The GL-S Lock Register bit 6 is Factory preset to 1 to indicate SSR Region 1 is unlocked and it can be programmed to 0 by the Customer to lock SSR Region 1. The GL-N and GL-P Lock Register DQ0 is by default preset to 1 to indicate the Secure Silicon Region is unlocked. It can be Factory set to 0 if Factory preprogramming is ordered or can be programmed to 0 by the Customer to lock the Secure Silicon Region.

In the GL-S flash, the Factory Lockable SSR Region 0 Lock Status bit is bit 7 at x16 offset 003h in the Autoselect Register and the User Lockable SSR Region 1 Lock Status bit is bit 6 at word offset 003h in the Autoselect Register. In the GL-N and GL-P flash, the Secure Silicon Sector Lock Status bit is bit 7 at word offset 003h in the Autoselect Register, and when set, indicates the Secure Silicon Region is locked.

The S70GL02GS flash has two 1 kB OTP areas, one in each GL01GS die. Access to the upper region requires flash input A26 = 1 during the Secure Silicon Register Access command cycles.

3.12 Individual Sector Write Protection

The GL-S flash supports the Advanced Sector Protection (ASP) feature that provides software enabled program and erase protection on a sector basis utilizing user configurable 8-byte password, non-volatile and volatile control, consistent with the GL-N and GL-P flash.

3.13 Data Polling

GL-S flash supports legacy data polling to determine the status of embedded programming and erase operations. The implementation is consistent with the GL-N and GL-P flash and no software modification is required to continue using data polling routines when migrating to the GL-S flash.

After the six Sector Erase command initiation cycles are written to GL-S, the DQ3 bit immediately transitions to logic 1, indicating the erase operation has begun.

If a DQ5 time out event occurs on GL-S flash, a software reset command is required to clear DQ5 and to return the flash to a ready state. It can take up to 2 µs for the flash to stop communicating that it is busy following this reset command.

Data polling may not be supported on future generations of MirrorBit Eclipse GL flash families. Status Register reads are required to determine the status of embedded program and erase operations if data polling is not supported.

3.14 Status Register

The GL-S flash supports Status Register reads as an alternative method to Data Polling for determination of embedded operation status. The Status Register feature is not supported on the GL-N and GL-P flash.

The 16-bit Status Register is accessed via a two cycle sequence consisting of a Read Status Register Command write cycle followed immediately by a read cycle to the same targeted sector address. Using the Status Register is advantageous because, unlike legacy data polling, software does not need to track active address regions or compare sequential polling read values to determine embedded algorithm status; one Status Register access provides all the information necessary to determine the flash state. A Clear Status Register command is available to reset the last completed embedded operation portion of the Status Register.

Status Register usage is optional and existing designs utilizing GL-N and GL-P flash do not have to use this feature. If desired, software can be modified to take advantage of this feature by querying the Lower Software Bits

at offset 000Ch in Autoselect mode. If bit 0 is set, Status Register functionality is supported. Section 7, *Appendix — Status Register Read Source Code Example* on page 15 contains sample C source code showing how to implement Data Polling and Status Register accesses.

Full details of the Status Register implementation are provided in the GL-S flash data sheet (S29GL_128S_01GS_00). The Status Register bit definitions are provided in Table 3.

Table 3. Status Register Bit Definition

Status Register Bit	Description	Name	Reset Value	Busy Status	Ready Status
15:08	Reserved		x	Invalid	x
7	Device Ready Bit	DRB	1	0	1
6	Erase Suspend Status Bit	ESSB	0	Invalid	0: No Erase In Suspension 1: Erase In Suspension
5	Erase Status Bit	ESB	0	Invalid	0: Erase Successful 1: Erase Failed
4	Program Status Bit	PSB	0	Invalid	0: Program Successful 1: Program Failed
3	Write Buffer Abort Status Bit	WBASB	0	Invalid	0: Program Not Aborted 1: Program Aborted During Write Buffer Command
2	Program Suspend Status Bit	PSSB	0	Invalid	0: No Program In Suspension 1: Program In Suspension
1	Sector Lock Status Bit	SLSB	0	Invalid	0: Sector Not Locked During Operation 1: Sector Locked Error Operation
0	Reserved		0	Invalid	x

Notes:

1. *Bits 15 thru 8, and 0 are reserved for future use and may display as 0 or 1. These bits should be ignored (masked) when checking status.*
2. *Bit 7 is 1 when there is no Embedded Algorithm in progress in the device.*
3. *Bits 6 thru 1 are valid only if bit 7 is 1.*
4. *All bits are put in their reset status by cold reset or warm reset.*
5. *Bits 5, 4, 3, and 1 are cleared to 0 by the Clear Status Register command or Reset command.*
6. *Upon issuing the Erase Suspend Command, the user must continue to read status until DRB becomes 1.*
7. *ESSB is cleared to 0 by the Erase Resume Command.*
8. *ESB reflects success or failure of the most recent erase operation.*
9. *PSB reflects success or failure of the most recent program operation.*
10. *During erase suspend, programming to the suspended sector will cause program failure and set the Program status bit to 1.*
11. *Upon issuing the Program Suspend Command, the user must continue to read status until DRB becomes 1.*
12. *PSSB is cleared to 0 by the Program Resume Command.*
13. *SLSB indicates that a program or erase operation failed because the sector was locked.*
14. *SLSB reflects the status of the most recent program or erase operation.*

3.15 Blank Check

The GL-S flash supports a sector Blank Check feature that enables system software to minimize latency associated with erasures prior to code updates. Use of this feature is optional and is not supported in the GL-N and GL-P flash. The new Blank Check feature has no effect on existing designs. Please refer to the GL-S data sheets for Blank Check feature implementation details.

3.16 CFI Register

Table 4 provides a listing of all the Common Flash Interface (CFI) register values that are different for the GL-N, GL-P, and GL-S flash families. Software can access the CFI register to determine device specific features such as array size, command set, page size, and programming time and use these values to self-configure for optimal

performance. GL-S supports the CFI version 1.5, which is an extended address range revision of the legacy CFI version 1.3 supported by the GL-N and GL-P flash.

Table 4. CFI Register Differences

CFI Register	Word Offset	GL-N	GL-P	GL-S
Typical timeout for single word write = 2^N µs	1Fh	0007h	0006h	0008h
Typical timeout for maximum multi-byte program = 2^N µs	20h	0007h	0009h	0009h
Typical timeout for individual block erase = 2^N ms	21h	000Ah	0009h	0008h
Typical timeout for full chip erase = 2^N ms 0000h = Not Supported	22h	0000h	0013h (1 Gb) 0012h (512 Mb) 0011h (256 Mb) 0010h (128 Mb)	0012h (1 Gb) 0011h (512 Mb) 0010h (256 Mb) 000Fh (128 Mb)
Maximum timeout for single word = 2^N times typical	23h	0003h	0003h	0001h
Maximum timeout for maximum multi-byte program = 2^N times typical	24h	0005h	0005h	0002h
Maximum timeout for individual block erase = 2^N times typical	25h	0004h	0003h	0003h
Maximum timeout for full chip erase = 2^N times typical 0000h = Not Supported	26h	0000h	0002h	0003h
Flash Device Interface Description 0000h = x8-only, 0001h = x16-only, 0002h = x8/x16-capable	28h	0002h	0002h	0001h
Maximum number of bytes in multi-byte write = 2^N	2Ah	0005h	0006h	0009h
Minor version number, ASCII	44h	0033h	0033h	0035h
Process Technology (Bits 5-2): 0100b = 110 nm MirrorBit, 0101b = 90 nm MirrorBit, 0111b = 65 nm MirrorBit Eclipse Address Sensitive Unlock (Bits 1-0): 00b = Required, 01b = Not Required	45h	0010h	0014h	001Ch
Page Mode Type 0002h = 8-word Page, 0003h = 16-word Page	4Ch	0002h	0002h	0003h
ACC (Acceleration) Supply Minimum 0000h = Not Supported, D[7:4] = V, D[3:0] = 100 mV	4Dh	00B5h	00B5h	0000h
ACC (Acceleration) Supply Maximum 0000h = Not Supported, D[7:4] = V, D[3:0] = 100 mV	4Eh	00C5h	00C5h	0000h
Unlock Bypass 0000h = Not Supported, 0001h = Supported	51h	–	–	0000h
Secure Silicon Sector (Customer OTP Area) Size = 2^N bytes	52h	–	–	0009h
Software Features	53h	–	–	008Fh
Read Page Size = 2^N bytes	54h	–	–	0005h
Erase Suspend Timeout Maximum < 2^N µs	55h	–	–	0006h
Program Suspend Timeout Maximum < 2^N µs	56h	–	–	0006h
Embedded Hardware Reset Timeout Maximum < 2^N µs	78h	–	–	0006h
Non-embedded Hardware Reset Timeout Maximum < 2^N µs	79h	–	–	0009h

Certain software drivers verify specific values in the CFI register to assure support for the specific flash. Linux MTD drivers verify the Major and Minor version number (ASCII) entries at CFI register word offsets 43h and 44h, respectively, to determine if device specific support is built into the MTD. GL-N and GL-P flash have a Major version number value of 0031h (ASCII ‘1’) and a Minor version number value of 0033h (ASCII ‘3’) to indicate the CFI Register follows the CFI 1.3 standard. The GL-S flash have a Major version number value of 0031h (ASCII ‘1’) and a Minor version number value of 0035h (ASCII ‘5’) to indicate the CFI Register follows the CFI 1.5 standard. This difference causes older MTD drivers to not recognize the GL-S flash. In that case, a software patch is required to update the MTD for GL-S support. An appropriate Linux driver patch can be downloaded from www.cypress.com.

3.17 Lock Register Differences

There are several changes to the Lock Register for GL-S flash, see Table 5.

The DQ7 ‘Reserved’ bit is factory set to either 0 or 1 on GL-S flash. This bit is factory preset to 1 on GL-N and GL-P flash.

- The DQ6 ‘SSR Region 1 (Customer) Lock Bit’ on GL-S is factory set to 1 and can be customer set to 0 to permanently write protect the 512 byte SSR 1 OTP region. This bit is ‘Reserved’ and factory preset to 1 on GL-N and GL-P flash.
- The DQ0 ‘SSR Region 0 (Factory) Lock Bit’ on GL-S is preset at the factory to 0 to permanently write protect the 512 byte SSR 0 OTP region. On GL-N and GL-P this bit enables locking of the Secure Silicon Region by either the Factory or Customer. If the Secure Silicon Region is Factory preprogrammed, this bit is 0, which indicates the Secure Silicon Region is locked; otherwise it is Factory preset to 1 and can be set to 0 by the Customer to lock the Secure Silicon Region.

Note: The Customer is not required to program DQ2 or DQ1, and DQ6 or DQ0 bits at the same time on GL-S flash. This allows the Customer to lock the SSR before or after the device protection scheme has been selected. When programming the Lock Register, all ‘Reserved’ bits should be written as 1 (masked).

Table 5. Lock Register Differences

Lock Register	GL-N and GL-P Flash			GL-S Flash		
Bit	Definition	Default	Customer Alterable	Definition	Default	Customer Alterable
DQ[15:9]	Reserved	1111111b	no	Reserved	1111111b	no
DQ8	Reserved	1b	no	Reserved	0b	no
DQ7	Reserved	1b	no	Reserved	0b/1b	no
DQ6	Reserved	1b	no	SSR Region 1 (Customer) Lock Bit	1b	yes
DQ5	Reserved	1b	no	Reserved	1b	no
DQ4	Reserved	1b	no	Reserved	1b	no
DQ3	Reserved	1b	no	Reserved	1b	no
DQ2	Password Protection Mode Lock Bit	1b	yes	Read Password Protection Mode Lock Bit	1b	yes
DQ1	Persistent Protection Mode Lock Bit	1b	yes	Persistent Protection Mode Lock Bit	1b	yes
DQ0	Secure Silicon Sector Protection Bit	1b	yes	SSR Region 0 (Factory) Lock Bit	0b	no

4 Power On and Warm Reset Timing

At power on, the flash requires more time in the reset state to initialize than it does during a warm reset. Table 6 and Figure 1 and Figure 2 detail the power on and warm reset timing requirements for the GL-N, GL-P, and GL-S flash.

Table 6. Power On and Warm Reset Timing Requirements

Parameter	Description	Type	GL-N	GL-P	GL-S
Power on Reset					
t_{VCS}	V_{CC} Setup Time to first access	min	35 µs	35 µs	300 µs
t_{VIOS}	V_{IO} Setup Time to first access	min	35 µs	35 µs	300 µs
t_{RPH}	RESET# Low to CE# Low	min	35 µs	35 µs	35 µs
t_{RP}	RESET# Low to RESET# High	min	500 ns	35 µs	200 ns (2)
t_{RH}	RESET# High to CE# Low	min	50 ns	200 ns	50 ns (2)
t_{CEH}	CE# High to CE# Low	min	N/A	N/A	20 ns
Warm Reset					
t_{RPH}	RESET# Low to CE# Low	min	20 µs (3)	35 µs	35 µs

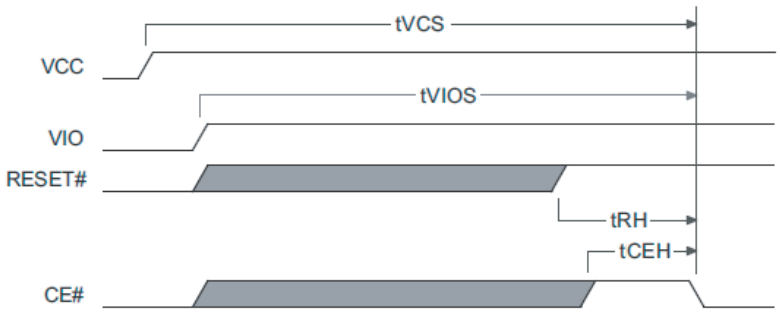
Table 6. Power On and Warm Reset Timing Requirements

Parameter	Description	Type	GL-N	GL-P	GL-S
t_{RP}	RESET# Low to RESET# High	min	500 ns	35 µs	200 ns (2)
t_{RH}	RESET# High to CE# Low	min	50 ns	200 ns	50 ns (2)
t_{CEH}	CE# High to CE# Low	min	N/A	N/A	20 ns

Notes:

1. *N/A = Not Applicable.*
2. *For GL-S, t_{RP} + t_{RH} must not be less than t_{RPH}.*
3. *For GL-N, t_{RP} = 20 µs during embedded operation and t_{RP} = 500 ns.*

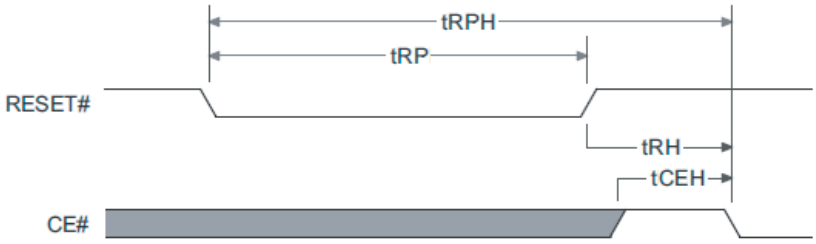
Figure 1. Power-Up Reset Timing



Note:

The sum of t_{RP} and t_{RH} must be equal to or greater than t_{RPH}.

Figure 2. Warm Reset Timing



Note:

The sum of t_{RP} and t_{RH} must be equal to or greater than t_{RPH}.

The differences in power-on timing should not present a migration challenge for most applications where the flash interfaces directly with a Host that requires oscillator and PLL lock prior to initiating the first boot read access to the flash. In applications which may access the flash within 300 µs of power application, some circuit modification is required to accommodate migration to GL-S flash.

To initiate the first read or write cycle after power on, the GL-S requires CE# to transition from High to Low no sooner than t_{VCS} after V_{CC} exceeds V_{CC_min} and V_{IO} exceeds V_{IO_min}. CE# must be High at least t_{CEH} = 20 ns prior to CE# falling edge which initiates the first access. These are not requirements for GL-N and GL-P so designs that have CE# fixed low cannot migrate to GL-S without modification to enable active CE# control.

CE# is ignored during Warm Reset; however, to initiate the first read or write cycle after warm reset, the GL-S requires CE# to transition from High to Low no sooner than t_{RH} after RESET# transitions from Low to High. CE# must be high at least t_{CEH} = 20 ns prior to CE# falling edge, which initiates first access. These are not requirements for GL-N and GL-P so designs that have CE# fixed low cannot migrate to GL-S without modification to enable active CE# control.

The GL-S allows V_{IO} to ramp concurrently with or after V_{CC} with no restriction on time or voltage differential. During power ramp no input is allowed to exceed V_{IO}. The GL-S data sheet provides enhanced direction on power management and control to design a robust and reliable system. In general, this additional guidance in the GL-S data sheet also applies to GL-N and GL-P flash.

5 DC and AC Parameter Differences

Table 7 provides a side by side reference of DC specification differences. The GL-S does not support the application of more than 4.0V to any input. The GL-S maximum logic levels are specified differently than GL-N and GL-P and are more in line with industry standards for CMOS application specifications, e.g.; it is unrealistic for low current CMOS inputs to have a steady state logic '0' at more than 20% of V_{IO}. The difference in maximum V_{IH} does not impact logic transition points or timing parameter specifications and should not cause migration issues.

Table 7. DC Specification Differences

Parameter	Description	Type	GL-N	GL-P	GL-S
Input Levels					
V_{IO}	All I/O other than A9 and ACC	max	4.0V	4.0V	4.0V
V_{IO}	A9 and ACC	max	12.5V	12.5V	4.0V
Logic Levels					
V_{IH}	Input High Voltage	max	V_{IO} + 0.3V	V_{IO} + 0.3V	V_{IO} + 0.4V
Power Usage					
I_{CC1}	Active V_{CC} + V_{IO} Read (5 MHz)	max	50 mA	55 mA	60 mA
I_{CC2}	Active V_{CC} Intra-Page Read (33 MHz)	max	20 mA	20 mA	25 mA
I_{CC3}	Active Program or Erase	max	90 mA	90 mA	100 mA
I_{CC4}	Standby Current	max	5 µA	5 µA	100 µA
I_{CC5}	Reset Current	max	5 µA	500 µA	20 mA
I_{CC6}	Automatic Sleep Current (1)	max	5 µA	5 µA	150 µA

Note:

1. *GL-S enables the specified automatic sleep current consumption within t_{ACC} + 30 ns of Address change while CE# < V_{IL} and transitions to standby mode within 8 µs of additional control signal inactivity.*

Table 8 provides side by side comparisons of AC parameter specification differences between GL families (minus reset timing parameter differences documented in Table 6). All parameters should be reviewed against actual application implementations to ensure a successful migration. For applications that use the erase suspend and/or program suspend features, it is important to review the system software ramifications of the GL-S having longer latency between issuance of the suspend and resume commands and the flash updating status and completing the transition between modes.

Table 8. AC Specification Differences

Parameter	Description	Type	GL-N (1)	GL-P (1)	GL-S (1)
Async Read					
t_{ACC} / t_{CE} / t_{RC}	Read Cycle Time	min	100 ns	100 ns	100 ns
t_{PACC}	Intra-Page Access Time	min	25 ns	25 ns	15 ns
t_{DF}	Control negate to data High-Z	min	20 ns	20 ns	15 ns
Async Write					
t_{WC}	Write Cycle Time	min	100 ns	100 ns	60 ns
t_{WP}	WE# Enable to Disable	min	35 ns	35 ns	25 ns
t_{WPH}	WE# Disable to Enable	min	30 ns	30 ns	20 ns
t_{DS}	Data Setup to WE# Disable	min	45 ns	30 ns	30 ns
t_{BUSY}	Erase/Program Valid to RY/BY# Delay	max	90 ns	90 ns	80 ns
Suspend Resume					
t_{ESL}	Erase Suspend / Erase Resume	max	20 µs	20 µs	40 µs
t_{PSL}	Program Suspend / Program Resume	max	15 µs	15 µs	40 µs
Array Update					
	Full Buffer Write Program Time (2)	typ	240 µs	480 µs	340 µs

Table 8. AC Specification Differences (Continued)

Parameter	Description	Type	GL-N (1)	GL-P (1)	GL-S (1)
	Effective per Word Write Buffer Program Time	typ	15 µs	15 µs	1.33 µs
	Single Word Program Time	typ	60 µs	60 µs	125 µs
	128 kB Sector Erase Time (4)	typ	500 ms	500 ms	200 ms
	Sector Erase Timeout	max	50 µs	50 µs	0s
Throughput					
	x16 Async Read	max	20 MB/s	18 MB/s	22 MB/s
	x16 Page Mode Read (3)	max	58 MB/s	56 MB/s	98 MB/s
	Programming	typ	133 kB/s	133 kB/s	1.5 MB/s
	Erase (4)	typ	262 kB/s	262 kB/s	655 kB/s

Notes:

1. *All table specifications apply to I-temp rated 512 Mbit density devices with V_{CC} = V_{IO} = 2.7-3.6V. Refer to individual data sheets for performance specifications for other densities and operating conditions.*
2. *Maximum Write Buffer Size varies: GL-N = 32B, GL-P = 64B, GL-S = 512B.*
3. *Page mode read throughput based on 8 word page accesses for GL-N and GL-P and 16 word page accesses for GL-S.*
4. *Excluding pre-programming to 0x00 prior to erasure*

6 Packaging

Standard S29GL-S flash are available in SnPb and Pb-free in 56-lead TSO056 leadframe package and two 64-ball ball grid array packages – LAA064 and LAE064. The electrical contact dimensions and footprints are compatible with the GL-N and GL-P flash in x16 data bus width applications.

The outer dimensions of the LAE064 package are 9 x 9 mm, forty three percent smaller than the 11 x 13 mm LAA064 package. Printed Circuit Board (PCB) layout changes are not required to use the LAE064 package on existing GL-N and GL-P designs; however, surface mount placement programs require modification for proper part placement.

The S70GL02GS flash is available in the same 64-ball LAA064 ball grid array package as the S70GL02GP flash. The footprint is compatible with the S70GL02GP flash in x16 data bus width applications.

Several of the connection definitions have changed, see Table 9.

Table 9. GL-N / GL-P / GL-S Pin Out Differences

Pin or Ball	GL-N	GL-P	GL-S	Migration Issue
TSOP Package				
16	WP#/ACC	WP#/ACC	WP#	Potential
27	NC	NC	RFU	No
28	NC	NC	DNU	Potential
30	NC	NC	RFU	No
51	DQ15/A-1	DQ15/A-1	DQ15	No
53	BYTE#	BYTE#	RFU	No
55	NC	NC/A25 (1)	NC/A25 (1)	No
BGA Package				
B1	NC	NC/A26 (2)	NC/A26 (2)	No
B4	WP#/ACC	WP#/ACC	WP#	Potential
E1	NC	NC	DNU	Potential
F7	BYTE#	BYTE#	RFU	No
G1	NC	NC	RFU	No
G7	DQ15/A-1	DQ15/A-1	DQ15	No
G8	NC	NC/A25 (1)	NC/A25 (1)	No

Legend:

NC = Not Connected internally (okay to use pad for routing).

RFU = Reserved for Future Use (not connected internally on current product).

DNU = Do Not Use (must be left floating, not okay to use pad for routing).

Notes:

1. *A25 only for S29GL01GP, S29GL01GS, S70GL02GP, and S70GL02GS.*
2. *A26 only for S70GL02GP and S70GL02GS.*

Figure 3 and Figure 4 depict the pin definitions for the GL-N/GL-P and GL-S flash TSOP packages, respectively.

Figure 5 and Figure 6 illustrate the pin definitions for GL-N/GL-P and GL-S flash BGA packages, respectively.

Figure 3. TSOP Pin Definitions: GL-N and GL-P

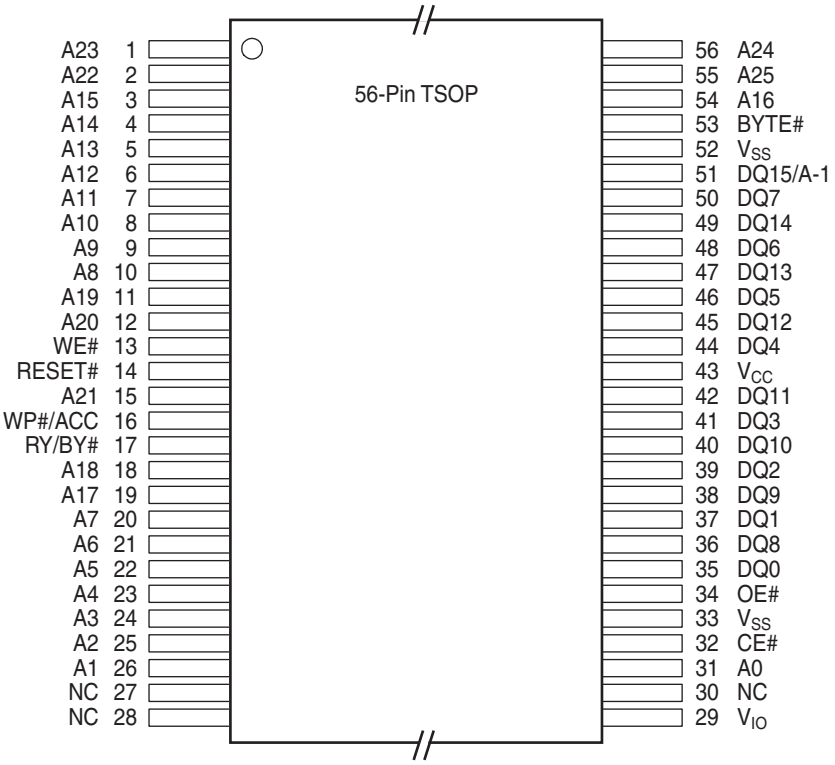


Figure 4. TSOP Pin Definitions: GL-S

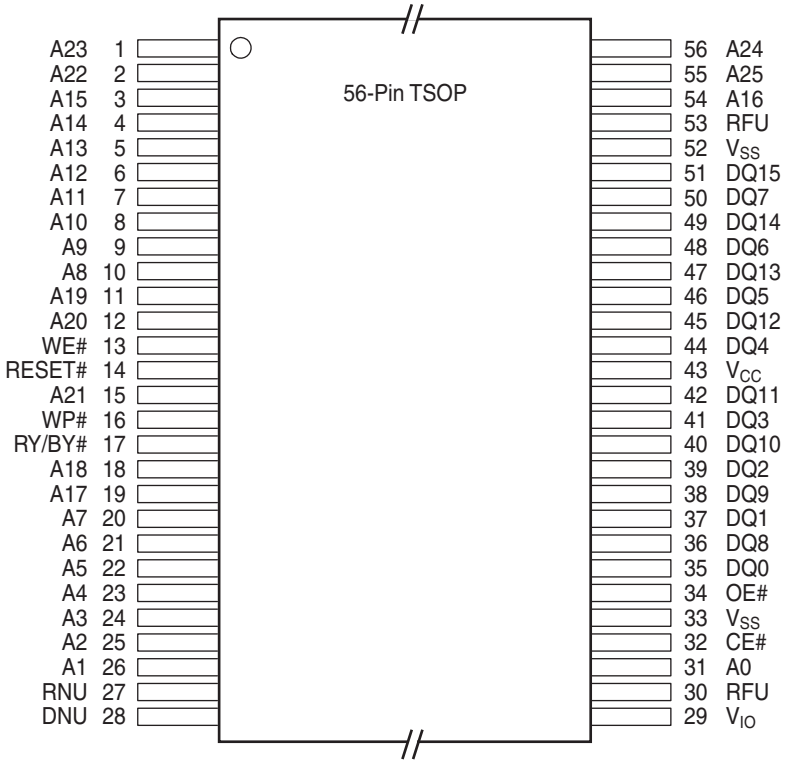


Figure 5. BGA Pin Definitions: GL-N and GL-P

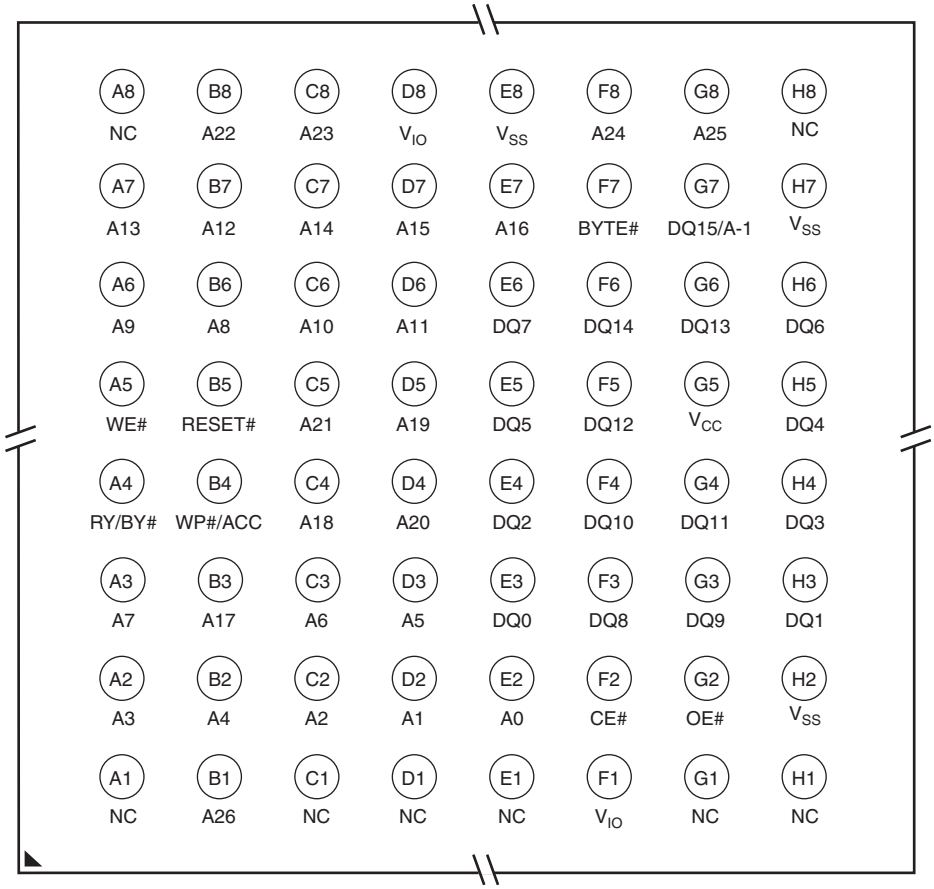
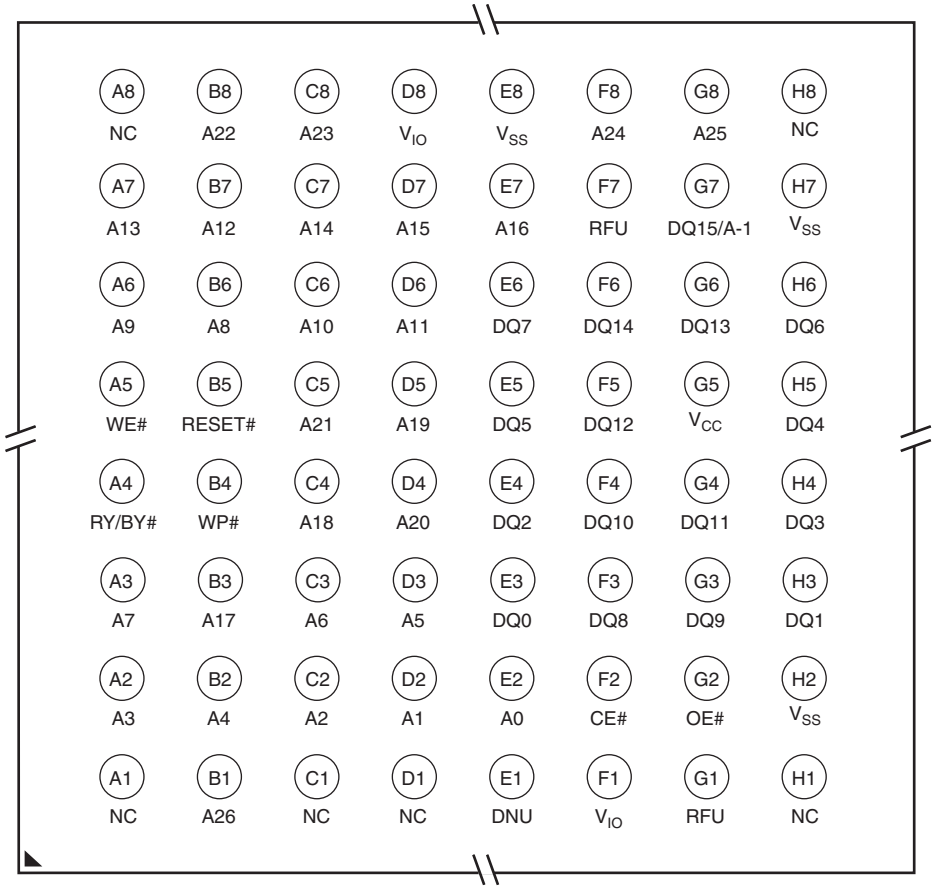


Figure 6. BGA Pin Definitions: GL-S



The WP#/ACC connection on GL-N and GL-P flash is the WP# input on GL-S flash. The ACC feature is not supported on GL-S and this input does not tolerate application of more than V_{IO} voltage. This input definition difference does not cause migration issues in most applications. Only applications that switch V_{HH} (nominally 12V) onto WP#/ACC to accelerate in-system program and erase operations are impacted by this difference.

The DQ15/A-1 connection on GL-N and GL-P flash is the DQ15 input/output on GL-S flash. Migration to GL-S is only possible in x16 Data Bus applications where the alternate A-1 input feature (on GL-N and GL-P flash) is not active. This input definition difference does not cause migration issues in x16 Data Bus applications.

The BYTE# input on GL-N and GL-P flash is electrically isolated and labeled RFU on GL-S flash. Migration to GL-S is only possible in x16 Data Bus applications where the BYTE# input is pulled > V_{IH}. Since this connection on the GL-S is electrically isolated, this input difference does not cause migration issues.

The GL-S flash has one connection labeled DNU that is labeled NC on GL-N and GL-P flash. DNU connections must be left floating. This input definition difference does not cause migration issues in most applications since No Connect pads are generally left floating. Applications that use this specific pad for signal routing purposes are impacted.

7 Appendix — Status Register Read Source Code Example

```
/*************************************************************
*
* wlld_StatusRegReadCmd - Status register read command
*
* This function sends the status register read command before
* actualy read it.
*
```

```
* RETURNS: void
*
* ERRNO:
*/
#ifndef REMOVE_LLD_STATUS_REG_READ_CMD
void wlld_StatusRegReadCmd
(
FLASHDATA * base_addr,    /* device base address in system */
ADDRESS offset           /* address offset from base address */
)
{
FLASH_WR(base_addr, (offset & SA_OFFSET_MASK) + LLD_UNLOCK_ADDR1, NOR_STATUS_REG_READ_CMD);
}

/*************************************************************
*
* lld_StatusGet - Determines Flash Status for GL-S device
*
* Note: This routine implements both (1) read status and check
* toggles (2) use read status command(GL-S device). The
* enable_status_cmd_g flag switch between these two status get
* methods. When calling this function, the WriteBufferProgramming
* flag needs to be set to 1 if the caller wants to check DQ1 for
* WriteBuffer abort. Then the flag needs to be set back to 0. See
* lld_poll for example of how to use the WriteBufferProgramming
* flag.
*
* RETURNS: DEVSTATUS
*
*/
#define DQ1_MASK   (0x02 * LLD_DEV_MULTIPLIER)  /* DQ1 mask for all interleave devices */
#define DQ2_MASK   (0x04 * LLD_DEV_MULTIPLIER)  /* DQ2 mask for all interleave devices */
#define DQ5_MASK   (0x20 * LLD_DEV_MULTIPLIER)  /* DQ5 mask for all interleave devices */
#define DQ6_MASK   (0x40 * LLD_DEV_MULTIPLIER)  /* DQ6 mask for all interleave devices */

#define DQ6_TGL_DQ1_MASK (dq6_toggles >> 5) /* Mask for DQ1 when device DQ6 toggling */
#define DQ6_TGL_DQ5_MASK (dq6_toggles >> 1) /* Mask for DQ5 when device DQ6 toggling */

DEVSTATUS lld_StatusGet
(
FLASHDATA *  base_addr,    /* device base address in system */
ADDRESS      offset        /* address offset from base address */
)
{

 FLASHDATA dq6_toggles;
 FLASHDATA status_read_1;
 FLASHDATA status_read_2;
 FLASHDATA status_read_3;


if( enable_status_cmd_g == 0 )   /* Do not use Read Status */
                /* Command */
 {
     status_read_1 = FLASH_RD(base_addr, offset);
     status_read_2 = FLASH_RD(base_addr, offset);
     status_read_3 = FLASH_RD(base_addr, offset);

     /* Any DQ6 toggles */
     dq6_toggles = ((status_read_1 ^ status_read_2) &        /* Toggles between read1 and read2 */
                    (status_read_2 ^ status_read_3) &        /* Toggles between read2 and read3 */
```

```
                    DQ6_MASK );                              /* Check for DQ6 only */

      if (dq6_toggles)
      {
/* Checking WriteBuffer Abort condition: Check for all devices */
/* that have DQ6 toggling also have Write Buffer Abort DQ1 set */
          if (WriteBufferProgramming &&
              ((DQ6_TGL_DQ1_MASK & status_read_1) == DQ6_TGL_DQ1_MASK) )
              return DEV_WRITE_BUFFER_ABORT;

/* Checking Timeout condition: Check for all devices that have */
/* DQ6 toggling also have Time Out DQ5 set. */
          if ((DQ6_TGL_DQ5_MASK & status_read_1) == DQ6_TGL_DQ5_MASK )
             return DEV_EXCEEDED_TIME_LIMITS;

          /* No timeout, no WB error */
          return DEV_BUSY;
      }
      else   /* no DQ6 toggles on all devices */
      {
          /* Checking Erase Suspend condition */
          status_read_1 = FLASH_RD(base_addr, offset);
          status_read_2 = FLASH_RD(base_addr, offset);

          /* Checking Erase Suspend condition */
             if ( ((status_read_1 ^ status_read_2) & DQ2_MASK) == 0)
             return DEV_NOT_BUSY;
/* All devices DQ2 not toggling */

          if (((status_read_1 ^ status_read_2) & DQ2_MASK) == DQ2_MASK)
             return DEV_SUSPEND;
/* All devices DQ2 toggling */
          else
             return DEV_BUSY;
/* Wait for all devices DQ2 toggling */
      }
  }
  else
  {
/*.........................................................*/
/* Use Status Register Read command to read the status      */
/* register. This is for GL-S devices only                  */
/*.........................................................*/
#ifdef STATUS_REG
      volatile FLASHDATA status_reg;

      wlld_StatusRegReadCmd( base_addr, offset );  /* Issue status register read command */
      status_reg = FLASH_RD(base_addr, offset);
/* read the status register */

      if ( (status_reg & DEV_RDY_MASK) != DEV_RDY_MASK  ) /* Are all devices done bit 7 is 1 */
        return DEV_BUSY ;

      if ( status_reg & DEV_ERASE_MASK )    /* Any erase error */
        return DEV_ERASE_ERROR;

      if ( status_reg & DEV_PROGRAM_MASK )/* Any program error */
        return DEV_PROGRAM_ERROR;

      if ( status_reg & DEV_SEC_LOCK_MASK )/* Any sector lock error */
        return DEV_SECTOR_LOCK;
```

```
      return DEV_NOT_BUSY ;
#endif
      return DEV_STATUS_UNKNOWN;
/* should never get here */
  }
}
```

Document History Page

Document Title: AN98586 - Migration from GL-N and GL-P to GL-S Flash Document Number: 001-98586				
Rev.	**ECN No.**	**Orig. of Change**	**Submission Date**	**Description of Change**
**	-	MSWI	03/28/2011	Initial release. Replaces the Application Note, How To Prepare Designs for 65 nm MirrorBit GL and 65 nm MirrorBit Eclipse GL Product Families.
*A	-	MSWI	05/20/2011	Added CFI Register section.
*B	-	MSWI	07/22/2011	Added text clarifications globally Updated table for GL-N, GL-P, and GL-S Flash Feature Comparison Added table: Secure Flash Device IDs. Added section: Lock Register Differences Updated table in DC and AC Parameter Differences Power On and Warm Reset Timing: Added tCEH specification Updated figures: Power-Up Reset Timing, Warm Reset Timing
*C	-	MSWI	08/05/2011	Secure Silicon OTP Area: Corrected preset and factory set conditions Lock Register Differences: Corrected factory lock bit and factory set conditions.
*D	-	MSWI	06/22/2012	GL Family Feature Comparison table: updated Packaging and Ordering Options GL Flash Device IDs table: corrected Device ID entries CFI Register Differences table: corrected technology definition Updated text in Updated text.
*E	-	MSWI	02/11/2013	Corrected Table 4.1: Power On and Warm Reset Timing Requirements
*F	-	MSWI	03/38/2013	Device ID: Corrected GL-S specific Device Technology CFI data
*G	-	MSWI	05/10/2013	GL Family Feature Comparison table: corrected Status via Status Register values for S29GL-N and S29GL-P
*H	4848347	MSWI	09/02/2015	Updated in Cypress template Added references to corresponding sections in Table 1 Edited throughout the document
*I	5844206	AESATMP8	08/04/2017	Updated logo and Copyright.
*J	6020425	MSWI	01/10/2018	Excluded GL064S from the scope of the AN. Updated template.

Worldwide Sales and Design Support

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

Products

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

PSoC®Solutions

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

Cypress Developer Community

Community | Projects | Video | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2011-2018. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

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

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