# THE DATASHEET OF

# S29GL512T12DHN020

www.win-source.net 0086-755-83957316

# 1 Gb (128 MB)/512 Mb (64 MB) GL-T MIRRORBIT™ Flash

## Parallel, 3.0 V

## General description

The S29GL01GT/512T are MIRRORBIT™ 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.

## Features

- 45-nm MIRRORBIT™ 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 $V_{CC}$
- ×8/×16 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-KB 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 for 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 for 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 |

Performance summary

### 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 MBps | 1.14 MBps | 1.14 MBps |
| Sector erase (128 KB) | 245 KBps | 245 KBps | 245 KBps |

### 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 |

# Table of contents

**General description** ..... **1**
**Features** ..... **1**
**Performance summary** ..... **2**
**Table of contents** ..... **4**
**1 Product overview** ..... **6**
**2 Address space overlays** ..... **8**
2.1 Flash memory array ..... 10
2.2 Device ID and CFI (ID-CFI) ASO ..... 11
2.3 Status Register ASO ..... 12
2.4 Data polling status ASO ..... 12
2.5 SSR ASO ..... 13
2.6 Sector protection control ..... 14
2.7 ECC status ASO ..... 15
**3 Data protection** ..... **16**
3.1 Device protection methods ..... 16
3.2 Command protection ..... 16
3.3 SSR (OTP) ..... 16
3.4 Sector protection methods ..... 17
**4 Read operations** ..... **22**
4.1 Asynchronous read ..... 22
4.2 Page mode read ..... 22
**5 Embedded operations** ..... **23**
5.1 Embedded algorithm controller (EAC) ..... 23
5.2 Program and erase summary ..... 24
5.3 Automatic ECC ..... 25
5.4 Command set ..... 27
5.5 Status monitoring ..... 43
5.6 Error types and clearing procedures ..... 50
5.7 Embedded Algorithm Performance table ..... 53
**6 Data integrity** ..... **57**
6.1 Erase endurance ..... 57
6.2 Data retention ..... 57
**7 Software interface reference** ..... **58**
7.1 Command summary ..... 58
7.2 Device ID and Common Flash Interface (ID-CFI) ASO map ..... 64
**8 Signal descriptions** ..... **69**
8.1 Address and data configuration ..... 69
8.2 Input/Output summary ..... 69
8.3 Word/Byte configuration ..... 70
8.4 Versatile I/O feature ..... 70
8.5 Ready/Busy# (RY/BY#) ..... 70
8.6 Hardware reset ..... 70
**9 Signal protocols** ..... **71**
9.1 Interface states ..... 71
9.2 Power-off with hardware data protection ..... 72
9.3 Power Conservation modes ..... 72
9.4 Read ..... 73
9.5 Write ..... 74
**10 Electrical specifications** ..... **75**
10.1 Absolute maximum ratings ..... 75
10.2 Thermal resistance ..... 75

Table of contents

10.3 Latchup characteristics ..... 75
10.4 Operating ranges ..... 76
10.5 DC characteristics ..... 79
10.6 Capacitance characteristics ..... 82
**11 Timing specifications ..... 84**
11.1 Key to switching waveforms ..... 84
11.2 AC test conditions ..... 85
11.3 Power-on reset (POR) and warm reset ..... 86
11.4 AC characteristics ..... 89
**12 Physical interface ..... 106**
12.1 56-pin TSOP ..... 106
12.2 64-ball FBGA ..... 108
12.3 56-ball FBGA ..... 111
**13 Special handling instructions for FBGA package ..... 113**
**14 Ordering information ..... 114**
14.1 Valid combinations — standard ..... 114
14.2 Valid combinations — automotive grade / AEC-Q100 ..... 117
**Revision history ..... 120**

# 1 Product overview

The GL-T family consists of 512-Mb to 1-Gb, 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.

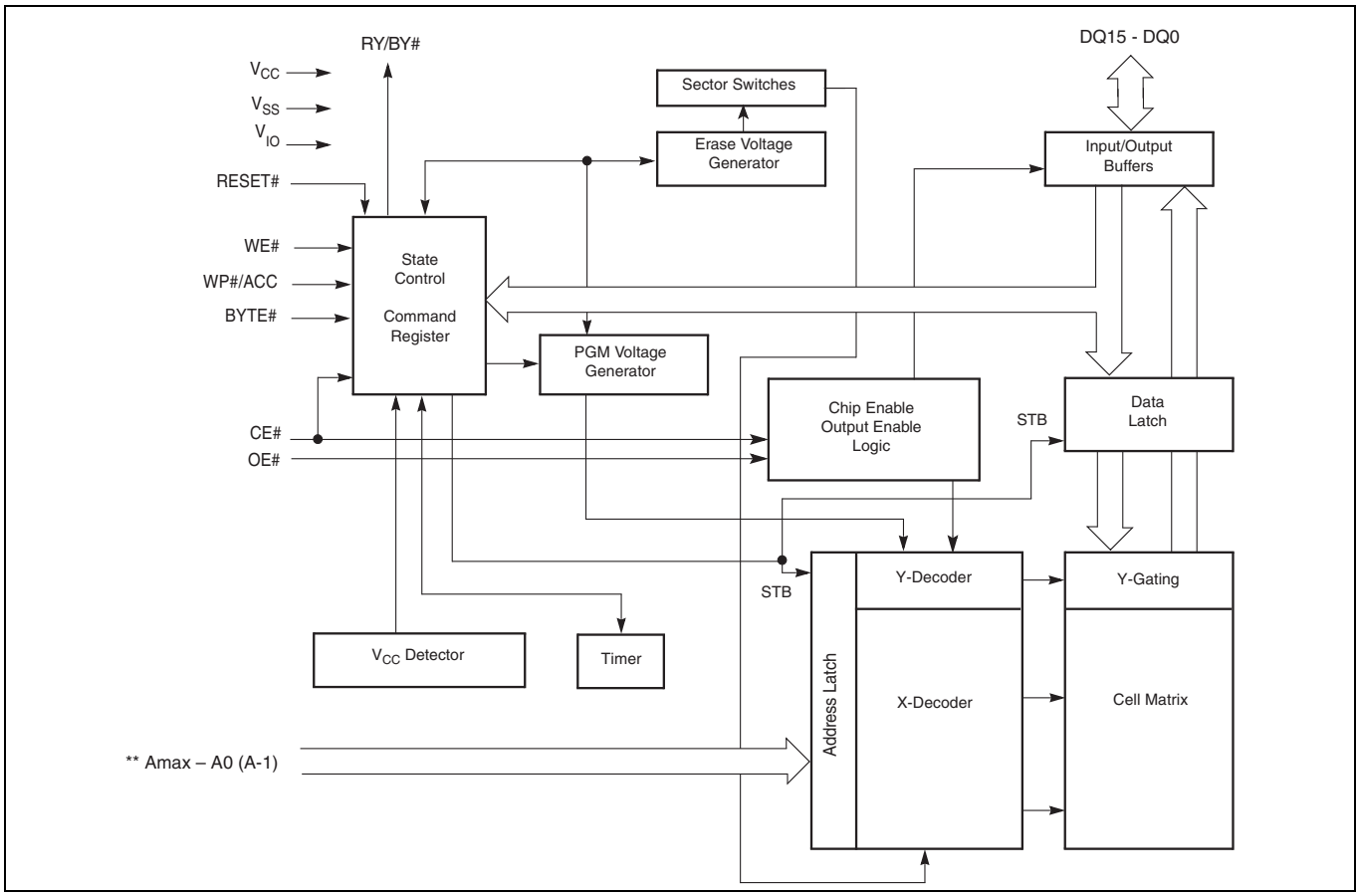



**Figure 1 Block diagram**[1]

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.

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

**Table 1 S29GL-T address map**

| Type | ×16 | | ×8 | |
|---|---|---|---|---|
| | Count | Addresses | Count | Addresses |
| Address within page | 16 | A3–A0 | 32 | A3–A1 |
| Address within write buffer | 256 | A7–A0 | 256 | 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)<br>512 (512 Mb) | Amax–A16 | 1024 (1 Gb)<br>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 ‘0’ to ‘1’. An erase operation must be performed on an entire 128-KB aligned and length group of data call a sector. When shipped from Infineon all sectors are erased.

Programming is done via a 512-byte write buffer. In ×16 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 ×8 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.

# 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 Infineon 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 Infineon 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

Address space overlays

In Read mode the entire flash memory array may be directly read by the host system memory controller. The memory device EAC, puts the device in Read mode during power-on, after a hardware reset, after a command reset, or after an 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 EA, 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 EA 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 EA is completed, the data polling ASO is exited and the device goes to the previous (calling) mode (from which the EA 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 23**, 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 S29GL01GT sector and memory address map**

| Sector size (KB) | 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 | 7FE0000h–7FFFFFFh | Sector ending address |

**Table 3 S29GL512T sector and memory address map**

| Sector size (KB) | 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 | 3FE0000h–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 ×16 and XXX0000h–XXX1FFFF in ×8.

## 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 CFI information, respectively.

**Table 4 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 25**.

### 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. Infineon 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. ×8, ×16, ×32. 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’. Infineon 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 25**.

### 2.2.2 Common flash memory interface

The JEDEC 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 64.

Like the device ID information, CFI information is structured to work with any memory data bus width e. g. ×8, ×16, ×32. 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, 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.

## 2.3 Status Register ASO

The Status Register ASO contains a single word of registered volatile status for EA's. 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 SSR ASO

The 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 SSR 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 5 SSR**

| 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 KB |

## 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. **"Automatic ECC"** on page 25 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 23** and **Table 24**, selects the ECC page.

**Table 6 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 7 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<br>1 = ECC disabled | 0 = No error corrected<br>1 = Single bit error corrected | 0 = No error corrected<br>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 cannot 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

EA's 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 23** for a listing of all commands accepted by the device.

## 3.3 SSR (OTP)

See **"SSR ASO"** on page 12 for a description of the secure silicon region. See **"Secure Silicon Region ASO"** on page 39 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 Advanced sector protection (ASP)

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 2**.

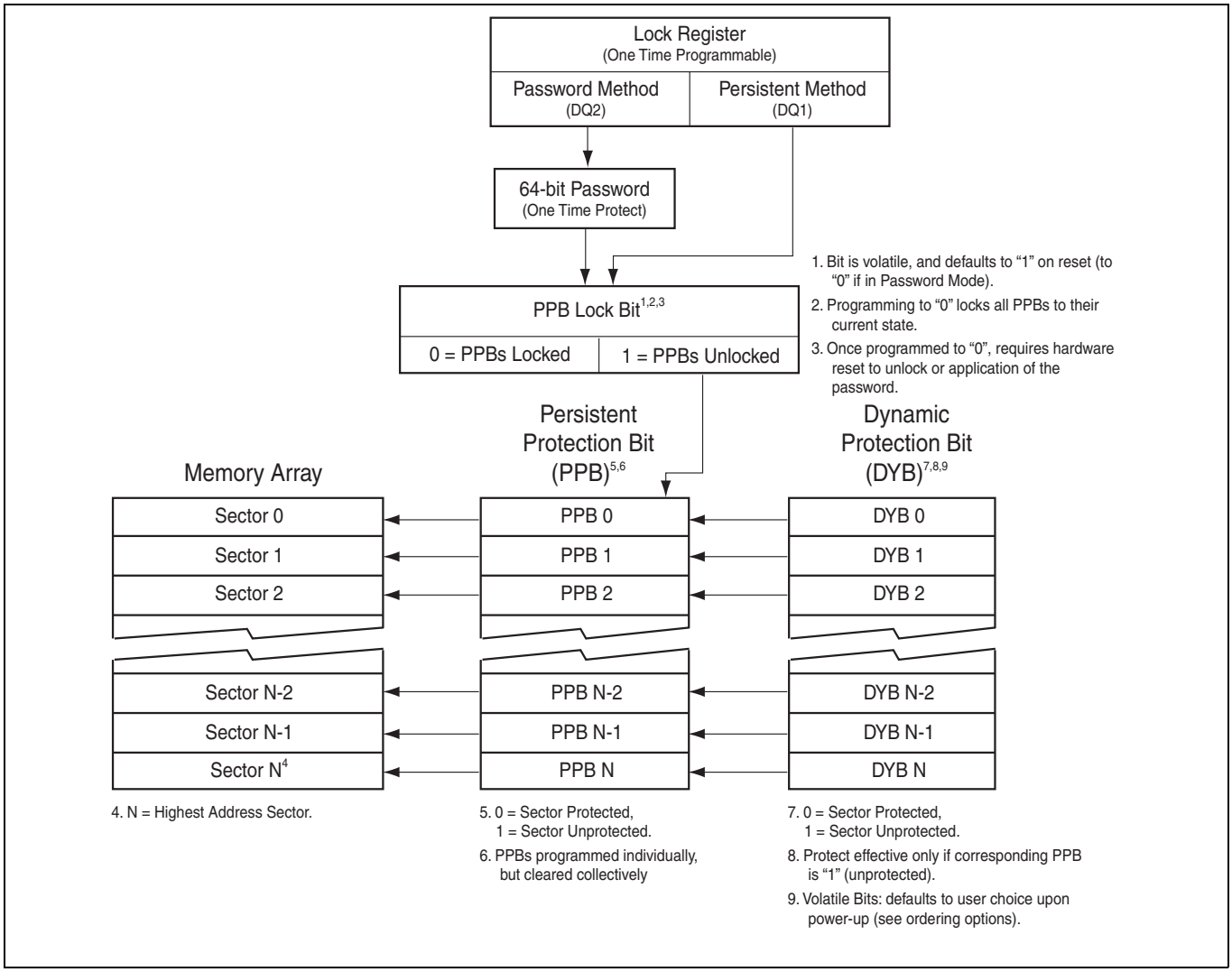



**Figure 2 ASP overview**

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 Infineon. The SSR can be factory protected or left unprotected depending on the ordering option (model) ordered.

### 3.4.3 PPB lock

The PPB 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 PPB's are located in a separate non-volatile 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. Pre-programming 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)

DYB's are volatile and unique for each sector and can be individually modified. DYB's 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 8 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 9 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 SSR protection bits must be used with caution, as once locked, there is no procedure available for unlocking the protected portion of the SSR and none of the bits in the protected SSR memory space can be modified in any way. Once the SSR 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 43 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 Infineon. 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–A1 in ×8 mode). The higher address bits are kept constant and only A3–A0 (A3–A1 in ×8 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 ×8 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 79 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).

EA's 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 58 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 EA. The algorithms are managed entirely by the device internal 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 '1'. A succeeding read shows that the data is still '0'. Only erase operations can convert '0' to '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 53.

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 53.
- When an 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 43 for information on these status bits. Refer to **"Data polling status"** on page 45 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 79 represents the active current specification for a write (EA) 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. Infineon 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 43 for information on these status bits. See **"Data polling status"** on page 45 for information on these status bits. See **Figure 3** 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 23** for program command sequences.

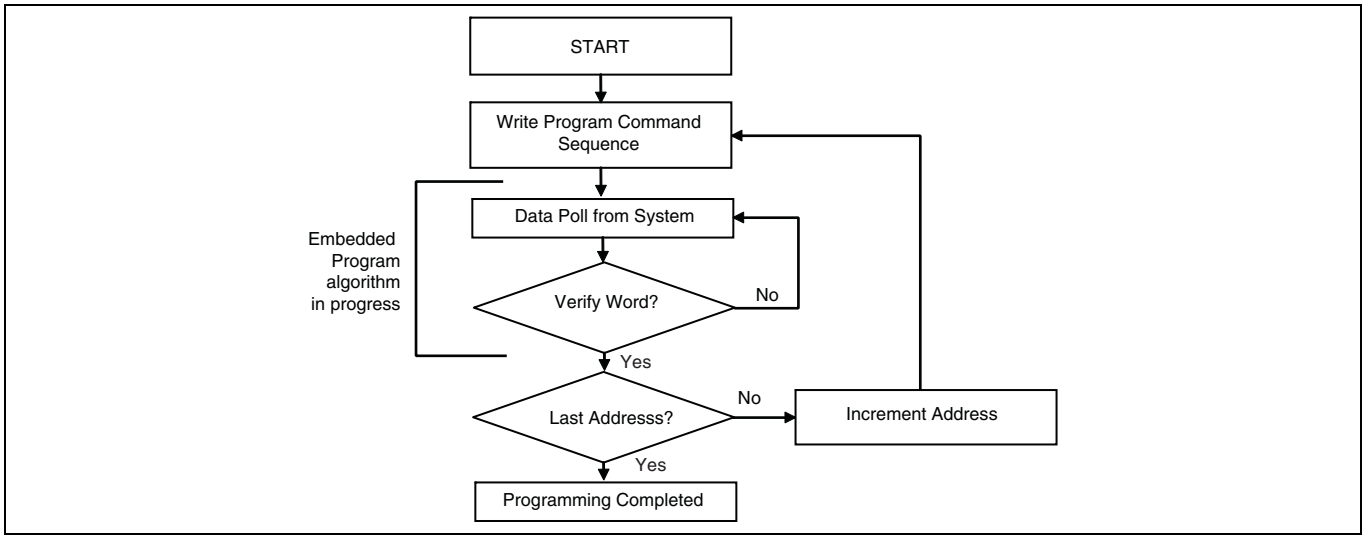



**Figure 3 Word program operation**

### 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 6 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 43 for information on these status bits. See **"Data polling status"** on page 45 for information on these status bits. See **Figure 4** 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.

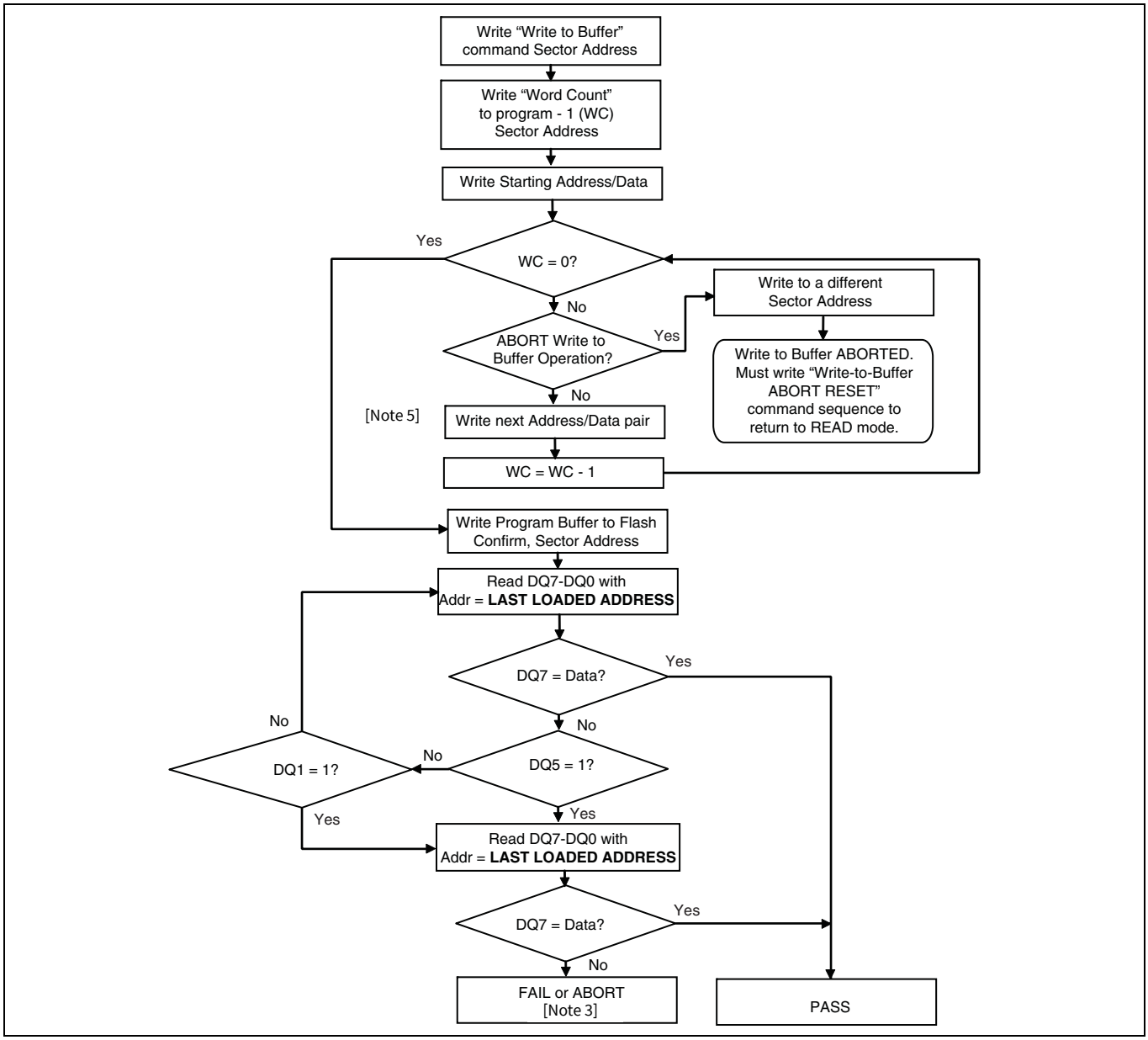



**Figure 4 Write buffer programming operation with data polling status**

**Notes**

2. DQ7 should be rechecked even if DQ5 = 1 because DQ7 may change simultaneously with DQ5.
3. 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.
4. See **Table 23** for the command sequence as required for write buffer programming.
5. 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.

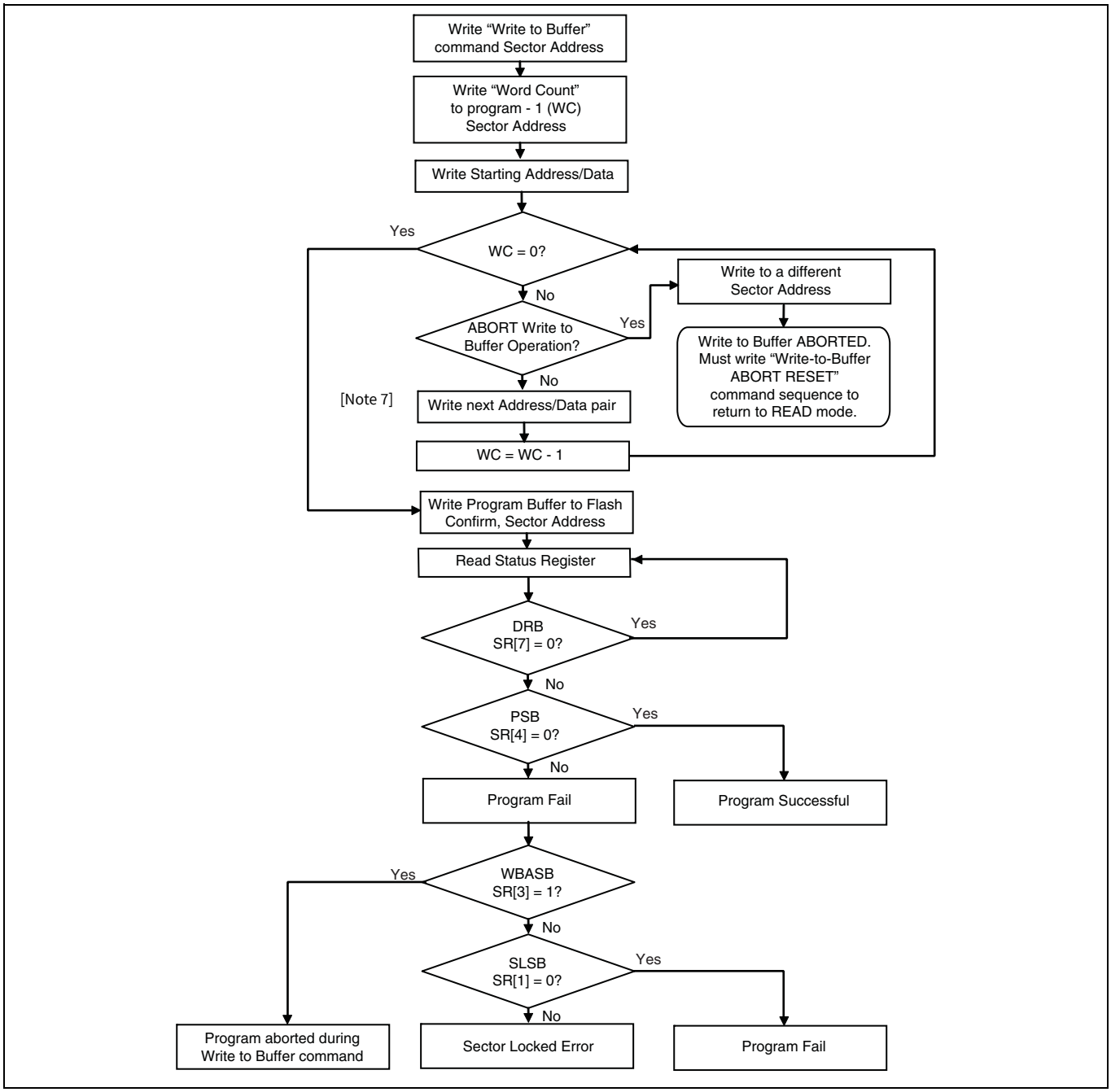



**Figure 5 Write buffer programming operation with Status Register**

**Notes**

6. See **Table 23** for the command sequence as required for write buffer programming.
7. 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.

**Table 10 Write Buffer Programming command sequence**

| Sequence | ×16 | | ×8 | | Comment |
|---|---|---|---|---|---|
| | Address | Data | Address | Data | |
| Issue Unlock command 1 | 555 | AA | AAA | AA | |
| Issue Unlock command 2 | 2AA | 55 | 555 | 55 | |
| Issue Write to Buffer command at sector address | SA | 0025h | SA | 25h | |
| Issue number of locations at sector address<br>Example:<br>WC of 0 = 1 word to pgm<br>WC of 1 = 2 words to pgm | SA | WC | SA | WC | WC = number of words to program – 1<br>(in ×8 mode WC = number of bytes to program – 1) |
| Load starting address / data pair | Starting address | PD | Starting address | PD | Selects write-buffer-page and loads first address/data pair. |
| Load next address / data pair | WBL | PD | WBL | PD | All addresses must be within the selected write-buffer-page boundaries, and have to be loaded in sequential order. |
| Load last address/data pair | WBL | PD | WBL | PD | All addresses must be within the selected write-buffer-page boundaries, and have to be loaded in sequential order. |
| Issue write buffer program confirm at sector address | SA | 0029h | SA | 29h | This command must follow the last write buffer location loaded, or the operation will abort. |
| Device goes busy. | | | | | |

**Legend:**

SA = Sector address (Non-sector address bits are don’t care. Any address within the sector is sufficient.)

WBL = Write buffer location (Must be within the boundaries of the write-buffer-line specified by the starting address.)

WC = Word count

PD = Program data

### 5.4.2 Program Suspend / Program Resume commands

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

There are two commands available for program suspend. The legacy combined Erase / Program suspend command (B0h command code) and the separate Program Suspend command (51h command code). There are also two commands for program resume. The legacy combined Erase / Program resume command (30h command code) and the separate Program Resume command (50h command code). It is recommended to use the separate program suspend and resume commands for programming and use the legacy combined command only for erase suspend and resume.

After the programming operation has been suspended, the system can read array data from any non-suspended Line. The Program Suspend command may also be issued during a programming operation while an erase is suspended. In this case, data may be read from any addresses not in Erase Suspend or Program Suspend.

After the Program Resume command is written, the device reverts to programming and the status bits are updated. The system can determine the status of the program operation by reading the Status Register or using data polling. Refer to **"Status Register"** on page 43 for information on these status bits. Refer to **"Data polling status"** on page 45 for more information.

Accesses and commands that are valid during Program Suspend are:

- Read to any other non-erase-suspended sector
- Read to any other non-program-suspended line
- Status Read command
- Status Register Clear
- Exit ASO or Command Set Exit
- Program Resume command

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

Program operations can be interrupted as often as necessary but in order for a program operation to progress to completion there must be some periods of time between resume and the next suspend command greater than or equal to $t_{PRS}$ in **"Embedded algorithm controller (EAC)"** on page 23.

Program suspend and resume is not supported while entered in an ASO.

### 5.4.3 Accelerated programming

The device supports program operations when the system asserts $V_{HH}$ on the WP#/ACC or ACC pin. When WP#/ACC or ACC pin is lowered back to $V_{IH}$ or $V_{IL}$ the device exits the accelerated programming mode and returns to normal operation. The WP#/ACC is $V_{HH}$ tolerant but is not designed to accelerate the program functions. If the system asserts $V_{HH}$ on this input, the device automatically enters the unlock bypass mode. The system can then use the Write Buffer Load command sequence provided by the unlock bypass mode. Note that if a 'Write-to-Buffer-Abort Reset' is required while in unlock bypass mode, the full 3-cycle RESET command sequence must be used to reset the device. Removing $V_{HH}$ from the ACC input, upon completion of the embedded program operation, returns the device to normal operation. Note that the WP#/ACC pin must not be at $V_{HH}$ for operations other than accelerated programming, or device damage may result. WP# contains an internal pull-up; when unconnected, WP# is at $V_{IH}$. Accelerated programming is supported at room temperature only.

- Sectors must be unlocked prior to raising WP#/ACC to $V_{HH}$.
- It is recommended that WP#/ACC apply $V_{HH}$ after power-up sequence is completed. In addition, it is recommended that WP#/ACC apply from $V_{HH}$ to $V_{IH}/V_{IL}$ before powering down $V_{CC}/V_{IO}$.

### 5.4.4 Unlock bypass

This device features an unlock bypass mode to facilitate shorter programming commands. Once the device enters the unlock bypass mode, only two write cycles are required to program data, instead of the normal four cycles. The device will also support the Write to Buffer command and will only require four+ write cycles.

This mode dispenses with the initial two unlock cycles required in the standard program command sequence, resulting in faster total programming time. The **"Command summary"** on page 58 shows the requirements for the unlock bypass command sequences.

During the unlock bypass mode, only the Read, Program, Write Buffer Programming, Write-to-Buffer-Abort Reset, Status Register Read, Status Register Clear, Soft Reset, Unlock Bypass Sector Erase, Unlock Bypass Chip Erase, Unlock Erase Suspend/Resume, Unlock Bypass Suspend/Resume, and Unlock Bypass Reset commands are valid. To exit the unlock bypass mode, the system must issue the two-cycle unlock bypass reset command sequence. The first cycle address is 'don't care' and the data 90h. The second cycle need only contain the data 00h. The sector then returns to the read mode.

**Software functions and sample code**

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

**Table 11 Unlock bypass entry (LLD Function = lld_UnlockBypassEntryCmd)**

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

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

**Table 12 Unlock bypass program (LLD Function = lld_UnlockBypassProgramCmd)**

| Cycle | Description | Operation | Byte address | Word address | Data |
|---|---|---|---|---|---|
| 1 | Program setup | Write | Base + XXXh | Base + XXXh | 00A0h |
| 2 | Program command | Write | Program address | Program address | Program data |

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

**Table 13 Unlock bypass reset (LLD Function = lld_UnlockBypassResetCmd)**

| Cycle | Description | Operation | Byte address | Word address | Data |
|---|---|---|---|---|---|
| 1 | Reset cycle 1 | Write | Base + XXXh | Base + XXXh | 0090h |
| 2 | Reset cycle 2 | Write | Program address | Program address | 0000h |

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

### 5.4.5 Evaluate Erase Status (EES)

The EES command verifies that the last erase operation on the addressed sector was completed successfully (i.e. “Trust Worthy”). The EES command can be used to detect erase operations failed due to loss of power, reset, or failure during the erase operation.

To initiate a EES on a sector, write 35h to address 555h in the sector, while the EAC is in the standby state.

The EES command may not be written while the device is actively programming or erasing or suspended.

The EES command does not allow for reads to the array during the operation.

Use the Status Register or polling method (only DQ6 toggles) to determine if the device is busy or completed. Once completed use the Status Register read to confirm if the sector is trust worthy or not. Bit 5 of the Status Register (SR[5]) will be cleared to ‘0’, if the sector is trust worthy. If the sector is not trust worthy than SR[5] will be set to ‘1’, RD/BY# will stay LOW, and either a Software Reset / ASO Exit command or a Status Register Clear command is required to return the device to the Standby State.

Once the EES is completed, the EAC will return to the Standby State.

The EES command requires $t_{EES}$ to complete and update the erase status in SR. The DRB bit (SR[7]) may be read to determine when the EES command is finished. If a sector is found not erased with SR[5] = 1, the sector must be erased again to ensure reliable storage of data in the sector.

### 5.4.6 Blank Check

The Blank Check command will confirm if the selected main flash array sector is currently erased (i.e. “Trust Worthy” and “Blank”). The Blank Check command does not allow for reads to the array during the Blank Check. Reads to the array while this command is executing will return polling data.

To initiate a Blank Check on a sector, write 33h to address 555h in the sector, while the EAC is in the Standby state.

The Blank Check command may not be written while the device is actively programming or erasing or suspended.

Use the Status Register or polling method (equivalent to an embedded erase operation) to determine if the device is busy or completed. Once completed the Status Register and the polling method will display if the sector is blank (equivalent to a successful erase operation) or if the sector is not erased. Bit 5 of the Status Register (SR[5]) will be cleared to ‘0’ if the sector is blank. If the sector is not blank than SR[5] will be set to ‘1’, RD/BY# will stay LOW, and either a Software Reset / ASO Exit command or a Status Register Clear command is required to return the device to the Standby State.

As soon as any bit is found to not be erased, the device will halt the operation and report the results.

Once the Blank Check is completed, the EAC will return to the Standby State.

### 5.4.7 Erase methods

#### 5.4.7.1 Chip erase

The chip erase function erases the entire main flash memory array. The device does not require the system to pre-program prior to erase. The embedded erase algorithm automatically programs and verifies the entire memory for an all 0 data pattern prior to electrical erase. After a successful chip erase, all locations within the device contain FFFFh. The system is not required to provide any controls or timings during these operations. The chip erase command sequence is initiated by writing two unlock cycles, followed by a set up command. Two additional unlock write cycles are then followed by the chip erase command, which in turn invokes the embedded erase algorithm. When WE# goes HIGH, at the end of the 6th cycle, the RY/BY# goes LOW.

When the embedded erase algorithm is complete, the EAC returns to the Standby state. Note that while the embedded erase operation is in progress, the system can not read data from the device. The system can determine the status of the erase operation by reading the RY/BY#, Status Register or using data polling. Refer to **“Ready/Busy# (RY/BY#)”** on page 70 for information on RY/BY#. Refer to **“Status Register”** on page 43 for information on these status bits. Refer to **“Data polling status”** on page 45 for more information.

Once the chip erase operation has begun, only a Status Read, Hardware RESET or Power cycle are valid. All other commands are ignored. However, a Hardware Reset or Power Cycle immediately terminates the erase operation and returns to read mode after $t_{RPH}$ time. If a chip erase operation is terminated, the Chip Erase command sequence must be reinitiated once the device has returned to the idle state to ensure data integrity.

See **Table 18**, **“Asynchronous Write operations”** on page 96 and **“Alternate CE# Controlled Write operations”** on page 104 for parameters and timing diagrams.

Sectors protected by the ASP DYB and PPB bits will not be erased. See **“Advanced sector protection (ASP)”** on page 17. If a sector is protected during chip erase, chip erase will skip the protected sector and continue with next sector erase. The status register erase status bit and sector lock bit are not set to ‘1’ by a failed erase on a protected sector.

### 5.4.7.2 Sector erase

The sector erase function erases one sector in the memory array. The device does not require the system to pre-program prior to erase. The embedded erase algorithm automatically programs and verifies the entire sector for an all 0 data pattern prior to electrical erase. After a successful sector erase, all locations within the erased sector contain FFFFh. The system is not required to provide any controls or timings during these operations. The sector erase command sequence is initiated by writing two unlock cycles, followed by a set up command. Two additional unlock write cycles are then followed by the address of the sector to be erased, and the sector erase command. When WE# goes HIGH, at the end of the 6th cycle, the RY/BY# goes LOW.

After the command sequence is written, a sector erase time-out of $t_{SEA}$ occurs. During the time-out period, additional sector addresses and sector erase commands may be written. Invalid commands will be ignored during the time-out period. Loading the sector erase buffer may be done in any sequence, and the number of sectors may be from one sector to all sectors. The time between these additional cycles must be less than $t_{SEA}$, otherwise erasure may begin. Any sector erase address and command following the exceeded time-out may or may not be accepted. It is recommended that processor interrupts be disabled during this time to ensure all commands are accepted. The interrupts can be re-enabled after the last Sector Erase command is written. Note that the secured silicon sector, autoselect, and CFI functions are unavailable when an erase operation in is progress. The system must rewrite the command sequence and any additional addresses and commands.

The system can determine the status of the erase operation by reading the RY/BY#, Status Register or using data polling. Refer to **“Ready/Busy# (RY/BY#)”** on page 70 for information on RY/BY#. Refer to **“Status Register”** on page 43 for information on these status bits. Refer to **“Data polling status”** on page 45 for more information.

Once the sector erase operation has begun, the Status Register Read and Erase Suspend commands are valid. All other commands are ignored. However, note that a hardware reset immediately terminates the erase operation and returns to read mode after $t_{RPH}$ time. If a sector erase operation is terminated, the sector erase command sequence must be reinitiated once the device has reset operation to ensure data integrity.

Sector(s) protected by the ASP DYB and PPB bits or password protection will not be erased. See **“Advanced sector protection (ASP)”** on page 17. If a sector is protected during multi-sector erase, sector erase will skip the protected sector and continue with next sector erase. The status register erase status bit and sector lock bit are not set to ‘1’ by a failed erase on a protected sector. See **“Embedded algorithm controller (EAC)”** on page 23 for parameters and timing diagrams. Sectors protected by the ASP DYB and PPB bits will not be erased. See **“Advanced sector protection (ASP)”** on page 17.

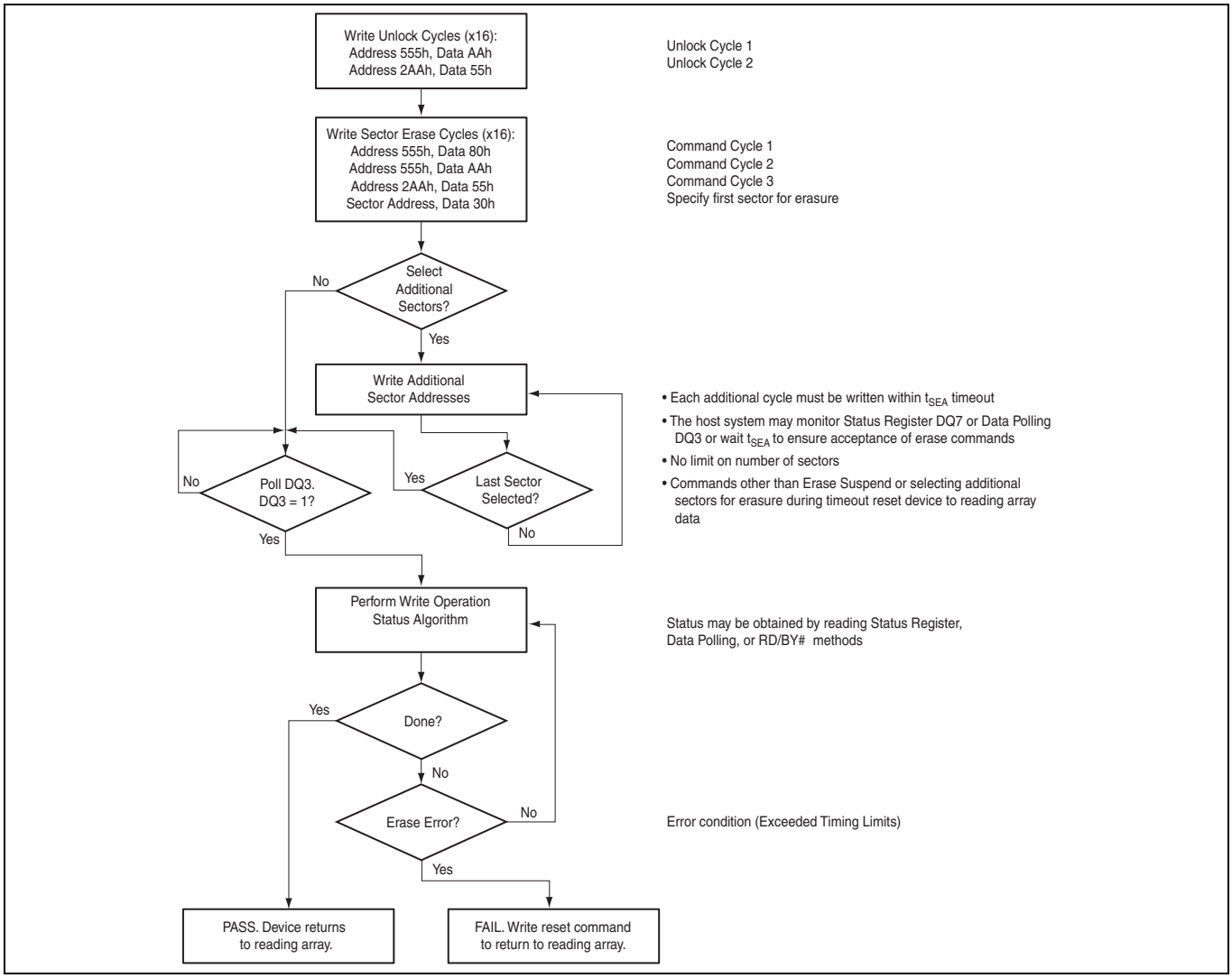



**Figure 6 Sector erase operation**[8]

**Note**
8. See command summary for ×8 bus cycles.

### 5.4.8 Erase Suspend / Erase Resume

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

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

After the erase operation has been suspended, the part enters the erase-suspend mode. The system can read data from or program data to the main flash array. Reading at any address within erase-suspended sectors produces undetermined data. The system can determine if a sector is actively erasing or is erase-suspended by reading the Status Register or using data polling. Refer to **"Status Register"** on page 43 for information on these status bits. Refer to **"Data polling status"** on page 45 for more information.

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

If a program failure occurs during erase suspend the Status Register Clear or Soft Reset commands will return the device to the erase suspended state. Erase will need to be resumed and completed before again trying to program the memory array.

Accesses and commands that are valid during Erase Suspend are:

- Read to any other non-suspended sector
- Program to any other non-suspended sector
- Status Register Read
- Status Register Clear
- Erase Resume command

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

Erase suspend and resume is not supported while entered in an ASO.

### 5.4.9 ASO Entry and Exit

#### 5.4.9.1 ID-CFI ASO

The system can access the ID-CFI ASO by issuing the ID-CFI Entry command sequence during read mode. See the detail description **Table 25**.

The ID-CFI ASO allows the following activities:

- Read ID-CFI ASO, using the same SA as used in the entry command.
- Read Sector Protection State at Sector Address (SA) + 2h. Location 2h provides volatile information on the current state of sector protection for the sector addressed. Bit 0 of the word at location 2h shows the logical NAND of the PPB and DYB bits related to the addressed sector such that if the sector is protected by either the PPB = 0 or the DYB = 0 bit for that sector the state shown is protected. (1 = Sector protected, 0 = Sector unprotected.)
- ASO Exit.

The following is a C source code example of using the CFI Entry and Exit functions. Refer to the **Infineon Low Level Driver User's Guide** for general information on Infineon flash memory software development guidelines.

```
/* Example: CFI Entry command */
*( (UINT16 *)base_addr + 0x55 ) = 0x0098; /* write CFI entry command */

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

#### 5.4.9.2 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.

#### 5.4.9.3 Secure Silicon Region ASO

The system can access the Secure Silicon Region by issuing the Secure Silicon Region Entry command sequence during Read Mode. This entry command uses the sector address (SA) in the command to determine which sector will be overlaid.

The Secure Silicon Region ASO allows the following activities:

- Read Secure Silicon Regions.
- Program the customer Secure Silicon Region is allowed using the Word or Write Buffer Programming commands. The Unlock Bypass commands and using ACC is not allowed.
- ASO Exit using legacy Secure Silicon Exit command for backward software compatibility.
- ASO Exit using the common exit command for all ASO - alternative for a consistent exit method.
- The recommended procedure for using the SSR region 3 read password mode is as follows:
- Program the data you want in SSR region 3.
- Clear lock register bit 10 to '0', which disable further program operations.
- Program the SSR region 3 password.
- Clear lock register bit 11 to '0', which will enable the SSR region 3 password feature which requires that a password be applied before reading SSR region 3 is allowed.

### 5.4.9.4 Lock Register ASO

The system can access the lock register by issuing the Lock Register entry command sequence during read mode. This entry command does not use a sector address from the entry command. The lock register appears at word location 0 in the device address space. All other locations in the device address space are undefined.

The lock register ASO allows the following activities:

- Read lock register, using device address location 0.
- Program the customer lock register using a modified Word Programming command.
- ASO Exit using legacy Command Set Exit command for backward software compatibility.
- ASO Exit using the common exit command for all ASO — alternative for a consistent exit method.

### 5.4.9.5 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 for a specific page or if the ECC logic corrected a single bit error on the selected Page.

The ECC Status ASO allows the following activities:

- Read ECC Status for the selected page.

### 5.4.9.6 Password ASO

The system can access the Password ASO by issuing the Password entry command sequence during read mode. This entry command does not use a sector address from the entry command. The password appears at word locations 0 to 3 in the device address space. All other locations in the device address space are undefined.

The Password ASO allows the following activities:

- Read password, using device address location 0 to 3 (if not locked).
- Program the password using a modified Word Programming command.
- Unlock the PPB lock bit with the Password Unlock command.
- ASO Exit using legacy Command Set Exit command for backward software compatibility.
- ASO Exit using the common exit command for all ASO — alternative for a consistent exit method.

### 5.4.9.7 PPB ASO

The system can access the PPB ASO by issuing the PPB entry command sequence during read mode. This entry command does not use a sector address from the entry command. The PPB bit for a sector appears in bit 0 of all word locations in the sector.

The PPB ASO allows the following activities:

- Read PPB protection status of a sector in bit 0 of any word in the sector.
- Program the PPB bit using a modified Word Programming command.
- Erase all PPB bits with the PPB erase command.
- ASO Exit using legacy Command Set Exit command for backward software compatibility.
- ASO Exit using the common exit command for all ASO — alternative for a consistent exit method.

### 5.4.9.8 PPB Lock ASO

The system can access the PPB Lock ASO by issuing the PPB Lock entry command sequence during Read Mode. This entry command does not use a sector address from the entry command. The global PPB Lock bit appears in bit 0 of all word locations in the device.

The PPB Lock ASO allows the following activities:

- Read PPB lock protection status in bit 0 of any word in the device address space.
- Set the PPB lock bit using a modified Word Programming command.
- ASO Exit using legacy Command Set Exit command for backward software compatibility.
- ASO Exit using the common exit command for all ASO — alternative for a consistent exit method.

### 5.4.9.9 DYB ASO

The system can access the DYB ASO by issuing the DYB entry command sequence during read mode. This entry command does not use a sector address from the entry command. The DYB bit for a sector appears in bit 0 of all word locations in the sector.

The DYB ASO allows the following activities:

- Read DYB protection status of a sector in bit 0 of any word in the sector.
- Set the DYB bit using a modified Word Programming command.
- Clear the DYB bit using a modified Word Programming command.
- ASO Exit using legacy Command Set Exit command for backward software compatibility.
- ASO Exit using the common exit command for all ASO — alternative for a consistent exit method.

### 5.4.9.10 Software (command) Reset / ASO exit

Software reset is part of the command set (See **Table 23**) that also returns the EAC to standby state and must be used for the following conditions:

- Exit ASO modes
- Clear timeout bit (DQ5) for data polling when timeout occurs

Software Reset does not affect EA mode. Reset commands are ignored once programming or erasure has begun, until the operation is complete. Software Reset does not affect outputs; it serves primarily to return to read mode from an ASO mode or from a failed program or erase operation.

Software Reset may cause a return to read mode from undefined states that might result from invalid command sequences. However, a hardware reset may be required to return to normal operation from some undefined states.

There is no software reset latency requirement. The reset command is executed during the $t_{WPH}$ period.

### 5.4.9.11 Continuity check feature

The continuity check provides a basic test of connectivity from package connectors to each die pad and to each individual die in a DDP. This feature is an extension of the legacy unlock cycle sequence used at the beginning of several commands. The unlock sequence is two writes with alternating ones and zeros pattern on the lower portion of the address and data lines with the pattern inverted between the first and second write. To perform a continuity check these patterns are extended to cover all address (Amax to '0') and data lines (DQ15 to '0'). A logic comparison circuit looks for the alternating one and zero pattern that is inverted between the two write cycles.

In the case of a DDP the A26 input is used to select which die the writes are sent to. When the correct patterns are detected the status register bit 0 is set to '1'. The status register clear command will clear the status register bit 0 to '0'.

The following table describes the continuity check sequence for a single die (e.g. GL01GT) in ×16.

**Table 14 Continuity check sequence for a single die**

| Phase | Access type | Address A26 | Address A25 to A0 | Data | Comment |
|---|---|---|---|---|---|
| Set-up | Write | N/A | XXXX555 | XX71 | Clear die zero status |
| | Write | N/A | 555 | XX70 | Write Status Register Read command to die zero |
| | Read | N/A | x | RD | Read status from die zero to confirm status bit zero = 0 |
| Continuity pattern | Write | N/A | 2AAAA55 | FF00 | First continuity cycle |
| | Write | N/A | 15555AA | 00FF | Second continuity cycle |
| Verify continuity pattern detected | Write | N/A | 555 | XX70 | Write Status Register Read command to die zero |
| | Read | N/A | x | RD | Read status from die zero to confirm status bit zero = 1 for continuity pattern detected |

The following table describes the continuity check sequence for a single die (e.g. GL01GT)in ×8.

**Table 15 Continuity check sequence for a single die**

| Phase | Access type | Address A26 | Address A25 to A1 | Data | Comment |
|---|---|---|---|---|---|
| Set-up | Write | N/A | XXXX555 | 71 | Clear die zero status |
| | Write | N/A | AAA | 70 | Write Status Register Read command to die zero |
| | Read | N/A | x | RD | Read status from die zero to confirm status bit zero = 0 |
| Continuity pattern | Write | N/A | 55554AB | FF | First continuity cycle |
| | Write | N/A | 2AAAB54 | 00 | Second continuity cycle |
| Verify continuity pattern detected | Write | N/A | 555 | 70 | Write Status Register Read command to die zero |
| | Read | N/A | x | RD | Read status from die zero to confirm status bit zero = 1 for continuity pattern detected |

## 5.5 Status monitoring

There are three methods for monitoring EA status. Previous generations of the S29GL flash family used the methods called data polling and Ready/Busy# (RY/BY#) Signal. These methods are still supported by the S29GL-T family. One additional method is reading the Status Register.

### 5.5.1 Status Register

The status of program and erase operations is provided by a single 16-bit Status Register. The Status Register Read command is written followed by a read access of the Status Register information. 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 contents of the Status Register is aliased (overlaid) in the full memory address space. Valid read (CE# and OE# LOW) access in the Status Register ASO exits the ASO (with the rising edge of CE# or OE# for $t_{CEPH}$/$t_{OEPH}$ time) and returns to the address space map in use when the Status Register Read command was issued. While in ×8 mode the full Status Register can be read (both the upper byte and lower byte) with one Status Register entry by keeping CE# and OE# LOW and having a transition on A1. Write operations are ignored and the device will stay in Status Register ASO.The Status Register contains bits related to the results – success or failure – of the most recently completed embedded algorithms (EA):

- Erase status (bit 5),
- Program status (bit 4),
- Write buffer abort (bit 3),
- Sector locked status (bit 1),
- Continuity Check Pattern Detected (bit 0).

and, bits related to the current state of any in process EA:

- Device busy (bit 7),
- Erase suspended (bit 6),
- Program suspended (bit 2)

The current state bits indicate whether an EA is in process, suspended, or completed.

The upper 8 bits (bits 15:8) are reserved. These have undefined HIGH or LOW value that can change from one status read to another. These bits should be treated as don’t care and ignored by any software reading status.

The Soft Reset Command will clear to 0 bits [5, 4, 1, 0] of the Status Register if Status Register bit 3 = 0. It will not affect the current state bits.

The Clear Status Register Command will clear to 0 bits [5, 4, 3, 1, 0] of the status register but will not affect the current state bits.

**Table 16 Status Register**

| Bit # | 15:8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit description | Reserved | Device ready bit | Erase suspend status bit[9] | Erase status bit[10] | Program status bit[11, 12] | Write buffer abort status bit | Program suspend status bit[13] | Sector lock status bit[14, 15] | Continuity check |
| Bit name | Note 18 | DRB | ESSB | ESB | PSB | WBASB | PSSB | SLSB | CC |
| Reset status[16, 17] | X | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Busy status | Invalid | 0 | Invalid | Invalid | Invalid | Invalid | Invalid | Invalid | Invalid |
| Ready status | X | 1[19, 20, 21] | 0 = No erase in suspension[22]<br>1 = Erase in suspension | 0 = Erase successful<br>1 = Erase fail | 0 = Program successful<br>1 = Program fail | 0 = Program not aborted<br>1 = Program aborted during Write to Buffer command | 0 = No program in suspension<br>1 = Program in suspension | 0 = Sector not locked during operation<br>1 = Sector locked error | 0 = Continuity check pattern not detected<br>1 = Continuity check pattern detected |

**Notes**

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

### 5.5.2 Data polling status

During an active embedded algorithm the EAC switches to the data polling ASO to display EA status to any read access. A single word of status information is aliased in all locations of the device address space. In the status word there are several bits to determine the status of an EA. These are referred to as DQ bits as they appear on the data bus during a read access while an EA is in progress. DQ bits 15 to 8, DQ4, and DQ0 are reserved and provide undefined data. Status monitoring software must mask the reserved bits and treat them as don't care. In ×8 mode A1 is ignored when performing Data Polling. **Table 17** and the following subsections describe the functions of the remaining bits.

#### 5.5.2.1 DQ7: Data# polling

The Data# polling bit, DQ7, indicates to the host system whether an embedded algorithm is in progress or has completed. Data# polling is valid after the rising edge of the final WE# pulse in the program or erase command sequence. Note that the Data# polling is valid only for the last word being programmed in the write-buffer-page during write buffer programming. Reading Data# polling status on any word other than the last word to be programmed in the write-buffer-page will return false status information.

During the embedded program algorithm, the device outputs on DQ7 the complement of the data bit programmed to DQ7. This DQ7 status also applies to programming during erase suspend. When the embedded program algorithm is complete, the device outputs the data bit programmed to bit 7 of the last word programmed. In case of a program suspend, the device allows only reading array data. If a program address falls within a protected sector, Data# polling on DQ7 is active for $t_{DP}$, then the device returns to reading array data.

During the Embedded Erase, Evaluate Erase Status, or Blank Check algorithms, Data# polling produces '0' on DQ7. When the algorithm is complete, or if the device enters the erase suspend mode, Data# polling produces '1' on DQ7. This is analogous to the complement / true datum output described for the embedded program algorithm: the erase function changes all the bits in a sector to '1'; prior to this, the device outputs the complement or '0'. The system must provide an address within the sector selected for erasure to read valid status information on DQ7.

After an erase command sequence is written, if the sector selected for erasing is protected, Data# polling on DQ7 is active for $t_{DP}$, then the device returns to reading array data.

Just prior to the completion of an embedded program or erase operation, DQ7 may change asynchronously with DQ6–DQ0 while Output Enable (OE#) is asserted LOW. That is, the device may change from providing status information to valid data on DQ7. Depending on when the system samples the DQ7 output, it may read the status or valid data. Even if the device has completed the program or erase operation and DQ7 has valid data, the data outputs on DQ6–DQ0 may be still invalid. Valid data on DQ7–D00 appears on successive read cycles.

When the system detects DQ7 has changed from the complement to true data, it can read valid data at DQ15–DQ0 (DQ7–DQ0 in ×8 mode) on the following read cycles. This is because DQ7 may change asynchronously with DQ6–DQ0 while Output Enable (OE#) is asserted LOW. This is illustrated in **Figure 29**. **Figure 17** shows the outputs for Data# polling on DQ7. **Figure 4** shows the Data# polling algorithm use in Write Buffer Programming.

Valid DQ7 data polling status may only be read from:

- the address of the last word loaded into the write buffer for a write buffer programming operation;
- the location of a single word programming operation;
- a location in a sector being erased, or evaluate erase status, or blank checked;
- or a location in any sector during chip erase.

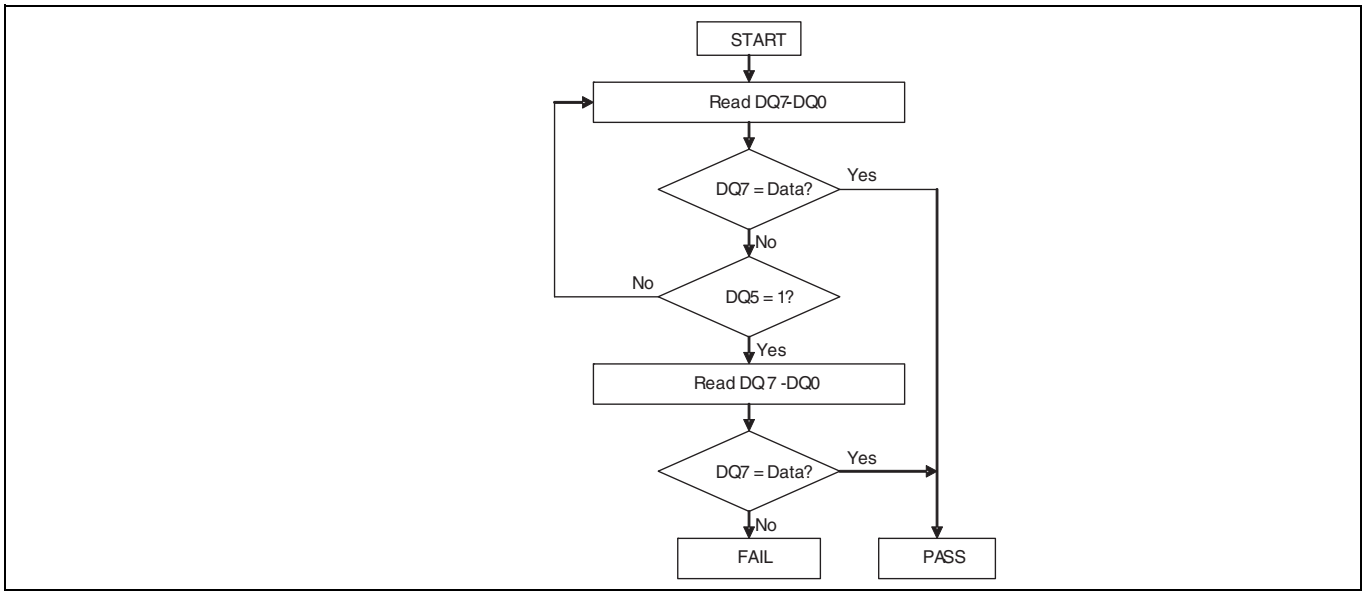



**Figure 7 Data# polling algorithm**[23]

**Note**
23.DQ7 should be rechecked even if DQ5 = 1 because DQ7 may change simultaneously with DQ5.

### 5.5.2.2 DQ6: Toggle Bit I

Toggle Bit I on DQ6 indicates whether an embedded program or erase algorithm is in progress or complete, or whether the device has entered the program suspend or erase suspend mode. Toggle Bit I may be read at any address, and is valid after the rising edge of the final WE# pulse in the command sequence (prior to the program or erase operation).

During an embedded program or erase algorithm operation, successive read cycles to any address cause DQ6 to toggle (The system may use either OE# or CE# to control the read cycles). When the operation is complete, DQ6 stops toggling.

After an erase command sequence is written, if the sector selected for erasing is protected, DQ6 toggles for $t_{DP}$, then the EAC returns to standby (Read Mode). If the selected sector is not protected, the embedded erase algorithm erases the unprotected sector.

The system can use DQ6 and DQ2 together to determine whether a sector is actively erasing or erase-suspended. When the device is actively erasing (that is, the embedded erase algorithm is in progress), DQ6 toggles. When the device enters the program suspend mode or erase suspend mode, DQ6 stops toggling. However, the system must also use DQ2 to determine which sectors are erasing, or erase-suspended. Alternatively, the system can use DQ7 (see **"DQ7: Data# polling"** on page 45).

DQ6 also toggles during the erase-suspend-program mode, and stops toggling once the embedded program algorithm is complete.

**Table 17** shows the outputs for Toggle Bit I on DQ6. **Figure 8** shows the toggle bit algorithm in flowchart form, and the **"Reading Toggle Bits DQ6/DQ2"** on page 48 explains the algorithm. **Figure 8** shows the toggle bit timing diagrams. See also **"DQ2: Toggle Bit II"** on page 47.

### 5.5.2.3 DQ3: Sector Erase Timer

After writing a sector erase command sequence, the system may read DQ3 to determine whether or not erasure has begun. See **"Sector erase"** on page 36 for more details (The sector erase timer does not apply to the chip erase command). If additional sectors are selected for erasure, the entire time-out also applies after each additional sector erase command. When the time-out period is complete, DQ3 switches from '0' to '1'. If the time between additional sector erase commands from the system can be assumed to be less than $t_{SEA}$, then the system need not monitor DQ3.

After the sector erase command is written, the system should read the status of DQ7 (Data# polling) or DQ6 (Toggle Bit I) to ensure that the device has accepted the command sequence, and then read DQ3. If DQ3 is '1', the embedded erase algorithm has begun; all further commands (except Erase Suspend) are ignored until the erase operation is complete. If DQ3 is '0', the device accepts additional sector erase commands. To ensure the command has been accepted, the system software should check the status of DQ3 prior to and following each sub-sequent sector erase command. If DQ3 is high on the second status check, the last command might not have been accepted. **Table 17** shows the status of DQ3 relative to the other status bits.

### 5.5.2.4 DQ2: Toggle Bit II

Toggle Bit II on DQ2, when used with DQ6, indicates whether a particular sector is actively erasing (that is, the embedded erase algorithm is in progress), or whether that sector is erase-suspended. Toggle Bit II is valid after the rising edge of the final WE# pulse in the command sequence.

DQ2 toggles when the system reads at addresses within the sector selected for erasure (or all sectors selected for erase operation during multi-sector erase). (The system may use either OE# or CE# to control the read cycles). But DQ2 cannot distinguish whether the sector is actively erasing or is erase-suspended. DQ6, by comparison, indicates whether the device is actively erasing, or is in erase suspend, but cannot distinguish if the sector is selected for erasure. Thus, both status bits are required for sector and mode information. Refer to **Table 17** to compare outputs for DQ2 and DQ6. **Figure 7** shows the toggle bit algorithm in flowchart form, and the **"Reading Toggle Bits DQ6/DQ2"** on page 48 explains the algorithm. See also **Figure 8** shows the toggle bit timing diagram.

### 5.5.2.5 Reading Toggle Bits DQ6/DQ2

Refer to **Figure 7** for the following discussion. Whenever the system initially begins reading toggle bit status, it must read DQ7–DQ0 at least twice in a row to determine whether a toggle bit is toggling. Typically, the system would note and store the value of the toggle bit after the first read. After the second read, the system would compare the new value of the toggle bit with the previous value. If the toggle bit is not toggling, the device has completed the program or erases operation. The system can read array data on DQ15–DQ0 (DQ7–DQ0 in ×8 mode) on the following read cycle.

However, if after the initial two read cycles, the system determines that the toggle bit is still toggling, the system also should note whether the value of DQ5 is HIGH (see **"DQ5: Exceeded timing limits"** on page 49). If it is, the system should then determine again whether the toggle bit is toggling, since the toggle bit may have stopped toggling just as DQ5 went HIGH. If the toggle bit is no longer toggling, the device has successfully completed the program or erase operation. If it is still toggling, the device did not complete the operation successfully, and the system must write the reset command to return to reading array data. It is recommended that data read for polling purposes only be used for polling purposes. Once toggling has stopped array data will be available on subsequent reads.

The remaining scenario is that the system initially determines that the toggle bit is toggling and DQ5 has not gone HIGH. The system may continue to monitor the toggle bit and DQ5 through successive read cycles, determining the status as described in the previous paragraph. Alternatively, it may choose to perform other system tasks. In this case, the system must start at the beginning of the algorithm when it returns to determine the status of the operation (operation as shown in **Figure 8**).

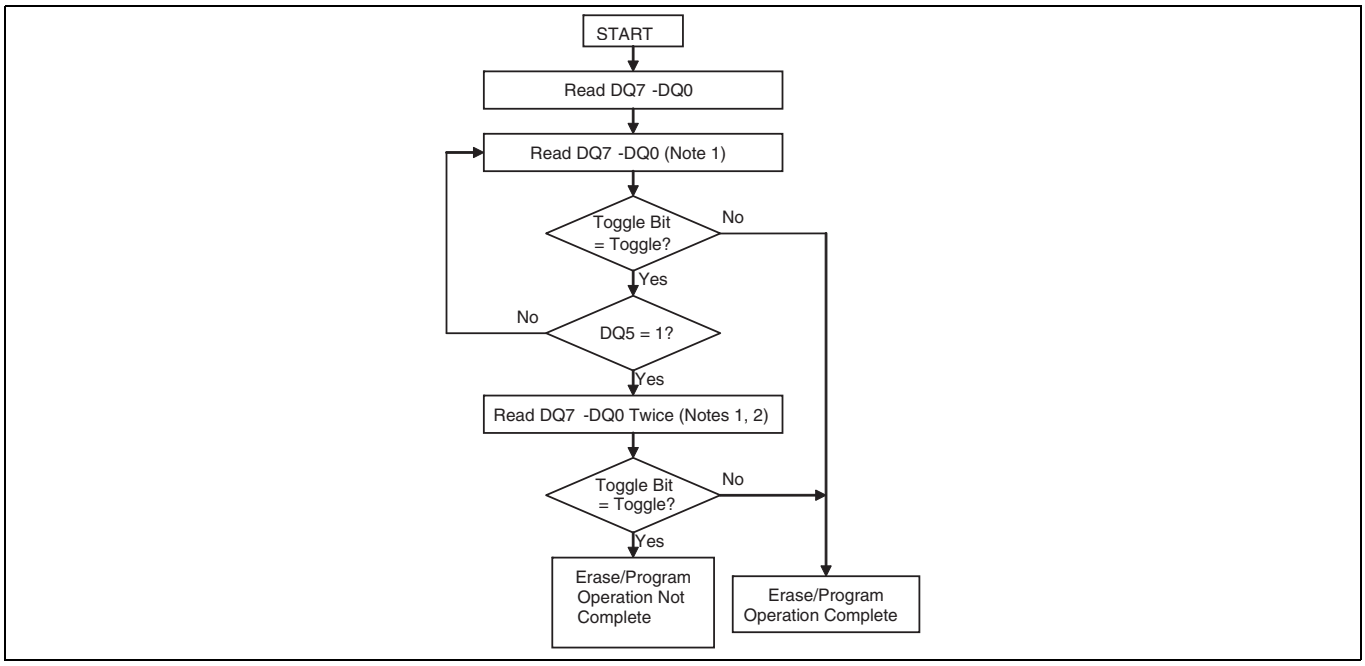



**Figure 8 Toggle Bit program**[24, 25]

**Notes**
24.Read toggle bit twice to determine whether or not it is toggling. See text.
25.Recheck toggle bit because it may stop toggling as DQ5 changes to '1'. See text.

### 5.5.2.6 DQ5: Exceeded timing limits

DQ5 indicates whether the program or erase time has exceeded a specified internal pulse count limit. Under these conditions DQ5 produces '1'. This is a failure condition that indicates the program or erase cycle was not successfully completed. The system must issue the reset command to return the device to reading array data.

When a timeout occurs, the software must send a Soft Reset or Status Register Reset command to clear the timeout bit (DQ5) and to return the EAC to the initial state. In this case, it is possible that the flash will continue to communicate busy for up to $t_{TOR}$ after the reset command is sent.

### 5.5.2.7 DQ1: Write-to-buffer abort

DQ1 indicates whether a write-to-buffer operation was aborted. Under these conditions DQ1 produces '1'. The system must issue the Write-to-Buffer-Abort-Reset command sequence or Status Register Clear command to return the EAC to standby (Read Mode) and the Status Register failed bits are cleared. See **"Write buffer programming"** on page 28 for more details.

**Table 17 Data polling status**

| Operation | | DQ7[27] | DQ6 | DQ5[26] | DQ3 | DQ2[27] | DQ1[29] | RY/BY# |
|---|---|---|---|---|---|---|---|---|
| Standard Mode | Embedded Program Algorithm | DQ7# | Toggle | 0 | N/A | No Toggle | 0 | 0 |
| | Reading within Erasing Sector[30] | 0 | Toggle | 0 | 1 | Toggle | N/A | 0 |
| | Reading Outside erasing Sector[30] | 0 | Toggle | 0 | 1 | No Toggle | N/A | 0 |
| Program Suspend Mode[28] | Reading within Program Suspended Sector | INVALID (Not allowed) | INVALID (Not allowed) | INVALID (Not allowed) | INVALID (Not allowed) | INVALID (Not allowed) | INVALID (Not allowed) | 1 |
| | Reading within Non-Program Suspended Sector | Data | Data | Data | Data | Data | Data | 1 |
| Erase Suspend Mode[32] | Reading within Erase Suspended Sector | 1 | No Toggle | 0 | N/A | Toggle | N/A | 1 |
| | Reading within Non-Erase Suspend Sector | Data | Data | Data | Data | Data | Data | 1 |
| | Programming within Non-Erase Suspended Sector | DQ7# | Toggle | 0 | N/A | N/A | N/A | 0 |
| Write-to-Buffer[29, 31] | BUSY State | DQ7# | Toggle | 0 | N/A | No Toggle | 0 | 0 |
| | Exceeded Timing Limits | DQ7# | Toggle | 1 | N/A | N/A | 0 | 0 |
| | ABORT State | DQ7# | Toggle | 0 | N/A | N/A | 1 | 0 |

**Notes**

26. DQ5 switches to '1' when an embedded program or embedded erase operation has exceeded the maximum timing limits. See **"DQ5: Exceeded timing limits"** on page 49 for more information.
27. DQ7 and DQ2 require a valid address when reading status information. Refer to the appropriate subsection for further details.
28. Data are invalid for addresses in a program suspended line. All addresses other than the program suspended line can be read for valid data.
29. DQ1 indicates the Write-to-Buffer abort status during Write-Buffer-Programming operations.
30. DQ3 = 0 for 50 µs after last sector is loaded during a multi-sector erase.
31. Applies only to program operations.
32. If SECSI is over laid on a suspended sector, if a program operation is initiated while in the SECSI mode, DQ6 will toggle and DQ2 will not toggle during the embedded operation.

## 5.6 Error types and clearing procedures

There are three types of errors reported by the embedded operation status methods. Depending on the error type, the status reported and procedure for clearing the error status is different. Following is the clearing of error status:

- If an ASO was entered before the error the device remains entered in the ASO awaiting ASO read or a command write.
- If an erase was suspended before the error the device returns to the erase suspended state awaiting flash array read or a command write.
- Otherwise, the device will be in standby state awaiting flash array read or a command write.

### 5.6.1 Embedded operation error

If an error occurs during an embedded operation (program, erase, blank check, or password unlock) the device (EAC) remains busy. The RY/BY# output remains LOW, data polling status continues to be overlaid on all address locations, and the status register shows ready with valid status bits. The device remains busy until the error status is detected by the host system status monitoring and the error status is cleared.

During embedded algorithm error status the data polling status will show the following:

- DQ7 is the inversion of the DQ7 bit in the last word loaded into the write buffer or last word of the password in the case of the password unlock command. DQ7 = 0 for an erase, evaluate erase status, blank check failure
- DQ6 continues to toggle
- DQ5 = 1; failure of the embedded operation
- DQ4 is RFU and should be treated as don’t care (masked)
- DQ3 = 1 to indicate an embedded sector erase was in progress or 0 to indicate an embedded program was in progress
- DQ2 continues to toggle, independent of the address used to read status
- DQ1 = 0; write buffer abort error
- DQ0 is RFU and should be treated as don’t care (masked)

During embedded algorithm error status the Status Register will show the following:

- SR[7] = 1; valid status displayed
- SR[6] = X; may or may not be erase suspended during the EA error
- SR[5] = 1 on erase or blank check error; else = 0
- SR[4] = 1 on program or password unlock error; else = 0
- SR[3] = 0; Write buffer abort
- SR[2] = 0; program suspended
- SR[1] = 0; protected sector
- SR[0] = X; RFU, treat as don’t care (masked)

When the embedded algorithm error status is detected, it is necessary to clear the error status in order to return to normal operation, with RY/BY# HIGH, ready for a new read or command write. The error status can be cleared by writing:

- Reset command
- Status Register Clear command

Commands that are accepted during embedded algorithm error status are:

- Status Register Read
- Reset command
- Status Register Clear command

### 5.6.2 Protection error

If an embedded algorithm attempts to change data within a protected area (program, or erase of a protected sector or OTP area) the device (EAC) goes busy for a period of $t_{DP}$ then returns to normal operation. During the busy period the RY/BY# output remains LOW, data polling status continues to be overlaid on all address locations, and the status register shows not ready with invalid status bits (SR[7] = 0).

During the protection error status busy period the data polling status will show the following:

- DQ7 is the inversion of the DQ7 bit in the last word loaded into the write buffer. DQ7 = 0 for an erase failure
- DQ6 continues to toggle, independent of the address used to read status
- DQ5 = 0; to indicate no failure of the embedded operation during the busy period
- DQ4 is RFU and should be treated as don't care (masked)
- DQ3 = 1 to indicate embedded sector erase in progress
- DQ2 continues to toggle, independent of the address used to read status
- DQ1 = 0; Write buffer abort error
- DQ0 is RFU and should be treated as don't care (masked)

Commands that are accepted during the protection error status busy period are:

- Status Register Read

When the busy period ends the device returns to normal operation, the data polling status is no longer overlaid, RY/BY# is HIGH, and the status register shows ready with valid status bits. The device is ready for flash array read or write of a new command.

After the protection error status busy period the Status Register will show the following:

- SR[7] = 1; valid status displayed
- SR[6] = X; may or may not be erase suspended after the protection error busy period
- SR[5] = 1 on erase error, else = 0
- SR[4] = 1 on program error, else = 0
- SR[3] = 0; program not aborted
- SR[2] = 0; no program in suspension
- SR[1] = 1; error due to attempting to change a protected location
- SR[0] = X; RFU, treat as don't care (masked)

Commands that are accepted after the protection error status busy period are:

- Any command

### 5.6.3 Write buffer abort

If an error occurs during a Write to Buffer command the device (EAC) remains busy. The RY/BY# output remains LOW, data polling status continues to be overlaid on all address locations, and the status register shows ready with valid status bits. The device remains busy until the error status is detected by the host system status monitoring and the error status is cleared.

During write to buffer abort (WBA) error status the data polling status will show the following:

- DQ7 is the inversion of the DQ7 bit in the last word loaded into the write buffer
- DQ6 continues to toggle, independent of the address used to read status
- DQ5 = 0; to indicate no failure of the programming operation. WBA is an error in the values input by the Write to Buffer command before the programming operation can begin
- DQ4 is RFU and should be treated as don't care (masked)
- DQ3 is don't care after program operation as no erase is in progress. If the write buffer program operation was started after an erase operation had been suspended then DQ3 = 1. If there was no erase operation in progress then DQ3 is a don't care and should be masked.
- DQ2 does not toggle after program operation as no erase is in progress. If the write buffer program operation was started after an erase operation had been suspended then DQ2 will toggle in the sector where the erase operation was suspended and not in any other sector. If there was no erase operation in progress then DQ2 is a don't care and should be masked.
- DQ1 = 1: Write buffer abort error
- DQ0 is RFU and should be treated as don't care (masked)

During write to buffer abort (WBA) error status the Status Register will show the following:

- SR[7] = 1; valid status displayed
- SR[6] = X; may or may not be erase suspended during the WBA error status
- SR[5] = 0; erase successful
- SR[4] = 1; programming related error
- SR[3] = 1; Write buffer abort
- SR[2] = 0; no program in suspension
- SR[1] = 0; sector not locked during operation
- SR[0] = X; RFU, treat as don't care (masked)

When the WBA error status is detected, it is necessary to clear the error status in order to return to normal operation, with RY/BY# HIGH, ready for a new read or command write. The error status can be cleared and device returned to normal operation by writing:

- Write Buffer Abort Reset command
- Status Register Clear command
- Commands that are accepted during write to buffer abort (WBA) error status are:
- Status Register Read
  - Reads the status register and returns to WBA busy state
- Write Buffer Abort Reset command
- Status Register Clear command

## 5.7 Embedded Algorithm Performance table

The Joint Electron Device Engineering Council (JEDEC) standard JESD22-A117 defines the procedural requirements for performing valid endurance and retention tests based on a qualification specification. This methodology is intended to determine the ability of a flash device to sustain repeated data changes without failure (program/erase endurance) and to retain data for the expected life (data retention). Endurance and retention qualification specifications are specified in JESD47 or may be developed using knowledge-based methods as in JESD94.

**Table 18 Embedded algorithm characteristics (–40°C to +85°C)**

| Parameter | | Min | Typ[34] | Max[35] | Unit | Comments |
|---|---|---|---|---|---|---|
| Sector Erase Time 128 KB | | – | 535 | 3500 | ms | Includes pre-programming prior to erasure[37] |
| Chip Erase | GL512T | – | 274 | 1792[33] | s | |
| | GL01GT | – | 548 | 3584[33] | s | |
| Single Word Programming Time[33] | | – | 160 | 750 | µs | |
| Buffer Programming Time | 2-byte[33] | – | 160 | 750 | µs | |
| | 32-byte[33] | – | 195 | 750 | | |
| | 64-byte[33] | – | 219 | 750 | | |
| | 128-byte[33] | – | 258 | 750 | | |
| | 256-byte[33] | – | 327 | 750 | | |
| | 512-byte[36] | – | 451 | 750 | | |
| Effective Write Buffer Program Operation per Word | 512-byte | – | 1.76 | – | µs | |
| Sector Programming Time 128 KB (full Buffer Programming) | | – | 115.4 | 192 | ms | See Note [38]. |
| Erase Suspend Latency ($t_{ESL}$) | | – | – | 40 | µs | |
| Program Suspend Latency ($t_{PSL}$) | | – | – | 40 | µs | |
| Erase Resume to next Erase Suspend ($t_{ERS}$) | | – | 100 | – | µs | Minimum of 60 µs but ≥ typical periods are needed for Erase to progress to completion. |
| Program Resume to next Program Suspend ($t_{PRS}$) | | – | 100 | – | µs | Minimum of 60 µs but ≥ typical periods are needed for Program to progress to completion. |
| Evaluate Erase Status ($t_{EES}$) | | – | 25 | 30 | µs | |
| Blank Check | | – | 6.2 | 8.5 | ms | |
| NOP (Number of Program-operations, per Line) | | – | – | 256 | | |

**Notes**
33. Not 100% tested.
34. Typical program and erase times assume the following conditions: 25°C, 3.0 V $V_{CC}$, 10,000 cycle, and a random data pattern.
35. Effective write buffer specification is based upon a 512-byte write buffer operation.
36. 512-byte load is not supported in ×8 mode.
37. In the pre-programming step of the Embedded Erase algorithm, all words are programmed to 0000h before sector and chip erasure.
38. System-level overhead is the time required to execute the bus-cycle sequence for the program command. See **Table 23** for further information on command definitions.

**Table 19 Embedded algorithm characteristics (–40°C to +105°C)**

| Parameter | | Min | Typ[40] | Max[41] | Unit | Comments |
|---|---|---|---|---|---|---|
| Sector Erase Time 128 KB | | – | 535 | 3500 | ms | Includes pre-programming prior to erasure (See Note 43) |
| Chip Erase | GL512T | – | 274 | 1792[39] | s | |
| | GL01GT | – | 548 | 3584[39] | | |
| Single Word Programming Time[39] | | – | 160 | 1050 | µs | |
| Buffer Programming Time | 2-byte[39] | – | 160 | 1050 | µs | |
| | 32-byte[39] | – | 195 | 1050 | | |
| | 64-byte[39] | – | 219 | .1050 | | |
| | 128-byte[39] | – | 258 | 1050 | | |
| | 256-byte[39] | – | 327 | 1050 | | |
| | 512-byte[39] | – | 451 | 1050 | | |
| Effective Write Buffer Program Operation per Word | 512-byte | – | 1.76 | – | µs | |
| Sector Programming Time 128 kB (full Buffer Programming) | | – | 115.4 | 269 | ms | See Note 44. |
| Erase Suspend Latency ($t_{ESL}$) | | – | – | 50 | µs | |
| Program Suspend Latency ($t_{PSL}$) | | – | – | 50 | µs | |
| Erase Resume to next Erase Suspend ($t_{ERS}$) | | – | 100 | – | µs | Minimum of 60 ns but ≥ typical periods are needed for Erase to progress to completion. |
| Program Resume to next Program Suspend ($t_{PRS}$) | | – | 100 | – | µs | Minimum of 60 ns but ≥ typical periods are needed for Program to progress to completion. |
| Evaluate Erase Status ($t_{EES}$) | | – | 25 | 30 | µs | |
| Blank Check | | – | 7.6 | 9.0 | ms | |
| NOP (Number of Program-operations, per Line) | | – | – | 1 per 16 word | | |

**Notes**
39. Not 100% tested.
40. Typical program and erase times assume the following conditions: 25°C, 3.0 V $V_{CC}$, 10,000 cycle, and a random data pattern.
41. Effective write buffer specification is based upon a 512-byte write buffer operation.
42. 512-byte load is not supported in ×8 mode.
43. In the pre-programming step of the Embedded Erase algorithm, all words are programmed to 0000h before sector and chip erasure.
44. System-level overhead is the time required to execute the bus-cycle sequence for the program command. See **Table 23** for further information on command definitions.

**Table 20 Embedded algorithm characteristics (–40°C to +125°C)**

| Parameter | | Min | Typ[46] | Max[47] | Unit | Comments |
|---|---|---|---|---|---|---|
| Sector Erase Time 128 KB | | – | 535 | 3500 | ms | |
| Chip Erase | GL512T | – | 274 | 1792[45] | s | Includes pre-programming prior to erasure[49] |
| | GL01GT | – | 548 | 3584[45] | µs | |
| Single Word Programming Time[45] | | – | 160 | 1050 | µs | |
| Buffer Programming Time | 2-byte[45] | – | 160 | 1050 | | |
| | 32-byte[45] | – | 195 | 1050 | | |
| | 64-byte[45] | – | 219 | 1050 | | |
| | 128-byte[45] | – | 258 | 1050 | | |
| | 256-byte[45] | – | 327 | 1050 | | |
| | 512-byte[45] | – | 451 | 1050 | | |
| Effective Write Buffer Program Operation per Word | 512-byte | – | 1.76 | – | µs | |
| Sector Programming Time 128 kB (full Buffer Programming) | | – | 115.4 | 269 | ms | See Note [50]. |
| Erase Suspend Latency (tESL) | | – | – | 50 | µs | |
| Program Suspend Latency (tPSL) | | – | – | 50 | µs | |
| Erase Resume to next Erase Suspend (tERS) | | – | 100 | – | µs | Minimum of 60 ns but ≥ typical periods are needed for Erase to progress to completion. |
| Program Resume to next Program Suspend (tPRS) | | – | 100 | – | µs | Minimum of 60 ns but ≥ typical periods are needed for Program to progress to completion. |
| Evaluate Erase Status (tEES) | | – | 25 | 30 | µs | |
| Blank Check | | – | 7.6 | 9.0 | ms | |
| NOP (Number of Program-operations, per Line) | | – | – | 1 per 16 word | | |

**Notes**
45. Not 100% tested.
46. Typical program and erase times assume the following conditions: 25°C, 3.0 V VCC, 1,000 cycle, and a random data pattern.
47. Effective write buffer specification is based upon a 512-byte write buffer operation.
48. 512-byte load is not supported in ×8 mode.
49. In the pre-programming step of the Embedded Erase algorithm, all words are programmed to 0000h before Sector and Chip erasure.
50. System-level overhead is the time required to execute the bus-cycle sequence for the program command. See **Table 23** for further information on command definitions.

# 6 Data integrity

## 6.1 Erase endurance

**Table 21 Erase endurance**

| Parameter | Minimum | Unit |
|---|---|---|
| Program/erase cycles per main flash array sectors | 100K | P/E cycle |
| Program/erase cycles per PPB array or non-volatile register array [51] | 100K | P/E cycle |

## 6.2 Data retention

**Table 22 Data retention**

| Parameter | Test conditions | Minimum time | Unit |
|---|---|---|---|
| Data retention time | 1K program/erase cycles | 20 | Years |
| | 10K program/erase cycles | 2 | Years |
| | 100K program/erase cycles | 0.2 | Years |

Contact Infineon Sales or an FAE representative for additional information on the data integrity.

**Note**
51. Each write command to a non-volatile register causes a P/E cycle on the entire non-volatile register array. OTP bits and registers internally reside in a separate array that is not P/E cycled.

# 7 Software interface reference

## 7.1 Command summary

**Table 23 Command definitions ×16**

| Command sequence[52] | | Cycles | Bus cycles[53, 54, 55, 56] | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | First | | Second | | Third | | Fourth | | Fifth | | Sixth | | Seventh | |
| | | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| Read[57] | | 1 | RA | RD | | | | | | | | | | | | |
| Reset/ASO Exit[58, 68] | | 1 | XXX | F0 | | | | | | | | | | | | |
| Status Register Read | | 2 | 555 | 70 | XXX | RD | | | | | | | | | | |
| Status Register Clear | | 1 | 555 | 71 | | | | | | | | | | | | |
| Word Program | | 4 | 555 | AA | 2AA | 55 | 555 | A0 | PA | PD | | | | | | |
| Write to Buffer | | 6 | 555 | AA | 2AA | 55 | SA | 25 | SA | WC | WBL | PD | WBL | PD | | |
| Program Buffer to Flash (confirm) | | 1 | SA | 29 | | | | | | | | | | | | |
| Write-to-Buffer-Abort Reset[64] | | 3 | 555 | AA | 2AA | 55 | 555 | F0 | | | | | | | | |
| Unlock Bypass | Enter | 3 | 555 | AA | 2AA | 55 | 555 | 20 | | | | | | | | |
| | Program[60] | 2 | XXX | A0 | PA | PD | | | | | | | | | | |
| | Write-to-Buffer[60] | 4 | SA | 25 | SA | WC | WBL | PD | WBL | PD | | | | | | |
| | Program Buffer to Flash (confirm) | 1 | SA | 29 | | | | | | | | | | | | |
| | Write-to-Buffer-Abort Reset[64] | 3 | 555 | AA | 2AA | 55 | 555 | F0 | | | | | | | | |
| | Sector Erase[60] | 2 | XXX | 80 | SA | 30 | | | | | | | | | | |
| | Chip Erase[60] | 2 | XXX | 80 | XXX | 10 | | | | | | | | | | |
| | Command Set Exit[61] | 2 | XXX | 90 | XXX | 00 | | | | | | | | | | |
| Chip Erase | | 6 | 555 | AA | 2AA | 55 | 555 | 80 | 555 | AA | 2AA | 55 | 555 | 10 | | |
| Sector Erase[71] | | 6 | 555 | AA | 2AA | 55 | 555 | 80 | 555 | AA | 2AA | 55 | SA | 30 | | |
| Erase Suspend/Program Suspend Legacy Method[62] | | 1 | XXX | B0 | | | | | | | | | | | | |
| Erase Suspend Enhanced Method | | | | | | | | | | | | | | | | |
| Erase Resume/Program Resume Legacy Method[63] | | 1 | XXX | 30 | | | | | | | | | | | | |
| Erase Resume Enhanced Method | | | | | | | | | | | | | | | | |
| Program Suspend Enhanced Method | | 1 | XXX | 51 | | | | | | | | | | | | |
| Program Resume Enhanced Method | | 1 | XXX | 50 | | | | | | | | | | | | |
| Evaluate Erase State | | 1 | (SA) 555 | 35 | | | | | | | | | | | | |
| Blank Check | | 1 | (SA) 555 | 33 | | | | | | | | | | | | |
| CFI Enter[59] | | 1 | (SA) 55 | 98 | | | | | | | | | | | | |
| Continuity Check | | 7 | 555 | 71 | 555 | 70 | XX | RD | 2AAAA55 | FF00 | 15555AA | 00FF | 555 | 70 | XX | RD |
| ID-CFI (Autoselect) ASO | ID (Autoselect) Entry | 3 | 555 | AA | 2AA | 55 | 555 | 90 | | | | | | | | |
| | CFI Enter[59] | 1 | 55 | 98 | | | | | | | | | | | | |
| | ID-CFI Read | 1 | RA | RD | | | | | | | | | | | | |
| | CFI Exit | 1 | XXX | FF | | | | | | | | | | | | |
| | Reset/ASO Exit[58, 69] | 1 | XXX | F0 | | | | | | | | | | | | |

**Table 23 Command definitions ×16** (Continued)

| | Command sequence[52] | Cycles | Bus cycles[53, 54, 55, 56] | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | First | | Second | | Third | | Fourth | | Fifth | | Sixth | | Seventh | |
| | | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| **Secure Silicon Region Command Definitions** | | | | | | | | | | | | | | | | |
| Secure Silicon Region (SSR) ASO | SSR Entry | 3 | 555 | AA | 2AA | 55 | (SA) 555 | 88 | | | | | | | | |
| | Read[57] | 1 | RA | RD | | | | | | | | | | | | |
| | Word Program | 4 | 555 | AA | 2AA | 55 | 555 | A0 | PA | PD | | | | | | |
| | Write to Buffer | 6 | 555 | AA | 2AA | 55 | SA | 25 | SA | WC | WBL | PD | WBL | PD | | |
| | Program Buffer to Flash (confirm) | 1 | SA | 29 | | | | | | | | | | | | |
| | Write-to-Buffer-Abort Reset[64] | 3 | 555 | AA | 2AA | 55 | 555 | F0 | | | | | | | | |
| | SSR Exit[64] | 4 | 555 | AA | 2AA | 55 | 555 | 90 | XX | 0 | | | | | | |
| | Reset/ASO Exit[58, 69] | 1 | XXX | F0 | | | | | | | | | | | | |
| **Lock Register Command Set Definitions** | | | | | | | | | | | | | | | | |
| Lock Register ASO | Lock Register Entry | 3 | 555 | AA | 2AA | 55 | 555 | 40 | | | | | | | | |
| | Program[68] | 2 | XXX | A0 | XXX | PD | | | | | | | | | | |
| | Read[68] | 1 | 0 | RD | | | | | | | | | | | | |
| | Command Set Exit[65, 69] | 2 | XXX | 90 | XXX | 0 | | | | | | | | | | |
| | Reset/ASO Exit[58, 69] | 1 | XXX | F0 | | | | | | | | | | | | |
| **Password Protection Command Set Definitions** | | | | | | | | | | | | | | | | |
| Password ASO | Password ASO Entry | 3 | 555 | AA | 2AA | 55 | 555 | 60 | | | | | | | | |
| | Program[67] | 2 | XXX | A0 | PWAx | PWDx | | | | | | | | | | |
| | Read[68] | 4 | 0 | PWD0 | 1 | PWD1 | 2 | PWD2 | 3 | PWD3 | | | | | | |
| | Unlock[68] | 7 | 0 | 25 | 0 | 3 | 0 | PWD0 | 1 | PWD1 | 2 | PWD2 | 3 | PWD3 | 0 | 29 |
| | Command Set Exit[65, 69] | 2 | XXX | 90 | XXX | 0 | | | | | | | | | | |
| | Reset/ASO Exit[58, 69] | 1 | XXX | F0 | | | | | | | | | | | | |
| **Non-Volatile Sector Protection Command Set Definitions** | | | | | | | | | | | | | | | | |
| PPB (Non-Volatile | PPB Entry | 3 | 555 | AA | 2AA | 55 | 555 | C0 | | | | | | | | |
| | PPB Program[70] | 2 | XXX | A0 | SA | 0 | | | | | | | | | | |
| | All PPB Erase[70] | 2 | XXX | 80 | 0 | 30 | | | | | | | | | | |
| | PPB Read[70] | 1 | SA | RD (0) | | | | | | | | | | | | |
| | Command Set Exit[65, 69] | 2 | XXX | 90 | XXX | 0 | | | | | | | | | | |
| | Reset/ASO Exit[58, 69] | 1 | XXX | F0 | | | | | | | | | | | | |
| **Global Non-Volatile Sector Protection Freeze Command Set Definitions** | | | | | | | | | | | | | | | | |
| PPB Lock Bit | PPB Lock Entry | 3 | 555 | AA | 2AA | 55 | 555 | 50 | | | | | | | | |
| | PPB Lock Bit Cleared | 2 | XXX | A0 | XXX | 0 | | | | | | | | | | |
| | PPB Lock Status Read[70] | 1 | XXX | RD (0) | | | | | | | | | | | | |
| | Command Set Exit[65, 69] | 2 | XXX | 90 | XXX | 0 | | | | | | | | | | |
| | Reset/ASO Exit[69] | 1 | XXX | F0 | | | | | | | | | | | | |
| **Volatile Sector Protection Command Set Definitions** | | | | | | | | | | | | | | | | |
| DYB (Volatile Sector | DYB ASO Entry | 3 | 555 | AA | 2AA | 55 | 555 | E0 | | | | | | | | |
| | DYB Set[70] | 2 | XXX | A0 | SA | 0 | | | | | | | | | | |
| | DYB Clear[70] | 2 | XXX | A0 | SA | 1 | | | | | | | | | | |
| | DYB Status Read[70] | 1 | SA | RD (0) | | | | | | | | | | | | |
| | Command Set Exit[65, 69] | 2 | XXX | 90 | XXX | 0 | | | | | | | | | | |
| | Reset/ASO Exit[69] | 1 | XXX | F0 | | | | | | | | | | | | |
| **Command Set Definitions ECC** | | | | | | | | | | | | | | | | |
| ECC ASO | ECC ASO Entry | 3 | 555 | AA | 2AA | 55 | 555 | 75 | | | | | | | | |
| | ECC Status Read | 1 | RA | RD | | | | | | | | | | | | |
| | Command Set Exit[65, 69] | 1 | XXX | F0 | | | | | | | | | | | | |

**Legend:**
X = Don't care.
RA = Address of the memory to be read.
RD = Data read from location RA during read operation.
PA = Address of the memory location to be programmed.
PD = Data to be programmed at location PA.
SA = Address of the sector selected. Address bits Amax–A16 uniquely select any sector.
WBL = Write Buffer Location. The address must be within the same Line.
WC = Word Count is the number of write buffer locations to load minus 1.
PWAx = PPB Password address for word0 = 00h, word1 = 01h, word2 = 02h, and word3 = 03h. SSR3 Password address for word0 = 10h, word1 = 11h, word2 = 12h, and word3 = 13h.
PWDx = Password data word0, word1, word2, and word3.
Gray vs. White Box = Read vs. Write Operation.

**Notes**
52. See **Table 31** for description of bus operations.
53. All values are in hexadecimal.
54. Except for the following, all bus cycles are write cycle: read cycle during Read, ID/CFI Read (Manufacturing ID / device ID), indicator bits, secure silicon region read, SSR lock read, and 2nd cycle of Status Register read.
55. Data bits DQ15–DQ8 are don't care in command sequences, except for RD, PD, WC and PWD.
56. Address bits Amax–A11 are don't cares for unlock and command cycles, unless SA or PA required (Amax is the highest Address pin).
57. No unlock or command cycles required when reading array data.
58. The Reset command is required to return to reading array data when device is in the ASO mode, or if DQ5 goes HIGH (while the device is providing status data).
59. Command is valid when device is ready to read array data.
60. The Unlock-Bypass command is required prior to the Unlock-Bypass-Program and the unlock bypass write to buffer commands.
61. The Unlock-Bypass-Reset command is required to return to reading array data when the device is in the unlock bypass mode.
62. The system can read and program/program suspend in non-erasing sectors, or enter the ID-CFI ASO, when in the erase suspend mode. The Erase Suspend command is valid only during a sector erase operation.
63. The Erase Resume/Program Resume command is valid only during the erase suspend/program suspend modes.
64. Issue this command sequence to return to read state after detecting device is in a Write-to-Buffer-Abort state. IMPORTANT: the full command sequence is required if resetting out of abort.
65. The Exit command returns the device to reading the array.
66. The password portion can be entered or read in any order as long as the entire 64-bit password is entered or read. Addresses are 10h–13h if the SSR3 is being accessed.
67. For PWDx, only one portion of the password can be programmed per each A0 command. Portions of the password must be programmed in sequential order (PWD0–PWD3).
68. All lock register bits are one-time programmable. The program state = 0 and the erase state = 1. Also, both the persistent protection mode lock bit and the password protection mode lock bit cannot be programmed at the same time or the lock register bits program operation halts and returns the device to read State. Lock register bits that are reserved for future use are undefined and may be 0's or 1's.
69. If any of the Entry commands was issued, an Exit command must be issued to reset the device into Read State.
70. Protected State = 00h, Unprotected State = 01h. The sector address for DYB set, DYB clear, or PPB Program command may be any location within the sector - the lower order bits of the sector address are don't care.
71. See **"Sector erase"** on page 36 for description of Multi-Sector Erase.

## Table 24 Command definitions ×8

| Command sequence[72] | | Cycles | Bus cycles[73, 74, 75, 76] | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | First | | Second | | Third | | Fourth | | Fifth | | Sixth | | Seventh | |
| | | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| Read[76] | | 1 | RA | RD | | | | | | | | | | | | |
| Reset/ASO Exit[77, 88] | | 1 | XXX | F0 | | | | | | | | | | | | |
| Status Register Read | | 2 | AAA | 70 | XXX | RD | | | | | | | | | | |
| Status Register Clear | | 1 | AAA | 71 | | | | | | | | | | | | |
| Word Program | | 4 | AAA | AA | 555 | 55 | AAA | A0 | PA | PD | | | | | | |
| Write to Buffer[90] | | 6 | AAA | AA | 555 | 55 | SA | 25 | SA | WC | WBL | PD | WBL | PD | | |
| Program Buffer to Flash (confirm) | | 1 | SA | 29 | | | | | | | | | | | | |
| Write-to-Buffer-Abort Reset[83] | | 3 | AAA | AA | 555 | 55 | AAA | F0 | | | | | | | | |
| Unlock Bypass | Enter | 3 | AAA | AA | 555 | 55 | AAA | 20 | | | | | | | | |
| | Program[79] | 2 | XXX | A0 | PA | PD | | | | | | | | | | |
| | Write-to-Buffer[79] | 4 | SA | 25 | SA | WC | WBL | PD | WBL | PD | | | | | | |
| | Program Buffer to Flash (confirm)[79] | 1 | SA | 29 | | | | | | | | | | | | |
| | Write-to-Buffer-Abort Reset[83] | 3 | AAA | AA | 555 | 55 | AAA | F0 | | | | | | | | |
| | Sector Erase[79] | 2 | XXX | 80 | SA | 30 | | | | | | | | | | |
| | Chip Erase[79] | 2 | XXX | 80 | XXX | 10 | | | | | | | | | | |
| | Command Set Exit[80] | 2 | XXX | 90 | XXX | 00 | | | | | | | | | | |
| Chip Erase | | 6 | AAA | AA | 555 | 55 | AAA | 80 | AAA | AA | 555 | 55 | AAA | 10 | | |
| Sector Erase[90] | | 6 | AAA | AA | 555 | 55 | AAA | 80 | AAA | AA | 555 | 55 | SA | 30 | | |
| Erase Suspend/Program Suspend Legacy Method[81] | | 1 | XXX | B0 | | | | | | | | | | | | |
| Erase Suspend Enhanced Method | | | | | | | | | | | | | | | | |
| Erase Resume/Program Resume Legacy Method[82] | | 1 | XXX | 30 | | | | | | | | | | | | |
| Erase Resume Enhanced Method | | | | | | | | | | | | | | | | |
| Program Suspend Enhanced Method | | 1 | XXX | 51 | | | | | | | | | | | | |
| Program Resume Enhanced Method | | 1 | XXX | 50 | | | | | | | | | | | | |
| Evaluate Erase State | | 1 | (SA) AAA | 35 | | | | | | | | | | | | |
| Blank Check | | 1 | (SA) AAA | 33 | | | | | | | | | | | | |
| CFI Enter[78] | | 1 | (SA) AA | 98 | | | | | | | | | | | | |
| Continuity Check | | 7 | AAA | 71 | AAA | 70 | XX | RD | 55554AB | FF | 2AAAB54 | 00 | AAA | 70 | XX | RD |
| ID-CFI (Autoselect) | ID (Autoselect) Entry | 3 | AAA | AA | 555 | 55 | AAA | 90 | | | | | | | | |
| | CFI Enter[78] | 1 | AA | 98 | | | | | | | | | | | | |
| | ID-CFI Read | 1 | RA | RD | | | | | | | | | | | | |
| | CFI Exit | 1 | XXX | FF | | | | | | | | | | | | |
| | Reset/ASO Exit[77, 88] | 1 | XXX | F0 | | | | | | | | | | | | |

**Table 24 Command definitions ×8** (Continued)

| | Command sequence[72] | Cycles | Bus cycles[73, 74, 75, 76] | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | First | | Second | | Third | | Fourth | | Fifth | | Sixth | | Seventh | |
| | | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| **Secure Silicon Region Command Definitions** | | | | | | | | | | | | | | | | |
| Secure Silicon Region | SSR Entry | 3 | AAA | AA | 555 | 55 | (SA) AAA | 88 | | | | | | | | |
| | Read[76] | 1 | RA | RD | | | | | | | | | | | | |
| | Word Program | 4 | AAA | AA | 555 | 55 | AAA | A0 | PA | PD | | | | | | |
| | Write to Buffer[90] | 6 | AAA | AA | 555 | 55 | SA | 25 | SA | WC | WBL | PD | WBL | PD | | |
| | Program Buffer to Flash (confirm) | 1 | SA | 29 | | | | | | | | | | | | |
| | Write-to-Buffer-Abort Reset[83] | 3 | AAA | AA | 555 | 55 | AAA | F0 | | | | | | | | |
| | SSR Exit[83] | 4 | AAA | AA | 555 | 55 | AAA | 90 | XX | 0 | | | | | | |
| | Reset/ASO Exit[77, 88] | 1 | XXX | F0 | | | | | | | | | | | | |
| **Lock Register Command Set Definitions** | | | | | | | | | | | | | | | | |
| Lock Register ASO | Lock Register Entry | 3 | AAA | AA | 555 | 55 | AAA | 40 | | | | | | | | |
| | Program[87] | 2 | XXX | A0 | XXX | PD | | | | | | | | | | |
| | Read[87] | 1 | 0 | RD | | | | | | | | | | | | |
| | Command Set Exit[84, 88] | 2 | XXX | 90 | XXX | 0 | | | | | | | | | | |
| | Reset/ASO Exit[77, 88] | 1 | XXX | F0 | | | | | | | | | | | | |
| **Password Protection Command Set Definitions** | | | | | | | | | | | | | | | | |
| Password ASO | Password ASO Entry | 3 | AAA | AA | 555 | 55 | AAA | 60 | | | | | | | | |
| | Program[86] | 2 | XXX | A0 | PWAx | PWDx | | | | | | | | | | |
| | Read[85] | 8 | 0 | PWD0 | 1 | PWD1 | 2 | PWD2 | 3 | PWD3 | 4 | PWD4 | 5 | PWD5 | 6 | PWD6 |
| | | | 7 | PWD7 | | | | | | | | | | | | |
| | Unlock[85] | 11 | 0 | 25 | 0 | 3 | 0 | PWD0 | 1 | PWD1 | 2 | PWD2 | 3 | PWD3 | 4 | PWD4 |
| | | | 5 | PWD5 | 6 | PWD6 | 7 | PWD7 | 0 | 29 | | | | | | |
| | Command Set Exit[84, 88] | 2 | XXX | 90 | XXX | 0 | | | | | | | | | | |
| | Reset/ASO Exit[77, 88] | 1 | XXX | F0 | | | | | | | | | | | | |
| **Non-Volatile Sector Protection Command Set Definitions** | | | | | | | | | | | | | | | | |
| PPB (Non-Volatile | PPB Entry | 3 | AAA | AA | 555 | 55 | AAA | C0 | | | | | | | | |
| | PPB Program[89] | 2 | XXX | A0 | SA | 0 | | | | | | | | | | |
| | All PPB Erase[89] | 2 | XXX | 80 | 0 | 30 | | | | | | | | | | |
| | PPB Read[89] | 1 | SA | RD (0) | | | | | | | | | | | | |
| | Command Set Exit[84, 88] | 2 | XXX | 90 | XXX | 0 | | | | | | | | | | |
| | Reset/ASO Exit[77, 88] | 1 | XXX | F0 | | | | | | | | | | | | |
| **Global Non-Volatile Sector Protection Freeze Command Set Definitions** | | | | | | | | | | | | | | | | |
| PPB Lock Bit | PPB Lock Entry | 3 | AAA | AA | 555 | 55 | AAA | 50 | | | | | | | | |
| | PPB Lock Bit Cleared | 2 | XXX | A0 | XXX | 0 | | | | | | | | | | |
| | PPB Lock Status Read[89] | 1 | XXX | RD (0) | | | | | | | | | | | | |
| | Command Set Exit[84, 88] | 2 | XXX | 90 | XXX | 0 | | | | | | | | | | |
| | Reset/ASO Exit[88] | 1 | XXX | F0 | | | | | | | | | | | | |
| **Volatile Sector Protection Command Set Definitions** | | | | | | | | | | | | | | | | |
| DYB (Volatile Sector | DYB ASO Entry | 3 | AAA | AA | 555 | 55 | AAA | E0 | | | | | | | | |
| | DYB Set[89] | 2 | XXX | A0 | SA | 0 | | | | | | | | | | |
| | DYB Clear[89] | 2 | XXX | A0 | SA | 1 | | | | | | | | | | |
| | DYB Status Read[89] | 1 | SA | RD (0) | | | | | | | | | | | | |
| | Command Set Exit[84, 88] | 2 | XXX | 90 | XXX | 0 | | | | | | | | | | |
| | Reset/ASO Exit[88] | 1 | XXX | F0 | | | | | | | | | | | | |
| **ECC Command Set Definitions** | | | | | | | | | | | | | | | | |
| ECC ASO | ECC ASO Entry | 3 | AAA | AA | 555 | 55 | AAA | 75 | | | | | | | | |
| | ECC Status Read | 1 | RA | RD | | | | | | | | | | | | |
| | Command Set Exit[84, 88] | 1 | XXX | F0 | | | | | | | | | | | | |

**Legend:**
X = Don't care.
RA = Address of the memory to be read.
RD = Data read from location RA during read operation.
PA = Address of the memory location to be programmed.
PD = Data to be programmed at location PA.
SA = Address of the sector selected. Address bits Amax–A16 uniquely select any sector.
WBL = Write Buffer Location. The address must be within the same Line.
WC = Word Count is the number of write buffer locations to load minus 1.
PWAx = PPB Password address for byte0 = 00h, byte1 = 01h, byte2 = 02h, byte3 = 03h, byte04= 04h, byte5 = 05h, byte6 = 06h, and byte7 = 07h. SSR3 Password address for byte0 = 20h, byte1 = 21h, byte2 = 22h, byte3 = 23h, byte04= 24h, byte5 = 25h, byte6 = 26h, and byte7 = 27h.
PWDx = Password data byte0, byte1, byte2, byte3, byte4, byte5, byte6, and byte7
Gray vs. White Box = Read vs. Write Operation.

**Notes**
72. See **Table 31** for description of bus operations.
73. All values are in hexadecimal.
74. Except for the following, all bus cycles are write cycle: read cycle during read, ID/CFI read (Manufacturing ID / Device ID), indicator bits, secure silicon region read, SSR lock read, and 2nd cycle of Status Register read.
75. Address bits Amax–A11 are don't cares for unlock and command cycles, unless SA or PA required (Amax is the highest address pin).
76. No unlock or command cycles required when reading array data.
77. The Reset command is required to return to reading array data when device is in the ASO mode, or if DQ5 goes HIGH (while the device is providing status data).
78. Command is valid when device is ready to read array data.
79. The Unlock-Bypass command is required prior to the Unlock-Bypass-Program command and the unlock bypass write to buffer commands.
80. The Unlock-Bypass-Reset command is required to return to reading array data when the device is in the unlock bypass mode.
81. The system can read and program/program suspend in non-erasing sectors, or enter the ID-CFI ASO, when in the erase suspend mode. The Erase Suspend command is valid only during a sector erase operation.
82. The Erase Resume/Program Resume command is valid only during the erase suspend/program suspend modes.
83. Issue this command sequence to return to read state after detecting device is in a Write-to-Buffer-Abort state. Important: the full command sequence is required if resetting out of abort.
84. The Exit command returns the device to reading the array.
85. The password portion can be entered or read in any order as long as the entire 64-bit password is entered or read. Addresses are 20h–27h if the SSR3 is being accessed.
86. For PWDx, only one portion of the password can be programmed per each A0 command. Portions of the password must be programmed in sequential order (PWD0–PWD7).
87. All lock register bits are one-time programmable. The program state = 0 and the erase state = 1. Also, both the persistent protection mode lock bit and the password protection mode lock bit cannot be programmed at the same time or the lock register bits program operation aborts and returns the device to read state. Lock register bits that are reserved for future use are undefined and may be 0's or 1's.
88. If any of the Entry commands was issued, an Exit command must be issued to reset the device into read state.
89. Protected State = 00h, Unprotected State = 01h. The sector address for DYB set, DYB clear, or PPB Program command may be any location within the sector - the lower order bits of the sector address are don't care.
90. See **"Sector erase"** on page 36 for description of multi-sector erase.
91. In ×8 mode, the WC represents 2 x8 WBL/PD cycles (e.g. if WC = 0, then 5th bus cycle would load data to lower byte address A1 = LOW and 6th bus cycle would load data to upper byte address A1 = HIGH).

## 7.2 Device ID and Common Flash Interface (ID-CFI) ASO map

The device ID portion of the ASO (word locations 0h to 0Fh) provides manufacturer ID, device ID, sector protection state, and basic feature set information for the device.

The access time to read location 02h is always $t_{ACC}$ and a read of this location requires CE# to go HIGH before the read and return LOW to initiate the read (asynchronous read access). Page mode read between location 02h and other ID locations is not supported. Page mode read between ID locations other than 02h is supported.

In ×8 mode, address A1 is ignored and the lower 8 bits of data will be returned for both address, in CFI only. While in ×8 only CFI or only Autoselect data can be read. In ×16 mode, able to read both memories from either command.

For additional information, see **"ID-CFI ASO"** on page 39.

**Table 25 ID (Autoselect) address map**

| Description | Address (×16) | Address (×8) | Read data |
|---|---|---|---|
| Manufacture ID | (SA) + 0000h | (SA) + 0000h | 0001h |
| Device ID | (SA) + 0001h | (SA) + 0002h | 227Eh |
| Protection verification | (SA) + 0002h | (SA) + 0004h | Sector protection state (1 = Sector protected, 0 = Sector unprotected).<br>To read a different SA protection state, only a new SA needs to be given. |
| Indicator bits | (SA) + 0003h | (SA) + 0006h | DQ15–DQ08 = 1 (Reserved)<br>DQ7 - Factory locked secure silicon region<br>1 = Locked,<br>0 = Not locked<br>DQ6 - Customer locked secure silicon region<br>1 = Locked<br>0 = Not locked<br>DQ5 = 1 (Reserved)<br>DQ4 - WP# protects<br>0 = Lowest address sector<br>1 = Highest address sector<br>DQ3–DQ0 = 1 (Reserved) |
| RFU | (SA) + 0004h | (SA) + 0008h | Reserved |
| | (SA) + 0005h | (SA) + 000Ah | Reserved |
| | (SA) + 0006h | (SA) + 000Ch | Reserved |
| | (SA) + 0007h | (SA) + 000Eh | Reserved |
| | (SA) + 0008h | (SA) + 0010h | Reserved |
| | (SA) + 0009h | (SA) + 0012h | Reserved |
| | (SA) + 000Ah | (SA) + 0014h | Reserved |
| | (SA) + 000Bh | (SA) + 0016h | Reserved |
| Lower software bits | (SA) + 000Ch | (SA) + 0018h | Bit 0 - Status Register support<br>1 = Status Register supported<br>0 = Status Register not supported<br>Bit 1 - DQ polling support<br>1 = DQ bits polling supported<br>0 = DQ bits polling not supported<br>Bit 3–2 - Command set support<br>11 = reserved<br>10 = reserved<br>01 = Reduced Command set<br>00 = Classic Command set<br>Bits 4–15 - Reserved = 0 |
| Upper software bits | (SA) + 000Dh | (SA) + 001Ah | Reserved |

**Table 25 ID (Autoselect) address map** (Continued)

| Description | Address (×16) | Address (×8) | Read data |
|---|---|---|---|
| Device ID | (SA) + 000Eh | (SA) + 001Ch | 2228h = 1 Gb<br>2223h = 512 Mb |
| Device ID | (SA) + 000Fh | (SA) + 001Eh | 2201h |

**Table 26 CFI query identification string**

| Word address | Byte address | Data | Description |
|---|---|---|---|
| (SA) + 0010h<br>(SA) + 0011h<br>(SA) + 0012h | (SA) + 0020h<br>(SA) + 0022h<br>(SA) + 0024h | 0051h<br>0052h<br>0059h | Query Unique ASCII string “QRY” |
| (SA) + 0013h<br>(SA) + 0014h | (SA) + 0026h<br>(SA) + 0028h | 0002h<br>0000h | Primary OEM Command set |
| (SA) + 0015h<br>(SA) + 0016h | (SA) + 002Ah<br>(SA) + 002Ch | 0040h<br>0000h | Address for Primary Extended table |
| (SA) + 0017h<br>(SA) + 0018h | (SA) + 002Eh<br>(SA) + 0030h | 0000h<br>0000h | Alternate OEM Command set<br>(00h = none exists) |
| (SA) + 0019h<br>(SA) + 001Ah | (SA) + 0032h<br>(SA) + 0034h | 0000h<br>0000h | Address for Alternate OEM Extended table<br>(00h = none exists) |

**Table 27 CFI system interface string**

| Word address | Byte address | Data | Description |
|---|---|---|---|
| (SA) + 001Bh | (SA) + 0036h | 0027h | $V_{CC}$ Min. (erase/program) (D7–D4: volts, D3–D0: 100 mV) |
| (SA) + 001Ch | (SA) + 0038h | 0036h | $V_{CC}$ Max. (erase/program) (D7–D4: volts, D3–D0: 100 mV) |
| (SA) + 001Dh | (SA) + 003Ah | 0000h | $V_{PP}$ Min. voltage (00h = no $V_{PP}$ pin present) |
| (SA) + 001Eh | (SA) + 003Ch | 0000h | $V_{PP}$ Max. voltage (00h = no $V_{PP}$ pin present) |
| (SA) + 001Fh | (SA) + 003Eh | 0008h | Typical timeout per single word write $2^N$ µs |
| (SA) + 0020h | (SA) + 0040h | 0009h | Typical timeout for max multi-byte program, $2^N$ µs (00h = not supported) |
| (SA) + 0021h | (SA) + 0042h | 000Ah | Typical timeout per individual block erase $2^N$ ms |
| (SA) + 0022h | (SA) + 0044h | 0014h (1 Gb)<br>0013h (512 Mb) | Typical timeout for full chip erase $2^N$ ms (00h = not supported) |
| (SA) + 0023h | (SA) + 0046h | 0002h (85°C)<br>0003h (105°C) | Max. timeout for single word write $2^N$ times typical |
| (SA) + 0024h | (SA) + 0048h | 0001h (85°C)<br>0002h (105°C) | Max. timeout for buffer write $2^N$ times typical |
| (SA) + 0025h | (SA) + 004Ah | 0002h | Max. timeout per individual block erase $2^N$ times typical |
| (SA) + 0026h | (SA) + 004Ch | 0002h | Max. timeout for full chip erase $2^N$ times typical (00h = not supported) |

**Table 28 CFI device geometry definition**

| Word address | Byte address | Data | Description |
|---|---|---|---|
| (SA) + 0027h | (SA) + 004Eh | 001Bh (1 Gb)<br>001Ah (512 Mb) | Device Size = $2^N$ byte; |
| (SA) + 0028h | (SA) + 0050h | 0002h | Flash Device Interface Description 0 = ×8-only, 1 = ×16-only, 2 = ×8/×16 capable |
| (SA) + 0029h | (SA) + 0052h | 0000h | |
| (SA) + 002Ah | (SA) + 0054h | 0009h | Max. number of byte in multi-byte write = $2^N$ (00 = not supported)<br>**Note** For ×16 (WORD) mode only. |
| (SA) + 002Bh | (SA) + 0056h | 0000h | |
| (SA) + 002Ch | (SA) + 0058h | 0001h | Number of Erase Block Regions within device<br>1 = Uniform device, 2 = Boot device |
| (SA) + 002Dh | (SA) + 005Ah | 00XXh | Erase Block Region 1 information (refer to JEDEC JESD68-01 or JEP137 specifications)<br>00FFh, 0003h, 0000h, 0002h =1 Gb<br>00FFh, 0001h, 0000h, 0002h = 512 Mb |
| (SA) + 002Eh | (SA) + 005Ch | 000Xh | |
| (SA) + 002Fh | (SA) + 005Eh | 0000h | |
| (SA) + 0030h | (SA) + 0060h | 000Xh | |
| (SA) + 0031h | (SA) + 0062h | 0000h | Erase Block Region 2 information (refer to CFI publication 100) |
| (SA) + 0032h | (SA) + 0064h | 0000h | |
| (SA) + 0033h | (SA) + 0066h | 0000h | |
| (SA) + 0034h | (SA) + 0068h | 0000h | |
| (SA) + 0035h | (SA) + 006Ah | 0000h | Erase Block Region 3 information (refer to CFI publication 100) |
| (SA) + 0036h | (SA) + 006Ch | 0000h | |
| (SA) + 0037h | (SA) + 006Eh | 0000h | |
| (SA) + 0038h | (SA) + 0070h | 0000h | |
| (SA) + 0039h | (SA) + 0072h | 0000h | Erase Block Region 4 information (refer to CFI publication 100) |
| (SA) + 003Ah | (SA) + 0074h | 0000h | |
| (SA) + 003Bh | (SA) + 0076h | 0000h | |
| (SA) + 003Ch | (SA) + 0078h | 0000h | |
| (SA) + 003Dh | (SA) + 007Ah | FFFFh | Reserved |
| (SA) + 003Eh | (SA) + 007Ch | FFFFh | |
| (SA) + 003Fh | (SA) + 007Eh | FFFFh | |

**Table 29 CFI primary vendor-specific extended query**

| Word address | Byte address | Data | Description |
|---|---|---|---|
| (SA) + 0040h | (SA) + 0080h | 0050h | Query-unique ASCII string “PRI” |
| (SA) + 0041h | (SA) + 0082h | 0052h | |
| (SA) + 0042h | (SA) + 0084h | 0049h | |
| (SA) + 0043h | (SA) + 0086h | 0031h | Major version number, ASCII |
| (SA) + 0044h | (SA) + 0088h | 0033h (CFI 1.3)<br>0035H (CFI 1.5) | Minor version number, ASCII<br>0033h = CFI Minor Version 3 (Model Number is 03, 04, V3, and V4)<br>0035h = CFI Minor Version 5 (Model Number is 01, 02, V1, and V2) |

**Table 29 CFI primary vendor-specific extended query** (Continued)

| Word address | Byte address | Data | Description |
|---|---|---|---|
| (SA) + 0045h | (SA) + 008Ah | 0024h | Address Sensitive Unlock (Bits 1–0)<br>00b = Required<br>01b = Not required<br>Process technology (Bits 5–2)<br>0000b = 0.23 µm Floating Gate<br>0001b = 0.17 µm Floating Gate<br>0010b = 0.23 µm MIRRORBIT™<br>0011b = 0.13 µm Floating Gate<br>0100b = 0.11 µm MIRRORBIT™<br>0101b = 0.09 µm MIRRORBIT™<br>0110b = 0.09 µm Floating Gate<br>0111b = 0.065 µm MIRRORBIT™ Eclipse<br>1000b = 0.065 µm MIRRORBIT™<br>1001b = 0.045 µm MIRRORBIT™ |
| (SA) + 0046h | (SA) + 008Ch | 0002h | Erase Suspend<br>0 = Not supported<br>1 = Read only<br>2 = Read and write |
| (SA) + 0047h | (SA) + 008Eh | 0001h | Sector Protect<br>00 = Not supported<br>X = Number of sectors in smallest group |
| (SA) + 0048h | (SA) + 0090h | 0000h | Temporary Sector Unprotect<br>00 = Not supported<br>01 = Supported |
| (SA) + 0049h | (SA) + 0092h | 0008h | Sector Protect/Unprotect Scheme<br>04 = High Voltage method<br>05 = Software Command Locking method<br>08 = Advanced Sector Protection method |
| (SA) + 004Ah | (SA) + 0094h | 0000h | Simultaneous Operation<br>00 = Not supported<br>X = Number of banks |
| (SA) + 004Bh | (SA) + 0096h | 0000h | Burst Mode type<br>00 = Not supported<br>01 = Supported |
| (SA) + 004Ch | (SA) + 0098h | 0003h | Page Mode type<br>00 = Not supported<br>01 = 4 Word Page<br>02 = 8 Word Page<br>03 =16 Word Page |
| (SA) + 004Dh | (SA) + 009Ah | 00B5h | ACC (Acceleration) Supply Minimum<br>00 = Not supported<br>D7–D4: Volt<br>D3–D0: 100 mV |
| (SA) + 004Eh | (SA) + 009Ch | 00C5h | ACC (Acceleration) Supply Maximum<br>00 = Not supported<br>D7–D4: Volt<br>D3–D0: 100 mV |
| (SA) + 004Fh | (SA) + 009Eh | 0004h (Bottom)<br>0005h (Top) | WP# Protection<br>00h = Flash device without WP Protect (No Boot)<br>01h = Eight 8 KB Sectors at TOP and Bottom with WP (Dual Boot)<br>02h = Bottom Boot Device with WP Protect (Bottom Boot)<br>03h = Top Boot Device with WP Protect (Top Boot)<br>04h = Uniform, Bottom WP Protect (Uniform Bottom Boot)<br>05h = Uniform, Top WP Protect (Uniform Top Boot)<br>06h = WP Protect for all sectors<br>07h = Uniform, Top and Bottom WP Protect |

**Table 29 CFI primary vendor-specific extended query** (Continued)

| Word address | Byte address | Data | Description |
|---|---|---|---|
| (SA) + 0050h | (SA) + 00A0h | 0001h | Program Suspend<br>00 = Not supported<br>01 = Supported |
| **Below Queries Only available for CFI Version 1.5** | | | |
| (SA) +0051h | (SA) +00A2h | 0001h | Unlock Bypass<br>00 = Not supported<br>01 = Supported |
| (SA) + 0052h | (SA) + 00A4h | 0009h | Secured Silicon Sector (Customer OTP Area) Size $2^N$ (bytes) |
| (SA) + 0053h | (SA) + 00A6h | 008Fh | Software Features<br>bit 0: status register polling (1 = supported, 0 = not supported)<br>bit 1: DQ polling (1 = supported, 0 = not supported)<br>bit 2: new program suspend/resume commands (1 = supported, 0 = not supported)<br>bit 3: word programming (1 = supported, 0 = not supported)<br>bit 4: bit-field programming (1 = supported, 0 = not supported)<br>bit 5: autodetect programming (1 = supported, 0 = not supported)<br>bit 6: RFU<br>bit 7: multiple writes per Line (1 = supported, 0 = not supported) |
| (SA) + 0054h | (SA) + 00A8h | 0005h | Page Size = $2^N$ bytes |
| (SA) + 0055h | (SA) + 00AAh | 0006h | Erase Suspend Timeout Maximum < $2^N$ (µs) |
| (SA) + 0056h | (SA) + 00ACh | 0006h | Program Suspend Timeout Maximum < $2^N$ (µs) |
| (SA) + 0057h to (SA) + 0077h | (SA) + 00AEh to (SA) + 00ACh | FFFFh | Reserved |
| (SA) + 0078h | (SA) + 00F0h | 0006h | Embedded Hardware Reset Timeout Maximum < $2^N$ (µs)<br>Reset with Reset Pin |
| (SA) + 0079h | (SA) + 00F2h | 0009h | Non-Embedded Hardware Reset Timeout Maximum < $2^N$ (µs)<br>Power-on-Reset |

# 8 Signal descriptions

## 8.1 Address and data configuration

Address and data are connected in parallel (ADP) via separate signal inputs and I/Os.

## 8.2 Input/Output summary

**Table 30 I/O summary**

| Symbol | Type | Description |
|---|---|---|
| RESET# | Input | Hardware Reset. At $V_{IL}$, causes the device to reset control logic to its standby state, ready for reading array data. |
| CE# | Input | Chip Enable. At $V_{IL}$, selects the device for data transfer with the host memory controller. |
| OE# | Input | Output Enable. At $V_{IL}$, causes outputs to be actively driven. At $V_{IH}$, causes outputs to be high impedance (High-Z). |
| WE# | Input | Write Enable. At $V_{IL}$, indicates data transfer from host to device. At $V_{IH}$, indicates data transfer is from device to host. |
| Amax–A0 | Input | Address inputs.<br>A25–A0 for S29GL01GT<br>A24–A0 for S29GL512T |
| DQ14–DQ0 | Input/Output | Data inputs and outputs |
| DQ15/A1 | Input/Output | DQ15: Data inputs and outputs<br>A1: LSB address input in byte mode |
| WP#/ACC | Input | Write Protect. At $V_{IL}$, disables program and erase functions in the lowest or highest address 64-Kword (128-KB) sector of the device. At $V_{IH}$, the sector is not protected. At $V_{HH}$, automatically places device in unlock bypass mode. WP# has an internal pull up; When unconnected WP# is at $V_{IH}$. |
| RY/BY# | Output – open drain | Ready/Busy. Indicates whether an Embedded Algorithm is in progress or complete. At $V_{IL}$, the device is actively engaged in an Embedded Algorithm such as erasing or programming. At High-Z, the device is ready for read or a new command write - requires external pull-up resistor to detect the High-Z state. Multiple devices may have their RY/BY# outputs tied together to detect when all devices are ready. |
| BYTE# | Input | Selects data bus width. At $V_{IL}$, the device is in byte configuration and data I/O pins DQ7–DQ0 are active and DQ15/A1 becomes the LSB address input. At $V_{IH}$, the device is in word configuration and data I/O pins DQ15–DQ0 are active. |
| $V_{CC}$ | Power Supply | Core power supply |
| $V_{IO}$ | Power Supply | Versatile I/O power supply. |
| $V_{SS}$ | Power Supply | Power supplies ground |
| NC | No Connect | Not Connected internally. The pin/ball location may be used in Printed Circuit Board (PCB) as part of a routing channel. |
| RFU | No Connect | Reserved for Future Use. Not currently connected internally but the pin/ball location should be left unconnected and unused by PCB routing channel for future compatibility. The pin/ball may be used by a signal in the future. |
| DNU | Reserved | Do Not Use. Reserved for use by Infineon. The pin/ball is connected internally. The input has an internal pull down resistance to $V_{SS}$. The pin/ball can be left open or tied to $V_{SS}$ on the PCB. |

## 8.3 Word/Byte configuration

The BYTE# pin controls whether the device data I/O pins operate in the byte or word configuration. If the BYTE# pin is set at logic 1, the device is in word configuration, DQ0–DQ15 are active and controlled by CE# and OE#.

If the BYTE# pin is set at logic 0, the device is in byte configuration, and only data I/O pins DQ0–DQ7 are active and controlled by CE# and OE#. The data I/O pins DQ8–DQ14 are tri-stated, and the DQ15 pin is used as an input for the LSB (A1) address function.

The BYTE# pin can only be switch while the device is in standby (read mode).

The BYTE# pin has an internal pull-up. Though not required in a ×16 only system, the pin should be connected to high (e.g. $V_{IO}$)

## 8.4 Versatile I/O feature

The maximum output voltage level driven by, and input levels acceptable to, the device are determined by the $V_{IO}$ power supply. This supply allows the device to drive and receive signals to and from other devices on the same bus having interface signal levels different from the device core voltage.

## 8.5 Ready/Busy# (RY/BY#)

RY/BY# is a dedicated, open drain output pin that indicates whether an EA, POR, or hardware reset is in progress or complete. The RY/BY# status is valid after the rising edge of the final WE# pulse in a command sequence, when $V_{CC}$ is above $V_{CC}$ minimum during POR, or after the falling edge of RESET#. Since RY/BY# is an open drain output, several RY/BY# pins can be tied together in parallel with a pull up resistor to $V_{IO}$.

If the output is LOW (Busy), the device is actively erasing, programming, or resetting (This includes programming in the Erase Suspend mode). If the output is HIGH (Ready), the device is ready to read data (including during the erase suspend mode), or is in the standby mode. **Table 17** shows the outputs for RY/BY# in each operation.

If EA has failed (Program/erase failure as result of max pulses or program abort), RY/BY# will stay LOW (busy) until status register bits 4 and 5 are cleared and the reset command is issued. If EA has failed (Sector is locked), RY/BY# will return HIGH (ready). This includes erase or programming on a locked sector.

## 8.6 Hardware reset

The RESET# input provides a hardware method of resetting the device to standby state. When RESET# is driven LOW for at least a period of $t_{RP}$, the device immediately:

- terminates any operation in progress,
- exits any ASO,
- tristates all outputs,
- resets the Status Register,
- resets the EAC to standby state.
- CE# is ignored for the duration of the reset operation ($t_{RPH}$).
- To meet the Reset current specification ($I_{CC5}$) CE# must be held HIGH.

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

# 9 Signal protocols

The following sections describe the host system interface signal behavior and timing for the 29GL-T family flash devices.

## 9.1 Interface states

**Table 31** describes the required value of each interface signal for each interface state.

**Table 31 Interface states**

| Interface State | $V_{CC}$ | $V_{IO}$ | RESET# | CE# | OE# | WE# | BYTE#[97] | WP#/ACC | Amax–A0[92] | DQ0–DQ7 | DQ8-DQ15 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | BYTE# = $V_{IH}$ | BYTE = $V_{IL}$ |
| Power-Off with Hardware Data Protection | $< V_{LKO}$ | $\leq V_{CC}$ | X | X | X | X | L or H | X | X | High-Z | High-Z | High-Z |
| Power-On (Cold) Reset | $\geq V_{CC}$ min | $\geq V_{IO}$ min $\leq V_{CC}$ | X | X | X | X | L or H | X | X | High-Z | High-Z | High-Z |
| Hardware (Warm) Reset | $\geq V_{CC}$ min | $\geq V_{IO}$ min $\leq V_{CC}$ | L | X | X | X | L or H | X | X | High-Z | High-Z | High-Z |
| Interface Standby | $\geq V_{CC}$ min | $\geq V_{IO}$ min $\leq V_{CC}$ | H | H | X | X | L or H | H | X | High-Z | High-Z | High-Z |
| Automatic Sleep[93, 95] | $\geq V_{CC}$ min | $\geq V_{IO}$ min $\leq V_{CC}$ | H | L | X | X | L or H | H | Valid | Output Available | Output Available | DQ8–DQ14 = High-Z, DQ15 = A1 |
| Read with Output Disable[94] | $\geq V_{CC}$ min | $\geq V_{IO}$ min $\leq V_{CC}$ | H | L | H | H | L or H | X | Valid | High-Z | High-Z | High-Z |
| Random Read | $\geq V_{CC}$ min | $\geq V_{IO}$ min | H | L | L | H | | X | Valid | Output Valid | Output Valid | DQ8–DQ14 = High-Z, DQ15 = A1 |
| Page Read | $\geq V_{CC}$ min | $\geq V_{IO}$ min $\leq V_{CC}$ | H | L | L | H | L or H | X | Amax–A4 Valid A3–A0 (or A3–A1) Modified | Output Valid | Output Valid | DQ8–DQ14 = High-Z, DQ15 = A1 |
| Write | $\geq V_{CC}$ min | $\geq V_{IO}$ min $\leq V_{CC}$ | H | L | H | L | L or H | Note [96] | Valid | Input Valid | Input Valid | DQ8–DQ14 = High-Z, DQ15 = A1 |

**Legend:**
L = $V_{IL}$
H = $V_{IH}$
X = either $V_{IL}$ or $V_{IH}$
L/H = rising edge
H/L = falling edge
Valid = all bus signals have stable L or H level
Modified = valid state different from a previous valid state
Available = read data is internally stored with output driver controlled by OE#

**Notes**
92. Address are Amax:A0 in word mode; Amax:A1 in byte mode.
93. WE# and OE# can not be at $V_{IL}$ at the same time.
94. Read with Output Disable is a read initiated with OE# HIGH.
95. Automatic Sleep is a read/write operation where data has been driven on the bus for an extended period, without CE# going HIGH and the device internal logic has gone into standby mode to conserve power.
96. If WP# = $V_{IL}$, on the outermost sector remains protected. If WP# = $V_{IH}$, the outermost sector is unprotected. WP# has an internal pull-up; when unconnected, WP# is at $V_{IH}$.
97. $V_{IL} = V_{SS}$ and $V_{IH} = V_{IO}$.

## 9.2 Power-off with hardware data protection

The memory is considered to be powered off when the core power supply ($V_{CC}$) drops below the lock-out voltage ($V_{LKO}$). When $V_{CC}$ is below $V_{LKO}$, the entire memory array is protected against a program or erase operation. This ensures that no spurious alteration of the memory content can occur during power transition. During a power supply transition down to power-off, $V_{IO}$ should remain less than or equal to $V_{CC}$.

If $V_{CC}$ goes below $V_{RST}$ (Min) then returns above $V_{RST}$ (Min) to $V_{CC}$ minimum, the POR interface state is entered and the EAC starts the cold reset embedded algorithm.

## 9.3 Power Conservation modes

### 9.3.1 Interface Standby

Standby is the default, low power, state for the interface while the device is not selected by the host for data transfer (CE# = HIGH). All inputs are ignored in this state and all outputs except RY/BY# are high impedance. RY/BY# is a direct output of the EAC, not controlled by the host interface.

### 9.3.2 Automatic Sleep

The automatic sleep mode reduces device interface energy consumption to the sleep level ($I_{CC6}$) following the completion of a random read access time. The device automatically enables this mode when addresses remain stable for $t_{ACC}$ + 30 ns. While in automatic sleep mode, output data is latched and always available to the system. Output of the data depends on the level of the OE# signal but, the automatic sleep mode current is independent of the OE# signal level. Standard address access timings ($t_{ACC}$ or $t_{PACC}$) provide new data when addresses are changed. Refer the **"DC characteristics"** on page 79 for the automatic sleep mode current specification $I_{CC6}$.

Automatic sleep helps reduce current consumption especially when the host system clock is slowed for power reduction. During slow system clock periods, read and write cycles may extend many times their length versus when the system is operating at high speed. Even though CE# may be LOW throughout these extended data transfer cycles, the memory device host interface will go to the Automatic Sleep current at $t_{ACC}$ + 30 ns. The device will remain at the Automatic Sleep current for $t_{ASSB}$. Then the device will transition to the standby current level. This keeps the memory at the Automatic Sleep or standby power level for most of the long duration data transfer cycles, rather than consuming full read power all the time that the memory device is selected by the host system.

However, the EAC operates independent of the automatic sleep mode of the host interface and will continue to draw current during an active EA. Only when both the host interface and EAC are in their standby states is the standby level current achieved.

## 9.4 Read

### 9.4.1 Read with output disable

When the CE# signal is asserted LOW, the host system memory controller begins a read or write data transfer. Often there is a period at the beginning of a data transfer when CE# is LOW, Address is valid, OE# is HIGH, and WE# is HIGH. During this state a read access is assumed and the random read process is started while the data outputs remain at high impedance. If the OE# signal goes LOW, the interface transitions to the random read state, with data outputs actively driven. If the WE# signal is asserted LOW, the interface transitions to the write state. Note, OE# and WE# should never be LOW at the same time to ensure no data bus contention between the host system and memory.

### 9.4.2 Random (Asynchronous) read

When the host system interface selects the memory device by driving CE# LOW, the device interface leaves the Standby state. If WE# is HIGH when CE# goes LOW, a random read access is started. The data output depends on the address map mode and the address provided at the time the read access is started.

The data appears on DQ15–DQ0 (DQ7–DQ0 in ×8 mode) when CE# is LOW, OE# is LOW, WE# remains HIGH, address remains stable, and the asynchronous access times are satisfied. Address access time ($t_{ACC}$) is equal to the delay from stable addresses to valid output data. The chip enable access time ($t_{CE}$) is the delay from stable CE# to valid data at the outputs. In order for the read data to be driven on to the data outputs the OE# signal must be LOW at least the output enable time ($t_{OE}$) before valid data is available.

At the completion of the random access time from CE# active ($t_{CE}$), address stable ($t_{ACC}$), or OE# active ($t_{OE}$), whichever occurs latest, the data outputs will provide valid read data from the currently active address map mode. If CE# remains LOW and any of the Amax to A4 address signals change to a new value, a new random read access begins. If CE# remains LOW and OE# goes HIGH the interface transitions to the read with output disable state. If CE# remains LOW, OE# goes HIGH, and WE# goes LOW, the interface transitions to the Write state. If CE# returns HIGH, the interface goes to the Standby state. Back to back accesses, in which CE# remains LOW between accesses, requires an address change to initiate the second access. See **“Asynchronous read operations”** on page 89.

### 9.4.3 Page read

After a random read access is completed, if CE# remains LOW, OE# remains LOW, the Amax to A4 address signals remain stable, and any of the A3 to A0 address signals change, a new access within the same page begins. In ×8 mode, when any of the A3 to A1 address signals change, a new access within the same page begins. The page read completes much faster ($t_{PACC}$) than a random read access.

## 9.5 Write

### 9.5.1 Asynchronous write

When WE# goes LOW after CE is LOW, there is a transition from one of the read states to the write state. If WE# is LOW before CE# goes LOW, there is a transition from the Standby state directly to the write state without beginning a read access.

When CE# is LOW, OE# is HIGH, and WE# goes LOW, a write data transfer begins. Note, OE# and WE# should never be LOW at the same time to ensure no data bus contention between the host system and memory. When the asynchronous write cycle timing requirements are met the WE# can go HIGH to capture the address and data values in to EAC command memory.

Address is captured by the falling edge of WE# or CE#, whichever occurs later. Data is captured by the rising edge of WE# or CE#, whichever occurs earlier.

When CE# is LOW before WE# goes LOW and stays LOW after WE# goes HIGH, the access is called a WE# controlled write. When WE# is HIGH and CE# goes HIGH, there is a transition to the Standby state. If CE# remains LOW and WE# goes HIGH, there is a transition to the read with output disable state.

When WE# is LOW before CE# goes LOW and remains LOW after CE# goes HIGH, the access is called a CE# controlled write. A CE# controlled write transitions to the Standby state.

If WE# is LOW before CE# goes LOW, the write transfer is started by CE# going LOW. If WE# is LOW after CE# goes HIGH, the address and data are captured by the rising edge of CE#. These cases are referred to as CE# controlled write state transitions.

Write followed by read accesses, in which CE# remains LOW between accesses, requires an address change to initiate the following read access.

Back to back accesses, in which CE# remains LOW between accesses, requires an address change to initiate the second access.

The EAC command memory array is not readable by the host system and has no ASO. 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.

### 9.5.2 Write Pulse “Glitch” protection

Noise pulses of less than 5 ns (typical) on WE# will not initiate a write cycle.

### 9.5.3 Logical inhibit

Write cycles are inhibited by holding OE# at $V_{IL}$, or CE# at $V_{IH}$, or WE# at $V_{IH}$. To initiate a write cycle, CE# and WE# must be LOW ($V_{IL}$) while OE# is HIGH ($V_{IH}$).

# 10 Electrical specifications

## 10.1 Absolute maximum ratings

**Table 32 Absolute maximum ratings**

| | |
|---|---|
| Storage temperature plastic packages | –65°C to +150°C |
| Ambient temperature with power applied | –65°C to +125°C |
| **Voltage with respect to ground** | |
| All pins other than RESET#[98] | –0.5 V to ($V_{IO}$ + 0.5 V) |
| RESET#[98] | –0.5 V to ($V_{CC}$ + 0.5 V) |
| Output short circuit current[99] | 100 mA |
| $V_{CC}$ | –0.5 V to +4.0 V |
| $V_{IO}$ | –0.5 V to +4.0 V |
| ACC | –0.5 V to +12.5 V |

## 10.2 Thermal resistance

**Table 33 Thermal resistance**

| Parameter | Description | Test condition | Device | TS056 | LAE064 | LAA064 | VBU056 | Unit |
|---|---|---|---|---|---|---|---|---|
| Theta JA | Thermal resistance (Junction to ambient) | Test conditions follow standard test methods and procedures for measuring thermal impedance in accordance with EIA/JESD51. with Still Air (0 m/s). | 1G | 43.5 | 30 | 24 | 30.5 | °C/W |
| | | | 512M | 45 | 32 | 26 | 33 | °C/W |
| Theta JB | Thermal resistance (Junction to board) | | 1G | 40.5 | 8.4 | 10.5 | 8.3 | °C/W |
| | | | 512M | 14 | 12.4 | 11.8 | 10.4 | °C/W |
| Theta JC | Thermal resistance (Junction to case) | | 1G | 20 | 7.5 | 6.7 | 8.1 | °C/W |
| | | | 512M | 19.5 | 10.1 | 7.3 | 10 | °C/W |

## 10.3 Latchup characteristics

This product complies with JEDEC standard JESD78C latch-up testing requirements.

**Notes**

98. Minimum DC voltage on input or I/O pins is –0.5 V. During voltage transitions, input or I/O pins may undershoot $V_{SS}$ to –2.0 V for periods of up to 20 ns. See **Figure 11**. Maximum DC voltage on input or I/O pins is $V_{CC}$ + 0.5 V. During voltage transitions, input or I/O pins may overshoot to $V_{CC}$ + 2.0 V for periods up to 20 ns. See **Figure 12**.
99. No more than one output may be shorted to ground at a time. Duration of the short circuit should not be greater than one second.
100. Stresses above those listed under **Absolute maximum ratings** may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational sections of this data sheet is not implied. Exposure of the device to absolute maximum rating conditions for extended periods may affect device reliability.

## 10.4 Operating ranges

### 10.4.1 Temperature ranges

**Table 34 Temperature ranges**

| Parameter | Symbol | Devices | Spec | | Unit |
|---|---|---|---|---|---|
| | | | Min | Max | |
| Ambient temperature | $T_A$ | Industrial (I) | –40 | +85 | °C |
| | | Industrial Plus (V) | –40 | +105 | |
| | | Extended (N) | –40 | +125 | |
| | | Automotive, AEC-Q100 grade 3 (A) | –40 | +85 | |
| | | Automotive, AEC-Q100 grade 2 (B) | –40 | +105 | |

### 10.4.2 Power supply voltages

**Table 35 Power supply voltages**

| | |
|---|---|
| $V_{CC}$ | 2.7 V to 3.6 V |
| $V_{IO}$ | 1.65 V to $V_{CC}$ + 200 mV |

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

### 10.4.3 Power-up and power-down

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

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

During power-down or voltage drops below $V_{CC}$ Lockout maximum ($V_{LKO}$), the $V_{CC}$ and $V_{IO}$ voltages must drop below $V_{CC}$ Reset ($V_{RST}$) minimum for a period of $t_{PD}$ for the part to initialize correctly when $V_{CC}$ and $V_{IO}$ again rise to their operating ranges. See **Figure 10**. If during a voltage drop the $V_{CC}$ stays above $V_{LKO}$ maximum the part will stay initialized and will work correctly when $V_{CC}$ is again above $V_{CC}$ minimum. If the part locks up from improper initialization, a hardware reset can be used to initialize the part correctly.

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

**Table 36 Power-up/power-down voltage and timing**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $V_{CC}$ | $V_{CC}$ power supply | 2.7 | 3.6 | V |
| $V_{LKO}$ | $V_{CC}$ level below which re-initialization is required[102] | – | 2.5 | V |
| $V_{RST}$ | $V_{CC}$ and $V_{IO}$ Low voltage needed to ensure initialization will occur[102] | 1.0 | – | V |
| $t_{VCS}$ | $V_{CC}$ and $V_{IO} \geq$ minimum to first access[102] | 300 | – | µs |
| $t_{PD}$ | Duration of $V_{CC} \leq V_{RST}$(min)[102] | 15 | – | µs |

**Note**
102.Not 100% tested.

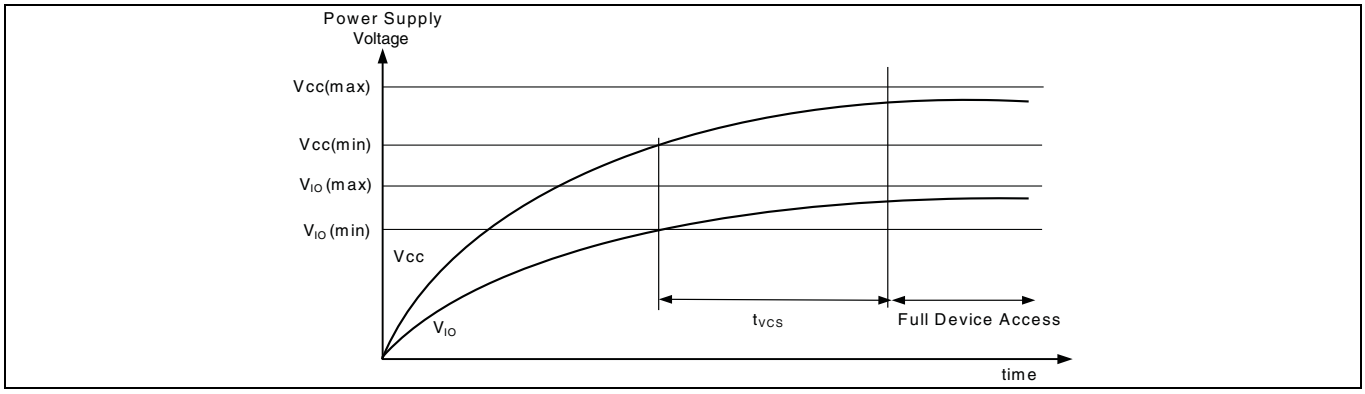



**Figure 9 Power-up**

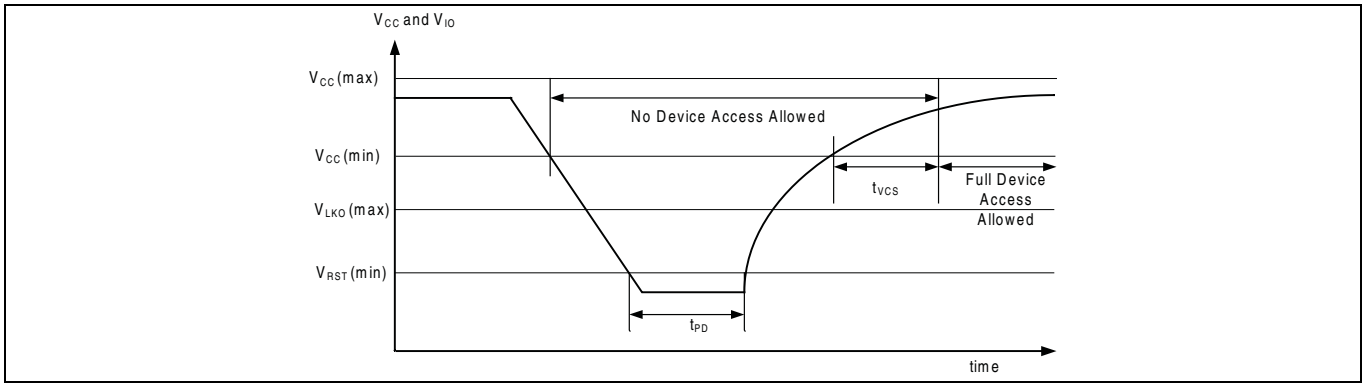



**Figure 10 Power-down and voltage drop**

## 10.4.4 Input signal overshoot

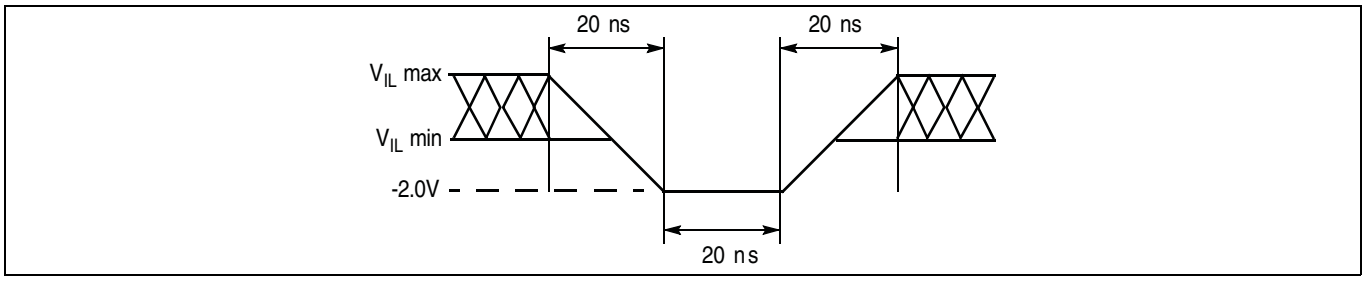



**Figure 11 Maximum negative overshoot waveform**

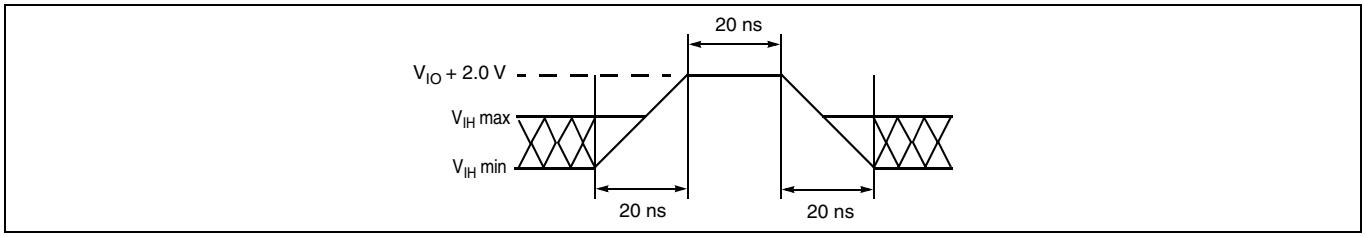



**Figure 12 Maximum positive overshoot waveform**

## 10.5 DC characteristics

**Table 37 DC characteristics (–40°C to +85°C)**

| Parameter | Description | Test conditions | | Min | Typ[103] | Max | Unit |
|---|---|---|---|---|---|---|---|
| $I_{LI}$ | Input load current | $V_{IN} = V_{SS}$ to $V_{CC}$, $V_{CC} = V_{CC}$ max | All others | – | ±0.02 | ±1.0 | µA |
| | | | WP#, BYTE# | – | ±0.5 | ±2.0 | |
| $I_{LO}$ | Output leakage current | $V_{OUT} = V_{SS}$ to $V_{CC}$, $V_{CC} = V_{CC}$ max | | – | ±0.02 | ±1.0 | µA |
| $I_{CC1}$ | $V_{CC}$ active read current | CE# = $V_{IL}$, OE# = $V_{IH}$, address switching @ 5 MHz, $V_{CC} = V_{CC}$ max | | – | 55 | 60 | mA |
| $I_{CC2}$ | $V_{CC}$ intra-page read current | CE# = $V_{IL}$, OE# = $V_{IH}$, address switching @ 33 MHz, $V_{CC} = V_{CC}$ max | | – | 9 | 25 | mA |
| $I_{CC3}$ | $V_{CC}$ active erase/program current[103, 104] | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC}$ max | | – | 45 | 100 | mA |
| $I_{CC4}$ | $V_{CC}$ standby current | CE#, RESET#, OE# = $V_{IH}$, $V_{IH} = V_{IO}$, $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ max | | – | 70 | 100 | µA |
| $I_{CC5}$ | $V_{CC}$ reset current[103, 105] | CE# = $V_{IH}$, RESET# = $V_{IL}$, $V_{CC} = V_{CC}$ max | | – | 10 | 20 | mA |
| $I_{CC6}$ | Automatic sleep mode[106] | $V_{IH} = V_{IO}$, $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ max, $t_{ACC}$ + 30 ns | | – | 3 | 6 | mA |
| | | $V_{IH} = V_{IO}$, $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ max, $t_{ASSB}$ | | – | 100 | 150 | µA |
| $I_{CC7}$ | $V_{CC}$ current during power-up[103, 107] | RESET# = $V_{IO}$, CE# = $V_{IO}$, OE# = $V_{IO}$, $V_{CC} = V_{CC}$ max | | – | 53 | 80 | mA |
| $V_{IL}$ | Input low voltage[108] | – | | –0.5 | – | 0.3 × $V_{IO}$ | V |
| $V_{IH}$ | Input high voltage[108] | – | | 0.7 × $V_{IO}$ | – | $V_{IO}$ + 0.4 | V |
| $V_{HH}$ | Voltage for ACC program acceleration | $V_{CC}$ = 2.7 V–3.6 V | | 11.5 | – | 12.5 | V |
| $V_{OL}$ | Output low voltage[108, 110] | $I_{OL}$ = 100 µA for DQ15–DQ0; $I_{OL}$ = 2 mA for RY/BY# | | – | – | 0.15 × $V_{IO}$ | V |
| $V_{OH}$ | Output high voltage[108] | $I_{OH}$ = 100 µA | | 0.85 × $V_{IO}$ | – | – | V |
| $V_{LKO}$ | Low $V_{CC}$ lock-out voltage[103] | – | | 2.25 | – | 2.5 | V |
| $V_{RST}$ | Low $V_{CC}$ power-on-reset voltage[103] | – | | – | 1.0 | – | V |

**Notes**

103. Not 100% tested.
104. $I_{CC}$ active while EA is in progress.
105. If an embedded operation is in progress at the start of reset, the current consumption will remain at the embedded operation specification until the embedded operation is stopped by the reset. If no embedded operation is in progress when reset is started, or following the stopping of an embedded operation, $I_{CC5}$ will be drawn during the remainder of $t_{RPH}$. After the end of $t_{RPH}$ the device will go to standby mode until the next read or write.
106. Automatic sleep mode enables the lower power mode when addresses remain stable for the specified designated time.
107. During power-up there are spikes of current demand, the system needs to be able to supply this current to insure the part initializes correctly.
108. $V_{IO}$ = 1.65 V to $V_{CC}$ or 2.7 V to $V_{CC}$ depending on the model.
109. $V_{CC}$ = 3 V and $V_{IO}$ = 3 V or 1.8 V. When $V_{IO}$ is at 1.8 V, I/O pins cannot operate at >1.8 V.
110. The recommended pull-up resistor for RY/BY# output is 5k to 10k ohms.

**Table 38 DC characteristics (–40°C to +105°C)**

| Parameter | Description | Test conditions | | Min | Typ[111] | Max | Unit |
|---|---|---|---|---|---|---|---|
| $I_{LI}$ | Input load current | $V_{IN} = V_{SS}$ to $V_{CC}$, $V_{CC} = V_{CC}$ max | All others | – | ±0.02 | ±1.0 | µA |
| | | | WP#, BYTE# | – | ±0.5 | ±2.0 | |
| $I_{LO}$ | Output leakage current | $V_{OUT} = V_{SS}$ to $V_{CC}$, $V_{CC} = V_{CC}$ max | | – | ±0.02 | ±1.0 | µA |
| $I_{CC1}$ | $V_{CC}$ active read current | CE# = $V_{IL}$, OE# = $V_{IH}$, address switching @ 5 MHz, $V_{CC} = V_{CC}$ max | | – | 55 | 60 | mA |
| $I_{CC2}$ | $V_{CC}$ intra-page read current | CE# = $V_{IL}$, OE# = $V_{IH}$, address switching @ 33 MHz, $V_{CC} = V_{CC}$ max | | – | 9 | 25 | mA |
| $I_{CC3}$ | $V_{CC}$ active erase/program current[111, 112] | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC}$ max | | – | 45 | 100 | mA |
| $I_{CC4}$ | $V_{CC}$ standby current | CE#, RESET#, OE# = $V_{IH}$, $V_{IH} = V_{IO}$, $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ max | | – | 70 | 200 | µA |
| $I_{CC5}$ | $V_{CC}$ reset current[111, 113] | CE# = $V_{IH}$, RESET# = $V_{IL}$, $V_{CC} = V_{CC}$ max | | – | 10 | 20 | mA |
| $I_{CC6}$ | Automatic sleep mode[114] | $V_{IH} = V_{IO}$, $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ max, $t_{ACC}$ + 30 ns | | – | 3 | 6 | mA |
| | | $V_{IH} = V_{IO}$, $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ max, $t_{ASSB}$ | | – | 100 | 200 | µA |
| $I_{CC7}$ | $V_{CC}$ current during power-up[111, 115] | RESET# = $V_{IO}$, CE# = $V_{IO}$, OE# = $V_{IO}$, $V_{CC} = V_{CC}$ max | | – | 53 | 80 | mA |
| $V_{IL}$ | Input low voltage[116] | | | –0.5 | – | 0.3 × $V_{IO}$ | V |
| $V_{IH}$ | Input high voltage[116] | | | 0.7 × $V_{IO}$ | – | $V_{IO}$ + 0.4 | V |
| $V_{HH}$ | Voltage for ACC program acceleration | $V_{CC}$ = 2.7 V–3.6 V | | 11.5 | – | 12.5 | V |
| $V_{OL}$ | Output low voltage[116, 118] | $I_{OL}$ = 100 µA for DQ15–DQ0; $I_{OL}$ = 2 mA for RY/BY# | | – | – | 0.15 × $V_{IO}$ | V |
| $V_{OH}$ | Output high voltage[116] | $I_{OH}$ = 100 µA | | 0.85 × $V_{IO}$ | – | – | V |
| $V_{LKO}$ | Low $V_{CC}$ lock-out voltage[111] | | | 2.25 | – | 2.5 | V |
| $V_{RST}$ | Low $V_{CC}$ power-on-reset voltage[111] | | | – | 1.0 | – | V |

**Notes**

111. Not 100% tested.
112. $I_{CC}$ active while EA is in progress.
113. If an embedded operation is in progress at the start of reset, the current consumption will remain at the embedded operation specification until the embedded operation is stopped by the reset. If no embedded operation is in progress when reset is started, or following the stopping of an embedded operation, $I_{CC7}$ will be drawn during the remainder of $t_{RPH}$. After the end of $t_{RPH}$ the device will go to standby mode until the next read or write.
114. Automatic sleep mode enables the lower power mode when addresses remain stable for the specified designated time.
115. During power-up there are spikes of current demand, the system needs to be able to supply this current to insure the part initializes correctly.
116. $V_{IO}$ = 1.65 V to $V_{CC}$ or 2.7 V to $V_{CC}$ depending on the model.
117. $V_{CC}$ = 3 V and $V_{IO}$ = 3 V or 1.8 V. When $V_{IO}$ is at 1.8 V, I/O pins cannot operate at >1.8 V.
118. The recommended pull-up resistor for RY/BY# output is 5k to 10k ohms.

**Table 39 DC characteristics (–40°C to +125°C)**

| Parameter | Description | Test conditions | | Min | Typ[119] | Max | Unit |
|---|---|---|---|---|---|---|---|
| $I_{LI}$ | Input load current | $V_{IN} = V_{SS}$ to $V_{CC}$, $V_{CC} = V_{CC}$ max | All others | – | ±0.02 | ±1.0 | µA |
| | | | WP#, BYTE# | – | ±0.5 | ±2.0 | |
| $I_{LO}$ | Output leakage current | $V_{OUT} = V_{SS}$ to $V_{CC}$, $V_{CC} = V_{CC}$ max | | – | ±0.02 | ±1.0 | µA |
| $I_{CC1}$ | $V_{CC}$ active read current | CE# = $V_{IL}$, OE# = $V_{IH}$, address switching @ 5 MHz, $V_{CC} = V_{CC}$ max | | – | 55 | 60 | mA |
| $I_{CC2}$ | $V_{CC}$ intra-page read current | CE# = $V_{IL}$, OE# = $V_{IH}$, Address switching @ 33 MHz, $V_{CC} = V_{CC}$ max | | – | 9 | 25 | mA |
| $I_{CC3}$ | $V_{CC}$ active erase/program current[119, 120] | CE# = $V_{IL}$, OE# = $V_{IH}$, $V_{CC} = V_{CC}$ max | | – | 45 | 100 | mA |
| $I_{CC4}$ | $V_{CC}$ standby current | CE#, RESET#, OE# = $V_{IH}$, $V_{IH} = V_{IO}$, $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ max | | – | 70 | 215 | µA |
| $I_{CC5}$ | $V_{CC}$ reset current[119, 121] | CE# = $V_{IH}$, RESET# = $V_{IL}$, $V_{CC} = V_{CC}$ max | | – | 10 | 20 | mA |
| $I_{CC6}$ | Automatic sleep mode[122] | $V_{IH} = V_{IO}$, $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ max, $t_{ACC}$ + 30 ns | | – | 3 | 6 | mA |
| | | $V_{IH} = V_{IO}$, $V_{IL} = V_{SS}$, $V_{CC} = V_{CC}$ max, $t_{ASSB}$ | | – | 100 | 215 | µA |
| $I_{CC7}$ | $V_{CC}$ current during power-up[119, 123] | RESET# = $V_{IO}$, CE# = $V_{IO}$, OE# = $V_{IO}$, $V_{CC} = V_{CC}$ max | | – | 53 | 80 | mA |
| $V_{IL}$ | Input low voltage[124] | – | | –0.5 | – | 0.3 × $V_{IO}$ | V |
| $V_{IH}$ | Input high voltage[124] | – | | 0.7 × $V_{IO}$ | – | $V_{IO}$ + 0.4 | V |
| $V_{HH}$ | Voltage for ACC program acceleration | $V_{CC}$ = 2.7 V–3.6 V | | 11.5 | – | 12.5 | V |
| $V_{OL}$ | Output low voltage[124, 126] | $I_{OL}$ = 100 µA for DQ15–DQ0; $I_{OL}$ = 2 mA for RY/BY# | | – | – | 0.15 × $V_{IO}$ | V |
| $V_{OH}$ | Output high voltage[124] | $I_{OH}$ = 100 µA | | 0.85 × $V_{IO}$ | – | – | V |
| $V_{LKO}$ | Low $V_{CC}$ lock-out voltage[119] | – | | 2.25 | – | 2.5 | V |
| $V_{RST}$ | Low $V_{CC}$ power-on-reset voltage[119] | – | | – | 1.0 | – | V |

**Notes**

119. Not 100% tested.
120. $I_{CC}$ active while EA is in progress.
121. If an embedded operation is in progress at the start of reset, the current consumption will remain at the embedded operation specification until the embedded operation is stopped by the reset. If no embedded operation is in progress when reset is started, or following the stopping of an embedded operation, $I_{CC7}$ will be drawn during the remainder of $t_{RPH}$. After the end of $t_{RPH}$ the device will go to standby mode until the next read or write.
122. Automatic sleep mode enables the lower power mode when addresses remain stable for the specified designated time.
123. During power-up there are spikes of current demand, the system needs to be able to supply this current to insure the part initializes correctly.
124. $V_{IO}$ = 1.65 V to $V_{CC}$ or 2.7 V to $V_{CC}$ depending on the model.
125. $V_{CC}$ = 3 V and $V_{IO}$ = 3 V or 1.8 V. When $V_{IO}$ is at 1.8 V, I/O pins cannot operate at >1.8 V.
126. The recommended pull-up resistor for RY/BY# output is 5k to 10k ohms.

## 10.6 Capacitance characteristics

**Table 40 Connector capacitance for FBGA (LAA) package**

| Symbol | Description | Test setup | Typ | Max | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input capacitance | $V_{IN} = 0$ | 4 | 5.5 | pF |
| $C_{OUT}$ | Output capacitance | $V_{OUT} = 0$ | 3.5 | 5 | pF |
| $C_{IN2}$ | Control pin capacitance | $V_{IN} = 0$ | 4 | 8 | pF |
| RY/BY# | Output capacitance | $V_{OUT} = 0$ | 3 | 4 | pF |
| RESET# | Reset input capacitance | $V_{IN} = 0$ | 21 | 23 | pF |

**Table 41 Connector capacitance for FBGA (LAE) package**

| Symbol | Description | Test setup | Typ | Max | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input capacitance | $V_{IN} = 0$ | 3.5 | 5 | pF |
| $C_{OUT}$ | Output capacitance | $V_{OUT} = 0$ | 3.5 | 5 | pF |
| $C_{IN2}$ | Control pin capacitance | $V_{IN} = 0$ | 3.5 | 7 | pF |
| RY/BY# | Output capacitance | $V_{OUT} = 0$ | 2.5 | 3.5 | pF |
| RESET# | Reset input capacitance | $V_{IN} = 0$ | 20 | 22 | pF |

**Notes**
127.Sampled, not 100% tested.
128.Test conditions $T_A = 25°C$, f = 1.0 MHz.
129.Sampled, not 100% tested.
130.Test conditions $T_A = 25°C$, f = 1.0 MHz.

**Table 42 Connector capacitance for FBGA (VBU) package**

| Symbol | Description | Test setup | Typ | Max | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input capacitance | $V_{IN} = 0$ | 3.5 | 5 | pF |
| $C_{OUT}$ | Output capacitance | $V_{OUT} = 0$ | 3.5 | 5 | pF |
| $C_{IN2}$ | Control pin capacitance | $V_{IN} = 0$ | 3.5 | 7 | pF |
| RY/BY# | Output capacitance | $V_{OUT} = 0$ | 3 | 4 | pF |
| RESET# | Reset input capacitance | $V_{IN} = 0$ | 20 | 22 | pF |

**Table 43 Connector capacitance for TSOP package**

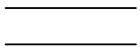
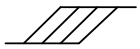
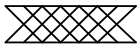
| Symbol | Description | Test setup | Typ | Max | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input capacitance | $V_{IN} = 0$ | 3 | 5 | pF |
| $C_{OUT}$ | Output capacitance | $V_{OUT} = 0$ | 3 | 4.5 | pF |
| $C_{IN2}$ | Control pin capacitance | $V_{IN} = 0$ | 3.5 | 7 | pF |
| RY/BY# | Output capacitance | $V_{OUT} = 0$ | 2.5 | 3.5 | pF |
| RESET# | Reset input capacitance | $V_{IN} = 0$ | 20 | 22 | pF |

**Notes**
131.Sampled, not 100% tested.
132.Test conditions $T_A = 25°C$, f = 1.0 MHz.
133.Sampled, not 100% tested.
134.Test conditions $T_A = 25°C$, f = 1.0 MHz.

# 11 Timing specifications

## 11.1 Key to switching waveforms

**Table 44 Switching waveforms**

| Waveform | Inputs | Outputs |
|---|---|---|
|  | Steady | |
| | Changing from H to L | |
|  | Changing from L to H | |
|  | Don’t care, any change permitted | Changing, state unknown |
| | Does not apply | Center line is high impedance state (High-Z) |

## 11.2 AC test conditions

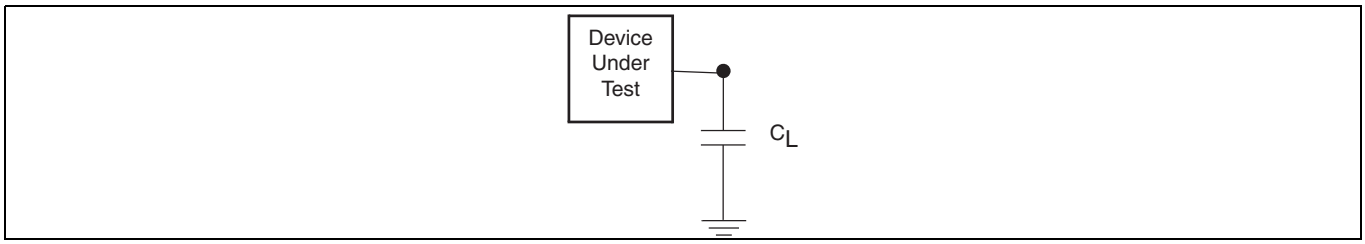



**Figure 13 Test setup**

**Table 45 Test specification**

| Parameter | All speeds | Units |
|---|---|---|
| Output load capacitance, $C_L$ | 30 | pF |
| Input rise and fall times[135] | 1.5 | ns |
| Input pulse levels | 0.0–$V_{IO}$ | V |
| Input timing measurement reference levels | $V_{IO}/2$ | V |
| Output timing measurement reference levels | $V_{IO}/2$ | V |

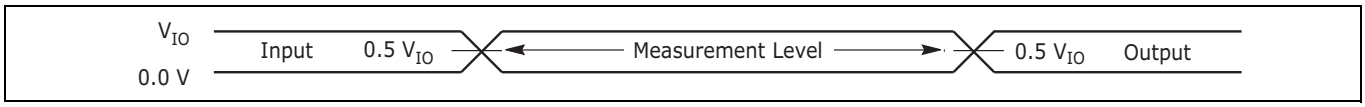



**Figure 14 Input waveforms and measurement levels**

**Note**
135.Measured between $V_{IL}$ max and $V_{IH}$ min.

## 11.3 Power-on reset (POR) and warm reset

Normal precautions must be taken for supply decoupling to stabilize the $V_{CC}$ and $V_{IO}$ power supplies. Each device in a system should have the $V_{CC}$ and $V_{IO}$ power supplies decoupled by a suitable capacitor close to the package connections (this capacitor is generally on the order of 0.1 µF).

**Table 46 Power-on and reset parameters**

| Parameter | Description | Limit | Value | Unit |
|---|---|---|---|---|
| $t_{VCS}$ | $V_{CC}$ setup time to first access[136, 137] | Min | 300 | µs |
| $t_{VIOS}$ | $V_{IO}$ setup time to first access[136, 137] | Min | 300 | µs |
| $t_{RPH}$ | RESET# LOW to CE# LOW | Min | 35 | µs |
| $t_{RP}$ | RESET# pulse width | Min | 200 | ns |
| $t_{RH}$ | Time between RESET# (HIGH) and CE# (LOW) | Min | 50 | ns |
| $t_{CEH}$ | CE# pulse width HIGH | Min | 20 | ns |

**Notes**

136. Not 100% tested.
137. Timing measured from $V_{CC}$ reaching $V_{CC}$ minimum and $V_{IO}$ reaching $V_{IO}$ minimum to $V_{IH}$ on Reset and $V_{IL}$ on CE#.
138. RESET# LOW is optional during POR. If RESET is asserted during POR, the later of $t_{RPH}$, $t_{VIOS}$, or $t_{VCS}$ will determine when CE# may go LOW. If RESET# remains LOW after $t_{VIOS}$, or $t_{VCS}$ is satisfied, $t_{RPH}$ is measured from the end of $t_{VIOS}$, or $t_{VCS}$. RESET must also be HIGH $t_{RH}$ before CE# goes LOW.
139. $V_{CC} \geq V_{IO} - 200$ mV during power-up.
140. $V_{CC}$ and $V_{IO}$ ramp rate can be non-linear.
141. Sum of $t_{RP}$ and $t_{RH}$ must be equal to or greater than $t_{RPH}$.

### 11.3.1 Power-on (Cold) reset (POR)

During the rise of power supplies the $V_{IO}$ supply voltage must remain less than or equal to the $V_{CC}$ supply voltage. $V_{IH}$ also must remain less than or equal to the $V_{IO}$ supply.

The cold reset EA requires a relatively long, hundreds of µs, period ($t_{VCS}$) to load all of the EAC algorithms and default state from non-volatile memory. During the cold reset period all control signals including CE# and RESET# are ignored. If CE# is LOW during $t_{VCS}$ the device may draw higher than normal POR current during $t_{VCS}$ but the level of CE# will not affect the cold reset EA. RESET# may be HIGH or LOW during $t_{VCS}$. If RESET# is LOW during $t_{VCS}$ it may remain LOW at the end of $t_{VCS}$ to hold the device in the hardware reset state. If RESET# is HIGH at the end of $t_{VCS}$ the device will go to the Standby state.

When power is first applied, with supply voltage below $V_{RST}$ then rising to reach operating range minimum, internal device configuration and warm reset activities are initiated. CE# is ignored for the duration of the POR operation ($t_{VCS}$ or $t_{VIOS}$). RESET# LOW during this POR period is optional. If RESET# is driven LOW during POR it must satisfy the hardware reset parameters $t_{RP}$ and $t_{RPH}$. In which case the reset operations will be completed at the later of $t_{VCS}$ or $t_{VIOS}$ or $t_{RPH}$. A CE#, OE#, or Address transition will initiate the 1st read operation. If CE# is held LOW during POR than the current address will be automatically read.

During cold reset the device will draw $I_{CC7}$ current.

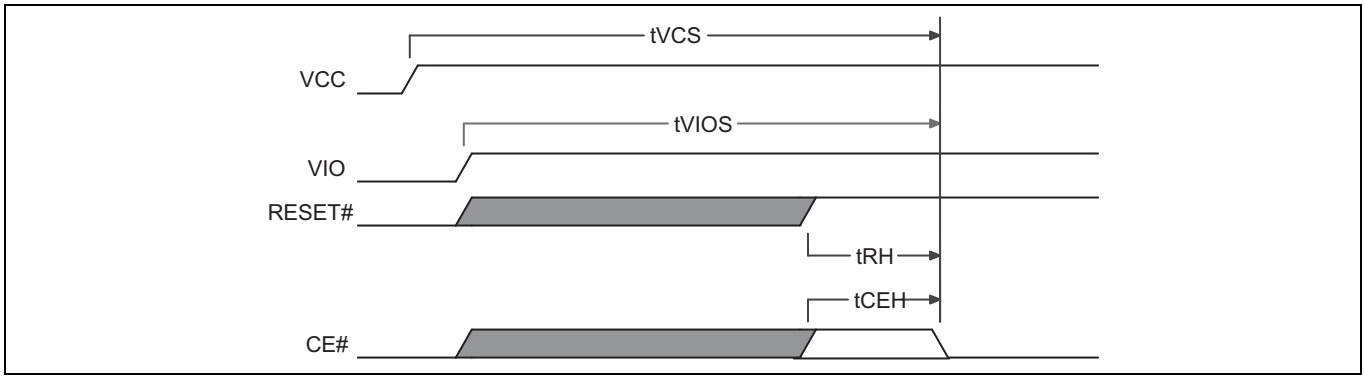



**Figure 15 Power-up diagram**

### 11.3.2 Hardware (Warm) reset

During hardware reset ($t_{RPH}$) the device will draw $I_{CC5}$ current.

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

If a cold reset has not been completed by the device when RESET# is asserted LOW after $t_{VCS}$, the Cold Reset# EA will be performed instead of the Warm RESET#, requiring $t_{VCS}$ time to complete. See **Figure 16**.

After the device has completed POR and entered the Standby state, any later transition to the hardware reset state will initiate the warm reset EA. A Warm Reset is much shorter than a cold reset, taking tens of µs ($t_{RPH}$) to complete. During the warm reset EA, any in progress EA is stopped and the EAC is returned to its POR state without reloading EAC algorithms from non-volatile memory. After the warm reset EA completes, the interface will remain in the hardware reset state if RESET# remains LOW. When RESET# returns HIGH the interface will transit to the Standby state. If RESET# is HIGH at the end of the warm reset EA, the interface will directly transit to the Standby state. If CE# is held LOW during warm reset than the current address will be automatically read.

If POR has not been properly completed by the end of $t_{VCS}$, a later transition to the hardware reset state will cause a transition to the power-on-reset interface state and initiate the cold reset EA. This ensures the device can complete a cold reset even if some aspect of the system power-on voltage ramp-up causes the POR to not initiate or complete correctly. The RY/BY# pin is LOW during cold or warm reset as an indication that the device is busy performing reset operations.

Hardware reset is initiated by the RESET# signal going to $V_{IL}$.

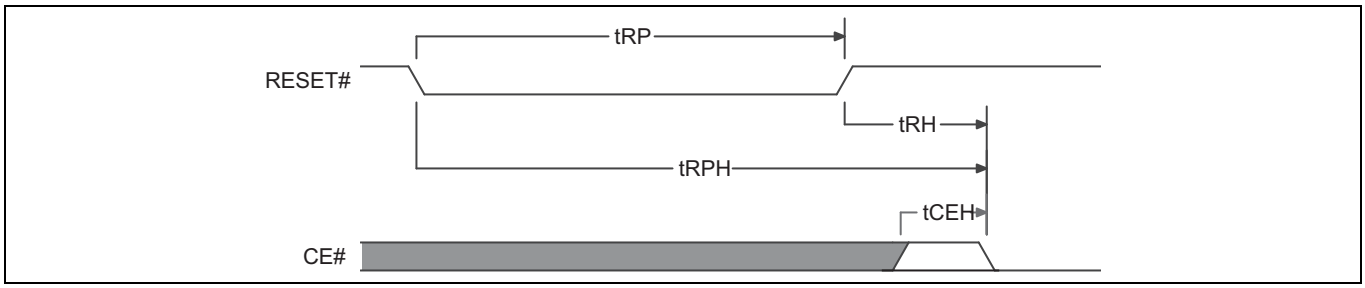



**Figure 16 Hardware reset**

## 11.4 AC characteristics

### 11.4.1 Asynchronous read operations

**Table 47 Read operation $V_{IO} = V_{CC}$ = 2.7 V to 3.6 V (–40°C to +85°C)**

| Parameter | | Description | | Test setup | | Speed option | Unit |
|---|---|---|---|---|---|---|---|
| JEDEC | Std | | | | | | |
| $t_{AVAV}$ | $t_{RC}$ | Read cycle time[142] | | 512 Mb, 1 Gb | Min | 100 | ns |
| $t_{AVQV}$ | $t_{ACC}$ | Address to output delay | CE# = $V_{IL}$<br>OE# = $V_{IL}$ | 512 Mb, 1 Gb | Max | 100 | ns |
| $t_{ELQV}$ | $t_{CE}$ | Chip Enable to output delay | OE# = $V_{IL}$ | 512 Mb, 1 Gb | Max | 100 | ns |
| | $t_{PACC}$ | Page access time | | 512 Mb, 1 Gb | Max | 15 | ns |
| $t_{GLQV}$ | $t_{OE}$ | Output Enable to output delay | | Read | Max | 25 | ns |
| | | | | Poll | Max | 35 | |
| $t_{AXQX}$ | $t_{OH}$ | Output hold time from addresses, CE# or OE#, whichever occurs first | | | Min | 0 | ns |
| $t_{EHQZ}$ | $t_{DF}$ | Chip Enable or Output Enable to output High-Z[142] | | | Max | 15 | ns |
| | $t_{OEH}$ | Output Enable hold time[142] | | Read | Min | 0 | ns |
| | | | | Poll | Min | 10 | ns |
| | $t_{ASO}$ | Address setup time to OE# LOW | | Poll | Min | 15 | ns |
| | $t_{AHT}$ | Address hold time from CE# or OE# HIGH | | Poll | Min | 0 | ns |
| | $t_{CEPH}$ | CE# HIGH | | Poll | Min | 20 | ns |
| | $t_{OEPH}$ | OE# HIGH | | Poll | Min | 20 | ns |
| | $t_{ASSB}$ | Automatic sleep to Standby time[142] | | CE# = $V_{IL}$, Address stable | Typ | 5 | µs |
| | | | | | Max | 8 | µs |
| $t_{BLEL}$ | $t_{FLEL}$ | BYTE# LOW to CE# LOW | | | Min | 10 | ns |
| $t_{BHEL}$ | $t_{FHEL}$ | BYTE# HIGH to CE# LOW | | | Min | 10 | ns |
| $t_{BLQV}$ | $t_{FLQV}$ | BYTE# LOW to output High-Z[142] | | | Max | 1 | µs |
| $t_{BHQV}$ | $t_{FHQV}$ | BYTE# HIGH to output delay | | | Max | 1 | µs |

**Note**
142.Not 100% tested.

**Table 48 Read operation $V_{IO}$ = 1.65 V to $V_{CC}$, $V_{CC}$ = 2.7 V to 3.6 V (–40°C to +85°C)**

| Parameter | | Description | | Test setup | | Speed option | Unit |
|---|---|---|---|---|---|---|---|
| JEDEC | Std | | | | | | |
| $t_{AVAV}$ | $t_{RC}$ | Read cycle time[143] | | 512 Mb, 1 Gb | Min | 110 | ns |
| $t_{AVQV}$ | $t_{ACC}$ | Address to output delay | CE# = $V_{IL}$<br>OE# = $V_{IL}$ | 512 Mb, 1 Gb | Max | 110 | ns |
| $t_{ELQV}$ | $t_{CE}$ | Chip Enable to output delay | OE# = $V_{IL}$ | 512 Mb, 1 Gb | Max | 110 | ns |
| | $t_{PACC}$ | Page access time | | 512 Mb, 1 Gb | Min | 25 | ns |
| $t_{GLQV}$ | $t_{OE}$ | Output Enable to output delay | | Read and poll | Max | 35 | ns |
| $t_{AXQX}$ | $t_{OH}$ | Output Hold time from addresses, CE# or OE#, whichever occurs first | | | Min | 0 | ns |
| $t_{EHQZ}$ | $t_{DF}$ | Chip Enable or Output Enable to output High-Z[143] | | | Max | 20 | ns |
| | $t_{OEH}$ | Output Enable hold time[143] | | Read | Min | 0 | ns |
| | | | | Poll | Min | 10 | ns |
| | $t_{ASO}$ | Address setup time to OE# LOW | | Poll | Min | 15 | ns |
| | $t_{AHT}$ | Address hold time from CE# or OE# HIGH | | Poll | Min | 0 | ns |
| | $t_{CEPH}$ | CE# HIGH | | Poll | Min | 20 | ns |
| | $t_{OEPH}$ | OE# HIGH | | Poll | Min | 20 | ns |
| | $t_{ASSB}$ | Automatic sleep to standby time[143] | | CE# = $V_{IL}$, Address stable | Typ | 5 | µs |
| | | | | | Max | 8 | µs |
| $t_{BLEL}$ | $t_{FLEL}$ | BYTE# LOW to CE# LOW | | | Min | 10 | ns |
| $t_{BHEL}$ | $t_{FHEL}$ | BYTE# HIGH to CE# LOW | | | Min | 10 | ns |
| $t_{BLQV}$ | $t_{FLQV}$ | BYTE# LOW to output High-Z[143] | | | Max | 1 | µs |
| $t_{BHQV}$ | $t_{FHQV}$ | BYTE# HIGH to output delay | | | Max | 1 | µs |

**Note**
143.Not 100% tested.

**Table 49 Read operation $V_{IO} = V_{CC}$ = 2.7 V to 3.6 V (–40°C to +105°C)**

| Parameter | | Description | | Test setup | | Speed option | Unit |
|---|---|---|---|---|---|---|---|
| JEDEC | Std | | | | | | |
| $t_{AVAV}$ | $t_{RC}$ | Read cycle time[144] | | 512 Mb, 1 Gb | Min | 110 | ns |
| $t_{AVQV}$ | $t_{ACC}$ | Address to output delay | CE# = $V_{IL}$<br>OE# = $V_{IL}$ | 512 Mb, 1 Gb | Max | 110 | ns |
| $t_{ELQV}$ | $t_{CE}$ | Chip Enable to output delay | OE# = $V_{IL}$ | 512 Mb, 1 Gb | Max | 110 | ns |
| | $t_{PACC}$ | Page access time | | 512 Mb, 1 Gb | Max | 15 | ns |
| $t_{GLQV}$ | $t_{OE}$ | Output Enable to output delay | | Read | Max | 25 | ns |
| | | | | Poll | Max | 35 | |
| $t_{AXQX}$ | $t_{OH}$ | Output hold time from addresses, CE# or OE#, whichever occurs first | | | Min | 0 | ns |
| $t_{EHQZ}$ | $t_{DF}$ | Chip Enable or Output Enable to output High-Z[144] | | | Max | 15 | ns |
| | $t_{OEH}$ | Output Enable hold time[144] | | Read | Min | 0 | ns |
| | | | | Poll | Min | 10 | ns |
| | $t_{ASO}$ | Address setup time to OE# LOW | | Poll | Min | 15 | ns |
| | $t_{AHT}$ | Address hold time from CE# or OE# HIGH | | Poll | Min | 0 | ns |
| | $t_{CEPH}$ | CE# HIGH | | Poll | Min | 20 | ns |
| | $t_{OEPH}$ | OE# HIGH | | Poll | Min | 20 | ns |
| | $t_{ASSB}$ | Automatic sleep to standby time[144] | | CE# = $V_{IL}$, Address stable | Typ | 5 | µs |
| | | | | | Max | 8 | µs |
| $t_{BLEL}$ | $t_{FLEL}$ | BYTE# LOW to CE# LOW | | | Min | 10 | ns |
| $t_{BHEL}$ | $t_{FHEL}$ | BYTE# HIGH to CE# LOW | | | Min | 10 | ns |
| $t_{BLQV}$ | $t_{FLQV}$ | BYTE# LOW to output High-Z[144] | | | Max | 1 | µs |
| $t_{BHQV}$ | $t_{FHQV}$ | BYTE# HIGH to output delay | | | Max | 1 | µs |

**Note**
144.Not 100% tested.

**Table 50 Read operation $V_{IO}$ = 1.65 V to $V_{CC}$, $V_{CC}$ = 2.7 V to 3.6 V (–40°C to +105°C)**

| Parameter | | Description | | Test setup | | Speed option | Unit |
|---|---|---|---|---|---|---|---|
| JEDEC | Std | | | | | | |
| $t_{AVAV}$ | $t_{RC}$ | Read cycle time[145] | | 512 Mb, 1 Gb | Min | 120 | ns |
| $t_{AVQV}$ | $t_{ACC}$ | Address to output delay | CE# = $V_{IL}$<br>OE# = $V_{IL}$ | 512 Mb, 1 Gb | Max | 120 | ns |
| $t_{ELQV}$ | $t_{CE}$ | Chip Enable to output delay | OE# = $V_{IL}$ | 512 Mb, 1 Gb | Max | 120 | ns |
| | $t_{PACC}$ | Page access time | | 512 Mb, 1 Gb | Max | 25 | ns |
| $t_{GLQV}$ | $t_{OE}$ | Output Enable to output delay | | Read and Poll | Max | 35 | ns |
| $t_{AXQX}$ | $t_{OH}$ | Output Hold time from addresses, CE# or OE#, whichever occurs first | | | Min | 0 | ns |
| $t_{EHQZ}$ | $t_{DF}$ | Chip Enable or Output Enable to output High-Z[145] | | | Max | 15 | ns |
| | $t_{OEH}$ | Output Enable hold time[145] | | Read | Min | 0 | ns |
| | | | | Poll | Min | 10 | ns |
| | $t_{ASO}$ | Address setup time to OE# LOW | | Poll | Min | 15 | ns |
| | $t_{AHT}$ | Address hold time from CE# or OE# HIGH | | Poll | Min | 0 | ns |
| | $t_{CEPH}$ | CE# HIGH | | Poll | Min | 20 | ns |
| | $t_{OEPH}$ | OE# HIGH | | Poll | Min | 20 | ns |
| | $t_{ASSB}$ | Automatic sleep to standby time[145] | | CE# = $V_{IL}$, Address stable | Typ | 5 | µs |
| | | | | | Max | 8 | µs |
| $t_{BLEL}$ | $t_{FLEL}$ | BYTE# LOW to CE# LOW | | | Min | 10 | ns |
| $t_{BHEL}$ | $t_{FHEL}$ | BYTE# HIGH to CE# LOW | | | Min | 10 | ns |
| $t_{BLQV}$ | $t_{FLQV}$ | BYTE# LOW to output High-Z[145] | | | Max | 1 | µs |
| $t_{BHQV}$ | $t_{FHQV}$ | BYTE# HIGH to output delay | | | Max | 1 | µs |

**Note**
145.Not 100% tested.

**Table 51 Read operation $V_{IO} = V_{CC}$ = 2.7 V to 3.6 V (–40°C to +125°C)**

| Parameter | | Description | | Test setup | | Speed option | Unit |
|---|---|---|---|---|---|---|---|
| JEDEC | Std | | | | | | |
| $t_{AVAV}$ | $t_{RC}$ | Read cycle time[146] | CE# = $V_{IL}$ OE# = $V_{IL}$ | 512 Mb, 1 Gb | Min | 120 | ns |
| $t_{AVQV}$ | $t_{ACC}$ | Address to output delay | OE# = $V_{IL}$ | 512 Mb, 1 Gb | Max | 120 | ns |
| $t_{ELQV}$ | $t_{CE}$ | Chip Enable to output delay | | 512 Mb, 1 Gb | Max | 120 | ns |
| | $t_{PACC}$ | Page access time | | 512 Mb, 1 Gb | Max | 15 | ns |
| $t_{GLQV}$ | $t_{OE}$ | Output Enable to output delay | | Read | Max | 25 | ns |
| | | | | Poll | Max | 35 | ns |
| $t_{AXQX}$ | $t_{OH}$ | Output Hold time from addresses, CE# or OE#, whichever occurs first | | | Min | 0 | ns |
| $t_{EHQZ}$ | $t_{DF}$ | Chip Enable or Output Enable to output High-Z[146] | | | Max | 15 | ns |
| | $t_{OEH}$ | Output Enable hold time[146] | | Read | Min | 0 | ns |
| | | | | Poll | Min | 10 | ns |
| | $t_{ASO}$ | Address setup time to OE# LOW | | Poll | Min | 15 | ns |
| | $t_{AHT}$ | Address hold time from CE# or OE# HIGH | | Poll | Min | 0 | ns |
| | $t_{CEPH}$ | CE# HIGH | | Poll | Min | 20 | ns |
| | $t_{OEPH}$ | OE# HIGH | | Poll | Min | 20 | ns |
| | $t_{ASSB}$ | Automatic sleep to standby time[146] | | CE# = $V_{IL}$, Address stable | Typ | 5 | µs |
| | | | | | Max | 8 | µs |
| $t_{BLEL}$ | $t_{FLEL}$ | BYTE# LOW to CE# LOW | | | Min | 10 | ns |
| $t_{BHEL}$ | $t_{FHEL}$ | BYTE# HIGH to CE# LOW | | | Min | 10 | ns |
| $t_{BLQV}$ | $t_{FLQV}$ | BYTE# LOW to output High-Z[146] | | | Max | 1 | µs |
| $t_{BHQV}$ | $t_{FHQV}$ | BYTE# HIGH to output delay | | | Max | 1 | µs |

**Note**
146.Not 100% tested.

**Table 52 Read operation $V_{IO}$ = 1.65 V to $V_{CC}$, $V_{CC}$ = 2.7 V to 3.6 V (–40°C to +125°C)**

| Parameter | | Description | | Test setup | | Speed option | Unit |
|---|---|---|---|---|---|---|---|
| JEDEC | Std | | | | | | |
| $t_{AVAV}$ | $t_{RC}$ | Read cycle time[147] | CE# = $V_{IL}$ OE# = $V_{IL}$ | 512 Mb, 1 Gb | Min | 130 | ns |
| $t_{AVQV}$ | $t_{ACC}$ | Address to output delay | OE# = $V_{IL}$ | 512 Mb, 1 Gb | Max | 130 | ns |
| $t_{ELQV}$ | $t_{CE}$ | Chip Enable to output delay | | 512 Mb, 1 Gb | Max | 130 | ns |
| | $t_{PACC}$ | Page access time | | 512 Mb, 1 Gb | Max | 20 | ns |
| $t_{GLQV}$ | $t_{OE}$ | Output Enable to output delay | | Read | Max | 25 | ns |
| | | | | Poll | Max | 35 | ns |
| $t_{AXQX}$ | $t_{OH}$ | Output Hold time from addresses, CE# or OE#, whichever occurs first | | | Min | 0 | ns |
| $t_{EHQZ}$ | $t_{DF}$ | Chip Enable or Output Enable to output High-Z[147] | | | Max | 15 | ns |
| | $t_{OEH}$ | Output Enable hold time[147] | | Read | Min | 0 | ns |
| | | | | Poll | Min | 10 | ns |
| | $t_{ASO}$ | Address setup time to OE# LOW | | Poll | Min | 15 | ns |
| | $t_{AHT}$ | Address hold time from CE# or OE# HIGH | | Poll | Min | 0 | ns |
| | $t_{CEPH}$ | CE# HIGH | | Poll | Min | 20 | ns |
| | $t_{OEPH}$ | OE# HIGH | | Poll | Min | 20 | ns |
| | $t_{ASSB}$ | Automatic sleep to standby time[147] | | CE# = $V_{IL}$, Address stable | Typ | 5 | µs |
| | | | | | Max | 8 | µs |
| $t_{BLEL}$ | $t_{FLEL}$ | BYTE# LOW to CE# LOW | | | Min | 10 | ns |
| $t_{BHEL}$ | $t_{FHEL}$ | BYTE# HIGH to CE# LOW | | | Min | 10 | ns |
| $t_{BLQV}$ | $t_{FLQV}$ | BYTE# LOW to output High-Z[147] | | | Max | 1 | µs |
| $t_{BHQV}$ | $t_{FHQV}$ | BYTE# HIGH to output delay | | | Max | 1 | µs |

**Note**
147.Not 100% tested.

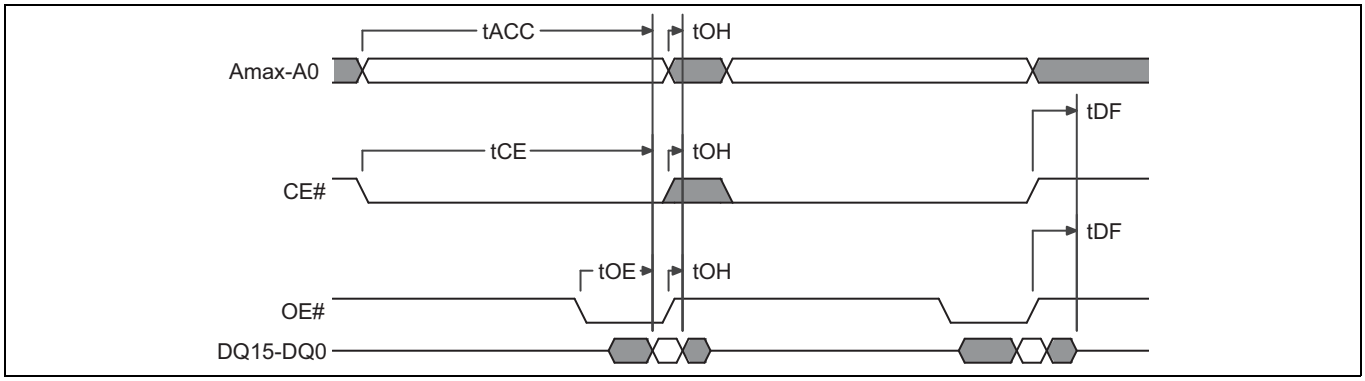



**Figure 17 Back to Back Read ($t_{ACC}$) operation timing diagram**[151]

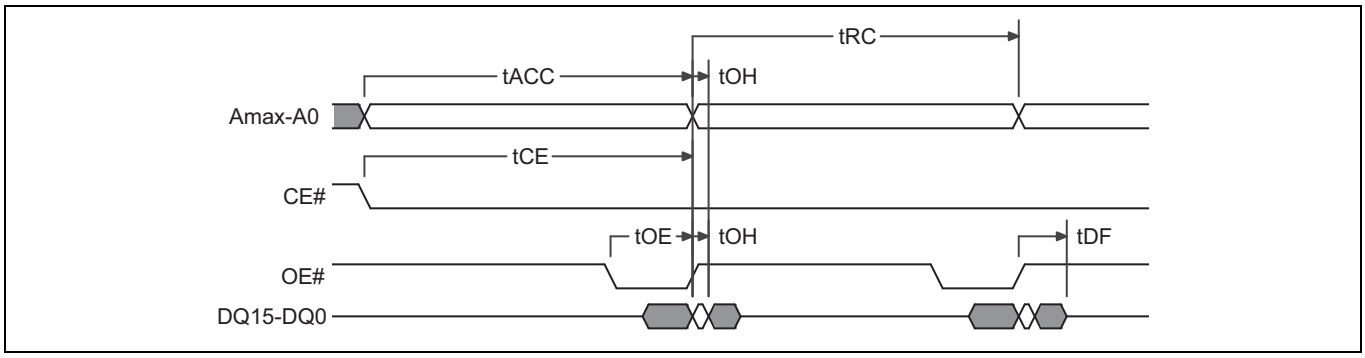



**Figure 18 Back to Back Read ($t_{RC}$) operation timing diagram**[148, 149]

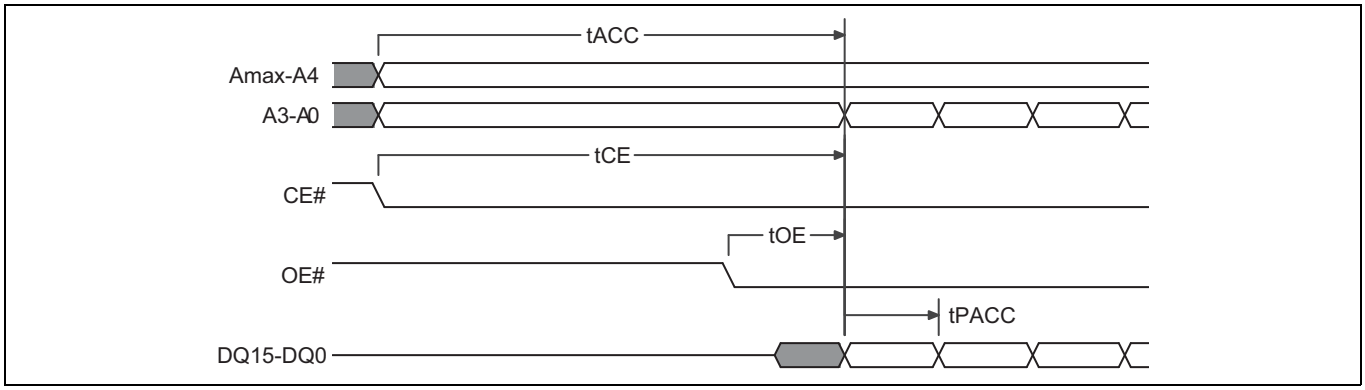



**Figure 19 Page Read timing diagram**[148, 150]

**Notes**

148. Address are Amax:A0 in word mode; Amax:A1 in byte mode, Data are DQ15–DQ0 in word mode; DQ7–DQ0 in byte mode.
149. Back to Back operations, in which CE# remains LOW between accesses, requires an address change to initiate the second access.
150. Toggle A3:A0. in word mode; A3:A1 in byte mode.
151. Address are Amax:A0 in word mode; Amax:A1 in byte mode, Data are DQ15–DQ0 in word mode; DQ7–DQ0 in byte mode.

## 11.4.2 Asynchronous Write operations

**Table 53 Write operations**

| Parameter | | Description | | $V_{IO}$ = 2.7 V to $V_{CC}$ | $V_{IO}$ = 1.65 V to $V_{CC}$ | Unit |
|---|---|---|---|---|---|---|
| JEDEC | Std | | | | | |
| $t_{AVAV}$ | $t_{WC}$ | Write cycle time[152] | Min | 60 | | ns |
| $t_{AVWL}$ | $t_{AS}$ | Address setup time | Min | 0 | | ns |
| | $t_{ASO}$ | Address setup time to OE# LOW during toggle bit polling | Min | 15 | | ns |
| $t_{WLAX}$ | $t_{AH}$ | Address hold time | Min | 45 | | ns |
| | $t_{AHT}$ | Address hold time From CE# or OE# HIGH during toggle bit polling | Min | 0 | | ns |
| $t_{DVWH}$ | $t_{DS}$ | Data setup time | Min | 30 | | ns |
| $t_{WHDX}$ | $t_{DH}$ | Data hold time | Min | 0 | | ns |
| $t_{GHWL}$ | $t_{GHWL}$ | Read recovery time before write (OE# HIGH to WE# LOW) | Min | 0 | | ns |
| $t_{ELWL}$ | $t_{CS}$ | CE# setup time | Min | 0 | | ns |
| $t_{WHEH}$ | $t_{CH}$ | CE# hold time | Min | 0 | | ns |
| $t_{WLWH}$ | $t_{WP}$ | WE# pulse width | Min | 25 | | ns |
| $t_{WHWL}$ | $t_{WPH}$ | WE# pulse width HIGH | Min | 20 | | ns |
| | $t_{SEA}$ | Sector erase time-out | Min | 50 | | µs |

**Note**
152.Not 100% tested.

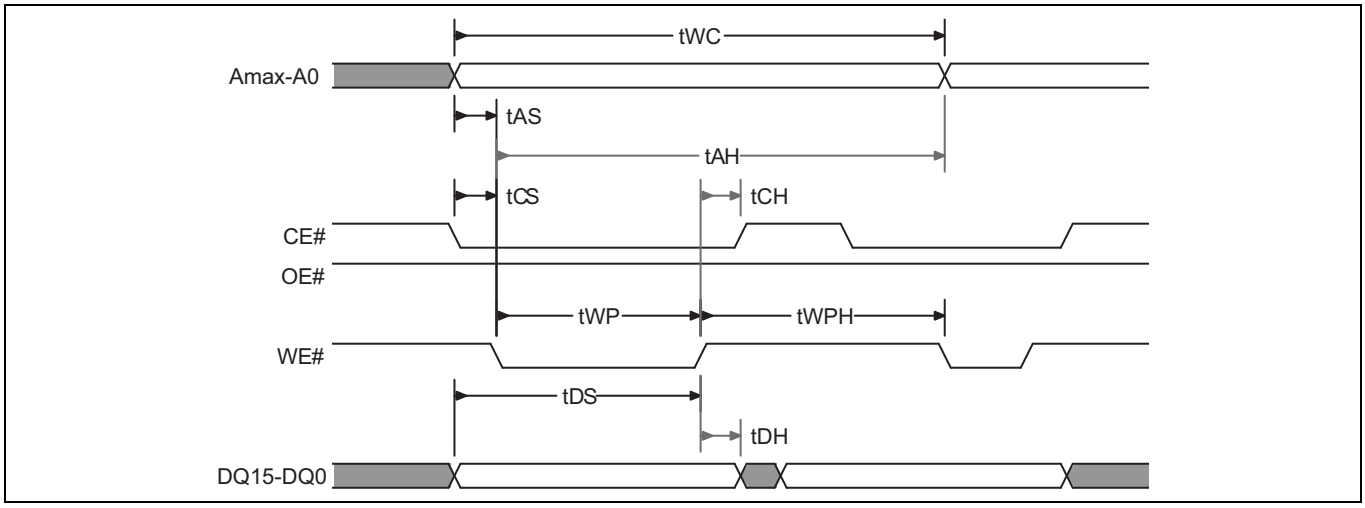



**Figure 20 Back to Back Write operation timing diagram**[153]

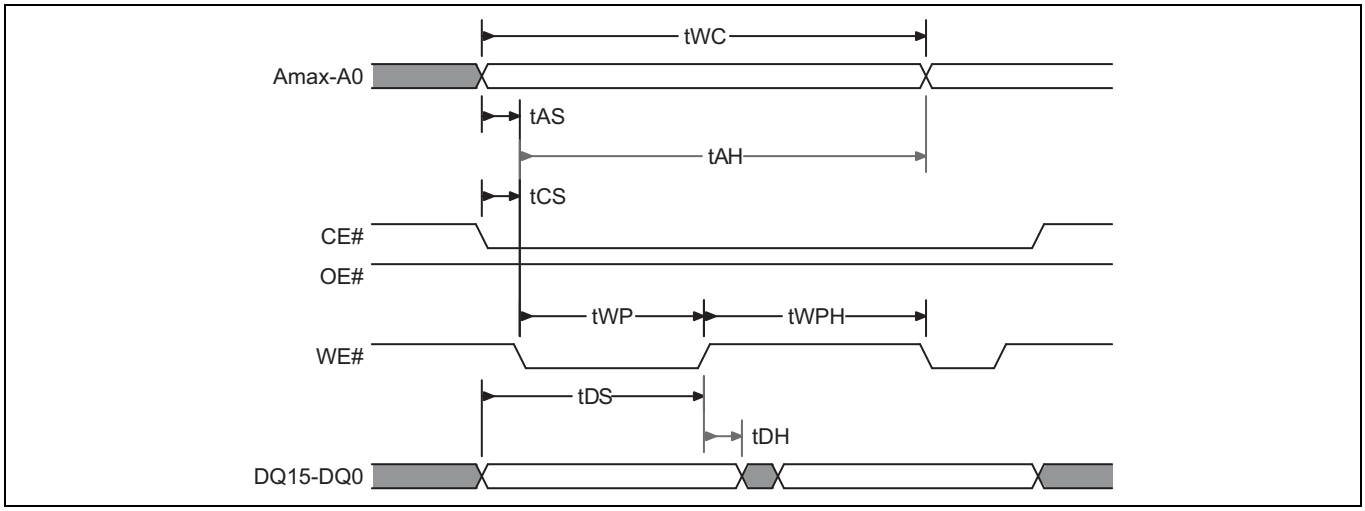



**Figure 21 Back to Back (CE#VIL) Write operation timing diagram**[153]

**Note**
153. Address are Amax:A0 in word mode; Amax:A1 in byte mode, Data are DQ15–DQ0 in word mode; DQ7–DQ0 in byte mode.

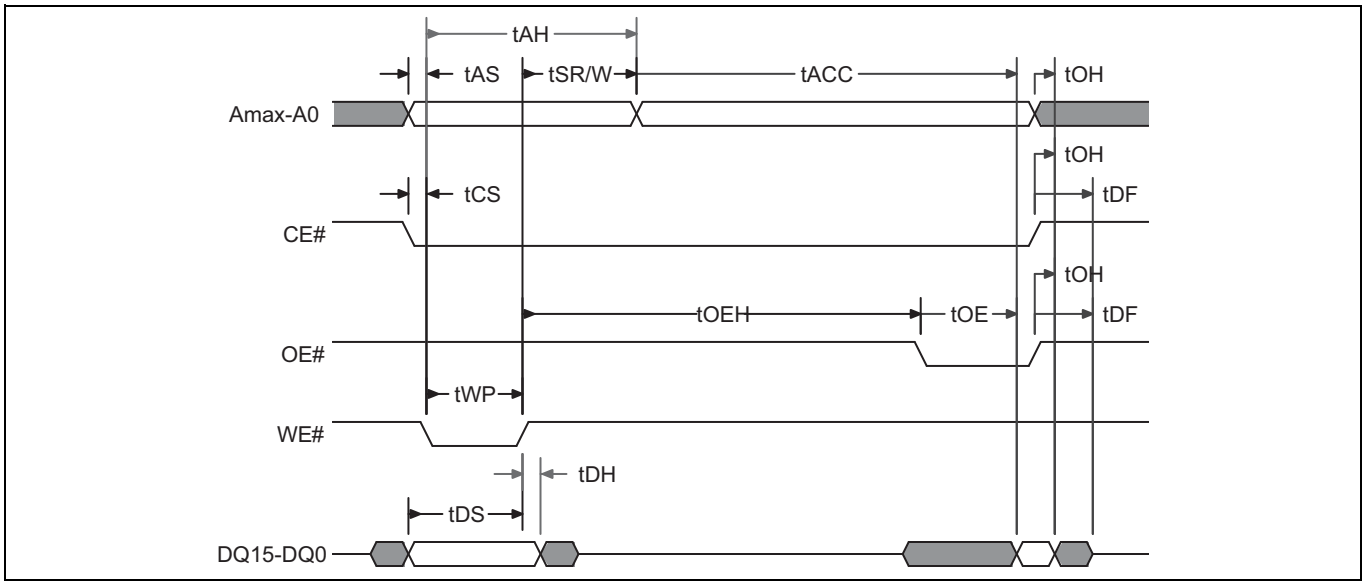



**Figure 22 Write to Read ($t_{ACC}$) operation timing diagram**[154]

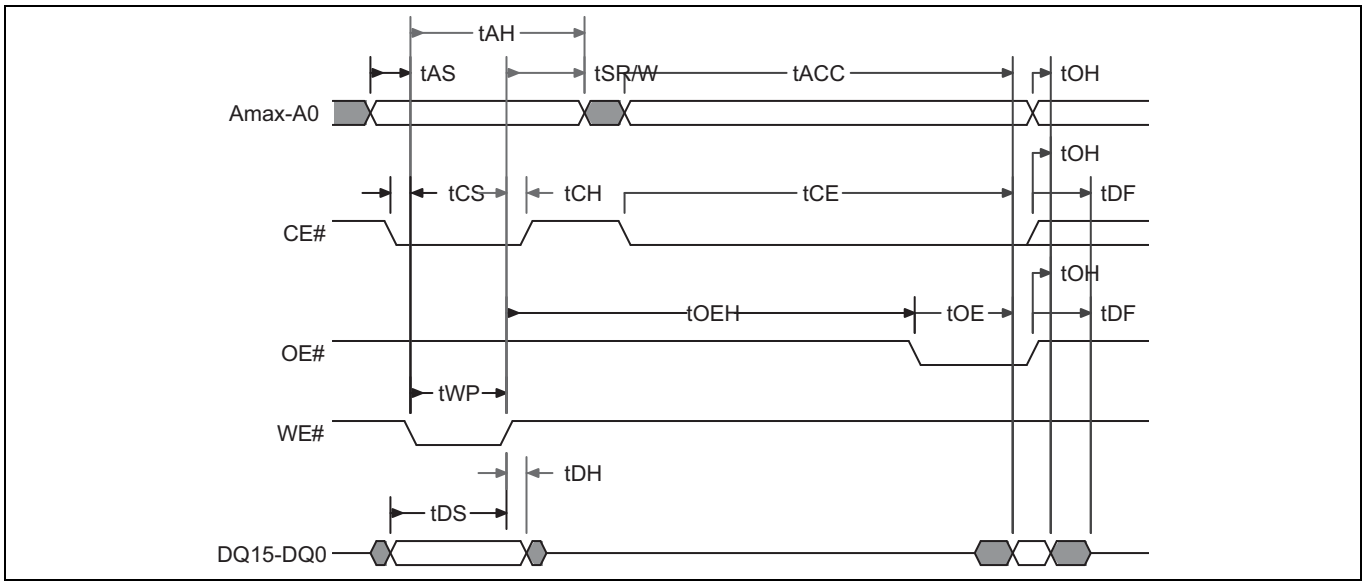



**Figure 23 Write to Read ($t_{CE}$) operation timing diagram**[154]

**Note**
154.Address are Amax:A0 in word mode; Amax:A1 in byte mode, Data are DQ15–DQ0 in word mode; DQ7–DQ0 in byte mode.

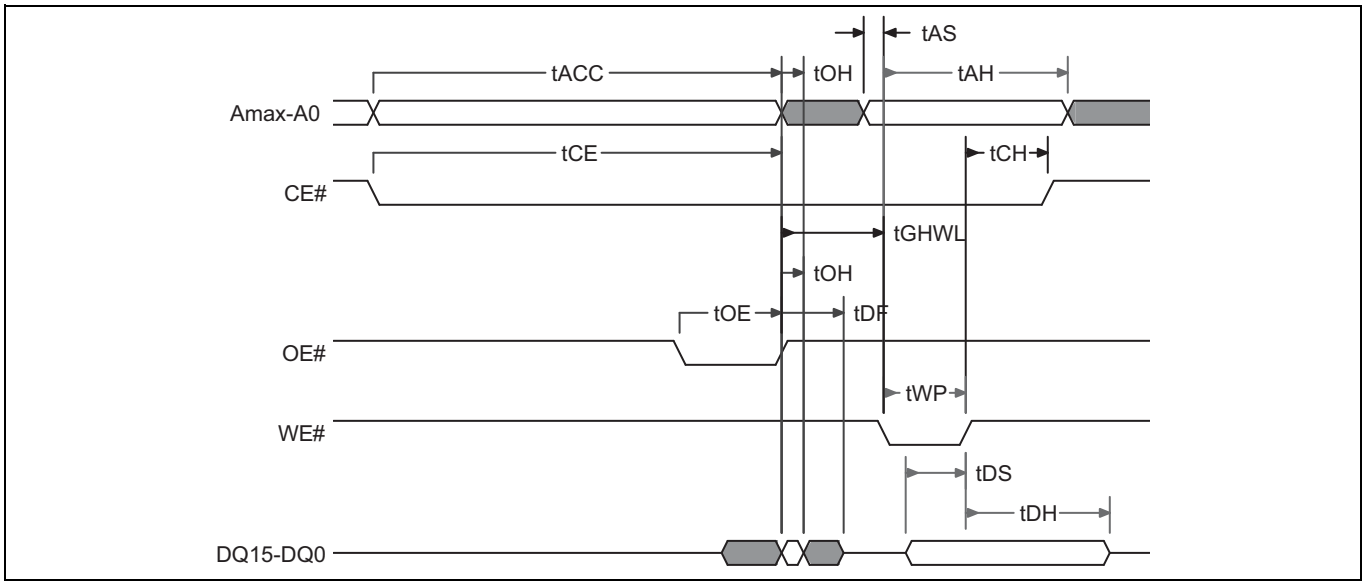



**Figure 24 Read to Write (CE# $V_{IL}$) operation timing diagram**[155]

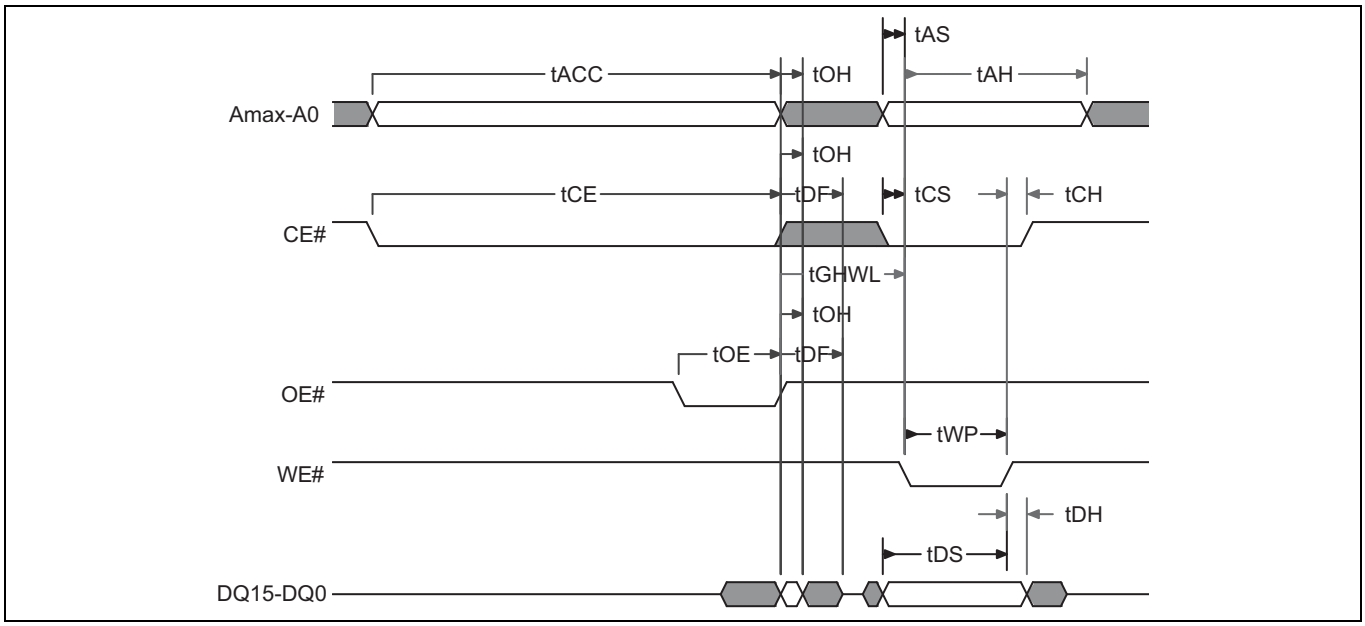



**Figure 25 Read to Write (CE# Toggle) operation timing diagram**[155]

**Note**
155.Address are Amax:A0 in word mode; Amax:A1 in byte mode, Data are DQ15–DQ0 in word mode; DQ7–DQ0 in byte mode.

**Table 54 Erase/program operations**

| Parameter | | Description | | $V_{IO}$ = 2.7 V to $V_{CC}$ | $V_{IO}$ = 1.65 V to $V_{CC}$ | Unit |
|---|---|---|---|---|---|---|
| JEDEC | Std | | | | | |
| $t_{WHWH1}$ | $t_{WHWH1}$ | Write Buffer Program operation | Typ | Note [158] | | µs |
| | | Effective Write Buffer Program operation per word | Typ | Note [158] | | µs |
| | | Program operation per word or page | Typ | Note [158] | | µs |
| $t_{WHWH2}$ | $t_{WHWH2}$ | Sector Erase operation[156] | Typ | Note [158] | | ms |
| | $t_{BUSY}$ | Erase/Program valid to RY/BY# delay | Max | 80 | | ns |
| | $t_{SR/W}$ | Latency between Read and Write operations[157] | Min | 10 | | ns |
| | $t_{ESL}$ | Erase Suspend Latency | Max | Note [158] | | µs |
| | $t_{PSL}$ | Program Suspend Latency | Max | Note [158] | | µs |
| | $t_{RB}$ | RY/BY# Recovery time | Min | 0 | | µs |
| | $t_{PPB}$ | PPB Lock Unlock | Min | 80 | | µs |
| | | | Max | 120 | | |
| | $t_{DP}$ | Data Polling to Protected Sector (Program) | Min | 3 | | µs |
| | | | Max | 20 | | |
| | | Data Polling to Protected Sector (Erase) | Min | 3 | | |
| | | | Max | 100 | | |
| | $t_{VHH}$ | $V_{HH}$ Rise and Fall time[156] | Min | 250 | | ns |
| | $t_{TOR}$ | Exceeded timing cleared (DQ5) | Min | 100 | | ns |

**Notes**
156.Not 100% tested.
157.Upon the rising edge of WE#, must wait $t_{SR/W}$ before switching to another address.
158.See **Table 18** and **Table 19** for specific values.

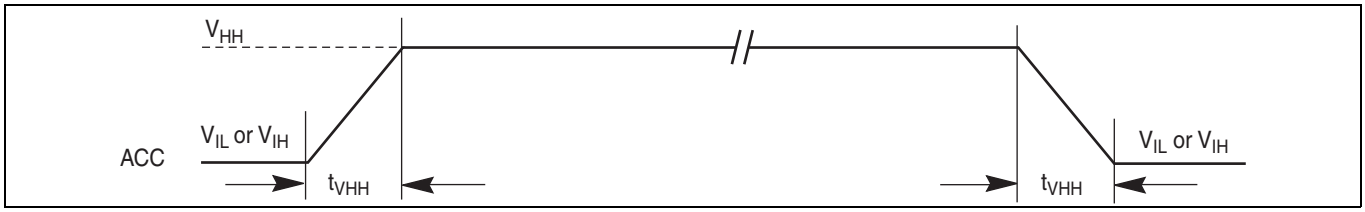



**Figure 26 Accelerated Program operation timing diagram**

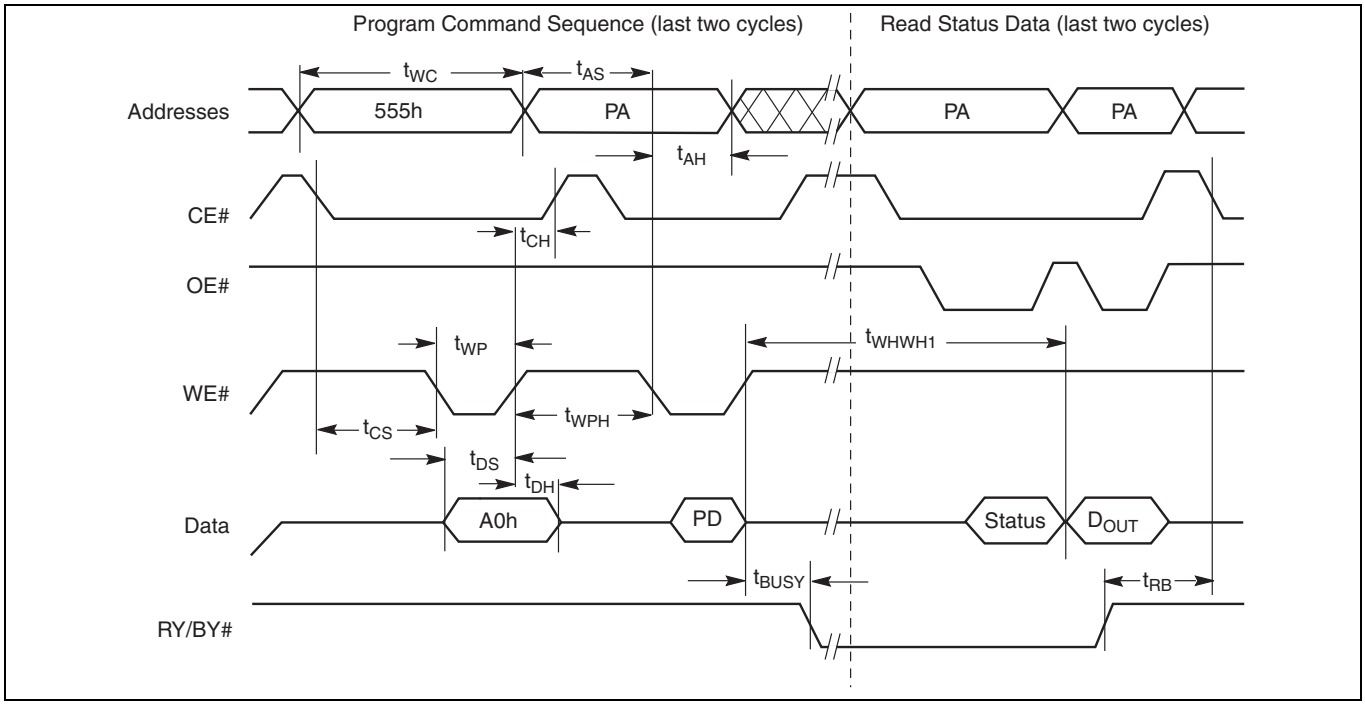



**Figure 27 Program operation timing diagram**[159, 160]

**Notes**

159. Address are Amax:A0 in word mode; Amax:A1 in byte mode, Data are DQ15–DQ0 in word mode; DQ7–DQ0 in byte mode.
160. PA = program address, PD = program data, $D_{OUT}$ is the true data at the program address.

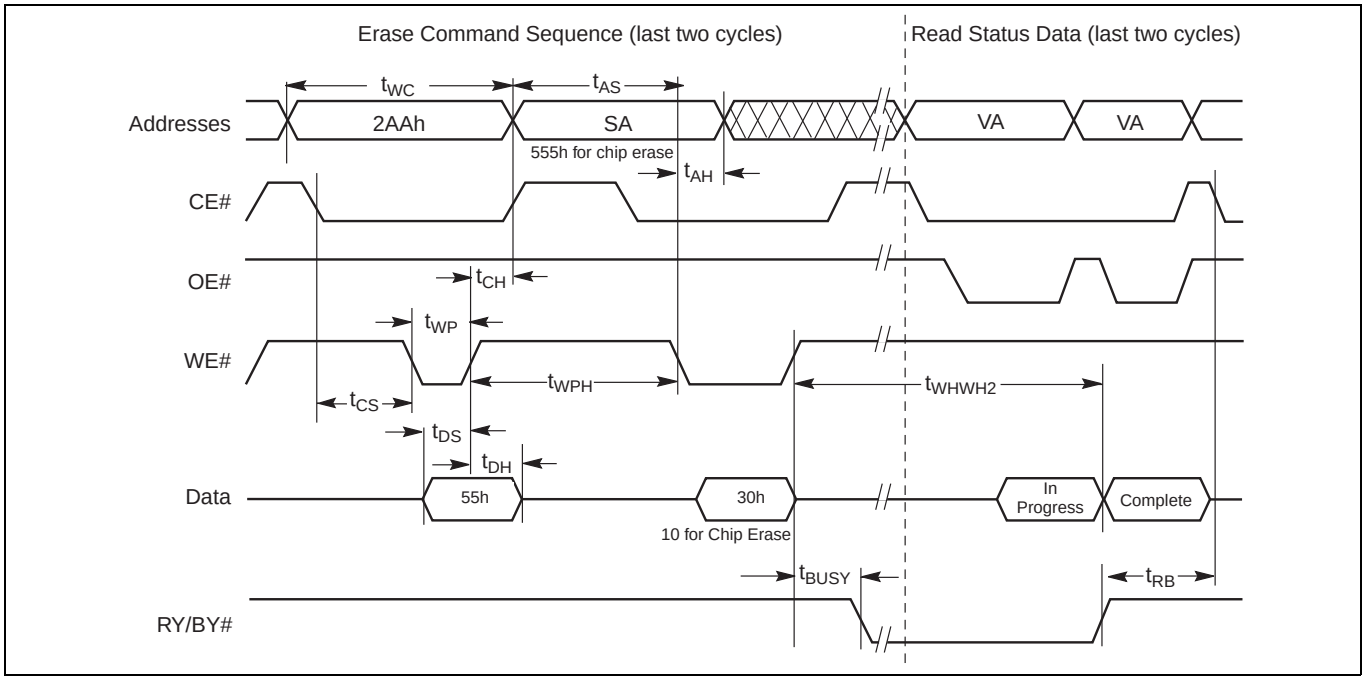



**Figure 28 Chip/Sector Erase operation timing diagram**[161, 162]

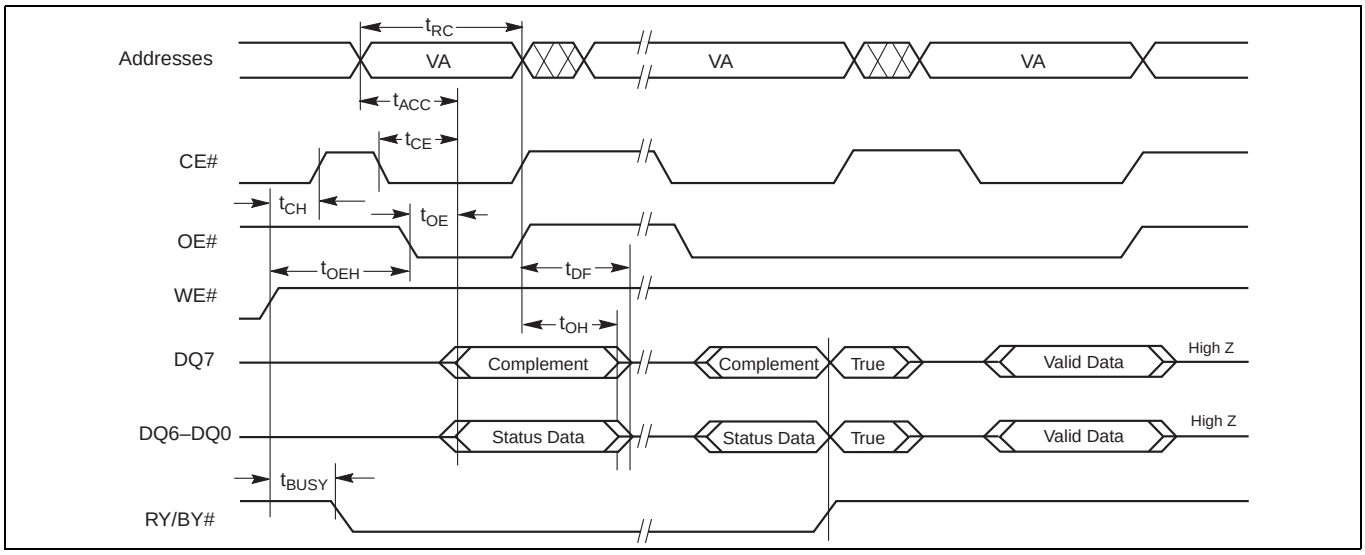



**Figure 29 Data# Polling timing diagram (During EA's)**[163]

**Notes**

161. Address are Amax:A0 in word mode; Amax:A1 in byte mode, Data are DQ15–DQ0 in word mode; DQ7–DQ0 in byte mode.
162. SA = sector address (for sector erase), VA = valid address for reading status data.
163. VA = Valid address. Illustration shows first status cycle after command sequence, last status read cycle, and array data read cycle.

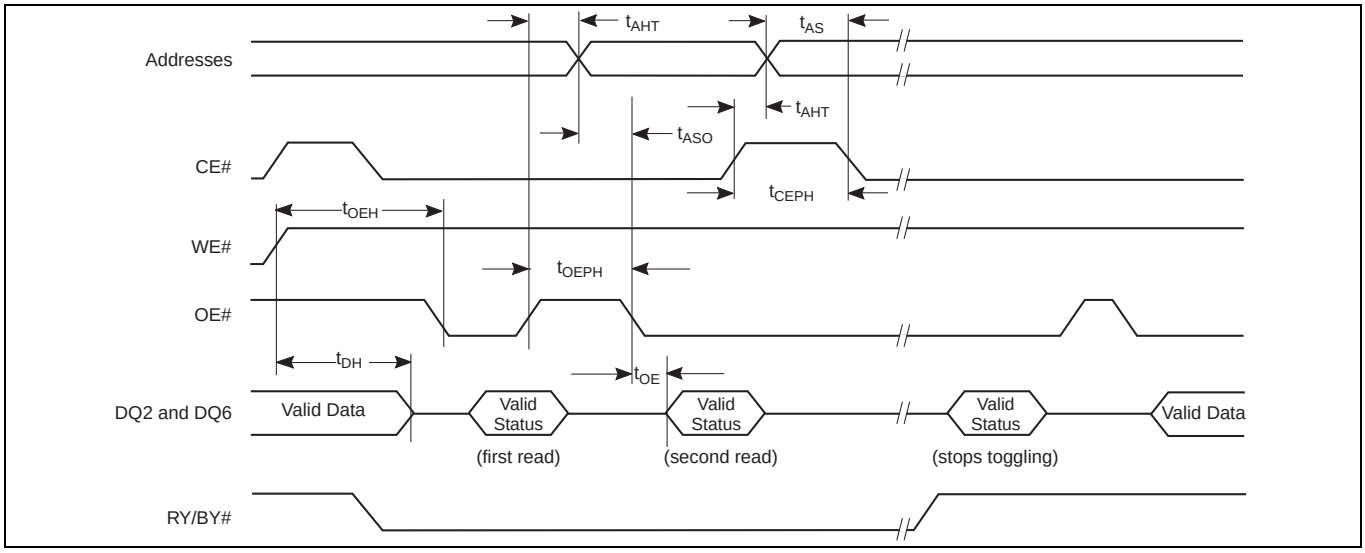



**Figure 30 Toggle Bit timing diagram (During EA's)**[164]

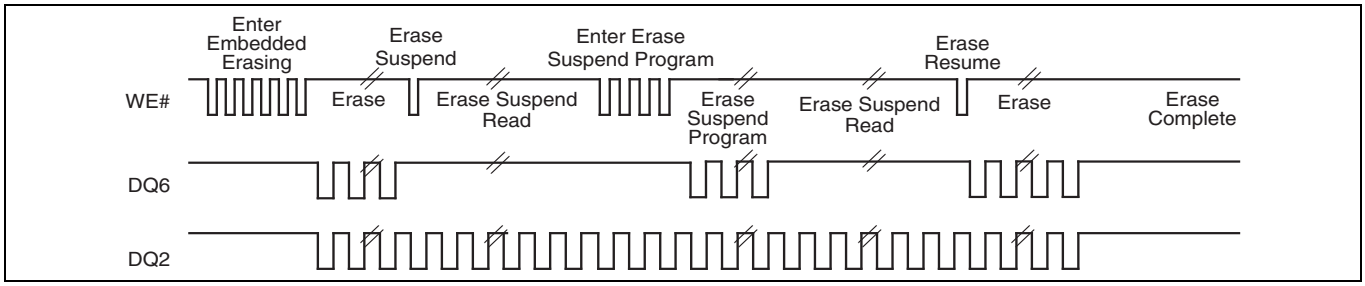



**Figure 31 DQ2 vs. DQ6 relationship diagram**[165]

**Notes**

164. DQ6 will toggle at any read address while the device is busy. DQ2 will toggle if the address is within the actively erasing sector.
165. The system may use OE# or CE# to toggle DQ2 and DQ6. DQ2 toggles only when read at an address within the erase-suspended sector.

### 11.4.3 Alternate CE# Controlled Write operations

**Table 55 Alternate CE# Controlled Write operations**

| Parameter | | Description | | $V_{IO}$ = 2.7 V to $V_{CC}$ | $V_{IO}$ = 1.65 V to $V_{CC}$ | Unit |
|---|---|---|---|---|---|---|
| JEDEC | Std | | | | | |
| $t_{AVAV}$ | $t_{WC}$ | Write Cycle time[166] | Min | 60 | | ns |
| $t_{AVWL}$ | $t_{AS}$ | Address Setup time | Min | 0 | | ns |
| | $t_{ASO}$ | Address Setup time to OE# LOW during toggle bit polling | Min | 15 | | ns |
| $t_{WLAX}$ | $t_{AH}$ | Address Hold time | Min | 45 | | ns |
| | $t_{AHT}$ | Address Hold time from CE# or OE# HIGH during toggle bit polling | Min | 0 | | ns |
| $t_{DVWH}$ | $t_{DS}$ | Data Setup time | Min | 30 | | ns |
| $t_{WHDX}$ | $t_{DH}$ | Data Hold time | Min | 0 | | ns |
| | $t_{CEPH}$ | CE# HIGH during toggle bit polling | Min | 20 | | ns |
| | $t_{0EPH}$ | OE# HIGH during toggle bit polling | Min | 20 | | ns |
| $t_{GHEK}$ | $t_{GHEL}$ | Read Recovery time before Write (OE# HIGH to WE# LOW) | Min | 0 | | ns |
| $t_{WLEL}$ | $t_{WS}$ | WE# Setup time | Min | 0 | | ns |
| $t_{ELWH}$ | $t_{WH}$ | WE# Hold time | Min | 0 | | ns |
| $t_{ELEH}$ | $t_{CP}$ | CE# Pulse Width | Min | 25 | | ns |
| $t_{EHEL}$ | $t_{CPH}$ | CE# Pulse Width HIGH | Min | 20 | | ns |
| | $t_{SEA}$ | Sector Erase time-out | Min | 50 | | µs |

**Note**
166.Not 100% tested.

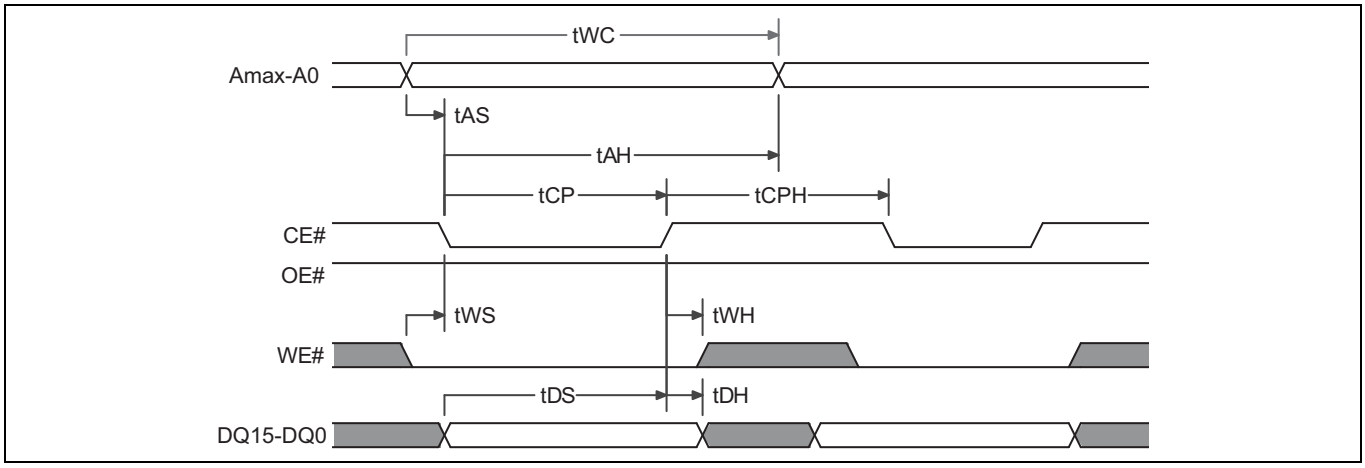



**Figure 32 Back to Back (CE#) Write operation timing diagram**[167]

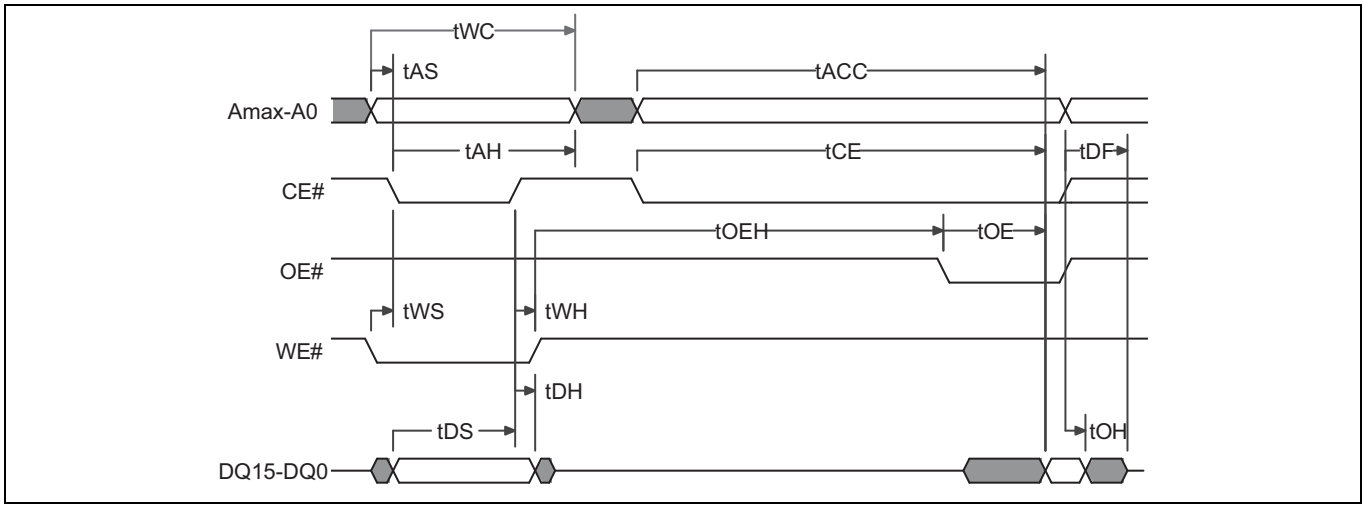



**Figure 33 (CE#) Write to Read operation timing diagram**[167]

**Note**
167.Address are Amax:A0 in word mode; Amax:A-1 in byte mode, Data are DQ15–DQ0 in word mode; DQ7–DQ0 in byte mode.

# 12 Physical interface

## 12.1 56-pin TSOP

### 12.1.1 Connection diagram

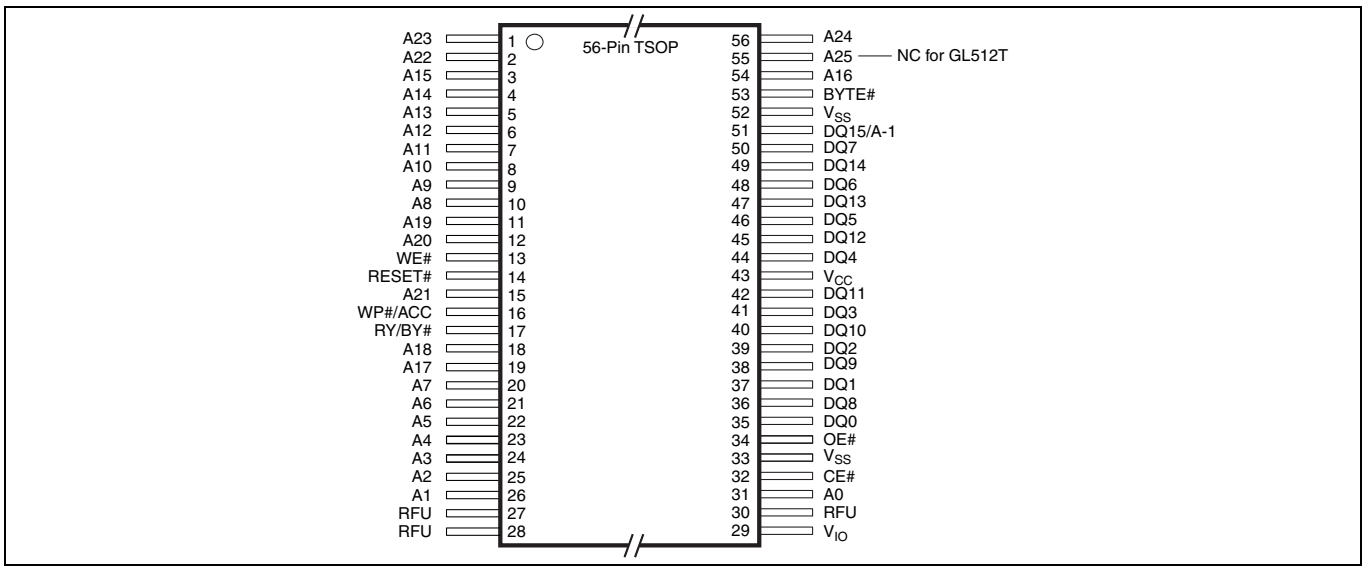
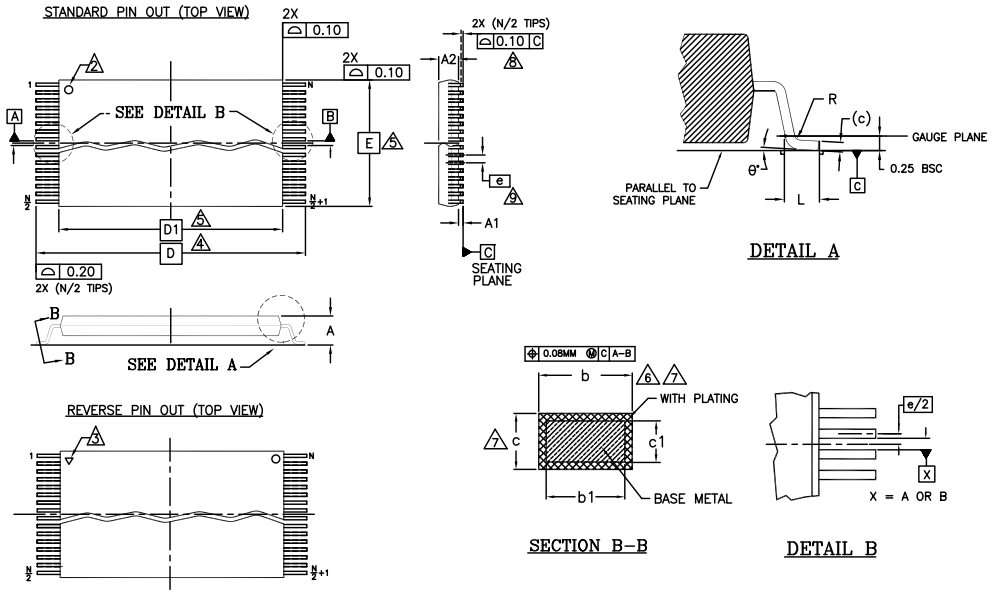



**Figure 34 56-pin standard TSOP**[168]

**Note**
168.Pin 27, 28, and 30 are Reserved for Future Use (RFU).

## 12.1.2 Physical diagram




| SYMBOL | DIMENSIONS | | |
|---|---|---|---|
| | MIN. | NOM. | MAX. |
| A | — | — | 1.20 |
| A1 | 0.05 | — | 0.15 |
| A2 | 0.95 | 1.00 | 1.05 |
| b1 | 0.17 | 0.20 | 0.23 |
| b | 0.17 | 0.22 | 0.27 |
| c1 | 0.10 | — | 0.16 |
| c | 0.10 | — | 0.21 |
| D | 20.00 BASIC | | |
| D1 | 18.40 BASIC | | |
| E | 14.00 BASIC | | |
| e | 0.50 BASIC | | |
| L | 0.50 | 0.60 | 0.70 |
| θ | 0° | — | 8 |
| R | 0.08 | — | 0.20 |
| N | 56 | | |

NOTES:

1. DIMENSIONS ARE IN MILLIMETERS (mm).
2. PIN 1 IDENTIFIER FOR STANDARD PIN OUT (DIE UP).
3. PIN 1 IDENTIFIER FOR REVERSE PIN OUT (DIE DOWN): INK OR LASER MARK.
4. TO BE DETERMINED AT THE SEATING PLANE -C- . THE SEATING PLANE IS DEFINED AS THE PLANE OF CONTACT THAT IS MADE WHEN THE PACKAGE LEADS ARE ALLOWED TO REST FREELY ON A FLAT HORIZONTAL SURFACE.
5. DIMENSIONS D1 AND E DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION ON E IS 0.15mm PER SIDE AND ON D1 IS 0.25mm PER SIDE.
6. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08mm TOTAL IN EXCESS OF b DIMENSION AT MAX. MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON LOWER RADIUS OR THE FOOT. MINIMUM SPACE BETWEEN PROTRUSION AND AN ADJACENT LEAD TO BE 0.07mm .
7. THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE LEAD BETWEEN 0.10mm AND 0.25mm FROM THE LEAD TIP.
8. LEAD COPLANARITY SHALL BE WITHIN 0.10mm AS MEASURED FROM THE SEATING PLANE.
9. DIMENSION "e" IS MEASURED AT THE CENTERLINE OF THE LEADS.
10. JEDEC SPECIFICATION NO. REF: MO-142(D)EC.

002-15549 *B

**Figure 35 56-pin TSOP (18.4 × 14.0 × 1.2 mm) package outline, 002-15549**

## 12.2 64-ball FBGA

### 12.2.1 Connection diagram

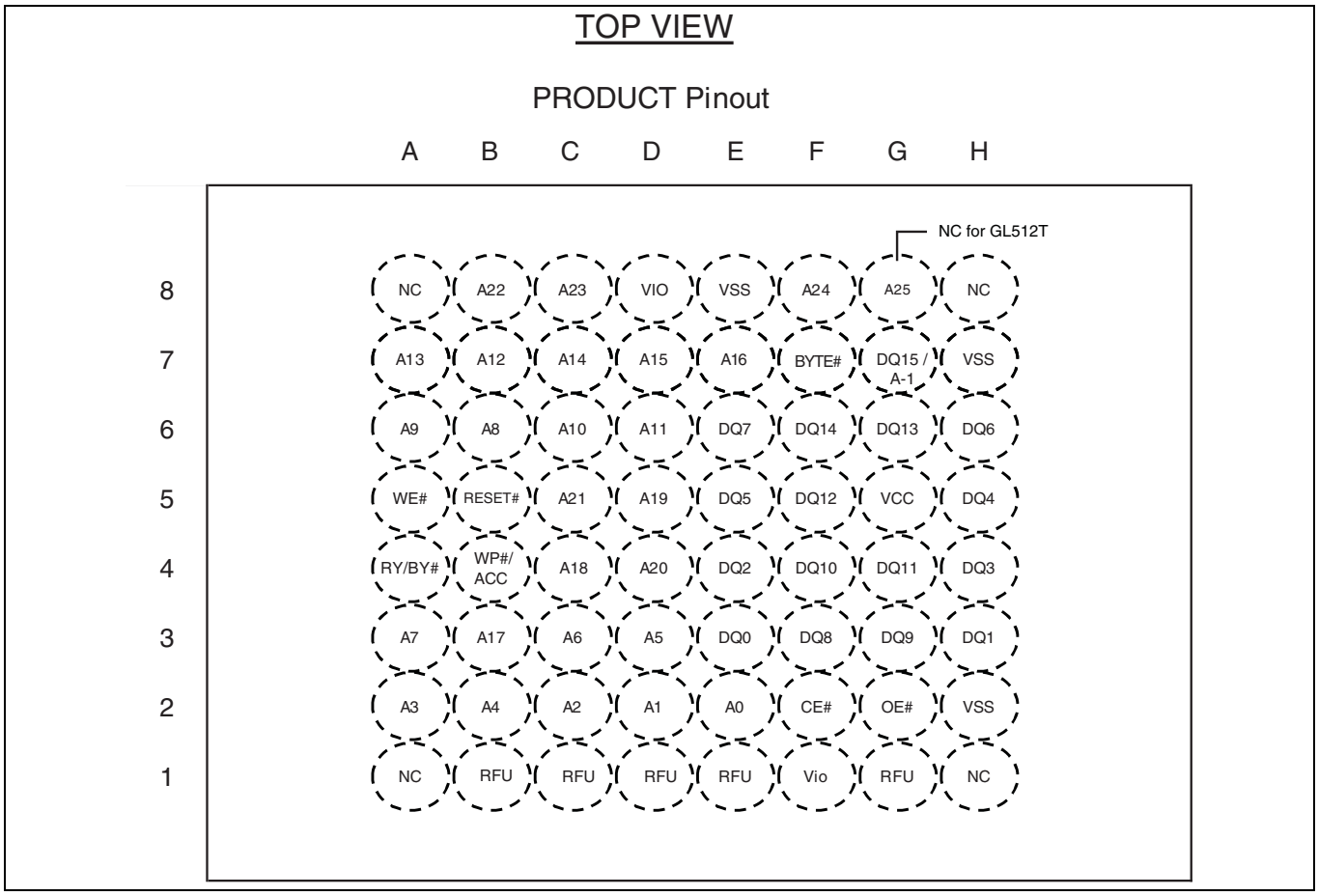
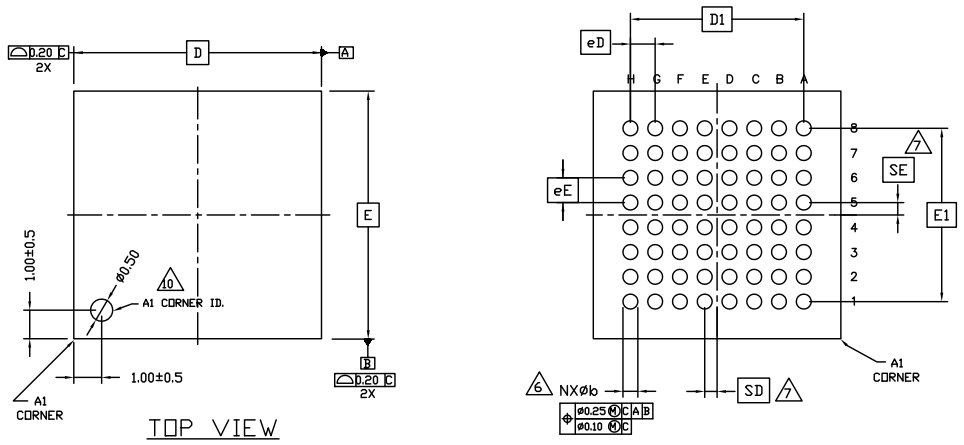



**Figure 36 64-ball fortified ball grid array**[169, 170]

**Notes**
169.Balls A1, A8, H1, and H8, No Connect (NC).
170.Balls B1, C1, D1, E1, and G1 Reserved for Future Use (RFU).

## 12.2.2 Physical diagram – LAE064




| SYMBOL | DIMENSIONS | | |
|---|---|---|---|
| | MIN. | NOM. | MAX. |
| A | - | - | 1.40 |
| A1 | 0.40 | - | - |
| A2 | 0.60 | - | - |
| D | 9.00 BSC. | | |
| E | 9.00 BSC. | | |
| D1 | 7.00 BSC. | | |
| E1 | 7.00 BSC. | | |
| MD | 8 | | |
| ME | 8 | | |
| N | 64 | | |
| Øb | 0.50 | 0.60 | 0.70 |
| eD | 1.00 BSC. | | |
| eE | 1.00 BSC. | | |
| SD/SE | 0.50 BSC. | | |

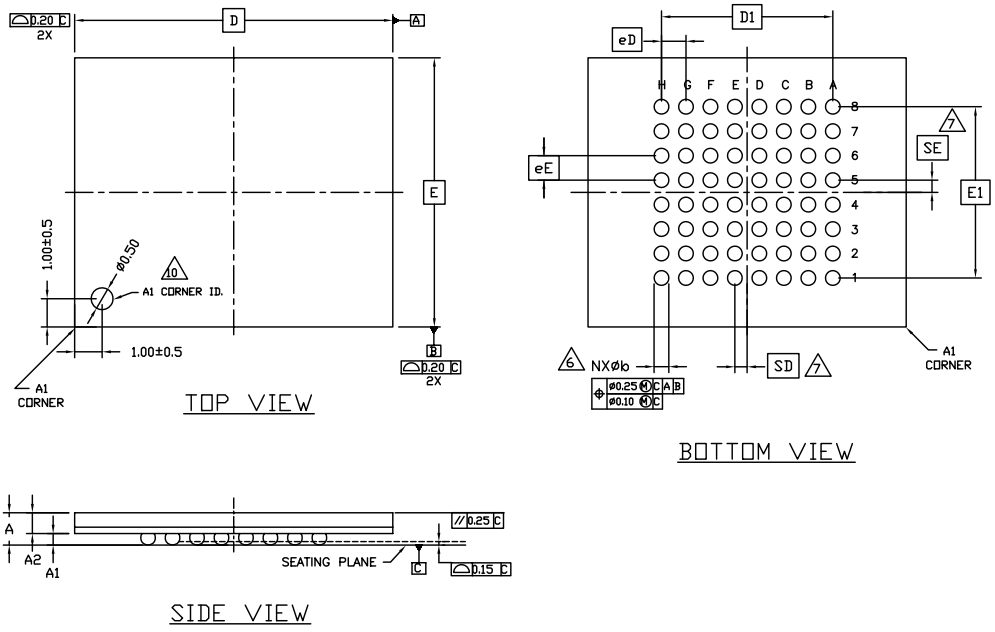
NOTES:

1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994 .
2. ALL DIMENSIONS ARE IN MILLIMETERS .
3. BALL POSITION DESIGNATION PER JEP95 SECTION 3, SPP-020 (RECTANGULAR) OR SPP-010 (SQUARE).
4. e REPRESENTS THE SOLDER BALL GRID PITCH .
5. SYMBOL "MD" IS THE BALL ROW MATRIX SIZE IN THE "D" DIRECTION.
   SYMBOL "ME" IS THE BALL COLUMN MATRIX SIZE IN THE "E" DIRECTION.
   N IS THE TOTAL NUMBER OF SOLDER BALLS.
6. DIMENSION "b" IS MEASURED AT THE MAXIMUM BALL DIAMETER IN A PLANE PARALLEL TO DATUM C .
7. "SD" AND "SE" ARE MEASURED WITH RESPECT TO DATUMS A AND B AND DEFINE THE POSITION OF THE CENTER SOLDER BALL IN THE OUTER ROW.
   WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW, "SD" OR "SE" = 0.
   WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, "SD" = eD/2 AND "SE" = eE/2.
8. "+" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.
9. A1 CORNER TO BE IDENTIFIED BY CHAMFER, LASER OR INK MARK, METALLIZED MARK INDENTATION OR OTHER MEANS.
10. JEDEC SPECIFICATION NO. REF : N/A

002-15537 *A

**Figure 37 64-ball FBGA (9.0 × 9.0 × 1.4 mm) package outline, 002-15537**

## 12.2.3 Physical diagram — LAA064




| SYMBOL | DIMENSIONS | | |
|---|---|---|---|
| | MIN. | NOM. | MAX. |
| A | - | - | 1.40 |
| A1 | 0.40 | - | - |
| A2 | 0.60 | - | - |
| D | 13.00 BSC. | | |
| E | 11.00 BSC. | | |
| D1 | 7.00 BSC. | | |
| E1 | 7.00 BSC. | | |
| MD | 8 | | |
| ME | 8 | | |
| N | 64 | | |
| Øb | 0.50 | 0.60 | 0.70 |
| eD | 1.00 BSC. | | |
| eE | 1.00 BSC. | | |
| SD/SE | 0.50 BSC. | | |

NOTES:

1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994 .
2. ALL DIMENSIONS ARE IN MILLIMETERS .
3. BALL POSITION DESIGNATION PER JEP95 SECTION 3, SPP-020 (RECTANGULAR) OR SPP-010 (SQUARE).
4. e REPRESENTS THE SOLDER BALL GRID PITCH .
5. SYMBOL "MD" IS THE BALL ROW MATRIX SIZE IN THE "D" DIRECTION.
   SYMBOL "ME" IS THE BALL COLUMN MATRIX SIZE IN THE "E" DIRECTION.
   N IS THE TOTAL NUMBER OF SOLDER BALLS.
6. DIMENSION "b" IS MEASURED AT THE MAXIMUM BALL DIAMETER IN A PLANE PARALLEL TO DATUM C .
7. "SD" AND "SE" ARE MEASURED WITH RESPECT TO DATUMS A AND B AND DEFINE THE POSITION OF THE CENTER SOLDER BALL IN THE OUTER ROW.
   WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW, "SD" OR "SE" = 0.
   WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, "SD" = eD/2 AND "SE" = eE/2.
8. "+" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.
9. A1 CORNER TO BE IDENTIFIED BY CHAMFER, LASER OR INK MARK, METALLIZED MARK INDENTATION OR OTHER MEANS.

002-15536 **

**Figure 38 64-ball FBGA (13.0 × 11.0 × 1.4 mm) package outline, 002-15536**

## 12.3 56-ball FBGA

### 12.3.1 Connection diagram

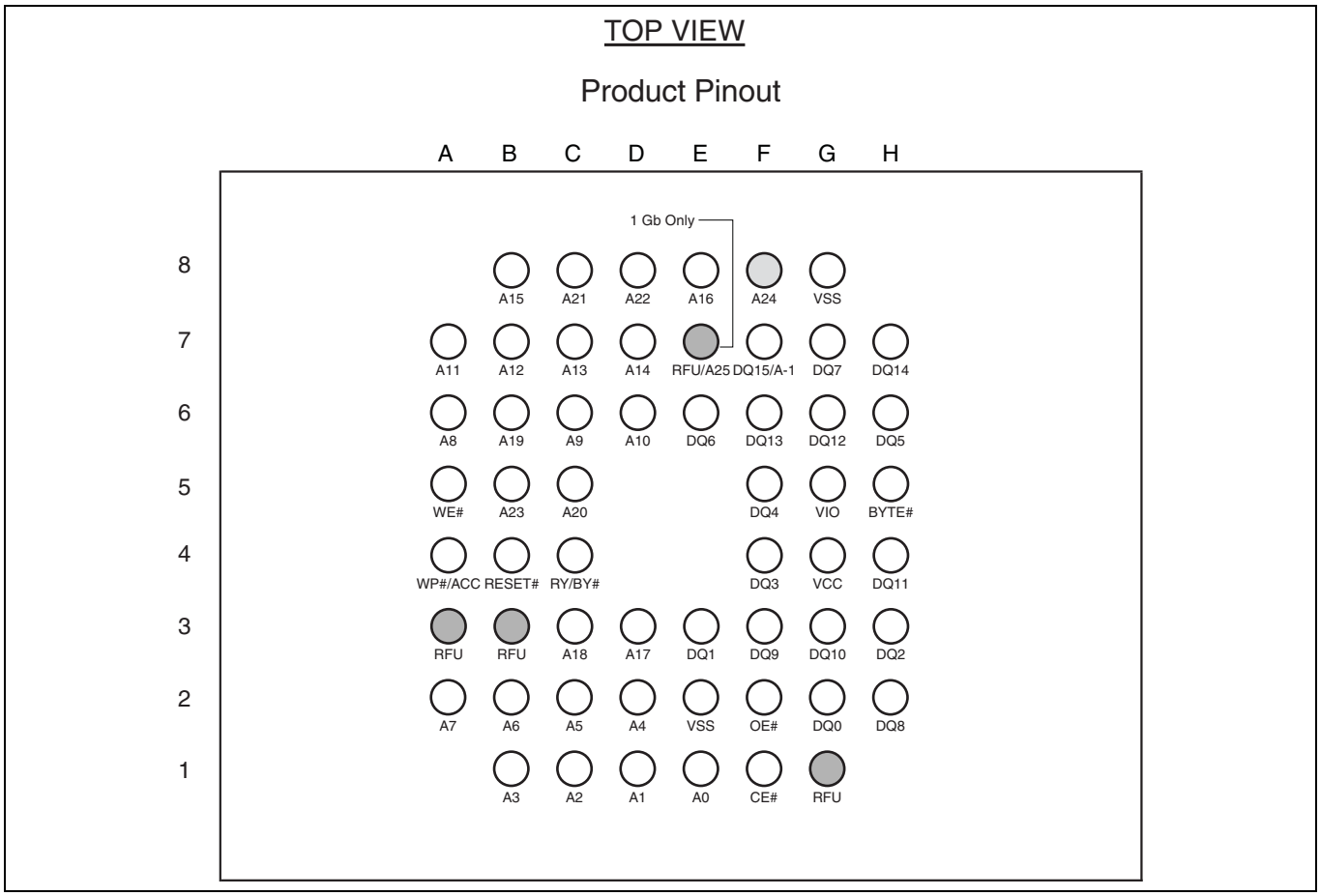
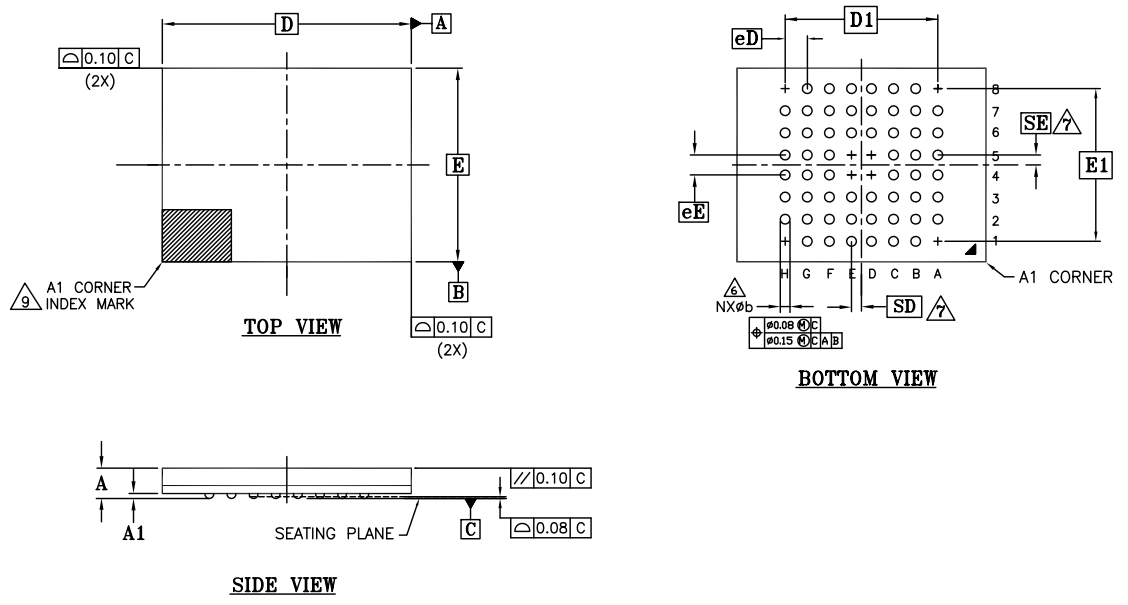



**Figure 39 56-ball fortified ball grid array**[171]

**Note**
171.Balls A3, B3, and G1 Reserved for Future Use (RFU).

## 12.3.2 Physical diagram




| SYMBOL | DIMENSIONS | | |
|---|---|---|---|
| | MIN. | NOM. | MAX. |
| A | - | - | 1.00 |
| A1 | 0.17 | - | - |
| D | 9.00 BSC. | | |
| E | 7.00 BSC. | | |
| D1 | 5.60 BSC. | | |
| E1 | 5.60 BSC. | | |
| MD | 8 | | |
| ME | 8 | | |
| n | 56 | | |
| Øb | 0.33 | - | 0.45 |
| eD/eE | 0.80 BSC. | | |
| SD/SE | 0.40 BSC. | | |

NOTES:

1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994 .
2. ALL DIMENSIONS ARE IN MILLIMETERS .
3. BALL POSITION DESIGNATION PER JEP95, SECTION 3, SPP-010/020.
4. e REPRESENTS THE SOLDER BALL GRID PITCH .
5. SYMBOL "MD" IS THE BALL ROW MATRIX SIZE IN THE "D" DIRECTION.
   SYMBOL "ME" IS THE BALL COLUMN MATRIX SIZE IN THE "E" DIRECTION.
   n IS THE TOTAL NUMBER OF POPULATED SOLDER BALLS FOR MATRIX SIZE MD AND ME.

6 DIMENSION "b" IS MEASURED AT THE MAXIMUM BALL DIAMETER IN A PLANE PARALLEL TO DATUM C.

7 "SD" AND "SE" ARE MEASURED WITH RESPECT TO DATUMS A AND B AND DEFINE THE POSITION OF THE CENTER SOLDER BALL IN THE OUTER ROW.
WHEN THERE IS AN ODD NUMBER OF SOLDER BALLS IN THE OUTER ROW, "SD" OR "SE" = 0.
WHEN THERE IS AN EVEN NUMBER OF SOLDER BALLS IN THE OUTER ROW, "SD" = eD/2 and "SE" = eE/2.

8. "+" INDICATES THE THEORETICAL CENTER OF DEPOPULATED BALLS.

9 A1 CORNER TO BE IDENTIFIED BY CHAMFER, LASER OR INK MARK, METALLIZED MARK INDENTATION OR OTHER MEANS.

002-15551 **

**Figure 40 56-ball FBGA (9.0 × 7.0 × 1.0 mm) package outline, 002-15551**

Special handling instructions for FBGA package

# 13 Special handling instructions for FBGA package

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

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

# 14 Ordering information

## 14.1 Valid combinations — standard

The recommended combinations table lists configurations planned to be available in volume. The table below will be updated as new combinations are released. Contact your local sales representative to confirm availability of specific combinations and to check on newly released combinations.

**Table 56 S29GL-T valid combinations for CFI version 1.3**

| S29GL-T valid combinations | | | | | |
|---|---|---|---|---|---|
| **Base OPN** | **Speed (ns)** | **Package and temperature**[172] | **Model number** | **Packing type**[173] | **Ordering part number (yy = Model number, x = Packing type)** |
| S29GL01GT | 100 | DHI, FAI, FHI, GHI, TFI | 03, 04 | 0, 3 | S29GL01GT10DHIyyx<br>S29GL01GT10FAIyyx<br>S29GL01GT10FHIyyx<br>S29GL01GT10GHIyyx<br>S29GL01GT10TFIyyx |
| | 110 | DHI, FAI, FHI, GHI, TFI | V3, V4 | | S29GL01GT11DHIyyx<br>S29GL01GT11FAIyyx<br>S29GL01GT11FHIyyx<br>S29GL01GT11GHIyyx<br>S29GL01GT11TFIyyx |
| | 110 | DHV, FHV, TFV | 03, 04 | | S29GL01GT11DHVyyx<br>S29GL01GT11FHVyyx<br>S29GL01GT11TFVyyx |
| | 120 | DHV, FHV, TFV | V3, V4 | | S29GL01GT12DHVyyx<br>S29GL01GT12FHVyyx<br>S29GL01GT12TFVyyx |
| | 120 | DHN, TFN | 03, 04 | | S29GL01GT12DHNyyxx<br>S29GL01GT12TFNyyxx |
| | 130 | DHN, TFN | V3, V4 | | S29GL01GT13DHNyyxx<br>S29GL01GT13TFNyyxx |
| S29GL512T | 100 | DHI, FAI, FHI, GHI, TFI | 03, 04 | 0, 3 | S29GL512T10DHIyyx<br>S29GL512T10FAIyyx<br>S29GL512T10FHIyyx<br>S29GL512T10GHIyyx<br>S29GL512T10TFIyyx |
| | 110 | DHI, FAI, FHI, GHI, TFI | V3, V4 | | S29GL512T11DHIyyx<br>S29GL512T11FAIyyx<br>S29GL512T11FHIyyx<br>S29GL512T11GHIyyx<br>S29GL512T11TFIyyx |
| | 110 | DHV, FHV, TFV | 03, 04 | | S29GL512T11DHVyyx<br>S29GL512T11FHVyyx<br>S29GL512T11TFVyyx |
| | 120 | DHV, FHV, TFV | V3, V4 | | S29GL512T12DHVyyx<br>S29GL512T12FHVyyx<br>S29GL512T12TFVyyx |
| | 120 | DHN, TFN | 03, 04 | | S29GL512T12DHNyyxx<br>S29GL512T12TFNyyxx |
| | 130 | DHN, TFN | V3, V4 | | S29GL512T13DHNyyxx<br>S29GL512T13TFNyyxx |

**Notes**
172. Additional speed, package, and temperature options maybe offered in the future. Check with your local sales representative for availability.
173. Package Type 0 is standard option.

**Table 57 S29GL-T valid combinations for CFI version 1.5**

| S29GL-T valid combinations | | | | | |
|---|---|---|---|---|---|
| **Base OPN** | **Speed (ns)** | **Package and temperature**[174] | **Model number** | **Packing type**[175] | **Ordering part number (yy = Model number, x = Packing type)** |
| S29GL01GT | 100 | DHI, FAI, FHI, GHI, TFI | 01, 02 | 0, 3 | S29GL01GT10DHIyyx<br>S29GL01GT10FAIyyx<br>S29GL01GT10FHIyyx<br>S29GL01GT10GHIyyx<br>S29GL01GT10TFIyyx |
| | 110 | DHI, FAI, FHI, GHI, TFI | V1, V2 | | S29GL01GT11DHIyyx<br>S29GL01GT11FAIyyx<br>S29GL01GT11FHIyyx<br>S29GL01GT11GHIyyx<br>S29GL01GT11TFIyyx |
| | 110 | DHV, FHV, TFV | 01, 02 | | S29GL01GT11DHVyyx<br>S29GL01GT11FHVyyx<br>S29GL01GT11TFVyyx |
| | 120 | DHV, FHV, TFV | V1, V2 | | S29GL01GT12DHVyyx<br>S29GL01GT12FHVyyx<br>S29GL01GT12TFVyyx |
| | 120 | DHN, TFN | 01, 02 | | S29GL01GT12DHNyyxx<br>S29GL01GT12FHNyyxx<br>S29GL01GT12TFNyyxx |
| | 130 | DHN, TFN | V1, V2 | | S29GL01GT13DHNyyxx<br>S29GL01GT13TFNyyxx |
| S29GL512T | 100 | DHI, FAI, FHI, GHI, TFI | 01, 02 | 0, 3 | S29GL512T10DHIyyx<br>S29GL512T10FAIyyx<br>S29GL512T10FHIyyx<br>S29GL512T10GHIyyx<br>S29GL512T10TFIyyx |
| | 110 | DHI, FAI, FHI, GHI, TFI | V1, V2 | | S29GL512T11DHIyyx<br>S29GL512T11FAIyyx<br>S29GL512T11FHIyyx<br>S29GL512T11GHIyyx<br>S29GL512T11TFIyyx |
| | 110 | DHV, FHV, TFV | 01, 02 | | S29GL512T11DHVyyx<br>S29GL512T11FHVyyx<br>S29GL512T11TFVyyx |
| | 120 | DHV, FHV, TFV | V1, V2 | | S29GL512T12DHVyyx<br>S29GL512T12FHVyyx<br>S29GL512T12TFVyyx |
| | 120 | DHN, TFN | 01, 02 | | S29GL512T12DHNyyxx<br>S29GL512T12TFNyyxx |
| | 130 | DHN, TFN | V1, V2 | | S29GL512T13DHNyyxx<br>S29GL512T13TFNyyxx |

**Notes**
174. Additional speed, package, and temperature options maybe offered in the future. Check with your local sales representative for availability.
175. Package Type 0 is standard option.

## 14.2 Valid combinations — automotive grade / AEC-Q100

The table below lists configurations that are automotive grade / AEC-Q100 qualified and are planned to be available in volume. The table will be updated as new combinations are released. Consult your local sales representative to confirm availability of specific combinations and to check on newly released combinations.

Production part approval process (PPAP) support is only provided for AEC-Q100 grade products.

Products to be used in end-use applications that require ISO/TS-16949 compliance must be AEC-Q100 grade products in combination with PPAP. Non–AEC-Q100 grade products are not manufactured or documented in full compliance with ISO/TS-16949 requirements.

AEC-Q100 grade products are also offered without PPAP support for end-use applications that do not require ISO/TS-16949 compliance.

**Table 58 S29GL-T valid combinations for CFI version 1.3 — automotive grade / AEC-Q100**

| S29GL-T valid combinations — automotive grade / AEC-Q100 | | | | | |
|---|---|---|---|---|---|
| **Base OPN** | **Speed (ns)** | **Package and temperature** | **Model number** | **Packing type** | **Ordering part number (yy = Model number, x = Packing type)** |
| S29GL01GT | 100 | DHA, FHA, TFA | 03, 04 | 0, 3 | S29GL01GT10DHAyyx<br>S29GL01GT10FHAyyx<br>S29GL01GT10TFAyyx |
| | 110 | DHA, FHA, TFA | V3, V4 | | S29GL01GT11DHAyyx<br>S29GL01GT11FHAyyx<br>S29GL01GT11TFAyyx |
| | 110 | DHB, FHB, TFB | 03, 04 | | S29GL01GT11DHByyx<br>S29GL01GT11FHByyx<br>S29GL01GT11TFByyx |
| | 120 | DHB, FHB, TFB | V3, V4 | | S29GL01GT12DHByyx<br>S29GL01GT12FHByyx<br>S29GL01GT12TFByyx |
| S29GL512T | 100 | DHA, FHA, TFA | 03, 04 | 0, 3 | S29GL512T10DHAyyx<br>S29GL512T10FHAyyx<br>S29GL512T10TFAyyx |
| | 110 | DHA, FHA, TFA | V3, V4 | | S29GL512T11DHAyyx<br>S29GL512T11FHAyyx<br>S29GL512T11TFAyyx |
| | 110 | DHB, FHB, TFB | 03, 04 | | S29GL512T11DHByyx<br>S29GL512T11FHByyx<br>S29GL512T11TFByyx |
| | 120 | DHB, FHB, TFB | V3, V4 | | S29GL512T12DHByyx<br>S29GL512T12FHByyx<br>S29GL512T12TFByyx |

**Table 59 S29GL-T valid combinations for CFI version 1.5 — automotive grade / AEC-Q100**

| S29GL-T valid combinations — automotive grade / AEC-Q100 | | | | | |
|---|---|---|---|---|---|
| **Base OPN** | **Speed (ns)** | **Package and temperature** | **Model number** | **Packing type** | **Ordering part number (yy = Model number, x = Packing type)** |
| S29GL01GT | 100, 110 | DHA, FHA, TFA | 01, 02 | 0, 3 | S29GL01GT10DHAyyx<br>S29GL01GT10FHAyyx<br>S29GL01GT10TFAyyx<br>S29GL01GT11DHAyyx<br>S29GL01GT11FHAyyx<br>S29GL01GT11TFAyyx |
| | 110 | DHA, FHA, TFA | V1, V2 | | S29GL01GT11DHAyyx<br>S29GL01GT11FHAyyx<br>S29GL01GT11TFAyyx |
| | 110 | DHB, FHB, TFB | 01, 02 | | S29GL01GT11DHByyx<br>S29GL01GT11FHByyx<br>S29GL01GT11TFByyx |
| | 120 | DHB, FHB, TFB | V1, V2 | | S29GL01GT12DHByyx<br>S29GL01GT12FHByyx<br>S29GL01GT12TFByyx |
| S29GL512T | 100 | DHA, FHA, TFA | 01, 02 | 0, 3 | S29GL512T10DHAyyx<br>S29GL512T10FHAyyx<br>S29GL512T10TFAyyx |
| | 110 | DHA, FHA, TFA | V1, V2 | | S29GL512T11DHAyyx<br>S29GL512T11FHAyyx<br>S29GL512T11TFAyyx |
| | 110 | DHB, FHB, TFB | 01, 02 | | S29GL512T11DHByyx<br>S29GL512T11FHByyx<br>S29GL512T11TFByyx |
| | 120 | DHB, FHB, TFB | V1, V2 | | S29GL512T12DHByyx<br>S29GL512T12FHByyx<br>S29GL512T12TFByyx |

Ordering information

The ordering part number for the General Market device is formed by a valid combination of the following:

**S29GL01GT 10 D H I 01 0**

**Packing type**
0 = Tray
3 = 13” Tape and reel

**Model number (CFI Version, $V_{IO}$, and $V_{CC}$ Range)**
CFI Version 1.3
03 = $V_{IO}$ = $V_{CC}$ = 2.7 V to 3.6 V, highest address sector protected
04 = $V_{IO}$ = $V_{CC}$ = 2.7 V to 3.6 V, lowest address sector protected
V3 = $V_{IO}$ = 1.65 to $V_{CC}$, $V_{CC}$ = 2.7 V to 3.6 V, highest address sector protected
V4 = $V_{IO}$ = 1.65 to $V_{CC}$, $V_{CC}$ = 2.7 V to 3.6 V, lowest address sector protected

CFI Version 1.5
01 = $V_{IO}$ = $V_{CC}$ = 2.7 V to 3.6 V, highest address sector protected
02 = $V_{IO}$ = $V_{CC}$ = 2.7 V to 3.6 V, lowest address sector protected
V1 = $V_{IO}$ = 1.65 to $V_{CC}$, $V_{CC}$ = 2.7 V to 3.6 V, highest address sector protected
V2 = $V_{IO}$ = 1.65 to $V_{CC}$, $V_{CC}$ = 2.7 V to 3.6 V, lowest address sector protected

**Temperature range**
I = Industrial (–40°C to +85°C)
V = Industrial Plus (–40°C to +105°C)
N = Extended (–40°C to +125°C)
A = Automotive, AEC-Q100 Grade 3 (–40°C to +85°C)
B = Automotive, AEC-Q100 Grade 2 (–40°C to +105°C)

**Package materials set**
A = Not Lead (Pb)-Free
F = Lead Free (Pb-Free)
H = Low Halogen, Pb-Free

**Package type**
D = Fortified ball-grid array package (LAE064) 9 mm x 9 mm
F = Fortified ball-grid array package (LAA064) 13 mm x 11 mm
G = Fortified ball-grid array package (VBU056) 9 mm x 7 mm
T = Thin small outline package (TSOP) standard pinout

**Speed option**
10 = 100 ns random access time
11 = 110 ns random access time
12 = 120 ns random access time
13 = 130 ns random access time

**Device number/description**
S29GL01GT, S29GL512T
3.0 V core, with $V_{IO}$ option, 1024, 512 Megabit page-mode flash memory,
Manufactured on 45-nm MIRRORBIT™ process technology

# Revision history

| Document revision | Date | Description of changes |
|---|---|---|
| ** | 01/19/2015 | Initial release. |
| *A | 2015-05-08 | Performance Summary: Typical Program and Erase Rates table: updated Sector Erase for –40°C to +85°C<br>Embedded Algorithm Performance Table: Embedded Algorithm Characteristics (–40°C to +85°C) table: updated Sector Erase Time, Chip Erase, and Max Single Word Programming Time<br>Device ID and Common Flash Interface (ID-CFI) ASO Map: CFI System Interface String table: updated '(SA) + 0023h' Data |
| *B | 2015-07-29 | Performance Summary: Typical Program and Erase Rates table: Updated Sector Erase for –40°C to +105°C<br>Embedded Algorithm Performance Table: Embedded Algorithm Characteristics (–40°C to +105°C) table: updated Sector Erase Time, Chip Erase, Single Word Programming Time, Buffer Programming Time, Effective Write Buffer Program Operation per Word, and Sector Programming Time 128 kB<br>Device ID and Common Flash Interface (ID-CFI) ASO Map: CFI System Interface String table: updated Data for Word Address (SA) + 0023h and (SA) + 0024h |
| *C | 2015-08-24 | Updated to Cypress template. |
| *D | 2015-10-07 | Added a note on Errata in page 1.<br>Added Errata. |
| *E | 2015-12-08 | Added Extended Temperature Range related information in all instances across the document.<br>Removed note on Errata in page 1.<br>Updated **Ordering information**:<br>Updated **Table 56**: Updated details in "Package and Temperature" column and "Ordering Part Number" column.<br>Removed Errata. |
| *F | 2016-03-09 | Updated **Performance summary**:<br>Replaced "Performance Summary Industrial Plus Temperature Range" with "Performance Summary Extended Temperature Range" in table title.<br>Replaced "200 µA" with "215 µA" in "–40°C to +125°C" column corresponding to "Standby" operation in "Maximum Current Consumption" table.<br>Updated **Product overview**:<br>Updated **Table 1**: Corrected typos in "×8" column.<br>Updated description below **Table 1** (Removed (A7 = 0 or A7 =1) from 7th paragraph of the section).<br>Updated **Data protection**:<br>Updated **Sector protection methods**:<br>Updated **Password protection mode**:<br>Updated **PPB password protection mode**:<br>Updated description.<br>Updated **Timing specifications**:<br>Updated **AC characteristics**:<br>Updated **Asynchronous read operations**:<br>Added **Table 51** and **Table 52**.<br>Updated **Physical interface**:<br>Updated **64-ball FBGA**:<br>Updated **Physical diagram – LAE064**:<br>Updated **Figure 37** (Updated with the latest revision).<br>Updated **Ordering information**:<br>No change in part numbers.<br>Updated Ordering Code Definitions below **Table 56**. |

Revision history

| Document revision | Date | Description of changes |
|---|---|---|
| *G | 2016-10-27 | Added “Automotive, AEC-Q100 Grade 3” and “Automotive, AEC-Q100 Grade 2” Temperature Range related information in all instances across the document.<br>Added “ECC” related information in all instances across the document.<br>Updated **Product overview**:<br>Updated **Table 1**.<br>Updated **Address space overlays**:<br>Added **ECC status ASO**.<br>Updated **Embedded operations**:<br>Added **Automatic ECC**.<br>Updated **Error types and clearing procedures**:<br>Removed Note “Under worst case conditions of 90°C, $V_{CC}$ = 2.70 V, 100,000 cycles, and a random data pattern.” below **Table 18**, **Table 19** and **Table 20**.<br>Removed Note “Data retention of 20 years is based on 1K erase cycles.” below **Table 18**, **Table 19** and **Table 20**.<br>Added **Data integrity**.<br>Updated **Software interface reference**:<br>Updated **Command summary**:<br>Updated **Table 23**:<br>Added “ECC ASO” Command Sequence and its details.<br>Updated **Table 24**:<br>Added “ECC ASO” Command Sequence and its details.<br>Updated **Device ID and Common Flash Interface (ID-CFI) ASO map**:<br>Updated **Table 29**:<br>Updated details in “Description” column corresponding to Word Address “(SA) + 0044h”.<br>Updated **Electrical specifications**:<br>Added **Thermal resistance**.<br>Updated **Ordering information**:<br>Removed “Valid Combinations”.<br>Added **Valid combinations — standard**.<br>Added **Valid combinations — automotive grade / AEC-Q100**.<br>Updated Ordering Code Definitions.<br>Updated **Other resources**:<br>Removed “Software”.<br>Removed “Application Notes”.<br>Added “Cypress Flash Memory Roadmap”.<br>Added “Links to Software”.<br>Added “Links to Application Notes”.<br>Updated to new template. |
| *H | 2017-01-18 | Updated **Electrical specifications**:<br>Updated **DC characteristics**:<br>Updated **Table 39**: Added minimum values for $V_{IL}$, $V_{IH}$, $V_{HH}$, $V_{OH}$, $V_{LKO}$ parameters.<br>Updated to new template. |
| *I | 2017-05-24 | Corrected the number of cycles mentioned for ECC ASO exit command.<br>Updated Cypress Logo and Copyright. |
| *J | 2018-09-03 | Updated **Product overview**:<br>Updated **Table 1**.<br>Updated **Software interface reference**:<br>Updated **Device ID and Common Flash Interface (ID-CFI) ASO map**:<br>Updated **Table 28**.<br>Updated **Electrical specifications**:<br>Updated **Thermal resistance**:<br>Updated **Table 33**.<br>Updated to new template. |
| *K | 2018-10-17 | Updated **Electrical specifications**:<br>Updated **Thermal resistance**:<br>Updated **Table 33** (Changed value of Theta JA from 46 °C/W to 43.5 °C/W in “TS056” column corresponding to 1G).<br>Updated **Ordering information**:<br>Updated **Table 57**. |

Revision history

| Document revision | Date | Description of changes |
|---|---|---|
| *L | 2019-04-05 | Updated **Address space overlays**:<br>Updated **Flash memory array**:<br>Updated **Table 2** (Updated details under “Address Range (8-Bit)” column).<br>Updated **Table 3** (Updated details under “Address Range (8-Bit)” column).<br>Updated **Timing specifications**:<br>Updated **AC characteristics**:<br>Updated **Asynchronous read operations**:<br>Updated **Table 47** through **Table 52**:<br>Changed name of parameter from $t_{ASH}$ to $t_{AHT}$.<br>Removed $t_{OEP}$, $t_{OEC}$ parameters and their details.<br>Updated **Physical interface**:<br>Updated **56-pin TSOP**:<br>Updated **Physical diagram**:<br>Replaced existing spec with 002-15549 *B.<br>Updated **64-ball FBGA**:<br>Updated **Physical diagram – LAE064**:<br>Replaced existing spec with 002-15537 *A.<br>Updated **Physical diagram — LAA064**:<br>Replaced existing spec with 002-15536 **.<br>Updated **56-ball FBGA**:<br>Updated **Physical diagram**:<br>Replaced existing spec with 002-15551 **.<br>Updated Copyright information. |
| *M | 2022-07-25 | Updated Document Title to read as “S29GL01GT, S29GL512T, 1 Gb (128 MB)/512 Mb (64 MB) GL-T MIRRORBIT™ Flash Parallel, 3.0 V”.<br>Replaced “MirrorBit® Eclipse” with “MIRRORBIT™” in all instances across the document.<br>Updated **Features**:<br>Replaced “Distinctive characteristics” with “Features” in heading.<br>Updated **Address space overlays**:<br>Updated **Device ID and CFI (ID-CFI) ASO**:<br>Updated **Common flash memory interface**:<br>Updated description.<br>Updated **Embedded operations**:<br>Updated **Status monitoring**:<br>Updated **Status Register**:<br>Updated **Table 16**.<br>Updated **Data integrity**:<br>Updated **Erase endurance**:<br>Updated **Table 21**.<br>Updated **Data retention**:<br>Updated description.<br>Removed “Software interface”.<br>Removed “Hardware interface”.<br>Updated **Electrical specifications**:<br>Updated **Thermal resistance**:<br>Updated **Table 33**.<br>Removed “Other resources”.<br>Migrated to Infineon template. |

**Trademarks**
All referenced product or service names and trademarks are the property of their respective owners.

**Edition 2022-07-25**
**Published by**
**Infineon Technologies AG**
**81726 Munich, Germany**

**© 2022 Infineon Technologies AG. All Rights Reserved.**

**Do you have a question about this document?**
**Go to www.infineon.com/support**

**Document reference**
**002-00247 Rev. *M**

**IMPORTANT NOTICE**
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

**WARNINGS**
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.

## Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- View S29GL512T12DHN020 on WIN SOURCE
- Cypress Semiconductor Corp Information

## Optimize Your Supply Chain with WIN SOURCE Solutions

- Global Sourcing Solution
- Obsolete Management
- Cost Control Management
- Shortage Management
- Alternative Solution
- Excess Inventory Management