# AT25XE161D

16-Mbit 1.65 V – 3.6 V Range SPI Serial Flash Memory with Multi I/O Support

## Features

- Voltage Range: 1.65 V - 3.6 V
- Serial Peripheral Interface (SPI) compatible
  - Supports SPI modes 0 and 3
  - Supports dual output operation (1-1-2)
  - Supports quad output operation (1-1-4)
  - Supports quad I/O / XiP operation (1-4-4 and 0-4-4)
- 133 MHz maximum operating frequency
- Flexible, optimized erase architecture
  - 256-Byte page erase
  - 4-kByte block erase
  - 32-kByte block erase
  - 64-kByte block erase
  - Full chip erase
- Flexible programming
  - Byte/Page program (1 to 256 Bytes)
  - Sequential program mode capability
- Erase program suspend resume
- 1 x 128-byte factory-programmed unique identifier
- 3 x 128-byte, One Time Programmable (OTP) security registers
- Software controlled *Reset* and *Terminate* commands
- Multiple memory protection schemes
  - Individual block protection
  - User-definable protected area at start or end of memory array (default)
- Read-modify-write (RMW) command – emulates SRAM write in a single command
- Active status interrupt – automatically generates interrupt to host when RDY/BSY bit is cleared
- Software controlled *Reset* and *Terminate* commands
- Hardware reset pin option
- JEDEC hardware reset
- Non-volatile / volatile Status Registers
- JEDEC standard manufacturer and device ID
- Serial Flash Discoverable Parameters (SFDP)
- Low power dissipation:
  - 30 µA standby current (typical)
  - 8.2 µA Deep Power-Down (DPD) current (typical)
  - 7 nA Ultra Deep Power Down (UDPD) current (typical)
  - 8.3 mA active read current (1-1-1 — 104 MHz)
  - 10.1 mA program current
  - 10.7 mA erase current
- User-configurable and auto I/O pin drive strength levels
- Endurance: 100,000 program/erase cycles
- Data Retention: 20 years
- -40 °C to +85 °C operation
- Industry standard green (Pb/Halide-free/RoHS Compliant) Package Options
  - 8-lead SOIC (150-mil)
  - 8-lead SOIC (208-mil)
  - 8-pad Ultra-thin DFN (2 x 3 x 0.6 mm)
  - 8-ball WLCSP (3 x 2 x 3 ball matrix)
  - Die/Wafer — Contact Renesas Electronics for more information

# Contents

## Figures

# Tables

# 1. Product Overview

The AT25XE161D is a 16-Mbit serial peripheral interface (SPI) Flash memory device designed for use in a wide variety of high-volume industrial, consumer, and connected applications.

It can be used for storing program memory that is copied from Flash memory into embedded or external RAM during system boot; it also can be used for directly executing program code from Flash memory (eXecute in Place [XiP]) and where small amounts of data are stored and updated locally in Flash memory.

XiP is specifically supported by features that enhance read speed:

- Quad-SPI, which allows reading four bits in one clock cycle.
- Continuous read mode (0-4-4 command format), which removes the need to send a command opcode.
- High SPI clock frequency.

These features allow fast response from the Flash memory whenever the host must fetch commands or data from it.

AT25XE161D has page erase feature which can erase a block as small as 256 bytes. This makes the write operation much more efficient, especially for storing small quantities of user or system data for data logging.

The AT25XE161D supports a wide Vcc voltage range: 1.65 V to 3.6 V, which is ideal for battery operated systems.

# 2. Block Diagram

Figure 1 shows a block diagram of the AT25XE161D device. The *Interface Control Logic* block connects to external device through a set of pins. The state of these pins is distributed *Interface Control Logic* block to other blocks as necessary. The design also contains a 16 Mbit serial Flash memory array, a 256-byte SRAM buffer, and an I/*O Interface Unit* that operates depending on the type of data transfer mode.

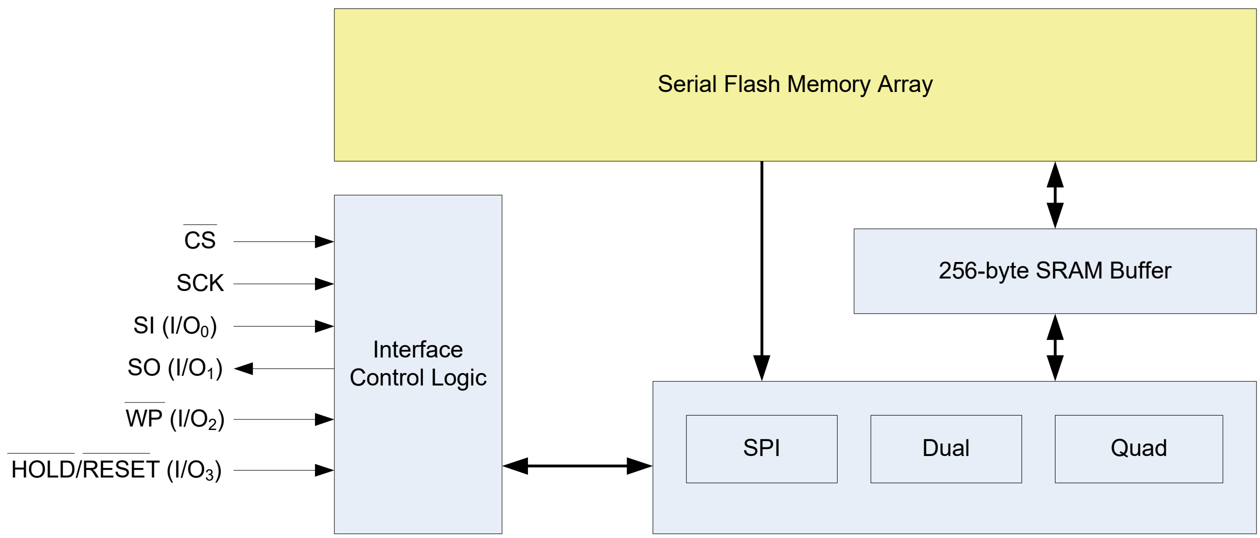
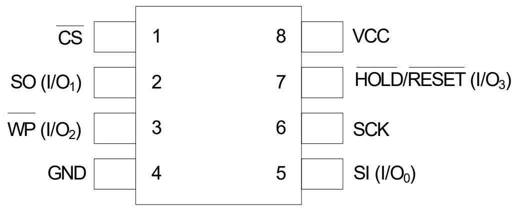



**Figure 1. Block Diagram**

# 3. Package Pinouts

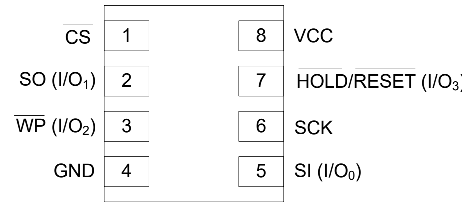
Figure 2 show the package pinouts for the following devices. Note that the Die Wafer Scale option is not shown.




8-lead SOIC Package (150-mil)

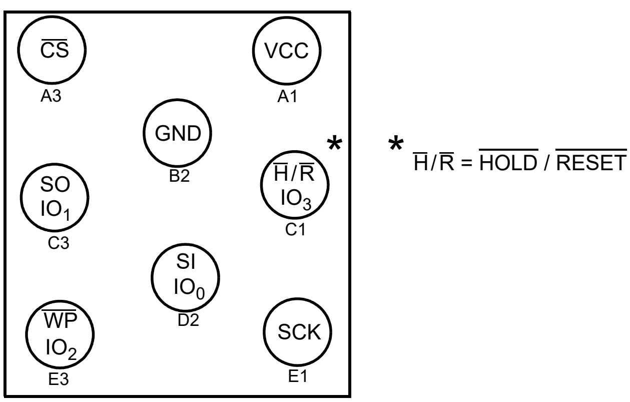
8-lead EIAJ SOIC Package (208-mil)

(Top View)




8-pad Ultra-Thin DFN Package

2 x 3 x 0.6 mm (Top View)




8-ball WLCSP Package

3 x 2 x 3 ball matrix (bottom view)

**Figure 2. Memory Package Types**

**Table 1. Pin Descriptions**

| Symbol | Name and Function | Asserted State | Type |
|---|---|---|---|
| $\overline{CS}$ | **CHIP SELECT**<br>Asserting the $\overline{CS}$ pin selects the device. When the $\overline{CS}$ pin is deasserted, the device is be deselected and normally be placed in standby mode (not Deep Power-Down mode), and the SO pin is in a high-impedance state. When the device is deselected, data are not accepted on the SI pin.<br>A high-to-low transition on the $\overline{CS}$ pin is required to start an operation, and a low-to-high transition is required to end an operation. When ending an internally self-timed operation such as a program or erase cycle, the device does not enter the standby mode until the completion of the operation.<br>To ensure correct power-up sequencing, it is recommended to add a 10k Ohm pull-up resistor from $\overline{CS}$ to $V_{CC}$. This ensures $\overline{CS}$ ramps together with $V_{CC}$ during power-up. | Low | Input |
| SCK | **SERIAL CLOCK**<br>This pin provides a clock to the device and controls the flow of data to and from the device. Command, address, and input data present on the SI pin is always latched in on the rising edge of SCK, while output data on the SO pin is always clocked out on the falling edge of SCK. | - | Input |
| SI ($I/O_0$) | **SERIAL INPUT**<br>The SI pin shifts data into the device. The SI pin is used for all data input including command and address sequences. Data on the SI pin is always latched in on the rising edge of SCK.<br>With the Multi I/O Read commands, the SI pin becomes an output pin ($I/O_0$) in conjunction with other pins to allow either two or four bits of data on ($I/O_{1:0}$ or $I/O_{3:0}$) to be clocked out on every falling edge of SCK.<br>Data present on the SI pin is ignored whenever the device is deselected ($\overline{CS}$ is deasserted and the device is in the reset condition). | - | Input/ Output |
| SO ($I/O_1$) | **SERIAL OUTPUT**<br>The SO pin shifts data out from the device. Data on the SO pin is always clocked out on the falling edge of SCK.<br>With the Multi I/O Read commands, the SO pin remains an output pin ($I/O_1$) in conjunction with other pins to allow either two or four bits of data on ($I/O_{1:0}$ or $I/O_{3:0}$) to be clocked out on every falling edge of SCK.<br>The SO pin is in a high-impedance state whenever the device is deselected ($\overline{CS}$ is deasserted and the device is in the reset condition). | - | Input/ Output |
| $\overline{WP}$ ($I/O_2$) | **WRITE PROTECT**<br>This pin is used either for write-protection, in which case it is referred to as $\overline{WP}$, or as one of the quad-SPI I/O pins, in which case it is referred to as $IO_2$.<br>When the Quad Enable (QE) bit of Status Register 2 is 0, and the SRP1 and SRP0 bits are 0 and 1, respectively, the pin can be used for write-protection. It then can be asserted (driven low) to protect the Status Registers from modification.<br>When the QE bit of Status Register 2 is 1, quad-SPI communication is enabled, and the pin is used as I/O pin $IO_2$ in any command that makes use of quad-SPI. In this setting, do not use the pin for write-protection.<br>The $\overline{WP}$ pin is internally pulled-high and can be left floating if not used. | Low | Input/O utput |

**Table 1. Pin Descriptions** (Continued)

| Symbol | Name and Function | Asserted State | Type |
|---|---|---|---|
| $\overline{\text{HOLD}}$ / RESET ($I/O_3$) | **$\overline{\text{HOLD}}$ / $\overline{\text{RESET}}$**<br>This pin is used either for pausing communication ($\overline{\text{HOLD}}$), as a hardware reset pin ($\overline{\text{RESET}}$), or as one of the quad-SPI I/O pins ($IO_3$).<br>When the Quad Enable (QE) bit of Status Register 2 is 0, this pin is used either as a $\overline{\text{HOLD}}$ or $\overline{\text{RESET}}$ pin, depending on the value of the HOLD/RESET bit of Status Register 3. When the QE bit of Status Register 2 is 1, quad-SPI communication is enabled, and the pin is used as I/O pin $IO_3$ in any command that makes use of quad-SPI. In this setting, do not use the pin for hold or reset.<br>The $\overline{\text{HOLD}}$ pin is used to pause a SPI sequence without resetting the clocking sequence. To enable the HOLD mode, the $\overline{\text{CS}}$ must be low. The HOLD mode effect is on with the falling edge of the $\overline{\text{HOLD}}$ signal with SCK being low. The HOLD mode ends on the rising edge of the $\overline{\text{HOLD}}$ signal with SCK being low.<br>When configured as $\overline{\text{RESET}}$, this pin can be used to perform a hardware reset on the device.<br>The $\overline{\text{HOLD}}$/$\overline{\text{RESET}}$/$IO_3$ pin is internally pulled-high and can be left floating if not used. | Low | Input/Output |
| $V_{CC}$ | **DEVICE POWER SUPPLY**<br>The $V_{CC}$ pin supplies the source voltage to the device.<br>Operations at invalid $V_{CC}$ voltages can produce spurious results; do not attempt this. | - | Power |
| GND | **GROUND**<br>The ground reference for the power supply. Connect GND to the system ground. | - | Power |

# 4. Device Operation

This section describes the various data transfer modes supported by the device, as well as other system operations.

## 4.1 Data Transfer Modes

The JEDEC specification uses a numerical system to indicate the type of transfer for a given command. The nomenclature for this system is defined as *(x-y-z)* to indicate the number of active pins used for the command (x), address (y), and data (z). For an example, a designation of 1-1-2 indicates that one pin (SI) transfers the command, one pin (SI) is for the address, and two pins (SI and SO) are for data. The AT25XE161D supports the following transfer types.

**Table 2. Bus Transfer Types**

| Transfer Type | Transfer Name | Command | Pin(s) Used for Command | Address | Pin(s) Used for Address | Data | Pin(s) Used for Data |
|---|---|---|---|---|---|---|---|
| 1-0-0 | SPI | Yes | SI | No | -- | No | -- |
| 1-1-0 | SPI | Yes | SI | Yes | SI | No | -- |
| 1-0-1 | SPI | Yes | SI | No | -- | Yes | SI (write)<br>SO (read) |
| 1-1-1 | SPI | Yes | SI | Yes | SI | Yes | SI (write)<br>SO (read) |
| 1-1-2 | Dual Output | Yes | SI | Yes | SI | Yes | SI, SO |
| 1-1-4 | Quad Output | Yes | SI | Yes | SI | Yes | SI, SO, $\overline{WP}$, $\overline{HOLD}$ |
| 1-4-4 | Quad I/O | Yes | SI | Yes | SI, SO, WP, $\overline{HOLD}$ | Yes | SI, SO, $\overline{WP}$, $\overline{HOLD}$ |
| 0-4-4 | XiP | No | -- | Yes | SI, SO, WP, $\overline{HOLD}$ | Yes | SI, SO, $\overline{WP}$, $\overline{HOLD}$ |

As shown in the table above, the AT25XE161D supports the following transfer formats, which are described in the following subsections.

- Standard SPI Operation
- Dual Output Operation
- Quad Output Operation
- Quad I/O Operation
- XiP Operation

## 4.2 Standard SPI Operation

Standard SPI transfers are divided into three elements; command, address, and data. SPI mode support the following four transfer types, as described in Table 2.

- Command only, no address or data (1-0-0)
- Command and address only, no data (1-1-0)
- Command and data only, no address (1-0-1)
- Command, address, and data (1-1-1)

For standard SPI transfers, command and address are always transferred on the SI pin. For write operations, data is also transferred on the SI pin. For read operations, data is transferred on the SO pin.

The AT25XE161D supports the two most common SPI modes, 0 and 3, meaning that data is always latched on the rising edge of SCK and always output on the falling edge of SCK.

### 4.2.1 Command Only, No Address or Data (1-0-0)

The following diagram shows a command-only transfer. In this type of transfer no address or data are required. An example is the *Chip Erase* (60h/C7h) command. A 1-0-0 transfer type is shown in Figure 3.

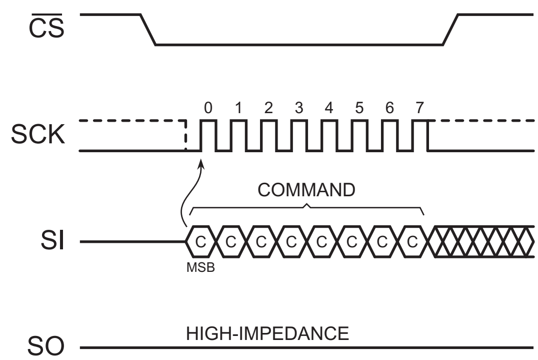



**Figure 3. SPI Transfer — Command Only**

### 4.2.2 Command and Address Only, No Data (1-1-0)

The following diagram shows a transfer with command and address only. In this type of transfer no data is required. An example is the *Block Erase* (20h) command, where the address indicates the location of the block to be erased.

The 1-1-0 transfer type is shown in Figure 4.

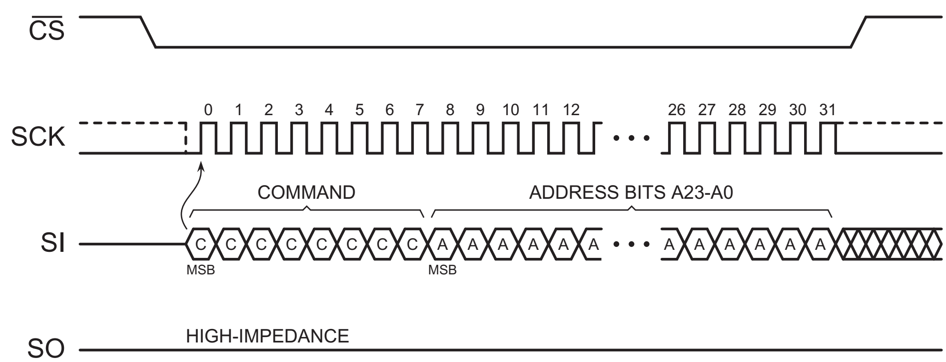



**Figure 4. SPI Transfer — Command and Address Only**

### 4.2.3 Command and Data Only, No Address (1-0-1)

The following diagrams show a transfer with command and data only. In this type of transfer no address is required. An example is the *Status Register Read* (05h/35h/15h) and *Status Register Write* (01h/31h/11h) commands, where the command itself indicates the location of the register. The 1-0-1 transfer type for a read operation is shown in Figure 5.

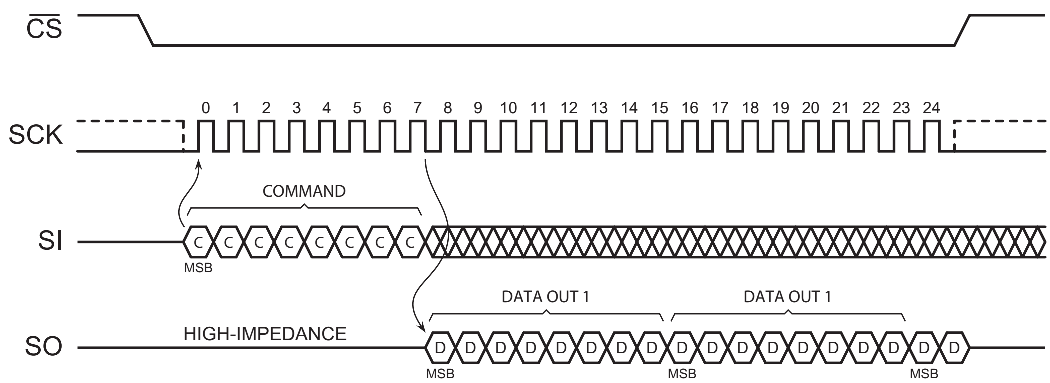



**Figure 5. SPI Transfer — Command and Data Only — Read Operation**

The 1-0-1 transfer type for a write operation is shown in Figure 6.

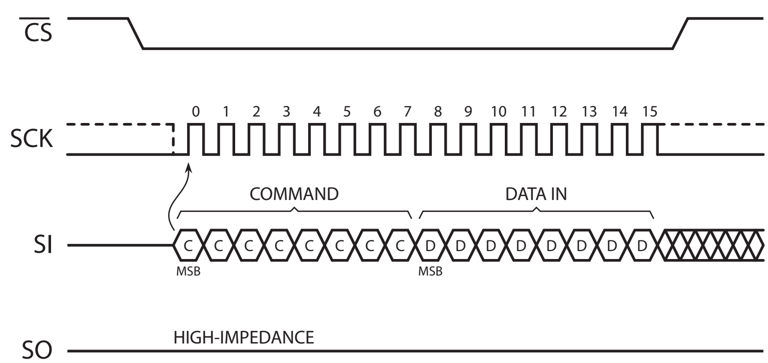



**Figure 6. SPI Transfer — Command and Data Only — Write Operation**

### 4.2.4 Command, Address, and Data (1-1-1)

The following diagrams show a command, address, and data transfer. In this type of transfer the command and address are followed by one or more data types, depending on the command type.

Note that this type of transfer can contain one or more dummy bytes between the end of the address and the beginning of the data output depending on the type of command. See Table 21 for more information.

The 1-1-1 transfer type for a read operation without dummy bytes is shown in Figure 7. An example is the *Read Array* (03h) command.

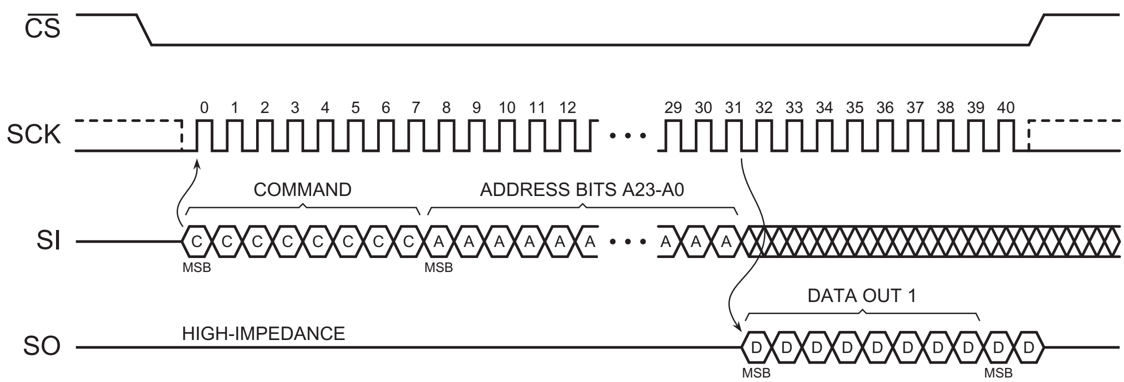



**Figure 7. SPI Transfer — Command, Address, and Data — Read Operation with No Dummy Bytes**

The 1-1-1 transfer type for a read operation with dummy bytes is shown in Figure 8. An example is the *Fast Read Array* (0Bh) command. Note that eight dummy clocks are shown in the figure below for illustration purposes only. More than eight clocks may be required, depending on the type of operation and operating frequency.

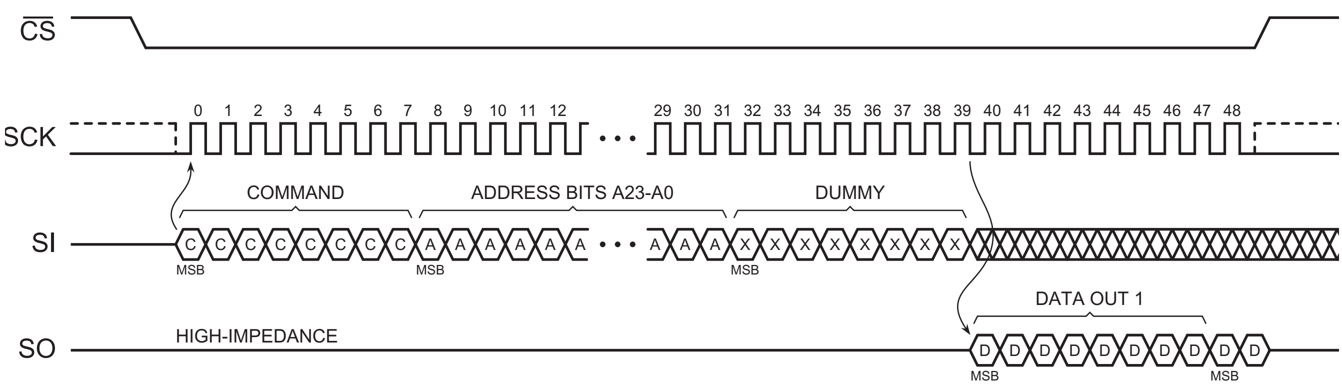



**Figure 8. SPI Transfer — Command, Address, and Data — Read Operation with Dummy Bytes**

The 1-1-1 transfer type for a write operation is shown in Figure 9. An example is the Byte/Page Program (02h) command.

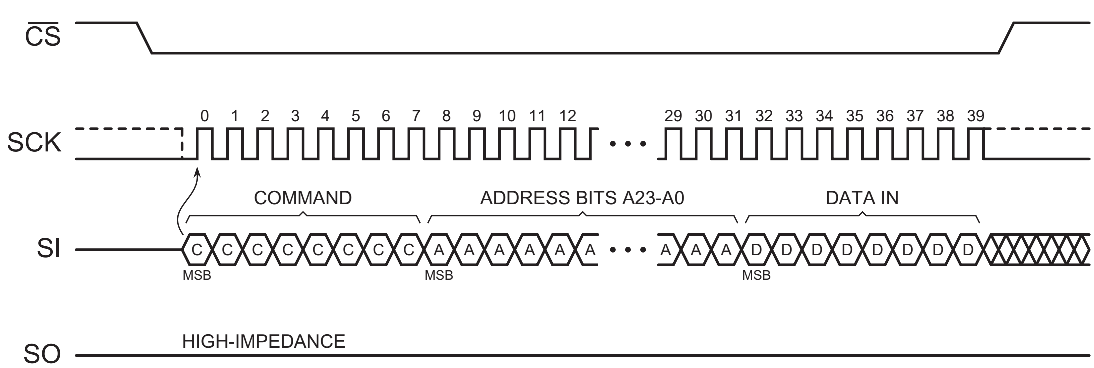



**Figure 9. SPI Transfer — Command, Address, and Data — Write Operation**

## 4.3 Dual Output Operation (1-1-2)

The AT25XE161D supports the Dual Output (1-1-2) transfer type which enhances overall throughput over the standard SPI mode. This mode transfer command and address on the SI pin like in SPI mode, but the data is transferred on to the SI and SO pins. This means that only half the number of clocks are required to transfer the data.

A Dual Output read operation is shown in Figure 10. An example of this operation is a *Dual Output Read* (3Bh). Note that this type of transfer can contain one or more dummy bytes between the end of the address and the beginning of the data output depending on the type of command and the operating frequency. See Table 21 for more information.

A Dual Output 1-1-2 read operation is shown in Figure 10.

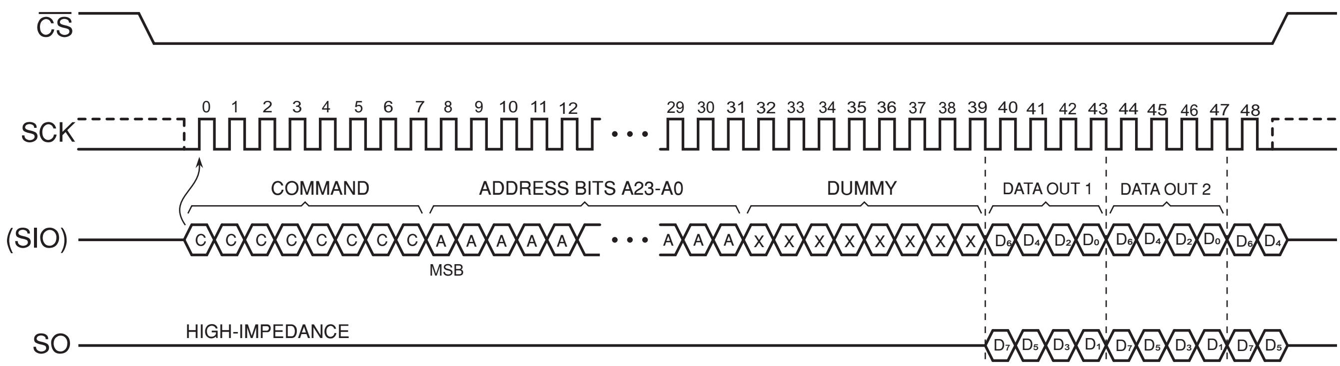



**Figure 10. Dual Output Read — 1-pin Command, 1-Pin Address, and 2-Pin Data**

A Dual Output (1-1-2) write operation is shown in Figure 11. An example of this operation is a *Dual Output Byte/Page Program* (A2h).

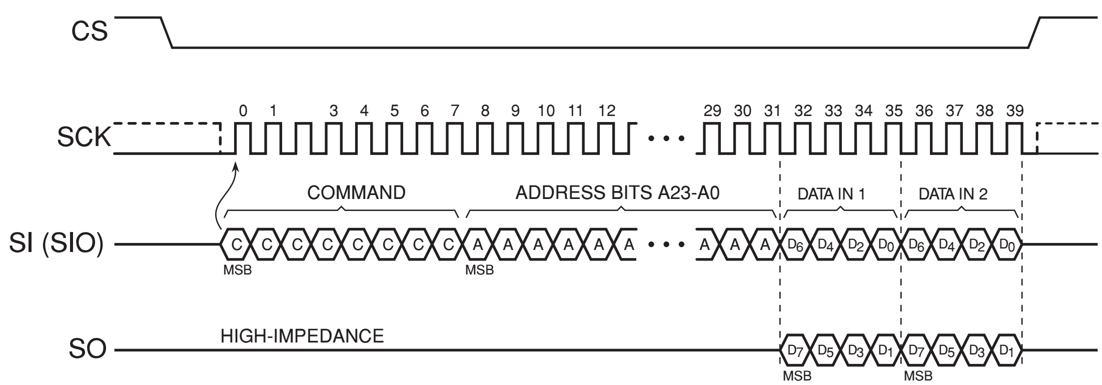



**Figure 11. Dual Output Write — 1-pin Command, 1-Pin Address, and 2-Pin Data**

## 4.4 Quad Output Operation (1-1-4)

The AT25XE161D supports the Quad Output mode which enhances overall throughput over the standard SPI and dual operation modes by increasing the data transfer rate. In this mode, data is transferred on four pins: $IO_0$, $IO_1$, $IO_2$, and $IO_3$. This means that only 1/4th the number of clocks are required to transfer the data relative to standard SPI mode. This is known as a 1-1-4 transfer which is defined as follows: 1-pin command, 1-pin address, and 4-pin data (1-1-4).

In Quad Output mode, the command (C) and address (A) are driven to the memory device on the $IO_0$ pin. During write operations, the $IO_0$, $IO_1$, $IO_2$, and $IO_3$ pins switched to inputs and the data is driven on all four pins, allowing four data bits to be transferred on every clock. During read operations, once the command and address are transferred on the SI ($IO_0$) pin, the $IO_0$, $IO_1$, $IO_2$, and $IO_3$ pins switched to outputs and the data is driven on all four pins, allowing four data bits to be transferred on every clock.

A 1-1-4 Quad Output read operation is shown in Figure 12. An example of this operation is a *Quad Output Read* (6Bh). Note that this type of transfer can contain one or more dummy bytes between the end of the address and the beginning of the data output depending on the type of command and the operating frequency. See Table 21 for more information.

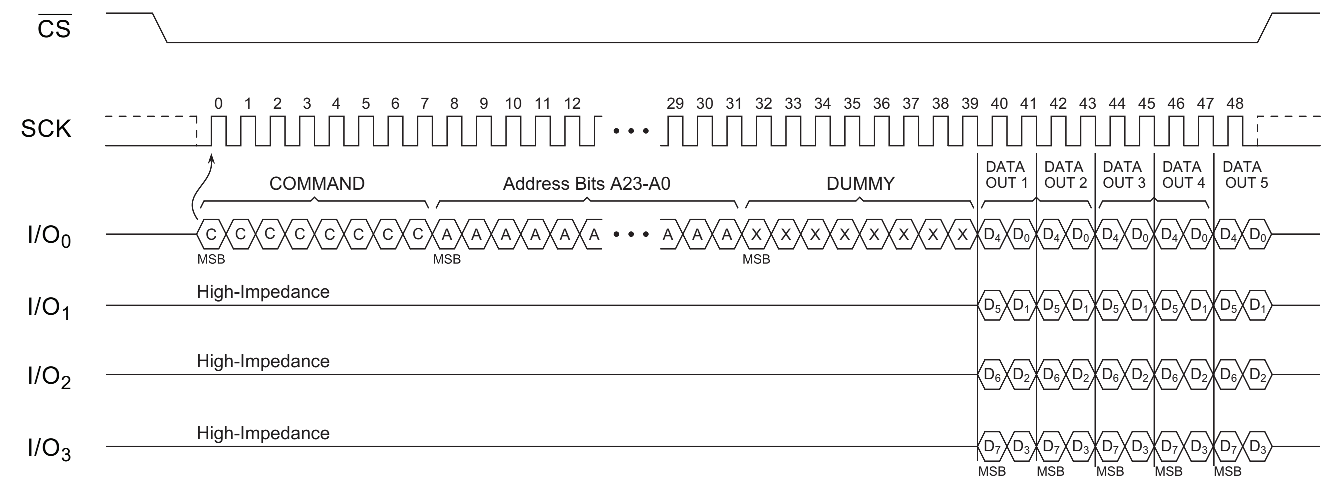



**Figure 12. Quad Output Transfer — 1-Pin Command, 1-Pin Address, and 4-Pin Data — Read Operation**

A 1-1-4 Quad Output write operation is shown in Figure 13. An example of this operation is a *Quad Output Byte/Page Program* (32h).

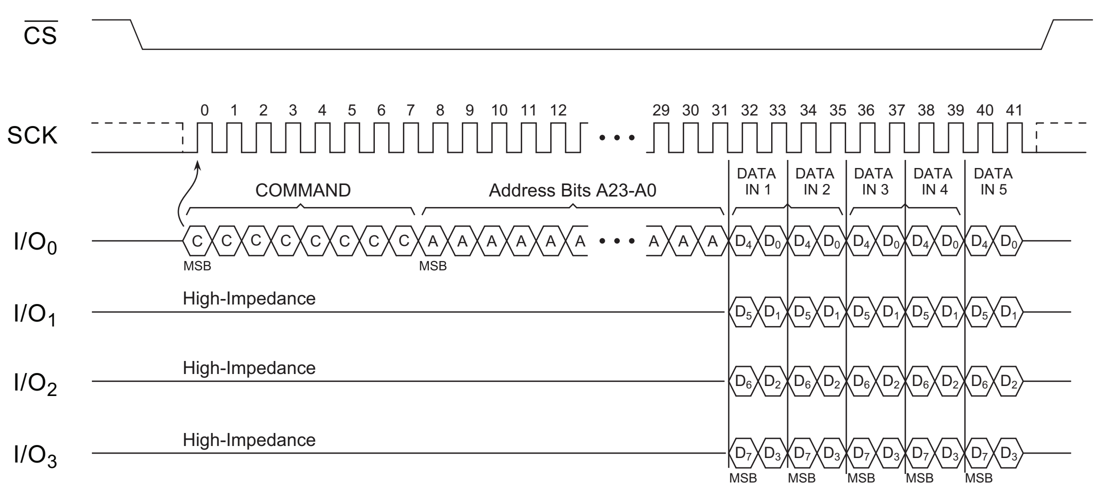



**Figure 13. Quad Output Transfer — 1-Pin Command, 1-Pin Address, and 4-Pin Data — Write Operation**

## 4.5 Quad I/O Operation (1-4-4)

The AT25XE161D supports the Quad I/O mode which enhances overall throughput over the standard SPI and dual operation modes by increasing the data transfer rate. In this mode, address and data are transferred on four pins: $IO_0$, $IO_1$, $IO_2$, and $IO_3$ . This means that only 1/4th the number of clocks are required to transfer the address and data relative to standard SPI mode. This is known as a 1-4-4 transfer which is defined as follows: 1-pin command, 4-pin address, and 4-pin data (1-4-4).

In Quad I/O mode, the command (C) is driven to the memory device on the SI ($IO_0$) pin. During write operations, the $IO_0$, $IO_1$, $IO_2$, and $IO_3$ pins switched to inputs and the address and data are driven on all four pins, allowing four bits to be transferred on every clock. During read operations, once the command is transferred on the SI ($IO_0$) pin, address is transferred on the $IO_0$, $IO_1$, $IO_2$, and $IO_3$ pins, allowing a 24-bit address to be transferred in only six clocks. Once the address transfer is complete, these pins are switched to outputs and the data is driven on all four pins, allowing four data bits to be transferred on every clock. Note that mode bits M[7:0] must not be in high-impedance (tri-state) mode. For optimal performance it is recommended to drive a value of 55h or FFh on these bits when not in XiP mode.

A 1-4-4 Quad I/O read operation is shown in Figure 14. An example of this operation is a *Manufacturer/Device ID Read* (94h). Note that this type of transfer can contain one or more dummy bytes between the end of the address and the beginning of the data output depending on the type of command and the operating frequency. See Table 21 for more information.

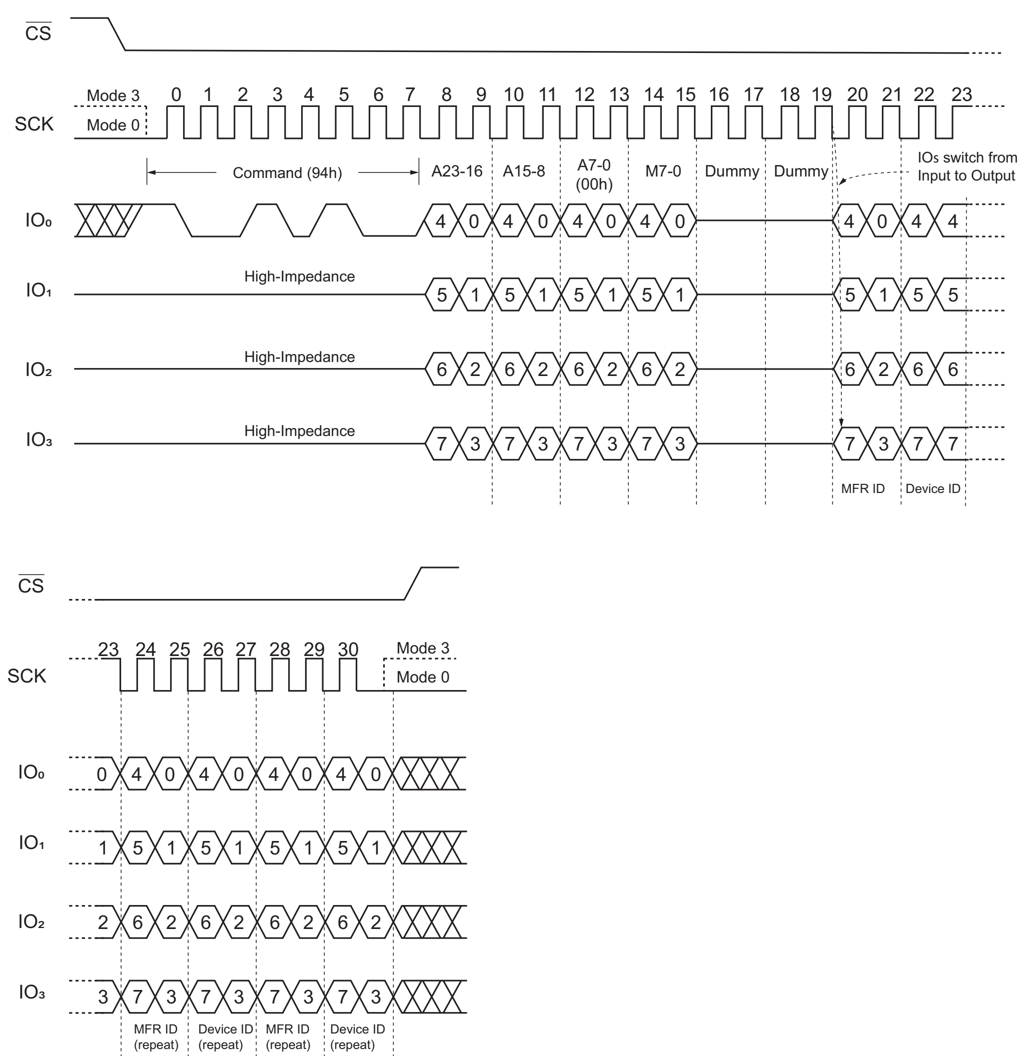



**Figure 14. Quad I/O Transfer — 1-Pin Command, 4-Pin Address, and 4-Pin Data — Read Operation**

## 4.6 XiP Mode Operation

The XiP mode is similar to the Quad I/O mode in that both the address and data are transferred on all four pins: $IO_0$, $IO_1$, $IO_2$, and $IO_3$ . Using all four pins to transfer address and data allows XiP mode to reduce the overall number of clocks required to complete the operation, thereby streamlining code execution.

The XiP mode is enabled by setting bit 2 (QE) of Status Register 1, and bit 3 (XiP) of Status Register 4. The QE bit enables Quad I/O mode, allowing address and data to be transferred on all four pins. Setting the XiP bit enables continuous read mode, allowing subsequent transfers to occur without driving the command each time. Continuous read mode is controlled by mode bits M[5:4] as explained below. Therefore, the initial EBh or E7h command requires the command to be driven, but for subsequent transfers the command is not required.

The XiP mode is only used for the following commands:

- EBh: XiP mode initial read (1-4-4)
- EBh: XiP mode subsequent reads (0-4-4)
- E7h: XiP mode initial read with Doubleword Aligned (DWA) address (1-4-4)
- E7h: XiP mode subsequent reads with DWA address (0-4-4)

As shown above, the only difference between the EBh and E7h commands is that the E7h command is performed only on a double-word-aligned address boundary. The EBh command does not have this restriction. Note that mode bits M[7:0] must not be in high-Z (tri-state) mode. For optimal performance it is recommended to drive a value of 55h or FFh on these bits when not in XiP mode.

The *Set Burst with Wrap* (77h) command does not access the memory directly, but rather is used in conjunction with the EBh and E7h commands to select 8-, 16-, 32-, or 64-Byte sections within a 256-byte page. The user selects the size by programming the wrap bits as part of the 77h command. See the table in Section 6.14, Set Burst Wrap (77h) for more information.

When the EBh/E7h command is driven onto the bus, the associated data immediately following the address contains eight mode bits, known as M[7:0]. Of these bits, M[5:4] are decoded and used by hardware to determine if the device is in XiP continuous read mode. If the value on M[5:4] equals 2'b10, the device is placed into XiP mode to allow for continuous read operations to occur, meaning that for subsequent operations the command field is not required. Each time a subsequent transfer is made, it contains only the address and mode bits.

### 4.6.1 Initial Transfer and XiP Mode Detection (M[5:4])

The initial XiP transfer follows the 1-4-4 format, where the EBh or E7h command is transferred on the SI ($IO_0$) pin, and address and data are transferred on the $IO_0$, $IO_1$, $IO_2$, and $IO_3$ pins. Because all four pins are used, the address requires only six clocks to transfer. The initial 1-4-4 XiP mode transfer is shown in Figure 15.

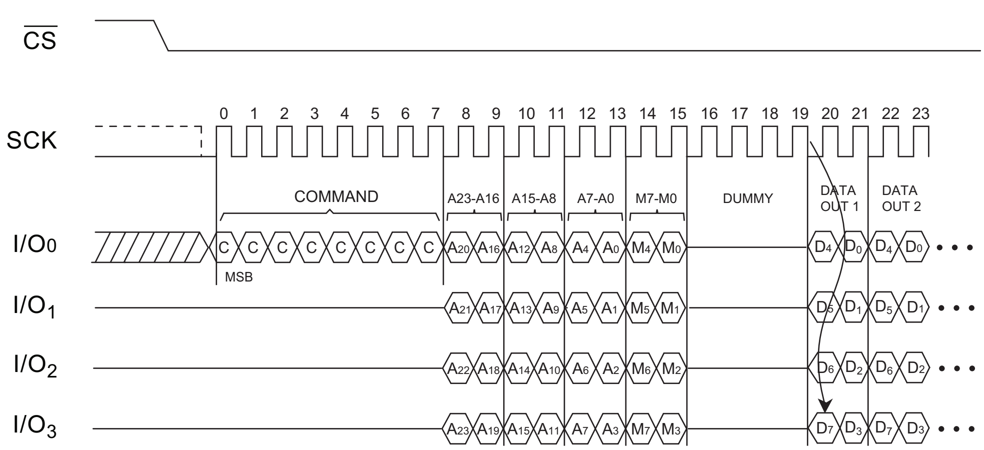



Figure 15. XiP Transfer — 1-Pin Command, 4-Pin Address, and 4-Pin Data — Initial Read (M[5:4] = 2'b10)

### 4.6.2 Subsequent Transfers

Once XiP mode is detected, subsequent operations do not require the command to be transferred. After the data stream starts, the user can deassert the $\overline{CS}$ pin. Once $\overline{CS}$ is deasserted, data output is suspended. Once $\overline{CS}$ is again driven low, only the address and M[7:0] mode data are required because the device is already in XiP mode. Each time a new operation is transferred on the bus, hardware decodes the M[7:0] bits. As long as M[5:4] have a value of 2’b10, the device is in XiP mode and it is not necessary to transfer the command. Once M[5:4] ≠ 2’b10 (any value other than 2’b10), the operation completes and the device exits XiP mode. A subsequent 0-4-4 XiP mode transfer is shown in Figure 16.

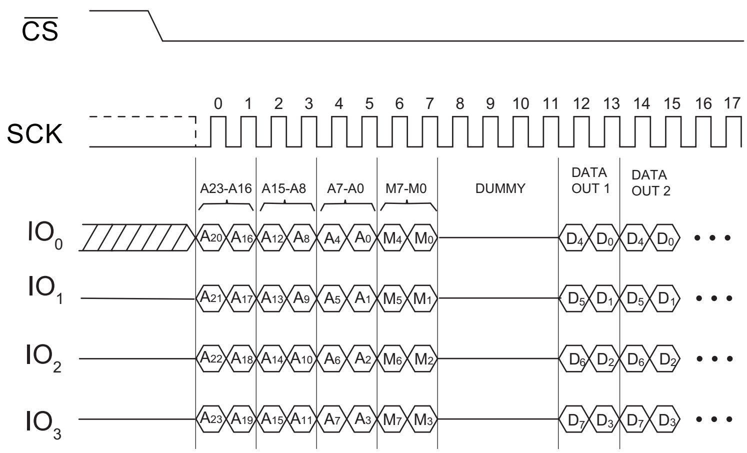



**Figure 16. XiP Transfer — No Command, 4-Pin Address, and 4-Pin Data — Subsequent Reads (M[5:4] = 2’b10)**

### 4.6.3 Set Burst with Wrap

As mentioned above, the *Set Burst with Wrap* (77h) command is used in conjunction with the EBh and E7h commands to select specific sections within a 256-byte page. When this command is transferred, the data field contains 8 wrap bits. Bits 6:4 of this field determine the wrap length, which can be between 8 and 64 bytes. See Section 6.14, Set Burst Wrap (77h) for more information.

Note that when the device receives the EBh/E7h command with M[5:4] = 2’b10, then the device enters XiP mode. While in the XiP (0-4-4) mode (see Subsequent Transfers subsection above) the only command that the device can execute are EBh (or E7h). This is mandatory since the command field for subsequent transfers does not exist.

During normal operation, the user sends the 77h command before entering XiP mode. If the user wants to issue a 77h command after XiP mode has been entered (to change the wrap length/properties), they must exit the XiP mode by sending a 0-4-4 command with M[5:4] ≠ 2’b10). This exits XiP mode. Then the user can issue the 77h command. The 77h command is a 1-4-4 XiP mode transfer as shown in Figure 17.

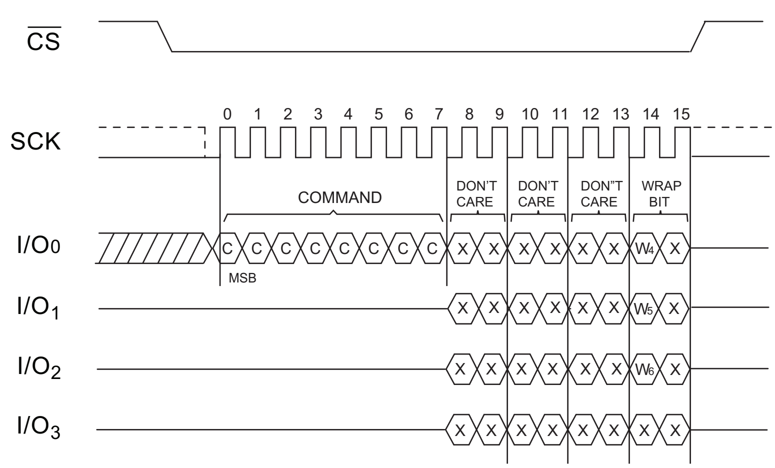



**Figure 17. XiP Transfer — 1-Pin Command, 4-Pin Address, and 4-Pin Data — Write Operation**

See Section 6.14, Set Burst Wrap (77h) for more information.

## 4.7 Memory Architecture

The memory array of the AT25XE161D memory array is divided into three levels of granularity comprising of blocks and pages;

- 64 kB blocks
- 32 kB blocks
- 4 kB blocks
- 256 byte pages

The size of the erase blocks is optimized for both code and data storage applications, allowing both code and data segments to reside in their own erase regions.

Table 3 details each level and the number of pages per block. The program operations to the memory array can be done at the full page level or at the byte level (a variable number of bytes). Erase operations can be performed at the chip, block, or page level.

The physical block size for this device is 2 Mbit.

**Table 3. Device Block Memory Map — Block Erase Address Ranges**

| 64 kB Block Erase (D8h) | 32 kB Block Erase (52h) | 4 kB Block Erase (20h) | Block Address Range |
|---|---|---|---|
| 64 kB (block 31) | 32 kB (block 63) | 4 kB (B511) [1] | 1FF000h - 1FFFFFh |
| | | 4 kB (B510) | 1FE000h - 1FEFFFh |
| | | 4 kB (B509) | 1FD000h - 1FDFFFh |
| | | 4 kB (B508) | 1FC000h - 1FCFFFh |
| | | 4 kB (B507) | 1FB000h - 1FBFFFh |
| | | 4 kB (B506) | 1FA000h - 1FAFFFh |
| | | 4 kB (B505) | 1F9000h - 1F9FFFh |
| | | 4 kB (B504) | 1F8000h - 1F8FFFh |
| | 32 kB (block 62) | 4 kB (B503) | 1F7000h - 1F7FFFh |
| | | 4 kB (B502) | 1F6000h - 1F6FFFh |
| | | 4 kB (B501) | 1F5000h - 1F5FFFh |
| | | 4 kB (B500) | 1F4000h - 1F4FFFh |
| | | 4 kB (B499) | 1F3000h - 1F3FFFh |
| | | 4 kB (B498) | 1F2000h - 1F2FFFh |
| | | 4 kB (B497) | 1F1000h - 1F1FFFh |
| | | 4 kB (B496) | 1F0000h - 1F0FFFh |
| 64 kB (block 30)<br>to<br>64 kB (block 1) | 32 kB (block 61)<br>to<br>32 kB (block 2) | 4 kB (B495)<br>to<br>4 kB (B16) | 1EF000h - 1EFFFFh<br>to<br>010000h - 010FFFh |
| 64 kB (block 0 | 32 kB (block 1) | 4 kB (B15) | 00F000h - 00FFFFh |
| | | 4 kB (B14) | 00E000h - 00EFFFh |
| | | 4 kB (B13) | 00D000h - 00DFFFh |
| | | 4 kB (B12) | 00C000h - 00CFFFh |
| | | 4 kB (B11) | 00B000h - 00BFFFh |
| | | 4 kB (B10) | 00A000h - 00AFFFh |
| | | 4 kB (B9) | 009000h - 009FFFh |
| | | 4 kB (B8) | 008000h - 008FFFh |
| | 32 kB (block 0) | 4 kB (B7) | 007000h - 007FFFh |
| | | 4 kB (B6) | 006000h - 006FFFh |
| | | 4 kB (B5) | 005000h - 005FFFh |
| | | 4 kB (B4) | 004000h - 004FFFh |
| | | 4 kB (B3) | 003000h - 003FFFh |
| | | 4 kB (B2) | 002000h - 002FFFh |
| | | 4 kB (B1) | 001000h - 001FFFh |
| | | 4 kB (B0) | 000000h - 000FFFh |

1. B = block.

Table 4 shows how one 4 kB block maps to sixteen 256 byte pages. The very top and bottom 4 Kbyte blocks of the address space are shown.

**Table 4. AT25XE161D Device Block Memory Map — Page Erase and Page Program**

| 4 kB Blocks | | 256 Byte Page Erase | 1 - 256 Byte Page Program |
|---|---|---|---|
| 4 kB (B511) [1] | | 256 Bytes | 1FFF00h - 1FFFFFh |
| 4 kB (B510) | | 256 Bytes | 1FFE00h - 1FFEFFh |
| 4 kB (B509) | | 256 Bytes | 1FFD00h - 1FFDFFh |
| 4 kB (B508) | | 256 Bytes | 1FFC00h - 1FFCFFh |
| 4 kB (B507) | | 256 Bytes | 1FFB00h - 1FFBFFh |
| 4 kB (B506) | | 256 Bytes | 1FFA00h - 1FFAFFh |
| 4 kB (B505) | | 256 Bytes | 1FF900h - 1FF9FFh |
| 4 kB (B504) | | 256 Bytes | 1FF800h - 1FF8FFh |
| 4 kB (B503) | | 256 Bytes | 1FF700h - 1FF7FFh |
| 4 kB (B502) | | 256 Bytes | 1FF600h - 1FF6FFh |
| 4 kB (B501) | | 256 Bytes | 1FF500h - 1FF5FFh |
| 4 kB (B500) | | 256 Bytes | 1FF400h - 1FF4FFh |
| 4 kB (B499) | | 256 Bytes | 1FF300h - 1FF3FFh |
| 4 kB (B498) | | 256 Bytes | 1FF200h - 1FF2FFh |
| 4 kB (B497) | | 256 Bytes | 1FF100h - 1FF1FFh |
| 4 kB (B496) | | 256 Bytes | 1FF000h - 1FF0FFh |
| 4kB (B494)<br>to<br>4kB (B16) | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| 4 kB (B15) | | 256 Bytes | 000F00h - 000FFFh |
| 4 kB (B14) | | 256 Bytes | 000E00h - 000EFFh |
| 4 kB (B13) | | 256 Bytes | 000D00h - 000DFFh |
| 4 kB (B12) | | 256 Bytes | 000C00h - 000CFFh |
| 4 kB (B11) | | 256 Bytes | 000B00h - 000BFFh |
| 4 kB (B10) | | 256 Bytes | 000A00h - 000AFFh |
| 4 kB (B9) | | 256 Bytes | 000900h - 0009FFh |
| 4 kB (B8) | | 256 Bytes | 000800h - 0008FFh |
| 4 kB (B7) | | 256 Bytes | 000700h - 0007FFh |
| 4 kB (B6) | | 256 Bytes | 000600h - 0006FFh |
| 4 kB (B5) | | 256 Bytes | 000500h - 0005FFh |
| 4 kB (B4) | | 256 Bytes | 000400h - 0004FFh |
| 4 kB (B3) | | 256 Bytes | 000300h - 0003FFh |
| 4 kB (B2) | | 256 Bytes | 000200h - 0002FFh |
| 4 kB (B1) | | 256 Bytes | 000100h - 0001FFh |
| 4 kB (B0) | | 256 Bytes | 000000h - 0000FFh |

1. B = Block

## 4.8 Memory Protection

The AT25XE161D device incorporates a robust memory protection scheme that allows memory locations to be protected down to the 4 kB block level. The device provides the ability to globally lock all blocks in one operation, or to lock individual blocks on a per-block basis.

### 4.8.1 Standard Memory Protection

The standard memory protection scheme is invoked by clearing bit 2 (WPS) of Status Register 3. When this bit is cleared, the AT25XE161D uses a combination of the following register fields to set the memory protection scheme:

- CMPRT: When this bit is set, unprotected areas of memory become protected, and protected areas of memory become unprotected. For example, when CMPRT = 0, a top 64 kB block can be protected while the rest of the array is not; when CMP = 1, the same 64 kB block becomes unprotected while the rest of the array becomes read-only. See bit 6 of Section 5.4, Status Register 2 for more information.
- BPSIZE: This bit works in conjunction with bits 4:2 (BP[2:0]) in Status Register 1 to determine the size of the blocks to be protected. Block sizes of 4 kB and 64 kB can be protected depending on the state of this bit. See bit 6 of Section 5.3, Status Register 1 for more information.
- TB: This bit indicates if the protection is from the bottom up of the top down of the memory. See bit 5 of Section 5.3, Status Register 1 for more information.
- BP[2:0]: This field can be programmed to protect all of the memory array, none of the memory array, or a portion of the memory array. When that portion of the memory if protected, it is protected from the program and erase commands. See bit 4:2 of Section 5.3, Status Register 1 for more information.

The protection scheme differs based on the setting of the CMPRT bit described above. Table 5 shows the relationship between the BPSIZE, BP[2:0], and TB bits when CMPRT = 0. The right-most column shows the protected address range. All addresses not shown are unprotected.

Table 5. AT25XE161D Device Block Protection Map — CMPRT = 0, WPS = 0

| CMPRT | BPSIZE | TB | BP[2:0] | Protected Address Range |
|---|---|---|---|---|
| 0 | 0 | 0 | 3'b000 | NONE |
| 0 | 0 | 0 | 3'b001 | 1F0000 - 1FFFFF |
| 0 | 0 | 0 | 3'b010 | 1E0000 - 1FFFFF |
| 0 | 0 | 0 | 3'b011 | 1C0000 - 1FFFFF |
| 0 | 0 | 0 | 3'b100 | 180000 - 1FFFFF |
| 0 | 0 | 0 | 3'b101 | 100000 - 1FFFFF |
| 0 | 0 | 0 | 3'b110 | 000000 - 1FFFFF |
| 0 | 0 | 0 | 3'b111 | 000000 - 1FFFFF |
| 0 | 0 | 1 | 3'b000 | NONE |
| 0 | 0 | 1 | 3'b001 | 000000 - 00FFFF |
| 0 | 0 | 1 | 3'b010 | 000000 - 01FFFF |
| 0 | 0 | 1 | 3'b011 | 000000 - 03FFFF |
| 0 | 0 | 1 | 3'b100 | 000000 - 07FFFF |
| 0 | 0 | 1 | 3'b101 | 000000 - 0FFFFF |
| 0 | 0 | 1 | 3'b110 | 000000 - 1FFFFF |
| 0 | 0 | 1 | 3'b111 | 000000 - 1FFFFF |
| 0 | 1 | 0 | 3'b000 | NONE |
| 0 | 1 | 0 | 3'b001 | 1FF000 - 1FFFFF |
| 0 | 1 | 0 | 3'b010 | 1FE000 - 1FFFFF |
| 0 | 1 | 0 | 3'b011 | 1FC000 - 1FFFFF |
| 0 | 1 | 0 | 3'b100 | 1F8000 - 1FFFFF |
| 0 | 1 | 0 | 3'b101 | 1F8000 - 1FFFFF |
| 0 | 1 | 0 | 3'b110 | 000000 - 1FFFFF |
| 0 | 1 | 0 | 3'b111 | 000000 - 1FFFFF |
| 0 | 1 | 1 | 3'b000 | NONE |
| 0 | 1 | 1 | 3'b001 | 000000 - 000FFF |
| 0 | 1 | 1 | 3'b010 | 000000 - 001FFF |
| 0 | 1 | 1 | 3'b011 | 000000 - 003FFF |
| 0 | 1 | 1 | 3'b100 | 000000 - 007FFF |
| 0 | 1 | 1 | 3'b101 | 000000 - 007FFF |
| 0 | 1 | 1 | 3'b110 | 000000 - 1FFFFF |
| 0 | 1 | 1 | 3'b111 | 000000 - 1FFFFF |

Table 6 shows the relationship between the BPSIZE, BP[2:0], and TB bits when CMPRT = 1.

**Table 6. AT25XE161D Device Block Protection Map — CMPRT = 1, WPS = 0**

| CMPRT | BPSIZE | TB | BP[2:0] | Protected Address Range |
|---|---|---|---|---|
| 1 | 0 | 0 | 3'b000 | 000000 - 1FFFFF |
| 1 | 0 | 0 | 3'b001 | 000000 - 1EFFFF |
| 1 | 0 | 0 | 3'b010 | 000000 - 1DFFFF |
| 1 | 0 | 0 | 3'b011 | 000000 - 1BFFFF |
| 1 | 0 | 0 | 3'b100 | 000000 - 17FFFF |
| 1 | 0 | 0 | 3'b101 | 000000 - 0FFFFF |
| 1 | 0 | 0 | 3'b110 | NONE |
| 1 | 0 | 0 | 3'b111 | NONE |
| 1 | 0 | 1 | 3'b000 | 000000 - 1FFFFF |
| 1 | 0 | 1 | 3'b001 | 010000 - 1FFFFF |
| 1 | 0 | 1 | 3'b010 | 020000 - 1FFFFF |
| 1 | 0 | 1 | 3'b011 | 040000 - 1FFFFF |
| 1 | 0 | 1 | 3'b100 | 080000 - 1FFFFF |
| 1 | 0 | 1 | 3'b101 | 100000 - 1FFFFF |
| 1 | 0 | 1 | 3'b110 | NONE |
| 1 | 0 | 1 | 3'b111 | NONE |
| 1 | 1 | 0 | 3'b000 | 000000 - 1FFFFF |
| 1 | 1 | 0 | 3'b001 | 000000 - 1FEFFF[1][2] |
| 1 | 1 | 0 | 3'b010 | 000000 - 1FDFFF(1) (2) |
| 1 | 1 | 0 | 3'b011 | 000000 - 1FBFFF(1) (2) |
| 1 | 1 | 0 | 3'b100 | 000000 - 1F7FFF(2) |
| 1 | 1 | 0 | 3'b101 | 000000 - 1F7FFF(2) |
| 1 | 1 | 0 | 3'b110 | NONE |
| 1 | 1 | 0 | 3'b111 | NONE |
| 1 | 1 | 1 | 3'b000 | 000000 - 1FFFFF |
| 1 | 1 | 1 | 3'b001 | 001000 - 1FFFFF[3][4] |
| 1 | 1 | 1 | 3'b010 | 002000 - 1FFFFF(3) (4) |
| 1 | 1 | 1 | 3'b011 | 004000 - 1FFFFF(3) (4) |
| 1 | 1 | 1 | 3'b100 | 008000 - 1FFFFF(4) |
| 1 | 1 | 1 | 3'b101 | 008000 - 1FFFFF(4) |
| 1 | 1 | 1 | 3'b110 | NONE |
| 1 | 1 | 1 | 3'b111 | NONE |

1. When the 32 kB Erase command is used, the protected region is 0x00_0000 - 0x1F_7FFF.
2. When the 64 kB Erase command is used, the protected region is 0x00_0000 - 0x1E_FFFF.
3. When the 32 kB Erase command is used, the protected region is 0x00_8000 - 0x1F_FFFF.
4. When the 64 kB Erase command is used, the protected region is 0x01_0000 - 0x1F_FFFF.

### 4.8.2 Individual Block Lock and Unlock

In addition to the standard protection scheme described in the previous subsection, the AT25XE161D device also provides the ability to lock individual memory locations. Protection of memory locations can be applied individually using the *Individual Block Lock* (36h) command and the *Individual Block Unlock* (39h) command.The user must execute a series of one or more *Individual Block Unlock* (39h) commands to unlock the desired memory locations.

The individual memory protection scheme is invoked by setting bit 2 (WPS) of Status Register 3. When this bit is set, the AT25XE161D uses the *Individual Block Lock* (36h) command that provides the address of the 4 kB or 64 kB block to be locked. See Table 17 for more information on the WPS bit.

Table 3 above shows the address ranges for each 4 kB block corresponding to the top and bottom 64 kB blocks of the memory map. The appropriate 24-bit address is driven on the SI pin after the 36h command. After decoding the command, hardware reads the address and sets the appropriate lock bit for that 4 kB block.

Note that the ability to lock 4 kB blocks only applies to the top and bottom 64 kB blocks of the map. This corresponds to blocks 0 and 31. For 64 kB blocks 1 - 30, the blocks can only be locked on the 64 kB boundary (1 lock bit per 64 kB). This equates to a total of 62 lock bits;

- 16 bits, one per 4 kB sub-block in the top 64 kB block
- 16 bits, one per 4 kB sub-block in the bottom 64 kB block
- 30 bits, one each for 64 kB blocks 1 - 30

See Section 6.21, Individual Block Lock (36h) and Section 6.22, Individual Block Unlock (39h) for more information.

### 4.8.3 Global Block Lock and Unlock

In addition to individual block protection of memory locations as described in the previous subsection, the AT25XE161D also allows for the blocks to be locked and unlocked globally using the *Global Block Lock* (7Eh) and the *Global Block Unlock* (98h) commands. The user must execute a *Global Unlock Block* command to unlock the memory locations.

Note that in the AT25XE161D the factory default setting is the Standard Memory Protection scheme; therefore, all memory locations are unprotected.

See Section 6.24, Global Block Lock (7Eh) and Section 6.25, Global Block Unlock (98h) for more information.

### 4.8.4 Reading the State of the Lock Bits

In addition to globally or individually locking and unlocking selected memory blocks as described above, the AT25XE161D device allows the user to poll any block in memory to determine if it has been locked. This is done by executing either the 3Ch or 3Dh command, along with the 24-bit address. Both of these command perform the exact same operation and can be used interchangeably.

Once this information is decoded, hardware fetches the 8-bit lock field from the requested location and outputs this information onto the SO pin. The most significant bit (MSB) of the value is transferred first, and the least significant bit (LSB) is transferred last. If the LSB is 1, the corresponding block is locked and no erase or program operation can be executed to that block. If the LSB is 0, the block or section is unlocked and program/erase operations are allowed. See Section 6.23, Read Block Lock (3Ch/3Dh) for more information.

## 4.9 Power-Down Considerations

The AT25XE161D device supports the *Deep Power-Down* (B9h) and *Ultra-Deep Power-Down* (79h/B9h) modes. Also, bit 7 (PDM) of Status Register 4 (SR4) can be used to select either of these modes using the B9h command. The 79h command is provided for backward compatibility.

There are three ways to enter power-down mode:

1. Set the PDM bit in SR4 (logic 1) and execute the B9h command to place the device into *Deep Power-Down* (DPD) mode. In this mode it is possible to execute the *Resume from Deep Power-Down* (ABh) command or *Enable Reset* (66h) and *Reset Device* (99h) commands in order to exit DPD mode. The content of SRAM buffer is preserved in this case. The device could also be reset by JEDEC reset, hardware reset, or power-on-reset in order to exit DPD mode. The exit from DPD mode time is defined by the $t_{RDPD}$ timing parameter in Section 7.5. Also see Section 4.9.2, below. Note that in the AT25XE161D device, simply deasserting $\overline{CS}$ as in other devices does not exit DPD mode.
2. Clear the PDM bit in SR4 (logic 0) and execute the B9h command to place the device into *Ultra-Deep Power-Down* (UDPD) mode. To exit this mode, it is necessary to execute the *Resume from Ultra-Deep Power-Down* (ABh) command or a JEDEC reset, hardware reset, or power-on-reset to initiate an internal reset of the device. The contents of the SRAM buffer are NOT preserved. The resume from UDPD mode recovery time is defined

by the $t_{RUDPD}$ timing parameter in Section 7.5. Also see Section 4.9.2, below. Note that in the AT25XE161D device, simply deasserting $\overline{CS}$ as in other devices does NOT exit UDPD mode.

3. Execute the 79h command to place the device into Ultra-Deep Power-Down (UDPD) mode. If this command is used, the state of the PDM bit in SR4 is ignored. This mode allows for software backward compatibility. A device reset is required to exit UPDP mode. In this mode it is necessary to execute the *Resume from Ultra-Deep Power-Down* (ABh) command or a JEDEC reset, hardware reset or power-on-reset to initiate an internal reset of the device. The contents of the SRAM buffer are NOT preserved. The exit from UDPD mode recovery time is defined by the $t_{RUDPD}$ timing parameter in Section 7.5. Also see Section 4.9.2, below for more information on exiting power-down mode.

### 4.9.1 Entering Power-Down Mode

The conditions for entering DPD or UDPD mode are shown in the Table 7. The PDM column indicates the state of the Power-Down Mode bit in Status Register 4.

**Table 7. Entering DPD or UDPD Mode**

| Command | PDM bit | Power-Down Mode | SRAM Contents | Power-Down Exit Recovery Time |
|---|---|---|---|---|
| B9h | 1 | Deep Power-Down | Retained | Short |
| B9h | 0 | Ultra-Deep Power-Down | Lost | Long |
| 79h | x | Ultra-Deep Power-Down | Lost | Long |

### 4.9.2 Exiting Power-Down Mode

The following methods can be used to exit DPD or UDPD mode.

**Table 8. Exiting DPD or UDPD Mode**

| Command | PDM bit | Power-Down Mode | Exit Command | Power On Reset | JEDEC Reset | Hardware Reset Pin[1] | 66h/99h Command | Terminate (F0h) | Status After Exit |
|---|---|---|---|---|---|---|---|---|---|
| B9h | 1 | DPD | ABh[2] | Y | Y | Y | Y | N | Idle |
| B9h | 0 | UDPD | ABh[3] | Y | Y | Y | N | N | Idle |
| 79h | x | UDPD | ABh(3) | Y | Y | Y | N | N | Idle |

1. Hardware reset function must be enabled by software before entering the Power-Down mode. See bit 7 of Status Register 3.
2. Executing the ABh command in DPD mode returns the device to an idle state. The SRAM contents are retained.
3. Executing the ABh command in UDPD mode causes the device to initiate an internal reset sequence. The SRAM contents are undefined.

**ABh Command**

Executing the ABh command while in the Deep Power-Down (DPD) mode causes the device to exit DPD and return back to an idle state. This command does not reset the device and no data is lost. Executing the ABh command while in the Ultra-Deep Power-Down (UDPD) mode causes the device to perform an internal reset. In this case the SRAM contents are undefined.

**Device Resets**

In addition to the ABh command described above, performing a Power On Reset (POR), a JEDEC reset, or asserting the hardware reset pin (RESET) also causes the device to exit DPD or UDPD mode. If the device is reset in any of these three ways, the SRAM contents are undefined.

**Enable Reset (66h) / Reset (99h) Command**

As shown in Table 8, the 66h/99h reset command is accepted in DPD mode. In this case the SRAM contents are retained. The device does not exit UDPD mode, regardless of whether the device entered UDPD mode using the B9h command or the 79h command. To exit UDPD mode, the device must be reset as described in the previous subsections.

**Terminate Command (F0h)**

The *Terminate* (F0) command does not cause exit from DPD or UDPD modes. To exit DPD or UDPD mode still requires either the ABh command, or resetting the device as described in the previous subsections.

### 4.9.3 Reset During Program and Erase Commands

The AT25XE161D device supports the following program and erase operations.

Program operations include:

- Byte/Page Program (02h)
- Sequential Program (AFh/ADh)
- Dual Output Byte/Page Program (A2h)
- Quad Output Byte/Page Program (32h)
- Read-Modify-Write (0Ah)
- Main Memory Page Program Without Erase (88h)
- Program Security Register (9Bh)

Erase operations include:

- Page Erase (81h/DBh)
- 4 kB Block Erase (20h)
- 32 kB Block Erase (52h)
- 64 kB Block Erase (D8h)
- Chip Erase (60h/C7h)

These commands are affected when resetting the device as shown in Table 9. In this table, a Y entry indicates that the device is reset when that type of reset occurs during the corresponding command. A N' entry indicates that the operation is terminated, but the device is not reset. For example, when the F0h command is executed during a program or erase operation, the operation is halted, but the device is not reset.

**Table 9. Resetting the Device During a Program or Erase Operation**

| Command Type | Power-On Reset | JEDEC Reset | Hardware Reset Pin | 66h/99h Command | Terminate (F0h) |
|---|---|---|---|---|---|
| Program | Y | Y | Y | Y | N |
| Erase | Y | Y | Y | Y | N |

## 4.10 Erase/Program Suspend Considerations and Nested Operations

The AT25XE161D device provides three Status Register bits to manage program and erase suspend operations.

- SUSP — Status Register 2, bit 7. The SUSP bit is set by hardware and indicates that the program or erase operation has been suspended.
- ES — Status Register 5, bit 3. The ES bit is set by hardware whenever an erase operation is suspended.
- PS — Status Register 5, bit 2. The PS bit is set by hardware whenever a program operation is suspended.

These three bits work in conjunction to define the state of a suspend operation, as shown in Table 10.

In this table, the SUSP is a logical OR of the ES and PS bits. When either the ES or PS bit is set, the SUSP bit is set. When both of the ES and PS bits are cleared, the SUSP bit is cleared.

These bits relate to the flow diagram in Figure 18 below as follows: when the erase operation is suspended, hardware sets the ES bit. When the program operation is suspended, hardware sets the PS bit. Once the program operation is complete, hardware clears the PS bit. Finally, when the erase operation is completed, hardware clears the ES bit.

**Table 10. Encoding of Erase/Program Suspend Operations**

| SUSP | ES | PS | Status |
|---|---|---|---|
| 0 | 0 | 0 | No suspend operation in progress. |
| 1 | 0 | 1 | Program suspend operation in progress. |
| 1 | 1 | 0 | Erase suspend operation in progress. |
| 1 | 1 | 1 | Nested erase/program suspend operation in progress. |

### 4.10.1 Nested Operations

The AT25XE161D device supports nested erase and program suspend operations. An erase operation can be suspended and a program operation started. This operation can then also be suspended and another operation commenced, such as a read. After the read operation is complete, the program operation can be resumed. Once the program operation is completed, the erase operation can be resumed. Nested operations adhere to the following constraints:

- Suspending an erase operation followed by a program operation is supported
- Suspending a program operation followed by an erase operation is NOT supported

The erase operation must be suspended first, followed by suspension of the program operation.

Figure 18 shows an example of a flow diagram of a nested operation.

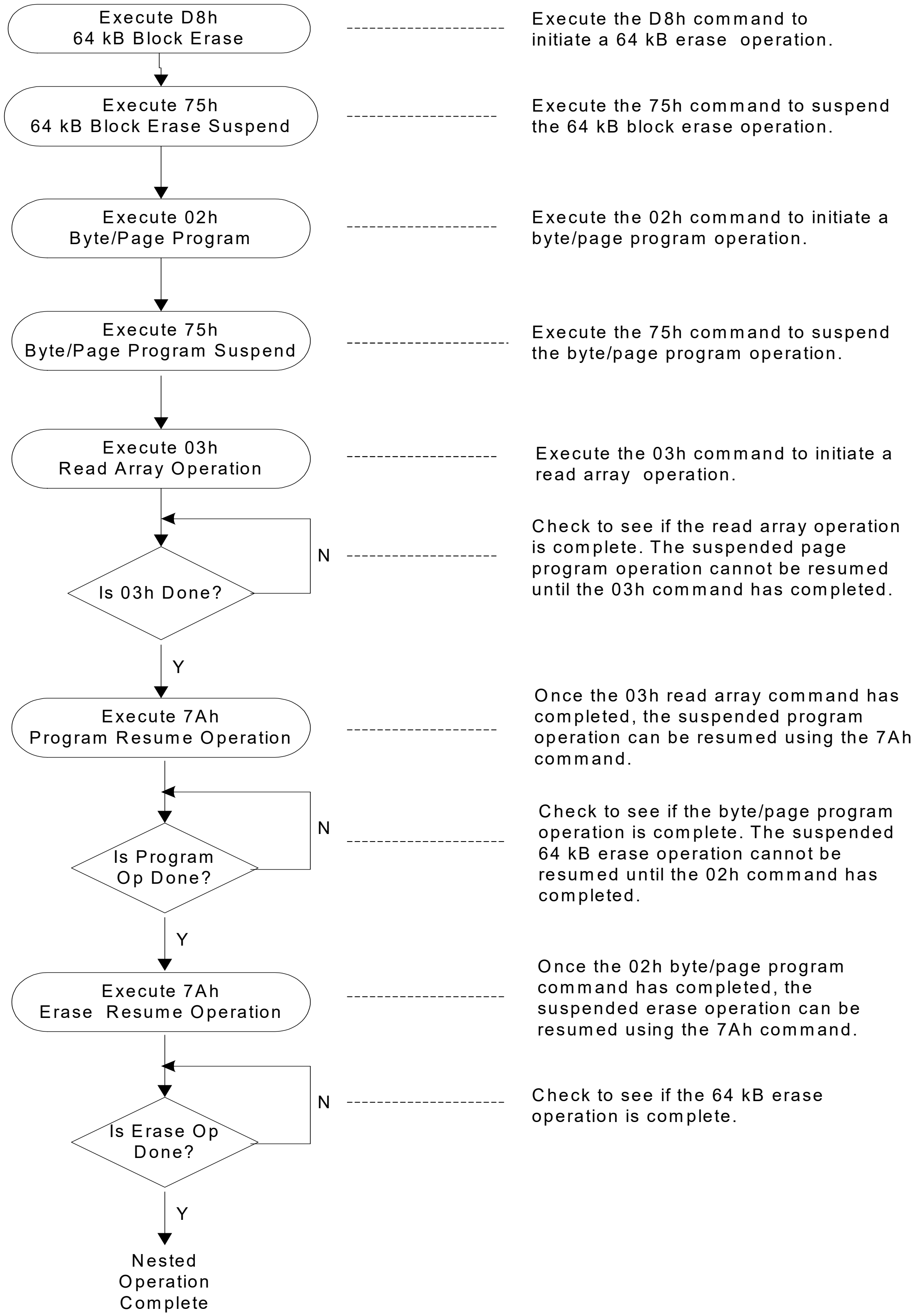



**Figure 18. Flow Diagram of Nested Operations Example**

### 4.10.2 Program and Erase Errors

The AT25XE161D device provides two Status Register bits to indicate program and erase errors.

- EE — Status Register 4, bit 4. The EE bit is set by hardware whenever an error occurs during an erase operation.
- PE — Status Register 4, bit 5. The PE bit is set by hardware whenever an error occurs during a program operation.

When a new program or erase command is accepted by the device, hardware clears the EE or PE bits, depending on the command. The device clears the EE bit when a new *Block Erase* or *Chip Erase* command is accepted. The device clears the PE bit when a new *Byte/Page Program, Buffer to M-M P-P without Erase Sequential Programming, Program OTP Security Register, Write Status Register* or *Status Register Lock* command is accepted. If an error is detected during the command execution, then the EE or PE flag is set. These flags hold their contents even if the program/erase operation is suspended. Note that the Resume command does not change the state of the EE and PE bits. If a *Terminate* command is issued while the device is busy (executing program/erase), then the PE or EE flag is set.

The PE and EE bits are volatile, meaning that any reset operation clears these bits. The ABh command exits Deep Power-Down (DPD) or Ultra-Deep Power-Down (UDPD) mode. If the device is in DPD mode and the user issues the ABh command, then the device exits DPD mode but does not change the state of the PE and EE bits. However, if the ABh command is executed and the device exits from UDPD mode, then the device initiates a reset, which causes both the PE and EE bits to be cleared.

When an error occurs during a given command, the EE and PE bits are set as shown in Table 11. Note that for any of the commands where a Yes appears in the EE or PE columns, when a new command is issued, the EE or PE bit is cleared. Therefore, when the bit is set initially due to an error being detected, it is incumbent on software to detect when these bits are set and take the necessary steps to determine the exact error.

**Table 11. Command Errors and Their Effect on the EE and PE Bits**

| Command Name | Command Code | Sets the EE Bit | Sets the PE Bit |
|---|---|---|---|
| Read Array | 03h, 3Bh, 6Bh, EBh, E7h | No | No |
| Buffer Read | D4h | No | No |
| Block Erase | 81h, DBh, 20h, 52h, D8h | Yes | No |
| Chip Erase | 60h, C7h | Yes | No |
| Buffer Write | 84h | No | No |
| Byte/Page Program | 02h, A2h, 32h | No | Yes |
| Buffer to M-M P-P without Erase | 88h | No | Yes |
| Read-Modify-Write | 0Ah | No | Yes |
| Sequential Programming | ADh, AFh | No | Yes |
| Program/Erase Suspend | B0h, 75h | No | No |
| Program/Erase Resume | D0h, 7Ah | No | No |
| Write Enable | 06h | No | No |
| Write Disable | 04h | No | No |
| Volatile Write Enable | 50h | No | No |
| Individual Block Lock | 36h | No | No |
| Individual Block Unlock | 39h | No | No |
| Global Block Lock | 7Eh | No | No |
| Global Block Unlock | 98h | No | No |
| Read Block Lock Status | 3Ch, 3Dh | No | No |
| Program OTP Security Register | 9Bh | No | Yes |
| Read OTP Security Register | 4Bh | No | No |

**Table 11. Command Errors and Their Effect on the EE and PE Bits** (Continued)

| Command Name | Command Code | Sets the EE Bit | Sets the PE Bit |
|---|---|---|---|
| Read Status Registers (direct) | 05h, 35h, 15h | No | No |
| Read Status Registers (indirect) | 65h | No | No |
| Write Status Registers (direct) | 01h, 31h, 11h | No | Yes |
| Write Status Registers (indirect) | 71h | No | Yes |
| Status Register Lock | 6Fh | No | Yes |
| Active Interrupt | 25h | No | No |
| Terminate | F0h | Yes<br>(if Erase command is terminated) | Yes<br>(if Program command is terminated) |
| Enable Reset | 66h | No | No |
| Reset Device | 99h | Cleared on POR | Cleared on POR |
| Read Mfg ID and Device ID | 9Fh, 90h, 94h | No | No |
| Deep Power-Down | B9h | No | No |
| Resume from Deep Power-Down | ABh | No | No |
| Resume from Ultra-Deep Power-Down | ABh | Cleared on POR | Cleared on POR |
| Ultra-Deep Power-Down | 79h | No | No |
| Read SFDP | 5Ah | No | No |
| Set Burst with Wrap | 77h | No | No |
| Low Battery Detect | EFh | No | No |

### 4.10.3 Suspending and Terminating a Program or Erase Operation

**Program Operation**

A self-timed program operation can be suspended using the *Suspend* (75h) command and terminated using the *Terminate* (F0h) command. The device responds to either of these commands by initiating a sequence which completes some internal sub-operations, brings the internal voltage supplies to their quiescent state, saves the intermediate address counters and states, and sets either the PS (Program Suspend) bit in Status Register 4 in the case of a *Suspend* command, or the PE (Program Error) bit in Status Register 5 in case of the *Terminate* command.

When the user issues the *Suspend* command the device requires a period of time equal to $t_{SUS}$ before the BUSY flag goes low and the PS flag goes high. Similarly, when the user issues the *Terminate* command the device requires a period of time equal to $t_{SWTERM}$ before the BUSY flag goes low and the PE flag goes high.

If the user issues the *Suspend* command, and then issues the *Terminate* command before the device has completed the internal suspend operation, the device honors the Terminate command and sets the PE flag. The PS flags is not set, and the BUSY flag is cleared. At this point the self-timed command that was terminated cannot be resumed, and the region of the memory that was being programmed/erased is left in an intermediate state.

Erase Operation

A self-timed erase operation can be suspended using the *Suspend* (75h) command and terminated using the *Terminate* (F0h) command. The device responds to either of these commands by initiating a sequence which completes some internal sub-operations, brings the internal voltage supplies to their quiescent state, saves the intermediate address counters and states, and sets either the ES (Erase Suspend) bit in Status Register 4 in the case of a Suspend command, or the EE (Erase Error) bit in Status Register 5 in case of the *Terminate* command.

When the user issues the *Suspend* command the device requires a period of time equal to $t_{SUS}$ before the BUSY flag goes low and the PE flag goes high. Similarly, when the user issues the *Terminate* command the device requires a period of time equal to $t_{SWTERM}$ before the BUSY flag goes low and the EE flag goes high.

If the user issues the *Suspend* command, and then issues the *Terminate* command before the device has completed the internal suspend operation, the device honors the *Terminate* command and sets the EE flag. The ES flags is not set, and the BUSY flag is cleared. At this point the self-timed command that was terminated cannot be resumed, and the region of the memory that was being programmed/erased is left in an intermediate state.

This concept is shown in Figure 19.

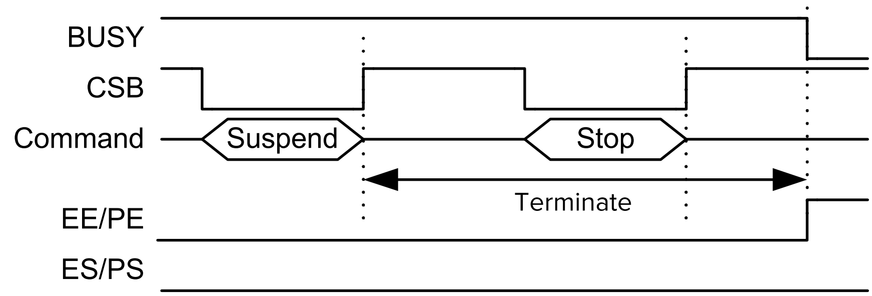



**Figure 19. Issuing a Terminate Command Before the Suspend Command has Completed**

**Terminate After Suspend**

If the time between the *Suspend* command and the *Terminate* command is large enough, then the device can finish the internal suspend operation before the *Terminate* command is received. In this case, the device clears the BUSY bit in Status Register 1, and the ES/PS flags in Status Register 5 are set. Now when the device receives the *Terminate* command, it is ignored by the hardware. This concept is shown in Figure 20.

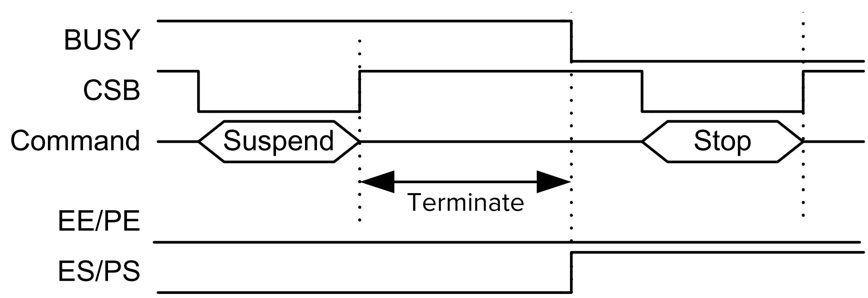



**Figure 20. Allowing Enough Time for the Suspend Operation to Complete**

If the user wishes to perform another program operation, they must check the status of both the PS and PE flags in Status Registers 4 and 5, respectively, to determine the next action.

If the PS bit is cleared and the PE bit is set, the user can issue a new *Program* command. Note that previous *Program* command that was terminated leaves that region of memory in an intermediate state, and it is recommended to erase this region of memory before attempting to program it.

If the ES/PS bit is set, then the user cannot issue another *Program*/*Erase* command. The user must first issue a *Resume* command to re-start the suspended program/erase operation, then either allow it to run to completion or terminate it using the *Terminate* command. The user can now issue a new *Program*/*Erase* command.

### 4.10.4 Terminating a Non-Volatile Register Operation in Progress

The *Terminate* (F0h) command, the *Software Reset* command (66h + 99h), the hardware reset (RESET on $IO_3$) pin, and the JEDEC Reset are ways to terminate any on-going internal self-timed program/erase operation (and in some cases reset the device). However, abruptly terminating the *Status Register Write, Status Register Lock*, or *OTP Security Register Program* command is not desirable because this can leave these non-volatile registers in an indeterminate state.

Therefore when the device is busy executing the *Status Register Write*, the *Status Register Lock* or the *OTP Security Register Program* command, then the *Terminate* command is ignored. The software reset, hardware reset, and JEDEC reset actions are delayed until after the internal self-timed operations are completed; once the internal operation is completed, the device is reset. See Section 4.12 for more information on how to perform a JEDEC hardware reset.

## 4.11 OTP Security Register Lock

The AT25XE161D device supports four 128-byte OTP security registers. One register is programmed by Renesas Electronics at the factory and is always locked. The other three OTP registers can be programmed by the user and become locked whenever any bit in the most significant byte of that register is written. In response to this write operation, hardware writes bits 5:3 (SL3:SL1) of Status Register 2 to indicate that the corresponding register has been locked. Software can read this field to determine which registers have been locked.

Each byte of each register can be written using address bits 8:0 that are driven with the Program OTP Security register command (9Bh). See Table 27 in Section 6.26, Program Security Register (9Bh) for an exact encoding of these address bits.

This concept is shown in Figure 21. In this figure, A[8:7] are used to select between the four OTP Security registers, and A[6:0] are used to select one of the 128 bytes within that register. A value of 7’b1111111 on A[6:0] indicates an access to the MSB of that register as shown below.

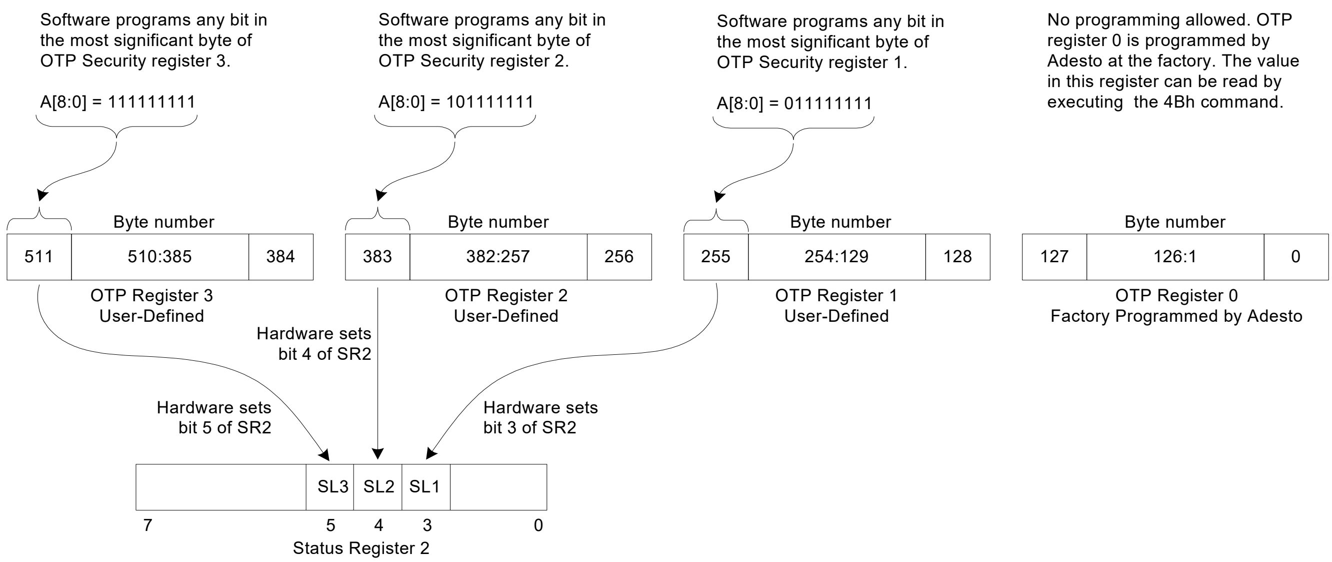



**Figure 21. OTP Security Register Program and Lock**

## 4.12 Standard JEDEC Hardware Reset

The JEDEC hardware reset sequence does not use the SCK pin. The SCK pin must be held low (mode 0) or high (mode 3) through the entire reset sequence. This prevents any confusion with a command, as no command bits are transferred.

A reset is commanded when the data on the SI pin is 0101 on four consecutive positive edges of the $\overline{CS}$ pin with no edge on the SCK pin throughout. This is a sequence where

1. $\overline{CS}$ is driven active low to select the device.
2. Clock (SCK) remains stable in either a high or low state.
3. SI is driven low by the bus master, simultaneously with $\overline{CS}$ going active low. No SPI bus slave drives SI during $\overline{CS}$ low before a transition of SCK.
4. $\overline{CS}$ is driven inactive. The slave captures the state of SI on the rising edge of $\overline{CS}$.

The above steps are repeated 4 times, each time alternating the state of SI.

After the fourth $\overline{CS}$ pulse, the slave triggers its internal reset. SI is low on the first $\overline{CS}$, high on the second, low on the third, high on the fourth. This provides a value of 5h, unlike random noise. Any activity on SCK during this time halts the sequence, and a Reset is not generated.

After a JEDEC hardware reset while the device is in Ultra-Deep Power-Down (UDPD) mode, the SRAM buffer resets to an undefined value. Hardware resets all volatile Status Registers, including the block protection bits, to their default values.

After a JEDEC hardware reset while the device is in any other mode than UDPD mode, the SRAM buffer keeps the values it had prior to Reset, with the following exception: If the reset sequence is initiated during an update of the SRAM buffer, the contents of the SRAM buffer can be corrupted. Hardware resets all volatile Status Registers, including the block protection bits, reset to their default values.

All non-volatile Status Registers keep the value they had prior to reset, with the following exception: If the reset sequence is initiated during a write to a non-volatile Status Register, the value of that register can be corrupted.

The device reverts to standard SPI mode after JEDEC hardware reset. Figure 22 shows the timing for the JEDEC hardware reset operation.

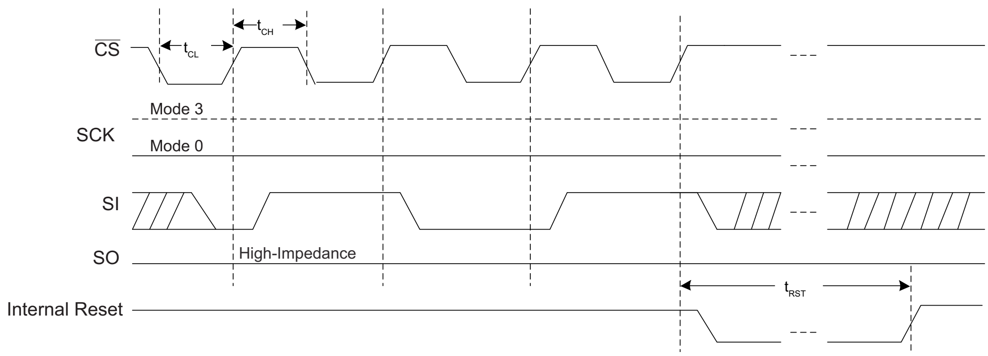



**Figure 22. JEDEC Standard Hardware Reset**

## 4.13 Chip Select Restrictions

The $\overline{CS}$ pin starts and ends operations in the device. Once the $\overline{CS}$ pin is asserted and the operation begins, it can only be deasserted on a byte boundary. If the $\overline{CS}$ pin is deasserted on a non-byte boundary, the operation is ignored.

For example, when executing the ABh command to exit power-down mode, only the command is required. No address and data are required to perform this operation. Therefore, only eight clocks are required to transfer the command. If the $\overline{CS}$ pin is raised after the 8th clock (most significant bit of the command is transferred), hardware performs the operation. If the $\overline{CS}$ pin is raised after the 9th or 10th clock (not a byte boundary), the operation is ignored and no ABh command is executed.

Similarly, if the ABh command fetches the device ID. Once the $\overline{CS}$ pin is asserted, 40 clock cycles are required: 8 clocks for the command, 24 dummy clocks, and 8 clocks to shift out the device ID. If the $\overline{CS}$ pin is raised after the 40th clock, hardware transfers the ID information, exits power-down mode, and completes the operation. However, if the $\overline{CS}$ pin is raised on the 41st of 42nd clock (not a byte boundary), the ID information is still returned as that occurred in clocks 32 - 40, but the device does not exit power-down mode.

## 4.14 Active Status Interrupt

When a program or erase operation is in progress, there are two ways to determine when the operation has finished.

- Repeatedly poll bit 0 (RDY/BSY) of Status Register 1 looking for a high to low transition.
- Once the program or erase operation begins, execute the *Active Status Interrupt* (25h) command, and watch for a high to low transition on the SO pin.

For example, assume a *Byte/Page Program* (02h) operation is initiated and 256 bytes are transferred into the buffer. Once the $\overline{CS}$ pin is deasserted, indicating the completion of the 02h command, hardware sets the RDY/BSY bit and starts moving the data to memory. Executing the 25h command causes hardware to drive the state of the RDY/BSY bit onto the SO pin. This allows external logic to monitor the state of the SO pin for a high to low transition.

When the *Active Status Interrupt* (25h) command is used, the SO pin can be connected to an external interrupt controller. The controller recognizes the high to low transition and takes the appropriate action. This command is an alternative to the polling of the RDY/BSY bit in Status Register 1 and uses less overhead than continually executing the 05h command to read the contents of Status Register 1.

## 4.15 Low Battery Detect

The AT25XE161D device includes a *Low Battery Detect* (EFh) command that can be used to probe the battery and determine its status. The low battery detect function is programmable. Software uses Status Register (SR6) to program parameters such as what voltage level constitutes a low battery, and how much load is applied to the battery to determine the amount of charge left. Another field sets the amount of time the desired load is applied to the battery.

Status Register 6 is used in the following manner to determine the low battery status. For the exact programmable values for these registers fields, see Section 5.4, Status Register 2.

1. Program the LBVL field (bits 5:3) of SR6 to set the low battery detect level. When a test is performed, the device does not indicate a low voltage level until that level falls below the value programmed into this field. Values range from 1.8V to 3.2V in 200 mV increments.
2. Program the LBLD field (bits 2:1) of SR6 to indicate the amount of load applied to the battery for testing purposes. The amount of load applied ranges from 0 µA to 10 mA.
3. Program the LBD field (bit 0) to indicate the amount of time the load selected by the LBLD field is applied to the battery. Values of 100 µs and 1 ms are supported.
4. Execute the *Low Battery Detect* (EFh) command. This operation sets bits 7:6 (LBST) of SR6 to a value of 2’b01, indicating that a low battery detect operation is in progress.
5. Software can poll bits 7:6 of SR6 to determine whether the test has completed. As long as the value of these bits is 2’b01 the battery test is in progress. Alternatively, the active status interrupt (25h) command can be used to determine when the test has completed. See the previous subsection for more information on the active status interrupt command.
6. Once the operation is complete, bits 7:6 of SR6 indicate the result of the test. If the field contains a value of 2’b10, the voltage of the battery is greater than the threshold set by the LBVL field. A value of 2’b11 indicates the voltage of the battery is less than the threshold set by the LBVL field.

Bits 7:6 of SR6 can be reset by hardware when certain internal conditions are met. The user can reset this field by executing a Terminate command (F0h). See Section 6.43, Low Battery Detect (EFh) for more information.

## 4.16 Read-Modify-Write

The AT25XE161D device incorporates a completely self-contained Read-Modify-Write (R-M-W) operation (command 0Ah) that can be used to reprogram any number of sequential bytes in a page in the main memory array without affecting the rest of the bytes in the same page. This command allows the device to easily emulate an EEPROM by providing a method to modify a single byte or more in the main memory in a single operation, without pre-erasing the memory or any external RAM buffers. The main advantage of this command is that it allows a memory location to be erased and reprogrammed in one operation.

The device also incorporates an intelligent erase and programming algorithm that can detect when a byte location fails to erase or program properly. If an erase or program error arises, hardware indicates this condition by setting bit 5:4 (EE and PE) in Status Register 4. See Section 5.6, Status Register 4 for more information on these bits. See Section 6.42, Single Command Read-Modify-Write — EEPROM Emulation (0Ah) for more information on the 0Ah command.

## 4.17 HOLD / RESET Function

The AT25XE161D device provides a configurable HOLD/RESET pin. This pin can be configured to operate as either a HOLD pin, or as a device RESET pin, by programming bit 7 (HOLD/RESET) of Status Register 3. When this bit is cleared, the HOLD/RESET pin functions as an active low HOLD pin. When this bit is set, the HOLD/RESET pin functions as a active low device RESET pin.

The HOLD/RESET function is only valid in the SPI and dual modes of operation. When bit 1 (QE) of Status Register 2 is set, indicating the device is in one of the quad modes (quad output or I/O) the HOLD/RESET functions are not available and the pin functions as the $I/O_3$ data pin.

When configured as a RESET pin, no commands are accepted while the pin is low and the device is in reset.

Configuring the pin for the HOLD function allows an operation to be paused, then resumed when the HOLD pin is deasserted. Once the HOLD pin is asserted (with CS low), the HOLD function takes effect on the next falling edge of SCK. Conversely, once the HOLD pin is deasserted, the HOLD function is removed on the next falling edge of SCK. Note that the CS pin must be low for the entire time in which the HOLD operation is in progress.

The HOLD function can be used in situations where the clock and data signals of the AT25XE161D device are shared with other external agents, allowing the device to share the bus with other high priority events such as interrupts or other events that need immediate attention. Once the condition is resolved, the HOLD pin can be deasserted, allowing the operation to resume.

During the time the HOLD pin is asserted, the Serial Output (SO) pin is forced to the high impedance state. The state of the Serial Input (SI) and Serial Clock (SCK) pins are ignored. Note that the CS pin must be low for the entire time in which the HOLD operation is in progress.

# 5. Status Registers

The device contains six Status Registers (SR) described in the following subsections. The Status Registers can be read to determine the device's ready/busy status, as well as the status of many other functions such as hardware locking and software protection.

## 5.1 Register Structure and Updates

In the AT25XE161D device there are two sets of Status Registers. One set is implemented in hardware and is accessed as Status Registers 1 - 6. These are known as volatile registers as their data is lost during a device reset. Also, the device keeps a copy of the Status Registers in memory. These are known as non-volatile registers because they are stored in the Flash and are not affected by a device reset. On power up, the non-volatile contents of the memory are copied to the Status Registers. See Section 5.2.2 for more information on volatile and non-volatile register types.

The AT25XE161D provides two methods for updating the Status Registers. When the *Write to Status Register* command is preceded by the *Volatile Status Register Write Enable* (50h) command, only the flip-flops holding the Status Register bits are updated; the non-volatile memory dedicated to the Status Registers is not updated. When the *Write to Status Register* command is preceded with a *Write Enable* (06h) command, then both the flip-flops holding the Status Register bits and the non-volatile memory dedicated to the Status Registers are updated. This concept is shown in Figure 23.

The main difference between these two commands is the time required to execute them. Accesses to memory are much slower by default. So for the 50h command, the write is to the volatile Status Registers only which is very fast. For the 06h command, the write is to non-volatile memory which is much slower.

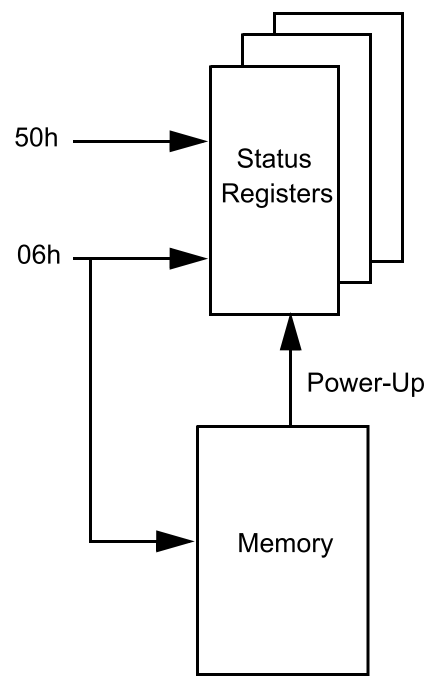



**Figure 23. AT25XE161D Register Structure**

## 5.2 Register Accesses

This section describes the various ways to access the AT25XE161D Status Registers. The AT25XE161D incorporates six Status Registers. Each of these registers can be written or read using both the direct and indirect addressing methods as described in Section 5.2.1. Also, the device supports both volatile and non-volatile registers as described in Section 5.2.2.

### 5.2.1 Direct and Indirect Addressing

The AT25XE161D device supports both direct and indirect addressing for accessing the Status Registers. The indirect addressing method can be used for all six Status Registers, and the direct addressing for the first three Status Registers. This means that Status Registers 4 - 6 can only be accessed using the indirect addressing method.

These registers have been accessed using the direct addressing method, where every register has a specific command used to access it and perform read or write operations. An example of these commands is as follows:

- 05h: Read Status Register 1
- 35h: Read Status Register 2
- 15h: Read Status Register 3
- 01h: Write Status Register 1
- 31h: Write Status Register 2
- 11h: Write Status Register 3

The above commands have been retained to provide backward compatibility with existing software. Also to the direct method, the AT25XE161D device also provides an indirect method. In the indirect method, a single command (65h) executes a read operation on all of the Status Registers. Another command (71h) executes a write operation on all of the Status Registers. Once the command is provided, an 8-bit address field determines which register to access. The encoding of the address field is shown in Table 12.

**Table 12. Indirect Addressing of Status Registers**

| Command | Address | Action |
|---|---|---|
| 65h | 01h | Read Status Register 1 |
| | 02h | Read Status Register 2 |
| | 03h | Read Status Register 3 |
| | 04h | Read Status Register 4 |
| | 05h | Read Status Register 5 |
| | 06h | Read Status Register 6 |
| 71h | 01h | Write Status Register 1 |
| | 02h | Write Status Register 2 |
| | 03h | Write Status Register 3 |
| | 04h | Write Status Register 4 |
| | 05h | Write Status Register 5 |
| | 06h | Write Status Register 6 |

Note that by using the indirect address method, an address width of 8 bits indicates that up to 256 registers can be accessed, allowing for future expansion without requiring additional commands or associated hardware complexity.

### 5.2.2 Volatile and Non-Volatile Register Accesses

As shown in Figure 23, the AT25XE161D device supports both volatile and non-volatile register accesses. Volatile register accesses are to the Status Registers implemented in hardware. Non-volatile register accesses are to the Status Registers stored in memory. During power-up, the contents of the registers in memory are copied to the hardware Status Registers.

As shown in Figure 23, the 50h command writes only to the volatile register set, whereas the 06h command writes to both the volatile and non-volatile register sets.

## 5.3 Status Register 1

Table 13 shows the bit assignments for Status Register 1.

**Table 13. Status Register 1 Format**

| Bit # | Acronym | Name | Type | Default | Description |
|---|---|---|---|---|---|
| 7 | SRP0 | Status Register Protect 0 | R/W | 0 | SRP0 works with the SRP1 bit in Status Register 1 and the $\overline{\text{WP}}$ pin to control write protection. Types of protection include software protection, hardware protection, one-time programmable (OTP) protection, and power supply lock-down protection. See Table 15 for an encoding of these bits. |
| 6 | BPSIZE | Block Protect Size | R/W | 0 | BPSIZE controls the size of the blocks protected by the Block Protect Bits (BP2, BP1, BP0 in bits 4:2 of this register). Its encoding is:<br>0: 64 kB block size<br>1: 4 kB block size<br>The blocks can be protected from the bottom up, or from the top down, as described in the TB bit of this register. |
| 5 | TB | Top/Bottom | R/W | 0 | TB controls the direction of the blocks to be protected by the Block Protect Bits (BP2, BP1, BP0 in bits 4:2 of this register). Its encoding is:<br>0: Protect from bottom up<br>1: Protect from top down<br>The size of the protected blocks can also be selected, as described in the BPSIZE bit of this register. |
| 4:2 | BP2:0 | Block Protect | R/W | 000 | The Block Protect field provides write protection control and status. These bits are set using the Write Status Register 1 (01h) command. This field can be programmed to protect all, none, or a portion of the memory array. When that portion of the memory is selected, it is protected from the Program and Erase commands as described in the Memory Protection table.<br>The default is 3'b000 for this field, indicating that none of the memory array is protected. |
| 1 | WEL | Write Enable Latch Status | R | 0 | WEL gives the current status of the internal Write Enable Latch. When WEL is logic 0, the device does not accept any program, erase, memory protection, or Write Status Register commands. WEL defaults to logic 0 after a device power-up or reset.<br>Its encoding is:<br>0: Device is not write enabled (default).<br>1: Device is write enabled.<br>If WEL is 1, it is not reset to a logic 0 if an operation aborts due to an incomplete or unrecognized command being clocked into the device before the $\overline{\text{CS}}$ pin is deasserted.<br>To reset the WEL bit when an operation aborts prematurely, the entire command for a program, erase, memory protection, or Write Status Register command must have been clocked into the device.<br>When the Write Enable (06h) command is executed, the WEL bit is set. Conversely, when the Volatile Status Register Write Enable (50h) command is executed, the WEL bit is not set. |
| 0 | $\overline{\text{RDY}}$/BSY | Ready/Busy Status | R | 0 | $\overline{\text{RDY}}$/BSY determines if an internal operation, such as a program or erase, is in progress. To poll the $\overline{\text{RDY}}$/BSY bit to detect the completion of a program or erase cycle, new Status Register data must be continually clocked out of the device until the state of the $\overline{\text{RDY}}$/BSY bit changes from a logic 1 to a logic 0.<br>Its encoding is:<br>0: Device is ready.<br>1: Device is busy with an internal operation. |

## 5.4 Status Register 2

Table 14 shows the bit assignments for Status Register 2.

**Table 14. Status Register 2 Format**

| Bit # | Acronym | Name | Type | Default | Description |
|---|---|---|---|---|---|
| 7 | SUSP | Suspend Status | R | 0 | SUSP is set by hardware and indicates that the program or erase operation has been suspended. This bit is set after software executes an Erase/Program Suspend (75h) command. Hardware clears this bit when it detects any of the following conditions:<br>Erase/Program Resume (7Ah) command, Hardware Reset, JEDEC Hardware Reset, 66h / 99h Reset command |
| 6 | CMPRT | Complement Protect | R/W | 0 | CMPRT is used with BPSIZE, TB, and BP2:BP0 bits to provide additional memory array protection. Its encoding is:<br>0: Current protection mechanism is unchanged<br>1: Current protection mechanism is reversed<br>When set, unprotected areas of memory become protected, and protected areas of memory become unprotected. For example, when CMPRT = 0, a top 64 kB block can be protected, while the rest of the array is not; when CMP = 1, the same 64 kB block becomes unprotected, the rest of the array is then read-only. |
| 5:3 | SL3:SL1 | Security Lock 3:1 | R | 000 | SL3:SL1 gives the One-Time-Program (OTP) lock status of Security Registers 1 - 3. This field determines which of these Security Registers have been locked. Note that Security register 0 is always locked as the value is programmed at the factory.<br>The default state of SL[3:1] is 0: all registers are unlocked. Each bit corresponds to a Status Register, as follows:<br>SL3 (bit 5): Security register 3<br>SL2 (bit 4): Security register 2<br>SL1 (bit 3): Security register 1<br>Each bit is set by hardware when the most significant byte of the corresponding Security Register (the 128th byte) is programmed. For example, when the MSB of Security register 1 is set, bit 3 (SL1) of this field is set. Conversely, when the MSB of Security register 3 is set, bit 5 (SL3) of this field is set. Each time a bit is set, it indicates that the corresponding 128-Byte Security Register has become read-only permanently. |
| 2 | R | Reserved | R | 0 | Reserved. |
| 1 | QE | Quad Enable | R/W | 0 | QE enables Quad SPI and XiP operation.Its encoding is:<br>0: QE mode is disabled<br>1: QE mode is enabled<br>When QE is logic 0, the $\overline{\text{WP}}$ and $\overline{\text{HOLD}}$ / RESET pins are enabled.<br>When QE is logic 1, the $\overline{\text{WP}}$ and $\overline{\text{HOLD}}$ / RESET pins function as the $IO_2$ and $IO_3$ pins, respectively, and the $\overline{\text{WP}}$ and $\overline{\text{HOLD}}$ / RESET pins functions are disabled.<br>This bit only pertains to the following commands:<br>6Bh: Quad Output Read<br>EBh: XiP Mode Read<br>E7h: Quad I/O Read with Double-word Aligned Address<br>32h: Quad Output Program<br>77h: Set Burst with Wrap<br>94h: Quad I/O Manufacturer/Device ID |
| 0 | SRP1 | Status Register Protect 1 | R/W | 0 | SRP1 works with the SRP0 bit in Status Register 0 and the $\overline{\text{WP}}$ pin to control write protection. Types of protection include software protection, hardware protection, one-time programmable (OTP) protection, and reset lock-down protection. See Table 15 for an encoding of these bits. |

Table 15 shows the operation of the SRP[1:0] volatile Status Register bits during normal operation.

**Table 15. Status Register Protection During Normal Operation**

| SRP1 | SRP0 | SRLOCK | WP | Type of Protection | Description |
|---|---|---|---|---|---|
| 0 | 0 | x | x | Software Protected (protection is controlled by the CMPRT, BPSIZE, TB, BP[2:0], or WPS bits) | When both the SRP0 and SRP1 bits are 0, the Status Registers can be written by executing a Write Status Register command (01h, 31h, 11h, 71h) and setting the WEL bit of this register. Note that the $\overline{WP}$ pin has no meaning during this type of operation. |
| 0 | 1 | x | 0 | Hardware Protected (protection is controlled by the $\overline{WP}$ pin) | When the SRP[1:0] bits are 0, 1 respectively, and the $\overline{WP}$ pin is low, the registers are hardware protected and cannot be written to. |
| 0 | 1 | x | 1 | Hardware Unprotected (protection is controlled by the CMPRT, BPSIZE, TB, BP[2:0], or WPS bits) | When the SRP[1:0] bits are 0, 1 respectively, and the $\overline{WP}$ pin is high, the registers are not hardware protected and can be written to by executing a Write Status Register command (01h, 31h, 11h, 71h) and setting the WEL bit of this register. |
| 1 | 0 | x | x | Reset Lockdown | When the SRP[1:0] bits are 1, 0 respectively, the Status Registers are write protected and cannot be written until the device is reset (caused by a power-cycle, hardware reset, or software reset). After reset, hardware changes the state of the volatile SRP[1:0] bits to 0,0 as shown in Table 16. |
| 1 | 1 | 0 | x | Reset Lockdown | When the SRP[1:0] bits are 1,1 respectively, and the SRLOCK bit is 0, the Status Registers are write protected and cannot be written until the device is reset (caused by a power-cycle, hardware reset or software reset). After reset, hardware changes the state of the volatile SRP[1:0] bits to 0,1 as shown in Table 16. |
| 1 | 1 | 1 | x | OTP | When the SRP[1:0] bits are 1,1 respectively, and the SRLOCK bit is 1, the Status Registers are permanently write protected. |

Table 16 shows the behavior of the SRP1 and SPR0 bits after a reset condition.

**Table 16. Status Register Protection During Reset**

| Non-Volatile Memory Bits | | | Type of Protection | Description | Volatile Register Bits | |
|---|---|---|---|---|---|---|
| SRP1 | SRP0 | SRLOCK | | | SRP1 | SRP0 |
| 0 | 0 | x | Software Protected | The reset operation maintains the volatile SRP1 and SRP0 at 0,0. | 0 | 0 |
| 0 | 1 | x | Hardware Protected | The reset operation maintains the volatile SRP1 and SRP0 at 0,1. | 0 | 1 |
| 1 | 0 | x | Reset Lockdown | The reset operation changes the volatile SRP1 and SRP0 bits to 0,0. | 0 | 0 |
| 1 | 1 | 0 | Reset Lockdown | The reset operation changes the volatile SRP1 and SRP0 bits to 0,1. | 0 | 1 |
| 1 | 1 | 1 | One-Time Program (OTP) | The reset operation maintains the volatile SRP1 and SRP0 at 1,1. This makes the Status Registers permanently write protected. | 1 | 1 |

## 5.5 Status Register 3

Table 17 shows the bit assignments for Status Register 3.

**Table 17. Status Register 3 Format**

| Bit # | Acronym | Name | Type | Default | Description |
|---|---|---|---|---|---|
| 7 | $\overline{\text{HOLD}}$/ RESET | $\overline{\text{HOLD}}$ or RESET Function | R/W | 0 | Renesas Electronics provides a variety of packages for the AT25XE161D, as shown in Section 3. Each of these packages provides a dedicated $\overline{\text{HOLD}}$ / RESET pin that can be configured as a RESET pin, or as a $\overline{\text{HOLD}}$ pin, depending on the programming of this bit. Its encoding is:<br>0: HOLD function is enabled<br>1: RESET function is enabled<br>Note that this pin can only be configured as $\overline{\text{HOLD}}$ or RESET when the QE bit (see bit 1 of Status Register 2) is cleared (logic 0). If the QE bit is set, the pin functions as a dedicated data pin and the $\overline{\text{HOLD}}$ / RESET functionality is disabled. |
| 6:5 | DRV1:0 | Drive Level | R/W | 01[1] | Drive level. The DRV1 and DRV0 bits are used to determine the output driver strength during read operations. The driver strength is automatically adjusted with VCC level. The driver strength is encoded as following:<br>00: Reserved<br>01: 100% (increase 1.66X at low VCC)<br>10: 66% (increase 2X at low VCC)<br>11: 33% (increase 3X at low VCC) |
| 4:3 | R | Reserved | R | 0 | Reserved |
| 2 | WPS | Write Protection Select | R/W | 0 | The WPS bit selects the Write Protect scheme. Its encoding is:<br>0: The AT25XE161D uses a combination of CMPRT, BPSIZE, TB, and BP[2:0] bits in Status Register 1 to protect a specific area of the memory array.<br>1: The AT25XE161D uses the individual block locks to protect any individual block. The default value for all individual block lock bits is 1 upon device power on or after reset.<br>When this bit is set, software uses the Block Lock (36h) and Block Unlock (39h) commands to lock and unlock blocks of memory.<br>For more information on this functionality, see Section 4.8, Memory Protection. |
| 1:0 | R | Reserved | R | 0 | Reserved. |

1. Default driver strength was used for device test and characterization.To achieve optimal performance, it is recommended to adjust driver strength setting to match the user system load under application-specific environmental conditions.

## 5.6 Status Register 4

The following table shows the bit assignments for Status Register 4.

**Table 18. Status Register 4 Format**

| Bit # | Acronym | Name | Type | Default | Description |
|---|---|---|---|---|---|
| 7 | PDM | Power-Down Mode | R/W | 0 | This bit is used in conjunction with the Deep Power-Down command (B9h) to place the device into either Deep Power-Down mode, or Ultra-Deep Power-Down mode. Its encoding is:<br>0: Execution of the B9h command invokes Ultra-Deep Power-Down mode. This is the same as executing the UDPD command 79h. In this mode the SRAM buffer contents are not preserved.<br>1: Execution of the B9h command invokes Deep Power-Down mode. In this mode the SRAM buffer contents are preserved.<br>In both modes, the ABh command is required to exit the corresponding Power-Down mode. Simply toggling the $\overline{CS}$ pin does not result in exiting from Power-Down mode. For more information see Section 4.9, Power-Down Considerations. |
| 6 | SPM | Sequential Program Mode Status | R | 0 | The SPM bit indicates whether the device is in the Byte/Page Program mode or the Sequential Program Mode. The default state after power-up or device reset is the Byte/Page Program mode.<br>Its encoding is:<br>0: Byte/Page Programming Mode (default)<br>1: Sequential Programming Mode entered<br>If software sets this bit to 1, the address is only required for the first transfer of the sequential operation. Therefore, on the first transfer, command, address, and data are required. If there are subsequent operations, the address is not required. Software must only supply the command and data (on a write). |
| 5 | PE | Program Error | R | 0 | This bit is set by hardware whenever an error occurs during a program operation. |
| 4 | EE | Erase Error | R | 0 | This bit is set by hardware whenever an error occurs during an erase operation. |
| 3 | XiP | XiP Mode Select | R/W | 0 | This bit determines whether a command is required each time a read operation is executed using the E7h or EBh (Quad Read) commands. Its encoding is:<br>0: XiP mode (continuous read mode) is disabled<br>1: XiP mode (continuous read mode) is enabled<br>If this bit is set and either the E7h or EBh commands are executed, the command is only required for the first access (1-4-4). In this case, the command is transferred on the SI pin, and the address and data are transferred on the SI ($I/O_0$), SO ($I/O_1$), $\overline{WP}$ ($I/O_2$), and $\overline{HOLD}$ ($I/O_3$) pins. Subsequent accesses require only address and data (0-4-4).<br>Note that if either this bit or the QE bit (see bit 1 of Status Register 2) is 0, the device can never be placed into 0-4-4 mode, and a command is required for each access. |
| 2:0 | BWS[2:0] | Burst Wrap Settings | R | 001 | This 3-bit field maps to the W6:W4 bits of the Set Burst Wrap command (77h) to determine the burst wrap status and the wrap length. See Section 6.14 for more information on the Set Burst Wrap command. |

## 5.7 Status Register 5

The following table shows the bit assignments for Status Register 5.

**Table 19. Status Register 5 Format**

| Bit # | Acronym | Name | Type | Default | Description |
|---|---|---|---|---|---|
| 7 | SRLOCK | Status Register Lock | R | 0 | Hardware sets this bit when the user executes the Status Register Lock (6Fh) command, immediately followed by two verification bytes, 4Dh and 67h. Once this action occurs, the Status Registers can be permanently locked. |
| 6:4 | DC[2:0] | Dummy Clocks | R/W | 000 | This field indicates the number of dummy clocks to be inserted between the address and data transactions for the EBh and E7h commands. Note that the dummy clocks include the 2 clocks required to clock in the M[7:0] bits.<br>000: 2 clocks<br>001: 4 clocks<br>010: 6 clocks<br>011: 8 clocks<br>100: 10 clocks.<br>101 - 111: Reserved |
| 3 | ES | Erase Suspend | R | 0 | This bit is set by hardware whenever an erase operation is suspended. |
| 2 | PS | Program Suspend | R | 0 | This bit is set by hardware whenever a program operation is suspended. |
| 1 | TERE | Terminate Enable | R/W | 0 | The TERE bit enables or disables the Terminate command. Its encoding is:<br>0: Terminate command is disabled<br>1: Terminate command is enabled<br>When the TERE bit is cleared (the default state after power-up), the Terminate command is disabled and any attempts to reset the device using the Terminate command are ignored. When the TERE bit is set, the Terminate command is enabled.<br>The TERE bit retains its state as long as power is applied to the device. Once set, the TERE bit remains in that state until it is modified using the Write Status Register Byte 5 command or until the device has been power cycled. |
| 0 | DWA | Doubleword Aligned | R/W | 0 | Setting the DWA bit indicates that the devices adheres to double-word aligned addressing. This bit is only used when the EBh (XiP DWA Read) command is executed and is encoded as follows:<br>0: Double-word addressing is disabled<br>1: Double-word addressing is enabled<br>When this bit is set, the lower 2 bits of address are ignored and assumed to be 00. |

## 5.8 Status Register 6

The following table shows the bit assignments for Status Register 6.

**Table 20. Status Register 6 Format**

| Bit # | Acronym | Name | Type | Default | Description |
|---|---|---|---|---|---|
| 7:6 | LBS | Low Battery Status | R | 0 | This 2-bit field indicates the state of the low battery test. The status is reported when the Battery Status command (EFh) is executed. This field is encoded as follows:<br>00: No test in progress. Normal operation.<br>01: Battery test in progress.<br>10: Battery test complete. Result is >Vth<br>11: Battery test complete. Result is <Vth<br>This field can be reset by hardware when certain internal conditions are met. The user can reset this field by executing a Terminate command (F0h).<br>The battery threshold voltage (Vth) is selected by the user by programming bits 5:3 (LBVL) of this register. See below. |
| 5:3 | LBVL | Low Battery Voltage Level | R/W | 0 | This field sets the voltage threshold used to indicate whether the battery is OK, or nearing failure. This field is encoded as follows:<br>000: 1.8V<br>001: 2.0V<br>010: 2.2V<br>011: 2.4V<br>100: 2.6V<br>101: 2.8V<br>110: 3.0V<br>111: 3.2V<br>For example, if this field is set to 3’b110, indicating a threshold of 3.0V, and the result of the test is 2.0V, hardware would program a value of 2’b11 in the LBS field above. This indicates that the battery is below the target threshold of 3.0V and could be nearing failure. |
| 2:1 | LBLD | Low Battery Load | R/W | 0 | Load battery. This field selects the amount of load applied to the battery during the test. This field is encoded as follows:<br>00: 10 µA of load<br>01: 100 µA of load<br>10: 1 mA of load<br>11: 10 mA of load<br>Once the desired amount of loading is applied to the battery for the time determined by the state of the LBD bit below, the measured voltage is compared to the value programmed into the LBVL field in bits 5:3. The result is reported in the LBS field in bits 7:6 of this register. |
| 0 | LBD | Load Battery Delay | R/W | 0 | Load battery delay. This bit determines the amount of time the load selected by the LBLD field in bits 2:1 is applied to the battery. Its encoding is:<br>0: 100 µs<br>1: 1 ms |

# 6. Commands and Addressing

A valid command or operation must always be started by first asserting the $\overline{CS}$ pin. After the $\overline{CS}$ pin has been asserted, the host controller must then clock out a valid 8-bit command on the SPI bus. Following the command, information such as address and data bytes would then be clocked out by the host controller. All command, address, and data bytes are transferred with the most-significant bit (MSB) first. An operation is ended by deasserting the $\overline{CS}$ pin.

Commands not supported by the AT25XE161D are ignored by the device and no operation is started. The device continues to ignore any data presented on the SI pin until the start of the next operation ($\overline{CS}$ pin being deasserted and then reasserted). Also, if the $\overline{CS}$ pin is deasserted before the complete command and address information are sent to the device, then no operation is performed and the device simply returns to the idle state and waits for the next operation.

Depending on the command, up to three bytes of information may be required, representing address bits A23 - A0. Since the upper address limit of the AT25XE161D memory array is 1FFFFFh, address bits A23 - A21 are always ignored by the device.

**Table 21. Command Listing**

| Command Name | Sect. in Doc. | Hex Cmd | Transfer Type[1] | | Command | | Address | | Dummy | | Data | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Mode | Format | Bytes | Clocks | Bytes | Clocks | Bytes | Clocks | Bytes[2] | Clocks |
| **Read Commands** | | | | | | | | | | | | |
| Read Array | 6.1 | 03h | SPI | 1-1-1 | 1 | 8 | 3 | 24 | 0 | 0 | Var. | Var. x 8 |
| Fast Read Array | 6.1 | 0Bh | SPI | 1-1-1 | 1 | 8 | 3 | 24 | 1 | 8 | Var. | Var. x 8 |
| Dual Output Read Array | 6.2 | 3Bh | Dual Output | 1-1-2 | 1 | 8 | 3 | 24 | 1 | 8 | Var. | Var. x 4 |
| Quad Output Read Array | 6.3 | 6Bh | Quad Output | 1-1-4 | 1 | 8 | 3 | 24 | 1 | 8 | Var. | Var. x 2 |
| XiP Mode Read Array (initial transfer) | 6.4 | EBh | XiP | 1-4-4 | 1 | 8 | 3 | 6 | Var. | Var. x 2 | Var. | Var. x 2 |
| XiP Mode Read Array (subsequent transfers) | 6.4 | --- | XiP | 0-4-4 | 0 | 0 | 3 | 6 | Var. | Var. x 2 | Var. | Var. x 2 |
| XiP Mode Read Array - DWA addr (initial transfer) | 6.4 | E7h | XiP | 1-4-4 | 1 | 8 | 3 | 6 | Var. | Var. x 2 | Var. | Var. x 2 |
| XiP Mode Read Array - DWA addr, subsequent transfers | 6.4 | --- | XiP | 0-4-4 | 0 | 0 | 3 | 6 | Var. | Var. x 2 | Var. | Var. x 2 |
| **Program/Erase Commands** | | | | | | | | | | | | |
| Page Erase (256b) | 6.5 | 81h/ DBh | SPI | 1-1-0 | 1 | 8 | 3 | 24 | 0 | 0 | 0 | 0 |
| Block Erase (4 Kbytes) | 6.6 | 20h | SPI | 1-1-0 | 1 | 8 | 3 | 24 | 0 | 0 | 0 | 0 |
| Block Erase (32 Kbytes) | 6.6 | 52h | SPI | 1-1-0 | 1 | 8 | 3 | 24 | 0 | 0 | 0 | 0 |
| Block Erase (64 Kbytes) | 6.6 | D8h | SPI | 1-1-0 | 1 | 8 | 3 | 24 | 0 | 0 | 0 | 0 |
| Chip Erase | 6.7 | 60h | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 6.7 | C7h | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Byte/Page Program (1 - 256 bytes) | 6.8 | 02h | SPI | 1-1-1 | 1 | 8 | 3 | 24 | 0 | 0 | Var. (1 - 256) | Var. x 8 |
| Sequential Program Mode[3] (first command) | 6.9 | ADh/ AFh | SPI | 1-1-1 | 1 | 8 | 3 | 24 | 0 | 0 | 1 | 8 |

**Table 21. Command Listing** (Continued)

| Command Name | Sect. in Doc. | Hex Cmd | Transfer Type[1] | | Command | | Address | | Dummy | | Data | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Mode | Format | Bytes | Clocks | Bytes | Clocks | Bytes | Clocks | Bytes[2] | Clocks |
| Sequential Program Mode (subsequent commands) | 6.9 | ADh/ AFh | SPI | 1-0-1 | 1 | 8 | 0 | 0 | 0 | 0 | 1 | 8 |
| Dual Output Byte/Page Program | 6.10 | A2h | Dual Output | 1-1-2 | 1 | 8 | 3 | 24 | 0 | 0 | Var. (1 - 256) | Var. x 4 |
| Quad Output Byte/Page Program | 6.11 | 32h | Quad Output | 1-1-4 | 1 | 8 | 3 | 24 | 0 | 0 | Var. (1 - 256) | Var. x 2 |
| Erase/Program Suspend | 6.12 | 75h/ B0h | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Erase/Program Resume | 6.13 | 7Ah/ D0h | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Set Burst with Wrap | 6.14 | 77h | Quad I/O | 1-4-4 | 1 | 8 | 3[4] | 6 | 0 | 0 | 1[5] | 2 |
| **Buffer Commands** | | | | | | | | | | | | |
| Buffer Read | 6.15 | D4h | SPI | 1-1-1 | 1 | 8 | 3[6] | 24 | 1 | 8 | Var. (1 - 256) | Var. x 8 |
| Buffer Write | 6.16 | 84h | SPI | 1-1-1 | 1 | 8 | 3(6) | 24 | 0 | 0 | Var. (1 - 256) | Var. x 8 |
| Buffer to Main Memory Page Program without Erase | 6.17 | 88h | SPI | 1-1-0 | 1 | 8 | 3[7] | 24 | 0 | 0 | 0 | 0 |
| Protection Commands | | | | | | | | | | | | |
| Write Enable | 6.18 | 06h | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Write Disable | 6.19 | 04h | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Volatile Status Register Write Enable | 6.20 | 50h | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Individual Block Lock | 6.21 | 36h | SPI | 1-1-0 | 1 | 8 | 3 | 24 | 0 | 0 | 0 | 0 |
| Individual Block Unlock | 6.22 | 39h | SPI | 1-1-0 | 1 | 8 | 3 | 24 | 0 | 0 | 0 | 0 |
| Read Block Lock | 6.23 | 3Ch/ 3Dh | SPI | 1-1-1 | 1 | 8 | 3 | 24 | 0 | 0 | 1[8] | 8 |
| Global Block Lock | 6.24 | 7Eh | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Global Block Unlock | 6.25 | 98h | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Security Register Commands** | | | | | | | | | | | | |
| Program OTP Security Register | 6.26 | 9Bh | SPI | 1-1-1 | 1 | 8 | 3[9] | 24 | 0 | 0 | Var. (1 - 128) | Var. x 8 |
| Read OTP Security Register | 6.27 | 4Bh | SPI | 1-1-1 | 1 | 8 | 3(9) | 24 | 1 | 8 | Var. (1 - 128) | Var. x 8 |
| **Status Register Commands** | | | | | | | | | | | | |
| Read Status Register 1 | 6.28 | 05h | SPI | 1-0-1 | 1 | 8 | 0 | 0 | 0 | 0 | 1 | 8 |
| Read Status Register 2 | 6.28 | 35h | SPI | 1-0-1 | 1 | 8 | 0 | 0 | 0 | 0 | 1 | 8 |
| Read Status Register 3 | 6.28 | 15h | SPI | 1-0-1 | 1 | 8 | 0 | 0 | 0 | 0 | 1 | 8 |
| Read Status Register 4 | 6.29 | 65h | SPI | 1-1-1 | 1 | 8 | 1 | 8 | 1 | 8 | 1 | 8 |
| Read Status Register 5 | 6.29 | 65h | SPI | 1-1-1 | 1 | 8 | 1 | 8 | 1 | 8 | 1 | 8 |

**Table 21. Command Listing** (Continued)

| Command Name | Sect. in Doc. | Hex Cmd | Transfer Type[1] | | Command | | Address | | Dummy | | Data | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Mode | Format | Bytes | Clocks | Bytes | Clocks | Bytes | Clocks | Bytes[2] | Clocks |
| Read Status Register 6 | 6.29 | 65h | SPI | 1-1-1 | 1 | 8 | 1 | 8 | 1 | 8 | 1 | 8 |
| Write Status Register 1 | 6.30 | 01h | SPI | 1-0-1 | 1 | 8 | 0 | 0 | 0 | 0 | 1[10] | 8 |
| Write Status Register 2 | 6.30 | 31h | SPI | 1-0-1 | 1 | 8 | 0 | 0 | 0 | 0 | 1 | 8 |
| Write Status Register 3 | 6.30 | 11h | SPI | 1-0-1 | 1 | 8 | 0 | 0 | 0 | 0 | 1 | 8 |
| Write Status Register 4 | 6.31 | 71h | SPI | 1-1-1 | 1 | 8 | 1 | 8 | 0 | 0 | 1 | 8 |
| Write Status Register 5 | 6.31 | 71h | SPI | 1-1-1 | 1 | 8 | 1 | 8 | 0 | 0 | 1 | 8 |
| Write Status Register 6 | 6.31 | 71h | SPI | 1-1-1 | 1 | 8 | 1 | 8 | 0 | 0 | 1 | 8 |
| Read Status Registers | 6.29 | 65h | SPI | 1-1-1 | 1 | 8 | 1[11] | 8 | 1 | 8 | Var.[12] (1 - 6) | Var. x 8 |
| Write Status Registers | 6.31 | 71h | SPI | 1-1-1 | 1 | 8 | 1(11) | 8 | 0 | 0 | 1 | 8 |
| Status Register Lock | 6.32 | 6Fh | SPI | 1-0-1 | 1 | 8 | 0 | 0 | 0 | 0 | 2 | 16 |
| Power-Down Commands | | | | | | | | | | | | |
| Deep Power-Down[13] | 6.33 | B9h | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Ultra-Deep Power-Down[14] | 6.35 | 79h/B9h | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Resume from Deep Power-Down | 6.34 | ABh | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Resume from Deep Power-Down with Device ID | 6.34 | ABh | SPI | 1-1-1 | 1 | 8 | 3 | 24 | 0 | 0 | 1 | 8 |
| Resume from Ultra-Deep Power-Down | 6.34 | ABh | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Reset Commands | | | | | | | | | | | | |
| Enable Reset | 6.36 | 66h [15] | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Reset Device | 6.36 | 99h | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Terminate | 6.37 | F0h | SPI | 1-0-1 | 1 | 8 | 0 | 0 | 0 | 0 | 1 | 8 |
| **Manufacturer/Device Commands** | | | | | | | | | | | | |
| Manufacturer/Device ID | 6.38 | 90h | SPI | 1-1-1 | 1 | 8 | 3 | 24 | 0 | 0 | 2[16] | 16 |
| Manufacturer/Device ID | 6.39 | 94h | Quad I/O | 1-4-4 | 1 | 8 | 3 | 6 | 1 | 2 | 2(16) | 4 |
| JEDEC ID | 6.40 | 9Fh | SPI | 1-0-1 | 1 | 8 | 0 | 0 | 0 | 0 | 5[17] | 40 |
| Miscellaneous Commands | | | | | | | | | | | | |
| Active Status Interrupt[18] | 6.41 | 25h | SPI | 1-0-1 | 1 | 8 | 0 | 0 | Var. | Var. x 8 | 0 | 0 |
| Read-Modify-Write | 6.42 | 0Ah | SPI | 1-1-1 | 1 | 8 | 3 | 24 | 0 | 0 | Var. | Var. x 8 |
| Start Low Battery Detect | 6.43 | EFh | SPI | 1-0-0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read SFDP | 6.44 | 5Ah | SPI | 1-1-1 | 1 | 8 | 3 | 24 | 1 | 8 | Var.[19] | Var. x 8 |

1. Indicates command, address, and data, and the number of pins each type is driven on. For example, 1-1-2 indicates that the command is driven on one pin (SI), address is driven on one pin (SI), and data is driven on two pins (SI and SO).
2. Var = Variable.

3. For the initial transfer of the ADh/AFh command, the command, address, and data are required. For all subsequent command, only the command and data are necessary. The address field is not required and increments automatically within the device.
4. For the 77h command, 24 bits are transferred in the address field, but these bits are don't care and are not used in the operation.
5. For the 77h command, the data byte consists of wrap bits W[7:0]. Bits 6:4 of this value are the only valid bits of the byte and are used to set the wrap mode.
6. For the D4h/84h commands, only the least significant byte of address is used. The upper 16 bits of address are don't care and are not used in the operation.
7. For the 88h command, only the upper 16 bits of address are used. The lower 8 bits are don't care and are not used in the operation.
8. For the 3Ch/3Dh command, the data byte returned provides the lock status for the 4 kB block relative to the address provided.
9. For the 9Bh and 4Bh commands, only the lower 9 bits of address (A[8:0]) are valid. Address bits 23:9 are don't care.
10. For compatibility with legacy devices, command 01h can also be used with 2 bytes of data. In such case, the second byte is written to Status Register 2.
11. The 65h and 71h commands require only one byte of address.
12. For the 65h command, if the initial value in the address field is 01h, pointing to SR1, all of the Status Registers can be read in one operation as long as the $\overline{CS}$ pin is held low. Once the data for SR1 is fetched, hardware increments the address automatically and fetch the contents of SR2, etc. until all six registers have been read.
13. Deep Power-Down mode can be entered by executing the B9h command with bit 7 (PDM) of Status Register 4 set.
14. Ultra-Deep Power-Down mode can be entered by either executing the 79h command, or by executing the B9h command with bit 7 (PDM) of Status Register 4 cleared.
15. The 66h and 99h commands are used together and form back-to-back 1-0-0 sequences on the bus.
16. For the 90h and 94h commands, two data bytes are output. The first byte is the manufacturer ID, and the second byte is the device ID.
17. For the 9Fh command, five bytes are output. First byte is the manufacturer ID, second byte is Device ID 1, third byte is Device ID 2, fourth byte is the extended string length, and the fifth byte is the extended string value.
18. For the 25h command, the RDY/BSY status is continually output onto the SO pin as long as the $\overline{CS}$ pin remains asserted. In SPI mode 0, the number of dummy bytes/ is 0 and the number of dummy clocks is 0. In SPI mode 3, the number of dummy bytes/ is 1 and the number of dummy clocks is 8.
19. For the 5Ah command, hardware outputs the data associated with the address provided, then increments the address automatically and continues to output successive locations as long as the $\overline{CS}$ pin remains asserted. To read the entire SFDP value, software must program the initial address as 000000h to access the first address, the leave $\overline{CS}$ asserted until the entire SFDP is read out.

## 6.1 Read Array (03h, 0Bh)

The *Read Array* command can be used to sequentially read a continuous stream of data from the device by simply providing the clock signal once the initial starting address is specified. The device incorporates an internal address counter that automatically increments every time one byte of data is output by the device.

Two commands (0Bh and 03h) can be used to read the memory array. The use of each command depends on the maximum clock frequency to read data from the device. The 0Bh command can be used at any clock frequency up to the maximum specified by $f_{SCK}$, and the 03h command can be used for lower frequency read operations up to the maximum specified by $f_{RDLF}$.

### 6.1.1 Transfer Format

The 03h/0Bh command follows the 1-1-1 transfer format described in Section 4.2, where the command and address are transferred on the SI pin, and data is transferred on the SO pin. See Figure 7 for timing diagram of a 1-1-1 transfer showing the toggling of external bus signals for a read operation without dummy bytes. Figure 8 shows a read operation with dummy bytes.

### 6.1.2 Transfer Sequence

To perform the *Read Array* command, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- Assert the $\overline{CS}$ pin.
- The appropriate command (0Bh or 03h) is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to read within the memory array. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first.
- Following the three address bytes, an additional dummy byte must be clocked into the device if the 0Bh command is used.
- Data is output on the SO pin. Each byte transfer requires eight clock cycles. The data is always output with the most significant bit of the byte transferred first. The number of bytes transferred is determined by software. The transfer can be anywhere from a single byte to the entire memory array. The most significant bit of each data byte is transferred first.
- Deasserting the $\overline{CS}$ pin terminates the read operation and puts the SO pin into high-impedance state. The $\overline{CS}$ pin can be deasserted at any time and does not require a full byte of data be read.

When the last byte (1FFFFFh) of the memory array has been read, the device loops back and continues reading at the beginning of the array (000000h). No delays are incurred when wrapping around from the end of the array to the beginning of the array.

## 6.2 Dual Output Read Array (3Bh)

The Dual Output Read Array command is similar to the standard Read Array (03h/0Bh) command and can be used to sequentially read a continuous stream of data from the device by simply providing the clock signal once the initial starting address has been specified. Unlike the standard Read Array command, however, the Dual-Output Read Array command allows two bits of data to be clocked out of the device on every clock cycle, rather than just one.

### 6.2.1 Transfer Format

The 3Bh command follows the 1-1-2 transfer format described in Section 4.3, where the command and address are transferred on the SI pin, and output data is transferred on the SI and SO pin. To facilitate this transfer, hardware switches the SI pin to an output as soon as the command has been decoded and the address transferred. See Figure 10 for a timing diagram of this command.

### 6.2.2 Transfer Sequence

To perform the Dual Output Read Array operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 3Bh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to read within the memory array. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first.
- Following the three address bytes, an additional dummy byte must be clocked into the device. This requires eight clock cycles to complete.
- Data is output on the SI and SO pins. As a result, each byte transfer requires four clock cycles, half that of the 03h/0Bh commands. The data is always output with the most significant bit of the byte transferred first. The number of bytes transferred is determined by software. The transfer can be anywhere from a single byte to the entire memory array. The data is driven onto the bus as follows:
  - 1st data clock: bit 7 is output on the SO pin
  - 1st data clock: bit 6 is output on the SI pin
  - 2nd data clock: bit 5 is output on the SO pin
  - 2nd data clock: bit 4 is output on the SI pin
  - 3rd data clock: bit 3 is output on the SO pin
  - 3rd data clock: bit 2 is output on the SI pin
  - 4th data clock: bit 1 is output on the SO pin
  - 4th data clock: bit 0 is output on the SI pin
  - Subsequent bytes are output with each additional 4 clocks. The sequence above continues with each byte of data being output after every four clock cycles.
- Deasserting the $\overline{CS}$ pin terminates the read operation and puts the SO and SI pins into a high-impedance state. The $\overline{CS}$ pin can be deasserted at any time and does not require that a full byte of data be read.

When the last byte (1FFFFFh) of the memory array has been read, the device continues reading from the beginning of the array (000000h). No delays are incurred when wrapping around from the end of the array to the beginning of the array.

## 6.3 Quad Output Read Array (6Bh)

The Quad Output Read Array command is similar to the Dual Output Read Array command, except that the Quad Output Read Array command allows four bits of data to be clocked out of the device on every clock cycle, rather than just one or two. Note that the Quad Enable bit (QE) in Status Register 2 (SR2) must be set to enable this command. The Quad Output Read Array command can be used at any clock frequency, up to the maximum specified by $f_{SCK.}$

### 6.3.1 Transfer Format

The 6Bh command follows the 1-1-4 transfer format described in Section 4.4, where the command and address are transferred on the SI pin, and data is transferred on the $\overline{\text{HOLD}}$, $\overline{\text{WP}}$, SO, and SI pins. See Figure 12 for a timing diagram of this transaction.

### 6.3.2 Transfer Sequence

To perform the Quad Output Read Array operation, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The 6Bh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to read within the memory array. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first.
- Following the three address bytes, an additional dummy byte must be clocked into the device. This requires eight clock cycles to complete.
- Following transfer of the dummy byte, hardware switches the $\overline{\text{HOLD}}$, $\overline{\text{WP}}$, and SI pins to outputs.
- Data is output on the $\overline{\text{HOLD}}$, $\overline{\text{WP}}$, SO, and SI pins. As a result, each byte transfer requires two clock cycles, one-fourth that of the 03h/0Bh commands. The data is always output with the most significant bit of the byte transferred first. The number of bytes transferred is determined by software. The transfer can be anywhere from a single byte to the entire memory array. Each data byte is shifted out of the device as follows:
  - First data clock: bit 7 is output on the $\overline{\text{HOLD}}$ pin
  - First data clock: bit 6 is output on the $\overline{\text{WP}}$ pin
  - First data clock: bit 5 is output on the SO pin
  - First data clock: bit 4 is output on the SI pin
  - Second data clock: bit 3 is output on the $\overline{\text{HOLD}}$ pin
  - Second data clock: bit 2 is output on the $\overline{\text{WP}}$ pin
  - Second data clock: bit 1 is output on the SO pin
  - Second data clock: bit 0 is output on the SI pin
- Deasserting the $\overline{\text{CS}}$ pin terminates the read operation and puts the $\overline{\text{HOLD}}$, $\overline{\text{WP}}$, SO, SI pins into a high-impedance state. The $\overline{\text{CS}}$ pin can be deasserted at any time and does not require that a full byte of data be read.

When the last byte (1FFFFFh) of the memory array has been read, the device continues reading from the beginning of the array (000000h). No delays are incurred when wrapping around from the end of the array to the beginning of the array.

## 6.4 XiP Mode Read (EBh), XiP Mode Read with Double-word Aligned Address (E7h)

The XiP Read command (EBh) is similar to the Quad Output Read Array command, except that four bits of address are clocked into the device on every clock cycle, rather than just one. The E7h command is similar to the EBh command but works only on double-word aligned (DWA) addresses. As such, the E7h command ignores address bits A[1:0] and internally assumes that these two bits are always 2'b00. This allows us to run the E7h command at faster clock speeds or fewer dummy clock cycles compared to the EBh command.

The EBh command can operate in either DWA mode or non-DWA mode depending on the state of the DWA bit in Status Register 5. See Section 5.7, Status Register 5 for more information.

### 6.4.1 Transfer Format

The initial EBh/E7h command follows the 1-4-4 transfer format described in Section 4.6, where the command is transferred on the SI pin, and the address and data are transferred on the $\overline{\text{HOLD}}$, $\overline{\text{WP}}$, SO, and SI pins. See Figure 15 for a timing diagram of this command.

Subsequent EBh/E7h commands follow the 0-4-4 transfer format described in Section 4.6, where the address and data are transferred on the $\overline{\text{HOLD}}$, $\overline{\text{WP}}$, SO, and SI pins. No command transfer is required. See Figure 16 for a timing diagram of this command.

### 6.4.2 Mode Bits

The EBh and E7h commands follow the 1-4-4 and 0-4-4 transfer formats as described above. During the initial transfer, an 8-bit mode field is transferred immediately following the last address byte; it is decoded by hardware to determine if the device is placed into XiP continuous read mode. If so, then subsequent transfers do not require the command field, resulting in a 0-4-4 transfer type. This mode increases performance by saving 8 clocks cycles per transfer. For more information on this mode, see Section 4.6, XiP Mode Operation.

### 6.4.3 Transfer Sequence — Initial Transfer

To perform the initial XiP Mode Read operation, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The EBh/E7h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first. This field is transferred on the SI pin.
- Three address bytes are clocked in to specify the starting address location of the first byte to read within the memory array. A total of 6 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first. Each address byte is shifted out of the device as follows:
  - First clock: address bit 23 is input on the $\overline{\text{HOLD}}$ pin
  - First clock: address bit 22 is input on the $\overline{\text{WP}}$ pin
  - First clock: address bit 21 is input on the SO pin
  - First clock: address bit 20 is input on the SI pin
  - Second clock: address bit 19 is input on the $\overline{\text{HOLD}}$ pin
  - Second clock: address bit 18 is input on the $\overline{\text{WP}}$ pin
  - Second clock: address bit 17 is input on the SO pin
  - Second clock: address bit 16 is input on the SI pin
  - Third and fourth clocks: A15:A8 are shifted in same as above
  - Fifth and sixth clocks: A7:A0 are shifted in same as above
- Following the three address bytes, a programmable number of dummy bytes must be clocked into the device. See the Section 6.4.5 below for more information. Note that the Mode bits are treated as part of the dummy bytes and must be clocked in to the device.
- Following transfer of the dummy bytes, hardware switches the $\overline{\text{HOLD}}$, $\overline{\text{WP}}$, and SI pins to outputs.

- Data is output on the $\overline{\text{HOLD}}$, $\overline{\text{WP}}$, SO, and SI pins. As a result, each byte transfer requires two clock cycles. The data is always output with the most significant bit of the byte transferred first. Each data byte is shifted out of the device as follows:
    - First clock: bit 7 is output on the $\overline{\text{HOLD}}$ pin
    - First clock: bit 6 is output on the $\overline{\text{WP}}$ pin
    - First clock: bit 5 is output on the SO pin
    - First clock: bit 4 is output on the SI pin
    - Second clock: bit 3 is output on the $\overline{\text{HOLD}}$ pin
    - Second clock: bit 2 is output on the $\overline{\text{WP}}$ pin
    - Second clock: bit 1 is output on the SO pin
    - Second clock: bit 0 is output on the SI pin
- Deasserting the $\overline{\text{CS}}$ pin terminates the read operation and puts the SI ($IO_0$), SO ($IO_1$), $\overline{\text{WP}}$ ($IO_2$), and $\overline{\text{HOLD}}$ ($IO_3$) pins into a high-impedance state. The $\overline{\text{CS}}$ pin can be deasserted at any time and does not require that a full byte of data be read.
- When the last byte (1FFFFFh) of the memory array has been read, the device continues reading from the beginning of the array (000000h). No delays are incurred when wrapping around from the end of the array to the beginning of the array.

### 6.4.4 Transfer Sequence — Subsequent Transfers

To perform subsequent XiP Mode Read operations, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- Three address bytes are clocked in to specify the starting address location of the first byte to read within the memory array. A total of 6 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first. The address is transferred on the bus in the same manner described in Section 6.4.3 above.
- Following the three address bytes, a programmable number of dummy bytes must be clocked into the device. See Section 6.4.5 below for more information. Note that the Mode bits are treated as part of the dummy bytes and must be clocked in to the device.
- Data is output on the SI ($IO_0$), SO ($IO_1$), $\overline{\text{WP}}$ ($IO_2$), and $\overline{\text{HOLD}}$ ($IO_3$) pins. As a result, each byte transfer requires two clock cycles. The data is always output with the most significant bit of the byte transferred first. Data is shifted out onto the bus in the same manner described in Section 6.4.3 above.
- Deasserting the $\overline{\text{CS}}$ pin terminates the read operation and puts the SI ($IO_0$), SO ($IO_1$), $\overline{\text{WP}}$ ($IO_2$), and $\overline{\text{HOLD}}$ ($IO_3$) pins into a high-impedance state. The $\overline{\text{CS}}$ pin can be deasserted at any time and does not require that a full byte of data be read.

When the last byte (1FFFFFh) of the memory array has been read, the device continues reading from the beginning of the array (000000h). No delays are incurred when wrapping around from the end of the array to the beginning of the array.

### 6.4.5 Programmable Number of Dummy Clocks

The number of dummy clocks required differs depending on frequency and the type of command being performed and the state of bit 0 (DWA) of Status Register 5. This relationship is shown in Table 22 and Table 23. The number of Dummy clocks is set using bits 6:4 (DC[2:0]) of Status Register 5. See Section 5.7, Status Register 5 for more information. Note that the number of dummy clocks includes the 2 SCK clock periods required to clock in the M[7:0] mode bits.

**Table 22. Frequency and Number of Dummy Clocks Based on Command Type in Non-Wrap Mode (default)**

| Cmd | DWA Bit | DC[2:0] | Dummy Clocks | XiP enabled (044) (1.65 V - 3.6 V) | XiP enabled (044) (2.7 V - 3.6 V) | XiP disabled (144) (1.65 V - 3.6 V) | XiP disabled (144) (2.7 V - 3.6 V) |
|---|---|---|---|---|---|---|---|
| | | | | SCK Frequency (MHz) | | | |
| EBh | 0 | 000 | 2 | 30 | 35 | 20 | 25 |
| | 0 | 001 | 4 | 50 | 60 | 40 | 45 |
| | 0 | 010 | 6 | 80 | 80 | 55 | 65 |
| | 0 | 011 | 8 | 96 | 96 | 70 | 80 |
| | 0 | 100 | 10 | 96 | 96 | 90 | 96 |
| | 1 | 000 | 2 | 50 | 60 | 65 | 65 |
| | 1 | 001 | 4 | 120 | 133 | 120 | 133 |
| | 1 | 010 | 6 | 120 | 133 | 120 | 133 |
| | 1 | 011 | 8 | 120 | 133 | 120 | 133 |
| | 1 | 100 | 10 | 120 | 133 | 120 | 133 |
| E7h | x | 000 | 2 | 40 | 50 | 45 | 45 |
| | x | 001 | 4 | 96 | 96 | 80 | 90 |
| | x | 010 | 6 | 108 | 108 | 108 | 108 |
| | x | 011 | 8 | 108 | 108 | 108 | 108 |
| | x | 100 | 10 | 108 | 108 | 108 | 108 |

As shown in Table 22, the EBh command can operate on both doubleword aligned (DWA = 1) and non-doubleword aligned (DWA = 0) addresses. The E7h command works only on doubleword aligned addresses, so the state of the DWA bit is ignored.

Table 23. Frequency and Number of Dummy Clocks Based on Command Type in Wrap Mode (77h)

| Cmd | DWA Bit | DC[2:0] | Dummy Clocks | XiP enabled (044) (1.65 V - 3.6 V) | XiP enabled (044) (2.7 V - 3.6 V) | XiP disabled (144) (1.65 V - 3.6 V) | XiP disabled (144) (2.7 V - 3.6 V) |
|---|---|---|---|---|---|---|---|
| | | | | SCK Frequency (MHz) | | | |
| EBh | 0 | 000 | 2 | 30 | 30 | 30 | 30 |
| | 0 | 001 | 4 | 45 | 50 | 60 | 60 |
| | 0 | 010 | 6 | 75 | 80 | 80 | 85 |
| | 0 | 011 | 8 | 90 | 120 | 80 | 96 |
| | 0 | 100 | 10 | 96 | 120 | 85 | 100 |
| | 1 | 000 | 2 | 50 | 55 | 55 | 55 |
| | 1 | 001 | 4 | 85 | 108 | 80 | 96 |
| | 1 | 010 | 6 | 104 | 133 | 90 | 104 |
| | 1 | 011 | 8 | 108 | 133 | 96 | 108 |
| | 1 | 100 | 10 | 108 | 133 | 96 | 108 |
| E7h | x | 000 | 2 | 50 | 55 | 55 | 55 |
| | x | 001 | 4 | 90 | 108 | 80 | 96 |
| | x | 010 | 6 | 104 | 133 | 85 | 104 |
| | x | 011 | 8 | 108 | 133 | 90 | 108 |
| | x | 100 | 10 | 108 | 133 | 90 | 108 |

As shown in Table 23, the EBh command can operate on both doubleword aligned (DWA = 1) and non-doubleword aligned (DWA = 0) addresses. The E7h command works only on doubleword aligned addresses, so the state of the DWA bit is ignored.

## 6.5 Page Erase (81h/DBh)

The Page Erase command can be used to individually erase any page in the main memory array. The Main Memory Byte/Page Program command can be utilized at a later time.

### 6.5.1 Command Prerequisites

Before a Page Erase command can be issued, the Write Enable (06h) command must have been previously issued to the device to set the WEL bit of Status Register 1.

### 6.5.2 Transfer Format

The 81h/DBh command follows the 1-1-0 transfer format described in Section 4.2, where the command and address are transferred on the SI pin. The address represents the page to be erased. Hence no data is transferred for this command.

### 6.5.3 Transfer Sequence

To perform the Page Erase operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 81h/DBh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the address location to be erased within the memory array. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first. The address is transferred as follows:
  - A[23:21] are don't care bits
  - A[20:8] contain the PA[12:0] page address bits that identifies the page to be erased
  - A[7:0] are don't care bits

When a low-to-high transition occurs on the CS pin, the device begins the erase of the selected page (the erased state is a Logic 1). The erase operation is internally self-timed and takes place in a maximum time of $t_{PE}$. During this time, the RDY/BUSY bit in the Status Register indicates that the device is busy.

### 6.5.4 Active Status Interrupt

Alternatively, the Active Status Interrupt (25h) can be used to determine when the erase operation has completed. When this command is used, it is not necessary to continuously read the RDY/BSY bit to determine when the command has completed. Instead, hardware drives the value of the RDY/BSY bit into the SO pin. The state of the pin is updated every clock cycle. The host can monitor the SO pin until it toggles from a logic 1 to a logic 0, indicating that the operation has completed. For more information, see Section 4.14, Active Status Interrupt.

### 6.5.5 Command Status

While the device is executing a successful erase cycle, bit 0 in Status Register 1 (SR1) indicates that the device is busy. For faster throughput, it is recommended that Status Register 1 be polled rather than waiting the $t_{PE}$ time to determine if the device has finished erasing. At some point before the erase cycle completes, the WEL bit in the Status Register is reset back to the logical 0 state.

### 6.5.6 Programming Restrictions

The Page Erase command adheres to the following programming restrictions.

- The CS pin must be deasserted on a byte boundary (multiples of eight bits). Otherwise, the device aborts the operation and no action is taken.
- Despite the lower order address bits not being decoded by the device, the complete three address bytes must still be clocked into the device before the CS pin is deasserted. If the address is incomplete, the operation is aborted.

### 6.5.7 Error Reporting

If the memory is in the protected state, the Block Erase command cannot be executed and the device returns to the idle state once the CS pin has been deasserted.

- The device incorporates an intelligent erase algorithm that can detect when a byte location fails to erase properly. If an erase error occurs, it is indicated by the EE and PE bits in Status Register 4. See Section 5.6, Status Register 4 for more information.

## 6.6 Block Erase (20h, 52h, D8h)

A block of 4-, 32-, or 64 Kilobytes (kB) can be erased (all bits set to the logical 1 state) in a single operation by using one of three different forms of the Block Erase command.

- The 20h command is used for a 4 kB erase
- The 52h command is used for a 32 kB erase
- The D8h command is used for a 64 kB erase

### 6.6.1 Command Prerequisites

Before a Block Erase command can be issued, the Write Enable (06h) command must have been previously issued to the device to set the WEL bit of Status Register 1.

### 6.6.2 Transfer Format

The 20h/52h/D8h command follows the 1-1-0 transfer format described in Section 4.2, where the command and address are transferred on the SI pin. The address represents the block to be erased. No data is transferred for this command.

### 6.6.3 Transfer Sequence

To perform the Block Erase operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The appropriate command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the address location within the 4-, 32-, or 64-kB block to be erased. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first.

See Table 21 for more information on the transfer sequence for this command.

### 6.6.4 Erase Operation

When the $\overline{CS}$ pin is deasserted, the device starts the erase of the appropriate block. The erasing of the block is internally self-timed and takes place in a time of $t_{BLKE}$. Since the Block Erase command erases a region of bytes, the lower order address bits do not need to be decoded by the device. Depending on the command issued, the address is treated as follows:

- For a 4 kB erase, address bits A11 - A0 are ignored by the device and their values can be either a logical 1 or 0".
- For a 32 kB erase, address bits A14 - A0 are ignored by the device.
- For a 64 kB erase, address bits A15 - A0 are ignored by the device.

### 6.6.5 Command Status

While the device is executing a successful erase cycle, bit 0 in Status Register 1 (SR1) indicates that the device is busy. For faster throughput, it is recommended that Status Register 1 be polled rather than waiting the $t_{BLKE}$ time to determine if the device has finished erasing. At some point before the erase cycle completes, the WEL bit in the Status Register is reset back to the logical 0" state.

### 6.6.6 Active Status Interrupt

Alternatively, the Active Status Interrupt (25h) can be used to determine when the erase operation has completed. When this command is used, it is not necessary to continuously read the $\overline{RDY}$/BSY bit to determine when the command has completed. Instead, hardware drives the value of the $\overline{RDY}$/BSY bit into the SO pin. The state of the pin is updated every clock cycle. The host can monitor the SO pin until it toggles from a logic 1 to a logic 0, indicating that the operation has completed. For more information, see Section 4.14, Active Status Interrupt.

### 6.6.7 Programming Restrictions

The Block Erase commands adhere to the following programming restrictions. The following events can cause the block erase operation to be aborted. If any of these events occur, the WEL bit in Status Register 1 is reset back to the logic 0 state.

- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiples of eight bits). Otherwise, the device aborts the operation and no erase operation is performed.
- Despite the lower order address bits not being decoded by the device, the complete three address bytes must still be clocked into the device before the $\overline{CS}$ pin is deasserted. If the address is incomplete, the operation is aborted.
- If the memory is in the protected state, the Block Erase command cannot be executed and the device returns to the idle state once the $\overline{CS}$ pin has been deasserted.

### 6.6.8 Error Reporting

The device incorporates an intelligent erase algorithm that can detect when a byte location fails to erase properly. If an erase error occurs, it is indicated by the EE and PE bits in Status Register 4. See Section 5.6, Status Register 4 for more information.

## 6.7 Chip Erase (60h, C7h)

The entire memory array can be erased in a single operation by using the Chip Erase command. Two commands (60h and C7h) can be used for the Chip Erase command. There is no difference in device functionality when utilizing these two commands, so they can be used interchangeably.

### 6.7.1 Command Prerequisites

Before a Chip Erase command can be issued, the Write Enable (06h) command must have been previously issued to the device to set the WEL bit of Status Register 1. The Chip Erase command is not executed if any memory region is protected by either the block protect bits or the individual block locks. For more information on the block protect bits, see bits 4:2 (BP[2:0]) of Section 5.3. For more information on enabling individual lock bits, see bit 2 (WPS) of Section 5.5.

### 6.7.2 Transfer Format

The 60h/C7h commands follow the 1-0-0 transfer format described in Section 4.2, where the command is transferred on the SI pin. No address or data are transferred for this command. See Figure 3 for a timing diagram of this operation.

### 6.7.3 Transfer Sequence

To perform the Chip Erase operation, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The 60h/C7h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- When the $\overline{\text{CS}}$ pin is deasserted, the device begins to erase the entire memory array. The erasing of the device is internally self-timed and takes place in a time of $t_{CHPE}$.

See Table 21 for more information on the transfer sequence for this command.

### 6.7.4 Device Status

While the device is executing a successful erase cycle, the $\overline{\text{RDY}}$/BSY bit in Status Register 1 can be read and indicates the device is busy. For faster throughput, it is recommended that Status Register 1 be polled rather than waiting the $t_{CHPE}$ time to determine if the device has finished erasing. At some point before the erase cycle completes, the WEL bit in SR1 is reset back to the logical 0 state.

### 6.7.5 Active Status Interrupt

Alternatively, the Active Status Interrupt (25h) can be used to determine when the erase operation has completed. When this command is used, it is not necessary to continuously read the $\overline{\text{RDY}}$/BSY bit to determine when the command has completed. Instead, hardware drives the value of the $\overline{\text{RDY}}$/BSY bit onto the SO pin. The state of the pin is updated every clock cycle. The host can monitor the SO pin until it toggles from a logic 1 to a logic 0, indicating that the operation has completed. For more information, see Section 4.14, Active Status Interrupt.

### 6.7.6 Programming Restrictions

The Chip Erase command adheres to the following programming restrictions. The following events can cause the chip erase operation to be aborted. If either one occurs, the WEL bit in Status Register 1 is reset back to the logical 0 state.

- The $\overline{\text{CS}}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no erase operation is performed.
- If any block in the memory is in the protected state, the Chip Erase command cannot be executed and the device returns to the idle state once the $\overline{\text{CS}}$ pin has been deasserted.

### 6.7.7 Error Reporting

A Chip Erase command cannot be suspended by executing the Program/Erase Suspend (75h) command. Hardware ignores the Program/Erase Suspend command if it is issued during a Chip Erase.

- The device incorporates an intelligent erase algorithm that can detect when a byte location fails to erase properly. If an erase error occurs, it is indicated by the EE and PE bits in Status Register 4. For more information, see Section 5.6, Status Register 4.

## 6.8 Byte/Page Program (02h)

The Byte/Page Program command allows anywhere from a single byte of data to 256 bytes of data to be programmed into previously erased memory locations. An erased memory location is one that has all eight bits set to the logical 1 state (a byte value of FFh).

### 6.8.1 Command Prerequisites

Before a Byte/Page Program command can be issued, the Write Enable (06h) command must have been previously issued to the device to set the WEL bit of Status Register 1.

### 6.8.2 Transfer Format

The 02h command follows the 1-1-1 transfer format described in Section 4.2, where the command, address, and data are all transferred on the SI pin. See Figure 9 for a timing diagram of this transaction.

### 6.8.3 Transfer Sequence

To perform the Byte/Page Program operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 02h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to read within the memory array. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first.
- Each byte transfer requires eight clock cycles. The data is always input with the most significant bit of the byte transferred first. The number of bytes transferred is determined by software. The transfer can be anywhere from a single byte to the entire buffer. If less than 256 bytes of data were sent to the device, the remaining bytes within the page are not programmed and remain in the erased state (FFh).
- Once the $\overline{CS}$ pin is deasserted, hardware moves the data from the buffer to the memory. The programming of the data bytes is internally self-timed and takes place in a time of $t_{PP}$ or $t_{BP}$ if only programming a single byte.

See Table 21 for more information on the transfer sequence for this command.

### 6.8.4 Device Status

While the data is being transferred, bit 0 ($\overline{RDY}$/BSY) of Status Register 1 can be read and indicates that the device is busy. For faster throughput, it is recommended that Status Register 1 be polled rather than waiting the $t_{BP}$ or $t_{PP}$ time to determine if the data bytes have finished programming.

### 6.8.5 Active Status Interrupt

Alternatively, the Active Status Interrupt (25h) can be used to determine when the program operation has completed. When this command is used, it is not necessary to continuously read the $\overline{RDY}$/BSY bit to determine when the command has completed. Instead, hardware drives the value of the $\overline{RDY}$/BSY bit into the SO pin. The state of the pin is updated every clock cycle. The host can monitor the SO pin until it toggles from a logic 1 to a logic 0, indicating that the operation has completed. For more information, see Section 4.14, Active Status Interrupt.

### 6.8.6 Programming Restrictions

The Byte/Page Program command adheres to the following programming restrictions. If any of the following conditions occur, the programming cycle is aborted and the WEL bit in Status Register 1 is reset back to the logic 0 state.

**Deassertion of the $\overline{CS}$ Pin**
The $\overline{CS}$ pin must be deasserted on even byte boundaries (multiples of eight bits). Otherwise, the device aborts the operation and no data is programmed into the memory array.

**Protected Memory**
If the memory is in a protected state, the Byte/Page Program command is not executed, and the device returns to the idle state once the $\overline{CS}$ pin has been deasserted.

**Page Address Boundary**
If the starting memory address denoted by A[23:0] does not fall on an even 256-byte page boundary (A[7:0] are not all 0), then special circumstances regarding which memory locations to be programmed apply. In this situation, any data sent to the device that exceeds the page size wraps around back to the beginning of the same page.

For example, if the starting address denoted by A23-A0 is 0000FEh, and three bytes of data are sent to the device, then the first two bytes of data is programmed at addresses 0000FEh and 0000FFh while the last byte of data is programmed at address 000000h. The remaining bytes in the page (addresses 000001h through 0000FDh) are not programmed and remain in the current state. Also, if more than 256 bytes of data are sent to the device, then only the last 256 bytes sent are latched into the internal buffer.

**Incomplete Address or Data**
If the device receives either an incomplete address, or an incomplete data byte, the operation is aborted and hardware clears the WEL bit is Status Register 1.

### 6.8.7 Error Reporting

The device also incorporates an intelligent programming algorithm that can detect when a byte location fails to program properly. If a programming error arises, it is indicated by the PE bit in Status Register 4.

## 6.9 Sequential Program Mode (ADh/AFh)

The Sequential Program Mode improves throughput over the Byte/Page Program command when the Byte/Page Program command programs single bytes only into consecutive address locations. For example, some systems may be designed to program only a single byte of information at a time and cannot utilize a buffered Page Program operation due to design restrictions. In such a case, the system would normally have to perform multiple Byte Program operations in order to program data into sequential memory locations. This approach can add considerable system overhead and SPI bus traffic.

The Sequential Programming Mode helps reduce system overhead and bus traffic by incorporating an internal address counter that keeps track of the byte location to program; thus, it is not necessary to supply an address sequence to the device for every byte being programmed.

### 6.9.1 Command Prerequisites

Before a Sequential Program command can be issued, the Write Enable (06h) command must have been previously issued to the device to set the WEL bit of Status Register 1.

When using the Sequential Program mode, all address locations to be programmed must be in the erased state.

### 6.9.2 Transfer Format

The initial ADh/AFh command follows the 1-1-1 transfer format described in Section 4.2, where the command, address, and data are all transferred on the SI pin. See Figure 9 for a timing diagram of this transaction.

All subsequent ADh/AFh commands follow the 1-0-1 transfer format described in Section 4.2, where the command and data are both transferred on the SI pin. Subsequent accesses do not require an address. See Figure 6 for a timing diagram of this transaction.

### 6.9.3 Transfer Sequence — Initial Transfer

To perform the initial Sequential Program operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The AFh/ADh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to written to within the device (typically the memory array). A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first.
- Transfer the first byte of data. This requires eight clock cycles. The data is always input with the most significant bit of the byte transferred first.
- Deasserting the $\overline{CS}$ pin starts the internally self-timed program operation, and the byte of data is programmed into the memory location specified by the A23:A0.

See Table 21 for more information on the transfer sequence for this command.

### 6.9.4 Transfer Sequence — Subsequent Transfers

To perform subsequent Sequential Program operations, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The AFh/ADh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Because the initial transfer has already occurred and hardware knows already knows the location of the next byte to be programmed, the address field is not necessary. The command can be followed by the next data byte.
- Transfer the next byte of data. This requires eight clock cycles. The data is always input with the most significant bit of the byte transferred first.
- Deasserting the $\overline{CS}$ pin starts the internally self-timed program operation, and the byte of data is programmed into the memory location specified by the internal address counter. There is no need to reissue the Write Enable command once the Sequential Program Mode has been entered.
- When the last desired byte has been programmed into the memory array, the Sequential Program Mode operation can be terminated by reasserting the $\overline{CS}$ pin and sending the Write Disable command to the device to reset the WEL bit in Status Register 1.

### 6.9.5 Program Status

While the device is programming a byte, the $\overline{RDY}$/BSY bit in Status Register 1 can be read to determine if the device is busy. The programming of the data bytes is internally self-timed and takes place in a time of $t_{PP}$ (page) or $t_{BP}$ (single byte).

For faster throughput, it is recommended that the Status Register be polled at the end of each program cycle rather than waiting the $t_{BP}$ or $t_{PP}$ time to determine if the byte has finished programming before starting the next operation.

### 6.9.6 Commands Allowed and Not Allowed in Sequential Program Mode

When the device is busy executing a self-timed program/erase operation ($\overline{RDY}$/BSY bit in SR1 = 1), then it normally ignores most commands from the user. However, there are a handful of commands which are accepted by the device.

Similarly, if the device is not busy ($\overline{RDY}$/BSY bit in SR1 = 0), but is in Sequential Programming mode, only some commands are accepted, and others are rejected by the device.

Table 24 shows which commands can and cannot be executed with the device is in Sequential Program mode.

**Table 24. Command Behavior During Sequential Programming Mode**

| Command Name | Command Code(s) | During Sequential Programming Mode (RDY/BUSY = 0, SPM = 1) |
|---|---|---|
| Read Array | 03h, 3Bh, 6Bh, EBh, E7h, 0Bh | Not allowed |
| Buffer Read | D4h | Not allowed |
| Page/Block Erase | 81h, DBh, 20h, 52h, D8h | Not allowed |
| Chip Erase | 60h, C7h | Not allowed |
| Buffer Write | 84h | Not allowed |
| Byte/Page Program | 02h, A2h, 32h | Not allowed |
| Buffer to M-M P-P without Erase | 88h | Not allowed |
| Read-Modify-Write | 0Ah | Not allowed |
| Sequential Programming | ADh, AFh | Allowed |
| Program/Erase Suspend | B0h, 75h | Not Allowed |
| Program/Erase Resume | D0h, 7Ah | Not allowed |
| Write Enable | 06h | Allowed |
| Write Disable | 04h | Allowed |
| Volatile Write Enable | 50h | Not allowed |
| Individual Block Lock | 36h | Not allowed |
| Individual Block Unlock | 39h | Not allowed |
| Global Block Lock | 7Eh | Not allowed |
| Global Block Unlock | 98h | Not allowed |
| Read Block Lock Status | 3Ch, 3Dh | Not allowed |
| Program OTP Security Register | 9Bh | Not allowed |
| Read OTP Security Register | 4Bh | Not allowed |
| Read Status Registers (direct) | 05h, 35h, 15h | Allowed |
| Read Status Registers (indirect) | 65h | Allowed |
| Write Status Registers (direct) | 01h, 31h, 11h | Not allowed |
| Write Status Registers (indirect) | 71h | Not allowed |
| Status Register Lock | 6Fh | Not allowed |
| Active Interrupt | 25h | Allowed |
| Terminate | F0h | Allowed |
| Enable Reset | 66h | Allowed |
| Reset Device | 99h | Allowed |
| Read Mfg ID and Device ID | 9Fh, 90h, 94h | Allowed |
| Deep Power-Down | B9h | Not allowed |
| Resume from Deep Power-Down | ABh | Not allowed |
| Ultra-Deep Power-Down | 79h | Not allowed |
| Read SFDP | 5Ah | Not allowed |
| Set Burst with Wrap | 77h | Not allowed |
| Start Low Battery Detect | EFh | Not allowed |

### 6.9.7 Programming Restrictions

The Sequential Program mode adheres to the following programming restrictions.

**Multiple Data Bytes per Program Cycle**

If more than one byte of data is clocked in during a program cycle, then only the last byte of data sent on the SI pin is stored in the internal latches. The programming of each byte is internally self-timed and takes place in a time of $t_{BP}$.

**Deasserting the $\overline{CS}$ Pin**

For each program cycle, a complete byte of data must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on even byte boundaries (multiples of eight bits). Otherwise, the device aborts the operation, and the byte of data is not programmed into the memory array. Also, hardware resets the WEL bit in Status Register 1.

**Protected Memory**

If the address initially specified by A[23:0] points to a memory location within a block that is in the protected state, the Sequential Program Mode command is not executed, and the device returns to the idle state once the $\overline{CS}$ pin has been deasserted. The WEL bit in the Status Register is also reset back to the logical 0 state.

Sequential Program mode does not automatically skip over protected blocks. Therefore, once the highest unprotected memory location in a programming sequence has been programmed, the device automatically exits the Sequential Program mode and resets the WEL bit in Status Register 1.

For example, if block 1 was protected and block 0 was currently being programmed, once the last byte of block 0 was programmed, the Sequential Program mode would automatically end. To continue programming with block 2, the Sequential Program mode would have to be restarted by supplying the ADh or AFh command, the three address bytes, and the first byte of block 2 to program.

**Address Wrapping**

There is no address wrapping when using the Sequential Program Mode. Therefore, when the last byte (1FFFFFh) of the memory array has been programmed, the device automatically exits the Sequential Program mode and resets the WEL bit in Status Register 1.

**64 kB Block Accesses**

If during a Sequential Program operation, the address increments into a 64 kB block where an erase operation has been suspended, hardware exits Sequential Program mode.

**Clearing the WEL Bit in Status Register 1**

If the WEL bit in Status Register 1 is cleared at any time during a sequential programming operation, hardware exits Sequential Program mode.

### 6.9.8 Error Reporting

The device also incorporates an intelligent programming algorithm that can detect when a byte location fails to program properly. If a programming error arises, it is indicated by the EE and PE bits in Status Register 4.

## 6.10 Dual Output Byte/Page Program (A2h)

The Dual Output Byte/Page Program (A2h) command is similar to the standard Byte/Page Program command (02h) and can be used to program anywhere from a single byte of data up to 256 bytes of data into previously erased memory locations. Unlike the standard Byte/Page Program command, however, the Dual Output Byte/Page Program command allows two bits of data to be clocked into the device on every clock cycle rather than just one.

### 6.10.1 Command Prerequisites

Before the Dual Output Byte/Page Program command can be executed, the Write Enable command (06h) must have been previously issued to set the Write Enable Latch (WEL) bit in Status Register 1.

### 6.10.2 Transfer Format

The A2h command follows the 1-1-2 transfer format described in Section 4.3, where the command and address are transferred on the SI pin, and input data is transferred on the SI and SO pins. To facilitate this transfer, hardware switches the SO pin to an input as soon as the command has been decoded and the address transferred. See Figure 11 for a timing diagram of this transaction.

### 6.10.3 Transfer Sequence

To perform a Dual Output Page Program operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The A2h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to written. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first.
- The first byte of data is transferred on the SI and SO pins. This requires four clock cycles. The data is always input with the most significant bit on the SO pin during each clock as shown below. Like the standard Byte/Page Program command, all data clocked into the device is stored to an internal buffer.
  - First data clock: bit 7 is input on the SO pin
  - First data clock: bit 6 is input on the SI pin
  - Second data clock: bit 5 is input on the SO pin
  - Second data clock: bit 4 is input on the SI pin
  - Third data clock: bit 3 is input on the SO pin
  - Third data clock: bit 2 is input on the SI pin
  - Fourth data clock: bit 1 is input on the SO pin
  - Fourth data clock: bit 0 is input on the SI pin
  - Subsequent data bytes are transmitted in the same sequence as above

### 6.10.4 Program Status

While the device is programming a byte, Status Register 1 can be read to determine if the device is busy. The programming of the data bytes is internally self-timed and takes place in a time of $t_{PP}$ (page) or $t_{BP}$ (single byte).

For faster throughput, it is recommended that the Status Register be polled at the end of each program cycle rather than waiting the $t_{BP}$ or $t_{PP}$ time to determine if the byte has finished programming before starting the next operation.

### 6.10.5 Active Status Interrupt

Alternatively, the Active Status Interrupt (25h) can be used to determine when the program operation has completed. When this command is used, it is not necessary to continuously read the $\overline{RDY}$/BSY bit to determine when the command has completed. Instead, hardware drives the value of the $\overline{RDY}$/BSY bit into the SO pin. The state of the pin is updated every clock cycle. The host can monitor the SO pin until it toggles from a logic 1 to a logic 0, indicating that the operation has completed. For more information, see Section 4.14, Active Status Interrupt.

### 6.10.6 Programming Restrictions

The Dual Output Byte/Page Program operation adheres to the following programming restrictions.

**Page Address Boundary**

If the starting memory address denoted by A[23:0] does not fall on a 256-byte page boundary (A[7:0] are not all 0), special circumstances regarding which memory locations are to be programmed apply. In this situation, any data sent to the device that exceeds the end of the page wraps around to the beginning of the same page.

For example: If the starting address denoted by A[23:0] is 0000FEh and three bytes of data are sent to the device, then the first two bytes of data is programmed at addresses 0000FEh and 0000FFh, while the last byte of data is programmed at address 000000h. The remaining bytes in the page (addresses 000001h through 0000FDh) are not programmed and remain in the erased state (FFh).

**Data Transfers**

When the $\overline{\text{CS}}$ pin is deasserted, the device programs the data stored in the internal buffer into the appropriate memory array locations based on the starting address specified by A[23:0] and the number of data bytes sent to the device. If fewer than 256 bytes of data is sent to the device, then the remaining bytes within the page are not programmed and remain in the erased state (FFh). Conversely, if more than 256 bytes of data are sent to the device, then only the last 256 bytes sent are latched into the internal buffer.

**Deassertion of the $\overline{\text{CS}}$ Pin**

The three address bytes and at least one complete byte of data must be clocked into the device before the $\overline{\text{CS}}$ pin can be deasserted. Also, the $\overline{\text{CS}}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no data is programmed into the memory array.

**Protected Memory**

If the address specified by A[23:0] points to a memory location within a block that is in the protected state, then the Byte/Page Program command is not executed, and the device returns to the idle state once the $\overline{\text{CS}}$ pin has been deasserted. The WEL bit in the Status Register is reset back to the Logical 0 state if the program cycle is aborted

### 6.10.7 Error Reporting

The device also incorporates an intelligent programming algorithm that can detect when a byte location fails to program properly. If a programming error arises, it is indicated by the EE and PE bits in Status Register 4.

## 6.11 Quad Output Page Program (32h)

The Quad Output Page Program command allows between 1 to 256 bytes of data to be programmed at previously erased memory locations.

### 6.11.1 Command Prerequisites

Before a Sequential Program command can be issued, the Write Enable (06h) command must have been previously issued to the device to set the WEL bit of Status Register 1. For more information on the QE bit, see Section 5.3. Also, the Quad Enable bit (QE bit in Status Register 2) must be set. For more information on the QE bit, see Section 5.4.

### 6.11.2 Transfer Format

The 32h command follows the 1-1-4 transfer format described in Section 4.4, where the command and address are transferred on the SI pin, and data is transferred on the $\overline{\text{HOLD}}$, $\overline{\text{WP}}$, SO, and SI pins. See Figure 13 for a timing diagram of this operation.

### 6.11.3 Transfer Sequence

To perform the Quad Output Page Program operation, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The 32h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to read within the memory array. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first.
- Following transfer of the last address byte, hardware switches the SO pin to an input.
- Data is input on the $\overline{\text{HOLD}}$, $\overline{\text{WP}}$, SO, and SI pins. As a result, each byte transfer requires only two clock cycles to complete. The data is always input with the most significant bit of the byte transferred first. The number of bytes transferred is determined by software. The transfer can be anywhere from a single byte to the entire memory array. Each data byte is shifted into the device as follows:
  - First data clock: bit 7 is input on the $\overline{\text{HOLD}}$ pin
  - First data clock: bit 6 is input on the $\overline{\text{WP}}$ pin

- First data clock: bit 5 is input on the SO pin
- First data clock: bit 4 is input on the SI pin
- Second data clock: bit 3 is input on the $\overline{\text{HOLD}}$ pin
- Second data clock: bit 2 is input on the $\overline{\text{WP}}$ pin
- Second data clock: bit 1 is input on the SO pin
- Second data clock: bit 0 is input on the SI pin
- Additional data bytes are transferred on the four pins in the same manner as shown above.

### 6.11.4 Programming Restrictions

The Quad Output Byte/Page Program command adheres to the following programming restrictions

- The $\overline{\text{CS}}$ pin must be deasserted on a byte boundary (multiples of eight bits). Otherwise, the device aborts the operation and no action is taken.
- If the memory is in a protected state, then the Byte/Page Program command is not executed, and the device returns to the idle state once the $\overline{\text{CS}}$ pin has been deasserted.
- If the starting memory address denoted by A[23:0] does not fall on an even 256-byte page boundary (A[7:0] are not all 0), special circumstances regarding which memory locations to be programmed apply. In this situation, any data sent to the device that exceeds the page size wraps around back to the beginning of the same page.

  For example, if the starting address denoted by A23-A0 is 0000FEh, and three bytes of data are sent to the device, then the first two bytes of data is programmed at addresses 0000FEh and 0000FFh while the last byte of data is programmed at address 000000h. The remaining bytes in the page (addresses 000001h through 0000FDh) are not programmed and remain in the current state. Also, if more than 256 bytes of data are sent to the device, then only the last 256 bytes sent are latched into the internal buffer.
- If the device receives either an incomplete address, or an incomplete data byte, the operation is aborted and hardware clears the WEL bit is Status Register 1.

### 6.11.5 Error Reporting

The device also incorporates an intelligent programming algorithm that can detect when a byte location fails to program properly. If a programming error arises, it is indicated by the PE bit in Status Register 4.

## 6.12 Program/Erase Suspend (75h/B0h)

The device supports two Program/Erase Suspend commands, 75h and B0h, that perform the exact same function. The 75h command can be used for legacy software, and the B0h command is used for compatibility with future implementations.

In some code plus data storage applications, it is often necessary to process certain high-level system interrupts that require relatively immediate reading of code or data from the Flash memory. In such an instance, it might not be possible for the system to wait the microseconds or milliseconds required for the Flash memory to complete a program or erase cycle. The Program/Erase Suspend command allows a program or erase operation in progress to be suspended so that other device operations can be performed. For example, by suspending an erase operation to a particular block, the system can perform functions such as a program or read to a different block.

### 6.12.1 Transfer Format

The 75h and B0h commands follow the 1-0-0 transfer format described in Section 4.2, where the command is transferred on the SI pin. No address or data are transferred for this command. See Figure 3 for a timing diagram of this operation.

### 6.12.2 Transfer Sequence

To perform the Program Erase/Suspend operation, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The 75h or B0h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- When the $\overline{CS}$ pin is deasserted, the device suspends the program.

See Table 21 for more information on the transfer sequence for this command.

### 6.12.3 Command Behavior During a Program/Erase Operation

If an attempt is made to perform an operation that is not allowed during a program or erase suspend, such as a Write Status Register operation, then the device simply ignores the command and no operation is performed. The state of the WEL bit in Status Register 1 is not affected. Note that in the table below, the RDY/BSY bit is located in Status Register 1. The PS and ES bits are located in Status Register 5.

**Table 25. Command Behavior During Program/Erase or Program/Erase Suspend Operations**

| Command | Command Code(s) | During Program or Erase (RDY/BSY = 1, PS = x, ES = x) | During Program Suspend (RDY/BSY = 0, PS = 1, ES = x) | During Erase Suspend (RDY/BSY = 0, PS = 0, ES = 1) |
|---|---|---|---|---|
| Read Array | 03h, 0Bh, 3Bh, 6Bh, EBh, E7h | Not allowed | Allowed | Allowed |
| Buffer Read | D4h | Not allowed | Allowed | Allowed |
| Page/Block Erase | 81h, DBh, 20h, 52h, D8h | Not allowed | Not allowed | Not allowed |
| Chip Erase | 60h, C7h | Not allowed | Not allowed | Not allowed |
| Buffer Write | 84h | Not allowed | Not allowed | Allowed |
| Byte/Page Program | 02h, A2h, 32h | Not allowed | Not allowed | Allowed[1] |
| Buffer to M-M P-P without Erase | 88h | Not allowed | Not allowed | Allowed(1) |
| Read-Modify-Write | 58h/0Ah | Not allowed | Not allowed | Not allowed |
| Sequential Programming | ADh, AFh | Not allowed | Not allowed | Allowed(1) |
| Program/Erase Suspend | B0h, 75h | Allowed | Not allowed | Not allowed |
| Program/Erase Resume | D0h, 7Ah | Not allowed | Allowed | Allowed |
| Write Enable | 06h | Not allowed | Allowed | Allowed |
| Write Disable | 04h | Not allowed | Allowed | Allowed |
| Volatile Write Enable | 50h | Not allowed | Allowed | Allowed |
| Individual Block Lock | 36h | Not allowed | Not allowed | Not allowed |
| Individual Block Unlock | 39h | Not allowed | Not allowed | Not allowed |
| Global Block Lock | 7Eh | Not allowed | Not allowed | Not allowed |
| Global Block Unlock | 98h | Not allowed | Not allowed | Not allowed |
| Read Block Lock Status | 3Ch, 3Dh | Not allowed | Allowed | Allowed |
| Program OTP Security Register | 9Bh | Not allowed | Not allowed | Not allowed |
| Read OTP Security Register | 4Bh | Not allowed | Allowed | Allowed |

**Table 25. Command Behavior During Program/Erase or Program/Erase Suspend Operations** (Continued)

| Command | Command Code(s) | During Program or Erase (RDY/BSY = 1, PS = x, ES = x) | During Program Suspend (RDY/BSY = 0, PS = 1, ES = x) | During Erase Suspend (RDY/BSY = 0, PS = 0, ES = 1) |
|---|---|---|---|---|
| Read Status Registers (direct) | 05h, 35h, 15h | Allowed | Allowed | Allowed |
| Read Status Registers (indirect) | 65h | Allowed | Allowed | Allowed |
| Write Status Registers (direct) | 01h, 31h, 11h | Not allowed | Not allowed | Not allowed |
| Write Status Registers (indirect) | 71h | Not allowed | Not allowed | Not allowed |
| Status Register Lock | 6Fh | Not allowed | Not allowed | Not allowed |
| Active Interrupt | 25h | Allowed | Allowed | Allowed |
| Terminate | F0h | Allowed | Allowed | Allowed |
| Enable Reset | 66h | Allowed | Allowed | Allowed |
| Reset Device | 99h | Allowed | Allowed | Allowed |
| Read Mfg ID and Device ID | 9Fh, 90h, 94h | Allowed | Allowed | Allowed |
| Deep Power-Down | B9h | Not allowed | Not allowed | Not allowed |
| Resume from Deep Power-Down | ABh | Allowed | Allowed | Allowed |
| Ultra-Deep Power-Down | 79h | Not allowed | Not allowed | Not allowed |
| Read SFDP | 5Ah | Not allowed | Allowed | Allowed |
| Set Burst with Wrap | 77h | Not allowed | Allowed | Allowed |
| Start Low Battery Detect | EFh | Not allowed | Allowed | Allowed |

1. The operation is allowed in a different 64 kB block.

### 6.12.4 Device Status

When the $\overline{\text{CS}}$ pin is deasserted, the program or erase operation currently in progress suspends within a time of $t_{SUSE}$. Hardware sets the Suspend (SUSP) bit in Status Register 2 and the ES or PS bits in Status Register 5 to indicate that the program or erase operation has been suspended. Also, hardware clears the $\overline{\text{RDY}}$/BSY bit in Status Register 1 to indicate that the device is ready for another operation.

### 6.12.5 Programming Restrictions

The Erase/Program Suspend command adheres to the following programming restrictions.

The following commands cannot be suspended:

- Write Status Register 1 (01h)
- Write Status Register 2 (31h)
- Write Status Register 3 (11h)
- Write Status Registers (71h)
- Status Register Lock (6Fh)
- Program OTP Security Registers (9Bh)
- Chip Erase (60h/C7h)
- Byte Write (R-M-W 0Ah)
- Sequential Programming (AFh/ADh)

Hardware ignores the Program/Suspend command if it is issued while these commands are in progress.

**Erase Suspend and Program Suspend Ordering**

A program operation can be performed while an erase operation is suspended. However, that operation must be completed before the erase operation can be resumed. Also, an erase operation cannot be performed while a program operation is suspended. Other device operations, such as a Read Status Register, can also be performed while a program or erase operation is suspended.

**Write Enable Latch Ignored**

Since the need to suspend a program or erase operation is immediate, the Write Enable command does not need to be issued prior to the Program/Erase Suspend command being issued. Therefore, the Program/Erase Suspend command operates independently of the state of the WEL bit in the Status Register.

**Deasserting the $\overline{\text{CS}}$ Pin**

The complete command must be clocked into the device before the $\overline{\text{CS}}$ pin is deasserted. Also, the $\overline{\text{CS}}$ pin must be deasserted on a byte boundary (multiples of eight bits). Otherwise, no suspend operation is performed.

**Accessing a Suspended Area**

If a read operation is attempted to a suspended area (page for programming, or block for erasing), the device outputs undefined data.

**Note:** See application note AN500 for operational guidance on implementing suspend and resume operations.

## 6.13 Program/Erase Resume (7Ah/D0h)

The device supports two Program/Erase Resume commands, 7Ah and D0h, that perform the exact same function. The 7Ah command can be used for legacy software, and the D0h command is used for compatibility with future implementations.

The Program/Erase Resume command allows a suspended program or erase operation to be resumed and continue programming a Flash page or erasing a Flash memory block where it left off.

### 6.13.1 Command Prerequisites

Hardware accepts the Program/Erase Resume command only if the ES bit (set when an erase operation is suspended) or PS bit (set when a program operation is suspended) in Status Register 5 is set and the $\overline{\text{RDY}}$/BSY bit in Status Register 1 is cleared. If the ES/PS bit is cleared or the $\overline{\text{RDY}}$/BSY bit is set, the Program/Erase Resume command is ignored by the device. See Section 5.3 and Section 5.7.

Note that the Write Enable command does not need to be issued prior to the Program/Erase Resume command being issued. Therefore, the Program/Erase Resume command operates independently of the state of the WEL bit in Status Register 1.

### 6.13.2 Transfer Format

The 7Ah and D0h commands follow the 1-0-0 transfer format described in Section 4.2, where the command is transferred on the SI pin. No address or data are transferred for this command. See Figure 3 for a timing diagram of this operation.

### 6.13.3 Transfer Sequence

To perform the Program/Erase Resume operation, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The 7Ah or D0h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first. No address of data are required for this command.
- When the $\overline{\text{CS}}$ pin is deasserted, the device resumes the program.

See Table 21 for more information on the transfer sequence for this command.

### 6.13.4 Command Behavior

When the command is executed, hardware performs the following:

- Hardware clears either the PS or the ES bits in Status Register 5 depending on whether a program or erase operation has been suspended. If both of these bits are cleared, hardware clears the SUSP bit in Status Register 2.
- Hardware sets the $\overline{\text{RDY}}$/BSY bit in Status Register 1 to indicate that the device is busy.
- When the $\overline{\text{CS}}$ pin is deasserted, the program or erase operation currently suspended resumes within a time of $t_{RES}$.

### 6.13.5 Programming Restrictions

The Program/Erase Resume command adheres to the following programming restrictions.

- The complete command must be clocked into the device before the $\overline{\text{CS}}$ pin is deasserted.
- The $\overline{\text{CS}}$ pin must be deasserted on a byte boundary (multiples of eight bits). Otherwise, no resume operation is performed.
- While the device is busy resuming a program or erase operation, any attempts at issuing the Program/Erase Suspend command are ignored. Therefore, if a resumed program or erase operation needs to be subsequently suspended again, the system must either wait the entire $t_{RES}$ time before issuing the Program/Erase Suspend command, or it can check the status of the $\overline{\text{RDY}}$/BSY bit in Status Register 1, or the ES/PS bits in Status Register 5 to determine if the previously suspended program or erase operation has resumed.
- During a simultaneous Erase Suspend/Program Suspend condition, issuing the Program/Erase Resume command results in the program operation resuming first. After the program operation has been completed, the Program/Erase Resume command must be issued again in order for the erase operation to be resumed. For more information, see Section 4.10.1, Nested Operations.

## 6.14 Set Burst Wrap (77h)

The Set Burst with Wrap (77h) command is used in conjunction with the EBh and E7h commands to support a cache line fill, regardless of the starting address. This type of operation, known as address wrapping, is an MCU-friendly feature that allows the Microcontroller Unit (MCU) cache controller to fill a cache line in one operation starting from a specific address (known as the critical byte) within the cache line, proceeding to the end of the line, then wrapping around the start of the cache line to complete the fill.

For example, if the wrap size is 16 bytes and the starting address is 0x1004, then the read sequence is as follows: 0x1004, 0x1005, 0x1006, 0x1007, 0x1008, 0x1009, 0x100A, 0x100B, 0x100C, 0x100D, 0x100E, 0x100F, 0x1000, 0x1001, 0x1002, 0x1003. As shown in this sequence, the fill starts at address 0x1004, proceeds to the end of the cache line (0x100F), then wraps around to 0x1000 to complete the fill.

The wrap feature can improve code execution performance in the MCU system, as the MCU first receives the command or data it requires at that instant, and then fetches the remainder of the cache line without requiring additional commands or addresses to be sent.

The 77h command both enables/disables the wrap-around feature, and determines the size of the wrap. The 77h command sets the state of bits W6, W5, and W4 as shown in Table 26.

### 6.14.1 Transfer Format

The 77h command follows the 1-4-4 transfer format described in Section 4.6, where the command is transferred on the SI pin, and the address and data are transferred on the $\overline{\text{HOLD}}$, $\overline{\text{WP}}$, SO, and SI pins. See Figure 17 for a timing diagram of this command.

### 6.14.2 Transfer Sequence

To perform the Set Burst with Wrap operation, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The 77h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in. A total of 6 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first. Even though the address must be clocked in, the address is treated as a don't care field internally.
- Following transfer of the last address byte, eight Wrap bits are transferred which indicate the length of the wrap operation as described in the table below.

### 6.14.3 Wrap Bits

After the last address byte is transferred, the device transfers eight wrap bits (W7-0), which indicate the wrap length. Note that only wrap bits W6:W4 are used during this operation. The W7 and W3:W0 bits are not used. This is shown in the following table.

**Table 26. Encoding of Burst Wrap Bits**

| W6 | W5 | W4 = 0 | | W4 = 1 | |
|---|---|---|---|---|---|
| | | Wrap Size | Wrap Around | Wrap Length | Wrap Around |
| 0 | 0 | 8-byte | Yes | N/A | No |
| 0 | 1 | 16-byte | Yes | N/A | No |
| 1 | 0 | 32-byte | Yes | N/A | No |
| 1 | 1 | 64-byte | Yes | N/A | No |

Once the 77h command is executed and the W6:W4 bits are set, the subsequent EBh or E7h command uses these bits to determine what section size to access within a given page. As shown in the table, returning to the normal read operation requires the execution of another 77h command with the W4 bit driven high (W4 = 1).

### 6.14.4 Programming Restrictions

The Set Burst with Wrap command adheres to the following programming restrictions.

- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiples of eight bits). Otherwise, the device aborts the operation and no action is taken.

## 6.15 Buffer Read (D4h)

The SRAM data buffers can be accessed independently from the main memory array using the Buffer Read command. This command allows data to be sequentially read directly from the buffer.

### 6.15.1 Transfer Format

The D4h command follows the 1-1-1 transfer format described in Section 4.2, where the command and address are transferred on the SI pin, and data is transferred on the SO pin. Figure 8 shows a buffer read operation with one dummy byte.

### 6.15.2 Transfer Sequence

To perform the Buffer Read operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The D4 command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to read within the buffer. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first. Since the buffer is only 256 bytes in size, the upper 16 bits of address (A[23:8) are don't care for this operation. Only the lower byte (A[7:0]) are valid and are used to select one of the 256 buffer entries.
- Following the three address bytes, an additional dummy byte must be clocked into the device.
- Data is output on the SO pin. Each byte transfer requires eight clock cycles. The data is always output with the most significant bit of the byte transferred first. The number of bytes transferred is determined by software. The transfer can be anywhere from a single byte to the entire 256 byte buffer.

- When the end of a buffer is reached, the device continues reading back at the beginning of the buffer.
- Deasserting the $\overline{CS}$ pin terminates the read operation and puts the SO pin into high-impedance state. The $\overline{CS}$ pin can be deasserted at any time and does not require a full byte of data be read.

## 6.16 Buffer Write (84h)

Utilizing the Buffer Write command allows data to be written directly into the internal buffer without starting a write to the Flash array at the same time. This write operation is typically used to write small amounts of data at a time to the buffer.

To prevent buffer data loss, do not perform a program or erase operation until the written data is being used, either through the *Buffer Read* command or *Buffer to Main Memory Program without Erase* command.

### 6.16.1 Command Prerequisites

The Write Enable command (06h) must have been previously issued to the device to set the Write Enable Latch (WEL) bit in Status Register 1.

### 6.16.2 Transfer Format

The 84h command follows the 1-1-1 transfer format described in Section 4.2, where the command, address, and data are transferred on the SI pin. Figure 9 shows a buffer write operation with a one byte transfer. Additional data bytes can be transferred as long as the $\overline{CS}$ pin remains low. Note that $\overline{CS}$ can only be deasserted on a byte boundary.

### 6.16.3 Transfer Sequence

To perform the Buffer Write operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 84 command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to written within the buffer. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first. Since the buffer is only 256 bytes in size, the upper 16 bits of address (A[23:8]) are don’t care for this operation. Only the lower byte (A[7:0]) is valid and selects one of the 256 buffer entries.
- Data is input on the SI pin. Each byte transfer requires eight clock cycles. The data is always input with the most significant bit of the byte transferred first. The number of bytes transferred is determined by software. The transfer can be anywhere from a single byte to the entire 256 byte buffer.
- After the last address byte has been clocked into the device, data can then be clocked in on subsequent clock cycles. If the end of the buffer is reached, the device wraps around back to the beginning of the buffer. Data continues to be loaded into the buffer until a low-to-high transition is detected on the $\overline{CS}$ pin.

### 6.16.4 Writing Buffer Entries

Hardware does not clear the buffer entries automatically prior to starting a Buffer Write command. This allows multiple Buffer Write commands to be performed to update different parts of the buffer before writing the buffer to the Flash memory, or to update the same buffer locations multiple times before writing to Flash memory. Buffer locations which are not updated can contain whatever data is left in the buffer from the previous write operation.

### 6.16.5 Writing the Buffer Entries to Flash Memory

Once the Buffer Write operation is completed, the Buffer to Main Memory Page Program without Built-In Erase (88h) command writes the full 256-byte page, even if less than 256 bytes were written to the buffer. For more information, see Section 6.17, Buffer to Main Memory Page Program without Erase (88h).

### 6.16.6 Programming Restrictions

The Buffer Write command adheres to the following programming restrictions.

- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no action is taken.

## 6.17 Buffer to Main Memory Page Program without Erase (88h)

The Buffer to Main Memory Page Program without Built-In Erase command allows data that is stored in the internal buffer to be written into a pre-erased page in the main memory array. The data in the buffer was previously written using the Buffer Write (84h) command. The 88h command writes the full 256-byte page, even if less than 256 bytes were written to the buffer.

### 6.17.1 Command Prerequisites

The page in main memory to be programmed must have been previously erased using one of the erase commands in order to avoid programming errors. The programming of the page is internally self-timed and takes place in a maximum time of $t_{PP}$.

The Write Enable command (06h) must have been previously issued to the device to set the Write Enable Latch (WEL) bit in Status Register 1.

### 6.17.2 Transfer Format

The 88h command follows the 1-1-0 transfer format described in Section 4.2, where the command and address are transferred on the SI pin. No data is required for this operation. Figure 4 shows a timing diagram for this type of transfer.

### 6.17.3 Transfer Sequence

To perform the Buffer Write operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 88h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to written within the buffer. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first.
  Since an entire 256 byte buffer entry is being written to the memory, the lower 8 bits of address (A[7:0]) are don't care for this operation, ensuring that the transfer starts at the beginning of a page boundary. Only the upper two bytes (A[23:8]) are valid and are used to select the page in memory.
- When a low-to-high transition occurs on the $\overline{CS}$ pin, the device programs the data stored in the buffer into the specified page in the main memory.

### 6.17.4 Device Status

During the transfer to memory, the $\overline{RDY}$/BSY bit in Status Register 1 indicates that the device is busy. Hardware clears this bit to indicate that the operation has completed and the device is ready for another operation. At some point before the program cycle completes, hardware resets the WEL bit in the Status Register back to the logic 0 state.

### 6.17.5 Protected Memory

Note that if the memory is in the protected state, the Buffer to Main Memory Page Program without Built-In Erase Program command is not executed, and the device returns to the idle state once the $\overline{CS}$ pin has been deasserted. Hardware resets the WEL bit in the Status Register back to the logical 0 state if either the program cycle aborts due to an incomplete address being sent, or because the memory location to be programmed is protected.

### 6.17.6 Programming Restrictions

The Buffer Main Memory Page Program without Erase command adheres to the following programming restrictions.

Protected Memory

Note that if the memory is in the protected state, the Buffer to Main Memory Page Program without Built-In Erase Program command is not executed, and the device returns to the idle state once the CS pin has been deasserted.

Incomplete Address

Even though the lower 8 bits of address are not used for this command, all 24 bits of address must be transferred to the device. If an incomplete address error occurs, the program cycle aborts and hardware resets the WEL bit in Status Register 1.

Deasserting the CS Pin

The CS pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no action is taken.

### 6.17.7 Error Reporting

The device incorporates an intelligent programming algorithm that can detect when a byte location fails to program properly. If a programming error arises, hardware sets the PE bit in Status Register 4 to indicate that a program error occurred.

## 6.18 Write Enable (06h)

The Write Enable command sets the Write Enable Latch (WEL) bit in Status Register 1. The WEL bit must be set for any of the following commands to be executed:

- Byte/Page Program (02h)
- Dual Output Byte/Page Program (A2h)
- Quad Output Byte/Page Program (32h)
- Sequential Program Mode (AFh/ADh)
- Buffer Main Memory Page Program without Erase (88h)
- Any Erase command (81h, DBh, 20h, 52h, D8h, 60h/C7h)
- Program OTP Security Register (9Bh)
- Write Status Register (01h, 31h, 11h, 71h)
- Read-Modify-Write (0Ah)

This makes the issuance of the above commands a two step process, thereby reducing the chances of a command being accidentally or erroneously executed. If the WEL bit in the Status Register is not set prior to the issuance of one of these commands, that command is not executed.

### 6.18.1 Transfer Format

The 06h command follows the 1-0-0 transfer format described in Section 4.2, where the command is transferred on the SI pin. No address or data are transferred for this command. See Figure 3 for a timing diagram of this operation.

### 6.18.2 Transfer Sequence

To perform the Write Enable operation, the CS pin is asserted and the information transferred as follows:

- The 06h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first. No address or data are required for this command.
- When the CS pin is deasserted, hardware sets the WEL bit in Status Register 1.

See Table 21 for more information on the transfer sequence for this command.

### 6.18.3 Programming Restrictions

The Write Enable command adheres to the following programming restrictions.

- The complete command must be clocked into the device before the CS pin is deasserted.

- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no action is taken.

## 6.19 Write Disable (04h)

The Write Disable command clears the Write Enable Latch (WEL) bit in Status Register 1. When the WEL bit is cleared, the following commands cannot be executed:

- Byte/Page Program (02h)
- Dual Output Byte/Page Program (A2h)
- Quad Output Byte/Page Program (32h)
- Sequential Program Mode (AFh/ADh)
- Buffer Main Memory Page Program without Erase (88h)
- Any Erase command (81h, DBh, 20h, 52h, D8h, 60h/C7h)
- Program OTP Security Register (9Bh)
- Write Status Register (01h, 31h, 11h, 71h)
- Read-Modify-Write (0Ah)

Other conditions can also cause the WEL bit to be cleared. For more information, see Table 13.

### 6.19.1 Transfer Format

The 04h command follows the 1-0-0 transfer format described in Section 4.2, where the command is transferred on the SI pin. No address or data are transferred for this command. See Figure 3 for a timing diagram of this operation.

### 6.19.2 Transfer Sequence

To perform the Write Disable operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 04h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first. No address or data are required for this command.
- When the $\overline{CS}$ pin is deasserted, hardware clears the WEL bit in Status Register 1.

See Table 21 for more information on the transfer sequence for this command.

### 6.19.3 Programming Restrictions

The Write Disable command adheres to the following programming restrictions.

- The complete command must be clocked into the device before the $\overline{CS}$ pin is deasserted.
- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no action is taken.

## 6.20 Volatile Status Register Write Enable (50h)

To write the volatile version of the Status Register bits, the Write Enable for Volatile Status Register (50h) command must be issued prior to each Write Status Register (01h) command. The Write Enable for the Volatile Status Register command does not set the Write Enable Latch (WEL) bit in Status Register 1. It is only valid for the next following Write Status Register command, to change the volatile Status Register bit values.

### 6.20.1 Transfer Format

The 50h command follows the 1-0-0 transfer format described in Section 4.2, where the command is transferred on the SI pin. No address or data are transferred for this command. See Figure 3 for a timing diagram of this operation.

### 6.20.2 Transfer Sequence

To perform the Write Disable operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 50h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first. No address or data are required for this command.
- When the $\overline{CS}$ pin is deasserted, hardware performs the operation.

See Table 21 for more information on the transfer sequence for this command.

### 6.20.3 Programming Restrictions

The Volatile Status Register Write Enable command adheres to the following programming restrictions.

- The complete command must be clocked into the device before the $\overline{CS}$ pin is deasserted.
- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no action is taken.

## 6.21 Individual Block Lock (36h)

The Individual Block Lock command can be used to protect individual pages in the memory array from being programmed or erased through one of the Program or Erase commands. To enable the individual block lock function, the WPS bit in Status Register 3 must be set. If the WPS bit is cleared, write protection is then determined by the combination of the CMPRT, BPSIZE, TB, and BP[2:0] bits in the Status Register 1. The default value for each individual lock bit is 1 at power-up or reset, so the entire memory array is protected.

### 6.21.1 Command Prerequisites

The Write Enable (06h) command must be executed before the device can accept the Individual Block Lock command. This is required to set the WEL bit in Status Register 1.

### 6.21.2 Transfer Format

The 36h command follows the 1-1-0 transfer format described in Section 4.2, where the command and address are transferred on the SI pin. The address represents the individual block to be locked. Hence no data is transferred for this command.

### 6.21.3 Transfer Sequence

To perform the Individual Block Lock operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 36h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the address location to be locked within the memory array. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first.
- When the $\overline{CS}$ pin is deasserted, hardware performs the operation and clears the WEL bit in Status Register 1.

See Table 21 for more information on the transfer sequence for this command.

#### 6.21.4 Programming Restrictions

The Individual Block Lock command adheres to the following programming restrictions.

- The complete command must be clocked into the device before the $\overline{CS}$ pin is deasserted.
- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no action is taken.

### 6.22 Individual Block Unlock (39h)

The Individual Block Unlock command can be used to unlock individual pages in the memory array to allow them to be programmed or erased through one of the Program or Erase commands. To enable the individual block unlock function, the WPS bit in Status Register 3 must be set. If WPS is cleared, write protection is then determined by the combination of the CMP, SEC, TB, and BP[2:0] bits in Status Register 1. The default value for each individual lock bit is 1 at power-up or reset, so the entire memory array is being protected.

#### 6.22.1 Command Prerequisites

The Write Enable (06h) command must be executed before the device can accept the Individual Block Unlock command. This is required to set the WEL bit in Status Register 1.

#### 6.22.2 Transfer Format

The 39h command follows the 1-1-0 transfer format described in Section 4.2, where the command and address are transferred on the SI pin. The address represents the individual block to be unlocked. Thus, no data is transferred for this command. See Figure 4 for a timing diagram of this transaction.

#### 6.22.3 Transfer Sequence

To perform the Individual Block Unlock operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 39h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the address location to be unlocked within the memory array. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first.
- When the $\overline{CS}$ pin is deasserted, hardware performs the operation and clears the WEL bit in Status Register 1.

See Table 21 for more information on the transfer sequence for this command.

#### 6.22.4 Programming Restrictions

The Individual Block Unlock command adheres to the following programming restrictions.

- The complete command must be clocked into the device before the $\overline{CS}$ pin is deasserted.
- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no action is taken.

## 6.23 Read Block Lock (3Ch/3Dh)

The Read Block Lock commands allow the user to read the status of the lock bits in each block. The individual block locks are used to protect any individual block. The default value for all individual block lock bits is 1 upon device power on or after reset.

The 3Ch and 3Dh command perform the exact same operation. These two commands are identical and are provided for backward compatibility purposes.

### 6.23.1 Command Prerequisites

The WPS bit in Status Register 3 must be set. When WPS is set, software uses the Block Lock (36h) and Block Unlock (39h) commands to lock and unlock blocks of memory. If the WPS bit is cleared, write protection is determined by the combination of CMP, SEC, TB, BP[2:0] bits in the Status Registers. The Read Block Lock command must not be used to determine the protection status of any region of the memory. See Section 4.8.1 for more information on the protection scheme.

### 6.23.2 Transfer Format

The 3Ch and 3Dh commands follow the 1-1-1 transfer format described in Section 4.2, where the command and address are transferred on the SI pin, and data is transferred on the SO pin. Figure 7 shows a timing diagram for this transaction.

### 6.23.3 Transfer Sequence

To perform the Read Block Lock operation, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The 3Ch or 3Dh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to read within the buffer. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first.
- Data is output on the SO pin and requires eight clock cycles. The MSB of the lock value is driven onto the SO pin first, and the LSB is driven out last. If the LSB is 1, the corresponding block is locked and no erase or program operation can be executed to that block. If the LSB is 0, the block is unlocked, indicating that program/erase operations are allowed. The remaining bits 7:1 of this value are undefined.

  If $\overline{\text{CS}}$ is kept low and additional data is clocked out, the device simply repeats the same byte over and over again until $\overline{\text{CS}}$ goes high. Out of the 8 bits in this byte, the LSB (bit 0) has the lock bit information, the rest of the are undefined. The lock bit information corresponds to the region of memory that encapsulates the 20-bit address provided by the user. So for example:
  - If the user provides an address of 0x000000, the device returns the lock status of the 4 kB region from 0x000000 - 0x000FFF.
  - If the user provides an address of 0x001234, the device returns the lock status of the 4 kB region 0x001000 - 0x001FFF.

### 6.23.4 Programming Restrictions

The Read Block Lock command adheres to the following programming restriction: The complete command must be clocked into the device before the $\overline{\text{CS}}$ pin is deasserted.

## 6.24 Global Block Lock (7Eh)

The Global Block Lock command sets all of the block bits to 1 with one operation.

### 6.24.1 Command Prerequisites

The Write Enable command (06h) must be executed to set WEL bit in Status Register 1 before the Global Block Lock command can be executed. See Section 6.18, Write Enable (06h) for more information.

### 6.24.2 Transfer Format

The 7Eh command follows the 1-0-0 transfer format described in Section 4.2, where the command is transferred on the SI pin. No address or data are transferred for this command. See Figure 3 for a timing diagram of this operation.

### 6.24.3 Transfer Sequence

To perform the Global Block Lock operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 7Eh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first. No address or data are required for this command.
- When the $\overline{CS}$ pin is deasserted, hardware performs the operation and clears the WEL bit in Status Register 1.

See Table 21 for more information on the transfer sequence for this command.

### 6.24.4 Programming Restrictions

The Global Block Lock command adheres to the following programming restrictions.

- The complete command must be clocked into the device before the $\overline{CS}$ pin is deasserted.
- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no action is taken.

## 6.25 Global Block Unlock (98h)

The Global Block UnLock command resets all of the block bits to 0 with one operation.

### 6.25.1 Command Prerequisites

The Write Enable command (06h) must be executed to set WEL bit in Status Register 1 before the Global Block Unlock command can be executed. See Section 6.18, Write Enable (06h) for more information.

### 6.25.2 Transfer Format

The 98h command follows the 1-0-0 transfer format described in Section 4.2, where the command is transferred on the SI pin. No address or data are transferred for this command. See Figure 3 for a timing diagram of this operation.

### 6.25.3 Transfer Sequence

To perform the Global Block Unlock operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 98h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first. Since all blocks are unlocked with this command, no address or data are required.
- When the $\overline{CS}$ pin is deasserted, hardware performs the operation and clears the WEL bit in Status Register 1.

See Table 21 for more information on the transfer sequence for this command

### 6.25.4 Programming Restrictions

The Global Block Unlock command adheres to the following programming restrictions.

- The complete command must be clocked into the device before the $\overline{CS}$ pin is deasserted.

- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no action is taken.

## 6.26 Program Security Register (9Bh)

There are four specialized 128-byte OTP (One-Time Programmable) Security Registers that can be used for purposes such as unique device serialization and locked key storage.

### 6.26.1 Command Prerequisites

Before the Program OTP Security Register command can be issued, the Write Enable command (06h) must be executed to set WEL bit in Status Register 1. See Section 6.18, Write Enable (06h) for more information.

### 6.26.2 OTP Security Register Layout

The OTP Security Registers are independent of the main Flash memory array. The four registers are organized as follows:

| Security Register Byte Number | | | | | | |
|---|---|---|---|---|---|---|
| 127 | 126 | 125 | ..... | 2 | 1 | 0 |
| Factory Programmed by Renesas Electronics | | | | | | |

| Security Register Byte Number | | | | | | |
|---|---|---|---|---|---|---|
| 255 | 254 | 253 | ..... | 130 | 129 | 128 |
| User Security OTP Register | | | | | | |

| Security Register Byte Number | | | | | | |
|---|---|---|---|---|---|---|
| 383 | 382 | 381 | ..... | 258 | 257 | 256 |
| User Security OTP Register | | | | | | |

| Security Register Byte Number | | | | | | |
|---|---|---|---|---|---|---|
| 511 | 510 | 509 | ..... | 386 | 385 | 384 |
| User Security OTP Register | | | | | | |

### 6.26.3 Transfer Format

The 9Bh command follows the 1-1-1 transfer format described in Section 4.2, where the command and address are transferred on the SI pin, and data is transferred on the SO pin. See Figure 9 for timing diagram of this transfer.

### 6.26.4 Transfer Sequence

To perform the Program OTP Register operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 9Bh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to read within the memory array. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first. For this command, address bits 23:9 are don’t care. Only bits 8:0 are used to address the appropriate Security register. See Table 27 for an encoding of the address.

- Data is input on the SI pin. Each byte transfer requires eight clock cycles. The data is always input with the most significant bit of the byte transferred first. The number of bytes transferred is determined by software. The transfer can be anywhere from a single byte to the entire OTP register array. Additional data bytes are transferred every eight clocks as long as the $\overline{CS}$ pin is asserted. The programming of the data bytes is internally self-timed and takes place in a time of $t_{OTPP}$ if the entire 128-byte register is programmed at once.
- When the $\overline{CS}$ pin is deasserted, the device takes the data stored in the internal buffer and programs it into the appropriate OTP Security Register locations based on the starting address specified by A8-A0 and the number of data bytes sent to the device.

Note that the Program OTP Security Register command utilizes an internal 256-buffer for processing. Therefore, the contents of the buffer is altered from its previous state when this command is issued.

See Table 21 for more information on the transfer sequence for this command.

### 6.26.5 Addressing the OTP Security Registers

Each byte within the four OTP security registers can be accessed using bits 8:0 of the address. Bits 23:9 of the address are ignored. The lower 7 bits (6:0) are used to select a byte within a given 128-byte OTP register. The next two bits (8:7) are used to select one of the four OTP registers. This is shown in Table 27.

**Table 27. OTP Register Access Map**

| Address | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **A[23:9]** | **A8** | **A7** | **A6** | **A5** | **A4** | **A3** | **A2** | **A1** | **A0** |
| Don't care | 0 | 0 | OTP Security Register 0 (Factory Programmed by Renesas Electronics)<br>A[6:0] = 0000000 - byte 0, A[6:0] = 1111111 - byte 127 | | | | | | |
| Don't care | 0 | 1 | User defined OTP Security Register 1<br>A[6:0] = 0000000 - byte 128, A[6:0] = 1111111 - byte 255 | | | | | | |
| Don't care | 1 | 0 | User defined OTP Security Register 2<br>A[6:0] = 0000000 - byte 256, A[6:0] = 1111111 - byte 383 | | | | | | |
| Don't care | 1 | 1 | User defined OTP Security Register 3<br>A[6:0] = 0000000 - byte 384, A[6:0] = 1111111 - byte 511 | | | | | | |

### 6.26.6 Programming Status

While the device is programming the OTP Security Register, the RDY/BSY bit in Status Register 1 can be read to determine if the operation is still in progress. For faster throughput, it is recommended that the Status Register be polled rather than waiting the $t_{OTPP}$ time to determine if the data bytes have finished programming. At some point before the OTP Security Register programming completes, hardware clears the WEL bit in the Status Register.

Alternatively, the Active Status Interrupt can also be used to determine when the operation has finished. When this command is executed, hardware drives the state of the RDY/BSY bit in Status Register 1 onto the SO pin. This pin can be monitored externally for a 1 to 0 transition, indicating completion of the program operation. This provides an alternative to repetitive polling of Status Register 1 using the 05h command.

### 6.26.7 Locking OTP Registers 1 - 3

The tables above show the bytes associated with each OTP register. OTP register 0 is locked by default as the value is programmed by Renesas Electronics at the factory and cannot be changed. For OTP registers 1 - 3, the most significant byte is as follows:

- OTP Register 1: MSB = byte 255
- OTP Register 2: MSB = byte 383
- OTP Register 3: MSB = byte 511

For OTP registers 1 - 3, bit-level programming is allowed for the first 127 bytes of the registers. However, the register is locked whenever one or more bits of the most significant byte are programmed. For example, if any bit of byte 255 in OTP register 1 is programmed, that register is locked by hardware can cannot be programmed further.

### 6.26.8 Verifying the Locked Status of a Security Register

As described in the previous subsection, once any bit in the MSB of Security registers 1 - 3 is written, the register is locked and not further programming is allowed. Software can determine the Locked status of each register by reading bits 5:3 (SL3:SL1) of Status Register 2. Each bit corresponds to one of the user defined OTP security registers. If the corresponding bit is cleared, the register is not locked. If the corresponding bit is set, the register is locked and cannot be programmed. See Section 5.4, Status Register 2 for more information.

### 6.26.9 Programming Restrictions

The Program OTP Register command adheres to the following program restrictions.

Early Power-Down

If the device is powered-down during the OTP Security Register program cycle, then the contents of the 128-byte user programmable portion of the OTP Security Register cannot be guaranteed and can not be programmed again.

Deasserting the $\overline{\text{CS}}$ Pin

The $\overline{\text{CS}}$ pin must be deasserted on a byte boundary (multiples of eight bits). Otherwise, the device aborts the operation and the user-programmable portion of the OTP Security Register is not programmed.

Incomplete Address or Data

The WEL bit in the Status Register is cleared if the OTP Security Register program cycle aborts due to an incomplete address being sent or an incomplete byte of data being sent.

Register Previously Programmed

Prior to programming an OTP register, hardware checks the state of the SL3:1 field in Status Register 2. For example, if an attempt is made to program OTP register 1, and the register was previously programmed as indicated by the SL1 bit in Status Register 2 being set, the operation is aborted.

## 6.27 Read OTP Security Register (4Bh)

The Read OTP Security Register command accesses the four specialized OTP Security registers. Any portion of the value can be read out depending on when the $\overline{\text{CS}}$ pin is deasserted.

### 6.27.1 Transfer Format

The 4Bh command follows the 1-1-1 transfer format described in Section 4.2, where the command and address are transferred on the SI pin, and data is transferred on the SO pin. See Figure 7 for a timing diagram of this transaction. In this diagram a single byte of data is shown.

### 6.27.2 Transfer Sequence — Single Register

To perform the Read OTP Security register operation of a single register, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The 4Bh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to read within the memory array. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first. The address breakdown is as follows:
  - A[23:9] are don't care
  - A[8:0] indicate the byte address
- Following the three address bytes, an additional dummy byte must be clocked into the device.

- Data is output on the SO pin. Each byte transfer requires eight clock cycles. The data is always output with the most significant bit of the byte transferred first. The number of bytes transferred depends on how long the $\overline{CS}$ pin remains asserted. A total of 1024 clocks is required to shift out all 128 bytes of data.

### 6.27.3 Transfer Sequence — All Registers

To perform the Read OTP Security register operation of all four registers, the transfer sequence is the same as above, except that the $\overline{CS}$ pin remains asserted until all registers are read. As noted above, a total of 1024 clocks is required to read one register. Therefore, a total of 4096 clocks is required to read out the contents of all four registers.

See Table 21 for more information on the transfer sequence for this command.

## 6.28 Read Status Registers 1 - 3 (05h, 35h, 15h)

Three Read Status Register commands are used to perform a direct read of Status Registers 1 - 3. These registers can also be indirectly accessed using the 65h command as described in the following section. In the direct method, a specific command is executed. Hardware decodes this command and retrieves data from the appropriate Status Register. No address field is required. Status Registers 1 - 3 are accessed by the following commands.

- Read Status Register 1: 05h
- Read Status Register 2: 35h
- Read Status Register 3: 15h

### 6.28.1 Transfer Format

The 05h/35h/15h commands follow the 1-0-1 transfer format described in Section 4.2, where the command is transferred on the SI pin, and data is transferred on the SO pin. No address is required for this command. See Figure 5 for a timing diagram of this transaction.

### 6.28.2 Transfer Sequence

To perform the Read Status Register register operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 05h/35h/15h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Data is output on the SO pin. Each byte transfer requires eight clock cycles. The data is always output with the most significant bit of the byte transferred first. After the last bit (0) of the Status Register has been clocked out, the sequence repeats itself, starting again with bit 7 of the selected register byte. This continues as long as the $\overline{CS}$ pin remains asserted and the clock pin is being pulsed. The data in the Status Register is constantly being updated, so each repeating sequence can output new data.
- Deasserting the $\overline{CS}$ pin terminates the Read Status Register operation and put the SO pin into a high-impedance state. The $\overline{CS}$ pin can be deasserted at any time and does not require that a full byte of data be read.

See Table 21 for more information on the transfer sequence for this command.

## 6.29 Read Status Registers (65h)

The Read Status Register command works in conjunction with an 8-bit address field to perform an indirect read of Status Registers 1 - 6. Registers 1 - 3 can also be directly accessed as described in the previous section. In the indirect method, a Read Status Register command (65h) is executed. Hardware decodes this command and retrieves data from the appropriate Status Register as directed by the 8-bit address field, which follows the command in the sequence. Status Registers 1 - 6 are read in the following manner.

**Table 28. Indirect Addressing of the Status Registers**

| Byte 0 Command | Byte 1 Address | Byte 2 Dummy | Byte 3 Output | Action |
|---|---|---|---|---|
| 65h | 01h | Dummy | S[7:0] | Read Status Register 1 |
| 65h | 02h | Dummy | S[15:8] | Read Status Register 2 |
| 65h | 03h | Dummy | S[23:16] | Read Status Register 3 |
| 65h | 04h | Dummy | S[31:24] | Read Status Register 4 |
| 65h | 05h | Dummy | S[39:32] | Read Status Register 5 |
| 65h | 06h | Dummy | S[47:40] | Read Status Register 6 |

### 6.29.1 Transfer Format

The 65h command follows the 1-1-1 transfer format described in Section 4.2, where the command and address are transferred on the SI pin, and data is transferred on the SO pin. Figure 24 shows a timing diagram for this operation.

### 6.29.2 Transfer Sequence

To perform the Read Status Register command, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 65h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- One address byte is clocked in to specify the address the register to be written. A total of 8 clocks are required to transfer the address. The most significant bit of the address (A[7]) is transferred first.
- Following the one address byte, one dummy byte is clocked into the device.
- Data is output on the SO pin. Each byte transfer requires eight clock cycles. The data is always output with the most significant bit of the byte transferred first. The number of bytes transferred is determined by software. The transfer can be anywhere from a single register to all six registers.
- Deasserting the $\overline{CS}$ pin terminates the read operation and puts the SO pin into high-impedance state. The $\overline{CS}$ pin can be deasserted at any time and does not require a full byte of data be read.

See Table 21 for more information on the transfer sequence for this command.

### 6.29.3 Reading a Single Status Register

To read a specific Status Register, the $\overline{CS}$ pin must first be asserted and the corresponding command of 65h must be clocked into the device. After the command has been clocked in, the 8-bit address of the register to be read is clocked into the device as shown in the above table. Once the address is decoded, hardware begins outputting Status Register data on the SO pin during every subsequent clock cycle.

### 6.29.4 Continuous Read of All Status Registers

To read all six Status Registers in sequence, execute a 65h command with the address field equivalent to 01h. Once the contents of Status Register 1 are read out, the hardware increments the address automatically and begins reading the contents of Status Register 2. This continues until all six Status Registers have been read out as long as the $\overline{CS}$ pin remains asserted and the clock pin is being pulsed. Once the process is started, data is input and output from the device as follows:

**Table 29. Indirect Status Register Read Sequence**

| Clocks | Action |
|---|---|
| 0 - 7 | Command 65h. Clocks 0 - 3 = 0110, and clocks 4 - 7 = 0101. |
| 8 - 15 | Address 01h. Clocks 8 - 11 = 0000, and clocks 12 - 15 = 0001. |
| 16 - 23 | Dummy byte on clocks 16 - 23. |
| 24 - 31 | Status Register 1 data. Clock 24 = bit 7, and clock 31 = bit 0. |
| 32 - 39 | Status Register 2 data. Clock 32 = bit 7, and clock 39 = bit 0. |
| 40 - 47 | Status Register 3 data. Clock 40 = bit 7, and clock 47 = bit 0. |
| 48 - 55 | Status Register 4 data. Clock 48 = bit 7, and clock 55 = bit 0. |
| 56 - 63 | Status Register 5 data. Clock 56 = bit 7, and clock 63 = bit 0. |
| 64 - 71 | Status Register 6 data. Clock 64 = bit 7, and clock 71 = bit 0. |
| 72 - xx | Data is undefined. |

### 6.29.5 Transfer Diagram

Figure 24 shows the bus signals during an indirect register read operation. This type of command requires only a single 8-bit address.

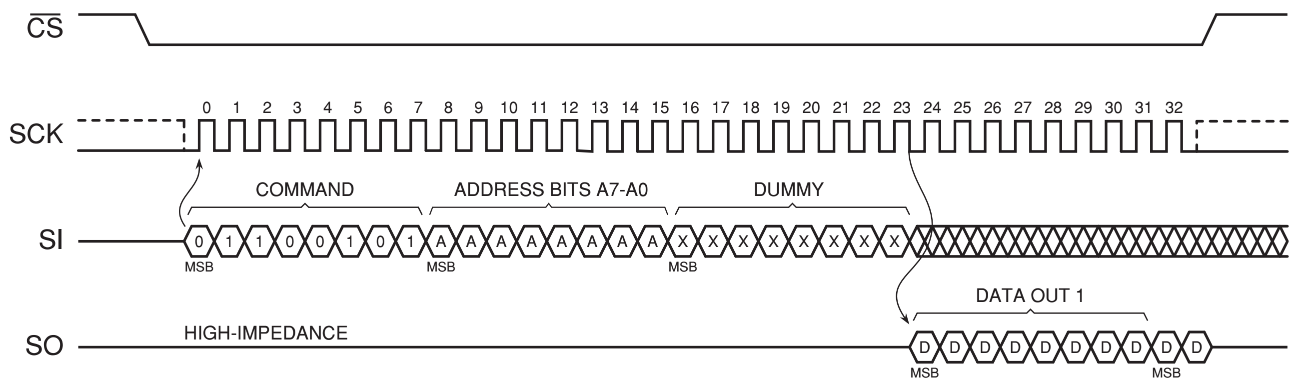



**Figure 24. Status Register Read Operation Showing 8-bit Address Field**

### 6.29.6 Programming Restrictions

The Read Status Register command adheres to the following programming restrictions.

**Reading Register Addresses 7 Through 255**

As the address is an 8-bit value, the device continues to read up to 255 register bytes as long as the $\overline{CS}$ pin remains asserted, even though only six registers are implemented at this time. Starting with a read of register 7 through register 255, the device returns undefined data. If the $\overline{CS}$ pin remains asserted and all 255 bytes are read out, the address wraps around from FFh to 00h. This continues until the $\overline{CS}$ pin is deasserted.

## 6.30 Write Status Registers 1 - 3 — Direct (01h, 31h, 11h)

The Write Status Register commands are used to perform a direct write of Status Registers 1 - 3. These registers can also be indirectly accessed as described in the following section. In the direct method, a specific command is executed. Hardware decodes this command and writes the data to the appropriate Status Register. No address field is required. Status Registers 1 - 3 are accessed by the following commands.

- Write Status Register 1: 01h
- Write Status Register 2: 31h
- Write Status Register 3: 11h

### 6.30.1 Command Prerequisites

To write the volatile version of the Status Register bits, the Write Enable for Volatile Status Register (50h) command must be issued prior to each Write Status Register (01h/31h/11h) command. The Write Enable for the Volatile Status Register command does not set the Write Enable Latch (WEL) bit in Status Register 1. It is only valid for the next following Write Status Register command, to change the volatile Status Register bit values.

To write the non-volatile version of the Status Register bits, the Write Enable (06h) command must be issued prior to each Write Status Register (01h/31h/11h) command.

### 6.30.2 Transfer Format

The 01h/31h/11h commands follow the 1-0-1 transfer format described in Section 4.2, where the command and data are transferred on the SI pin. No address is required for this command. See Figure 6 for a timing diagram of this transaction.

### 6.30.3 Transfer Sequence

To perform the Write Status Register register operation, the CS pin is asserted and the information transferred as follows:

- The 01h/31h/11h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Data is input on the SI pin. Each byte transfer requires eight clock cycles. The data is always input with the most significant bit of the byte transferred first.
- When the CS pin is deasserted, hardware clears the WEL bit in Status Register 1.

See Table 21 for more information on the transfer sequence for this command.

### 6.30.4 Programming Restrictions

The Write Status Register command adheres to the following programming restrictions.

- The complete command must be clocked into the device before the CS pin is deasserted.
- The CS pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no action is taken.
- For compatibility with legacy devices, command 01h can also be used with 2 bytes of data. In such case, the second byte is written to Status Register 2.
- To ensure that only Status Register 1 is written, use command 01h with one data byte only.

## 6.31 Write Status Registers — Indirect (71h)

The Write Status Register command works in conjunction with an 8-bit address field to perform an indirect write of Status Registers 1 - 6. Registers 1 - 3 can also be directly accessed as described in the previous section. In the indirect method, a Write Status Register command (71h) is executed. Hardware decodes this command and writes data to the appropriate Status Register using the 8-bit address field, which follows the command in the sequence. Status Registers 1 - 6 are written in the following manner.

**Table 30. Indirect Addressing of the Status Registers**

| Byte 0 | Byte 1 | Byte 2 | Action |
|---|---|---|---|
| Command | Address | Write Data | |
| 71h | 01h | S[7:0] | Write Status Register 1 |
| 71h | 02h | S[15:8] | Write Status Register 2 |
| 71h | 03h | S[23:16] | Write Status Register 3 |
| 71h | 04h | S[31:24] | Write Status Register 4 |
| 71h | 05h | S[39:32] | Write Status Register 5 |
| 71h | 06h | S[47:40] | Write Status Register 6 |

### 6.31.1 Command Prerequisites

The Write Enable command (06h) must be executed to set WEL bit in Status Register 1 before the Write Status Register (71h) command can be executed. See Section 6.18, Write Enable (06h) for more information.

To write the volatile version of the Status Register bits, the Write Enable for Volatile Status Register (50h) Command must be issued prior to each Write Status Register (71h) command. The Write Enable for the Volatile Status Register command does not set the Write Enable Latch (WEL) bit in Status Register 1. It is only valid for the next following Write Status Register command, to change the volatile Status Register bit values.

To write the non-volatile version of the Status Register bits, the Write Enable (06h) command must be issued prior to each Write Status Register (01h/31h/11h) command.

### 6.31.2 Transfer Format

The 71h command follows the 1-1-1 transfer format described in Section 4.2, where the command, address, and data are transferred on the SI pin. Figure 24 shows a timing diagram for this operation.

### 6.31.3 Transfer Sequence

To perform the Write Status Register operation, the CS pin is asserted and the information transferred as follows:

- The 71h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- One address byte is clocked in to specify the address of the register to be written. A total of 8 clocks are required to transfer the address. The most significant bit of the address (A[7]) is transferred first.
- Data is input on the SI pin. Each byte transfer requires eight clock cycles. The data is always input with the most significant bit of the byte transferred first. If more than one byte of data is provided before the CS pin is deasserted, no register is written.

See Table 21 for more information on the transfer sequence for this command.

### 6.31.4 Writing to a Status Register

To write a specific Status Register, the CS pin must first be asserted and the corresponding command of 71h must be clocked into the device. After the command has been clocked in, the 8-bit address of the register to be written is clocked into the device as shown in the above table. Once the address is decoded, hardware begins writing to the Status Register with the data on the SI pin. This sequence in Table 31 shows a write to Status Register 1.

**Table 31. Indirect Status Register Write Sequence**

| Clocks | Action |
|---|---|
| 0 - 7 | Command 71h. Clocks 0 - 3 = 0111, and clocks 4 - 7 = 0001. |
| 8 - 15 | Address 01h. Clocks 8 - 11 = 0000, and clocks 12 - 15 = 0001. |
| 16 - 23 | Status Register 1 write data. Clock 16 = bit 7, and clock 23 = bit 0. |

### 6.31.5 Transfer Diagram

Figure 25 shows the bus signals during an indirect register write operation. This type of command requires only a single 8-bit address as shown.

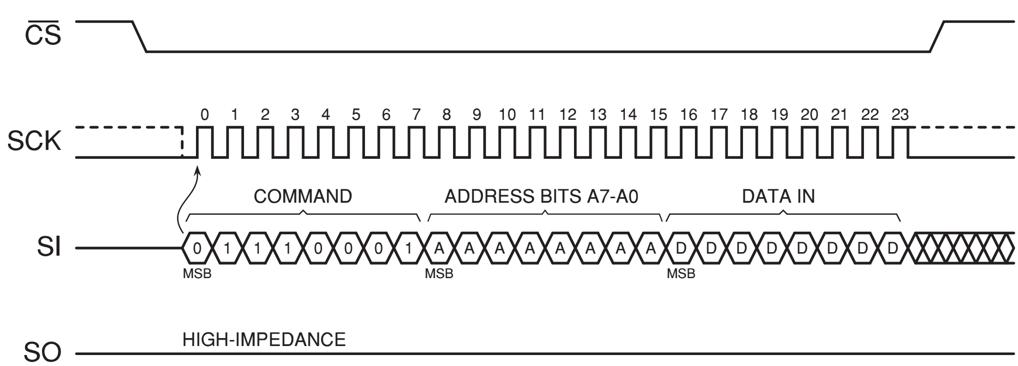



**Figure 25. Status Register Write Operation Showing 8-bit Address Field**

### 6.31.6 Programming Restrictions

The Write Status Registers command adheres to the following programming restrictions.

- The complete command must be clocked into the device before the CS pin is deasserted.
- The CS pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no action is taken.
- The AT25XE161D device implements Status Registers 1 - 6 at addresses 0x01 through 0x06. If the user attempts to write to a register that is not defined or reserved (addresses 0x00, or 0x07 - 0xFF), then the device does not write to any Status Register byte. Instead the operation is aborted and hardware clears the WEL bit in Status Register 1 once the CS pin is deasserted.

## 6.32 Status Register Lock (6Fh)

The Status Register Lock command explicitly locks the Status Registers when the SRP1 and SRP0 bits in Status Registers 2 and 1 are set. Once this occurs, they can no longer be programmed. This command is provided to eliminate the possibility of inadvertently locking the registers by accidentally setting the SRP1 and SRP0 bits in Status Registers 2 and 1. Even if these two bits are set to 2’b11, this command, along with two verification bytes, is required in order for hardware to lock the Status Registers.

### 6.32.1 Command Prerequisites

The Write Enable command (06h) must be executed to set the WEL bit in Status Register 1 before the Status Register Lock command can be executed. See Section 6.18, Write Enable (06h) for more information.

### 6.32.2 Transfer Format

The 6Fh command follows a special 1-0-1 transfer format described in Section 4.2, where the command, address, and data are transferred on the SI pin. However, for this command, two verification bytes are transferred. Figure 26 shows a timing diagram for this operation.

### 6.32.3 Transfer Sequence

To perform the Status Register Lock operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 6Fh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Data is input on the SI pin. The first verification byte of 4Dh is transferred, followed by a second verification byte of 67h. Each byte transfer requires eight clock cycles. The data is always input with the most significant bit of the byte transferred first. Note that these values must be transferred in the correct order. If a value other than 4Dh followed by 67h is transferred, the operation is aborted.

See Table 21 for more information on the transfer sequence for this command.

### 6.32.4 Transfer Diagram

Figure 26 shows the bus signals during a Status Register Lock operation. In this transfer the command is followed by two verification bytes.

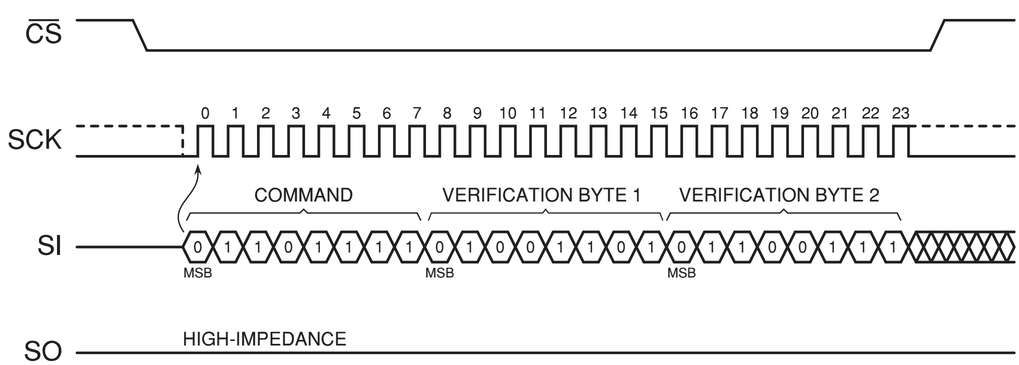



**Figure 26. Status Register Lock Operation with Two Verification Bytes**

### 6.32.5 Programming Restrictions

The Write Status Registers command adheres to the following programming restrictions.

- The complete command must be clocked into the device before the $\overline{CS}$ pin is deasserted.
- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no action is taken.

## 6.33 Deep Power-Down (B9h)

The Deep Power-Down command is used in conjunction with bit 7 (PDM) in Status Register 4 to place the device into Deep Power-Down mode. When the PDM bit is set and the B9h command is executed, the device enters Deep Power-Down mode.

When the device is in Deep Power-Down mode, most commands, including the Read Status Register command, are ignored. The only commands that are accepted while in this mode are the Resume from Deep Power-Down command (ABh), the Enable Reset command (66h), and the Reset Device (99h) commands. Since all other commands are ignored, the mode can be used as an extra protection mechanism against program and erase operations. When the $\overline{\text{CS}}$ pin is deasserted, the device enters the Deep Power-Down mode within the maximum time of $t_{EDPD}$ as shown in Figure 27.

### 6.33.1 Transfer Format

The B9h command follows the 1-0-0 transfer format described in Section 4.2, where the command is transferred on the SI pin. No address or data are transferred for this command. See Figure 3 for a timing diagram of this operation.

### 6.33.2 Transfer Sequence

To perform the Deep Power-Down operation, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The B9h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- When the $\overline{\text{CS}}$ pin is deasserted, the device begins the power-down operation within the time $t_{EDPD}$ defined in Section 7.5.

Figure 27 shows a diagram of this timing parameter.

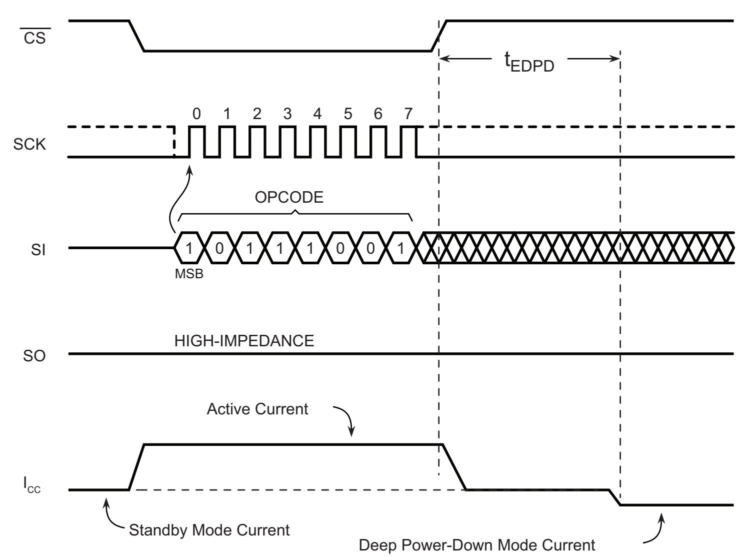



**Figure 27. Entering Deep Power-Down State**

See Table 21 for more information on the transfer sequence for this command.

### 6.33.3 Programming Restrictions

The Deep Power-Down command adheres to the following programming restrictions.

- The $\overline{\text{CS}}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation.
- The B9h command is ignored if an internally self-timed operation such as a program or erase cycle is in progress. In this case, the B9h command must be reissued after the internally self-timed operation has been

completed. Software can monitor the $\overline{\text{RDY}}$/BSY bit in Status Register 1 to determine when the program or erase operation has completed.

## 6.34 Resume from Ultra-Deep Power-Down / Deep Power-Down with Device ID (ABh)

In order to exit the Deep Power-Down (DPD) mode and resume normal device operation, the Resume from Deep Power-Down (ABh) command must be issued. This command, along with the 66h/99h Reset Enable/Reset command, are the only commands that the device recognizes while in the Deep Power-Down mode. This command allows both the exit from DPD mode, as well as a fetch of the Device ID. Which operation is executed depends on the way in which $\overline{\text{CS}}$ is used as described below.

The ABh command can also be used to exit from Ultra-Deep Power-Down (UDPD) mode. Execution of this command while in Ultra-Deep Power-Down causes the device to initialize, and then enters standby mode, where it is ready to accept commands.

### 6.34.1 Command Options

As mentioned above, the ABh command can be used to exit both DPD and UDPD modes. In DPD mode, execution of the ABh command causes the device to exit the DPD state, and also fetch the device ID as described in Section 6.34.3 below. In UDPD mode, the ABh command exits the UDPD state as described in Section 6.34.4, below. Table 32 shows how to exit the DPD and UDPD modes.

**Table 32. Options for Exiting DPD and UDPD Modes**

| Command | Currently in DPD Mode? | Currently in UDPD Mode? | Exit DPD Mode | Exit UDPD Mode | Fetch Device ID |
|---|---|---|---|---|---|
| ABh | No | No | No | No | Yes |
| | Yes | No | Yes | No | Yes |
| | No | Yes | No | Yes | No |

### 6.34.2 Transfer Format — Resume from Deep Power-Down

For this operation, the ABh command follows the 1-0-0 transfer format described in Section 4.2, where the command is transferred on the SI pin. No address or data are transferred for this command. See Figure 3 for a timing diagram of this operation.

### 6.34.3 Transfer Format — Resume from Deep Power-Down and Obtain Device ID

For this operation, the ABh command follows the 1-1-1 transfer format described in Section 4.2, where the command is transferred on the SI pin and address and data on the SO pin. See Figure 7 for a timing diagram of this operation. Note that no address is transferred for this command. Rather, three dummy bytes are transferred in place of the address.

### 6.34.4 Transfer Sequence — Resume from Deep Power-Down

To perform the Resume from Deep Power-Down operation, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The ABh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- For this operation, the $\overline{\text{CS}}$ pin must be deasserted on the 8th clock, immediately after the command is issued. This causes the device begins the resume from power-down operation. When the $\overline{\text{CS}}$ pin is deasserted after the 8th clock, the device exits the Deep Power-Down mode within the maximum time of $t_{RDPD}$ and returns to the standby mode. After the device has returned to the standby mode, normal command operations such as Read Array can be resumed.

See Table 21 for more information on the transfer sequence for this command.

### 6.34.5 Transfer Sequence — Resume from Deep Power-Down and Obtain Device ID

To perform the Resume from Deep Power-Down operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The ABh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three dummy bytes are then clocked in for the address. These clocks are required to provide time for the device to exit power-down mode and fetch the device ID. A total of 24 clocks are required to transfer the three dummy bytes.
- For this operation, the $\overline{CS}$ pin continues to be asserted after the 8th clock. In this operation, the $\overline{CS}$ pin is held low for the time it takes to transfer the device ID. Therefore, 40 clocks are required to complete this operation; 8 for the command, 24 for the dummy bytes, and 8 clocks for the device ID.
- The device ID is shifted out on the SO pin, with the most significant bit of the value being transmitted first. The device ID is read out continuously until the $\overline{CS}$ pin is deasserted.

### 6.34.6 Transfer Sequence — Resume from Ultra-Deep Power-Down

To perform the resume from Ultra-Deep Power-Down operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The ABh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Once $\overline{CS}$ is deasserted, hardware initiates an internal reset of the device and the SRAM contents are lost.
- The device exits the Ultra-Deep Power-Down mode within the time $t_{RUDPD}$ defined in Section 7.5.

Figure 28 shows a timing diagram of this timing parameter.

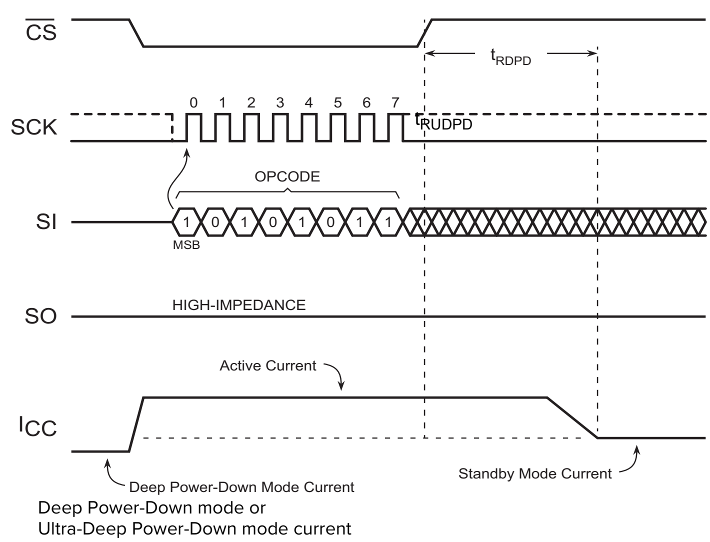



**Figure 28. Resume from Deep Power-Down or Ultra-Deep Power-Down**

### 6.34.7 Programming Restrictions

The Resume from Deep Power-Down command adheres to the following programming restrictions.

- The complete command must be clocked into the device before the $\overline{CS}$ pin is deasserted.
- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no action is taken.

## 6.35 Ultra-Deep Power-Down (79h)

The Ultra-Deep Power-Down (UDPD) mode allows the device to further reduce energy consumption compared to Deep Power-Down modes by shutting down additional internal circuitry. The UDPD mode can be entered by either executing the 79h command, or by executing the B9h command with bit 7 (PDM) of Status Register 4 cleared. When the device is in the Ultra-Deep Power-Down mode, all commands except the ABh command are ignored.

Execution of the ABh command allows the device to exit from Ultra-Deep Power-Down mode. Execution of this command while in Ultra-Deep Power-Down causes the device to initialize, and then enters standby mode.

### 6.35.1 Transfer Format

The 79h command follows the 1-0-0 transfer format described in Section 4.2, where the command is transferred on the SI pin. No address or data are transferred for this command. See Figure 3 for a timing diagram of this operation.

### 6.35.2 Transfer Sequence

To perform the Ultra-Deep Power-Down operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 79h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- When the $\overline{CS}$ pin is deasserted, the device enters the Ultra-Deep Power-Down mode within the maximum time of $t_{EUDPD}$. Any additional data clocked into the device after this command is ignored.

Figure 29 shows a diagram with this timing parameter.

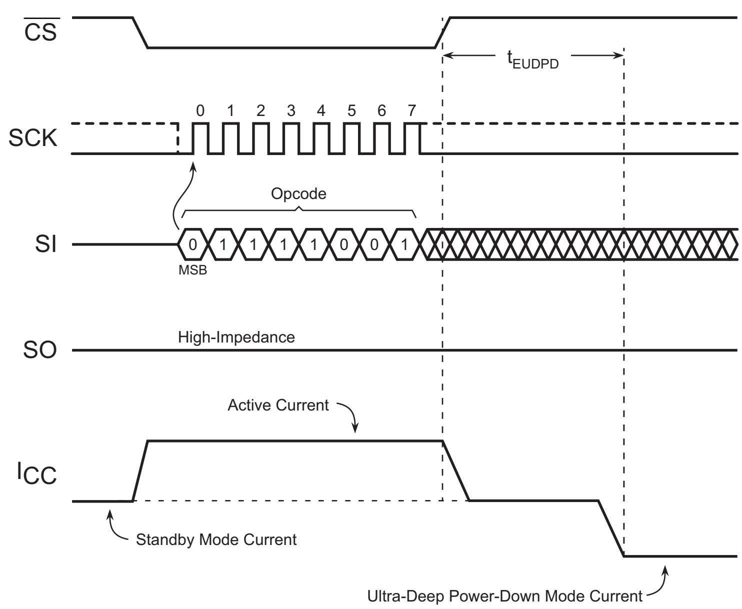



**Figure 29. Entering Ultra-Deep Power-Down State**

See Table 21 for more information on the transfer sequence for this command.

### 6.35.3 Programming Restrictions

The following events can cause the Ultra-Deep Power-Down operation to be aborted.

- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and return to the standby mode once the $\overline{CS}$ pin is deasserted. Also, the device defaults to the standby mode after a power cycle.
- The 79h command is ignored if an internally self-timed operation such as a program or erase cycle is in progress. The 79h command must be reissued after the internally self-timed operation has been completed.

Software can monitor the $\overline{\text{RDY}}$/BSY bit in Status Register 1 to determine when the program or erase operation has completed.

## 6.36 Enable Reset (66h) and Reset Device (99h)

Executing the Reset (66h/99h) command terminates all internal operations and causes the device to initialize. After reset, the device is set to its default parameters and all previous register settings are lost.

### 6.36.1 Transfer Format

The 66h/99h commands are issued as two back-to-back 1-0-0 commands described in Section 4.2. The first command (66h) enables the reset function and is transferred on the SI pin. The second command (99h) immediately follows the 66h command and initiates a reset of the device. This command is also transferred on the SI pin. No address or data are transferred for this command. See Figure 3 for a timing diagram of this operation.

### 6.36.2 Transfer Sequence

To perform a device reset operation, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The 66h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Deassert the $\overline{\text{CS}}$ pin for one clock cycle.
- Reassert the $\overline{\text{CS}}$ pin.
- The 99h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first. Note that no other command can be issued in between the 66h command and the 99h command.
- When the $\overline{\text{CS}}$ pin is deasserted after the 99h command, the device initiates the reset operation within a time of $t_{SWRST}$. During the reset sequence, no other command can be accepted by the device.

See Table 21 for more information on the transfer sequence for this command.

Figure 30 shows an example of the back-to-back command sequence.

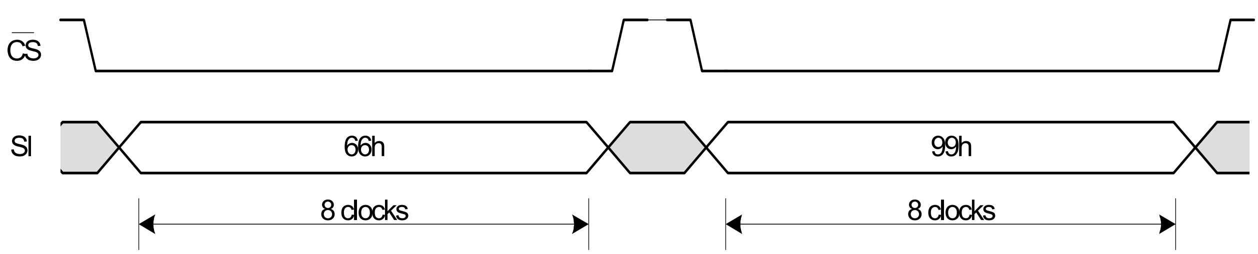



**Figure 30. Enable Reset and Reset Command Sequence (SPI Mode)**

### 6.36.3 Programming Restrictions

The Enable Reset command adheres to the following programming restrictions.

- The complete command must be clocked into the device before the $\overline{\text{CS}}$ pin is deasserted.
- The $\overline{\text{CS}}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and no action is taken.
- If an erase or program operation has been suspended (75h command has been executed prior to the 66h/99h command) and the device reset sequence is initiated, the data is corrupted. To prevent this from happening, check the $\overline{\text{RDY}}$/BSY in Status Register 1 and the SUSP bit in Status Register 2 to make sure each bit is cleared before issuing the 66h/99h Reset command sequence.

## 6.37 Terminate (F0h)

In some applications, it might be necessary to prematurely terminate a program or erase operation rather than wait the for the program or erase operation to complete normally. The Terminate command immediately aborts any operation in progress and returns the device to an idle state.

Since the need to terminate the operation is immediate, the Write Enable command does not need to be issued prior to the Terminate command. Therefore, the Terminate command operates independently of the state of the WEL bit in Status Register 1.

Once this command is issued, the operation in progress cannot be continued. To suspend and then continue an operation, use the Program/Erase Suspend (75h) and Program/Erase Resume (7Ah) commands.

### 6.37.1 Command Prerequisites

The Terminate command can be executed only if the command has been enabled by setting the Reset Enabled (RSTE) bit in Status Register 5 using the Write Status Register 5 command (71h + 05h address offset). This command must be executed before the Terminate command is executed. If the Reset command has not been enabled (the RSTE bit in is cleared), then any attempt at executing the Terminate command is ignored.

### 6.37.2 Transfer Format

The F0h command follows the 1-0-1 transfer format described in Section 4.2, where the command and data are transferred on the SI pin. Immediately after the command is transferred, a confirmation data byte is transferred. This byte is required by the hardware before the Terminate command can be executed. A timing diagram of this operation is shown in Figure 6.

### 6.37.3 Transfer Sequence

To perform device reset operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The F0h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- The D0h confirmation byte is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first. Note that is the confirmation byte is any value other than D0h, the entire command is ignored and the device returns to the idle state.
- When the $\overline{CS}$ pin is deasserted after the D0h confirmation byte, the device initiates the Terminate operation within a time of $t_{SWTERM}$. During the Terminate sequence, no other command can be accepted by the device.

See Table 21 for more information on the transfer sequence for this command.

### 6.37.4 Programming Restrictions

The F0h Terminate command adheres to the following programming restrictions.

- If a program or erase operation is in progress and cannot complete before the operation is terminated through the F0h command, the contents of the page being programmed or erased cannot be guaranteed to be valid.
- The F0h command does not reset the device configuration registers in volatile mode. If a reset of the internal state is desired, perform a 66h/99h command. This command resets the device and all programmable parameters and all volatile register contents are lost.
- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and returns to the standby mode once the $\overline{CS}$ pin is deasserted. Also, the device defaults to the standby mode after a power cycle.

## 6.38 Read Manufacturer/Device ID (90h)

The Read Manufacturer/Device ID (90h) command provides both the JEDEC assigned manufacturer ID, and the specific device ID.

### 6.38.1 Transfer Format

The 90h command follows the 1-1-1 transfer format described in Section 4.2, where the command and address are transferred on the SI pin, and data is transferred on the SO pin. See Figure 7 for timing diagram of this operation.

### 6.38.2 Transfer Sequence

To perform the Read Array operation, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The 90h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three dummy bytes are clocked in during the address phase. A total of 24 clocks are required. These three dummy bytes are not used
- Data is output on the SO pin. Two bytes are transferred, requiring a total of 16 clock cycles. The data is driven onto the bus as follows:
  - 1st data clock: bit 7 of the manufacturer ID is output on the SO pin
  - 2nd data clock: bit 6 of the manufacturer ID is output on the SO pin
  - .........
  - 8th data clock: bit 0 of the manufacturer ID is output on the SO pin
  - 9th data clock: bit 7 of the device ID is output on the SO pin
  - 10th data clock: bit 6 of the device ID is output on the SO pin
  - .........
  - 16th data clock: bit 0 of the device ID is output on the SO pin
- Deasserting the $\overline{\text{CS}}$ pin terminates the read operation and puts the SO pin into high-impedance state. If the $\overline{\text{CS}}$ pin remains asserted, the device continues to shift out the manufacturer ID followed by the device ID.

See Table 21 for more information on the transfer sequence for this command.

## 6.39 Quad Read Manufacturer/Device ID (94h)

The Read Manufacturer/Device ID command operates in Quad I/O mode, where the SI, SO, $\overline{\text{WP}}$, and $\overline{\text{HOLD}}$ pins are all bidirectional, to allow the manufacturer and device ID information to be transmitted at four times the speed of the Read Manufacturer/Device ID command (90h).

### 6.39.1 Transfer Format

The 94h command follows the 1-4-4 transfer format described in Section 4.6, where the command is transferred on the SI pin, and the address and data are transferred on the SI, SO, $\overline{\text{WP}}$, and $\overline{\text{HOLD}}$ pins. See Figure 15 for a timing diagram of this transaction.

### 6.39.2 Transfer Sequence

To perform the Quad Read Manufacturer/Device ID command, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The 94h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three dummy address bytes are clocked in during the address phase. A total of 6 clocks are required. These three dummy bytes are ignored.

- The mode byte is clocked into the device. Two clocks are required to transfer the command. The one dummy byte is ignored.
- Four dummy clocks are driven to the device prior to the output of data.
- Data is output on the SI, SO, $\overline{WP}$, and $\overline{HOLD}$ pins. Each byte transfer requires two clock cycles. The data is always output with the most significant bit of the byte transferred first. The first byte transferred is the manufacturer ID. The second byte transferred is the device ID. As long as $\overline{CS}$ is asserted, hardware continues to shift these two bytes out onto the bus. Each data byte is shifted out of the device as follows:
  - First data clock: bit 7 of the manufacturer ID is output on the $\overline{HOLD}$ pin
  - First data clock: bit 6 of the manufacturer ID is output on the $\overline{WP}$ pin
  - First data clock: bit 5 of the manufacturer ID is output on the SO pin
  - First data clock: bit 4 of the manufacturer ID is output on the SI pin
  - Second data clock: bit 3 of the manufacturer ID is output on the $\overline{HOLD}$ pin
  - Second data clock: bit 2 of the manufacturer ID is output on the $\overline{WP}$ pin
  - Second data clock: bit 1 of the manufacturer ID is output on the SO pin
  - Second data clock: bit 0 of the manufacturer ID is output on the SI pin
  - In clocks 3 and 4, the device ID is shifted out on byte 2 in the same manner as above

## 6.40 Read JEDEC ID (9Fh)

The Read JEDEC ID command allows software to identify the manufacturer and device ID information for the device while it is in the system. This command allows the ID information to be read out a relatively low clock frequency to ensure the device can be identified. Once the identification process is complete, the application can increase the clock frequency as necessary. For more information, see the Serial Flash Discoverable Parameters (SFDP) JEDEC specification, edition JESD216C.

### 6.40.1 Transfer Format

The 9Fh command follows the 1-0-1 transfer format described in Section 4.2, where the command is transferred on the SI pin, and data is transferred on the SO pin. See Figure 31 for timing diagram of this operation. Note that no address is required for this type of read operation.

### 6.40.2 Transfer Sequence

To perform the JEDEC ID operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 9Fh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first. No address is required for this transaction.
- Data is output on the SO pin. Five bytes are transferred, requiring a total of 40 clock cycles. The data is driven onto the bus as follows:
  - 1st data clock: bit 7 of the manufacturer ID is output on the SO pin
  - .........
  - 8th data clock: bit 0 of the manufacturer ID is output on the SO pin
  - 9th data clock: bit 7 of the device ID byte 1 is output on the SO pin
  - .........
  - 16th data clock: bit 0 of the device ID byte 1 is output on the SO pin
  - 17th data clock: bit 7 of the device ID byte 2 is output on the SO pin
  - .........
  - 24th data clock: bit 0 of the device ID byte 2 is output on the SO pin
  - 25th data clock: bit 7 of the extended device string length byte is output on the SO pin
  - .........

- 32nd data clock: bit 0 of the extended device string length byte is output on the SO pin
- 33rd data clock: bit 7 of the extended device string value byte is output on the SO pin
- .........
- 40th data clock: bit 0 of the extended device string value byte is output on the SO pin

▪ Deasserting the $\overline{CS}$ pin terminates the read operation and puts the SO pin into high-impedance state. If the $\overline{CS}$ pin remains asserted, the device continues to shift out the manufacturer ID followed by the device ID.

See Table 21 for more information on the transfer sequence for this command. The following tables below show the exact contents of each data byte.

**Table 33. Manufacturer and Device ID Details**

| Data Type | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Hex | Details |
|---|---|---|---|---|---|---|---|---|---|---|
| Manufacturer ID | JEDEC Assigned Code | | | | | | | | 1Fh | JEDEC Code: 0001 1111 (1Fh for Renesas Electronics) |
| | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | | |
| Device ID (Part 1) | Family Code | | | Density Code | | | | | 46h | Family Code: 0100 (4h)<br>Density Code: 0110 (6h) |
| | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | | |
| Device ID (Part 2) | Sub Code | | | Product Version Code | | | | | 0Ch | Sub Code: 0000 (0h)<br>Product Version: 1100 (Ch) |
| | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | | |
| Device ID (Part 3)<br>(EDI String Length) | Number of Extended Device ID Bytes to Follow | | | | | | | | 01h | Bytes to follow: 1 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | |
| Device ID (Part 4)<br>(EDI String Value) | Extended device Identity Value | | | | | | | | 0xh | x: Device Variant/Option. See table below. |
| | 0 | 0 | 0 | 0 | x | x | x | x | | |

**Table 34. Device ID Part 4 Variants — EDI String Value**

| EDI String Value | Variant |
|---|---|
| 00h | Initial device |
| 01h | TBD |
| 02h | TBD |
| 03h | TBD |
| 04h | TBD |
| 05h | TBD |
| 06h | TBD |
| 07h | TBD |
| 08h - 0Fh | Reserved |

### 6.40.3 Transfer Diagram

Figure 31 shows the bus signals during an indirect register read operation. This type of command requires only a single 8-bit address.

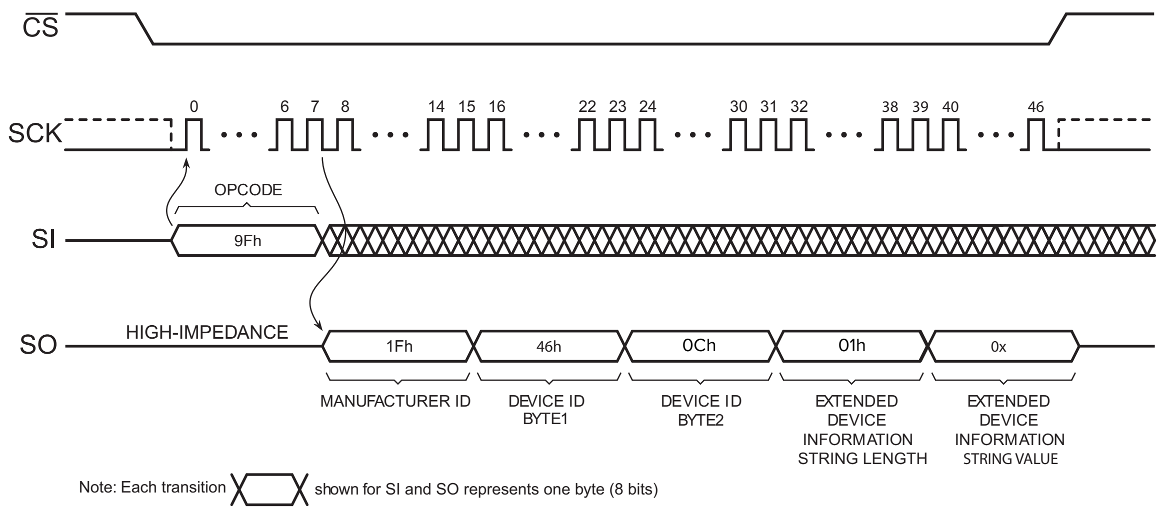



**Figure 31. Read JEDEC ID**

## 6.41 Active Status Interrupt (25h)

To simplify the readout of the $\overline{\text{RDY}}$/BSY bit in Status Register 1, the Active Status Interrupt command (25h) can be used. When this command is used, it is not necessary to continuously read the $\overline{\text{RDY}}$/BSY bit to determine when the command has completed. Instead, hardware drives the value of the $\overline{\text{RDY}}$/BSY bit into the SO pin. The state of the pin is updated in every internal clock cycle used for the self-timed operation. The host can monitor the SO pin until it toggles from a logic 1 to a logic 0, indicating that the operation has completed. At this point $\overline{\text{CS}}$ can be deasserted to complete the operation.

For example, the programming of all or part of the memory array using a command such as the Byte/Page Program (02h) can require many clock cycles to complete. In this case, once the 02h command and address have been transferred and the $\overline{\text{CS}}$ pin has been deasserted to signal the end of the bus transaction, the Active Status Interrupt command can be executed to monitor the state of the $\overline{\text{RDY}}$/BSY bit in Status Register 1. Once the program operation has been completed internally, hardware clears the $\overline{\text{RDY}}$/BSY bit, indicating the operation is complete. For more information, see Section 4.14, Active Status Interrupt.

### 6.41.1 Transfer Format

The 25h command is a version of the 1-0-1 transfer format described in Section 4.2, where the command is transferred on the SI pin, and data, in the form of an interrupt status, is output by the device onto the SO pin. See Figure 32 below for more information.

### 6.41.2 Transfer Sequence

To perform the Active Status Interrupt operation, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The 25h command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first. No address is required for this transaction.
- In SPI Mode 0, no dummy clocks are required. However, for SPI Mode 3, one dummy byte is required. The value of the SI pin after the command (or dummy byte) is clocked in has no significance to the operation.
- The state of the $\overline{\text{RDY}}$/BSY bit in Status Register 1 is output onto the SO pin, and is continuously updated by the device for as long as the $\overline{\text{CS}}$ pin remains asserted. Once the command and dummy byte are transferred, the SCK pin can be suspended. If the $\overline{\text{RDY}}$/BSY bit changes from 1 to 0 while the $\overline{\text{CS}}$ pin is asserted, the SO line

also changes from 1 to 0. Note that the $\overline{\text{RDY}}$/BSY bit cannot change from 0 to 1 during an operation, so if the SO line is already 0, it does not change.

- Deasserting the $\overline{\text{CS}}$ pin terminates the read operation and puts the SO pin into high-impedance state. The $\overline{\text{CS}}$ pin can be deasserted at any time and does not require that a full byte of data be read.

See Table 21 for more information on the transfer sequence for this command. Figure 32 shows the state of the SO pin during execution of an Active Status Interrupt command.

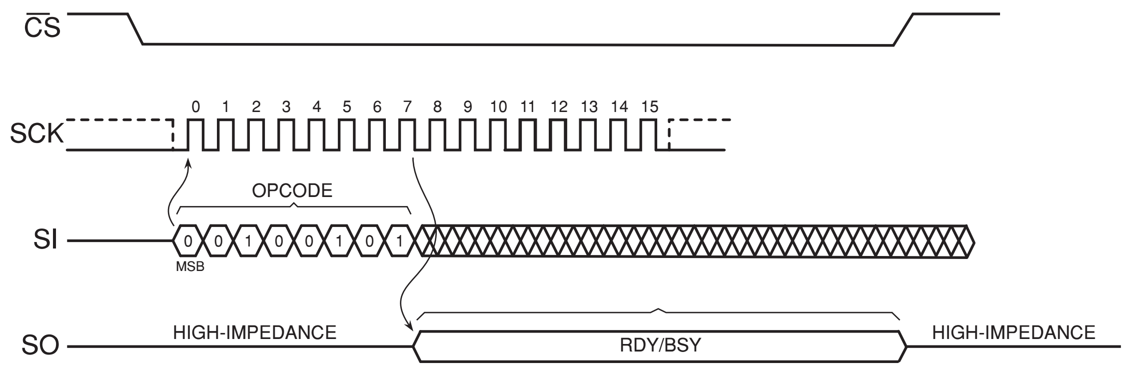



**Figure 32. Active Status Interrupt**

### 6.41.3 Programming Restrictions

The $\overline{\text{CS}}$ pin must be deasserted on a byte boundary (multiple of eight bits). Otherwise, the device aborts the operation and returns to the standby mode once the $\overline{\text{CS}}$ pin is deasserted. Also, the device defaults to the standby mode after a power cycle.

## 6.42 Single Command Read-Modify-Write — EEPROM Emulation (0Ah)

The device incorporates a completely self-contained read-modify-write operation that can be used to reprogram any number of sequential bytes in a page in the main memory array without affecting the rest of the bytes in the same page. This command allows the device to easily emulate an EEPROM by providing a method to modify a single byte or more in the main memory in a single operation, without the need for pre-erasing the memory or the need for any external RAM buffers. The main advantage of this command is that it allows a memory location to be erased and reprogrammed in one operation. For more information, see Section 4.16, Read-Modify-Write.

### 6.42.1 Command Prerequisites

Before the RMW command can be issued, the Write Enable command (06h) must have been previously issued to set the WEL bit in Status Register 1.

### 6.42.2 Transfer Format

The transfer format for the 0Ah command is comprised of a 1-1-1 write operation as described in Section 4.2. Once this command is executed, the read of the memory and subsequent modify operation are handled internally. For a timing diagram of this type of transaction, See Figure 9 for a timing diagram of this operation.

### 6.42.3 Transfer Sequence

To perform the RMW operation, the $\overline{\text{CS}}$ pin is asserted and the information transferred as follows:

- The 0Ah command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to read within the memory array. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first. The organization of the address field is as follows:
  - A[23:21]: These bits are don't care.

- A[20:8]: These 13 bits specify the page in the main memory to be written.
- A[7:0]: These address bits designate the starting byte address within the page to reprogram.

- After the address bytes have been clocked in, any number of sequential data bytes from 1 to 256 can be clocked into the device. If the end of the buffer is reached when clocking in the data, the device wraps around back to the beginning of the buffer.
- After all data bytes have been clocked into the device, a low-to-high transition on the $\overline{CS}$ pin starts the self-contained, internal read-modify-write operation. Only the data bytes that were clocked into the device is reprogrammed in the main memory. If only one data byte was clocked into the device, then only one byte in main memory is reprogrammed and the remaining bytes in the main memory page remain in their previous state.

See Table 21 for more information on the transfer sequence for this command.

### 6.42.4 Programming Restrictions

The Byte Write command adheres to the following programming restrictions.

- The $\overline{CS}$ pin must be deasserted on a byte boundary (multiples of 8 bits). Otherwise, the operation is aborted and no data is programmed. The reprogramming of the data bytes is internally self-timed. During this time, the $\overline{RDY}$/BSY bit in Status Register 1 indicates the device is busy.

### 6.42.5 Error Reporting

The device also incorporates an intelligent erase and programming algorithm that can detect when a byte location fails to erase or program properly. If an erase or program error arises, it is indicated by the EE and PE bits in Status Register 4.

## 6.43 Low Battery Detect (EFh)

The low battery detect command allows software to periodically check the battery level of the system. This is most important in battery operated devices, where knowing the voltage level of the battery and when it may be about to fail can be critical.

The device provides a number of programmable parameters that are used to monitor battery life in the system. The user can select at what level the battery reports a failure, how much load is applied during the test, and for how long. This flexibility allows the user to customize the battery test to their particular system requirements. For more information on how to set up the test variables and the programming sequence, see Section 4.15, Low Battery Detect. Note that the AT25XE161D device uses Status Register 6 to monitor battery usage.

This operation supports the Active Interrupt (25h) command. This command can be used to monitor the state of the $\overline{RDY}$/BSY bit in Status Register 1 to determine when an operation has completed. For more information, see Section 6.41, Active Status Interrupt (25h).

### 6.43.1 Transfer Format

The Low Battery Detect sequence is a two step operation. The initial EFh command follows the 1-0-0 transfer format, where the command is transferred on the SI pin. No address or data are required for this command. See Figure 3 for a timing diagram of this transfer type.

Once the operation is complete, a Status Register Read (65h) command can be used to read out the result of the battery test. Although this sequence follows the 1-1-1 format, only eight bits of address are required. For more information and a timing diagram of this type of transfer, see Section 6.29.5, Transfer Diagram.

### 6.43.2 Transfer Sequence

To perform the Low Battery Detect command, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The EFh command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.

- Deassert the $\overline{CS}$ pin. Once this pin is pulled high, hardware initiates the low battery test. The parameters used for the test were previously programmed into bits 5:0 of Status Register 6.
- To execute the optional 25h command, reassert the $\overline{CS}$ pin. Drive the Active Status Interrupt (25h) command onto the SI pin.
- Deassert the $\overline{CS}$ pin. Once this pin is pulled high, hardware drives the state of the $\overline{RDY}$/BSY bit in Status Register 1 onto the SO pin. The state of the $\overline{RDY}$/BSY bit is updated every clock cycle. During the low battery test, the host need only to monitor a high to low transition on this pin to determine when the test has completed.
- Once a high to low transition occurs on the SO pin indicating the end of the test, the test result can be determined by driving the Read Status Register (65h) command onto the SI pin. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- One address byte of 06h is clocked in to specify the location of Status Register 6. The most significant bit of the address (A[7]) is transferred first. Based on this address, hardware fetches the contents of Status Register 6 and drives the 8-bit value onto the SO pin.
- Once the contents of SR6 are driven onto the bus, the host reads bits 7:6 of this value to determine the result of the test. If this field contains a value of 2’b10, the voltage in the battery is greater than the threshold set by the LBVL field in bits 5:3. A value of 2’b11 indicates the voltage in the battery is less than the threshold set by the LBVL field.

## 6.44 Serial Flash Discoverable Parameters (5Ah)

The device contains a 256-byte Serial Flash Discoverable Parameter (SFDP) register. The SFDP register can be sequentially read in a similar fashion to the Read Array operation up to the maximum clock frequency specified by $f_{SCK}$.

### 6.44.1 Transfer Format

The 5Ah command follows the 1-1-1 transfer format described in Section 4.2, where the command, address, and dummy bytes are transferred on the SI pin, and data is transferred on the SO pin. Figure 8 shows a timing diagram of a read operation with dummy bytes. In this diagram one byte of data is transferred. Additional bytes can be transferred as long as the $\overline{CS}$ pin is asserted.

### 6.44.2 Transfer Sequence

To perform the SFDP operation, the $\overline{CS}$ pin is asserted and the information transferred as follows:

- The 5Ah command is clocked into the device. Eight clocks are required to transfer the command. The most significant bit of the command is transferred first.
- Three address bytes are clocked in to specify the starting address location of the first byte to read within the memory array. A total of 24 clocks are required to transfer the address. The most significant bit of the address (A[23]) is transferred first.
- Following the three address bytes, an additional dummy byte must be clocked into the device.
- Data is output on the SO pin. Each byte transfer requires eight clock cycles. The data is always output with the most significant bit of the byte transferred first. The number of bytes transferred is determined by software. The transfer can be anywhere from a single byte to the entire SFDP value. The most significant bit of each data byte is transferred first.
- Deasserting the $\overline{CS}$ pin terminates the read operation and puts the SO pin into high-impedance state. The $\overline{CS}$ pin can be deasserted at any time and does not require a full byte of data be read. The format of the SFDP register follows the format provided in JEDEC Standard No. 216 Rev B.

### 6.44.3 Programming Restrictions

When the last byte (0000FFh) of the SFDP Security Register has been read, the device continues reading back at the beginning of the register (000000h). No delays are incurred when wrapping around from the end of the register to the beginning of the register.

### 6.44.4 SFDP Organization

Contact Renesas Electronics for SFDP table information.

# 7. Electrical Specifications

## 7.1 Absolute Maximum Ratings

| | |
|---|---|
| Temperature under bias | -55 °C to +125 °C |
| Storage Temperature | -65 °C to +150 °C |
| All input voltages (including NC pins) with respect to ground | -0.6 V to ($V_{CC}$ + 0.5 V) |
| All output voltages with respect to ground | -0.6 V to ($V_{CC}$ + 0.5 V) |

*Notice: Stresses beyond those listed under Absolute Maximum Ratings can cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods can affect device reliability.

## 7.2 DC and AC Operating Range

| Parameter | AT25XE161D |
|---|---|
| Operating Temperature (Case) | -40 °C to 85 °C |
| $V_{CC}$ Power Supply | 1.65 V to 3.6 V |

## 7.3 DC Characteristics for 1.65 V to 3.6 V (-40 °C to 85 °C)

| Symbol | Parameter | Condition | 1.65 V to 3.6 V | | | Units |
|---|---|---|---|---|---|---|
| | | | Min | Typ[1] | Max | |
| $I_{UDPD}$[2] | Ultra-Deep Power-Down Current | $\overline{CS}$ = $V_{CC}$. All other inputs at 0V or $V_{CC}$ | | 5 - 7 | 600 | nA |
| $I_{DPD}$ | Deep Power-Down Current | $\overline{CS}$, $\overline{RESET}$, $\overline{WP}$ = $V_{CC}$. All other inputs at 0V or $V_{CC}$ | | 8.2 | 18 | µA |
| $I_{SB}$[3] | Standby Current | $\overline{CS}$, $\overline{RESET}$, $\overline{WP}$ = $V_{CC}$. All other inputs at 0V or $V_{CC}$ | | 30 | 55 | µA |
| $I_{CC1}$[4] | SPI (33 MHz) | $I_{OUT}$ = 0 mA | | 6.2 | 11.5 | mA |
| | SPI (104 MHz) | $I_{OUT}$ = 0 mA | | 8.3 | 15.0 | mA |
| | Dual (104 MHz) | $I_{OUT}$ = 0 mA | | 10.4 | 15.0 | mA |
| | Quad (104 MHz) | $I_{OUT}$ = 0 mA | | 12.6 | 17.5 | mA |
| $I_{CC3}$ | Active Current, Program Operation | $\overline{CS}$ = $V_{CC}$ | | 10.1 | 14.0 | mA |
| $I_{CC4}$ | Active Current, Erase Operation | $\overline{CS}$ = $V_{CC}$ | | 10.7 | 15.0 | mA |
| $I_{LI}$ [5] | Input Load Current | All inputs at CMOS levels | | | ±2 | µA |
| $I_{LO}$ | Output Leakage Current | All inputs at CMOS levels | | | ±2 | µA |
| $V_{IL}$ | Input Low Voltage | | | | $V_{CC}$ x 0.2 | V |
| $V_{IH}$ | Input High Voltage | | $V_{CC}$ x 0.8 | | | V |
| $V_{OL}$ | Output Low Voltage | $I_{OL}$ = 100 µA | | | 0.2 | V |
| $V_{OH}$ | Output High Voltage | $I_{OH}$ = -100 µA | $V_{CC}$ - 0.2 | | | V |

1. Typical values measured at 1.8V @ 25 °C for the 1.65 V to 3.6 V range.
2. $I_{UDPD}$ value is estimated. Not 100% tested.
3. During continuous read mode (0-4-4), $I_{SB}$ may exceed maximum limit.
4. Checkerboard pattern.
5. $\overline{WP}$ ($IO_2$) and $\overline{HOLD/RESET}$ ($IO_3$) pins have an internal pull-up resistor. It's input load current is +2 / -40 µA.

## 7.4 AC Clock Characteristics

| Symbol | Parameter | 1.65 V to 3.6 V | | | 2.7 V to 3.6 V | | | Units |
|---|---|---|---|---|---|---|---|---|
| | | Min | Typ | Max | Min | Typ | Max | |
| $f_{SCK}$ | Maximum clock frequency for all operations (except 03h, 0Bh, 3Bh, 6Bh, EBh, and E7h | | | 108 | | | 133 | MHz |
| | Maximum clock frequency for 0Bh and 3Bh | | | 96 | | | 108 | MHz |
| | Maximum clock frequency for 6Bh command | | | 90 | | | 104 | MHz |
| | Maximum clock frequency for EBh and E7h | | | 108 | | | 133 | MHz |
| $f_{RDLF}$ | Maximum clock frequency for 03h command (Read Array – Low Frequency) | | | 50 | | | 50 | MHz |

1. See Table 22 and Table 23 for the maximum frequency of EBh/E7h commands in various modes.

## 7.5 AC Characteristics – All Other Parameters

| Symbol [1] | Parameter | 1.65V to 3.6V | | | Units |
|---|---|---|---|---|---|
| | | Min | Typ | Max | |
| $t_{CLKH}$ | SCK High Time | 4.5 | | | ns |
| $t_{CLKL}$ | SCK Low Time | 4.5 | | | ns |
| $t_{CLKR}$[1] | Clock Rise Time, Peak-to-Peak (Slew Rate) | 0.75 | | | V/ns |
| $t_{CLKF}$[1] | Clock Fall Time, Peak-to-Peak (Slew Rate) | 0.75 | | | V/ns |
| $t_{CSH}$ | Minimum $\overline{\text{CS}}$ High Time | 35 | | | ns |
| $t_{CSLS}$ | $\overline{\text{CS}}$ Low Setup Time (relative to clock) | 6 | | | ns |
| $t_{CSLH}$ | $\overline{\text{CS}}$ Low Hold Time (relative to clock) | 6 | | | ns |
| $t_{CSHS}$ | $\overline{\text{CS}}$ High Setup Time (relative to clock) | 6 | | | ns |
| $t_{CSHH}$ | $\overline{\text{CS}}$ High Hold Time (relative to clock) | 6 | | | ns |
| $t_{DS}$ | Data In Setup Time | 2 | | | ns |
| $t_{DH}$ | Data In Hold Time | 1 | | | ns |
| $t_{DIS}$ | Output Disable Time | | | 8 | ns |
| $t_V$ | Output Valid Time | | | 8 | ns |
| $t_{OH}$ | Output Hold Time | 0 | | | ns |
| $t_{HLS}$ | $\overline{\text{HOLD}}$ Low Setup Time (relative to clock) | 6 | | | ns |
| $t_{HLH}$ | $\overline{\text{HOLD}}$ Low Hold Time (relative to clock) | 6 | | | ns |
| $t_{HHS}$ | $\overline{\text{HOLD}}$ High Setup Time (relative to clock) | 6 | | | ns |
| $t_{HHH}$ | $\overline{\text{HOLD}}$ High Hold Time (relative to clock) | 6 | | | ns |
| $t_{HLQZ}$ | $\overline{\text{HOLD}}$ Low to Output High-Z | | | 7 | ns |
| $t_{HHQX}$ | $\overline{\text{HOLD}}$ High to Output Low-Z | | | 7 | ns |
| $t_{WPS}$ | Write Protect Setup Time | 20 | | | ns |
| $t_{WPH}$ | Write Protect Hold Time | 100 | | | ns |
| $t_{EDPD}$ | $\overline{\text{CS}}$ High to Deep Power-Down | | | 3 | µs |
| $t_{EUDPD}$ | $\overline{\text{CS}}$ High to Ultra-Deep Power-Down | | | 3 | µs |
| $t_{SWTERM}$ | Resume from Terminate Command Time | | | 50 | µs |
| $t_{SWRST}$ | Resume from Software Reset Time | | | 200 | µs |
| $t_{RUDPD}$ [2] | Resume form Ultra-Deep Power-Down Time | | 160 | 200 | µs |
| $t_{RDPD}$ | Resume from Deep Power-Down Time | | | 35 | µs |

1. Not 100% tested (value guaranteed by design and characterization).
2. The maximum spec is valid for UDPD hold time (duration of Ultra-Deep Power-Down state) of >550 ms. For the shorter UDPD hold times, $T_{RUDPD}$ might reach 1200 µs under some conditions.

## 7.6 Program and Erase Characteristics

| Symbol | Parameter | | 1.65V - 3.6V | | | 2.7V - 3.6V | | | Units |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Typ [1] | Max [2] | Min | Typ [1] | Max [2] | |
| $t_{RWM}$ | Read-Modify-Write Time | | | 13.5 | 95 | | 13 | 95 | ms |
| $t_{PP}$ | Page Program Time (256 Bytes) | | | 2.5 | 7 | | 2.5 | 7 | ms |
| $t_{BP}$ [3] | Byte Program Time | | | 30 | 50 | | 30 | 50 | µs |
| $t_{PE}$ | Page Erase Time | | | 10.2 | 100 | | 10.2 | 100 | ms |
| $t_{BLKE}$ | Block Erase Time | 4 kB | | 45 | 130 | | 45 | 130 | ms |
| | | 32 kB | | 310 | 830 | | 310 | 830 | ms |
| | | 64 kB | | 600 | 1600 | | 600 | 1600 | ms |
| $t_{CHPE}$ | Chip Erase Time (16M density) | | | 20 | 37 | | 20 | 37 | sec |
| $t_{SUS}$ | Suspend Time (Program) | | | | 50 | | | 50 | µs |
| | Suspend Time (Erase) | | | | 50 | | | 50 | µs |
| $t_{RES}$ | Resume Time (Program) | | | 16 | 20 | | 16 | 20 | µs |
| | Resume Time (Erase) | | | 8 | 10 | | 8 | 10 | µs |
| $t_{OTPP}$ | OTP Security Register Program Time (<= 10K) | | | 5.5 | 15 | | 5.5 | 15 | ms |
| $t_{WRSR}$ | Write Status Register Time | | | 5.5 | 8.5 | | 5.5 | 8.5 | ms |

1. Typical values measured at 1.8 V at 25 ºC for 1.65 V - 3.6 V, and at 3.3 V at 25 ºC for 2.7 V - 3.6 V.
2. Unless otherwise specified, maximum is worst case measurement at cycling conditions after 100k cycles.
3. Program time after the first byte varies.

## 7.7 Power-On Timing

When power is first applied to the device, or when recovering from a reset condition, the output pin (SO) is in a high impedance state, and a high-to-low transition on the $\overline{CS}$ pin is required to start a valid command. The SPI mode (Mode 3 or Mode 0) is automatically selected on every falling edge of $\overline{CS}$ by sampling the inactive clock state.

As the device initializes, there is a transient current demand. The system needs to be capable of providing this current to ensure correct initialization. During power-up, the device must not be accessed for at least the minimum $t_{VCSL}$ time after the supply voltage reaches the minimum VCC level (VCC min). While the device is being powered-up, the internal Power-On Reset (POR) circuitry keeps the device in a reset mode until the supply voltage rises above the minimum VCC. During this time, all operations are disabled, and the device does not respond to any commands. (The $t_{VCSL}$ time is measured from when VCC reaches VCC min.)

If Power-On-Reset (POR) has not been properly completed by the end of $t_{VCSL}$, the execution of a JEDEC Reset sequence restarts the POR process. This ensures the device can complete the POR sequence, even if some aspect of system Power-On voltage ramp-up causes the POR to not initiate or complete correctly.

Figure 33 shows the AC timing during power-up.

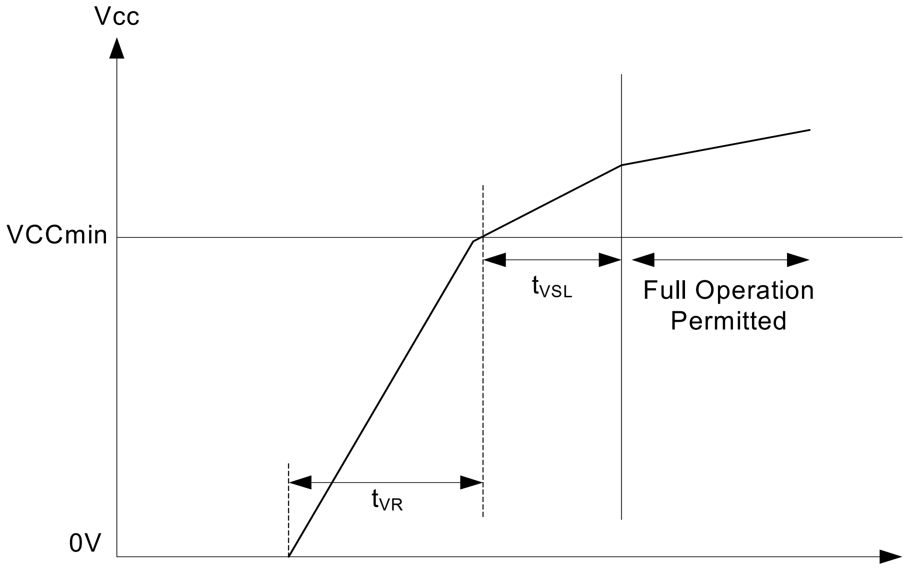



**Figure 33. AC Timing During Device Power Up**

**Table 35. Power On Timing Requirements**

| Symbol | Parameter | 1.65V - 3.6V | | | Units |
|---|---|---|---|---|---|
| | | Min | Typ | Max | |
| $t_{VSL}$ | Minimum $V_{CC}$ to full operation permitted | | | 200 | µs |
| $t_{VR}$ | $V_{CC}$ rise time (from 0V to VCCmin) | | 1 | 6000 [1] | µs/V |
| $t_{PWD}$ | $V_{CC}$ brown-out low time | 300 | | | µs |
| $V_{PWDMAX}$ | Maximum $V_{CC}$ brown-out | | | 0.2 | V |

1. 30000 µs/V at -10 °C to 85 °C.

Figure 34 shows the AC power-up timing after a brown-out condition.

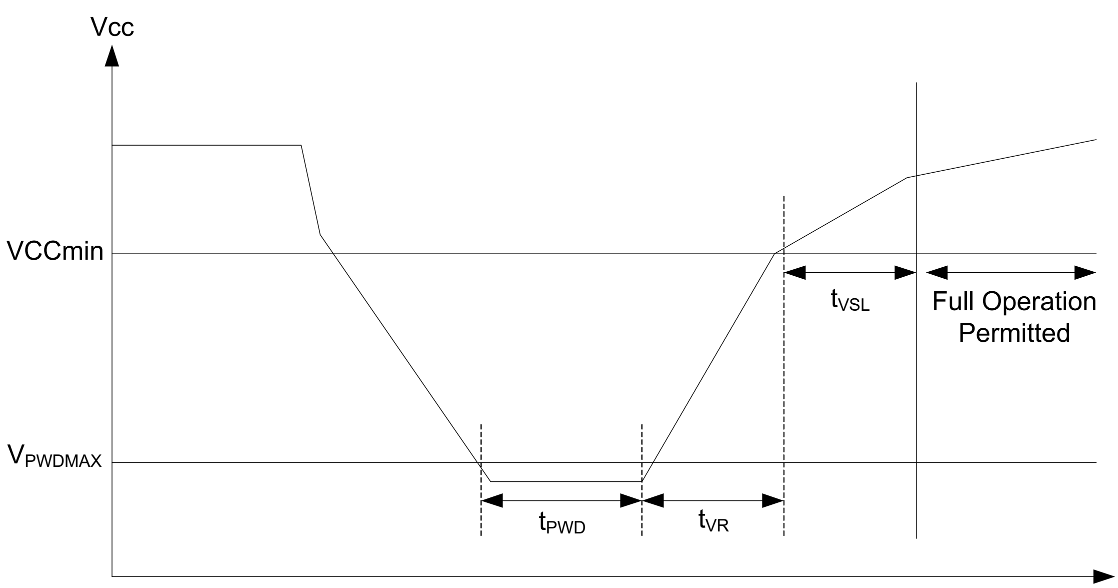



**Figure 34. AC Power-On Timing After a Brown-Out**

## 7.8 AC Timing Diagrams

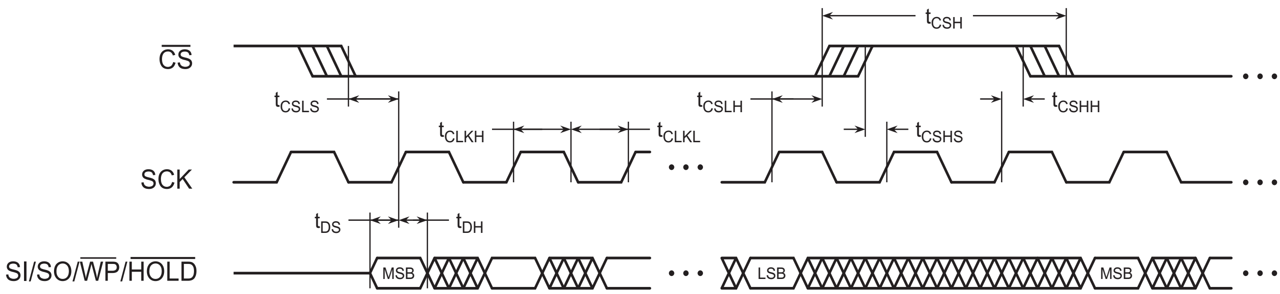



**Figure 35. Serial Input Timing**

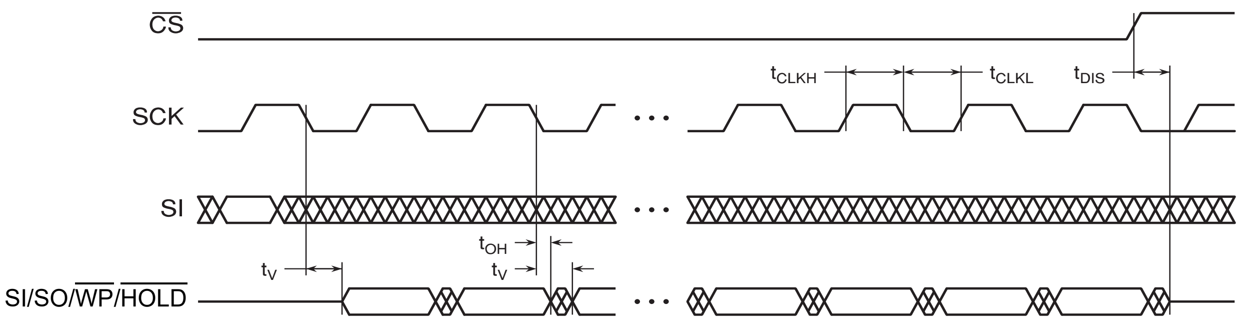



**Figure 36. Serial Output Timing**

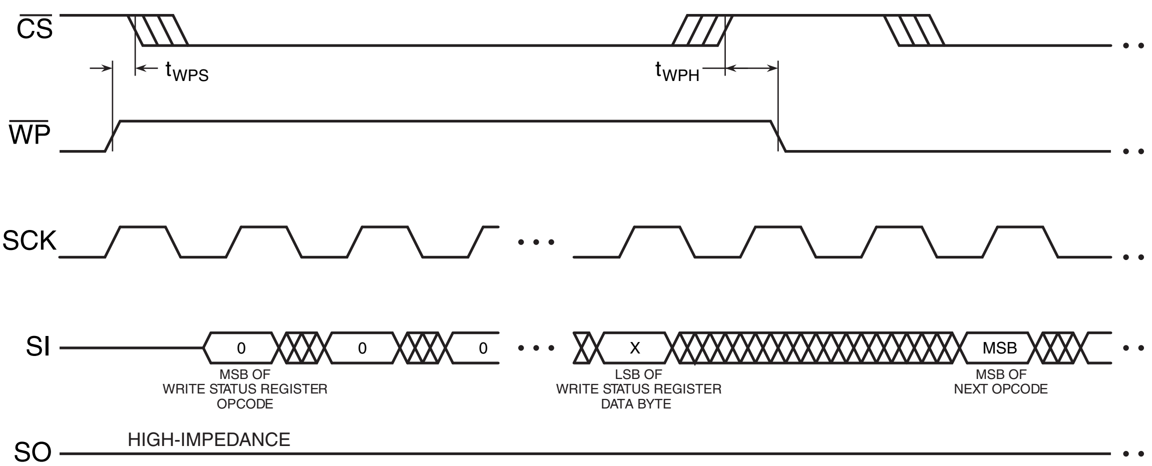



**Figure 37. $\overline{\text{WP}}$ Timing for Write Status Register Command**

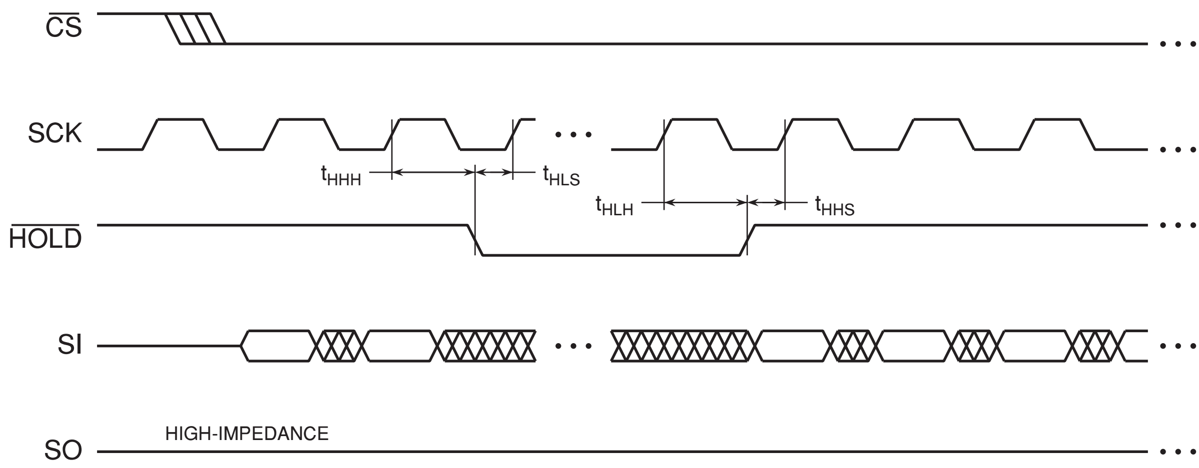



**Figure 38. HOLD Timing – Serial Input**

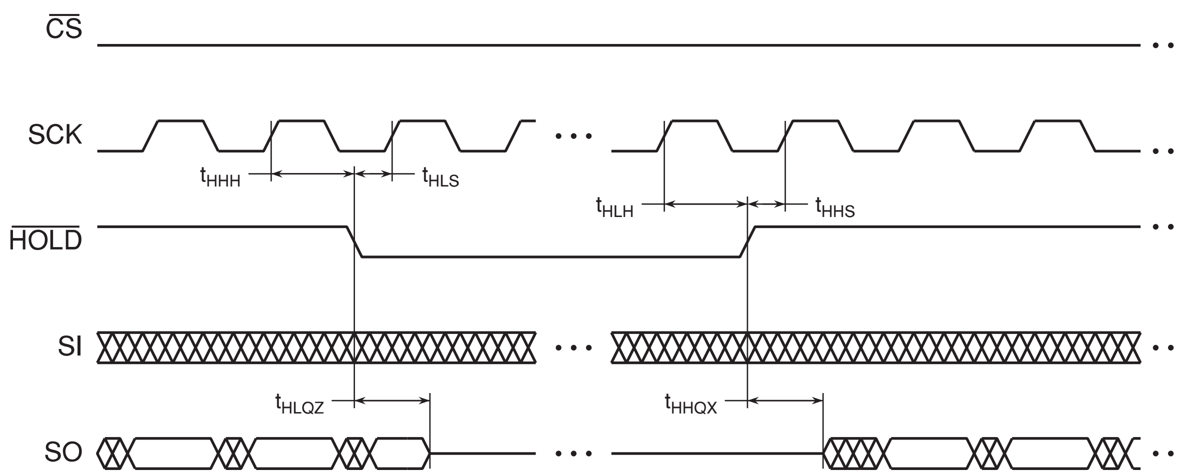



**Figure 39. HOLD Timing – Serial Output**

# 8. Ordering Information

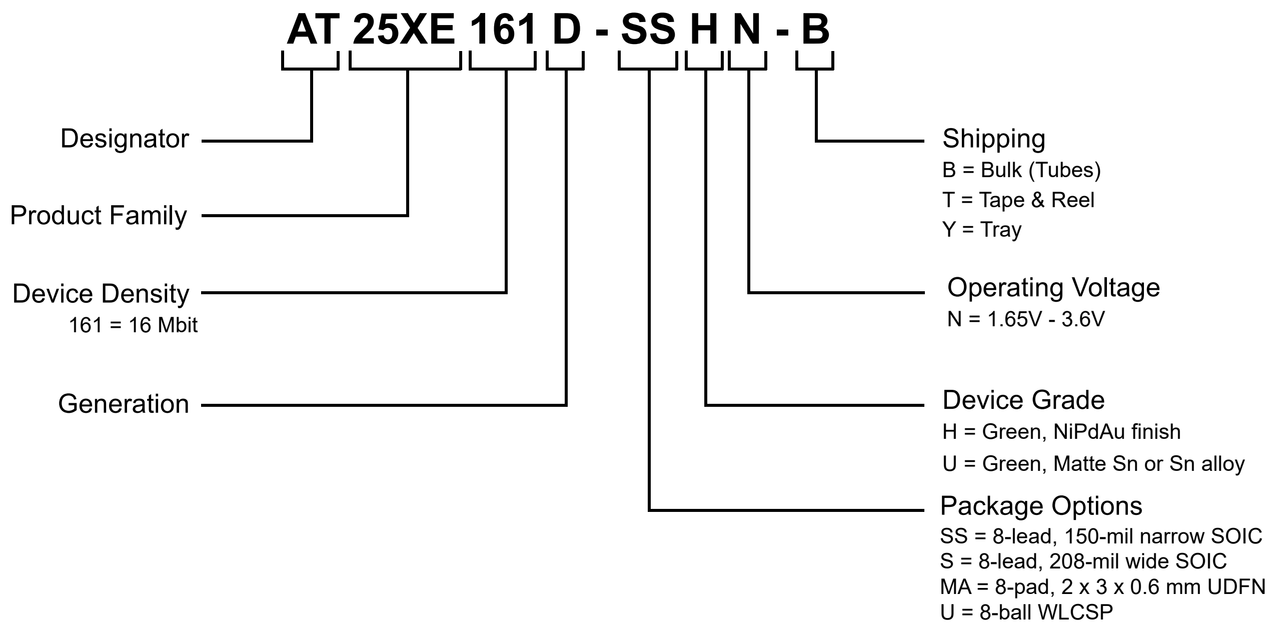


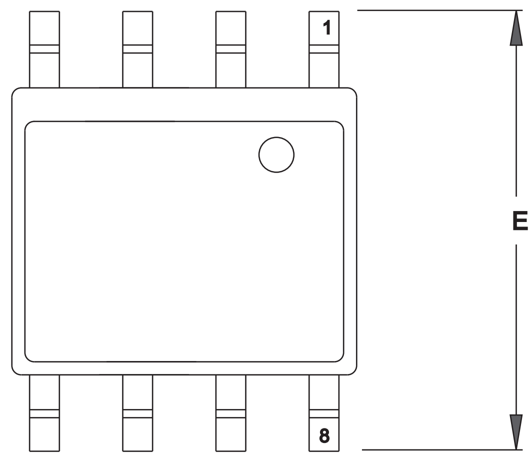

**Table 36. Valid Ordering Codes**

| Valid Part Number | Description | Lead Finish | Operating Voltage | Temperature Range |
|---|---|---|---|---|
| AT25XE161D-SSHN-B<br>AT25XE161D-SSHN-T | 8-lead, 150-mil Narrow Plastic Gull Wing Small Outline Package (JEDEC SOIC) | NiPdAu | 1.65 V to 3.6 V | Industrial (-40°C to +85°C) |
| AT25XE161D-SHN-B<br>AT25XE161D-SHN-T | 8-lead, 208-mil Wide Plastic Gull Wing Small Outline Package (EIAJ SOIC) | | | |
| AT25XE161D-MAHN-T | 8-pad, 2 x 3 x 0.6 mm, Thermally Enhanced Plastic Ultra Thin Dual Flat No Lead Package (UDFN) | | | |
| AT25XE161D-UUN-T | 8-ball, 3 x 2 x 3 Ball Matrix | Green, Matte Sn, or Sn Alloy | | |
| AT25XE161D-DWF [1] | Die in Wafer Form | – | | |

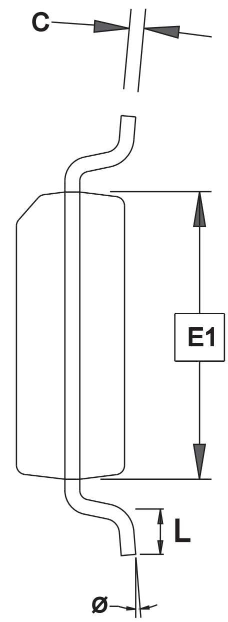
1. For more information on Die Wafer Form, contact Renesas Electronics.

# 9. Packaging Information

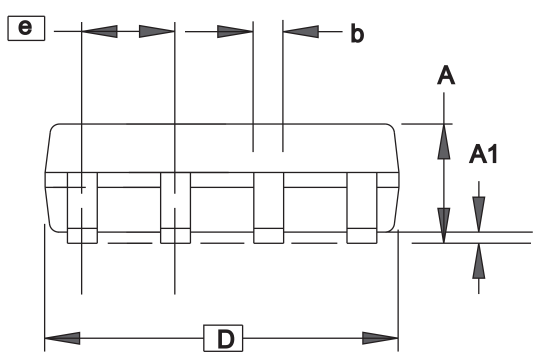
## 9.1 8-Lead, 150-mil Narrow Body JEDEC SOIC




TOP VIEW




END VIEW



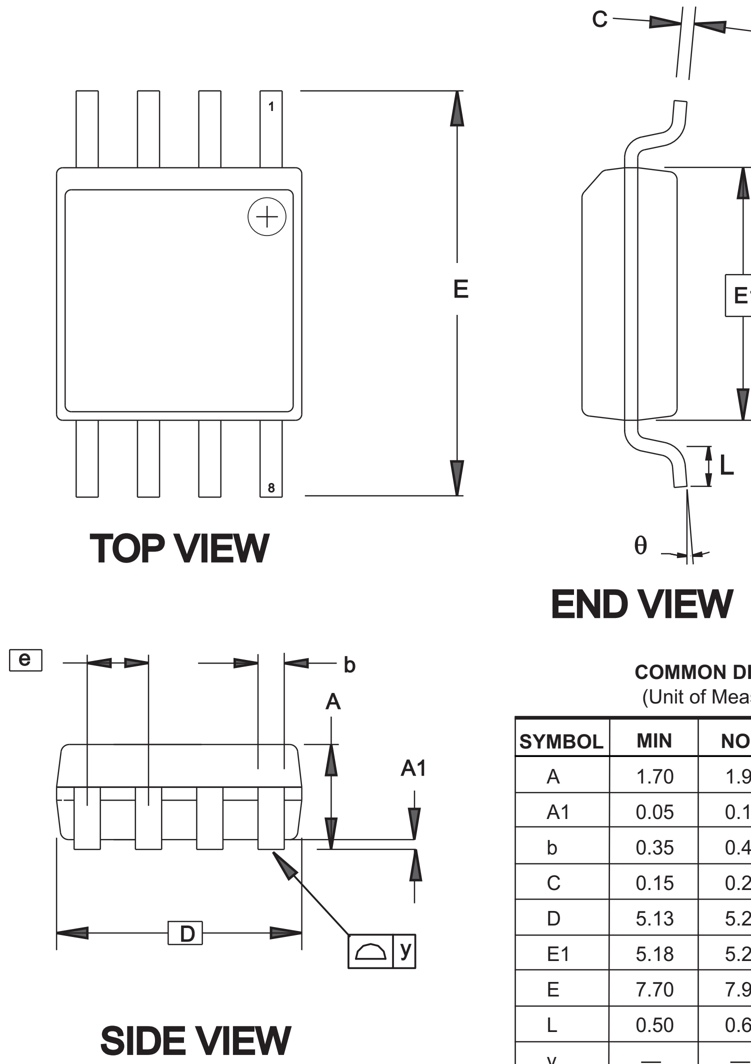

SIDE VIEW

Notes: This drawing is for general information only. See JEDEC Drawing MS-012, Variation AA, for proper dimensions, tolerances, datums, etc.

**COMMON DIMENSIONS**
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 1.35 | 1.60 | 1.75 | |
| A1 | 0.10 | 0.18 | 0.25 | |
| b | 0.31 | 0.43 | 0.51 | |
| C | 0.17 | 0.22 | 0.25 | |
| D | 4.80 | 4.83 | 5.05 | |
| E1 | 3.81 | 3.90 | 3.99 | |
| E | 5.79 | 6.00 | 6.20 | |
| e | | 1.27 BSC | | |
| L | 0.40 | 0.60 | 1.27 | |
| Ø | 0° | 3.75 | 8° | |

## 9.2 8-Lead, 208-mil Wide Body EIAJ SOIC

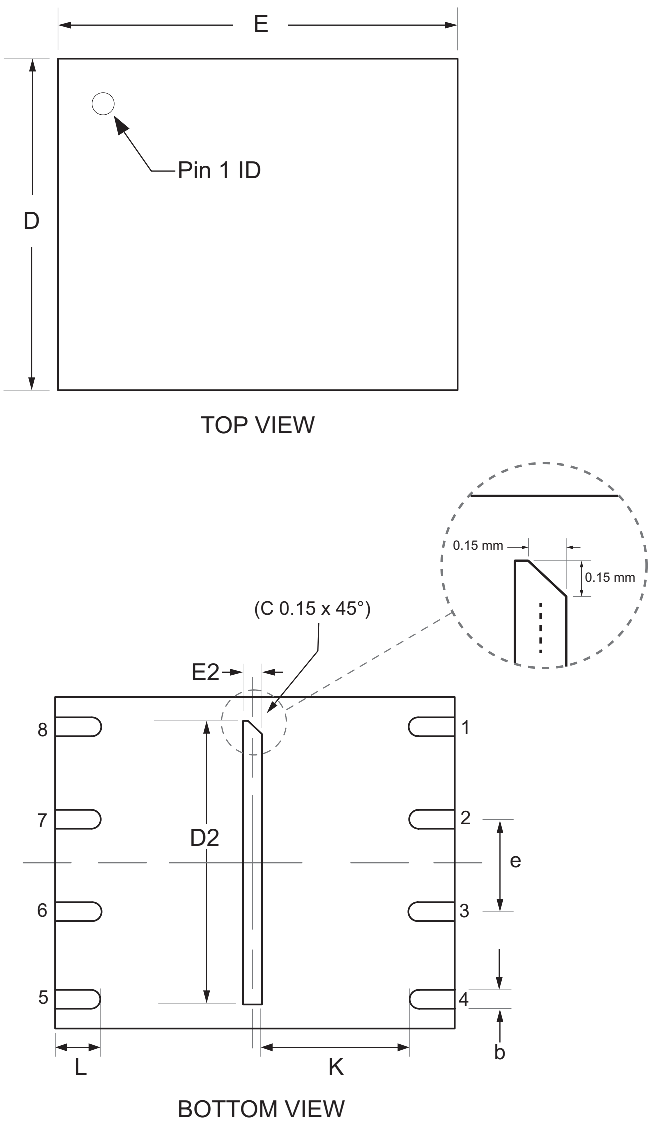


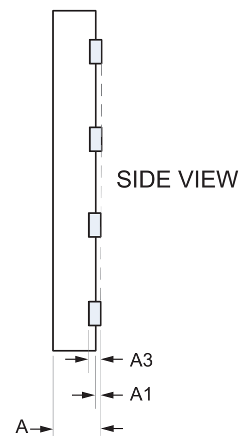

**COMMON DIMENSIONS**
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 1.70 | 1.95 | 2.16 | |
| A1 | 0.05 | 0.15 | 0.25 | |
| b | 0.35 | 0.42 | 0.48 | 4 |
| C | 0.15 | 0.20 | 0.35 | 4 |
| D | 5.13 | 5.28 | 5.38 | |
| E1 | 5.18 | 5.23 | 5.40 | 2 |
| E | 7.70 | 7.90 | 8.26 | |
| L | 0.50 | 0.65 | 0.85 | |
| y | — | — | 0.10 | |
| θ | 0° | — | 8° | |
| e | 1.27 BSC | | | 3 |

Notes:
1. This drawing is for general information only; refer to EIAJ Drawing EDR-7320 for additional information.
2. Mismatch of the upper and lower dies and resin burrs aren't included.
3. Determines the true geometric position.
4. Values b,C apply to plated terminal. The standard thickness of the plating layer shall measure between 0.007 to .021 mm.

## 9.3 8-Pad, 2 x 3 x 0.6 mm UDFN



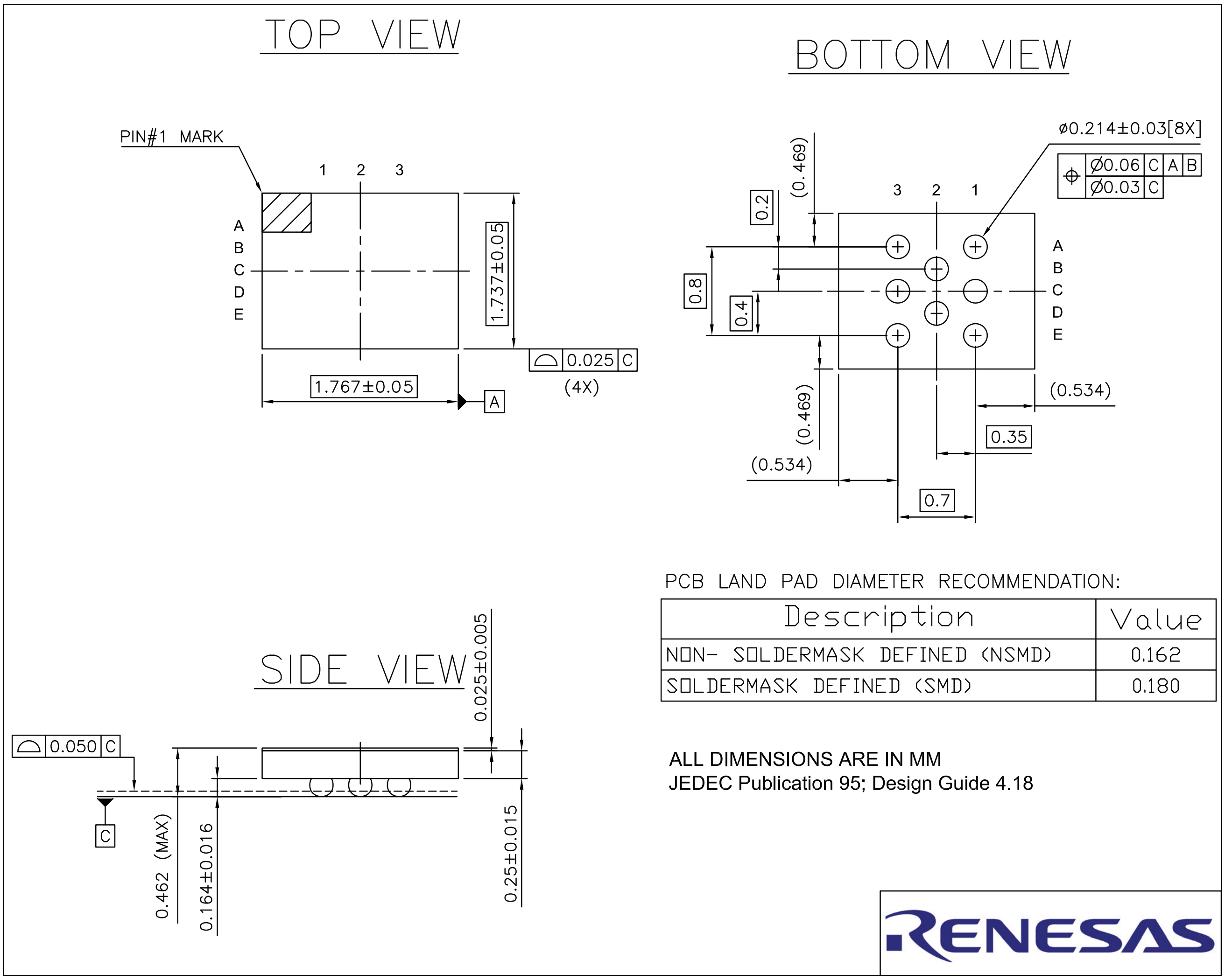




COMMON DIMENSIONS
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX |
|---|---|---|---|
| A | 0.50 | 0.55 | 0.60 |
| A1 | 0.00 | 0.02 | 0.05 |
| A3 | | 0.150 REF | |
| b | 0.20 | 0.25 | 0.30 |
| D | | 2.00 BSC | |
| D2 | 1.55 | 1.60 | 1.65 |
| E | | 3.00 BSC | |
| e | | 0.50 BSC | |
| E2 | 0.15 | 0.20 | 0.25 |
| K | 0.20 | | |
| L | 0.40 | 0.45 | 0.50 |

Notes: 1. This package conforms to JEDEC reference MO-229, Saw Singulation.
2. The terminal #1 ID is a Laser-marked feature.

## 9.4 8-ball 3 x 2 x 3 WLCSP




PCB LAND PAD DIAMETER RECOMMENDATION:

| Description | Value |
|---|---|
| NON- SOLDERMASK DEFINED (NSMD) | 0.162 |
| SOLDERMASK DEFINED (SMD) | 0.180 |

ALL DIMENSIONS ARE IN MM
JEDEC Publication 95; Design Guide 4.18

# 10. Revision History

| Revision | Date | Change History |
|---|---|---|
| A | 3/2018 | Initial release of Fusion 4 Mbit data sheet. |
| B | 10/2019 | Conversion to new Adesto template.<br>Updated Section 7.44, Read SFDP (5Ah).<br>Removed 5 x 6 UDFN package.<br>Updated text for steps 1, 2, and 3 in Section 5.4, Power Down Considerations.<br>Updated encoding of the DRV[1:0] bits in Table 6-6, Status Register 3.<br>Added Resume from Ultra-Deep Power Down mode (ABh) instruction to Table 7-10, Command Listing.<br>Updated text in Section 7.33, Deep Power Down.<br>Updated text in Section 7.34, Resume from Deep Power Down.<br>Updated text in Section 7.35, Ultra-Deep Power Down, to include mention of the ABh command as a way to Resume from Ultra-Deep Power Down.<br>Updated Section 7.39.3, Transfer Sequence, for the Quad Read Manufacturer/Device ID command in Section 7.39.<br>Changed $t_{UDPD}$ timing parameter acronym to $t_{RUDPD}$ in Section 8.5, AC Characteristics.<br>Changed $t_{DPD}$ timing parameter acronym to $t_{RDPD}$ in Section 8.5, AC Characteristics.<br>Added footnote to 8-WLCSP package in the tables of Section 9.1, Ordering Code Detail.<br>Updated dimension E of 8S2 package in Section 10.2.<br>Added Section 11, Glossary.<br>Removed R-M-W flow diagram in Section 5.11.<br>Updated memory protection tables in Section 5.3.1.<br>Updated X and Y dimensions of 8-ball WLCSP package.<br>Added deep power down timing diagram in Section 7.33.<br>Added resume from deep power down timing diagram in Section 7.34.<br>Added ultra-deep power down timing diagram in Section 7.35.<br>Removed references to Dual Input and Quad Input modes.<br>Updated product ID values in Table 7-26.<br>Updated WLCSP packaging options in Table 9-1.<br>Changed bit 1 of Status Register 5 from STPE to TERE. Modified all references throughout document as necessary. |
| C | 3/2020 | Updated Figure 7-9, Read JEDEC ID.<br>Deleted footnote about 50,000 cycles in Section 8.3, DC Characteristics.<br>Updated numbers in Section 8.6, Program and Erase Characteristics.<br>Removed bullet item on first page: Automatic checking and reporting of erase/program failures.<br>Added footnotes in Table 5-4, Device Block Protection Map, CMPRT = 1. |
| D | 6/2020 | Updated Section 8.3, DC Characteristics.<br>Updated Section 8.4, Maximum Operating Frequencies.<br>Updated Section 8.5, AC Characteristics.<br>Updated Section 8.6, Program and Erase Characteristics.<br>Added 2.7V - 3.6V voltage option to Section 8.6.<br>Updated Section 10.3, 8MA3 2 x 3 UDFN package drawing.<br>Changed designation from ADVANCED to PRELIMINARY.<br>Added footnote in Table 7.4, Command Behavior During Program/Erase or Program/Erase Suspend Operations.<br>Added 2.7V column in Table 7.2, Frequency and Number of Dummy Clocks Based on Command Type.<br>Added 2.7V column in Section 8.6, Program and Erase Characteristics.<br>Updated frequency values in 1.65V column of Table 7.2.<br>Changed 104 MHz to 108 MHz throughout document.<br>Updated 8-WLCSP package drawing.<br>Added footnote in Section 8.4, Maximum Clock Frequencies<br>Updated content on Mode bits in Section 5.5, Quad I/O Operation. |

| Revision | Date | Change History |
|---|---|---|
| E | 7/2020 | Added DWF entry to Table 9-1, Valid Ordering Codes<br>Added RMW (0Ah) command to bullet list in Section 7.18, Write Enable<br>Added RMW (0Ah) command to bullet list in Section 7.19, Write Disable<br>Updated Table 7-2, Frequency and Number of Dummy Clocks Based on Command Type<br>Updated frequency values in 1.65V column of Table 7.2.<br>Updated content on Mode bits in Section 5.5, Quad I/O Operation.<br>Updated $t_{BP}$ time in Section 8.6, Program and Erase Characteristics. |
| F | 01/2021 | Changed Table 8-19 to indicate temperature dependency.<br>Updated Table 7-2 and added Table 7-3 in Section 7.4, XiP Mode Read Command.<br>Updated Section 8.3, Maximum Clock Frequencies.<br>Changed value in second footnote of Section 8.5, AC Characteristics - All Other Parameters.<br>Removed footnote 4 from Section 8.3, DC Characteristics. |
| G | 06/2021 | Applied new corporate template to document.<br>Removed Preliminary designation.<br>Made changes to Tables 22 and 23, Tables in sections 7.3 through 7.6, and Table 39. |
| H | 08/2021 | Updated UDFN package drawing in Section 9.3. |
| I | 09/2021 | Replaced the paragraph for Accessing a Suspended Area at the end of Section 6.12.5. |
| J | 11/2021 | Updated package drawings in Sections 9.3 and 9.4 |
| K | 06/2022 | Applied new corporate template to document.<br>Added physical block size information to Section 1, Product Overview.<br>Updated values in Sections 7.3, 7.4, and 7.6.<br>Added the following to Section 4.8.3: “Note that in the AT25XE161D device all lock bits are set by default, making all memory locations protected.” and added: “Note that in the AT25XE161D the factory default setting is the Standard Memory Protection scheme; therefore, all memory locations are unprotected.”<br>Updated UDFN POD in Section 9.3.<br>Added the following to the end of Section 6.12.5: “Note: A read operation from a physical block that includes a suspended area might provide unreliable data. For the definition of the physical block and for the techniques to ensure high data integrity, see Application Note AN-500.”<br>Added the following sentence to the description of $\overline{CS}$ in Section 3: “To ensure correct power-up sequencing, it is recommended to add a 10k Ohm pull-up resistor from CS to VCC. This ensures CS ramps together with VCC during power-up.”<br>Revised the descriptions of $\overline{HOLD}/\overline{RESET}/IO_3$ and $\overline{WP}$ pins in Table 1. |
| L | 09/2023 | Corrected the WLCSP POD in Section 9.4. |
| M | 03/2024 | Changed order and text of some bullets in the Features list (front page).<br>Changed the text of Section 1, Product Overview.<br>Updated wording for opcode 75h (at the end of Section 6.12.5). See application note AN500 for operational guidance on implementing suspend and resume operations.<br>Moved physical block information from Section 1 to Section 4.7.<br>Updated Figure 16.<br>Updated the PODs in Section 9. |

## IMPORTANT NOTICE AND DISCLAIMER



### Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

### Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

### Trademarks