# AT25XE021A

## 2-Mbit, 1.65V Minimum SPI Serial Flash Memory with Dual-I/O Support

**PRELIMINARY DATASHEET**

## Features

- Single 1.65V - 3.6V Supply
- Serial Peripheral Interface (SPI) Compatible
  - Supports SPI Modes 0 and 3
  - Supports Dual-I/O Operation
- 70MHz Maximum Operating Frequency
  - Clock-to-Output ($t_V$) of 6 ns
- Flexible, Optimized Erase Architecture for Code + Data Storage Applications
  - Uniform 4-Kbyte Block Erase
  - Uniform 32-Kbyte Block Erase
  - Uniform 64-Kbyte Block Erase
  - Full Chip Erase
- Hardware Controlled Locking of Protected Sectors via $\overline{WP}$ Pin
- 128-byte, One-Time Programmable (OTP) Security Register
  - 64 bytes factory programmed with a unique identifier
  - 64 bytes user programmable
- Flexible Programming
  - Byte/Page Program (1 to 256 Bytes)
  - Dual-Input Byte/Page Program (1 to 256 Bytes)
  - Sequential Program Mode Capability
- Fast Program and Erase Times
  - 2ms Typical Page Program (256 Bytes) Time
  - 45ms Typical 4-Kbyte Block Erase Time
  - 360ms Typical 32-Kbyte Block Erase Time
  - 720ms Typical 64-Kbyte Block Erase Time
- Automatic Checking and Reporting of Erase/Program Failures
- Software Controlled Reset
- JEDEC Standard Manufacturer and Device ID Read Methodology
- Low Power Dissipation
  - 200nA Ultra Deep Power Down current (Typical)
  - 4.5µA Deep Power-Down Current (Typical)
  - 25uA Standby current (Typical)
  - 3mA Active Read Current (Typical)
- Endurance: 100,000 Program/Erase Cycles
- Data Retention: 20 Years
- Complies with Full Industrial Temperature Range
- Industry Standard Green (Pb/Halide-free/RoHS Compliant) Package Options
  - 8-lead SOIC (150-mil)
  - 8-pad Ultra Thin DFN (2 x 3 x 0.6 mm)
  - 8-pad Ultra Thin DFN (5 x 6 x 0.6 mm)
  - 8-lead TSSOP Package

# 1. Description

The Adesto® AT25XE021A is a serial interface Flash memory device designed for use in a wide variety of high-volume consumer based applications in which program code is shadowed from Flash memory into embedded or external RAM for execution. The flexible erase architecture of the AT25XE021A, with its page erase granularity it is ideal for data storage as well, eliminating the need for additional data storage devices.

The erase block sizes of the AT25XE021A have been optimized to meet the needs of today's code and data storage applications. By optimizing the size of the erase blocks, the memory space can be used much more efficiently. Because certain code modules and data storage segments must reside by themselves in their own erase regions, the wasted and unused memory space that occurs with large sectored and large block erase Flash memory devices can be greatly reduced. This increased memory space efficiency allows additional code routines and data storage segments to be added while still maintaining the same overall device density.

The device also contains a specialized OTP (One-Time Programmable) Security Register that can be used for purposes such as unique device serialization, system-level Electronic Serial Number (ESN) storage, locked key storage, etc.

Specifically designed for use in many different systems, the AT25XE021A supports read, program, and erase operations with a wide supply voltage range of 1.65V to 3.6V. No separate voltage is required for programming and erasing.

# 2. Pin Descriptions and Pinouts

**Table 2-1. Pin Descriptions**

| Symbol | Name and Function | Asserted State | Type |
|---|---|---|---|
| $\overline{CS}$ | **CHIP SELECT:** Asserting the $\overline{CS}$ pin selects the device. When the $\overline{CS}$ pin is deasserted, the device will be deselected and normally be placed in standby mode (not Deep Power-Down mode), and the SO pin will be in a high-impedance state. When the device is deselected, data will not be 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 will not enter the standby mode until the completion of the operation. | Low | Input |
| SCK | **SERIAL CLOCK:** This pin is used to provide a clock to the device and is used to control 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:** The SI pin is used to shift 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 Dual-Output Read commands, the SI Pin becomes an output pin ($I/O_0$) in conjunction with other pins to allow two bits of data on ($I/O_{1-0}$) to be clocked out on every falling edge of SCK.<br>To maintain consistency with the SPI nomenclature, the SI ($I/O_0$) pin will be referenced as the SI pin unless specifically addressing the Dual-I/O modes in which case it will be referenced as $I/O_0$.<br>Data present on the SI pin will be ignored whenever the device is deselected ($\overline{CS}$ is deasserted). | - | Input/ Output |

**Table 2-1. Pin Descriptions (Continued)**

| Symbol | Name and Function | Asserted State | Type |
|---|---|---|---|
| SO ($I/O_1$) | **SERIAL OUTPUT:** The SO pin is used to shift data out from the device. Data on the SO pin is always clocked out on the falling edge of SCK.<br>With the Dual-Output Read commands, the SO Pin remains an output pin ($I/O_1$) in conjunction with other pins to allow two bits of data on ($I/O_{1-0}$) to be clocked out on every falling edge of SCK.<br>To maintain consistency with the SPI nomenclature, the SO ($I/O_1$) pin will be referenced as the SO pin unless specifically addressing the Dual-I/O modes in which case it ise referenced as $I/O_1$.<br>The SO pin will be in a high-impedance state whenever the device is deselected ($\overline{CS}$ is deasserted). | - | Input/ Output |
| $\overline{WP}$ | **WRITE PROTECT:** The $\overline{WP}$ pin controls the hardware locking feature of the device. Please refer to "Protection Commands and Features" on page 17 for more details on protection features and the $\overline{WP}$ pin.<br>The $\overline{WP}$ pin is internally pulled-high and may be left floating if hardware controlled protection will not be used. However, it is recommended that the $\overline{WP}$ pin also be externally connected to $V_{CC}$ whenever possible. | Low | Input |
| $\overline{HOLD}$ | **HOLD:** The $\overline{HOLD}$ pin is used to temporarily pause serial communication without deselecting or resetting the device. While the $\overline{HOLD}$ pin is asserted, transitions on the SCK pin and data on the SI pin will be ignored, and the SO pin will be in a high-impedance state.<br>The $\overline{CS}$ pin must be asserted, and the SCK pin must be in the low state in order for a Hold condition to start. A Hold condition pauses serial communication only and does not have an effect on internally self-timed operations such as a program or erase cycle. Please refer to "Hold" on page 35 for additional details on the Hold operation.The $\overline{HOLD}$ pin is internally pulled-high and may be left floating if the Hold function will not be used. However, it is recommended that the $\overline{HOLD}$ pin also be externally connected to $V_{CC}$ whenever possible. | Low | Input |
| $V_{CC}$ | **DEVICE POWER SUPPLY:** The $V_{CC}$ pin is used to supply the source voltage to the device.<br>Operations at invalid $V_{CC}$ voltages may produce spurious results and should not be attempted. | - | Power |
| GND | **GROUND:** The ground reference for the power supply. GND should be connected to the system ground. | - | Power |

## Table 2-2. Pinouts

**Figure 2-1. 8-SOIC Top View**

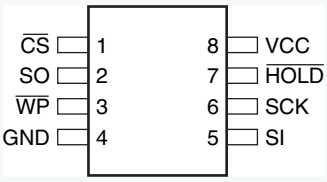



**Figure 2-2. 8-TSSOP Top View**

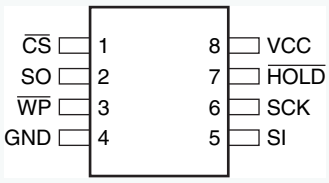
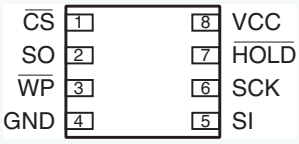
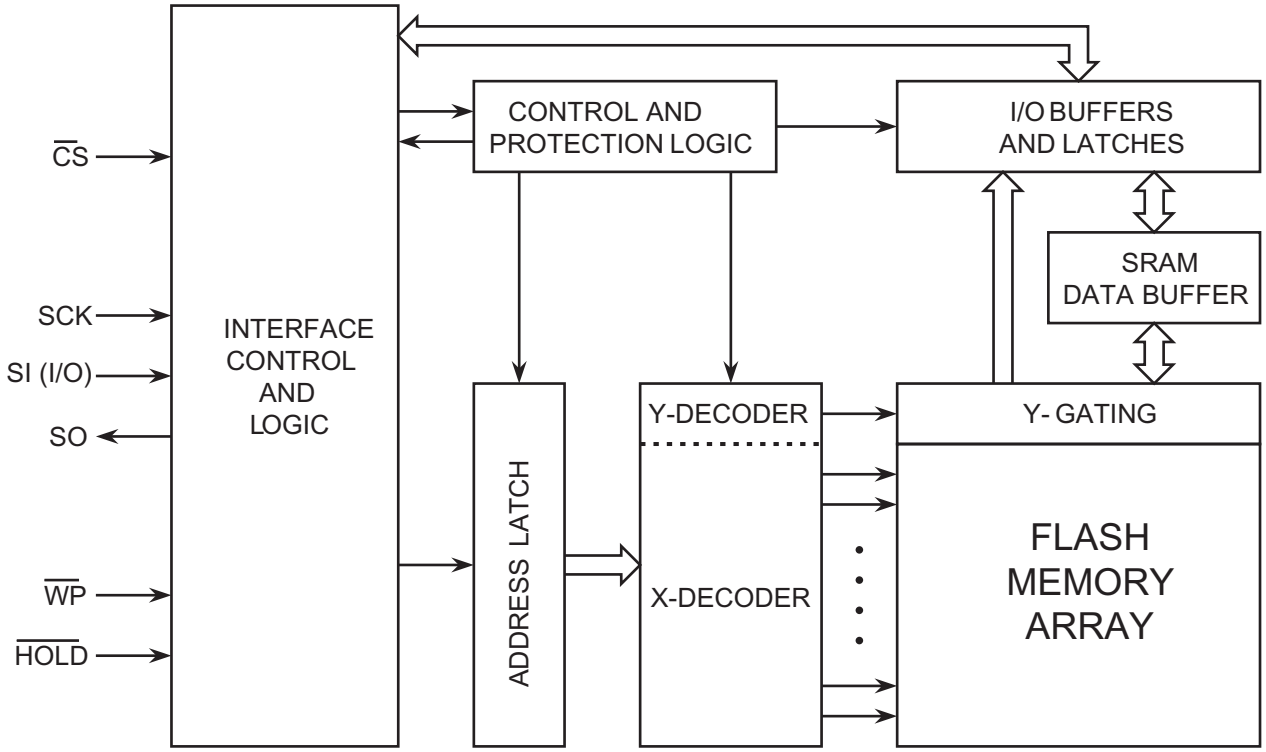



**Figure 2-3. 8-UDFN (Top View)**



# 3. Block Diagram

**Figure 3-1. Block Diagram**



# 4. Memory Array

To provide the greatest flexibility, the memory array of the AT25XE021A can be erased in three levels of granularity including a full chip erase. 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. The Memory Architecture Diagram illustrates the breakdown of each erase level.

**Figure 4-1. Memory Architecture Diagram**

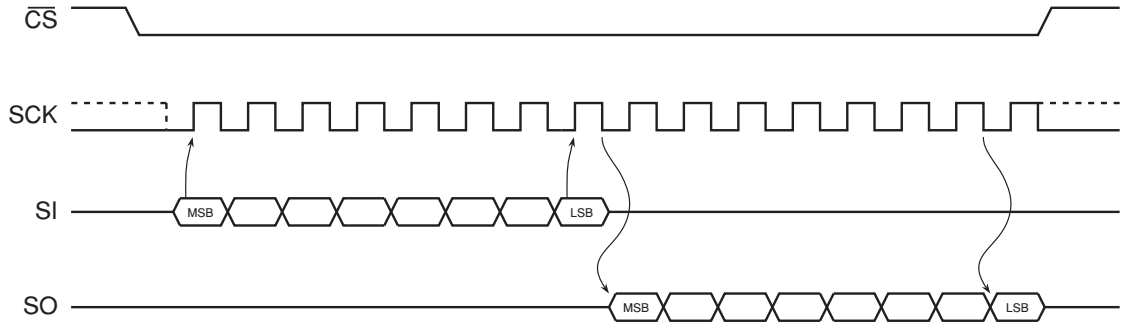
| | Block Erase Detail | | | | Page Program Detail | |
|---|---|---|---|---|---|---|
| Internal Sectoring for Sector Protection Function | 64KB Block Erase (D8h Command) | 32KB Block Erase (52h Command) | 4KB Block Erase (20h Command) | Block Address Range | 1-256 Byte Page Program (02h Command) | Page Address Range |
| 64KB (Sector 3) | 64KB | 32KB | 4KB | 03FFFFh – 03F000h | 256 Bytes | 03FFFFh – 03FF00h |
| | | | 4KB | 03EFFFh – 03E000h | 256 Bytes | 03FEFFh – 03FE00h |
| | | | 4KB | 03DFFFh – 03D000h | 256 Bytes | 03FDFFh – 03FD00h |
| | | | 4KB | 03CFFFh – 03C000h | 256 Bytes | 03FCFFh – 03FC00h |
| | | | 4KB | 03BFFFh – 03B000h | 256 Bytes | 03FBFFh – 03FB00h |
| | | | 4KB | 03AFFFh – 03A000h | 256 Bytes | 03FAFFh – 03FA00h |
| | | | 4KB | 039FFFh – 039000h | 256 Bytes | 03F9FFh – 03F900h |
| | | | 4KB | 038FFFh – 038000h | 256 Bytes | 03F8FFh – 03F800h |
| | | 32KB | 4KB | 037FFFh – 037000h | 256 Bytes | 03F7FFh – 03F700h |
| | | | 4KB | 036FFFh – 036000h | 256 Bytes | 03F6FFh – 03F600h |
| | | | 4KB | 035FFFh – 035000h | 256 Bytes | 03F5FFh – 03F500h |
| | | | 4KB | 034FFFh – 034000h | 256 Bytes | 03F4FFh – 03F400h |
| | | | 4KB | 033FFFh – 033000h | 256 Bytes | 03F3FFh – 03F300h |
| | | | 4KB | 032FFFh – 032000h | 256 Bytes | 03F2FFh – 03F200h |
| | | | 4KB | 031FFFh – 031000h | 256 Bytes | 03F1FFh – 03F100h |
| | | | 4KB | 030FFFh – 030000h | 256 Bytes | 03F0FFh – 03F000h |
| 64KB (Sector 2) | 64KB | 32KB | 4KB | 02FFFFh – 02F000h | 256 Bytes | 03EFFFh – 03EF00h |
| | | | 4KB | 02EFFFh – 02E000h | 256 Bytes | 03EEFFh – 03EE00h |
| | | | 4KB | 02DFFFh – 02D000h | 256 Bytes | 03EDFFh – 03ED00h |
| | | | 4KB | 02CFFFh – 02C000h | 256 Bytes | 03ECFFh – 03EC00h |
| | | | 4KB | 02BFFFh – 02B000h | 256 Bytes | 03EBFFh – 03EB00h |
| | | | 4KB | 02AFFFh – 02A000h | 256 Bytes | 03EAFFh – 03EA00h |
| | | | 4KB | 029FFFh – 029000h | 256 Bytes | 03E9FFh – 03E900h |
| | | | 4KB | 028FFFh – 028000h | 256 Bytes | 03E8FFh – 03E800h |
| | | 32KB | 4KB | 027FFFh – 027000h | ⋮ | |
| | | | 4KB | 026FFFh – 026000h | | |
| | | | 4KB | 025FFFh – 025000h | | |
| | | | 4KB | 024FFFh – 024000h | 256 Bytes | 0017FFh – 001700h |
| | | | 4KB | 023FFFh – 023000h | 256 Bytes | 0016FFh – 001600h |
| | | | 4KB | 022FFFh – 022000h | 256 Bytes | 0015FFh – 001500h |
| | | | 4KB | 021FFFh – 021000h | 256 Bytes | 0014FFh – 001400h |
| | | | 4KB | 020FFFh – 020000h | 256 Bytes | 0013FFh – 001300h |
| ⋮ | ⋮ | ⋮ | ⋮ | | 256 Bytes | 0012FFh – 001200h |
| | | | | | 256 Bytes | 0011FFh – 001100h |
| | | | | | 256 Bytes | 0010FFh – 001000h |
| 64KB (Sector 0) | 64KB | 32KB | 4KB | 00FFFFh – 00F000h | 256 Bytes | 000FFFh – 000F00h |
| | | | 4KB | 00EFFFh – 00E000h | 256 Bytes | 000EFFh – 000E00h |
| | | | 4KB | 00DFFFh – 00D000h | 256 Bytes | 000DFFh – 000D00h |
| | | | 4KB | 00CFFFh – 00C000h | 256 Bytes | 000CFFh – 000C00h |
| | | | 4KB | 00BFFFh – 00B000h | 256 Bytes | 000BFFh – 000B00h |
| | | | 4KB | 00AFFFh – 00A000h | 256 Bytes | 000AFFh – 000A00h |
| | | | 4KB | 009FFFh – 009000h | 256 Bytes | 0009FFh – 000900h |
| | | | 4KB | 008FFFh – 008000h | 256 Bytes | 0008FFh – 000800h |
| | | 32KB | 4KB | 007FFFh – 007000h | 256 Bytes | 0007FFh – 000700h |
| | | | 4KB | 006FFFh – 006000h | 256 Bytes | 0006FFh – 000600h |
| | | | 4KB | 005FFFh – 005000h | 256 Bytes | 0005FFh – 000500h |
| | | | 4KB | 004FFFh – 004000h | 256 Bytes | 0004FFh – 000400h |
| | | | 4KB | 003FFFh – 003000h | 256 Bytes | 0003FFh – 000300h |
| | | | 4KB | 002FFFh – 002000h | 256 Bytes | 0002FFh – 000200h |
| | | | 4KB | 001FFFh – 001000h | 256 Bytes | 0001FFh – 000100h |
| | | | 4KB | 000FFFh – 000000h | 256 Bytes | 0000FFh – 000000h |

# 5. Device Operation

The AT25XE021A is controlled by a set of instructions that are sent from a host controller, commonly referred to as the SPI Master. The SPI Master communicates with the AT25XE021A via the SPI bus which is comprised of four signal lines: Chip Select ($\overline{CS}$), Serial Clock (SCK), Serial Input (SI), and Serial Output (SO).

The SPI protocol defines a total of four modes of operation (mode 0, 1, 2, or 3) with each mode differing in respect to the SCK polarity and phase and how the polarity and phase control the flow of data on the SPI bus. The AT25XE021A supports the two most common modes, SPI Modes 0 and 3. The only difference between SPI Modes 0 and 3 is the polarity of the SCK signal when in the inactive state (when the SPI Master is in standby mode and not transferring any data). With SPI Modes 0 and 3, data is always latched in on the rising edge of SCK and always output on the falling edge of SCK.

**Figure 5-1. SPI Mode 0 and 3**




## 5.1 Dual Output Read

The AT25XE021A features a Dual-Output Read mode that allow two bits of data to be clocked out of the device every clock cycle to improve throughput. To accomplish this, both the SI and SO pins are utilized as outputs for the transfer of data bytes. With the Dual-Output Read Array command, the SI pin becomes an output along with the SO pin.

# 6. Commands and Addressing

A valid instruction 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 opcode on the SPI bus. Following the opcode, instruction dependent information such as address and data bytes would then be clocked out by the host controller. All opcode, address, and data bytes are transferred with the most-significant bit (MSB) first. An operation is ended by deasserting the $\overline{CS}$ pin.

Opcodes not supported by the AT25XE021A will be ignored by the device and no operation will be started. The device will continue to ignore any data presented on the SI pin until the start of the next operation ($\overline{CS}$ pin being deasserted and then reasserted). In addition, if the $\overline{CS}$ pin is deasserted before complete opcode and address information is sent to the device, then no operation will be performed and the device will simply return to the idle state and wait for the next operation.

Addressing of the device requires a total of three bytes of information to be sent, representing address bits A23-A0. Since the upper address limit of the AT25XE021A memory array is 03FFFFh, address bits A23-A18 are always ignored by the device.

**Table 6-1. Command Listing**

| Command | Opcode | | Clock Frequency | Address Bytes | Dummy Bytes | Data Bytes |
|---|---|---|---|---|---|---|
| **Read Commands** | | | | | | |
| Read Array | 0Bh | 0000 1011 | Up to 70MHz | 3 | 1 | 1+ |
| | 03h | 0000 0011 | Up to 25MHz | 3 | 0 | 1+ |
| Dual Output Read | 3Bh | 0011 1011 | Up to 40MHz | 3 | 1 | 1+ |
| **Program and Erase Commands** | | | | | | |
| Page Erase | 81h | 1000 0001 | Up to 70MHz | 3 | 0 | 0 |
| Block Erase (4 Kbytes) | 20h | 0010 0000 | Up to 70MHz | 3 | 0 | 0 |
| Block Erase (32 Kbytes) | 52h | 0101 0010 | Up to 70MHz | 3 | 0 | 0 |
| Block Erase (64 Kbytes) | D8h | 1101 1000 | Up to 70MHz | 3 | 0 | 0 |
| Chip Erase | 60h | 0110 0000 | Up to 70MHz | 0 | 0 | 0 |
| | C7h | 1100 0111 | Up to 70MHz | 0 | 0 | 0 |
| Byte/Page Program (1 to 256 Bytes) | 02h | 0000 0010 | Up to 70MHz | 3 | 0 | 1+ |
| Sequential Program Mode | ADh | 1010 1101 | Up to 70MHz | 3, 0 (1) | 0 | 1 |
| | AFh | 1010 1111 | Up to 70MHz | 3, 0 (1) | 0 | 1 |
| Dual-Input Byte/Page Program (1 to 256 bytes) | A2h | 1010 0010 | Up to 70MHz | 3 | 0 | 1+ |
| **Protection Commands** | | | | | | |
| Write Enable | 06h | 0000 0110 | Up to 70MHz | 0 | 0 | 0 |
| Write Disable | 04h | 0000 0100 | Up to 70MHz | 0 | 0 | 0 |
| Protect Sector | 36h | 0011 0110 | Up to 70MHz | 3 | 0 | 0 |
| Unprotect Sector | 39h | 0011 1001 | Up to 70MHz | 3 | 0 | 0 |
| Read Sector Protection Registers | 3Ch | 0011 1100 | Up to 70MHz | 3 | 0 | 1+ |
| **Security Commands** | | | | | | |
| Program OTP Security Register | 9Bh | 1001 1011 | Up to 70MHz | 3 | 0 | 1+ |
| Read OTP Security Register | 77h | 0111 0111 | Up to 70MHz | 3 | 2 | 1+ |
| **Status Register Commands** | | | | | | |
| Read Status Register | 05h | 0000 0101 | Up to 70MHz | 0 | 0 | 1+ |
| Active Status Interrupt | 25h | 0010 0101 | Up to 70MHz | 0 | 1 | 0 |
| Write Status Register Byte 1 | 01h | 0000 0001 | Up to 70MHz | 0 | 0 | 1 |
| Write Status Register Byte 2 | 31h | 0011 0001 | Up to 70MHz | 0 | 0 | 1 |
| **Miscellaneous Commands** | | | | | | |
| Reset | F0h | 1111 0000 | Up to 70MHz | 0 | 0 | 1(D0h) |

**Table 6-1. Command Listing**

| Command | Opcode | | Clock Frequency | Address Bytes | Dummy Bytes | Data Bytes |
|---|---|---|---|---|---|---|
| Read Manufacturer and Device ID | 9Fh | 1001 1111 | Up to 70MHz | 0 | 0 | 1 to 4 |
| Deep Power-Down | B9h | 1011 1001 | Up to 70MHz | 0 | 0 | 0 |
| Resume from Deep Power-Down | ABh | 1010 1011 | Up to 70MHz | 0 | 0 | 0 |
| Ultra Deep Power-Down | 79h | 0111 1001 | Up to 70MHz | 0 | 0 | 0 |

1. Three address bytes are required for the first operation to designate the address to start programming. Afterwards, the internal address counter automatically increments, so subsequent Sequential Program Mode operations only require clocking in of the opcode and the data byte until the Sequential Program Mode has been exited.

# 7. Read Commands

## 7.1 Read Array

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 clock cycle.

Two opcodes (0Bh and 03h) can be used for the Read Array command. The use of each opcode depends on the maximum clock frequency that will be used to read data from the device. The 0Bh opcode can be used at any clock frequency up to the maximum specified by $f_{CLK}$, and the 03h opcode can be used for lower frequency read operations up to the maximum specified by $f_{RDLF}$.

To perform the Read Array operation, the $\overline{CS}$ pin must first be asserted and the appropriate opcode (0Bh or 03h) must be clocked into the device. After the opcode has been clocked in, the three address bytes must be clocked in to specify the starting address location of the first byte to read within the memory array. Following the three address bytes, an additional dummy byte needs to be clocked into the device if the 0Bh opcode is used for the Read Array operation.

After the three address bytes (and the dummy byte if using opcode 0Bh) have been clocked in, additional clock cycles will result in data being output on the SO pin. The data is always output with the MSB of a byte first. When the last byte (03FFFFh) of the memory array has been read, the device will continue reading back at the beginning of the array (000000h). No delays will be incurred when wrapping around from the end of the array to the beginning of the array.

Deasserting the $\overline{CS}$ pin will terminate the read operation and put 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.

**Figure 7-1. Read Array - 03h Opcode**

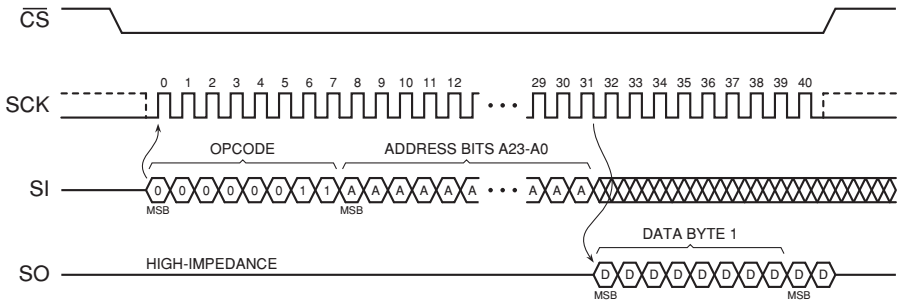
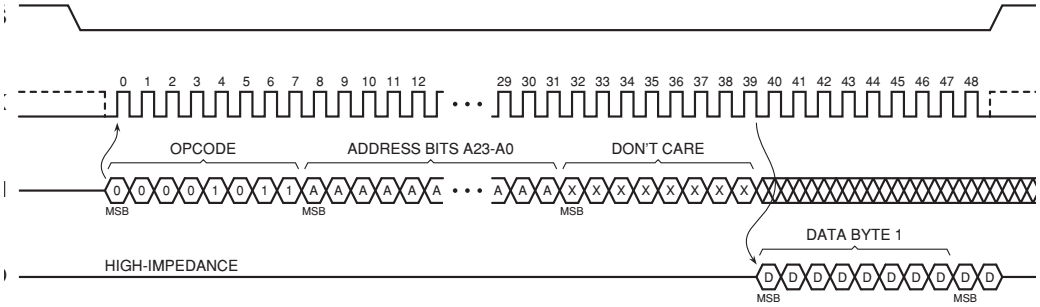


**Figure 7-2. Read Array - 0Bh Opcode**



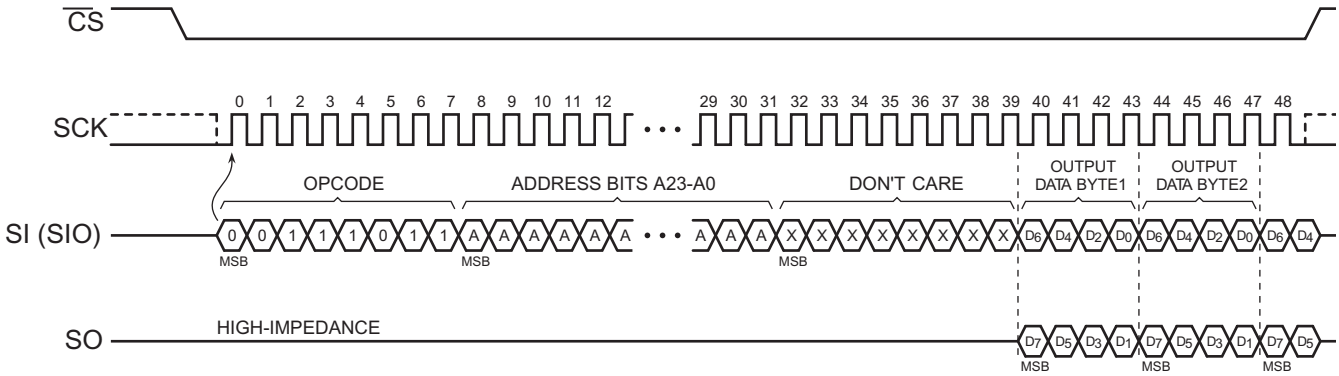

## 7.2 Dual-Output Read Array

The Dual-Output Read Array command is similar to the standard Read Array 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.

The Dual-Output Read Array command can be used at any clock frequency, up to the maximum specified by $f_{RDDO}$. To perform the Dual-Output Read Array operation, the $\overline{CS}$ pin must first be asserted and then the opcode 3Bh must be clocked into the device. After the opcode has been clocked in, the three address bytes must be clocked in to specify the location of the first byte to read within the memory array. Following the three address bytes, a single dummy byte must also be clocked into the device.

After the three address bytes and the dummy byte have been clocked in, additional clock cycles will result in data being output on both the SO and SI pins. The data is always output with the MSB of a byte first and the MSB is always output on the SO pin. During the first clock cycle, bit seven of the first data byte is output on the SO pin, while bit six of the same data byte is output on the SIO pin. During the next clock cycle, bits five and four of the first data byte are output on the SO and SIO pins, respectively. The sequence continues with each byte of data being output after every four clock cycles. When the last byte (03FFFFh) of the memory array has been read, the device will continue reading from the beginning of the array (000000h). No delays will be incurred when wrapping around from the end of the array to the beginning of the array.Deasserting the $\overline{CS}$ pin will terminate the read operation and put 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.

**Figure 7-3. Dual-Output Read Array**



# 8. Program and Erase Commands

## 8.1 Byte/Page Program

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). Before a Byte/Page Program command can be started, the Write Enable command must have been previously issued to the device (see “Write Enable” on page 17) to set the Write Enable Latch (WEL) bit of the Status Register to a logical “1” state.

To perform a Byte/Page Program command, an opcode of 02h must be clocked into the device followed by the three address bytes denoting the first byte location of the memory array to begin programming at. After the address bytes have been clocked in, data can then be clocked into the device and will be stored in an internal buffer.

If the starting memory address denoted by A23-A0 does not fall on an even 256-byte page boundary (A7-A0 are not all 0), then special circumstances regarding which memory locations to be programmed will apply. In this situation, any data that is sent to the device that goes beyond the end of the page will wrap 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 will be programmed at addresses 0000FEh and 0000FFh while the last byte of data will be programmed at address 000000h. The remaining bytes in the page (addresses 000001h through 0000FDh) will not be programmed and will remain in the erased state (FFh). In addition, if more than 256 bytes of data are sent to the device, then only the last 256 bytes sent will be latched into the internal buffer.

When the $\overline{CS}$ pin is deasserted, the device will take the data stored in the internal buffer and program it into the appropriate memory array locations based on the starting address specified by A23-A0 and the number of data bytes sent to the device. If less than 256 bytes of data were sent to the device, then the remaining bytes within the page will not be programmed and will remain in the erased state (FFh). The programming of the data bytes is internally self-timed and should take place in a time of $t_{PP}$ or $t_{BP}$ if only programming a single byte.

The three address bytes and at least one 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 will abort the operation and no data will be programmed into the memory array. In addition, if the memory is in the protected state (see “Protect Sector” on page 19), then the Byte/Page Program command will not be executed, and the device will return to the idle state once the $\overline{CS}$ pin has been deasserted. The WEL bit in the Status Register will be reset back to the logical “0” state if the program cycle aborts due to an incomplete address being sent, an incomplete byte of data being sent, the $\overline{CS}$ pin being deasserted on uneven byte boundaries, or because the memory location to be programmed is protected.

While the device is programming, the Status Register can be read and will indicate that the device is busy. For faster throughput, it is recommended that the Status Register be polled rather than waiting the $t_{BP}$ or $t_{PP}$ time to determine if the data bytes have finished programming. For fastest throughput and least power consumption, it is recommended that the Active Status Interrupt command 25h be used. After the initial 16 clks, no more clocks are required. Once the BUSY cycle is done, SO will be driven low immediately to signal the device has finished programming.At some point before the program cycle completes, the WEL bit in the Status Register will be reset back to the logical “0” state.

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 will be indicated by the EPE bit in the Status Register.

**Figure 8-1. Byte Program**

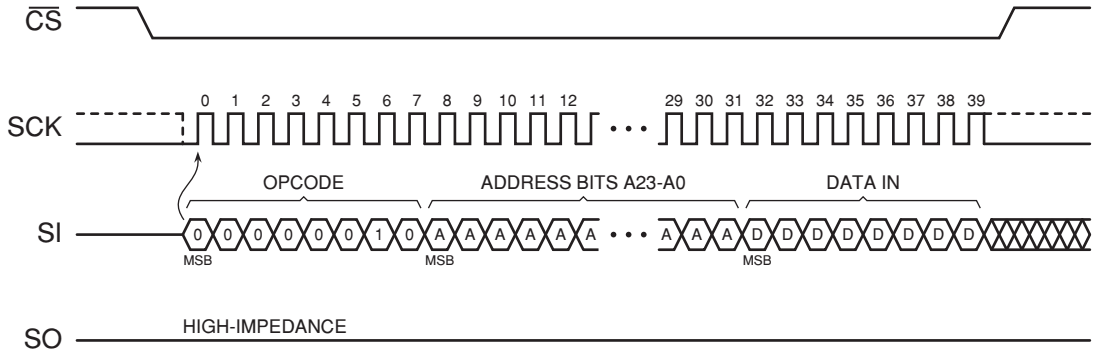
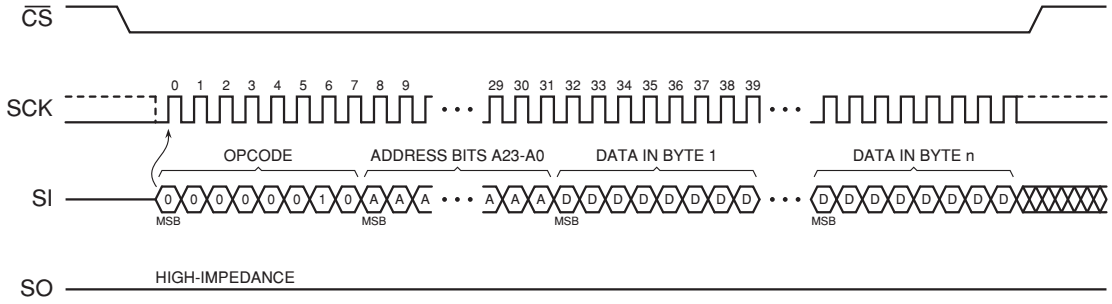



**Figure 8-2. Page Program**




## 8.2 Dual-Input Byte/Page Program

The Dual-Input Byte/Page Program command is similar to the standard Byte/Page Program command 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-Input Byte/Page Program command allows two bits of data to be clocked into the device on every clock cycle rather than just one.

Before the Dual-Input Byte/Page Program command can be started, the Write Enable command must have been previously issued to the device (see “Write Enable” on page 17) to set the Write Enable Latch (WEL) bit of the Status Register to a Logical 1 state. To perform a Dual-Input Byte/Page Program command, an A2h opcode must be clocked into the device followed by the three address bytes denoting the first location of the memory array to begin programming at. After the address bytes have been clocked in, data can then be clocked into the device two bits at a time on both the SO and SI pins.

The data is always input with the MSB of a byte first, and the MSB is always input on the SO pin. During the first clock cycle, bit seven of the first data byte is input on the SO pin while bit six of the same data byte is input on the SI pin. During the next clock cycle, bits five and four of the first data byte are input on the SO and SI pins, respectively. The sequence continues with each byte of data being input after every four clock cycles. Like the standard Byte/Page Program command, all data clocked into the device are stored in an internal buffer.

If the starting memory address denoted by A23-A0 does not fall on an even 256-byte page boundary (A7-A0 are not all 0), then special circumstances regarding which memory locations are to be programmed will apply. In this situation, any data that are sent to the device that go beyond the end of the page will wrap around to the beginning of the same page. In addition, if more than 256 bytes of data is sent to the device, then only the last 256 bytes sent will be latched into the internal buffer.

**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 will be programmed at addresses 0000FEh and 0000FFh, while the last byte of

data will be programmed at address 000000h. The remaining bytes in the page (addresses 000001h through 0000FDh) will not be programmed and will remain in the erased state (FFh).

When the $\overline{CS}$ pin is deasserted, the device will program the data stored in the internal buffer into the appropriate memory array locations based on the starting address specified by A23-A0 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 will not be programmed and will remain in the erased state (FFh). The programming of the data bytes is internally self-timed and should take place in a time of $t_{PP}$ or $t_{BP}$ if only programming a page ($t_{PP)}$ or a single byte ($t_{BP)}$.

The three address bytes and at least one 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 will abort the operation and no data will be programmed into the memory array. In addition, if the address specified by A23-A0 points to a memory location within a sector that is in the protected state (see “Protect Sector” on page 19), then the Byte/Page Program command will not be executed and the device will return to the idle state once the $\overline{CS}$ pin has been deasserted. The WEL bit in the Status Register will be reset back to the Logical 0 state if the program cycle aborts due to an incomplete address being sent, an incomplete byte of data being sent, the $\overline{CS}$ pin being deasserted on uneven byte boundaries, or because the memory location to be programmed is protected or locked down.

While the device is programming, the Status Register can be read and will indicate that the device is busy. For faster throughput, it is recommended that the Status Register be polled rather than waiting the $t_{BP}$ or $t_{PP}$ time to determine if the data bytes have finished programming. For fastest throughput and least power consumption, it is recommended that the Active Status Interrupt command 25h be used. At some point before the program cycle completes, the WEL bit in the Status Register will be reset back to the Logical 0 state.

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 will be indicated by the EPE bit in the Status Register.

**Figure 8-3. Dual-Input Byte Program**

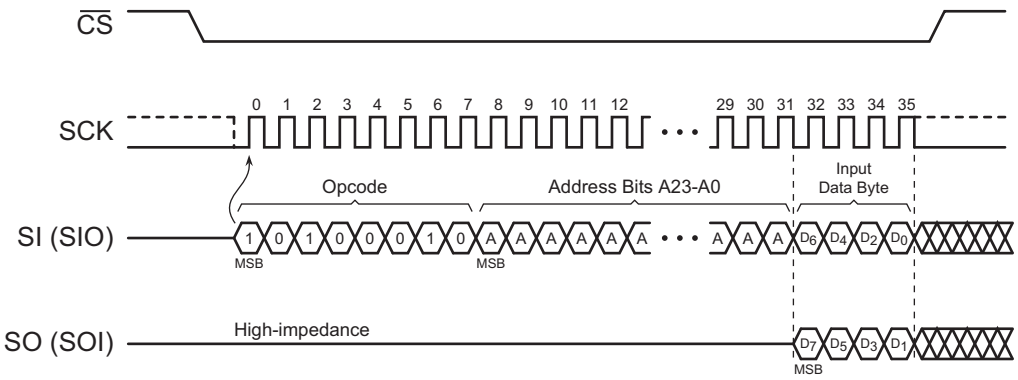
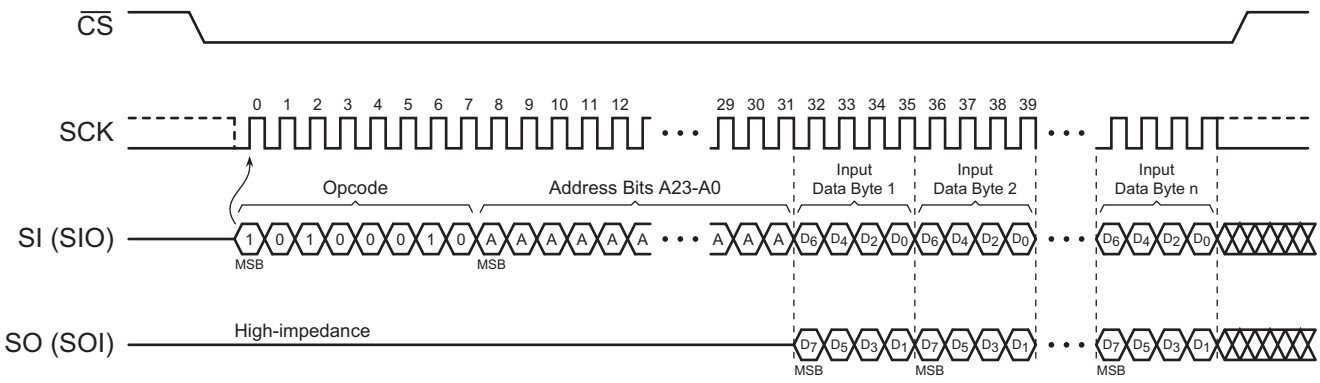


**Figure 8-4. Dual-Input Page Program**




## 8.3 Sequential Program Mode

The Sequential Program Mode improves throughput over the Byte/Page Program command when the Byte/Page Program command is used to program 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, thereby eliminating the need to supply an address sequence to the device for every byte to program. When using the Sequential Program mode, all address locations to be programmed must be in the erased state. Before the Sequential Program mode can first be entered, the Write Enable command must have been previously issued to the device to set the WEL bit of the Status Register to a logical “1” state.

To start the Sequential Program Mode, the $\overline{CS}$ pin must first be asserted, and either an opcode of ADh or AFh must be clocked into the device. For the first program cycle, three address bytes must be clocked in after the opcode to designate the first byte location to program. After the address bytes have been clocked in, the byte of data to be programmed can be sent to the device. Deasserting the $\overline{CS}$ pin will start the internally self-timed program operation, and the byte of data will be programmed into the memory location specified by A23 - A0.

After the first byte has been successfully programmed, a second byte can be programmed by simply reasserting the $\overline{CS}$ pin, clocking in the ADh or AFh opcode, and then clocking in the next byte of data. When the $\overline{CS}$ pin is deasserted, the second byte of data will be programmed into the next sequential memory location. The process would be repeated for any additional bytes. 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 the Status Register back to the logical “0” state.

If more than one byte of data is ever clocked in during each program cycle, then only the last byte of data sent on the SI pin will be stored in the internal latches. The programming of each byte is internally self-timed and should take place in a time of $t_{BP}$. 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 will abort the operation, the byte of data will not be programmed into the memory array, and the WEL bit in the Status Register will be reset back to the logical “0” state.

If the address initially specified by A23 - A0 points to a memory location within a sector that is in the protected state, then the Sequential Program Mode command will not be executed, and the device will return to the idle state once the $\overline{CS}$ pin has been deasserted. The WEL bit in the Status Register will also be reset back to the logical “0” state.

There is no address wrapping when using the Sequential Program Mode. Therefore, when the last byte (03FFFFh) of the memory array has been programmed, the device will automatically exit the Sequential Program mode and reset the WEL bit in the Status Register back to the logical “0” state. In addition, the Sequential Program mode will not automatically skip over protected sectors; therefore, once the highest unprotected memory location in a programming sequence has been programmed, the device will automatically exit the Sequential Program mode and reset the WEL bit in the Status Register. For example, if Sector 1 was protected and Sector 0 was currently being programmed, once the last byte of Sector 0 was programmed, the Sequential Program mode would automatically end. To continue programming with Sector 2, the Sequential Program mode would have to be restarted by supplying the ADh or AFh opcode, the three address bytes, and the first byte of Sector 2 to program.

While the device is programming a byte, the Status Register can be read and will indicate that the device is busy. For faster throughput, it is recommended that the Status Register be polled at the end of each program cycle rather than waiting the $t_{BP}$ time to determine if the byte has finished programming before starting the next Sequential Program mode cycle.

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 will be indicated by the EPE bit in the Status Register.

**Figure 8-5. Sequential Program Mode – Status Register Polling**

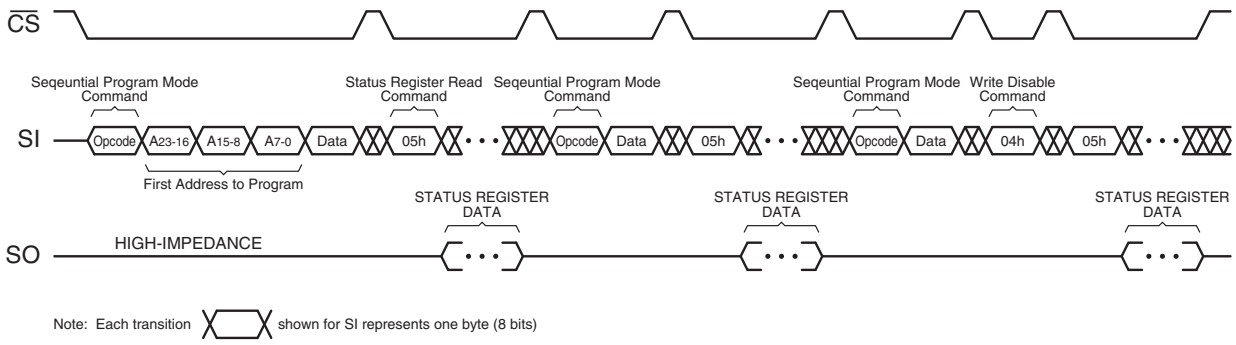
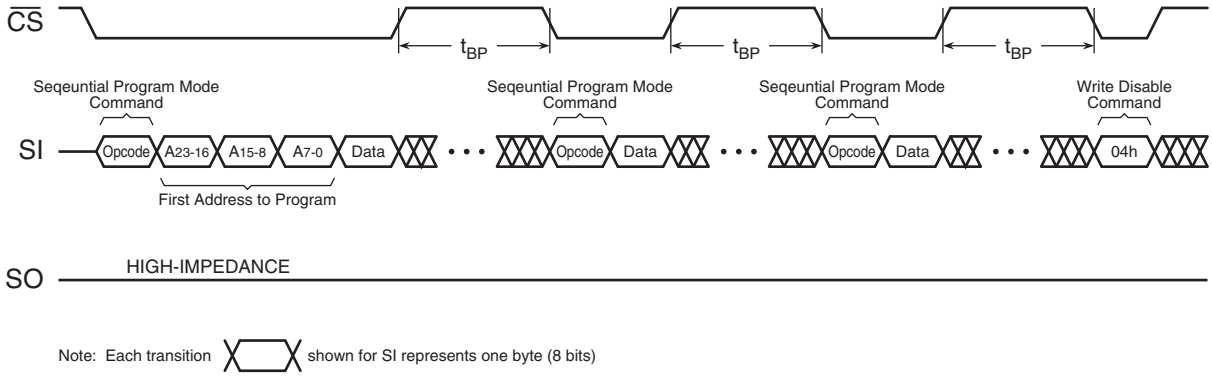



**Figure 8-6. Sequential Program Mode – Waiting Maximum Byte Program Time**



## 8.4 Page Erase

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.

To perform a Page Erase with the standard page size (256 bytes), an opcode of 81h must be clocked into the device followed by three address bytes comprised of eight dummy bits, 8 page address bits (PA7 - PA0) that specify the page in the main memory to be erased, and eight dummy bits.

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

The device also incorporates an intelligent erase algorithm that can detect when a byte location fails to erase properly. If an erase error arises, it will be indicated by the EPE bit in the Status Register.

## 8.5 Block Erase

A block of 4, 32, or 64Kbytes can be erased (all bits set to the logical “1” state) in a single operation by using one of three different opcodes for the Block Erase command. An opcode of 20h is used for a 4-Kbyte erase, an opcode of 52h for a 32-Kbyte erase, and an opcode of D8h is used for a 64-Kbyte erase. Before a Block Erase command can be started, the Write Enable command must have been previously issued to the device to set the WEL bit of the Status Register to a logical “1” state.

To perform a Block Erase, the $\overline{CS}$ pin must first be asserted and the appropriate opcode (20h, 52h, or D8h) must be clocked into the device. After the opcode has been clocked in, the three address bytes specifying an address within the 4-, 32-, or 64-Kbyte block to be erased must be clocked in. Any additional data clocked into the device will be ignored. When the $\overline{CS}$ pin is deasserted, the device will erase the appropriate block. The erasing of the block is internally self-timed and should take 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. Therefore, for a 4-Kbyte erase, address bits A11-A0 will be ignored by the device and their values can be either a logical “1” or “0”. For a 32-Kbyte erase, address bits A14-A0 will be ignored by the device. For a 64-Kbyte erase, address bits A15-A0 will be ignored by the device. 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, and the $\overline{CS}$ pin must be deasserted on an even byte boundary (multiples of eight bits); otherwise, the device will abort the operation and no erase operation will be performed.

If the memory is in the protected state, then the Block Erase command will not be executed, and the device will return to the idle state once the $\overline{CS}$ pin has been deasserted.

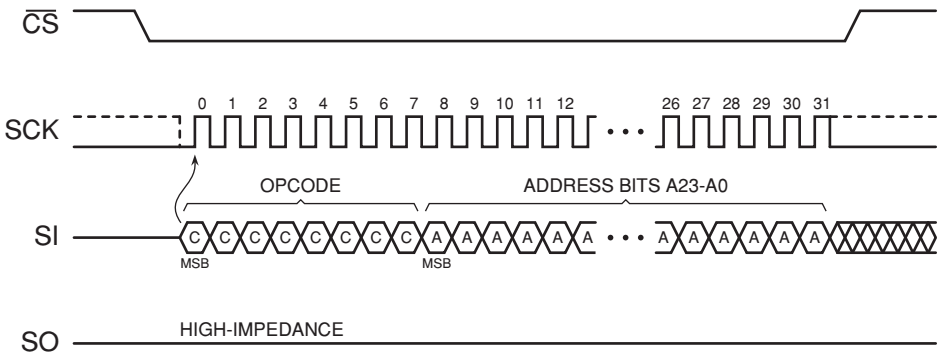
The WEL bit in the Status Register will be reset back to the logical “0” state if the erase cycle aborts due to an incomplete address being sent, the $\overline{CS}$ pin being deasserted on uneven byte boundaries, or because a memory location within the region to be erased is protected.

While the device is executing a successful erase cycle, the Status Register can be read and will indicate that the device is busy. For faster throughput, it is recommended that the Status Register 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 will be reset back to the logical “0” state.

For fastest throughput and least power consumption, it is recommended that the Active Status Interrupt command 25h be used. After the initial 16 clks, no more clocks are required.  Once the BUSY cycle is done, SO will be driven low immediately to signal the device has finished erasing.

The device also incorporates an intelligent erase algorithm that can detect when a byte location fails to erase properly. If an erase error occurs, it will be indicated by the EPE bit in the Status Register.

**Figure 8-7. Block Erase**




## 8.6 Chip Erase

The entire memory array can be erased in a single operation by using the Chip Erase command. Before a Chip Erase command can be started, the Write Enable command must have been previously issued to the device to set the WEL bit of the Status Register to a logical “1” state.

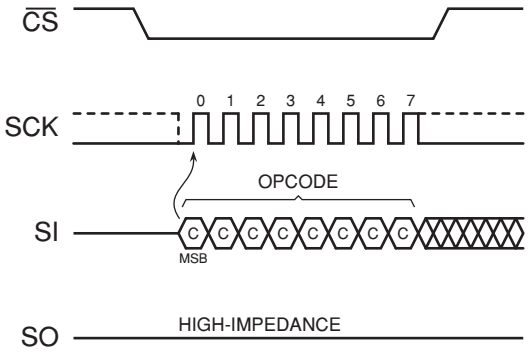
Two opcodes (60h and C7h) can be used for the Chip Erase command. There is no difference in device functionality when utilizing the two opcodes, so they can be used interchangeably. To perform a Chip Erase, one of the two opcodes must be clocked into the device. Since the entire memory array is to be erased, no address bytes need to be clocked into the device, and any data clocked in after the opcode will be ignored. When the $\overline{CS}$ pin is deasserted, the device will erase the entire memory array. The erasing of the device is internally self-timed and should take place in a time of $t_{CHPE}$.

The complete opcode must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on an even byte boundary (multiples of eight bits); otherwise, no erase will be performed. In addition, if any sector in the memory array is in the protected state, then the Chip Erase command will not be executed, and the device will return to the idle state once the $\overline{CS}$ pin has been deasserted. The WEL bit in the Status Register will be reset back to the logical “0” state if the $\overline{CS}$ pin is deasserted on uneven byte boundaries or if the memory is in the protected state.

While the device is executing a successful erase cycle, the Status Register can be read and will indicate that the device is busy. For faster throughput, it is recommended that the Status Register 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 the Status Register will be reset back to the logical “0” state.

The device also incorporates an intelligent erase algorithm that can detect when a byte location fails to erase properly. If an erase error occurs, it will be indicated by the EPE bit in the Status Register.

**Figure 8-8. Chip Erase**



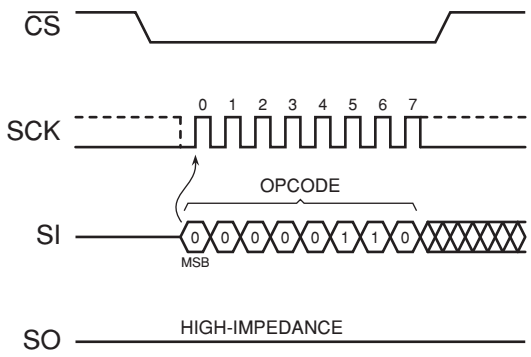
# 9. Protection Commands and Features

## 9.1 Write Enable

The Write Enable command is used to set the Write Enable Latch (WEL) bit in the Status Register to a logical “1” state. The WEL bit must be set before a Byte/Page Program, Erase, Program OTP Security Register, or Write Status Register command can be executed. This makes the issuance of these 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, then the command will not be executed.

To issue the Write Enable command, the $\overline{CS}$ pin must first be asserted and the opcode of 06h must be clocked into the device. No address bytes need to be clocked into the device, and any data clocked in after the opcode will be ignored. When the $\overline{CS}$ pin is deasserted, the WEL bit in the Status Register will be set to a logical “1”. The complete opcode must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on an even byte boundary (multiples of eight bits); otherwise, the device will abort the operation and the state of the WEL bit will not change.

**Figure 9-1. Write Enable**

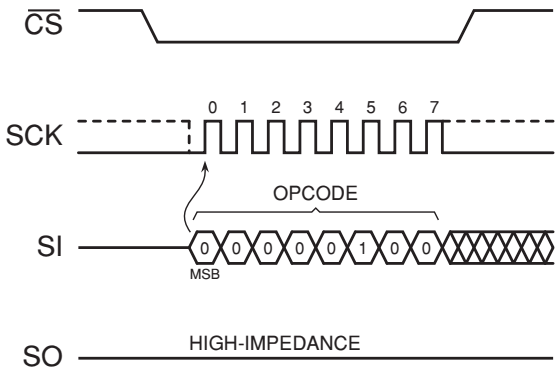



## 9.2 Write Disable

The Write Disable command is used to reset the Write Enable Latch (WEL) bit in the Status Register to the logical “0” state. With the WEL bit reset, all Byte/Page Program, Erase, Program OTP Security Register, and Write Status Register commands will not be executed. Other conditions can also cause the WEL bit to be reset; for more details, refer to the WEL bit section of the Status Register description.

To issue the Write Disable command, the $\overline{CS}$ pin must first be asserted and the opcode of 04h must be clocked into the device. No address bytes need to be clocked into the device, and any data clocked in after the opcode will be ignored. When the $\overline{CS}$ pin is deasserted, the WEL bit in the Status Register will be reset to a logical “0”. The complete opcode must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on an even byte boundary (multiples of eight bits); otherwise, the device will abort the operation and the state of the WEL bit will not change.

**Figure 9-2. Write Disable**




## 9.3 Protect Sector

Every physical sector of the device has a corresponding single-bit Sector Protection Register that is used to control the software protection of a sector. Upon device power-up or after a device reset, each Sector Protection Register will default to the logical “1” state indicating that all sectors are protected and cannot be programmed or erased.

Issuing the Protect Sector command to a particular sector address will set the corresponding Sector Protection Register to the logical “1” state. The following table outlines the two states of the Sector Protection Registers.

**Table 9-1. Sector Protection Register Values**

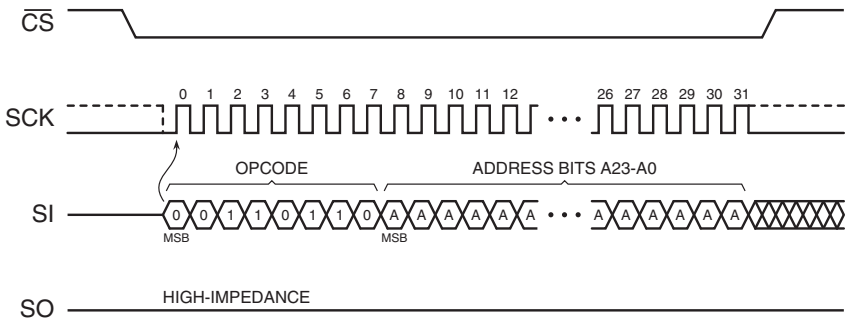
| Value | Sector Protection Status |
|---|---|
| 0 | Sector is unprotected and can be programmed and erased. |
| 1 | Sector is protected and cannot be programmed or erased. This is the default state. |

Before the Protect Sector command can be issued, the Write Enable command must have been previously issued to set the WEL bit in the Status Register to a logical “1”. To issue the Protect Sector command, the $\overline{CS}$ pin must first be asserted and the opcode of 36h must be clocked into the device followed by three address bytes designating any address within the sector to be protected. Any additional data clocked into the device will be ignored. When the $\overline{CS}$ pin is deasserted, the Sector Protection Register corresponding to the physical sector addressed by A23 - A0 will be set to the logical “1” state, and the sector itself will then be protected from program and erase operations. In addition, the WEL bit in the Status Register will be reset back to the logical “0” state.

The complete three address bytes must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on an even byte boundary (multiples of eight bits); otherwise, the device will abort the operation, the state of the Sector Protection Register will be unchanged, and the WEL bit in the Status Register will be reset to a logical “0”.

As a safeguard against accidental or erroneous protecting or unprotecting of sectors, the Sector Protection Registers can themselves be locked from updates by using the SPRL (Sector Protection Registers Locked) bit of the Status Register (please refer to the Status Register description for more details). If the Sector Protection Registers are locked, then any attempts to issue the Protect Sector command will be ignored, and the device will reset the WEL bit in the Status Register back to a logical “0” and return to the idle state once the $\overline{CS}$ pin has been deasserted.

**Figure 9-3. Protect Sector**




## 9.4 Unprotect Sector

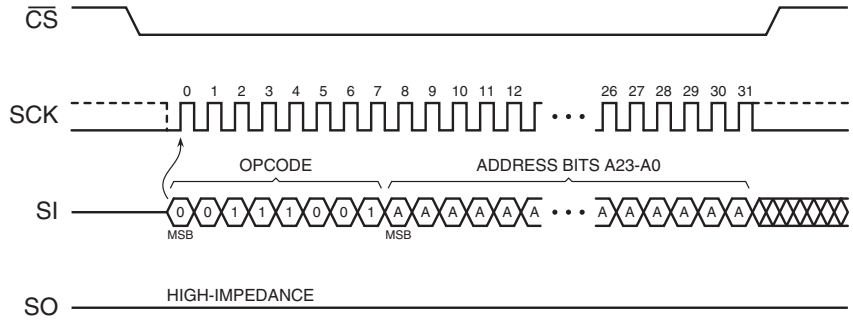
Issuing the Unprotect Sector command to a particular sector address will reset the corresponding Sector Protection Register to the logical “0” state (see Table 9-1 for Sector Protection Register values). Every physical sector of the device has a corresponding single-bit Sector Protection Register that is used to control the software protection of a sector.

Before the Unprotect Sector command can be issued, the Write Enable command must have been previously issued to set the WEL bit in the Status Register to a logical “1”. To issue the Unprotect Sector command, the $\overline{CS}$ pin must first be asserted and the opcode of 39h must be clocked into the device. After the opcode has been clocked in, the three address bytes designating any address within the sector to be unlocked must be clocked in. Any additional data clocked into the device after the address bytes will be ignored. When the $\overline{CS}$ pin is deasserted, the Sector Protection Register corresponding to the sector addressed by A23 - A0 will be reset to the logical “0” state, and the sector itself will be unprotected. In addition, the WEL bit in the Status Register will be reset back to the logical “0” state.

The complete three address bytes must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on an even byte boundary (multiples of eight bits); otherwise, the device will abort the operation, the state of the Sector Protection Register will be unchanged, and the WEL bit in the Status Register will be reset to a logical “0”.

As a safeguard against accidental or erroneous locking or unlocking of sectors, the Sector Protection Registers can themselves be locked from updates by using the SPRL (Sector Protection Registers Locked) bit of the Status Register (please refer to the Status Register description for more details). If the Sector Protection Registers are locked, then any attempts to issue the Unprotect Sector command will be ignored, and the device will reset the WEL bit in the Status Register back to a logical “0” and return to the idle state once the $\overline{CS}$ pin has been deasserted.

**Figure 9-4. Unprotect Sector**




## 9.5 Global Protect/Unprotect

The Global Protect and Global Unprotect features can work in conjunction with the Protect Sector and Unprotect Sector functions. For example, a system can globally protect the entire memory array and then use the Unprotect Sector command to individually unprotect certain sectors and individually reprotect them later by using the Protect Sector command. Likewise, a system can globally unprotect the entire memory array and then individually protect certain sectors as needed.

Performing a Global Protect or Global Unprotect is accomplished by writing a certain combination of data to the Status Register using the Write Status Register command (see “Write Status Register” section on page 31 for command execution details). The Write Status Register command is also used to modify the SPRL (Sector Protection Registers Locked) bit to control hardware and software locking.

To perform a Global Protect, the appropriate $\overline{WP}$ pin and SPRL conditions must be met, and the system must write a logical “1” to bits 5, 4, 3, and 2 of the Status Register. Conversely, to perform a Global Unprotect, the same $\overline{WP}$ and SPRL conditions must be met but the system must write a logical “0” to bits 5, 4, 3, and 2 of the Status Register. Table 9-2 details the conditions necessary for a Global Protect or Global Unprotect to be performed.

**Table 9-2. Valid SPRL and Global Protect/Unprotect Conditions**

| $\overline{WP}$ State | Current SPRL Value | New Write Status Register Data Bit 7 6 5 4 3 2 1 0 | Protection Operation | New SPRL Value |
|---|---|---|---|---|
| 0 | 0 | 0 x 0 0 0 0 x x | Global Unprotect – all Sector Protection Registers reset to 0 | 0 |
| | | 0 x 0 0 0 1 x x | No change to current protection. | 0 |
| | | ? | No change to current protection. | 0 |
| | | 0 x 1 1 1 0 x x | No change to current protection. | 0 |
| | | 0 x 1 1 1 1 x x | Global Protect – all Sector Protection Registers set to 1 | 0 |
| | | 1 x 0 0 0 0 x x | Global Unprotect – all Sector Protection Registers reset to 0 | 1 |
| | | 1 x 0 0 0 1 x x | No change to current protection. | 1 |
| | | ? | No change to current protection. | 1 |
| | | 1 x 1 1 1 0 x x | No change to current protection. | 1 |
| | | 1 x 1 1 1 1 x x | Global Protect – all Sector Protection Registers set to 1 | 1 |
| 0 | 1 | x x x x x x x x | No change to the current protection level. All sectors currently protected will remain protected and all sectors currently unprotected will remain unprotected.<br><br>The Sector Protection Registers are hard-locked and cannot be changed when the $\overline{WP}$ pin is LOW and the current state of SPRL is 1. Therefore, a Global Protect/Unprotect will not occur. In addition, the SPRL bit cannot be changed (the $\overline{WP}$ pin must be HIGH in order to change SPRL back to a 0). | |
| 1 | 0 | 0 x 0 0 0 0 x x | Global Unprotect – all Sector Protection Registers reset to 0 | 0 |
| | | 0 x 0 0 0 1 x x | No change to current protection. | 0 |
| | | ? | No change to current protection. | 0 |
| | | 0 x 1 1 1 0 x x | No change to current protection. | 0 |
| | | 0 x 1 1 1 1 x x | Global Protect – all Sector Protection Registers set to 1 | 0 |
| | | 1 x 0 0 0 0 x x | Global Unprotect – all Sector Protection Registers reset to 0 | 1 |
| | | 1 x 0 0 0 1 x x | No change to current protection. | 1 |
| | | ? | No change to current protection. | 1 |
| | | 1 x 1 1 1 0 x x | No change to current protection. | 1 |
| | | 1 x 1 1 1 1 x x | Global Protect – all Sector Protection Registers set to 1 | 1 |
| 1 | 1 | 0 x 0 0 0 0 x x<br>0 x 0 0 0 1 x x<br>?<br>0 x 1 1 1 0 x x<br>0 x 1 1 1 1 x x | No change to the current protection level. All sectors currently protected will remain protected, and all sectors currently unprotected will remain unprotected. | 0<br>0<br>0<br>0<br>0 |
| | | 1 x 0 0 0 0 x x<br>1 x 0 0 0 1 x x<br>?<br>1 x 1 1 1 0 x x<br>1 x 1 1 1 1 x x | The Sector Protection Registers are soft-locked and cannot be changed when the current state of SPRL is 1. Therefore, a Global Protect/Unprotect will not occur. However, the SPRL bit can be changed back to a 0 from a 1 since the $\overline{WP}$ pin is HIGH. To perform a Global Protect/Unprotect, the Write Status Register command must be issued again after the SPRL bit has been changed from a 1 to a 0. | 1<br>1<br>1<br>1<br>1 |

Essentially, if the SPRL bit of the Status Register is in the logical “0” state (Sector Protection Registers are not locked), then writing a 00h to the Status Register will perform a Global Unprotect without changing the state of the SPRL bit. Similarly, writing a 7Fh to the Status Register will perform a Global Protect and keep the SPRL bit in the logical “0” state. The SPRL bit can, of course, be changed to a logical “1” by writing an FFh if software-locking or hardware-locking is desired along with the Global Protect.

If the desire is to only change the SPRL bit without performing a Global Protect or Global Unprotect, then the system can simply write a 0Fh to the Status Register to change the SPRL bit from a logical “1” to a logical “0” provided the $\overline{WP}$ pin is deasserted. Likewise, the system can write an F0h to change the SPRL bit from a logical “0” to a logical “1” without affecting the current sector protection status (no changes will be made to the Sector Protection Registers).

When writing to the Status Register, bits 5, 4, 3, and 2 will not actually be modified but will be decoded by the device for the purposes of the Global Protect and Global Unprotect functions. Only bit 7, the SPRL bit, will actually be modified. Therefore, when reading the Status Register, bits 5, 4, 3, and 2 will not reflect the values written to them but will instead indicate the status of the $\overline{WP}$ pin and the sector protection status. Please refer to the “Read Status Register” section and Table 11-1 on page 25 for details on the Status Register format and what values can be read for bits 5, 4, 3, and 2.

## 9.6 Read Sector Protection Registers

The Sector Protection Registers can be read to determine the current software protection status of each sector. Reading the Sector Protection Registers, however, will not determine the status of the $\overline{WP}$ pin.

To read the Sector Protection Register for a particular sector, the $\overline{CS}$ pin must first be asserted and the opcode of 3Ch must be clocked in. Once the opcode has been clocked in, three address bytes designating any address within the sector must be clocked in. After the last address byte has been clocked in, the device will begin outputting data on the SO pin during every subsequent clock cycle. The data being output will be a repeating byte of either FFh or 00h to denote the value of the appropriate Sector Protection Register.

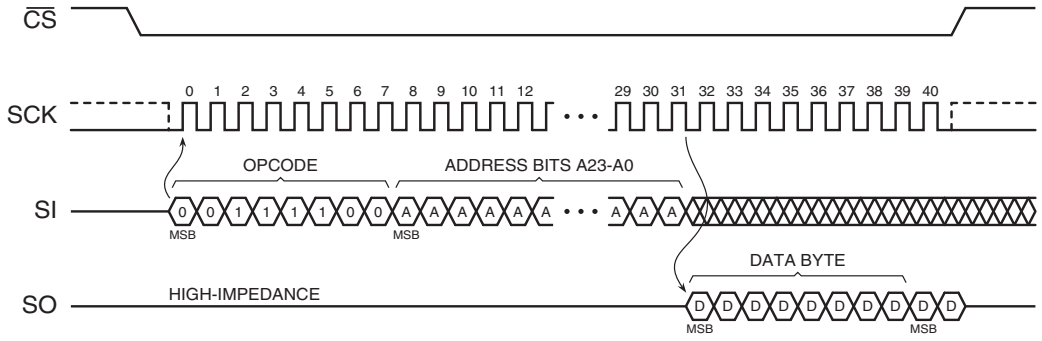
**Table 9-3. Read Sector Protection Register – Output Data**

| Output Data | Sector Protection Register Value |
|---|---|
| 00h | Sector Protection Register value is 0 (sector is unprotected). |
| FFh | Sector Protection Register value is 1 (sector is protected). |

Deasserting the $\overline{CS}$ pin will terminate the read 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.

In addition to reading the individual Sector Protection Registers, the Software Protection Status (SWP) bit in the Status Register can be read to determine if all, some, or none of the sectors are software protected (refer to the “Status Register Commands” on page 28 for more details).

**Figure 9-5. Read Sector Protection Register**



## 9.7 Protected States and the Write Protect ($\overline{WP}$) Pin

The $\overline{WP}$ pin is not linked to the memory array itself and has no direct effect on the protection status of the memory array. Instead, the $\overline{WP}$ pin, in conjunction with the SPRL (Sector Protection Registers Locked) bit in the Status Register, is used to control the hardware locking mechanism of the device. For hardware locking to be active, two conditions must be met – the $\overline{WP}$ pin must be asserted and the SPRL bit must be in the logical "1" state.

When hardware locking is active, the Sector Protection Registers are locked and the SPRL bit itself is also locked. Therefore, sectors that are protected will be locked in the protected state, and sectors that are unprotected will be locked in the unprotected state. These states cannot be changed as long as hardware locking is active, so the Protect Sector, Unprotect Sector, and Write Status Register commands will be ignored. In order to modify the protection status of a sector, the $\overline{WP}$ pin must first be deasserted, and the SPRL bit in the Status Register must be reset back to the logical "0" state using the Write Status Register command. When resetting the SPRL bit back to a logical "0", it is not possible to perform a Global Protect or Global Unprotect at the same time since the Sector Protection Registers remain soft-locked until after the Write Status Register command has been executed.

If the $\overline{WP}$ pin is permanently connected to GND, then once the SPRL bit is set to a logical "1", the only way to reset the bit back to the logical "0" state is to power-cycle or reset the device. This allows a system to power-up with all sectors software protected but not hardware locked. Therefore, sectors can be unprotected and protected as needed and then hardware locked at a later time by simply setting the SPRL bit in the Status Register.

When the $\overline{WP}$ pin is deasserted, or if the $\overline{WP}$ pin is permanently connected to VCC, the SPRL bit in the Status Register can still be set to a logical "1" to lock the Sector Protection Registers. This provides a software locking ability to prevent erroneous Protect Sector or Unprotect Sector commands from being processed. When changing the SPRL bit to a logical "1" from a logical "0", it is also possible to perform a Global Protect or Global Unprotect at the same time by writing the appropriate values into bits 5, 4, 3, and 2 of the Status Register.

Tables 9-4 and 9-5 detail the various protection and locking states of the device.

.

**Table 9-4. Sector Protection Register States**

| $\overline{WP}$ | Sector Protection Register n (1) | Sector n(1) |
|---|---|---|
| X (Don't Care) | 0 | Unprotected |
| | 1 | Protected |

1. "n" represents a sector number

**Table 9-5. Hardware and Software Locking**

| $\overline{WP}$ | SPRL | Locking | SPRL Change Allowed | Sector Protection Registers |
|---|---|---|---|---|
| 0 | 0 | | Can be modified from 0 to 1 | Unlocked and modifiable using the Protect and Unprotect Sector commands. Global Protect and Unprotect can also be performed. |
| 0 | 1 | Hardware Locked | Locked | Locked in current state. Protect and Unprotect Sector commands will be ignored. Global Protect and Unprotect cannot be performed. |
| 1 | 0 | | Can be modified from 0 to 1 | Unlocked and modifiable using the Protect and Unprotect Sector commands. Global Protect and Unprotect can also be performed. |
| 1 | 1 | Software Locked | Can be modified from 1 to 0 | Locked in current state. Protect and Unprotect Sector commands will be ignored. Global Protect and Unprotect cannot be performed. |

# 10. Security Commands

## 10.1 Program OTP Security Register

The device contains a specialized OTP (One-Time Programmable) Security Register that can be used for purposes such as unique device serialization, system-level Electronic Serial Number (ESN) storage, locked key storage, etc. The OTP Security Register is independent of the main Flash memory array and is comprised of a total of 128 bytes of memory divided into two portions. The first 64 bytes (byte locations 0 through 63) of the OTP Security Register are allocated as a one-time user-programmable space. Once these 64 bytes have been programmed, they cannot be erased or reprogrammed. The remaining 64 bytes of the OTP Security Register (byte locations 64 through 127) are factory programmed by Adesto and will contain a unique value for each device. The factory programmed data is fixed and cannot be changed

.

**Table 10-1. OTP Security Register**

| Security Register Byte Number | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | | 62 | 63 | 64 | 65 | | 126 | 127 |
| One-Time User Programmable | | | | | Factory Programmed by Adesto | | | | |

The user-programmable portion of the OTP Security Register does not need to be erased before it is programmed. In addition, the Program OTP Security Register command operates on the entire 64-byte user-programmable portion of the OTP Security Register at one time. Once the user-programmable space has been programmed with any number of bytes, the user-programmable space cannot be programmed again; therefore, it is not possible to only program the first two bytes of the register and then program the remaining 62 bytes at a later time.

Before the Program OTP Security Register command can be issued, the Write Enable command must have been previously issued to set the WEL bit in the Status Register to a logical “1”. To program the OTP Security Register, the $\overline{CS}$ pin must first be asserted and an opcode of 9Bh must be clocked into the device followed by the three address bytes denoting the first byte location of the OTP Security Register to begin programming at. Since the size of the user-programmable portion of the OTP Security Register is 64 bytes, the upper order address bits do not need to be decoded by the device. Therefore, address bits A23-A6 will be ignored by the device and their values can be either a logical “1” or “0”. After the address bytes have been clocked in, data can then be clocked into the device and will be stored in the internal buffer.

If the starting memory address denoted by A23-A0 does not start at the beginning of the OTP Security Register memory space (A5-A0 are not all 0), then special circumstances regarding which OTP Security Register locations to be programmed will apply. In this situation, any data that is sent to the device that goes beyond the end of the 64-byte user-programmable space will wrap around back to the beginning of the OTP Security Register. For example, if the starting address denoted by A23-A0 is 00003Eh, and three bytes of data are sent to the device, then the first two bytes of data will be programmed at OTP Security Register addresses 00003Eh and 00003Fh while the last byte of data will be programmed at address 000000h. The remaining bytes in the OTP Security Register (addresses 000001h through 00003Dh) will not be programmed and will remain in the erased state (FFh). In addition, if more than 64 bytes of data are sent to the device, then only the last 64 bytes sent will be latched into the internal buffer.

When the $\overline{CS}$ pin is deasserted, the device will take the data stored in the internal buffer and program it into the appropriate OTP Security Register locations based on the starting address specified by A23-A0 and the number of data bytes sent to the device. If less than 64 bytes of data were sent to the device, then the remaining bytes within the OTP Security Register will not be programmed and will remain in the erased state (FFh). The programming of the data bytes is internally self-timed and should take place in a time of $t_{OTPP}$.

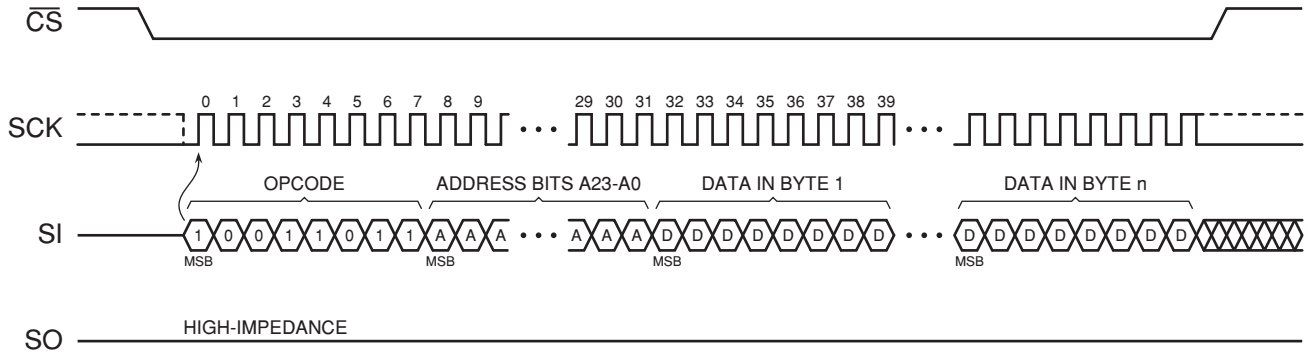
The three address bytes and at least one complete byte of data must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on even byte boundaries (multiples of eight bits); otherwise, the device will abort the operation and the user-programmable portion of the OTP Security Register will not be programmed. The WEL bit in the Status Register will be reset back to the logical “0” state if the OTP Security Register program cycle aborts due to an incomplete address being sent, an incomplete byte of data being sent, the $\overline{CS}$ pin being deasserted on uneven byte boundaries, or because the user-programmable portion of the OTP Security Register was previously programmed.

While the device is programming the OTP Security Register, the Status Register can be read and will indicate that the device is busy. 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, the WEL bit in the Status Register will be reset back to the logical “0” state.

If the device is powered-down during the OTP Security Register program cycle, then the contents of the 64-byte user programmable portion of the OTP Security Register cannot be guaranteed and cannot be programmed again.

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

**Figure 10-1. Program OTP Security Register**



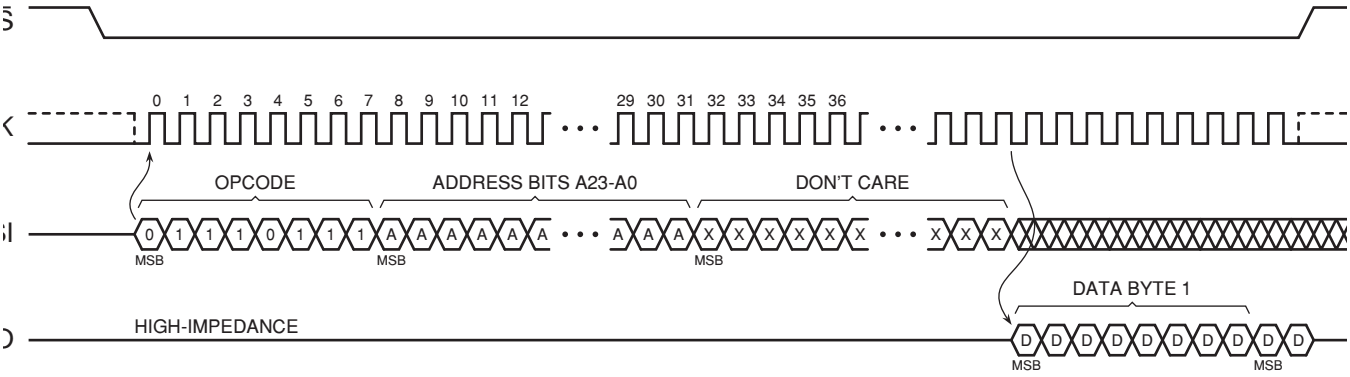

## 10.2 Read OTP Security Register

The OTP Security Register can be sequentially read in a similar fashion to the Read Array operation up to the maximum clock frequency specified by $f_{CLK}$. To read the OTP Security Register, the $\overline{CS}$ pin must first be asserted and the opcode of 77h must be clocked into the device. After the opcode has been clocked in, the three address bytes must be clocked in to specify the starting address location of the first byte to read within the OTP Security Register. Following the three address bytes, two dummy bytes must be clocked into the device before data can be output.

After the three address bytes and the dummy bytes have been clocked in, additional clock cycles will result in OTP Security Register data being output on the SO pin. When the last byte (00007Fh) of the OTP Security Register has been read, the device will continue reading back at the beginning of the register (000000h). No delays will be incurred when wrapping around from the end of the register to the beginning of the register.

Deasserting the $\overline{CS}$ pin will terminate the read 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.

**Figure 10-2. Read OTP Security Register**




# 11. Status Register Commands

## 11.1 Read Status Register

The Status Register 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. The Status Register can be read at any time, including during an internally self-timed program or erase operation.The Status Register consists of two bytes.

To read the Status Register, the $\overline{CS}$ pin must first be asserted and the opcode of 05h must be clocked into the device. After the opcode has been clocked in, the device will begin outputting Status Register data on the SO pin during every subsequent clock cycle. After the last bit (bit 0) of Status Register Byte 2 has been clocked out, the sequence will repeat itself, starting again with bit 7 of Status Register Byte 1, 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 will output new data.

Deasserting the $\overline{CS}$ pin will terminate 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.

**Table 11-1. Status Register Format**

| Bit (1) | Name | | Type (2) | Description | |
|---|---|---|---|---|---|
| 7 | SPRL | Sector Protection Registers Locked | R/W | 0 | Sector Protection Registers are unlocked (default). |
| | | | | 1 | Sector Protection Registers are locked. |
| 6 | SPM | Sequential Program Mode Status | R | 0 | Byte/Page Programming Mode (default). |
| | | | | 1 | Sequential Programming Mode entered. |
| 5 | EPE | Erase/Program Error | R | 0 | Erase or program operation was successful. |
| | | | | 1 | Erase or program error detected. |
| 4 | WPP | Write Protect ($\overline{WP}$) Pin Status | R | 0 | $\overline{WP}$ is asserted. |
| | | | | 1 | $\overline{WP}$ is deasserted. |

**Table 11-1. Status Register Format**

| Bit [1] | Name | | Type [2] | Description | |
|---|---|---|---|---|---|
| 3:2 | SWP | Software Protection Status | R | 00 | All sectors are software unprotected (all Sector Protection Registers are 0). |
| | | | | 01 | Some sectors are software protected. Read individual Sector Protection Registers to determine which sectors are protected. |
| | | | | 10 | *Reserved for future use.* |
| | | | | 11 | All sectors are software protected (all Sector Protection Registers are 1 – default). |
| 1 | WEL | Write Enable Latch Status | R | 0 | Device is not write enabled (default). |
| | | | | 1 | Device is write enabled. |
| 0 | $\overline{\text{RDY}}$/BSY | Ready/Busy Status | R | 0 | Device is ready. |
| | | | | 1 | Device is busy with an internal operation. |

1. Only bit 7 of the Status Register will be modified when using the Write Status Register command.
2. R/W = Readable and writable
   R = Readable only

### 11.1.1 SPRL Bit

The SPRL bit is used to control whether the Sector Protection Registers can be modified or not. When the SPRL bit is in the logical “1” state, all Sector Protection Registers are locked and cannot be modified with the Protect Sector and Unprotect Sector commands (the device will ignore these commands). In addition, the Global Protect and Global Unprotect features cannot be performed. Any sectors that are presently protected will remain protected, and any sectors that are presently unprotected will remain unprotected.

When the SPRL bit is in the logical “0” state, all Sector Protection Registers are unlocked and can be modified (the Protect Sector and Unprotect Sector commands, as well as the Global Protect and Global Unprotect features, will be processed as normal). The SPRL bit defaults to the logical “0” state after a power-up or a device reset.

The SPRL bit can be modified freely whenever the $\overline{\text{WP}}$ pin is deasserted. However, if the $\overline{\text{WP}}$ pin is asserted, then the SPRL bit may only be changed from a logical “0” (Sector Protection Registers are unlocked) to a logical “1” (Sector Protection Registers are locked). In order to reset the SPRL bit back to a logical “0” using the Write Status Register command, the $\overline{\text{WP}}$ pin will have to first be deasserted.The SPRL bit is the only bit of the Status Register that can be user modified via the Write Status Register command.

### 11.1.2 SPM Bit

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.

### 11.1.3 EPE Bit

The EPE bit indicates whether the last erase or program operation completed successfully or not. If at least one byte during the erase or program operation did not erase or program properly, then the EPE bit will be set to the logical “1” state. The EPE bit will not be set if an erase or program operation aborts for any reason such as an attempt to erase or program a protected region or if the WEL bit is not set prior to an erase or program operation. The EPE bit will be updated after every erase and program operation.

### 11.1.4 WPP Bit

The WPP bit can be read to determine if the $\overline{WP}$ pin has been asserted or not.

### 11.1.5 SWP Bits

The SWP bits provide feedback on the software protection status for the device. There are three possible combinations of the SWP bits that indicate whether none, some, or all of the sectors have been protected using the Protect Sector command or the Global Protect feature. If the SWP bits indicate that some of the sectors have been protected, then the individual Sector Protection Registers can be read with the Read Sector Protection Registers command to determine which sectors are in fact protected.

### 11.1.6 WEL Bit

The WEL bit indicates the current status of the internal Write Enable Latch. When the WEL bit is in the logical “0” state, the device will not accept any program, erase, Protect Sector, Unprotect Sector, or Write Status Register commands. The WEL bit defaults to the logical “0” state after a device power-up or reset. In addition, the WEL bit will be reset to the logical “0” state automatically under the following conditions:

- Write Disable operation completes successfully
- Write Status Register operation completes successfully or aborts
- Protect Sector operation completes successfully or aborts
- Unprotect Sector operation completes successfully or aborts
- Byte/Page Program operation completes successfully or aborts
- Sequential Program Mode reaches highest unprotected memory location
- Sequential Program Mode reaches the end of the memory array
- Sequential Program Mode aborts [1]
- Block Erase operation completes successfully or aborts
- Chip Erase operation completes successfully or aborts
- Hold condition aborts

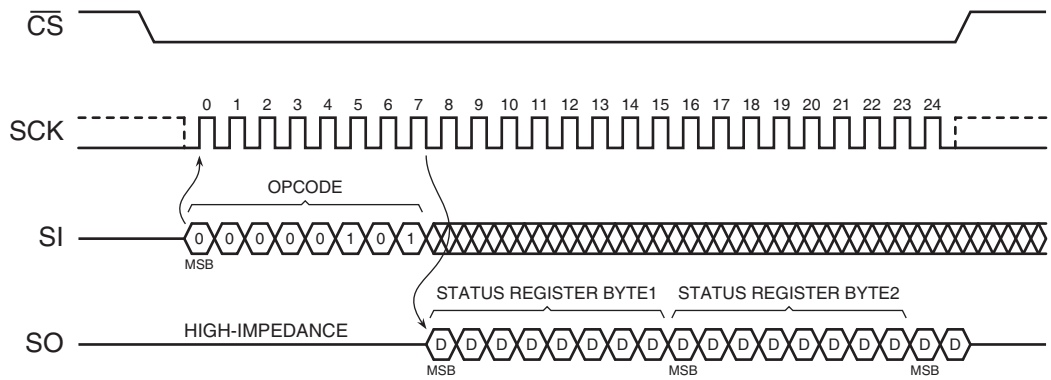
If the WEL bit is in the logical “1” state, it will not be reset to a logical “0” if an operation aborts due to an incomplete or unrecognized opcode being clocked into the device before the $\overline{CS}$ pin is deasserted. In order for the WEL bit to be reset when an operation aborts prematurely, the entire opcode for a program, erase, Protect Sector, Unprotect Sector, or Write Status Register command must have been clocked into the device.

### 11.1.7 $\overline{RDY}$/BSY Bit

The $\overline{RDY}$/BSY bit is used to determine whether or not an internal operation, such as a program or erase, is in progress. To poll the $\overline{RDY}$/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 $\overline{RDY}$/BSY bit changes from a logical “1” to a logical “0”.Note that the $\overline{RDY}$/BSY bit can be read either from Status Register Byte 1 or from Status Register Byte 2. See also the Active Status Interrupt command. (11.2)

1. WEL bit will not be reset if Software Reset command is entered.

**Figure 11-1. Read Status Register**




### 11.1.8 RSTE Bit

The RSTE bit is used to enable or disable the Reset command. When the RSTE bit is in the Logical 0 state (the default state after power-up), the Reset command is disabled and any attempts to reset the device using the Reset command will be ignored. When the RSTE bit is in the Logical 1 state, the Reset command is enabled.

The RSTE bit will retain its state as long as power is applied to the device. Once set to the Logical 1 state, the RSTE bit will remain in that state until it is modified using the Write Status Register Byte 2 command or until the device has been power cycled. The Reset command itself will not change the state of the RSTE bit.

**Table 11-2. Status Register Format – Byte 2**

| Bit[1] | Name | | Type[2] | Description | |
|---|---|---|---|---|---|
| 7 | RES | Reserved for future use | R | 0 | Reserved for future use |
| 6 | RES | Reserved for future use | R | 0 | Reserved for future use |
| 5 | RES | Reserved for future use | R | 0 | Reserved for future use |
| 4 | RSTE | Reset Enabled | R/W | 0 | Reset command is disabled (default) |
| | | | | 1 | Reset command is enabled |
| 3 | RES | Reserved for future use | R | 0 | Reserved for future use |
| 2 | RES | Reserved for future use | R | 0 | Reserved for future use |
| 1 | RES | Reserved for future use | R | 0 | Reserved for future use |
| 0 | $\overline{\text{RDY}}$/BSY | Ready/Busy Status | R | 0 | Device is ready |
| | | | | 1 | Device is busy with an internal operation |

Note: 1. Only bit 4 of Status Register Byte 2 will be modified when using the Write Status Register Byte 2 command

2. R/W = Readable and Writeable
   R = Readable only.

## 11.2 Active Status Interrupt

To simplify the readout of the $\overline{\text{RDY}}$/BSY bit, the Active Status Interrupt command (25h) may be used. It is then not necessary to continuously read the status register, it is sufficient to monitor the value of the SO line. If the SO line is connected to an interrupt line on the host controller, the host controller may be in sleep mode until the SO line indicates that the AT25XE021A is ready for the next command.

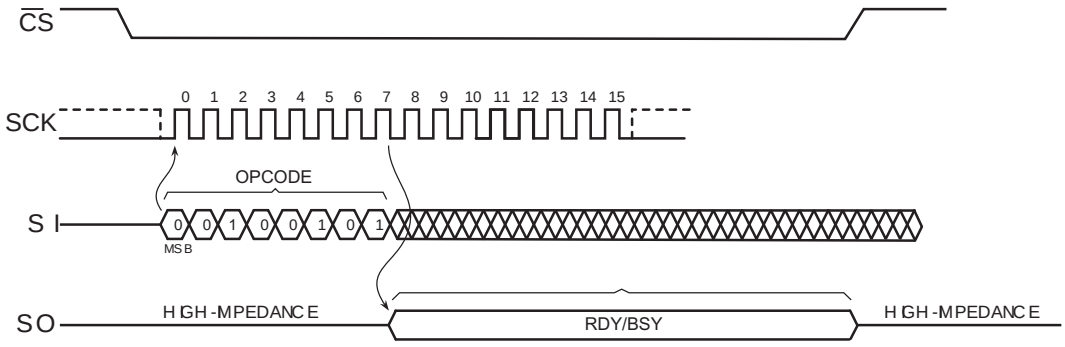
The $\overline{\text{RDY}}$/BSY bit can be read at any time, including during an internally self-timed program or erase operation.

To enable the Active Status Interrupt command, the $\overline{CS}$ pin must first be asserted and the opcode of 25h must be clocked into the device. For SPI Mode3, at least one dummy bit has to be clocked into the device after the last bit of the opcode has been clocked in. (In most cases, this is most easily done by sending a dummy byte to the device.) The value of the SI line after the opcode is clocked in is of no significance to the operation. For SPI Mode 0, this dummy bit (dummy byte) is not required.

The value of $\overline{RDY}$/BSY is then output on the SO line, and is continuously updated by the device for as long as the $\overline{CS}$ pin remains asserted. Additional clocks on the SCK pin are not required. If the $\overline{RDY}$/BSY bit changes from 1 to 0 while the $\overline{CS}$ pin is asserted, the SO line will change from 1 to 0. (The $\overline{RDY}$/BSY bit cannot change from 0 to 1 during an operation, so if the SO line already is 0, it will not change.)

Deasserting the $\overline{CS}$ pin will terminate the Active Status Interrupt 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.

**Figure 11-2. Active Status Interrupt**




## 11.3 Write Status Register

The Write Status Register command is used to modify the SPRL bit of the Status Register and/or to perform a Global Protect or Global Unprotect operation. Before the Write Status Register command can be issued, the Write Enable command must have been previously issued to set the WEL bit in the Status Register to a logical “1”.

To issue the Write Status Register command, the $\overline{CS}$ pin must first be asserted and the opcode of 01h must be clocked into the device followed by one byte of data. The one byte of data consists of the SPRL bit value, a don’t care bit, four data bits to denote whether a Global Protect or Unprotect should be performed, and two additional don’t care bits (see Table 11-3). Any additional data bytes that are sent to the device will be ignored. When the $\overline{CS}$ pin is deasserted, the SPRL bit in the Status Register will be modified, and the WEL bit in the Status Register will be reset back to a logical “0”. The values of bits 5, 4, 3, and 2 and the state of the SPRL bit before the Write Status Register command was executed (the prior state of the SPRL bit) will determine whether or not a Global Protect or Global Unprotect will be perfomed. Please refer to the “Global Protect/Unprotect” section on page 21 for more details.

The complete one byte of data must be clocked into the device before the $\overline{CS}$ pin is deasserted; otherwise, the device will abort the operation, the state of the SPRL bit will not change, no potential Global Protect or Unprotect will be performed, and the WEL bit in the Status Register will be reset back to the logical “0” state.

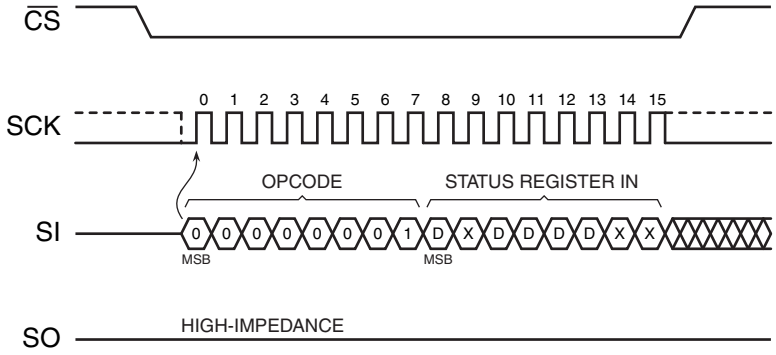
If the $\overline{WP}$ pin is asserted, then the SPRL bit can only be set to a logical “1”. If an attempt is made to reset the SPRL bit to a logical “0” while the $\overline{WP}$ pin is asserted, then the Write Status Register command will be ignored, and the WEL bit in the

Status Register will be reset back to the logical “0” state. In order to reset the SPRL bit to a logical “0”, the $\overline{WP}$ pin must be deasserted.

**Table 11-3. Write Status Register Format**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| SPRL | X | Global Protect/Unprotect | | | | X | X |

**Figure 11-3. Write Status Register**




## 11.4 Write Status Register Byte 2

The Write Status Register Byte 2 command is used to modify the RSTE. Using the Write Status Register Byte 2 command is the only way to modify the RSTE in the Status Register during normal device operation. Before the Write Status Register Byte 2 command can be issued, the Write Enable command must have been previously issued to set the WEL bit in the Status Register to a Logical 1.

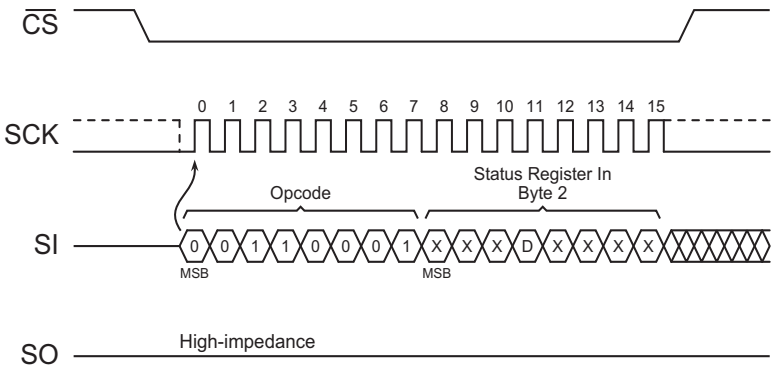
To issue the Write Status Register Byte 2 command, the $\overline{CS}$ pin must first be asserted and then the opcode 31h must be clocked into the device followed by one byte of data. The one byte of data consists of three don’t-care bits, the RSTE bit value, and four additional don’t-care bits (see Table 11-4). Any additional data bytes sent to the device will be ignored. When the $\overline{CS}$ pin is deasserted, the RSTE bit in the Status Register will be modified, and the WEL bit in the Status Register will be reset back to a Logical 0.

The complete one 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 will abort the operation, the state of the RSTE bit will not change, and the WEL bit in the Status Register will be reset back to the Logical 0 state.

**Table 11-4. Write Status Register Byte 2 Format**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| X | X | X | RSTE | X | X | X | X |

**Figure 11-4. Write Status Register Byte 2**




# 12. Other Commands and Functions

## 12.1 Read Manufacturer and Device ID

Identification information can be read from the device to enable systems to electronically query and identify the device while it is in system. The identification method and the command opcode comply with the JEDEC standard for “Manufacturer and Device ID Read Methodology for SPI Compatible Serial Interface Memory Devices”. The type of information that can be read from the device includes the JEDEC defined Manufacturer ID, the vendor specific Device ID, and the vendor specific Extended Device Information.

Since not all Flash devices are capable of operating at very high clock frequencies, applications should be designed to read the identification information from the devices at a reasonably low clock frequency to ensure all devices used in the application can be identified properly. Once the identification process is complete, the application can increase the clock frequency to accommodate specific Flash devices that are capable of operating at the higher clock frequencies.

To read the identification information, the $\overline{CS}$ pin must first be asserted and the opcode of 9Fh must be clocked into the device. After the opcode has been clocked in, the device will begin outputting the identification data on the SO pin during the subsequent clock cycles. The first byte that will be output will be the Manufacturer ID followed by two bytes of Device ID information. The fourth byte output will be the Extended Device Information String Length, which will be 00h indicating that no Extended Device Information follows. After the Extended Device Information String Length byte is output, the SO pin will go into a high-impedance state; therefore, additional clock cycles will have no affect on the SO pin and no data will be output. As indicated in the JEDEC standard, reading the Extended Device Information String Length and any subsequent data is optional.Deasserting the $\overline{CS}$ pin will terminate the Manufacturer and Device ID read 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.

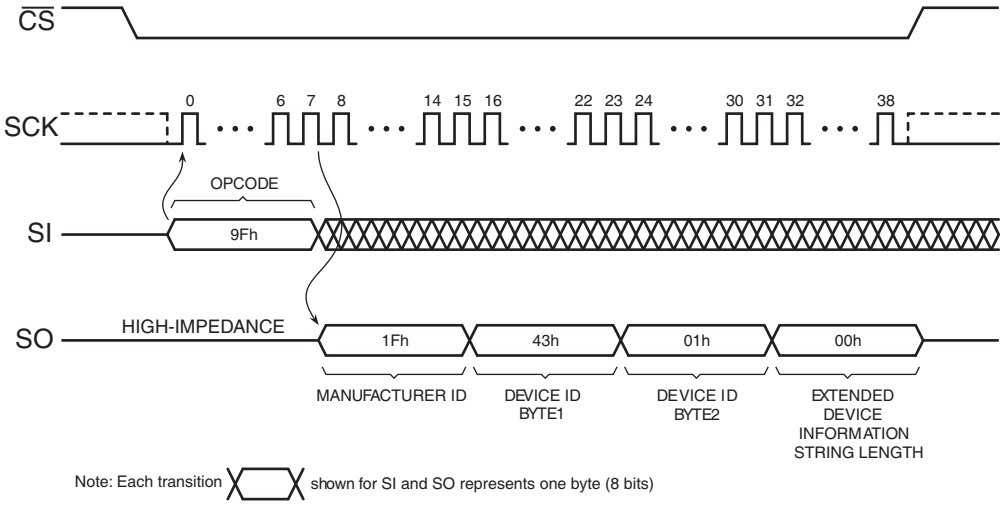
**Table 12-1. Manufacturer and Device ID Information**

| Byte No. | Data Type | Value |
|---|---|---|
| 1 | Manufacturer ID | 1Fh |
| 2 | Device ID (Part 1) | 43h |
| 3 | Device ID (Part 2) | 01h |
| 4 | Extended Device Information String Length | 00h |

**Table 12-2. Manufacturer and Device ID Details**

| Data Type | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Hex Value | Details |
|---|---|---|---|---|---|---|---|---|---|---|
| Manufacturer ID | JEDEC Assigned Code | | | | | | | | 1Fh | JEDEC Code: 0001 1111 (1Fh for Adesto) |
| | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | | |
| Device ID (Part 1) | Family Code | | | Density Code | | | | | 43h | Family Code: 010 (AT25F/AT25XExxx series)<br>Density Code: 00011 (2-Mbit) |
| | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | | |
| Device ID (Part 2) | Sub Code | | | Product Version Code | | | | | 01h | Sub Code: 000 (Standard series)<br>Product Version:00001 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | |

**Figure 12-1. Read Manufacturer and Device ID**




## 12.2 Deep Power-Down

During normal operation, the device will be placed in the standby mode to consume less power as long as the $\overline{CS}$ pin remains deasserted and no internal operation is in progress. The Deep Power-Down command offers the ability to place the device into an even lower power consumption state called the Deep Power-Down mode.

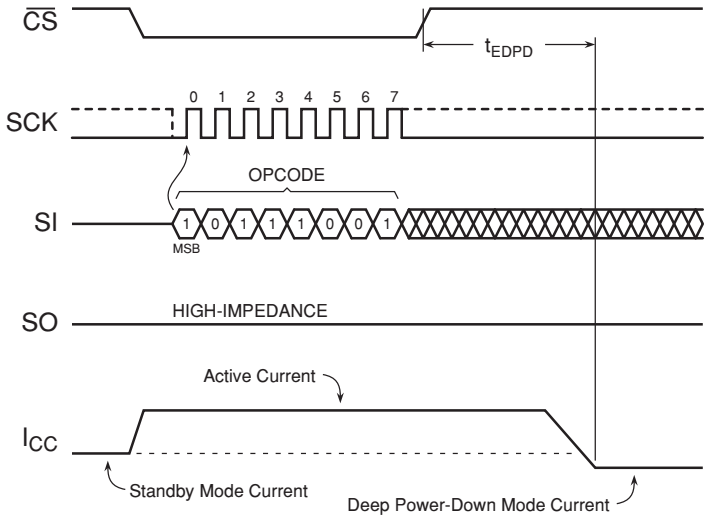
When the device is in the Deep Power-Down mode, all commands including the Read Status Register command will be ignored with the exception of the Resume from Deep Power-Down command. Since all commands will be ignored, the mode can be used as an extra protection mechanism against program and erase operations.

Entering the Deep Power-Down mode is accomplished by simply asserting the $\overline{CS}$ pin, clocking in the opcode of B9h, and then deasserting the $\overline{CS}$ pin. Any additional data clocked into the device after the opcode will be ignored. When the $\overline{CS}$ pin is deasserted, the device will enter the Deep Power-Down mode within the maximum time of $t_{EDPD}$.

The complete opcode must be clocked in before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on an even byte boundary (multiples of eight bits); otherwise, the device will abort the operation and return to the standby mode once the $\overline{CS}$ pin is deasserted. In addition, the device will default to the standby mode after a power-cycle.

The Deep Power-Down command will be ignored if an internally self-timed operation such as a program or erase cycle is in progress. The Deep Power-Down command must be reissued after the internally self-timed operation has been completed in order for the device to enter the Deep Power-Down mode.

**Figure 12-2. Deep Power-Down**



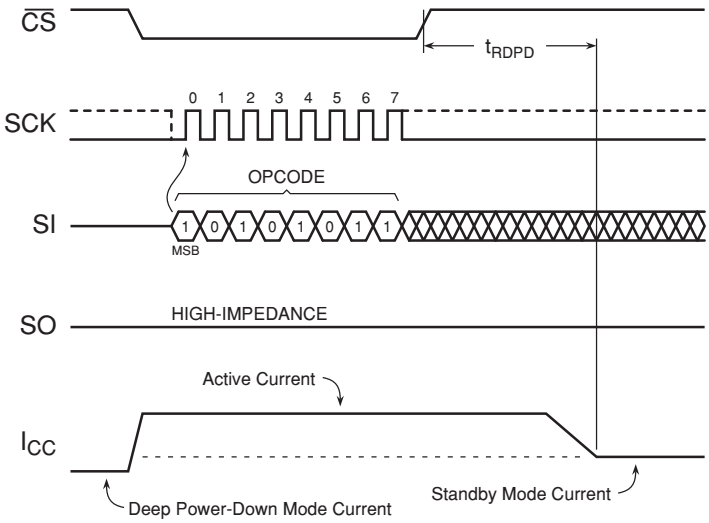

## 12.3 Resume from Deep Power-Down

In order to exit the Deep Power-Down mode and resume normal device operation, the Resume from Deep Power-Down command must be issued. The Resume from Deep Power-Down command is the only command that the device will recognized while in the Deep Power-Down mode.

To resume from the Deep Power-Down mode, the $\overline{CS}$ pin must first be asserted and opcode of ABh must be clocked into the device. Any additional data clocked into the device after the opcode will be ignored. When the $\overline{CS}$ pin is deasserted, the device will exit the Deep Power-Down mode within the maximum time of $t_{RDPD}$ and return to the standby mode. After the device has returned to the standby mode, normal command operations such as Read Array can be resumed.

If the complete opcode is not clocked in before the $\overline{CS}$ pin is deasserted, or if the $\overline{CS}$ pin is not deasserted on an even byte boundary (multiples of eight bits), then the device will abort the operation and return to the Deep Power-Down mode.

**Figure 12-3. Resume from Deep Power-Down**

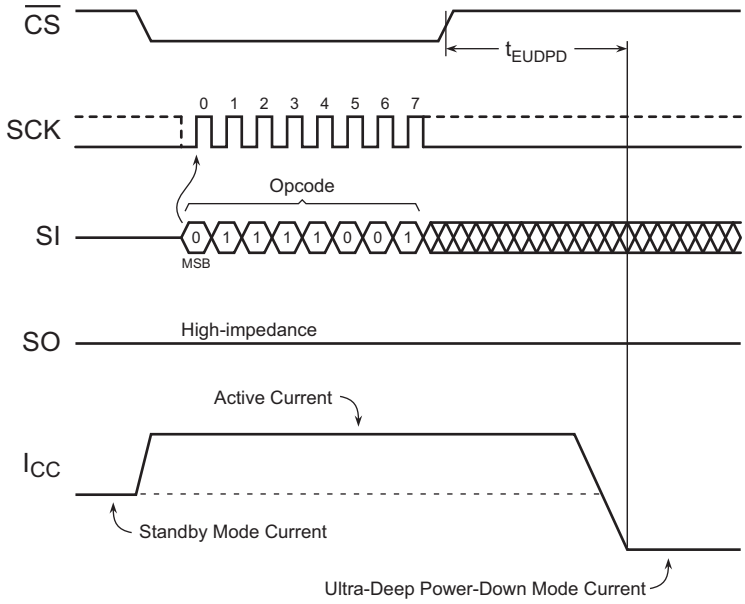


## 12.4 Ultra-Deep Power-Down

The Ultra-Deep Power-Down mode allows the device to further reduce its energy consumption compared to the existing standby and Deep Power-Down modes by shutting down additional internal circuitry. When the device is in the Ultra-Deep Power-Down mode, all commands including the Status Register Read and Resume from Deep Power-Down commands will be ignored. Since all commands will be ignored, the mode can be used as an extra protection mechanism against inadvertent or unintentional program and erase operations. Entering the Ultra-Deep Power-Down mode is accomplished by simply asserting the $\overline{CS}$ pin, clocking in the opcode 79h, and then deasserting the $\overline{CS}$ pin. Any additional data clocked into the device after the opcode will be ignored. When the $\overline{CS}$ pin is deasserted, the device will enter the Ultra-Deep Power-Down mode within the maximum time of $t_{EUDPD}$

The complete opcode must be clocked in before the $\overline{CS}$ pin is deasserted; otherwise, the device will abort the operation and return to the standby mode once the $\overline{CS}$ pin is deasserted. In addition, the device will default to the standby mode after a power cycle. The Ultra-Deep Power-Down command will be ignored if an internally self-timed operation such as a program or erase cycle is in progress.

**Figure 12-4. Ultra-Deep Power-Down**



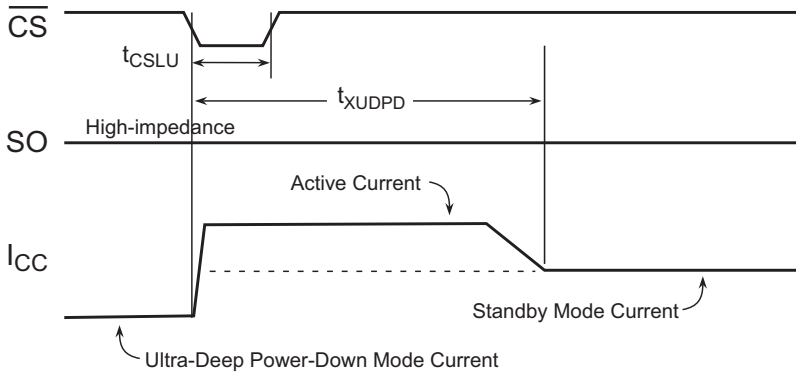

## 12.5 Exit Ultra-Deep Power-Down

To exit from the Ultra-Deep Power-Down mode, any one of three operations can be performed:

**Chip Select Toggle**

The $\overline{CS}$ pin must simply be pulsed by asserting the $\overline{CS}$ pin, waiting the minimum necessary $t_{CSLU}$ time, and then deasserting the $\overline{CS}$ pin again. To facilitate simple software development, a dummy byte opcode can also be entered while the $\overline{CS}$ pin is being pulsed; the dummy byte opcode is simply ignored by the device in this case. After the $\overline{CS}$ pin has been deasserted, the device will exit from the Ultra-Deep Power-Down mode and return to the standby mode within a maximum time of $t_{XUDPD}$ If the $\overline{CS}$ pin is reasserted before the $t_{XUDPD}$ time has elapsed in an attempt to start a new operation, then that operation will be ignored and nothing will be performed.

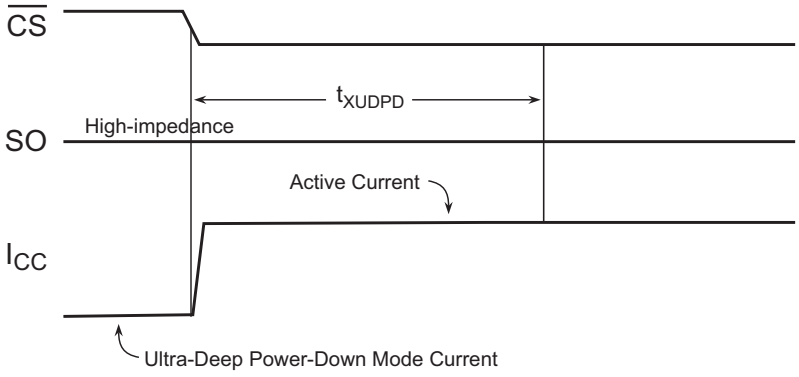
**Figure 12-5. Exit Ultra-Deep Power-Down (Chip Select Toggle)**




### Chip Select Low

By asserting the $\overline{CS}$ pin, waiting the minimum necessary $t_{XUDPD}$ time, and then clocking in the first bit of the next Opcode command cycle. If the first bit of the next command is clocked in before the $t_{XUDPD}$ time has elapsed, the device will exit Ultra Deep Power Down, however the intended operation will be ignored.

**Figure 12-6. Exit Ultra-Deep Power-Down (Chip Select Low)**




### Power Cycling

The device can also exit the Ultra Deep Power Mode by power cycling the device. The system must wait for the device to return to the standby mode before normal command operations can be resumed. Upon recovery from Ultra Deep Power Down all internal registers will be at there Power-On default state.

## 12.6 Hold

The $\overline{HOLD}$ pin is used to pause the serial communication with the device without having to stop or reset the clock sequence. The Hold mode, however, does not have an affect on any internally self-timed operations such as a program or erase cycle. Therefore, if an erase cycle is in progress, asserting the $\overline{HOLD}$ pin will not pause the operation, and the erase cycle will continue until it is finished.

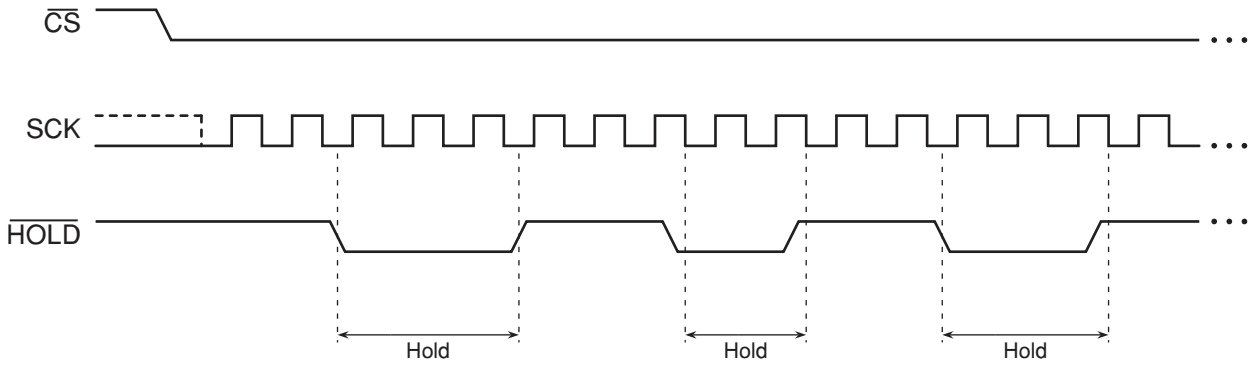
The Hold mode can only be entered while the $\overline{CS}$ pin is asserted. The Hold mode is activated simply by asserting the $\overline{HOLD}$ pin during the SCK low pulse. If the $\overline{HOLD}$ pin is asserted during the SCK high pulse, then the Hold mode won't be started until the beginning of the next SCK low pulse. The device will remain in the Hold mode as long as the $\overline{HOLD}$ pin and $\overline{CS}$ pin are asserted.

While in the Hold mode, the SO pin will be in a high-impedance state. In addition, both the SI pin and the SCK pin will be ignored. The $\overline{WP}$ pin, however, can still be asserted or deasserted while in the Hold mode.

To end the Hold mode and resume serial communication, the $\overline{\text{HOLD}}$ pin must be deasserted during the SCK low pulse. If the $\overline{\text{HOLD}}$ pin is deasserted during the SCK high pulse, then the Hold mode won't end until the beginning of the next SCK low pulse.

If the $\overline{\text{CS}}$ pin is deasserted while the $\overline{\text{HOLD}}$ pin is still asserted, then any operation that may have been started will be aborted, and the device will reset the WEL bit in the Status Register back to the logical "0" state.

**Figure 12-7. Hold Mode**




## 12.7 Reset

In some applications, it may be necessary to prematurely terminate a program or erase operation rather than wait the hundreds of microseconds or milliseconds necessary for the program or erase operation to complete normally. The Reset command allows a program or erase operation in progress to be ended abruptly and returns the device to an idle state. Since the need to reset the device is immediate, the Write Enable command does not need to be issued prior to the Reset command. Therefore, the Reset command operates independently of the state of the WEL bit in the Status Register.

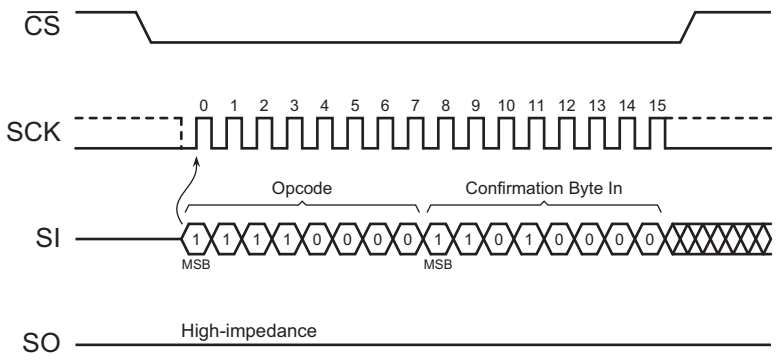
The Reset command can be executed only if the command has been enabled by setting the Reset Enabled (RSTE) bit in the Status Register to a Logical 1 using write status register byte 2 command 31h. This command should be entered before a program command is entered. If the Reset command has not been enabled (the RSTE bit is in the Logical 0 state), then any attempts at executing the Reset command will be ignored.

To perform a Reset, the $\overline{\text{CS}}$ pin must first be asserted, and then the opcode F0h must be clocked into the device. No address bytes need to be clocked in, but a confirmation byte of D0h must be clocked into the device immediately after the opcode. Any additional data clocked into the device after the confirmation byte will be ignored. When the $\overline{\text{CS}}$ pin is deasserted, the program operation currently in progress will be terminated within a time of $t_{SWRST}$. Since the program or erase operation may not complete before the device is reset, the contents of the page being programmed or erased cannot be guaranteed to be valid.

The Reset command has no effect on the states of the Configuration Register or RSTE bit in the Status Register. Apart from Sequential Programming, the WEL bit will be reset back to its default state.

The complete opcode and confirmation byte must be clocked into the device before the $\overline{\text{CS}}$ pin is deasserted, and the $\overline{\text{CS}}$ pin must be deasserted on an even byte boundary (multiples of eight bits); otherwise, no Reset operation will be performed.

**Table 12-3. Reset**




# 13. Electrical Specifications

## 13.1 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 . . . . . -0.6V to +4.1V |
| All Output Voltages with Respect to Ground . . . . . -0.6V to $V_{CC}$ + 0.5V |

*Notice: Stresses beyond those listed under “Absolute Maximum Ratings” may 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 may affect device reliability.

## 13.2 DC and AC Operating Range

| | | AT25XE021A |
|---|---|---|
| Operating Temperature (Case) | Ind. | -40°C to 85°C |
| $V_{CC}$ Power Supply | | 1.65V to 3.6V |

## 13.3 DC Characteristics

| Symbol | Parameter | Condition | 1.65V to 3.6V | | | 2.3V to 3.6V | | | Units |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Typ | Max | Min | Typ | Max | |
| $I_{UDPD}$ | Ultra-Deep Power-Down Current | $\overline{CS}$ = $V_{CC}$. All other inputs at 0V or $V_{CC}$ | | 0.2 | 0.6 | | 0.35 | 0.6 | µA |
| $I_{DPD}$ (1) | Deep Power-Down Current | $\overline{CS}$, $\overline{HOLD}$, $\overline{WP}$ = $V_{IH}$ All inputs at CMOS levels | | 4.5 | 15 | | 5 | 15 | µA |
| $I_{SB}$ | Standby Current | $\overline{CS}$, $\overline{HOLD}$, $\overline{WP}$ = $V_{IH}$ All inputs at CMOS levels | | 25 | 40 | | 25 | 40 | µA |
| $I_{CC1}$ (2)(3) | Active Current, Low Power Read (03h, 0Bh) Operation | f = 1MHz; $I_{OUT}$ = 0mA | | 3 | 4.5 | | 3.5 | 4.5 | mA |
| | | f = 20MHz; $I_{OUT}$ = 0mA | | 3.5 | 5.25 | | 4.2 | 5.25 | mA |
| $I_{CC2}$ (2) (3) | Active Current, Read Operation | f = 50MHz; $I_{OUT}$ = 0mA | | 3.5 | 5.25 | | 4.2 | 5.25 | mA |
| | | f = 85MHz; $I_{OUT}$ = 0mA | | 3.5 | 5.25 | | 4.2 | 5.25 | mA |
| $I_{CC3}$ (2)(3) | Active Current, Program Operation | $\overline{CS}$ = $V_{CC}$ | | 9 | 10.5 | | 9 | 10.5 | mA |
| $I_{CC4}$ (2)(3) | Active Current, Erase Operation | $\overline{CS}$ = $V_{CC}$ | | 8 | 11 | | 8 | 11 | mA |
| $I_{LI}$ | Input Load Current | All inputs at CMOS levels | | | 1 | | | 1 | µA |
| $I_{LO}$ | Output Leakage Current | All inputs at CMOS levels | | | 1 | | | 1 | µA |
| $V_{IL}$ | Input Low Voltage | | | | $V_{CC}$ x 0.2 | | | $V_{CC}$ x 0.3 | V |
| $V_{IH}$ | Input High Voltage | | $V_{CC}$ x 0.8 | | | $V_{CC}$ x 0.7 | | | V |
| $V_{OL}$ | Output Low Voltage | $I_{OL}$ = 100µA | | | 0.2 | | | 0.4 | V |
| $V_{OH}$ | Output High Voltage | $I_{OH}$ = -100µA | $V_{CC}$ - 0.2V | | | $V_{CC}$ - 0.2V | | | V |

1. Max. specification is 20µA @ 85°C.
2. Typical values measured at 1.8V @ 25°C for the 1.65V to 3.6V range.
3. Typical values measured at 3.0V @ 25°C for the 2.3V to 3.6V range.

## 13.4 AC Characteristics - Maximum Clock Frequencies

| Symbol | Parameter | 1.65V to 3.6V | | | 2.3V to 3.6V | | | Units |
|---|---|---|---|---|---|---|---|---|
| | | Min | Typ | Max | Min | Typ | Max | |
| $f_{CLK}$ | Maximum Clock Frequency for All Operations (including 0Bh opcode) | | | 70 | | | 70 | MHz |
| $f_{RDLF}$ | Maximum Clock Frequency for 03h Opcode (Read Array – Low Frequency) | | | 25 | | | 25 | MHz |
| $f_{RDDO}$ | Maximum Clock Frequency for 3B Opcode | | | 40 | | | 40 | MHz |

## 13.5 AC Characteristics – All Other Parameters

| Symbol | Parameter | 1.65V to 3.6V | | | 2.3V to 3.6V | | | Units |
|---|---|---|---|---|---|---|---|---|
| | | Min | Typ | Max | Min | Typ | Max | |
| $t_{CLKH}$ | Clock High Time | 4.5 | | | 4.5 | | | ns |
| $t_{CLKL}$ | Clock Low Time | 4 | | | 4 | | | ns |
| $t_{CLKR}$[(1)] | Clock Rise Time, Peak-to-Peak (Slew Rate) | 0.1 | | | 0.1 | | | V/ns |
| $t_{CLKF}$[(1)] | Clock Fall Time, Peak-to-Peak (Slew Rate) | 0.1 | | | 0.1 | | | V/ns |
| $t_{CSH}$ | Chip Select High Time | 35 | | | 25 | | | ns |
| $t_{CSLS}$ | Chip Select Low Setup Time (relative to Clock) | 6 | | | 5 | | | ns |
| $t_{CSLH}$ | Chip Select Low Hold Time (relative to Clock) | 6 | | | 5 | | | ns |
| $t_{CSHS}$ | Chip Select High Setup Time (relative to Clock) | 6 | | | 5 | | | ns |
| $t_{CSHH}$ | Chip Select High Hold Time (relative to Clock) | 6 | | | 5 | | | ns |
| $t_{DS}$ | Data In Setup Time | 2 | | | 2 | | | ns |
| $t_{DH}$ | Data In Hold Time | 1 | | | 1 | | | ns |
| $t_{DIS}$[(1)] | Output Disable Time | | | 8 | | | 6 | ns |
| $t_V$ | Output Valid Time | | | 7.5 | | | 6 | ns |
| $t_{OH}$ | Output Hold Time | 0 | | | 0 | | | ns |
| $t_{HLS}$ | $\overline{\text{HOLD}}$ Low Setup Time (relative to Clock) | 6 | | | 5 | | | ns |
| $t_{HLH}$ | $\overline{\text{HOLD}}$ Low Hold Time (relative to Clock) | 6 | | | 5 | | | ns |
| $t_{HHS}$ | $\overline{\text{HOLD}}$ High Setup Time (relative to Clock) | 6 | | | 5 | | | ns |
| $t_{HHH}$ | $\overline{\text{HOLD}}$ High Hold Time (relative to Clock) | 6 | | | 5 | | | ns |
| $t_{HLQZ}$[(1)] | $\overline{\text{HOLD}}$ Low to Output High-Z | | | 7 | | | 6 | ns |
| $t_{HHQX}$[(1)] | $\overline{\text{HOLD}}$ High to Output Low-Z | | | 7 | | | 6 | ns |
| $t_{WPS}$[(1)(1)] | Write Protect Setup Time | 20 | | | 20 | | | ns |
| $t_{WPH}$[(1)(1)] | Write Protect Hold Time | 100 | | | 100 | | | ns |
| $t_{EDPD}$[(1)] | Chip Select High to Deep Power-Down | | | 3 | | | 3 | µs |

## 13.5 AC Characteristics – All Other Parameters

| Symbol | Parameter | 1.65V to 3.6V | | | 2.3V to 3.6V | | | Units |
|---|---|---|---|---|---|---|---|---|
| | | Min | Typ | Max | Min | Typ | Max | |
| $t_{EUDPD}$ (1) | Chip Select High to Ultra Deep Power-Down | | | 3 | | | 3 | µs |
| $t_{SWRST}$ | Software Reset Time | | | 60 | | | 60 | µs |
| $t_{CSLU}$ | Minimum Chip Select Low to Exit Ultra Deep Power-Down | 20 | | | 20 | | | ns |
| $t_{XUDPD}$ | Exit Ultra Deep Power-Down Time | 70 | | | 70 | | | µs |
| $t_{RDPD}$ (1) | Chip Select High to Standby Mode | | | 8 | | | 8 | µs |

Notes: 1. Not 100% tested (value guaranteed by design and characterization).

## 13.6 Program and Erase Characteristics

| Symbol | Parameter | | 1.65V-3.6V | | | 2.3V-3.6V | | | Units |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Typ | Max | Min | Typ | Max | |
| $t_{PP}$ (1) | Page Program Time (256 Bytes) | | | 2 | 5 | | 2 | 2.5 | ms |
| $t_{BP}$ | Byte Program Time | | | 8 | | | 8 | | µs |
| $t_{PE}$ | Page Erase Time | | | 6 | 20 | | 6 | 15 | ms |
| $t_{BLKE}$ (1) | Block Erase Time | 4 Kbytes | | 45 | 100 | | 45 | 60 | ms |
| | | 32 Kbytes | | 360 | 600 | | 360 | 500 | |
| | | 64 Kbytes | | 720 | 1200 | | 720 | 1000 | |
| $t_{CHPE}$ (1)(2) | Chip Erase Time | | | 2.4 | 4.8 | | 2.4 | 4.0 | sec |
| $t_{OTPP}$ (1) | OTP Security Register Program Time | | | 400 | 950 | | 400 | 950 | µs |
| $t_{WRSR}$ | Write Status Register Time | | | | 200 | | | 200 | ns |

Note: 1. Maximum values indicate worst-case performance after 100,000 erase/program cycles.
2. Not 100% tested (value guaranteed by design and characterization).

# 14. Power-On/Reset State

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

## 14.1 Power-Up/Power-Down Voltage and Timing Requirements

During power-up, the device must not be READ for at least the minimum $t_{VCSL}$ time after the supply voltage reaches the minimum $V_{POR}$ level ($V_{POR}$ 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 $V_{cc}$. During this time, all operations are disabled and the device will not respond to any commands.

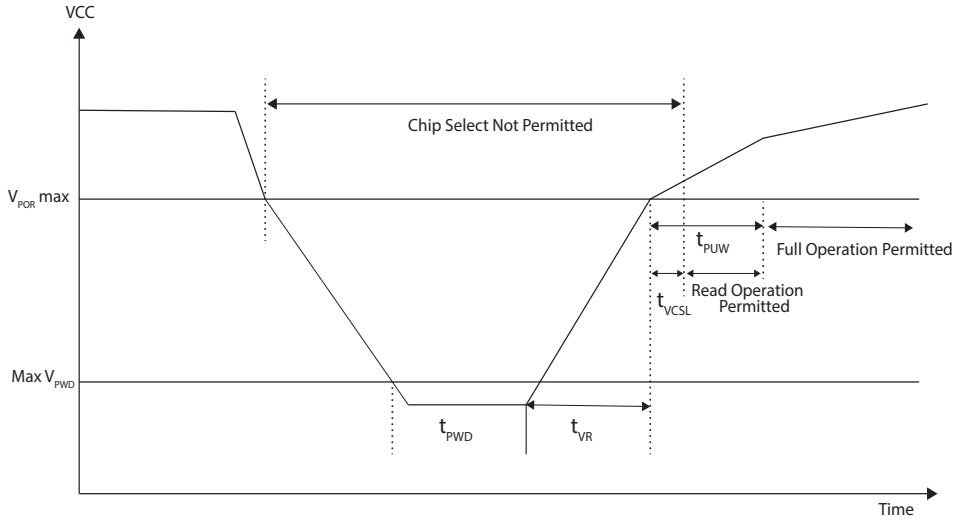
If the first operation to the device after power-up will be a program or erase operation, then the operation cannot be started until the supply voltage reaches the minimum $V_{CC}$ level and an internal device delay has elapsed. This delay will be a maximum time of $t_{PUW}$. After the $t_{PUW}$ time, the device will be in the standby mode if CSB is at logic high or active mode if CSB is at logic low. For the case of Power-down then Power-up operation, or if a power interruption occurs (such that VCC drops below $V_{POR}$ max), the $V_{cc}$ of the Flash device must be maintained below $V_{PWD}$ for at least the minimum specified $T_{PWD}$ time. This is to ensure the Flash device will reset properly after a power interruption.

**Table 14-1. Voltage and Timing Requirements for Power-Up/Power-Down**

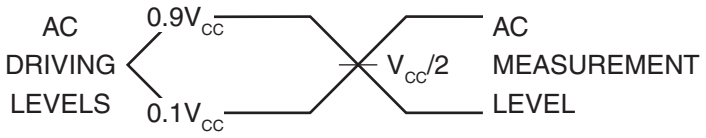
| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $V_{PWD}$ (1) | $V_{CC}$ for device initialization | | 1.0 | V |
| $t_{PWD}$ (1) | Minimum duration for device initialization | 300 | | µs |
| $t_{VCSL}$ | Minimum $V_{CC}$ to chip select low time for Read command | 70 | | µs |
| $t_{VR}$ (1) | $V_{CC}$ rise time | 1 | 500000 | µs/V |
| $V_{POR}$ | Power on reset voltage | 1.45 | 1.6 | V |
| $t_{PUW}$ | Power up delay time before Program or Erase is allowed | | 3 | ms |

1. Not 100% tested (value guaranteed by design and characterization).

**Figure 14-1. Power-Up Timing**

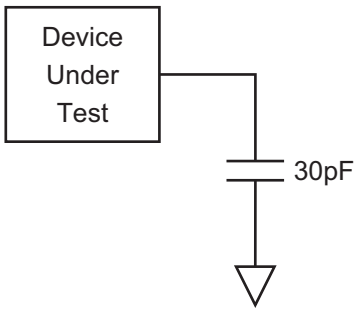


## 14.2 Input Test Waveforms and Measurement Levels




$t_R$, $t_F$ < 2 ns (10% to 90%)

## 14.3 Output Test Load




# 15. AC Waveforms

**Figure 15-1. Serial Input Timing**

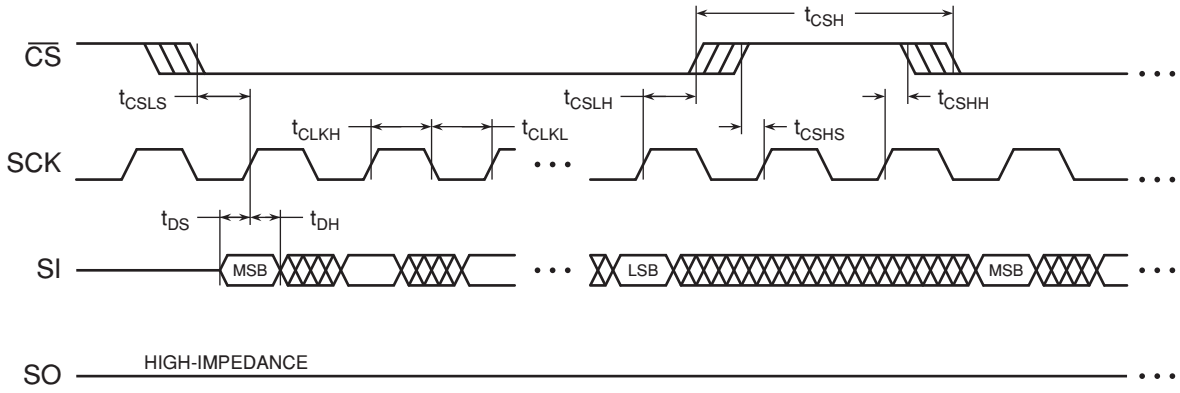



**Figure 15-2. Serial Output Timing**

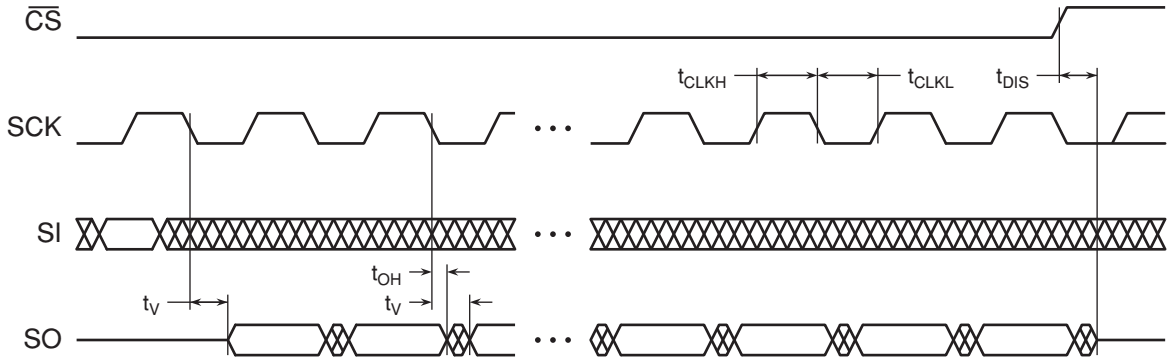


**Figure 15-3. $\overline{WP}$ Timing for Write Status Register Command When BPL = 1**

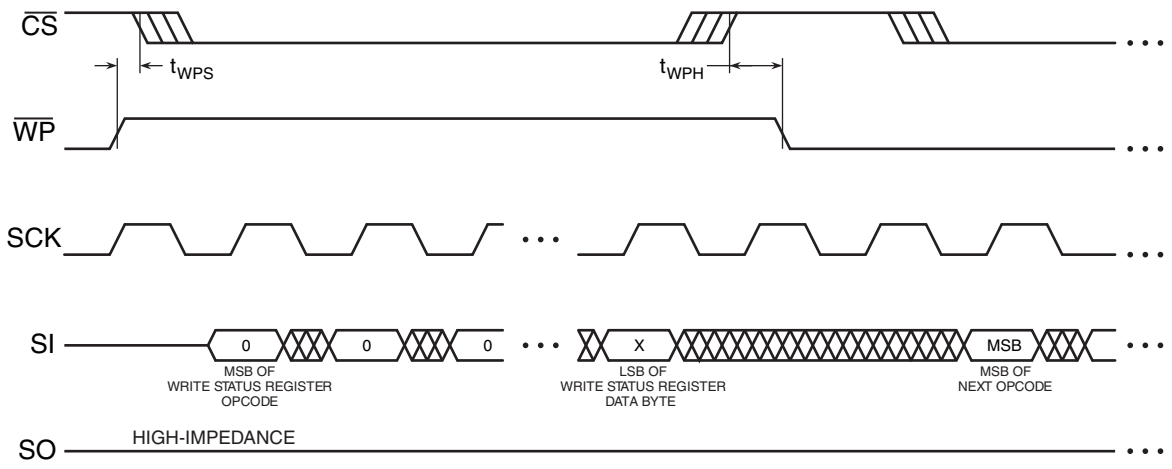



**Figure 15-4. $\overline{HOLD}$ Timing – Serial Input**

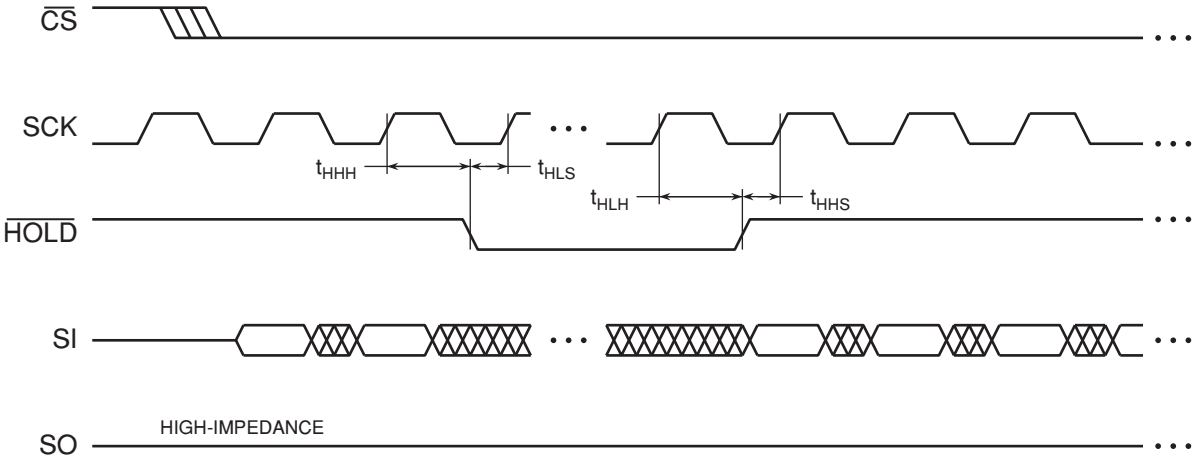
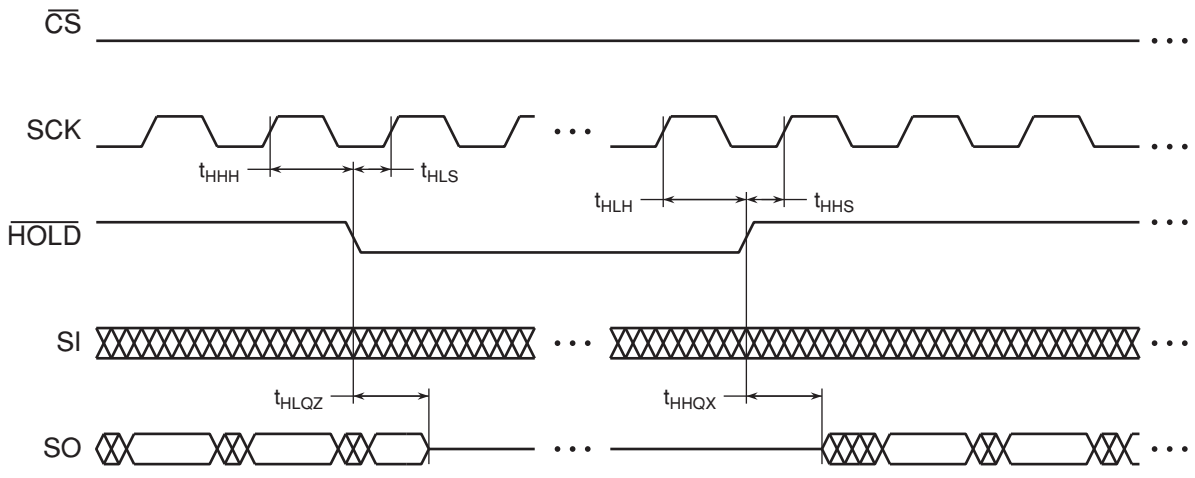
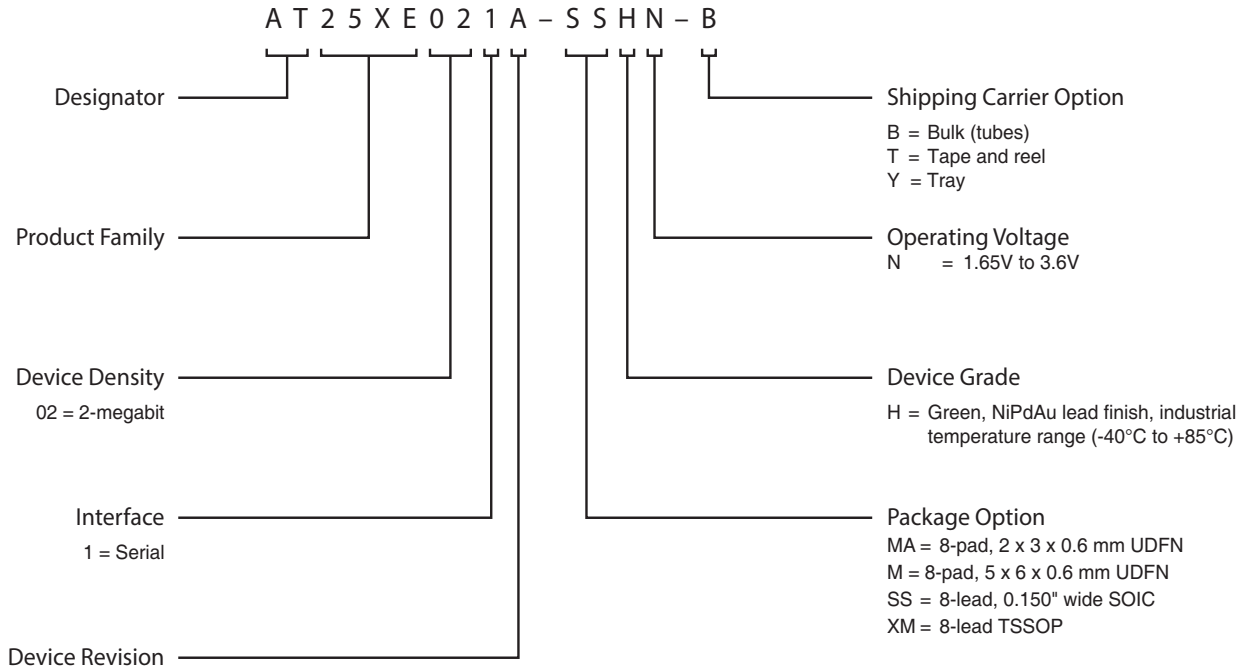



**Figure 15-5. $\overline{HOLD}$ Timing – Serial Output**



# 16. Ordering Information

## 16.1 Ordering Code Detail



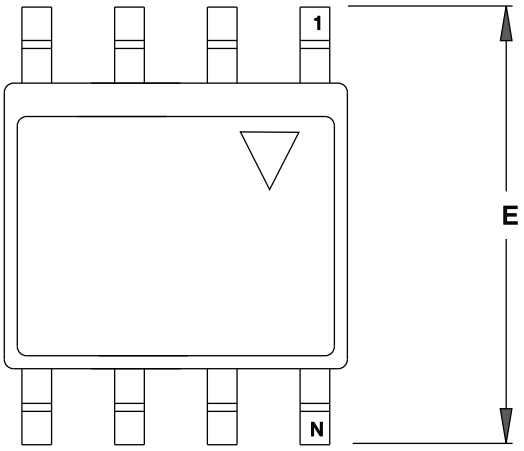

| Ordering Code | Package | Lead Finish | Operating Voltage | Max. Freq. (MHz) | Operation Range |
|---|---|---|---|---|---|
| AT25XE021A-SSHN-B<br>AT25XE021A-SSHN-T | 8S1 | NiPdAu | 1.65V to 3.6V | 70 | Industrial (-40°C to +85°C) |
| AT25XE021A-MHN-Y | 8MA1 | | | | |
| AT25XE021A-MHN-T | | | | | |
| AT25XE021A-MAHN-T | 8MA3 | | | | |
| AT25XE021A-XMHN-B | 8X | | | | |
| AT25XE021A-XMHN-T | | | | | |

Note: The shipping carrier option code is not marked on the devices.

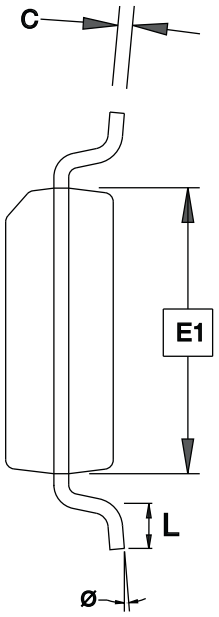
| Package Type | |
|---|---|
| **8S1** | 8-lead, 0.150" Wide, Plastic Gull Wing Small Outline Package (JEDEC SOIC) |
| **8MA1** | 8-pad, 5 x 6 x 0.6mm, Thermally Enhanced Plastic Ultra Thin Dual Flat No-lead (UDFN) |
| **8MA3** | 8-pad, 2 x 3 x 0.6mm, Thermally Enhanced Plastic Ultra Thin Dual Flat No Lead Package (UDFN) |
| **8X** | 8-lead, Thin Shrink Small Outline Package |

# 17. Packaging Information

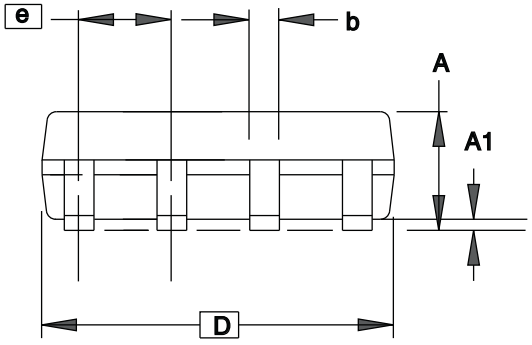
## 17.1 8S1 – JEDEC SOIC




TOP VIEW




END VIEW




SIDE VIEW

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

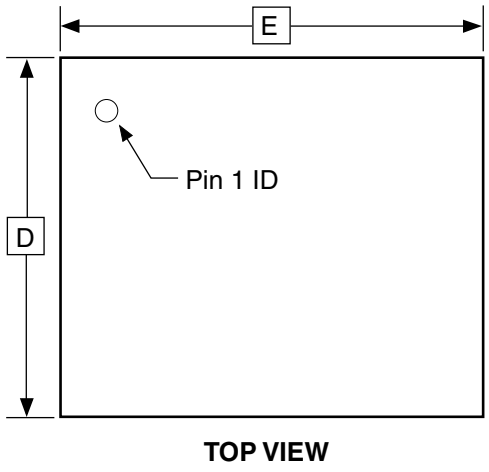
COMMON DIMENSIONS
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 1.35 | – | 1.75 | |
| A1 | 0.10 | – | 0.25 | |
| b | 0.31 | – | 0.51 | |
| C | 0.17 | – | 0.25 | |
| D | 4.80 | – | 5.05 | |
| E1 | 3.81 | – | 3.99 | |
| E | 5.79 | – | 6.20 | |
| e | 1.27 BSC | | | |
| L | 0.40 | – | 1.27 | |
| Ø | 0° | – | 8° | |

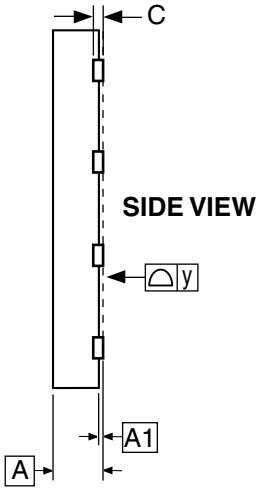
8/20/14

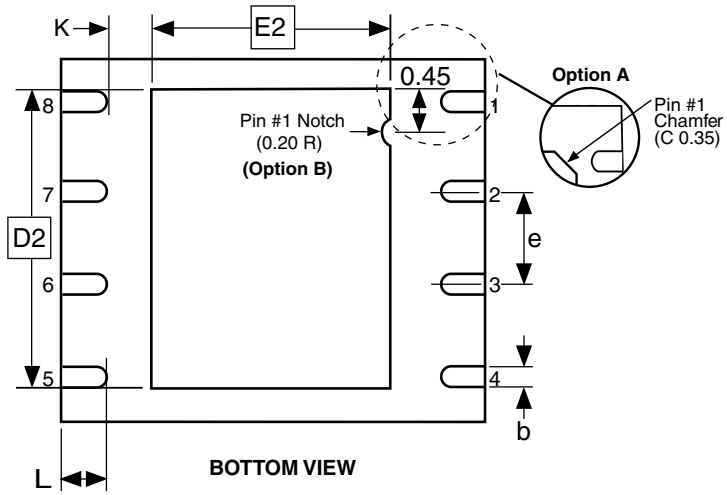
| adesto™ TECHNOLOGIES<br>Package Drawing Contact:<br>contact@adestotech.com | TITLE<br>8S1, 8-lead (0.150"Wide Body), Plastic Gull Wing Small Outline (JEDEC SOIC) | GPC<br>SWB | DRAWING NO.<br>8S1 | REV.<br>G |
|---|---|---|---|---|

## 17.2 8MA1 – 5 x 6 UDFN




TOP VIEW




BOTTOM VIEW

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

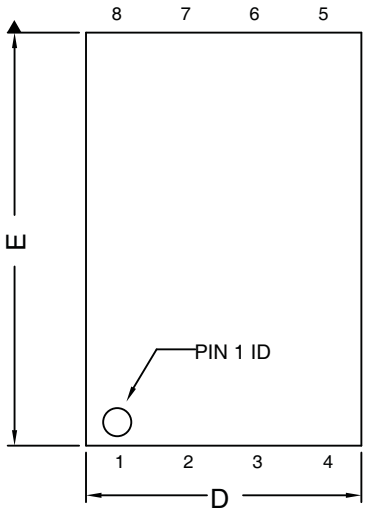
| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 0.45 | 0.55 | 0.60 | |
| A1 | 0.00 | 0.02 | 0.05 | |
| b | 0.35 | 0.40 | 0.48 | |
| C | | 0.152 REF | | |
| D | 4.90 | 5.00 | 5.10 | |
| D2 | 3.80 | 4.00 | 4.20 | |
| E | 5.90 | 6.00 | 6.10 | |
| E2 | 3.20 | 3.40 | 3.60 | |
| e | | 1.27 | | |
| L | 0.50 | 0.60 | 0.75 | |
| y | 0.00 | – | 0.08 | |
| K | 0.20 | – | – | |

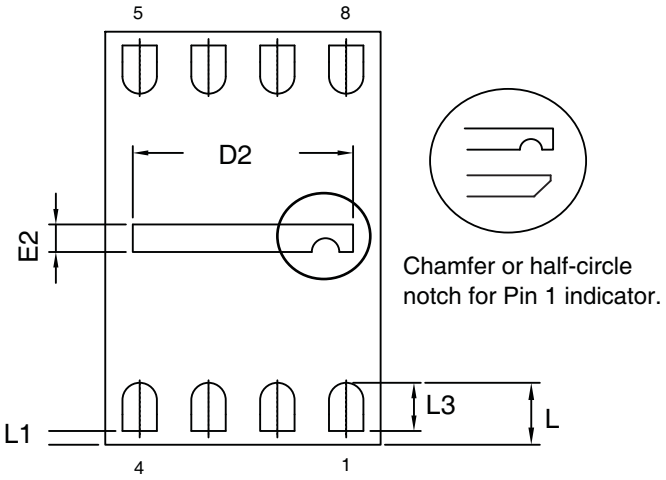
Notes: 1. This package conforms to JEDEC reference MO-229, Saw Singulation.
2. The terminal #1 ID is a Laser-marked Feature.

4/15/08

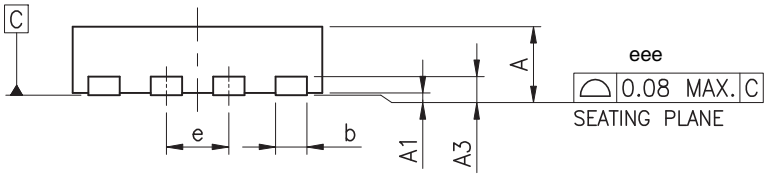
| adesto TECHNOLOGIES **Package Drawing Contact:** contact@adestotech.com | **TITLE** | **GPC** | **DRAWING NO.** | **REV.** |
|---|---|---|---|---|
| | **8MA1,** 8-pad (5 x 6 x 0.6 mm Body), Thermally Enhanced Plastic Ultra Thin Dual Flat No Lead Package (UDFN) | YFG | 8MA1 | D |

## 17.3 8MA3 – 2 x 3 UDFN




Chamfer or half-circle notch for Pin 1 indicator.




COMMON DIMENSIONS
(Unit of Measure = mm)

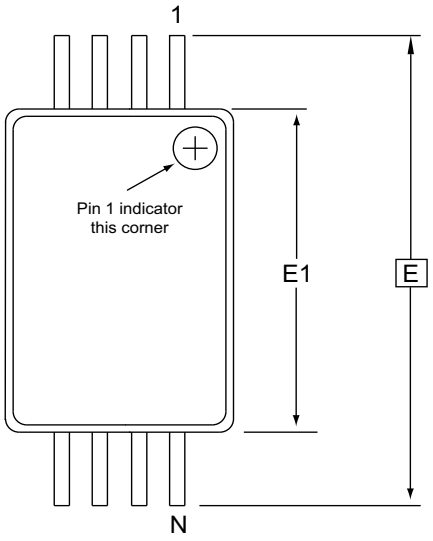
| SYMBOL | MIN | NOM | MAX |
|---|---|---|---|
| A | 0.45 | | 0.60 |
| A1 | 0.00 | | 0.05 |
| A3 | | 0.150 REF | |
| b | 0.20 | | 0.30 |
| D | | 2.00 BSC | |
| D2 | 1.50 | 1.60 | 1.70 |
| E | | 3.00 BSC | |
| E2 | 0.10 | 0.20 | 0.30 |
| e | | 0.50 BSC | |
| L | 0.40 | 0.45 | 0.50 |
| L1 | 0.00 | 0.10 | |
| L3 | 0.30 | | 0.50 |
| eee | – | – | 0.08 |

Notes: 1. All dimensions are in mm. Angles in degrees.
2. Bilateral coplanarity zone applies to the exposed heat sink slug as well as the terminals.

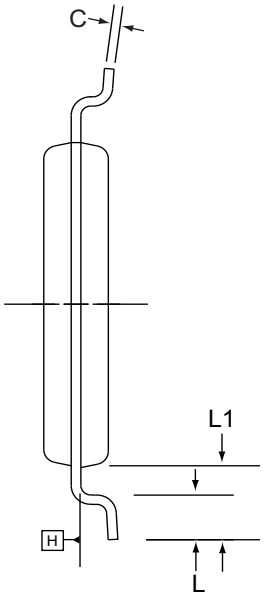
8/26/14

| | TITLE | GPC | DRAWING NO. | REV. |
|---|---|---|---|---|
| adesto TECHNOLOGIES Package Drawing Contact: contact@adestotech.com | 8MA3, 8-pad, 2 x 3 x 0.6 mm Body, 0.5 mm Pitch, 1.6 x 0.2 mm Exposed Pad, Saw Singulated Thermally Enhanced Plastic Ultra Thin Dual Flat No Lead Package (UDFN/USON) | YCQ | 8MA3 | GT |

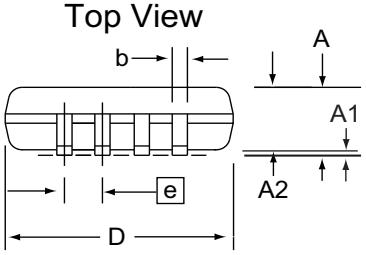
## 17.4 8X-TSSOP




Top View




End View




Side View

Notes:
1. This drawing is for general information only. Refer to JEDEC Drawing MO-153, Variation AA, for proper dimensions, tolerances, datums, etc.
2. Dimension D does not include mold Flash, protrusions or gate burrs. Mold Flash, protrusions and gate burrs shall not exceed 0.15mm (0.006in) per side.
3. Dimension E1 does not include inter-lead Flash or protrusions. Inter-lead Flash and protrusions shall not exceed 0.25mm (0.010in) per side.
4. Dimension b does not include Dambar protrusion. Allowable Dambar protrusion shall be 0.08mm total in excess of the b dimension at maximum material condition. Dambar cannot be located on the lower radius of the foot. Minimum space between protrusion and adjacent lead is 0.07mm.
5. Dimension D and E1 to be determined at Datum Plane H.

COMMON DIMENSIONS
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | - | - | 1.20 | |
| A1 | 0.05 | - | 0.15 | |
| A2 | 0.80 | 1.00 | 1.05 | |
| D | 2.90 | 3.00 | 3.10 | 2, 5 |
| E | | 6.40 BSC | | |
| E1 | 4.30 | 4.40 | 4.50 | 3, 5 |
| b | 0.19 | – | 0.30 | 4 |
| e | | 0.65 BSC | | |
| L | 0.45 | 0.60 | 0.75 | |
| L1 | | 1.00 REF | | |
| C | 0.09 | - | 0.20 | |

12/8/11

| adesto TECHNOLOGIES<br>**Package Drawing Contact:**<br>contact@adestotech.com | **TITLE**<br>8X, 8-lead 4.4mm Body, Plastic Thin Shrink Small Outline Package (TSSOP) | **GPC**<br>TNR | **DRAWING NO.**<br>8X | **REV.**<br>E |
|---|---|---|---|---|

# 18. Revision History

| Revision Level – Release Date | History |
|---|---|
| A – October 2014 | Initial release. |
| B – October 2014 | Updated document status to Preliminary. |
| C –February 2015 | Added $I_{DPD}$ specification footnote. |
| D – May 2015 | Updated AC and DC characteristics. Expanded explanation of Power up/Power down (Section 14). |

## Corporate Office

California | USA
Adesto Headquarters
1250 Borregas Avenue
Sunnyvale, CA 94089
Phone: (+1) 408.400.0578
Email: contact@adestotech.com



© 2015 Adesto Technologies. All rights reserved. / Rev.: DS-25XE021A–061D–5/2015

Adesto®, the Adesto logo, CBRAM®, and DataFlash® are registered trademarks or trademarks of Adesto Technologies. All other marks are the property of their respective owners.

Disclaimer: Adesto Technologies Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Adesto's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Adesto are granted by the Company in connection with the sale of Adesto products, expressly or by implication. Adesto's products are not authorized for use as critical components in life support devices or systems.

Общество с ограниченной ответственностью «МосЧип» ИНН 7719860671 / КПП 771901001
Адрес: 105318, г.Москва, ул.Щербаковская д.3, офис 1107

**Данный компонент на территории Российской Федерации**

**Вы можете приобрести в компании MosChip.**

Для оперативного оформления запроса Вам необходимо перейти по данной ссылке:

http://moschip.ru/get-element

Вы можете разместить у нас заказ для любого Вашего проекта, будь то серийное производство или разработка единичного прибора.

В нашем ассортименте представлены ведущие мировые производители активных и пассивных электронных компонентов.

Нашей специализацией является поставка электронной компонентной базы двойного назначения, продукции таких производителей как XILINX, Intel (ex.ALTERA), Vicor, Microchip, Texas Instruments, Analog Devices, Mini-Circuits, Amphenol, Glenair.

Сотрудничество с глобальными дистрибьюторами электронных компонентов, предоставляет возможность заказывать и получать с международных складов практически любой перечень компонентов в оптимальные для Вас сроки.

На всех этапах разработки и производства наши партнеры могут получить квалифицированную поддержку опытных инженеров.

Система менеджмента качества компании отвечает требованиям в соответствии с ГОСТ Р ИСО 9001, ГОСТ РВ 0015-002 и ЭС РД 009

**Офис по работе с юридическими лицами:**

105318, г.Москва, ул.Щербаковская д.3, офис 1107, 1118, ДЦ «Щербаковский»

Телефон: +7 495 668-12-70 (многоканальный)

Факс: +7 495 668-12-70 (доб.304)

E-mail: info@moschip.ru

Skype отдела продаж:
moschip.ru
moschip.ru_4
moschip.ru_6
moschip.ru_9