adesto® TECHNOLOGIES

# AT25SF321

## 32-Mbit, 2.5V Minimum SPI Serial Flash Memory with Dual-I/O and Quad-IO Support

## Features

- Single 2.5V - 3.6V Supply
- Serial Peripheral Interface (SPI) Compatible
  - Supports SPI Modes 0 and 3
  - Supports Dual and Quad Output Read
- 104MHz 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 Blocks via $\overline{WP}$ Pin
- 3 Protected Programmable Security Register Pages
- Flexible Programming
  - Byte/Page Program (1 to 256 Bytes)
- Fast Program and Erase Times
  - 0.7ms Typical Page Program (256 Bytes) Time
  - 70ms Typical 4-Kbyte Block Erase Time
  - 300ms Typical 32-Kbyte Block Erase Time
  - 600ms Typical 64-Kbyte Block Erase Time
- JEDEC Standard Manufacturer and Device ID Read Methodology
- Low Power Dissipation
  - 2µA Deep Power-Down Current (Typical)
  - 10µA Standby current (Typical)
  - 4mA 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 and 208-mil)
  - 8-pad Ultra Thin DFN (5 x 6 x 0.6 mm)
  - Die in Wafer Form

## Description

The Adesto® AT25SF321 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 AT25SF321 is ideal for data storage as well, eliminating the need for additional data storage devices.

The erase block sizes of the AT25SF321 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 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 three pages of Security Register that can be used for purposes such as unique device serialization, system-level Electronic Serial Number (ESN) storage, locked key storage, etc. These Security Register pages can be individually locked.

# 1. Pin Descriptions and Pinouts

**Table 1-1. 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 and Quad-Output Read commands, the SI Pin becomes an output pin ($I/O_0$) in conjunction with other pins to allow two or four bits of data on ($I/O_{3-0}$) to be clocked in 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 and Quad-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 1-1. 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 ($I/O_0$) in conjunction with other pins to allow two bits of data on ($I/O_{1-0}$) to be clocked in 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 will be 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 |
| $\overline{WP}$ ($I/O_2$) | **WRITE PROTECT:** The $\overline{WP}$ pin controls the hardware locking feature of the device.<br>With the Quad-Input Byte/Page Program command, the $\overline{WP}$ pin becomes an input pin ($I/O_2$) and, along with other pins, allows four bits (on $I/O_{3-0}$) of data to be clocked in on every rising edge of SCK. With the Quad-Output Read commands, the $\overline{WP}$ Pin becomes an output pin ($I/O_2$) in conjunction with other pins to allow four bits of data on ($I/O3_{3-0}$) to be clocked in on every falling edge of SCK.<br>To maintain consistency with the SPI nomenclature, the $\overline{WP}$ ($I/O_2$) pin will be referenced as the $\overline{WP}$ pin unless specifically addressing the Quad-I/O modes in which case it will be referenced as $I/O_2$<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. | - | Input/Output |
| $\overline{HOLD}$ ($I/O_3$) | **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 Function" on page 34 for additional details on the Hold operation.<br>With the Quad-Input Byte/Page Program command, the $\overline{HOLD}$ pin becomes an input pin ($I/O_3$) and, along with other pins, allows four bits (on $I/O_{3-0}$) of data to be clocked in on every rising edge of SCK. With the Quad-Output Read commands, the $\overline{HOLD}$ Pin becomes an output pin ($I/O_3$) in conjunction with other pins to allow four bits of data on ($I/O3_{3-0}$) to be clocked in on every falling edge of SCK.<br>To maintain consistency with the SPI nomenclature, the $\overline{HOLD}$ ($I/O_3$) pin will be referenced as the $\overline{HOLD}$ pin unless specifically addressing the Quad-I/O modes in which case it will be referenced as $I/O_3$<br>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. | - | Input/Output |
| $V_{CC}$ | **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 |

**Figure 1-1. 8-SOIC (Top View)**

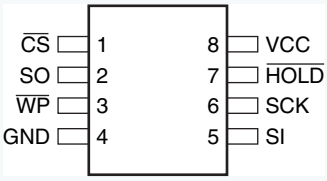
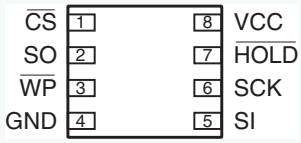
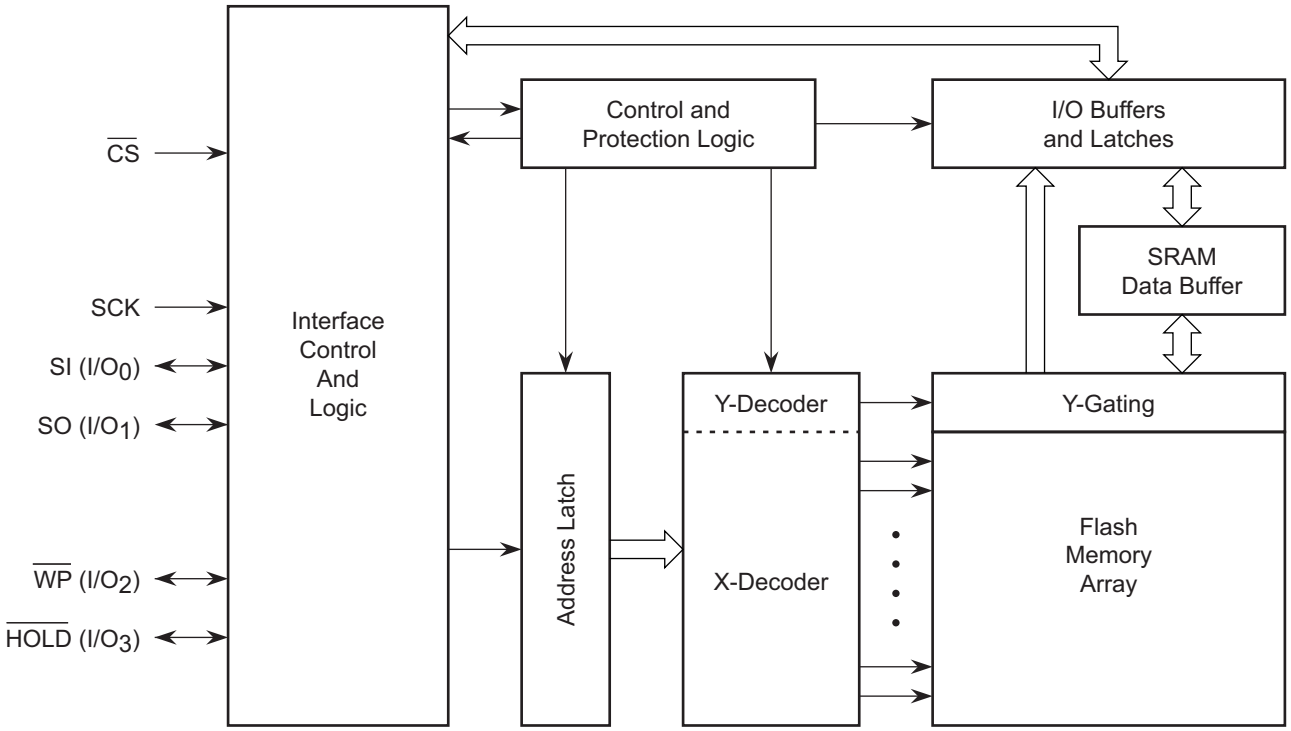



**Figure 1-2. 8-UDFN (Top View)**




## 2. Block Diagram

**Figure 2-1. Block Diagram**

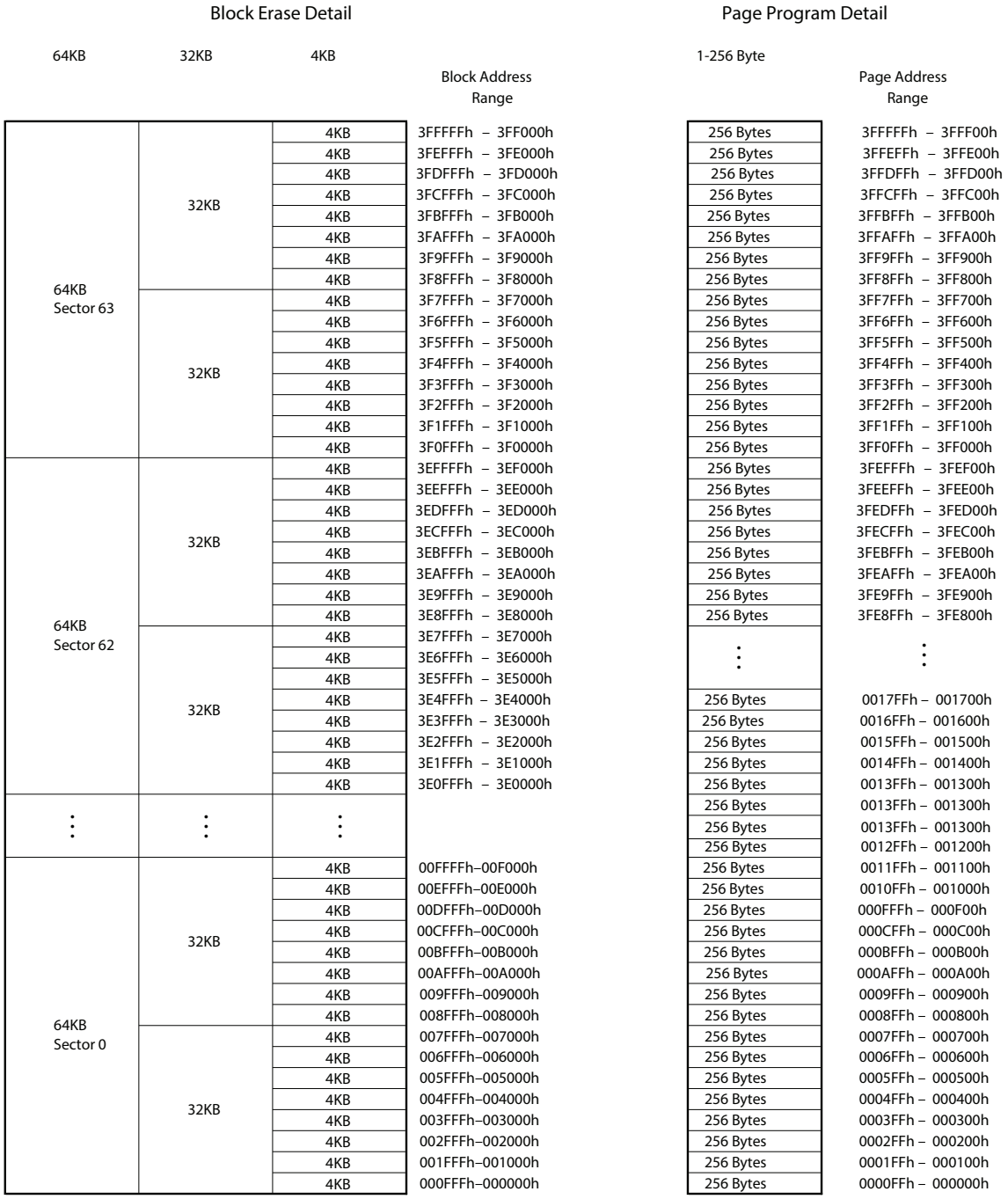



Note: $I/O_{3-0}$ pin naming convention is used for Dual-I/O and Quad-I/O commands.

## 3. Memory Array

To provide the greatest flexibility, the memory array of the AT25SF321 can be erased in four 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 3-1. Memory Architecture Diagram

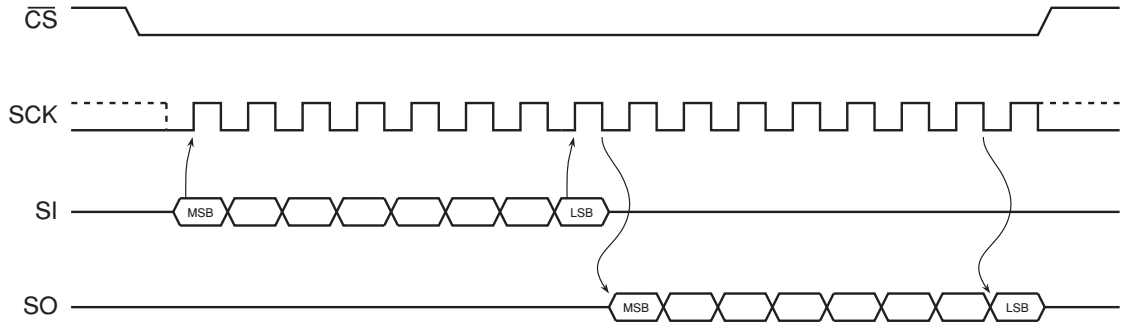


# 4. Device Operation

The AT25SF321 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 AT25SF321 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 AT25SF321 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 4-1. SPI Mode 0 and 3**




## 4.1 Dual Output Read

The AT25SF321 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.

## 4.2 Quad Output Read

The AT25SF321 features a Quad-Output Read mode that allow four bits of data to be clocked out of the device every clock cycle to improve throughput. To accomplish this, the SI, SO, $\overline{WP}$, $\overline{HOLD}$ pins are utilized as outputs for the transfer of data bytes. With the Quad-Output Read Array command, the SI, $\overline{WP}$, $\overline{HOLD}$ pins become outputs along with the SO pin.

# 5. 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 AT25SF321 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 AT25SF321 memory array is 3FFFFFh, address bits A23-A22 are always ignored by the device.

**Table 5-1. Command Listing**

| Command | Opcode | | Clock Frequency | Address Bytes | Dummy Bytes | Data Bytes | Section Link |
|---|---|---|---|---|---|---|---|
| **Read Commands** | | | | | | | |
| Read Array | 0Bh | 0000 1011 | Up to 85 MHz | 3 | 1 | 1+ | 6.1 |
| | 03h | 0000 0011 | Up to 50 MHz | 3 | 0 | 1+ | |
| Dual Output Read | 3Bh | 0011 1011 | Up to 85 MHz | 3 | 1 | 1+ | 6.2 |
| Dual I/O Read | BBh | 1011 1011 | Up to 85 MHz | 3 | 0 | 1+ | 6.3 |
| Quad Output Read | 6Bh | 0110 1011 | Up to 85 MHz | 3 | 1 | 1+ | 6.4 |
| Quad I/O Read | EBh | 1110 1011 | Up to 85 MHz | 3 | 1 | 1+ | 6.5 |
| Continuous Read Mode Reset - Dual | FFFFh | 1111 1111<br>1111 1111 | Up to 104 MHz | 0 | 0 | 0 | 6.6 |
| Continuous Read Mode Reset - Quad | FFh | 1111 1111 | Up to 104 MHz | 0 | 0 | 0 | 6.6 |
| **Program and Erase Commands** | | | | | | | |
| Block Erase (4 Kbytes) | 20h | 0010 0000 | Up to 104 MHz | 3 | 0 | 0 | 7.2 |
| Block Erase (32 Kbytes) | 52h | 0101 0010 | Up to 104 MHz | 3 | 0 | 0 | |
| Block Erase (64 Kbytes) | D8h | 1101 1000 | Up to 104MHz | 3 | 0 | 0 | |
| Chip Erase | 60h | 0110 0000 | Up to 104 MHz | 0 | 0 | 0 | 7.3 |
| | C7h | 1100 0111 | Up to 104 MHz | 0 | 0 | 0 | |
| Byte/Page Program (1 to 256 Bytes) | 02h | 0000 0010 | Up to 104 MHz | 3 | 0 | 1+ | 7.1 |
| Program/Erase Suspend | 75h | 0111 0101 | Up to 104 MHz | 0 | 0 | 0 | 7.4 |
| Program/Erase Resume | 7Ah | 0111 1010 | Up to 104 MHz | 0 | 0 | 0 | 7.5 |
| **Protection Commands** | | | | | | | |
| Write Enable | 06h | 0000 0110 | Up to 104 MHz | 0 | 0 | 0 | 8.1 |
| Write Disable | 04h | 0000 0100 | Up to 104 MHz | 0 | 0 | 0 | 8.2 |
| **Security Commands** | | | | | | | |
| Erase Security Register Page | 44h | 0100 0100 | Up to 104 MHz | 3 | 0 | 0 | 9.1 |
| Program Security Register Page | 42h | 0100 0010 | Up to 104 MHz | 3 | 0 | 1+ | 9.2 |
| Read Security Register Page | 48h | 0100 1000 | Up to 85MHz | 3 | 1 | 1+ | 9.3 |
| **Status Register Commands** | | | | | | | |
| Read Status Register Byte 1 | 05h | 0000 0101 | Up to 104 MHz | 0 | 0 | 1 | 10.1 |
| Read Status Register Byte 2 | 35h | 0011 0101 | Up to 104 MHz | 0 | 0 | 1 | |
| Write Status Register | 01h | 0000 0001 | Up to 104 MHz | 0 | 0 | 1 or 2 | 10.2 |
| Write Enable for Volatile Status Register | 50h | 0101 0000 | Up to 104MHz | 0 | 0 | 0 | 10.3 |

**Table 5-1. Command Listing**

| Command | Opcode | | Clock Frequency | Address Bytes | Dummy Bytes | Data Bytes | Section Link |
|---|---|---|---|---|---|---|---|
| **Miscellaneous Commands** | | | | | | | |
| Read Manufacturer and Device ID | 9Fh | 1001 1111 | Up to 104MHz | 0 | 0 | 3 | 11.1 |
| Read ID | 90h | 1001 0000 | Up to 104 MHz | 0 | 3 | 2 | 11.2 |
| Deep Power-Down | B9h | 1011 1001 | Up to 104 MHz | 0 | 0 | 0 | 11.3 |
| Resume from Deep Power-Down | ABh | 1010 1011 | Up to 104 MHz | 0 | 0 | 0 | 11.4 |
| Resume from Deep Power-Down and Read ID | ABh | 1010 1011 | Up to 104 MHz | 0 | 3 | 1 | 11.4 |

# 6. Read Commands

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

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 (3FFFFFh) 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 6-1. Read Array - 03h Opcode**

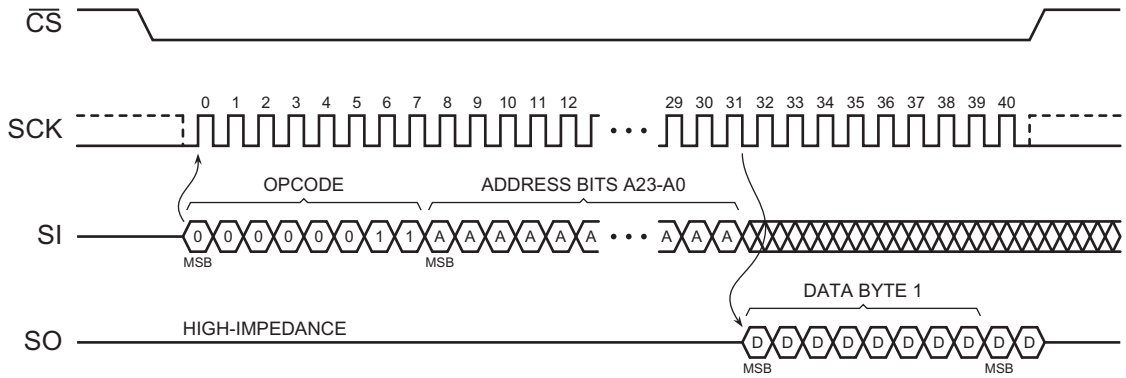
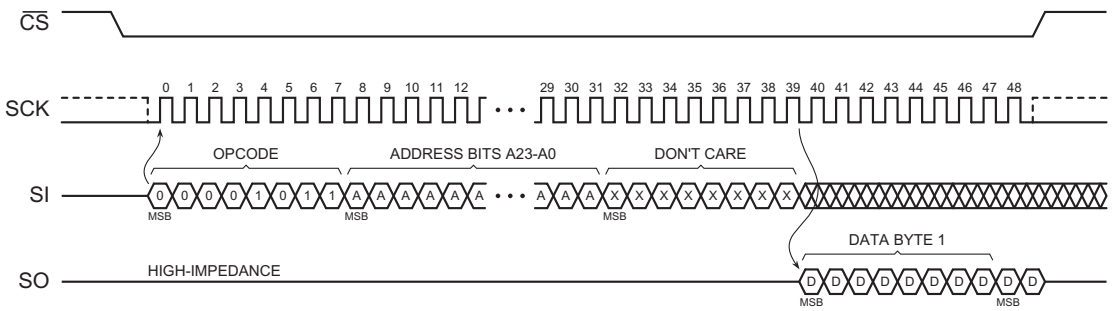


**Figure 6-2. Read Array - 0Bh Opcode**



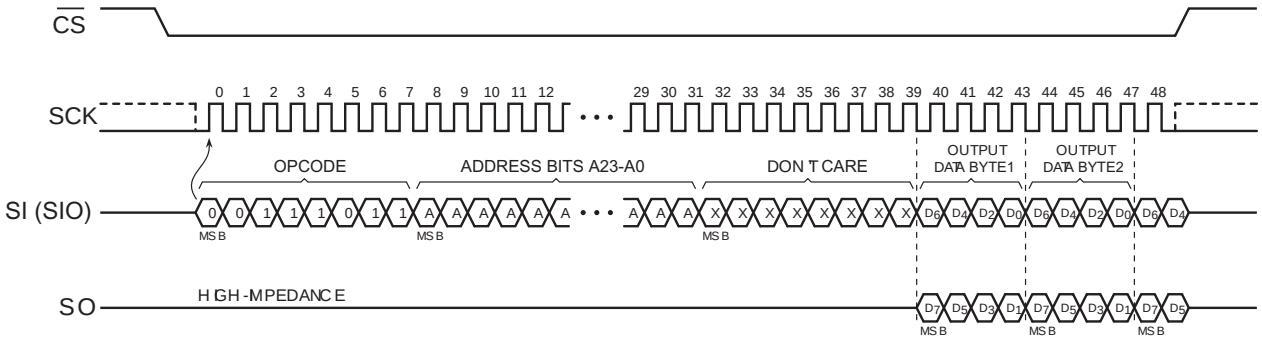

## 6.2 Dual-Output Read Array (3Bh)

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 SI pin. During the next clock cycle, bits five and four of the first data byte are output on the SO and SI pins, respectively. The sequence continues with each byte of data being output after every four clock cycles. When the last byte (3FFFFFh) 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 6-3. Dual-Output Read Array**



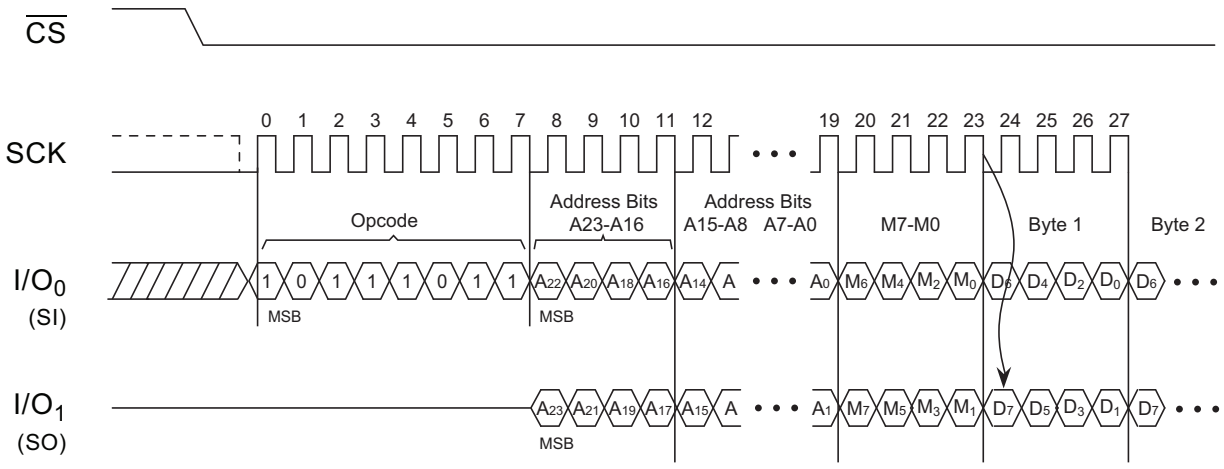
## 6.3 Dual-I/O Read Array (BBh)

The Dual-I/O Read Array command is similar to the Dual-Output 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 with two bits of address on each clock and two bits of data on every clock cycle.

The Dual-I/O Read Array command can be used at any clock frequency, up to the maximum specified by $f_{RDDO}$. To perform the Dual-I/O Read Array operation, the $\overline{CS}$ pin must first be asserted and then the opcode BBh 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 mode byte must also be clocked into the device.

After the three address bytes and the mode 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 SI pin. During the next clock cycle, bits five and four of the first data byte are output on the SO and SI pins, respectively. The sequence continues with each byte of data being output after every four clock cycles. When the last byte (3FFFFFh) 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.

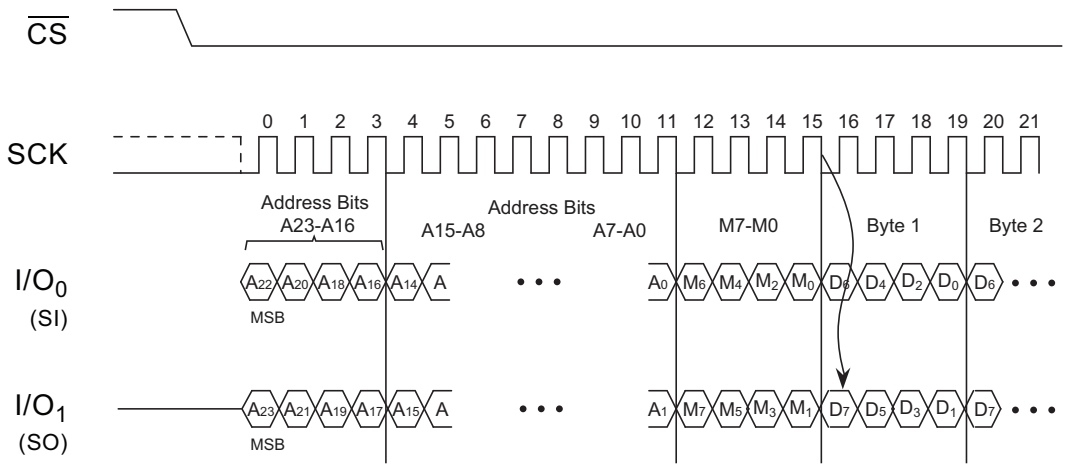
**Figure 6-4. Dual I/O Read Array (Initial command or previous M5, M4≠1,0)**




### 6.3.1 Dual-I/O Read Array (BBh) with Continuous Read Mode

The Fast Read Dual I/O command can further reduce instruction overhead through setting the Continuous Read Mode bits (M7-0) after the input Address bits (A23-0), as shown in Figure 6-5. The upper nibble of the (M7-4) controls the length of the next Fast Read Dual I/O command through the inclusion or exclusion of the first byte instruction code. The lower nibble bits of the (M3-0) are don't care ("x"). However, the IO pins should be high-impedance prior to the falling edge of the first data out clock. If the "Continuous Read Mode" bits M5-4 = (1,0), then the next Fast Read Dual I/O command (after $\overline{CS}$ is raised and then lowered) does not require the BBH instruction code, as shown in Figure 15. This reduces the command sequence by eight clocks and allows the Read address to be immediately entered after $\overline{CS}$ is asserted low. If the "Continuous Read Mode" bits M5-4 do not equal to (1,0), the next command (after $\overline{CS}$ is raised and then lowered) requires the first byte instruction code, thus returning to normal operation. A Continuous Read Mode Reset command can also be used to reset (M7-0) before issuing normal commands.

**Figure 6-5. Dual-I/O Read Array (Previous command set M5, M4 = 1,0)**




## 6.4 Quad-Output Read Array (6Bh)

The Quad-Output Read Array command is similar to the Dual-Output Read Array command. The Quad-Output Read Array command allows four bits of data to be clocked out of the device on every clock cycle, rather than just one or two.

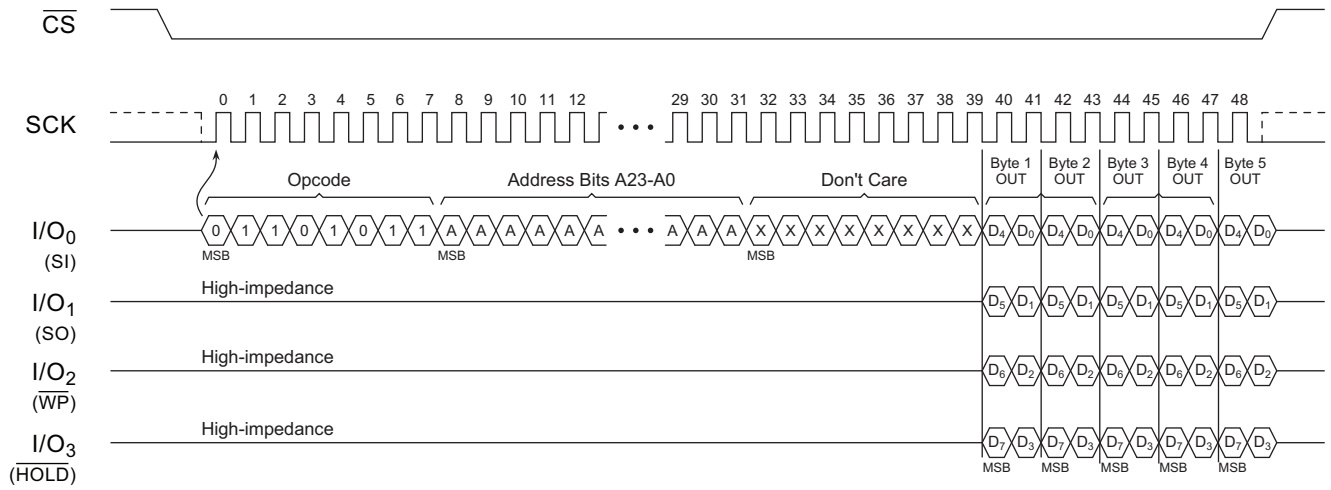
The Quad Enable bit (QE) of the Status Register must be set to enable for the Quad-Output Read Array instruction.

The Quad-Output Read Array command can be used at any clock frequency, up to the maximum specified by $f_{RDQO}$. To perform the Quad-Output Read Array operation, the $\overline{CS}$ pin must first be asserted and then the opcode 6Bh 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 the $I/O_{3-0}$ pins. The data is always output with the MSB of a byte first and the MSB is always output on the $I/O_3$ pin. During the first clock cycle, bit 7 of the first data byte will be output on the $I/O_3$ pin while bits 6, 5, and 4 of the same data byte will be output on the $I/O_2$, $I/O_1$, and $I/O_0$ pins, respectively. During the next clock cycle, bits 3, 2, 1, and 0 of the first data byte will be output on the $I/O_3$, $I/O_2$, $I/O_1$ and $I/O_0$ pins, respectively.

The sequence continues with each byte of data being output after every two clock cycles. When the last byte (3FFFFFh) 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 $\overline{WP}$, $\overline{HOLD}$, SO, 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 6-6. Quad-Output Read Array**




## 6.5 Quad-I/O Read Array(EBh)

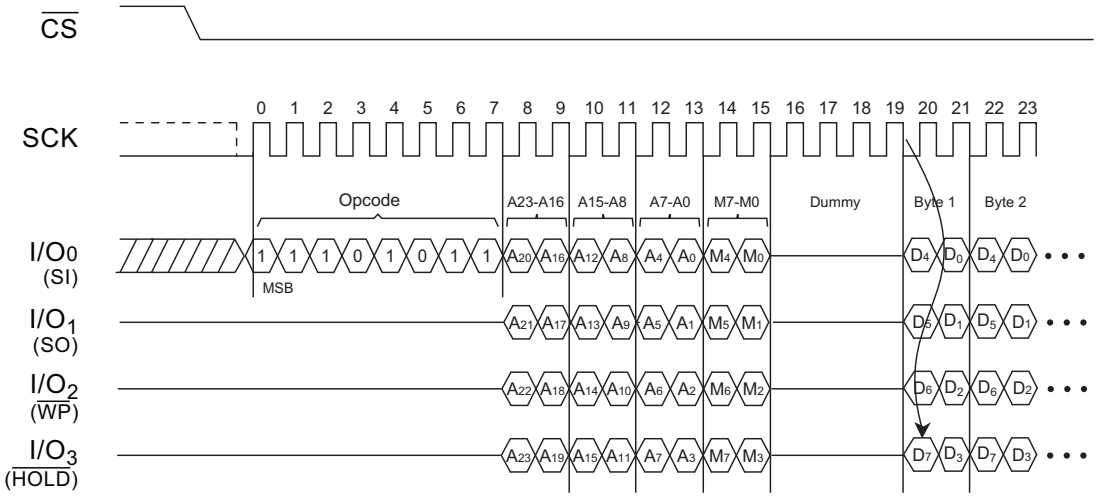
The Quad-I/O Read Array command is similar to the Quad-Output Read Array command. The Quad-I/O Read Array command allows four bits of address to be clocked into the device on every clock cycle, rather than just one.

The Quad-I/O Read Array command can be used at any clock frequency, up to the maximum specified by $f_{RDQO}$. To perform the Quad-I/O Read Array operation, the $\overline{CS}$ pin must first be asserted and then the opcode EBh 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 mode byte must also be clocked into the device.

After the three address bytes, the mode byte and two dummy bytes have been clocked in, additional clock cycles will result in data being output on the $I/O_{3-0}$ pins. The data is always output with the MSB of a byte first and the MSB is always output on the $I/O_3$ pin. During the first clock cycle, bit 7 of the first data byte will be output on the $I/O_3$ pin while bits 6, 5, and 4 of the same data byte will be output on the $I/O_2$, $I/O_1$ and $I/O_0$ pins, respectively. During the next clock cycle, bits 3, 2, 1, and 0 of the first data byte will be output on the $I/O_3$, $I/O_2$, $I/O_1$ and $I/O_0$ pins, respectively. The sequence continues with each byte of data being output after every two clock cycles.

When the last byte (3FFFFFh) 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 $I/O_3$, $I/O_2$, $I/O_1$ and $I/O_0$ 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.The Quad Enable bit (QE) of the Status Register must be set to enable for the Quad-I/O Read Array instruction.

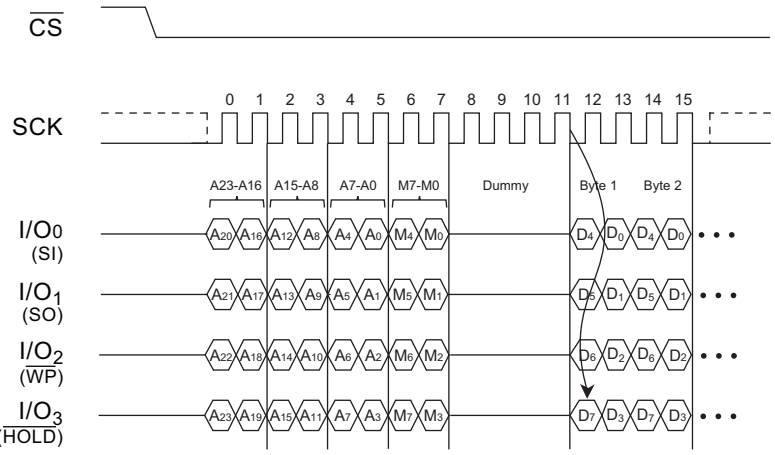
**Figure 6-7. Quad-I/O Read Array (Initial command or previous M5, M4 ≠ 1,0)**




### 6.5.1 Quad-I/O Read Array (EBh) with Continuous Read Mode

The Fast Read Quad I/O command can further reduce instruction overhead through setting the Continuous Read Mode bits (M7-0) after the input Address bits (A23-0), as shown in Figure 6-6. The upper nibble (M7-4) of the Continuous Read Mode bits controls the length of the next Fast Read Quad I/O command through the inclusion or exclusion of the first byte instruction code. The lower nibble bits (M3-0) of the Continuous Read Mode bits are don't care. However, the IO pins should be high-impedance prior to the falling edge of the first data out clock. If the Continuous Read Mode bits M5-4 = (1,0), then the next Quad-I/O Read Array command (after $\overline{CS}$ is raised and then lowered) does not require the EBh instruction code, as shown in Fig 6-8. This reduces the command sequence by eight clocks and allows the Read address to be immediately entered after $\overline{CS}$ is asserted low. If the "Continuous Read Mode" bits M5-4 do not equal to (1,0), the next command (after $\overline{CS}$ is raised and then lowered) requires the first byte instruction code, thus returning to normal operation. A Continuous Read Mode Reset command can also be used to reset (M7-0) before issuing normal commands.

**Figure 6-8. Quad I/O Read Array with Continuous Read Mode (Previous Command Set M5-4 =1,0)**




## 6.6 Continuous Read Mode Reset (FFh or FFFFh)

The Continuous Read Mode bits are used in conjunction with the Dual I/O Read Array and the Quad I/O Read Array commands to provide the highest random Flash memory access rate with minimum SPI instruction overhead, thus allowing more efficient XIP (execute in place) with this device family.

The "Continuous Read Mode" bits M7-0 are set by the Dual/Quad I/O Read Array commands. M5-4 are used to control whether the 8-bit SPI instruction code (BBh or EBh) is needed or not for the next instruction. When M5-4 = (1,0), the next instruction will be treated the same as the current Dual/Quad I/O Read Array command without needing the 8-bit instruction code. When M5-4 do not equal (1,0), the device returns to normal SPI instruction mode, in which all instructions can be accepted. M7-6 and M3-0 are reserved bits for future use; either 0 or 1 values can be used.

See Figure 6-9, the Continuous Read Mode Reset instruction (FFh or FFFFh) can be used to set M4 = 1, thus the device will release the Continuous Read Mode and return to normal SPI operation.

To reset Continuous Read Mode during Quad I/O operation, only eight clocks are needed to shift in instruction FFh. To reset Continuous Read Mode during Dual I/O operation, sixteen clocks are needed to shift in instruction FFFFh.

**Figure 6-9. Continuous Read Mode Reset (Quad)**

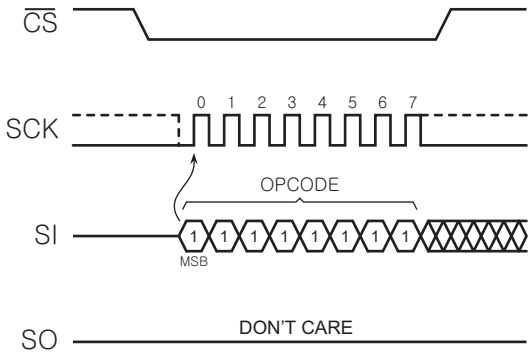
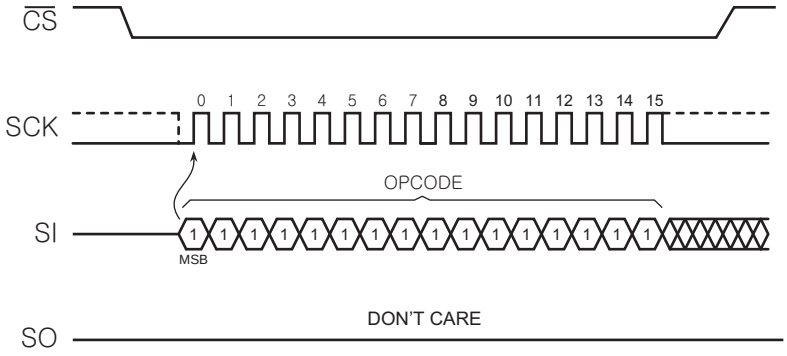



**Figure 6-10. Continuous Read Mode Reset (Dual)**




# 7. Program and Erase Commands

## 7.1 Byte/Page Program (02h)

The Byte/Page Program command allows anywhere from a single byte of data to 256 bytes of data to be programmed into previously erased memory locations. An erased memory location is one that has all eight bits set to the logical “1” state (a byte value of FFh). Before a Byte/Page Program command can be started, the Write Enable command must have been previously issued to the device (see “Write Enable (06h)” on page 20) 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 “Non-Volatile Protection” on page 21), 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. At some point before the program cycle completes, the WEL bit in the Status Register will be reset back to the logical “0” state.

**Figure 7-1. Byte Program**

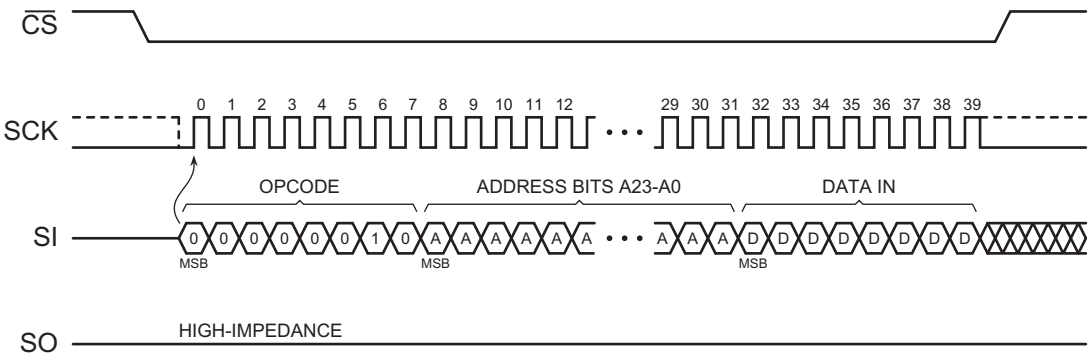
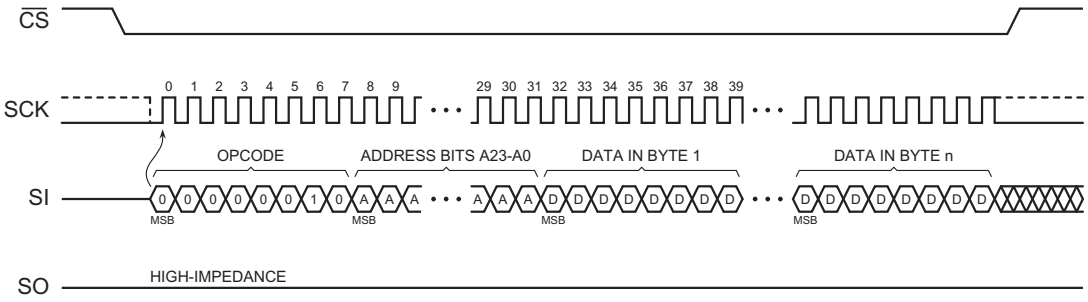



**Figure 7-2. Page Program**



## 7.2 Block Erase (20h, 52h, or D8h)

A block of 4, 32, or 64 Kbytes 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 is used for a 32-Kbyte erase, or 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- or 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 $t_{BLKE}$.

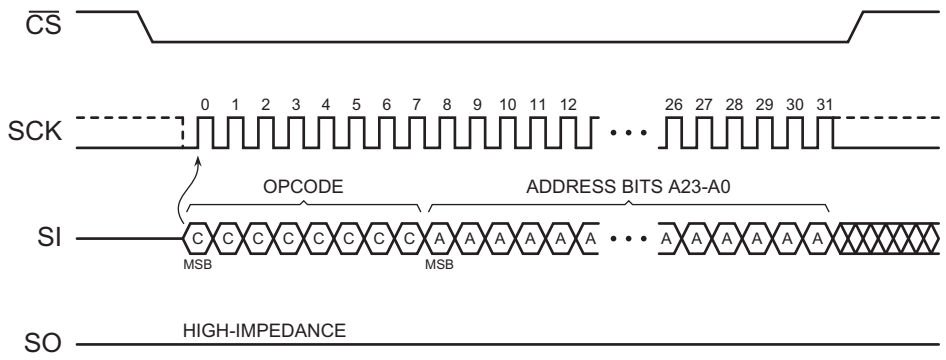
Since the Block Erase command erases a region of bytes, the lower order address bits do not need to be decoded by the device. Therefore, for a 4-Kbyte erase, address bits A11-A0 will be ignored by the device and their values can be either a logical “1” or “0”. For a 32-Kbyte erase, address bits A14-A0 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 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.

**Figure 7-3. Block Erase**




## 7.3 Chip Erase (60h or C7h)

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.

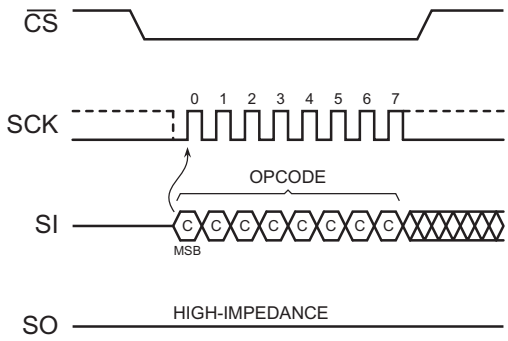
Two opcodes (60h and C7h) can be used for the Chip Erase command. There is no difference in device functionality when utilizing the two opcodes, so they can be used interchangeably. To perform a Chip Erase, one of the two opcodes must be clocked into the device. Since the entire memory array is to be 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 byte boundary (multiples of eight bits); otherwise, no erase will be performed. In addition, if 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.

**Figure 7-4. Chip Erase**




## 7.4 Program/Erase Suspend (75h)

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

Chip Erase cannot be suspended. The Program/Erase Suspend command will be ignored if it is issued during a Chip Erase.A program operation can be performed while an erase operation is suspended, but the program operation cannot be suspended while an erase operation is currently suspended.

Other device operations, such as a Read Status Register, can also be performed while a program or erase operation is suspended. Table 7-4 outlines the operations that are allowed and not allowed during a program or erase suspend.

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

To perform a Program/Erase Suspend, the $\overline{CS}$ pin must first be asserted and the opcode of 75h 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 program or erase operation currently in progress will be suspended within a time of $t_{SUSE}$. The Suspend (SUS) bit in the Write Status Register will be set to the logical "1" state to indicate that the program or erase operation has been suspended. In addition, the $\overline{RDY}$/BSY bit in the Status Register will indicate that the device is ready for another operation. 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 a byte boundary (multiples of eight bits). Otherwise, no suspend operation will be performed.

If a read operation is attempted to a suspended area (page for programming or block for erasing), then the device will output undefined data. Therefore, when performing a Read Array operation to an unsuspended area and the device's

internal address counter increments and crosses into the suspended area, the device will then start outputting undefined data until the internal address counter crosses to an unsuspended area.

A program operation is not allowed to a block that has been erase suspended. If a program operation is attempted to an erase suspended block, then the program operation will abort and the WEL bit in the Status Register will be reset back to a logical "0" state. Likewise, an erase operation is not allowed to a block that included the page that has been program suspended. If attempted, the erase operation will abort and the WEL bit in the Status Register will be reset to a logical "0" state.

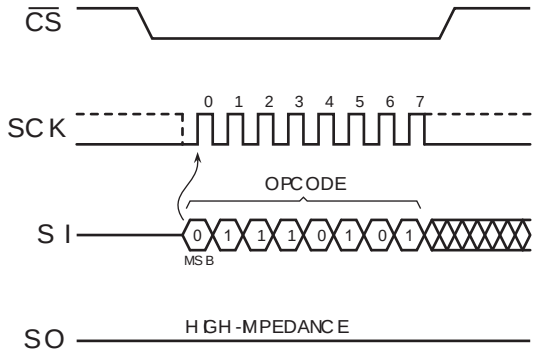
If an attempt is made to perform an operation that is not allowed during a program or erase suspend, such as a Write Status Register operation, then the device will simply ignore the opcode and no operation will be performed. The state of the WEL bit in the Status Register will not be affected.

**Table 7-1. Operations Allowed and Not Allowed During a Program/Erase Suspend Command**

| Command | During Program Suspend | During Erase Suspend |
|---|---|---|
| **Read Commands** | | |
| Read Array (03h, 0Bh, 3Bh, BBh, 6Bh, EBh) | Allowed (1) | Allowed(1) |
| Continuous Read Reset (FFh) | Allowed(1) | Allowed(1) |
| **Program and Erase Commands** | | |
| Block Erase (20h, 52h, D8h) | Not Allowed | Not Allowed |
| Chip Erase (C7h, 60h) | Not Allowed | Not Allowed |
| Byte/Page Program (02h) | Not Allowed | Allowed(1) |
| Program/Erase Suspend (75h) | Not Allowed | Not Allowed |
| Program/Erase Resume (7Ah) | Allowed | Allowed |
| **Protection Commands** | | |
| Write Enable (06h) | Allowed | Allowed |
| Write Disable (04h) | Allowed | Allowed |
| **Security Commands** | | |
| Erase Security Register Page (44h) | Not Allowed | Not Allowed |
| Program Security Register Page (42h) | Not Allowed | Not Allowed |
| Read Security Register Page (48h) | Allowed | Allowed |
| **Status Register Commands** | | |
| Read Status Register (05h, 35h) | Allowed | Allowed |
| Write Status Register (01h) | Not Allowed | Not Allowed |
| Volatile Write Enable Status Register (50h) | Not Allowed | Not Allowed |
| **Miscellaneous Commands** | | |
| Read Manufacturer and Device ID (9Fh) | Allowed | Allowed |
| Read ID (90h) | Allowed | Allowed |
| Deep Power-Down (B9h) | Not Allowed | Not Allowed |
| Resume from Deep Power-Down (ABh) | Allowed (2) | Allowed(2) |

1. Allowed for all 64K-byte blocks other than the one currently suspended.
2. Allowed for reading Device ID.

**Figure 7-5. Program/Erase Suspend**




## 7.5 Program/Erase Resume (7Ah)

The Program/Erase Resume command allows a suspended program or erase operation to be resumed and continue programming a Flash page or erasing a Flash memory block where it left off. The Program/Erase Resume instruction will be accepted by the device only if the SUS bit in the Write Status Register equals 1 and the RDY/BSY bit equals 0. If the SUS bit equals 0 or the RDY/BSY bit equals to 1, the Program/Erase Resume command will be ignored by the device. As with the Program/Erase Suspend command, the Write Enable command does not need to be issued prior to the Program/Erase Resume command being issued. Therefore, the Program/Erase Resume command operates independently of the state of the WEL bit in the Status Register.

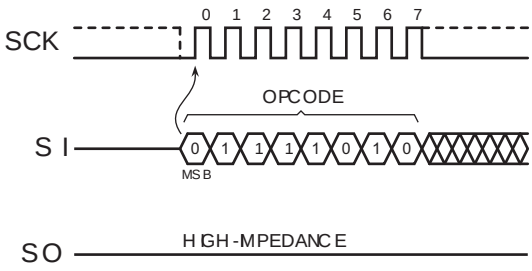
To perform Program/Erase Resume, the CS pin must first be asserted and opcode 7Ah must be clocked into the device.

No address bytes need to be clocked into the device, and any data clocked in after the opcode will be ignored. When the CS pin is deasserted, the program or erase operation currently suspended will resume within a time of $t_{RES}$. The SUS bit in the Status Register will then be reset back to the logical "0" state to indicate the program or erase operation is no longer suspended. In addition, the RDY/BSY bit in the Status Register will indicate that the device is busy performing a program or erase operation. The complete opcode must be clocked into the device before the CS pin is deasserted, and the CS pin must be deasserted on a byte boundary (multiples of eight bits). Otherwise, no resume operation will perform.

During a simultaneous Erase Suspend/Program Suspend condition, issuing the Program/Erase Resume command will result in the program operation resuming first. After the program operation has been completed, the Program/Erase Resume command must be issued again in order for the erase operation to be resumed.

While the device is busy resuming a program or erase operation, any attempts at issuing the Program/Erase Suspend command will be ignored. Therefore, if a resumed program or erase operation needs to be subsequently suspended again, the system must either wait the entire $t_{RES}$ time before issuing the Program/Erase Suspend command, or it must check the status of the RDY/BSY bit or the SUS bit in the Status Register to determine if the previously suspended program or erase operation has resumed.

**Figure 7-6. Program/Erase Resume.**



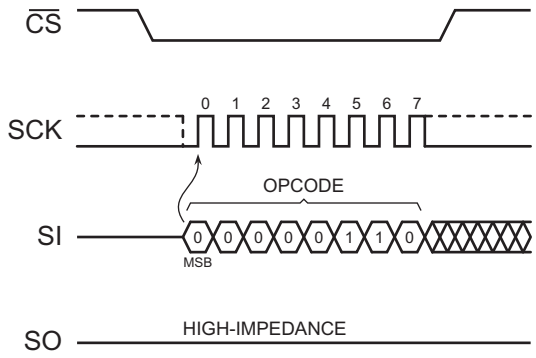
# 8. Protection Commands and Features

## 8.1 Write Enable (06h)

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 Security Register Pages, Erase Security Register Pages 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 byte boundary (multiples of eight bits); otherwise, the device will abort the operation and the WEL bit state will not change.

**Figure 8-1. Write Enable**

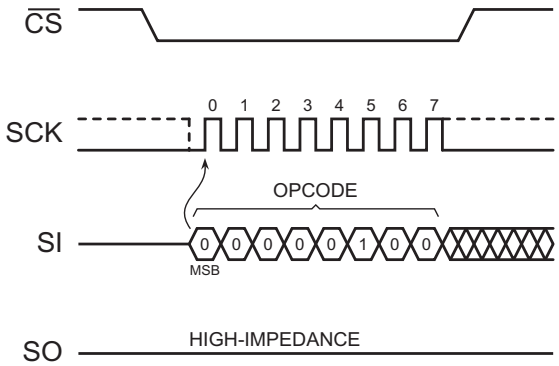



## 8.2 Write Disable (04h)

The Write Disable command is us0d 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 Security Register Page, 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 byte boundary (multiples of eight bits); otherwise, the device will abort the operation and the WEL bit state will not change.

**Figure 8-2. Write Disable**



## 8.3 Non-Volatile Protection

The device can be software protected against erroneous or malicious program or erase operations by utilizing the Non-Volatile Protection feature of the device. Non-Volatile Protection can be enabled or disabled by using the Write Status Register command to change the value of the Protection (CMP, SEC, TB, BP2, BP1, BP0) bits in the Status Register. The following table outlines the states of the Protection bits and the associated protection area

.

**Table 8-1. Memory Array with CMP=0**

| Protection Bits | | | | | Memory Content | |
|---|---|---|---|---|---|---|
| SEC | TB | BP2 | BP1 | BP0 | Address Range | Portion |
| X | X | 0 | 0 | 0 | None | None |
| 0 | 0 | 0 | 0 | 1 | 3F0000h-3FFFFFh | Upper 1/64 |
| 0 | 0 | 0 | 1 | 0 | 3E0000h-3FFFFFh | Upper 1/32 |
| 0 | 0 | 0 | 1 | 1 | 3C0000h-3FFFFFh | Upper 1/16 |
| 0 | 0 | 1 | 0 | 0 | 380000h-3FFFFFh | Upper 1/8 |
| 0 | 0 | 1 | 0 | 1 | 300000h-3FFFFFh | Upper 1/4 |
| 0 | 0 | 1 | 1 | 0 | 200000h-3FFFFFh | Upper 1/2 |
| 0 | 1 | 0 | 0 | 1 | 000000h-00FFFFh | Lower 1/64 |
| 0 | 1 | 0 | 1 | 0 | 000000h-01FFFFh | Lower 1/32 |
| 0 | 1 | 0 | 1 | 1 | 000000h-03FFFFh | Lower 1/16 |
| 0 | 1 | 1 | 0 | 0 | 000000h-07FFFFh | Lower 1/8 |
| 0 | 1 | 1 | 0 | 1 | 000000h-0FFFFFh | Lower 1/4 |
| 0 | 1 | 1 | 1 | 0 | 000000h-1FFFFFh | Lower 1/2 |
| X | X | 1 | 1 | 1 | 000000h-3FFFFFh | ALL |
| 1 | 0 | 0 | 0 | 1 | 3FF000h-3FFFFFh | Upper 1/512 |
| 1 | 0 | 0 | 1 | 0 | 3FE000h-3FFFFFh | Upper 1/256 |
| 1 | 0 | 0 | 1 | 1 | 3FC000h-3FFFFFh | Upper 1/128 |
| 1 | 0 | 1 | 0 | X | 3F8000h-3FFFFFh | Upper 1/64 |
| 1 | 0 | 1 | 1 | 0 | 3F8000h-3FFFFFh | Upper 1/64 |
| 1 | 1 | 0 | 0 | 1 | 000000h-000FFFh | Lower 1/512 |
| 1 | 1 | 0 | 1 | 0 | 000000h-001FFFh | Lower 1/256 |
| 1 | 1 | 0 | 1 | 1 | 000000h-003FFFh | Lower 1/128 |
| 1 | 1 | 1 | 0 | X | 000000h-007FFFh | Lower 1/64 |
| 1 | 1 | 1 | 1 | 0 | 000000h-007FFFh | Lower 1/64 |

**Table 8-2. Memory Array Protection with CMP=1**

| Protection Bits | | | | | Memory Content | |
|---|---|---|---|---|---|---|
| SEC | TB | BP2 | BP1 | BP0 | Address Range | Portion |
| X | X | 0 | 0 | 0 | 000000h-3FFFFFh | All |
| 0 | 0 | 0 | 0 | 1 | 000000h-3EFFFFh | Lower 63/64 |
| 0 | 0 | 0 | 1 | 0 | 000000h-3DFFFFh | Lower 31/32 |
| 0 | 0 | 0 | 1 | 1 | 000000h-3BFFFFh | Lower 15/16 |
| 0 | 0 | 1 | 0 | 0 | 000000h-37FFFFh | Lower 7/8 |
| 0 | 0 | 1 | 0 | 1 | 000000h-2FFFFFh | Lower 3/4 |
| 0 | 0 | 1 | 1 | 0 | 000000h-1FFFFFh | Lower 1/2 |
| 0 | 1 | 0 | 0 | 1 | 010000h-3FFFFFh | Upper 63/64 |
| 0 | 1 | 0 | 1 | 0 | 020000h-3FFFFFh | Upper 31/32 |
| 0 | 1 | 0 | 1 | 1 | 040000h-3FFFFFh | Upper 15/16 |
| 0 | 1 | 1 | 0 | 0 | 080000h-3FFFFFh | Upper 7/8 |
| 0 | 1 | 1 | 0 | 1 | 100000h-3FFFFFh | Upper 3/4 |
| 0 | 1 | 1 | 1 | 0 | 200000h-3FFFFFh | Upper 1/2 |
| X | X | 1 | 1 | 1 | NONE | NONE |
| 1 | 0 | 0 | 0 | 1 | 000000h-3FEFFFh | Lower 1023/1024 |
| 1 | 0 | 0 | 1 | 0 | 000000h-3FDFFFh | Lower 511/512 |
| 1 | 0 | 0 | 1 | 1 | 000000h-3FBFFFh | Lower 255/256 |
| 1 | 0 | 1 | 0 | X | 000000h-3F7FFFh | Lower 127/128 |
| 1 | 0 | 1 | 1 | 0 | 000000h-3F7FFFh | Lower 127/128 |
| 1 | 1 | 0 | 0 | 1 | 001000h-3FFFFFh | Upper 1023/1024 |
| 1 | 1 | 0 | 1 | 0 | 002000h-3FFFFFh | Upper 511/512 |
| 1 | 1 | 0 | 1 | 1 | 004000h-3FFFFFh | Upper 255/256 |
| 1 | 1 | 1 | 0 | X | 008000h-3FFFFFh | Upper 127/128 |
| 1 | 1 | 1 | 1 | 0 | 008000h-3FFFFFh | Upper 127/128 |

As a safeguard against accidental or erroneous protecting or unprotecting of the memory array, the Protection can be locked from updates by using the $\overline{WP}$ pin (see “Protected States and the Write Protect Pin” on page 22 for more details).

## 8.4 Protected States and the Write Protect 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, is used to control the hardware locking mechanism of the device.

If the $\overline{WP}$ pin is permanently connected to GND, then the protection bits cannot be changed.

# 9. Security Register Commands

The device contains three extra pages called Security Registers that can be used for purposes such as unique device serialization, system-level Electronic Serial Number (ESN) storage, locked key storage, etc. The Security Registers are independent of the main Flash memory.

Each page of the Security Register can be erased and programmed independently. Each page can also be independently locked to prevent further changes.

## 9.1 Erase Security Registers (44h)

Before an erase Security Register Page 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 an Erase Security Register Page command, the $\overline{CS}$ pin must first be asserted and the opcode 44h must be clocked into the device. After the opcode has been clocked in, the three address bytes specifying the Security Register Page to be erased must be clocked in. 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 $t_{BE}$.

Since the Erase Security Register Page command erases a region of bytes, the lower order address bits do not need to be decoded by the device. Therefore address bits A7-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 right after the last address bit (A0); otherwise, the device will abort the operation and no erase operation will be performed.

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 $t_{BE}$ time to determine if the device has finished erasing. At some point before the erase cycle completes, the $\overline{RDY}$/BSY bit in the Status Register will be reset back to the logical "0" state.

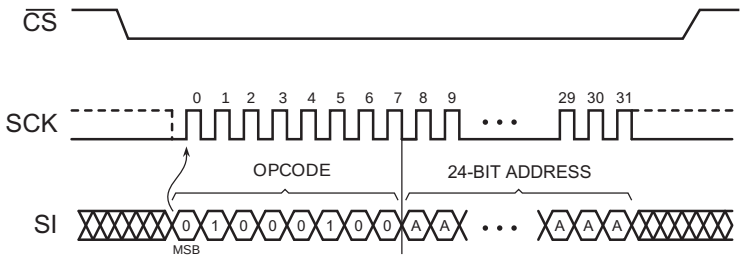
The WEL bit in the Status Register will be reset back to the logical "0" state if the erase cycle aborts due to an incomplete address being sent, the $\overline{CS}$ pin being deasserted on uneven byte boundaries, or because a memory location within the region to be erased is protected.

The Security Registers Lock Bits (LB3-LB1) in the Status Register can be used to OTP protect the security registers. Once a Lock Bit is set to 1, the corresponding Security Register will be permanently locked. The Erase Security Register Page instruction will be ignored for Security Registers which have their Lock Bit set.

**Table 9-1. Security Register Addresses for Erase Security Register Page Command**

| Address | A23-A16 | A15-A8 | A7-A0 |
|---|---|---|---|
| Security Register 1 | 00H | 01H | Don't Care |
| Security Register 2 | 00H | 02H | Don't Care |
| Security Register 3 | 00H | 03H | Don't Care |

**Figure 9-1. Erase Security Register Page**



## 9.2 Program Security Registers (42h)

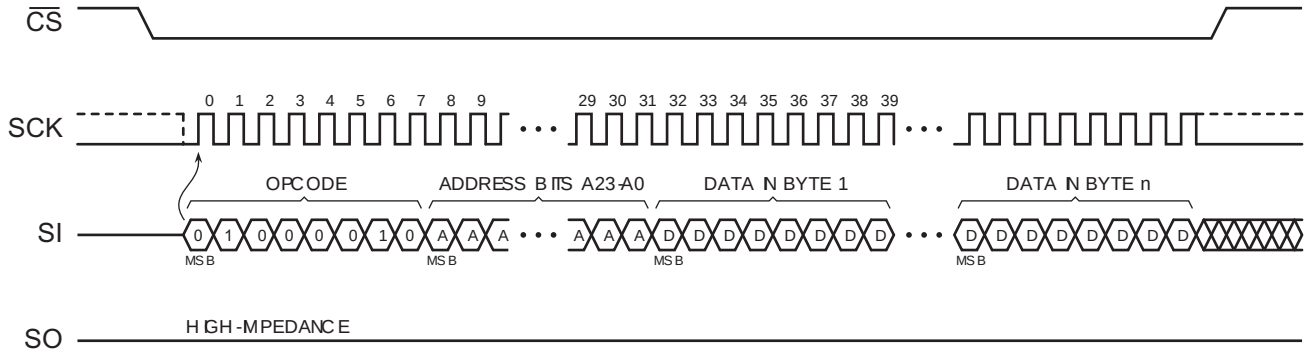
The Program Security Registers command utilizes the internal 256-byte buffer for processing. Therefore, the contents of the buffer will be altered from its previous state when this command is issued.

The Security Registers can be programmed in a similar fashion to the Program Array operation up to the maximum clock frequency specified by $f_{CLK}$. Before a Program Security Registers command can be started, the Write Enable command must have been previously issued to the device (see “Write Enable (06h)” on page 20) to set the Write Enable Latch (WEL) bit of the Status Register to a logical “1” state. To program the Security Registers, the $\overline{CS}$ pin must first be asserted and the opcode of 42h 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 program within the Security Register.

**Table 9-2. Security Register Addresses for Program Security Registers Command**

| Address | A23-A16 | A15-A8 | A7-A0 |
|---|---|---|---|
| Security Register 1 | 00H | 01H | Byte Address |
| Security Register 2 | 00H | 02H | Byte Address |
| Security Register 3 | 00H | 03H | Byte Address |

**Figure 9-2. Program Security Registers**




## 9.3 Read Security Registers (48h)

The 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 Security Register, the $\overline{CS}$ pin must first be asserted and the opcode of 48h 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 Security Register. Following the three address bytes, one dummy byte must be clocked into the device before data can be output.

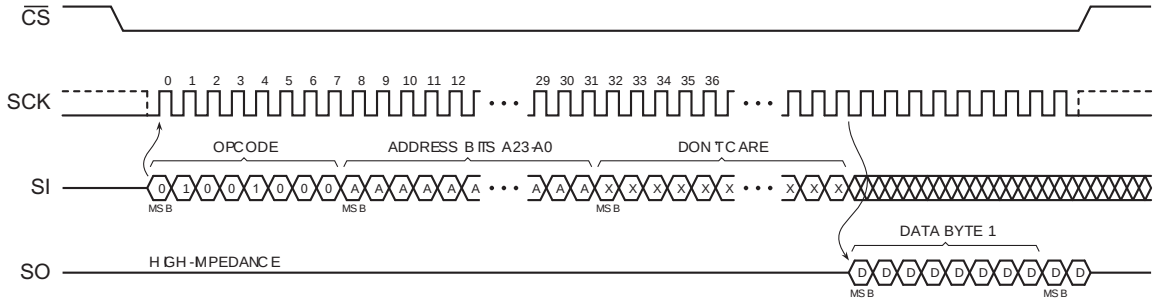
After the three address bytes and the dummy byte have been clocked in, additional clock cycles will result in Security Register data being output on the SO pin. When the last byte (0003FFh) of the Security Register has been read, the device will continue reading back at the beginning of the register (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.

**Table 9-3. Security Register Addresses for Read Security Registers Command**

| Address | A23-A16 | A15-A8 | A7-A0 |
|---|---|---|---|
| Security Register 1 | 00H | 01H | Byte Address |
| Security Register 2 | 00H | 02H | Byte Address |
| Security Register 3 | 00H | 03H | Byte Address |

**Figure 9-3. Read Security Registers**




# 10. Status Register Commands

## 10.1 Read Status Register (05h and 35h)

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 Block Protection. The Status Register can be read at any time, including during an internally self-timed program or erase operation.

To read Status Register Byte 1, 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 Byte 1 data on the SO pin during every subsequent clock cycle. After the last bit (bit 0) of Status Register Byte 1 has been clocked out, the sequence will repeat itself starting again with bit 7 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.

To read Status Register Byte 2, the $\overline{CS}$ pin must first be asserted and the opcode of 35h must be clocked into the device. After the opcode has been clocked in, the device will begin outputting Status Register Byte 2 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 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 10-1. Status Register Format - Byte 1**

| Bit[1] | Name | | Type[2] | Description | |
|---|---|---|---|---|---|
| 7 | SRP0 | Status Register Protection bit-0 | R/W | | See Table **10-3** on Status Register Protection |
| 6 | SEC | Block Protection | R/W | | See Table **8-1 and 8-2** on Non-Volatile Protection |
| 5 | TB | Top or Bottom Protection | R/W | | See Table **8-1 and 8-2** on Non-Volatile Protection |
| 4 | BP2 | Block Protection bit-2 | R/W | | See Table **8-1 and 8-2** on Non-Volatile Protection |
| 3 | BP1 | Block Protection bit-1 | R/W | | See Table **8-1 and 8-2** on Non-Volatile Protection |
| 2 | BP0 | Block Protection bit-0 | R/W | | See Table **8-1 and 8-2** on Non-Volatile Protection |
| 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 |

Notes: 1. Only bits 7 through 2 of the Status Register can be modified when using the Write Status Register command.

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

**Figure 10-1. Read Status Register Byte 1**

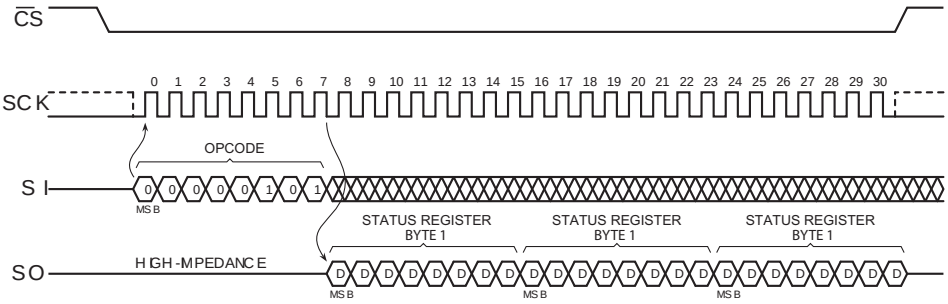



**Table 10-2. Status Register Format – Byte 2**

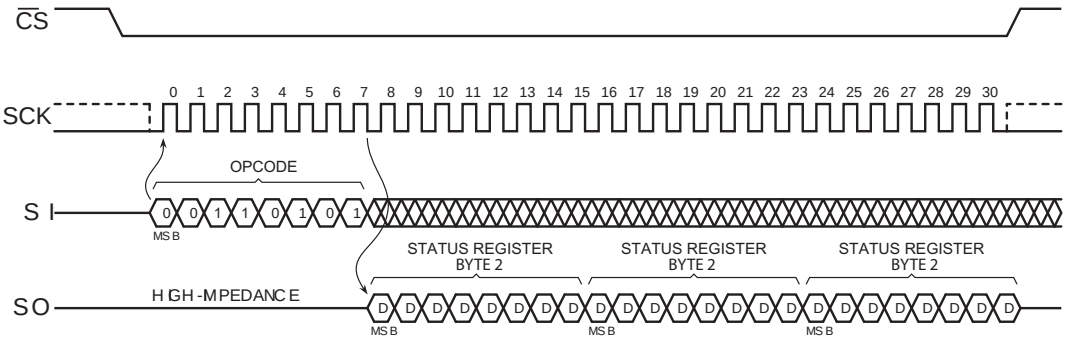
| Bit[1] | Name | | Type[2] | Description | |
|---|---|---|---|---|---|
| 7 | SUS | Suspend Status | R | 0 | Device is not suspended (default) |
| | | | | 1 | Device is suspended from erase |
| 6 | CMP | Complement Block Protection | R/W | 0 | See table on Block Protection |
| 5 | LB3 | Lock Security Register 3 | R/W | 0 | Security Register page-3 is not locked (default) |
| | | | | 1 | Security Register page-3 cannot be erased/programmed |
| 4 | LB2 | Lock Security Register 2 | R/W | 0 | Security Register page-2 is not locked (default) |
| | | | | 1 | Security Register page-2 cannot be erased/programmed |
| 3 | LB1 | Lock Security Register 1 | R/W | 0 | Security Register page-1 is not locked (default) |
| | | | | 1 | Security Register page-1 cannot be erased/programmed |
| 2 | RES | Reserved for future use | R | 0 | Reserved for future use |

**Table 10-2. Status Register Format – Byte 2**

| Bit[1] | Name | | Type[2] | Description | |
|---|---|---|---|---|---|
| 1 | QE | Quad Enable | R/W | 0 | $\overline{HOLD}$ and $\overline{WP}$ function normally (default) |
| | | | | 1 | $\overline{HOLD}$ and $\overline{WP}$ are I/O pins |
| 0 | SRP1 | Status Register Protect bit-1 | R/W | | See table on Status Register Protection |

Notes: 1. Only bits 6 through 3, 1, and 0 of the Status Register can be modified when using the Write Status Register command
2. R/W = Readable and writable
R = Readable only.

**Figure 10-2. Read Status Register Byte 2**




### 10.1.1 SRP1, SRP0 Bits

The SRP1 and SRP0 bits control whether the Status Register can be modified. The state of the $\overline{WP}$ pin along with the values of the SRP1 and SRP0 determine if the device is software protected, hardware protected, or permanently protected (see **Table 10-3**).

**Table 10-3. Status Register Protection Table**

| SRP1 | SRP0 | $\overline{WP}$ | Status Register | Description |
|---|---|---|---|---|
| 0 | 0 | X | Software Protected | The Status Register can be written to after a Write Enable instruction, WEL=1.(Factory Default) |
| 0 | 1 | 0 | Hardware Protected | $\overline{WP}$=0, the Status Register is locked and cannot be written. |
| 0 | 1 | 1 | Hardware Unprotected | $\overline{WP}$ =1, the Status Register is unlocked and can be written to after a Write Enable instruction, WEL=1. |
| 1 | 0 | X | Power Supply Lock-Down (1) | Status Register is protected and cannot be written to again until the next Power-Down, Power-Up cycle. |
| 1 | 1 | X | One Time Program | Status Register is permanently protected and cannot be written to. |

1. When SRP1, SRP0 = (1, 0), a Power-Down, Power-Up cycle will change SRP1, SRP0 to the (0, 0) state.

### 10.1.2 CMP, SEC, TB, BP2, BP1, BP0 Bits

The CMP, SEC, TB, BP2, BP1, and BP0 bits control which portions of the array are protected from erase and program operations (see Tables 8-1 and 8-2).

The CMP bit complements the effect of the other bits.

The SEC bit selects between large and small block size protection.

The TB bit selects between top of the array or bottom of the array protection.

The BP2, BP1, and BP0 bits determine how much of the array is protected.

### 10.1.3 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 Byte/Page Program, erase, Program Security Register, Erase Security Register, or Write Status Register commands. The WEL bit defaults to the logical “0” state after a device power-up or reset operation. 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
- Program Security Register operation completes successfully or aborts
- Erase Security Register operation completes successfully or aborts
- Byte/Page Program operation completes successfully or aborts
- Block Erase operation completes successfully or aborts
- Chip Erase operation completes successfully or 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 Byte/Page Program, erase, Program Security Register, Erase Security Register, or Write Status Register command must have been clocked into the device.

### 10.1.4 $\overline{RDY}$/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”.

### 10.1.5 LB3, LB2, LB1 Bits

The LB3, LB2, and LB1 bits are used to determine if any of the three Security Register pages are locked.

The LB3 bit is in the logical “1” state if Security Register page-2 is locked and cannot be erased or programmed.

The LB2 bit is in the logical “1” state if Security Register page-1 is locked and cannot be erased or programmed.

The LB1 bit is in the logical “1” state if Security Register page-0 is locked and cannot be erased or programmed.

### 10.1.6 QE Bit

The QE bit is used to determine if the device is in the Quad Enabled mode. If the QE bit is in the logical “1” state, then the $\overline{HOLD}$ and $\overline{WP}$ pins functions as input/output pins similar to the SI and SO. If the QE bit is in the logical “0” state, then the $\overline{HOLD}$ pin functions as an input only and the $\overline{WP}$ pin functions as an input only.

## 10.2 Write Status Register (01h)

The Write Status Register command is used to modify the Block Protection, Security Register Lock-down, Quad Enable, and Status Register Protection. 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 or two bytes of data. The first byte of data consists of the SRP0, SEC, TB, BP2, BP1, BP0 bit values and 2 dummy bits. The second byte is optional and consists of 1 dummy bit, the CMP, LB3, LB2, LB1, 1 dummy bit, the QE, and 1 dummy bit. When the $\overline{CS}$ pin is deasserted, the bit values 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 or two bytes 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 Status Register bits will not change, memory protection status will not change, and the WEL bit in the Status Register will be reset back to the logical “0” state

**Table 10-4. Write Status Register Format.**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| SRPO | SEC | TB | BP2 | BP1 | BP0 | WEL | $\overline{RDY}$/BSY |

**Figure 10-3. Write Status Register**

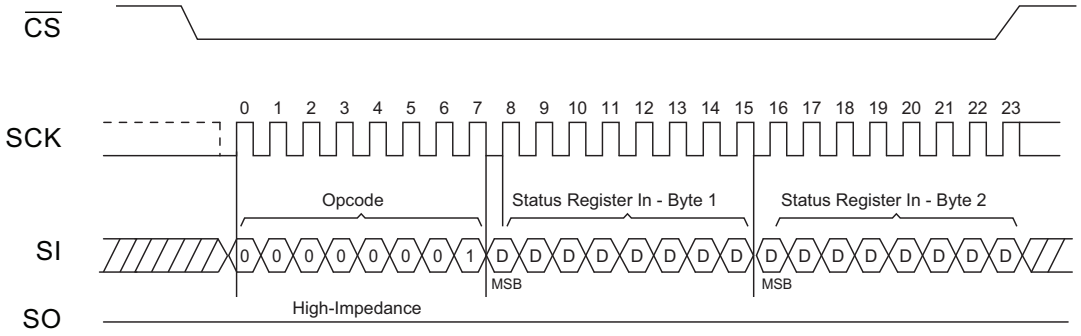



**Table 10-5. Write Status Register Byte 2**

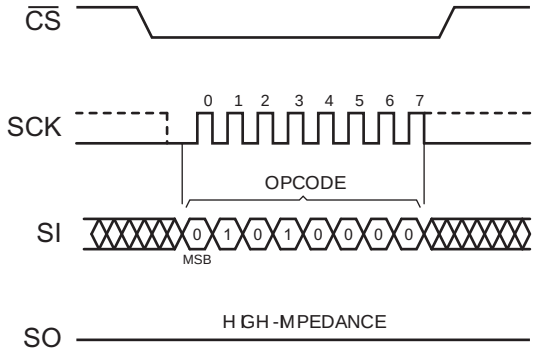
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| SUS | CMP | LB3 | LB2 | LB1 | reserved | QE | SRP1 |

## 10.3 Write Enable for Volatile Status Register (50h)

The non-volatile Status Register bits described in Table 10-1 and Table 10-2 can also be written to as volatile bits. During power up reset, the non-volatile Status Register bits are copied to a volatile version of the Status Register that is used during device operation. This gives more flexibility to change the system configuration and memory protection schemes quickly without waiting for the typical non-volatile bit write cycles or affecting the endurance of the Status Register non-volatile bits.

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

**Figure 10-4. Write Enable for Volatile Status Register**




# 11. Other Commands and Functions

The AT25SF321 supports three different commands to access device identification that indicates the manufacturer, device type, and memory density. The returned data bytes provide information as shown in Table 11-1.

**Table 11-1. Manufacturer and Device ID Information**

| Instruction | Opcode | Dummy Bytes | Manufacturer ID (Byte #1) | Device ID (Byte #2) | Device ID (Byte #3) |
|---|---|---|---|---|---|
| Read Manufacturer and Device ID | 9Fh | 0 | 1Fh | 87h | 01h |
| Read ID (Legacy Command) | 90h | 3 | 1Fh | | 15h |
| Resume from Deep Power-Down and Read Device ID | ABh | 3 | | | 15h |

## 11.1 Read Manufacturer and Device ID (9Fh)

Identification information can be read from the device to enable systems to electronically query and identify the device while it is in system.

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. 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.

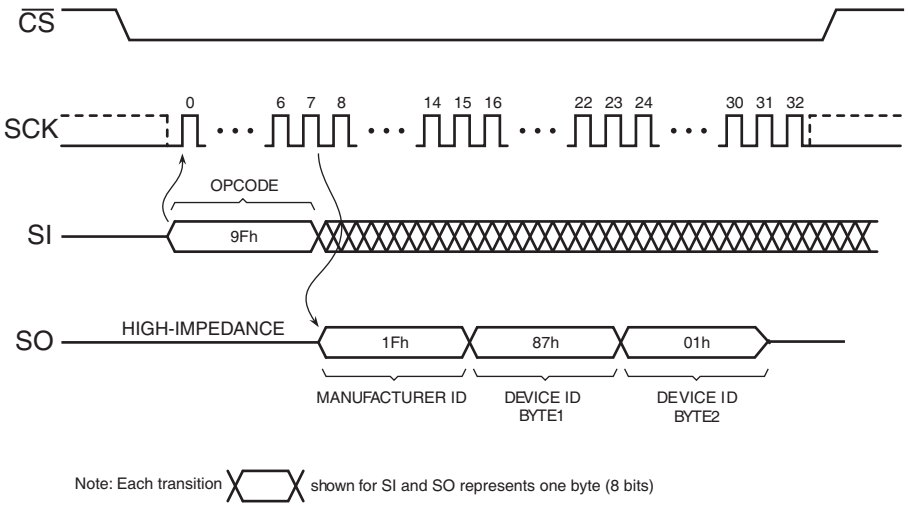
**Table 11-2. Manufacturer and Device ID Information**

| Byte No. | Data Type | Value |
|---|---|---|
| 1 | Manufacturer ID | 1Fh |
| 2 | Device ID (Part 1) | 87h |
| 3 | Device ID (Part 2) | 01h |

**Table 11-3. 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 | | | | | 87h | Family Code:100 (AT25SFxxx series)<br>Density Code: 00111 (32-Mbit) |
| | 1 | 0 | 0 | 0 | 0 | 1 | 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 11-1. Read Manufacturer and Device ID**



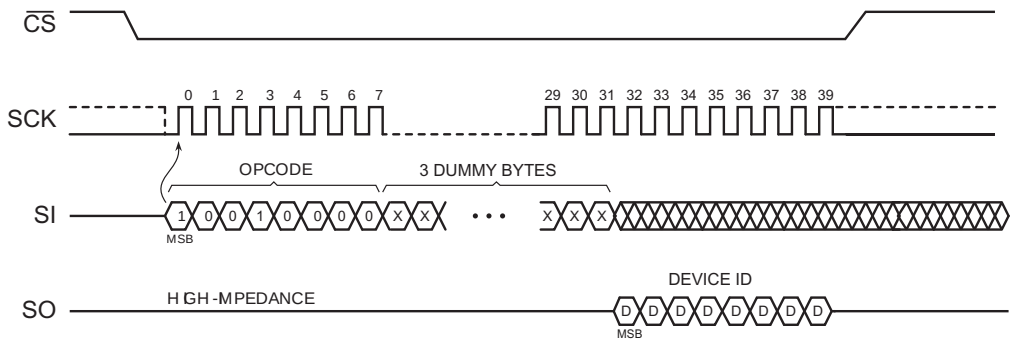

## 11.2 Read ID (Legacy Command) (90h)

Identification information can be read from the device to enable systems to electronically query and identify the device while it is in system. The preferred method for doing so is the JEDEC standard "Read Manufacturer and Device ID (9Fh)" method described in Section 11.1 on page 30; however, the legacy Read ID command is supported on the AT25SF321 to enable backwards compatibility to previous generation devices.

To read the identification information, the $\overline{CS}$ pin must first be asserted and the opcode of 90h must be clocked into the device, followed by three dummy bytes. After the opcode has been clocked in followed by three dummy bytes, 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 of 1Fh followed by a single byte of data representing a device code of 15h. After the device code is output, the sequence of bytes will repeat.

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

**Figure 11-2. Read ID (Legacy Command)**




## 11.3 Deep Power-Down (B9h)

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.

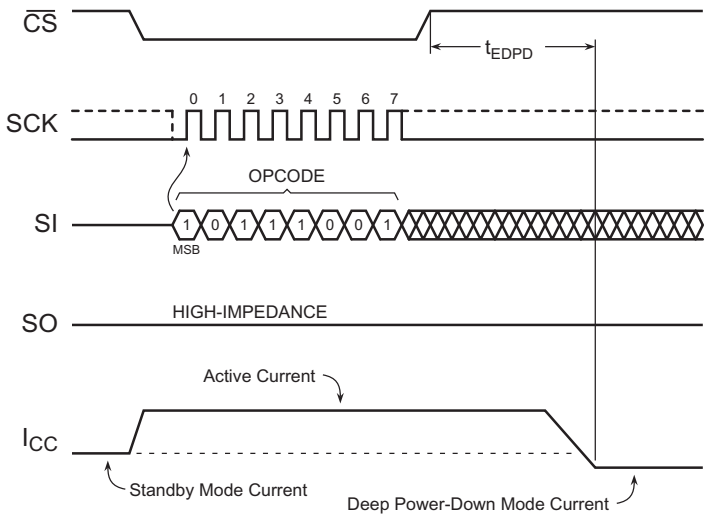
When the device is in the Deep Power-Down mode, all commands including the Read Status Register command will be ignored with the exception of the Resume from Deep Power-Down command. Since all commands will be ignored, the mode can be used as an extra protection mechanism against program and erase operations.

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 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 11-3. Deep Power-Down**



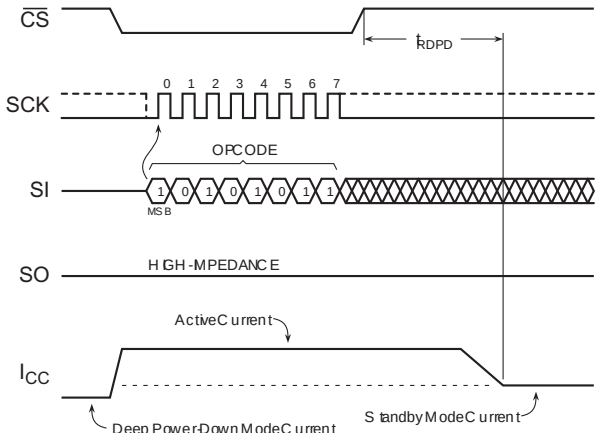
## 11.4 Resume from Deep Power-Down (ABh)

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 recognize while in the Deep Power-Down mode.

To resume from the Deep Power-Down mode, the $\overline{CS}$ pin must first be asserted and the 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 byte boundary (multiples of eight bits), then the device will abort the operation and return to the Deep Power-Down mode.

**Figure 11-4. Resume from Deep Power-Down**




### 11.4.1 Resume from Deep Power-Down and Read Device ID (ABh)

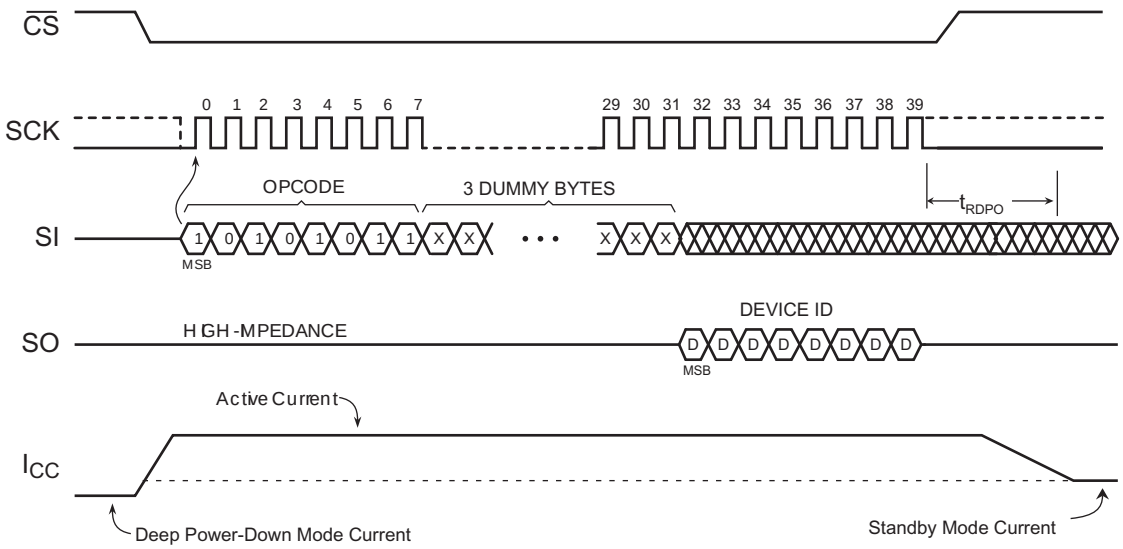
The Resume from Deep Power-Down command can also be used to read the Device ID.

When used to release the device from the Power-Down state and obtain the Device ID, the $\overline{CS}$ pin must first be asserted and opcode of ABh must be clocked into the device, followed by 3 dummy bytes. The Device ID bits are then shifted out on the falling edge of SCLK with most significant bit (MSB) first as shown in Figure 11-4. This command only outputs a single byte Device ID. The Device ID value for the AT25SF321 is listed in Table 11-1.

After the last bit (bit 0) of the Device ID has been clocked out, the sequence will repeat itself starting again with bit 7 as long as the CS pin remains asserted and the SCK pin is being pulsed. After $\overline{CS}$ is deasserted it must remain high for a time duration of $t_{RDPO}$ before new commands can be received.

The same instruction may be used to read device ID when not in power down. In that case, /CS does not have to remain high remain after it is deasserted.

**Figure 11-5. Resume from Deep Power-Down and Read Device ID**




## 11.5 Hold Function

The $\overline{\text{HOLD}}$ 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 $\overline{\text{HOLD}}$ pin will not pause the operation, and the erase cycle will continue until it is finished.

If the QE bit value in the Status Register has been set to logical “1”, then the $\overline{\text{HOLD}}$ pin does not function as a control pin. The $\overline{\text{HOLD}}$ pin will function as an output for Quad-Output Read and input/output for Quad-I/O Read.

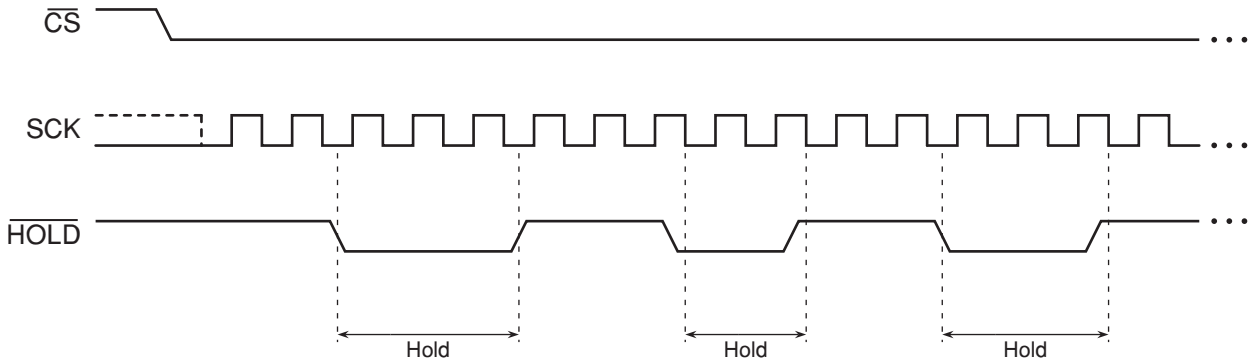
The Hold mode can only be entered while the $\overline{\text{CS}}$ pin is asserted. The Hold mode is activated simply by asserting the $\overline{\text{HOLD}}$ pin during the SCK low pulse. If the $\overline{\text{HOLD}}$ pin is asserted during the SCK high pulse, then the Hold mode won’t be started until the beginning of the next SCK low pulse. The device will remain in the Hold mode as long as the $\overline{\text{HOLD}}$ pin and $\overline{\text{CS}}$ 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 $\overline{\text{WP}}$ 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 11-6. Hold Mode**



# 12. Electrical Specifications

## 12.1 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.

## 12.2 DC and AC Operating Range

| | | AT25SF321 |
|---|---|---|
| Operating Temperature (Case) | Industrial | -40°C to 85°C |
| $V_{CC}$ Power Supply | | 2.5V to 3.6V |

## 12.3 DC Characteristics

| Symbol | Parameter | Condition | 2.5V to 3.6V | | | Units |
|---|---|---|---|---|---|---|
| | | | Min | Typ | Max | |
| $I_{DPD}$ | Deep Power-Down Current | $\overline{CS}$, $\overline{HOLD}$, $\overline{WP}$ = $V_{IH}$ All inputs at CMOS levels | | 2 | 5 | µA |
| $I_{SB}$ | Standby Current | $\overline{CS}$, $\overline{HOLD}$, $\overline{WP}$ = $V_{IH}$ All inputs at CMOS levels | | 13 | 25 | µA |
| $I_{CC1}$ (1)(3) | Active Current, Read (03h, 0Bh) Operation | f = 20MHz; $I_{OUT}$ = 0mA | | 3 | 6 | mA |
| | | f = 50MHz; $I_{OUT}$ = 0mA | | 4 | 7 | mA |
| | | f = 85MHz; $I_{OUT}$ = 0mA | | 5 | 8 | mA |
| $I_{CC2}$ (1) | Active Current,(3Bh, BBh Read Operation (Dual) | f = 50MHz; $I_{OUT}$ = 0mA | | 5 | 8 | mA |
| | | f = 85MHz; $I_{OUT}$ = 0mA | | 6 | 10 | mA |
| $I_{CC3}$ (1) | Active Current,(6Bh, EBh Read Operation (Quad) | f = 50MHz; $I_{OUT}$ = 0mA | | 6 | 10 | mA |
| | | f = 85MHz; $I_{OUT}$ = 0mA | | 8 | 14 | mA |
| $I_{CC4}$ (1) | Active Current, Program Operation | $\overline{CS}$ = $V_{CC}$ | | 10 | 16 | mA |
| $I_{CC5}$ (1) | Active Current, Erase Operation | $\overline{CS}$ = $V_{CC}$ | | 10 | 16 | mA |

| Symbol | Parameter | Condition | 2.5V to 3.6V | | | Units |
|---|---|---|---|---|---|---|
| | | | Min | Typ | Max | |
| $I_{LI}$ | 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 0.3 | V |
| $V_{IH}$ | Input High Voltage | | $V_{CC}$ x 0.7 | | | V |
| $V_{OL}$ | Output Low Voltage | $I_{OL}$ = 1.6mA; $V_{CC}$ = 2.5V | | | 0.4 | V |
| $V_{OH}$ | Output High Voltage | $I_{OH}$ = -100µA | $V_{CC}$ - 0.2V | | | V |

1. Typical values measured at 3.0V @ 25°C for the 2.5V to 3.6V range

## 12.4 AC Characteristics - Maximum Clock Frequencies

| Symbol | Parameter | 2.5V to 3.6V | | | 2.7V to 3.6V | | | Units |
|---|---|---|---|---|---|---|---|---|
| | | Min | Typ | Max | Min | Typ | Max | |
| $f_{CLK}$ | Maximum Clock Frequency for All Operations (excluding opcodes below) | | | 104 | | | 104 | MHz |
| $f_{RDLF}$ | Maximum Clock Frequency for 03h Opcode | | | 50 | | | 50 | MHz |
| $f_{RDHF}$ | Maximum Clock Frequency for 0Bh Opcode | | | 70 | | | 85 | MHz |
| $f_{RDDO}$ | Maximum Clock Frequency for 3B, BBh Opcode | | | 70 | | | 85 | MHz |
| $f_{RDQO}$ | Maximum Clock Frequency for 6B, EBh Opcode | | | 70 | | | 85 | MHz |

## 12.5 AC Characteristics - All Other Parameters

| Symbol | Parameter | 2.5V to 3.6V | | | 2.7V to 3.6V | | | Units |
|---|---|---|---|---|---|---|---|---|
| | | Min | Typ | Max | Min | Typ | Max | |
| $t_{CLKH}$ | Clock High Time | 5 | | | 5 | | | ns |
| $t_{CLKL}$ | Clock Low Time | 5 | | | 5 | | | 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 | 10 | | | 10 | | | ns |
| $t_{CSLS}$ | Chip Select Low Setup Time (relative to Clock) | 5 | | | 5 | | | ns |
| $t_{CSLH}$ | Chip Select Low Hold Time (relative to Clock) | 5 | | | 5 | | | ns |
| $t_{CSHS}$ | Chip Select High Setup Time (relative to Clock) | 5 | | | 5 | | | ns |

## 12.5 AC Characteristics - All Other Parameters

| Symbol | Parameter | 2.5V to 3.6V | | | 2.7V to 3.6V | | | Units |
|---|---|---|---|---|---|---|---|---|
| | | Min | Typ | Max | Min | Typ | Max | |
| $t_{CSHH}$ | Chip Select High Hold Time (relative to Clock) | 5 | | | 5 | | | ns |
| $t_{DS}$ | Data In Setup Time | 2 | | | 2 | | | ns |
| $t_{DH}$ | Data In Hold Time | 2 | | | 2 | | | ns |
| $t_{DIS}$ (1) | Output Disable Time | | | 7 | | | 6 | ns |
| $t_V$ | Output Valid Time (03h, 0Bh) | | | 7 | | | 6 | ns |
| | Output Valid Time (3Bh, BBh - Dual) | | | 7 | | | 7 | |
| | Output Valid Time (6Bh, EBh - Quad) | | | 8 | | | 8 | |
| $t_{OH}$ | Output Hold Time | 0 | | | 0 | | | ns |
| $t_{HLS}$ | $\overline{HOLD}$ Low Setup Time (relative to Clock) | 5 | | | 5 | | | ns |
| $t_{HLH}$ | $\overline{HOLD}$ Low Hold Time (relative to Clock) | 5 | | | 5 | | | ns |
| $t_{HHS}$ | $\overline{HOLD}$ High Setup Time (relative to Clock) | 5 | | | 5 | | | ns |
| $t_{HHH}$ | $\overline{HOLD}$ High Hold Time (relative to Clock) | 5 | | | 5 | | | ns |
| $t_{HLQZ}$ (1) | $\overline{HOLD}$ Low to Output High-Z | | | 6 | | | 6 | ns |
| $t_{HHQZ}$ (1) | $\overline{HOLD}$ High to Output High-Z | | | 6 | | | 6 | ns |
| $t_{WPS}$ (1) (2) | Write Protect Setup Time | 20 | | | 20 | | | ns |
| $t_{WPH}$ (1)(2) | Write Protect Hold Time | 100 | | | 100 | | | ns |
| $t_{EDPD}$ (1) | Chip Select High to Deep Power-Down | | | 1 | | | 1 | µs |
| $t_{RDPD}$ (1) | Chip Select High to Standby Mode | | | 5 | | | 5 | µs |
| $t_{RDPO}$ (1) | Resume Deep Power-Down, $\overline{CS}$ High to ID | | | 5 | | | 5 | µs |

1. Not 100% tested (value guaranteed by design and characterization).
2. Only applicable as a constraint for the Write Status Register command when BPL = 1.

## 12.6 Program and Erase Characteristics

| Symbol | Parameter | | 2.5 to 3.6V | | | 2.7V to 3.6V | | | Units |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Typ | Max | Min | Typ | Max | |
| $t_{PP}$ (1) | Page Program Time (256 Bytes) | | | 0.7 | 5 | | 0.7 | 3.0 | ms |
| $t_{BP}$ | Byte Program Time | | | 5 | | | 5 | | µs |
| $t_{BLKE}$ (1) | Block Erase Time | 4 Kbytes | | 60 | 300 | | 60 | 300 | ms |
| | | 32 Kbytes | | 300 | 1300 | | 300 | 1300 | |
| | | 64 K bytes | | 500 | 3000 | | 500 | 3000 | |
| $t_{CHPE}$ (1) (2) | Chip Erase Time | | | 25 | 60 | | 25 | 60 | sec |

## 12.6 Program and Erase Characteristics

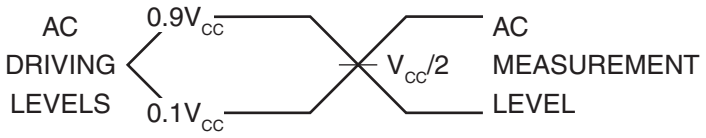
| Symbol | Parameter | 2.5 to 3.6V | | | 2.7V to 3.6V | | | |
|---|---|---|---|---|---|---|---|---|
| | | Min | Typ | Max | Min | Typ | Max | Units |
| $t_{SRP}$[1] | Security Register Program Time | | | 2.5 | | | 2.5 | ms |
| $t_{SRP}$[1] | Security Register Erase Time | | | 15 | | | 15 | ms |
| $t_{WRSR}$[2] | Write Status Register Time | | | 15 | | | 15 | ms |

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

## 12.7 Power-Up Conditions

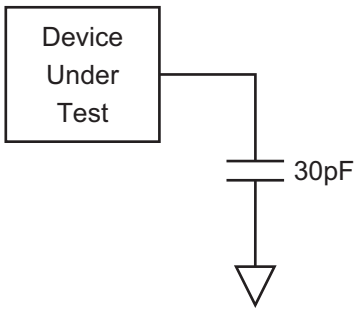
| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_{VCSL}$ | Minimum $V_{CC}$ to Chip Select Low Time | 20 | | µs |
| $t_{PUW}$ | Power-up Device Delay Before Program or Erase Allowed | | 10 | ms |
| $V_{POR}$ | Power-on Reset Voltage | | 2.3 | V |

## 12.8 Input Test Waveforms and Measurement Levels




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

## 12.9 Output Test Load




# 13. AC Waveforms

**Figure 13-1. Serial Input Timing**

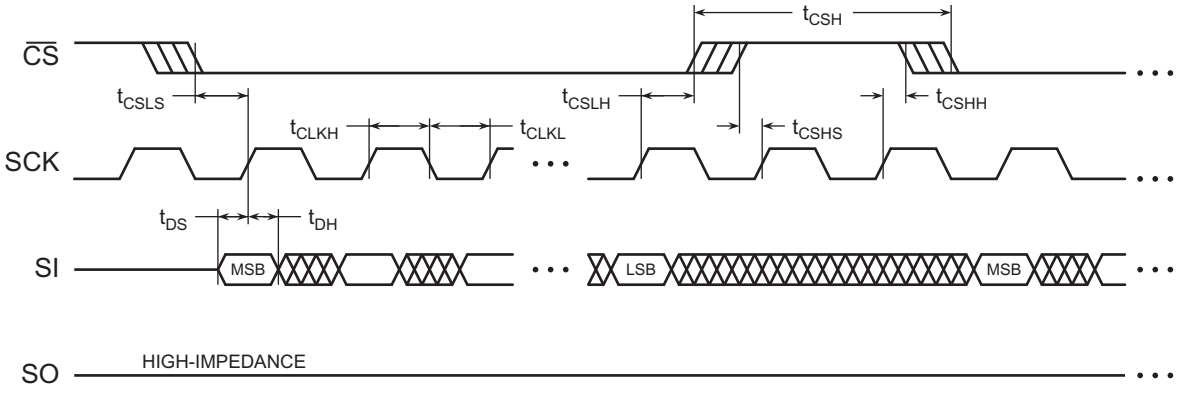



**Figure 13-2. Serial Output Timing**

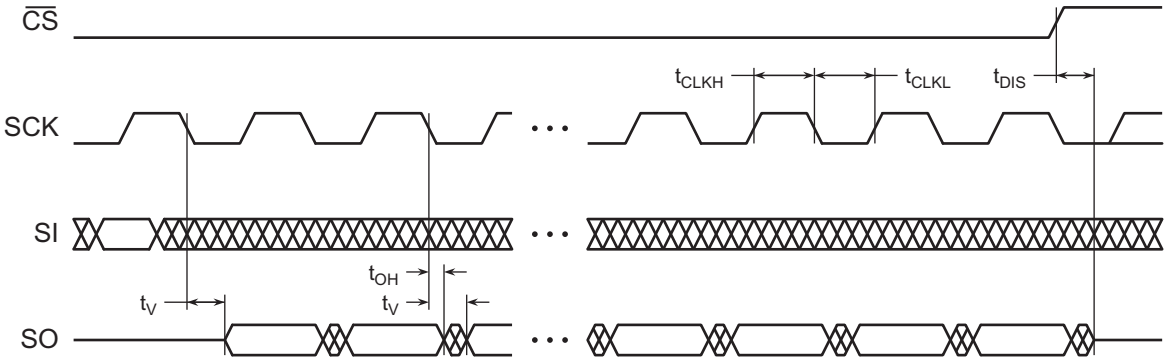


**Figure 13-3. $\overline{WP}$ Timing for Write Status Register Command When BPL = 1**

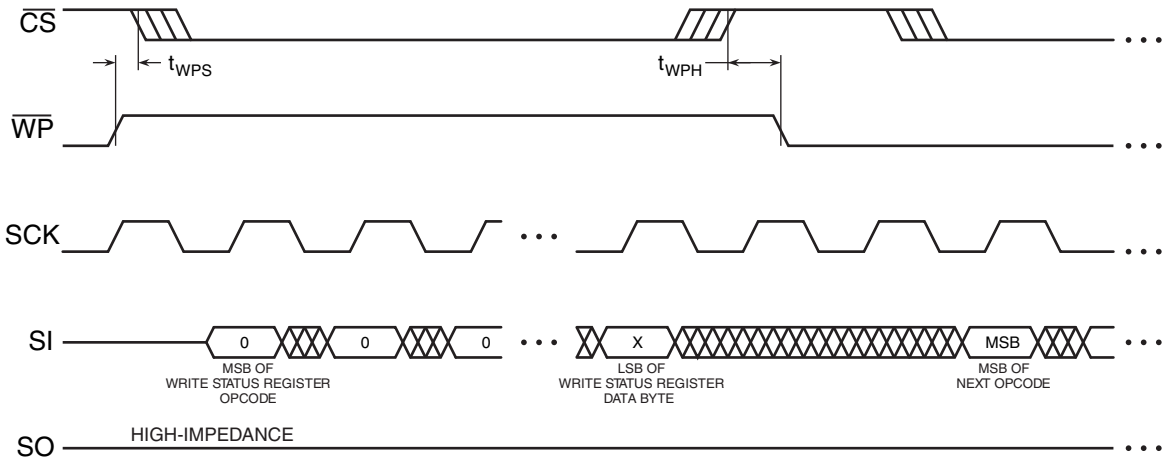



**Figure 13-4. $\overline{HOLD}$ Timing – Serial Input**

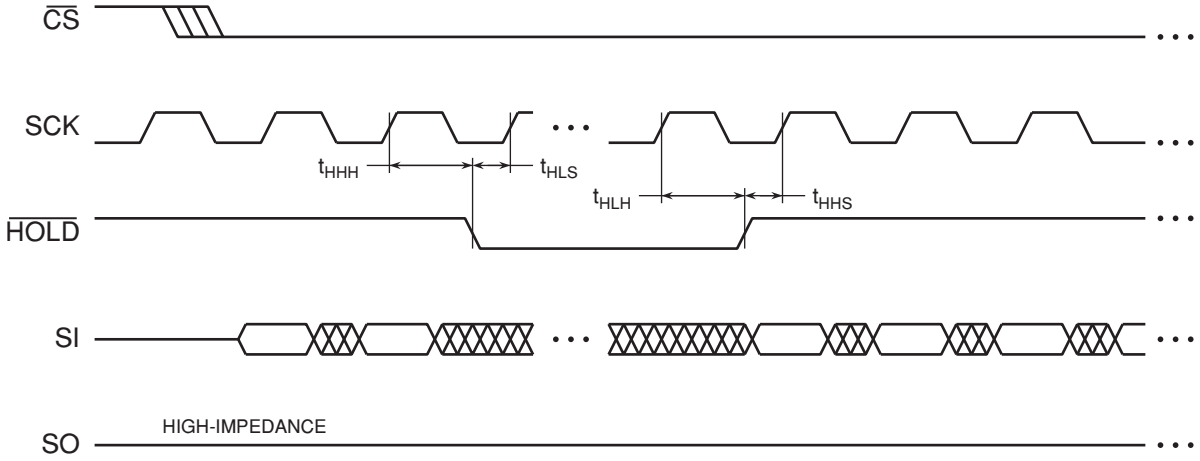
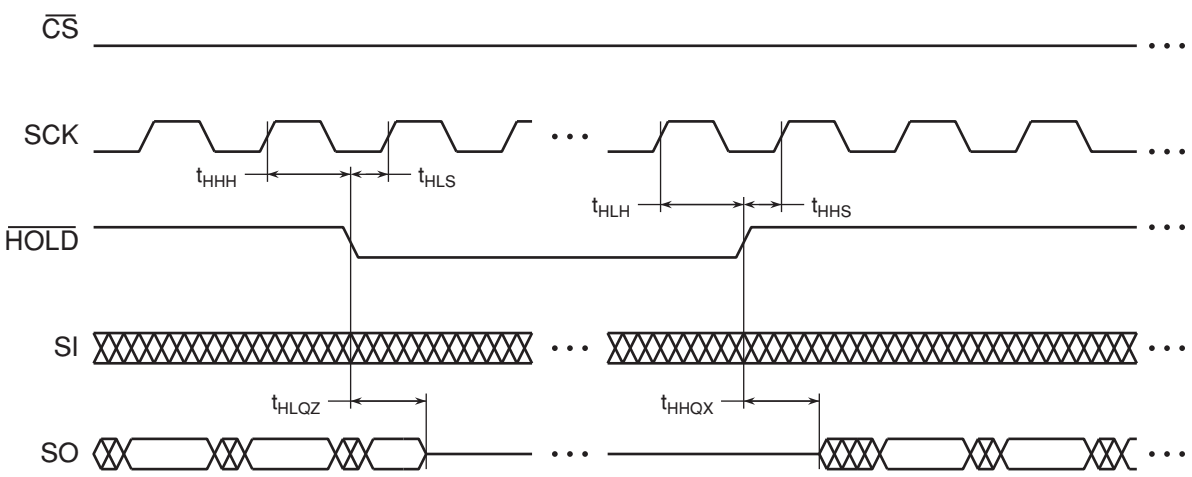
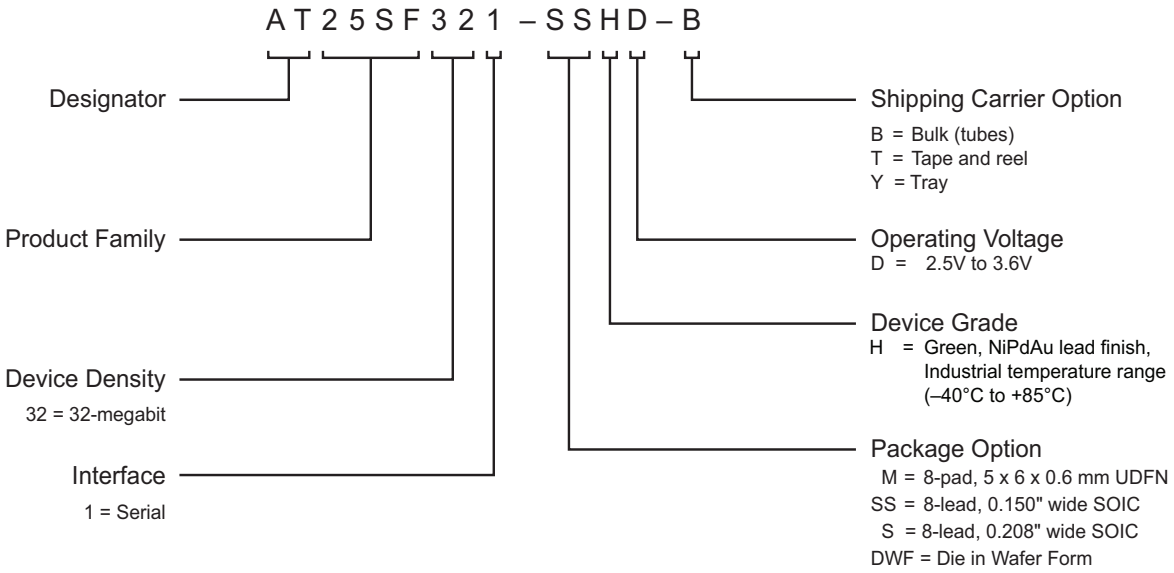



**Figure 13-5. $\overline{HOLD}$ Timing – Serial Output**



# 14. Ordering Information

## 14.1 Ordering Code Detail



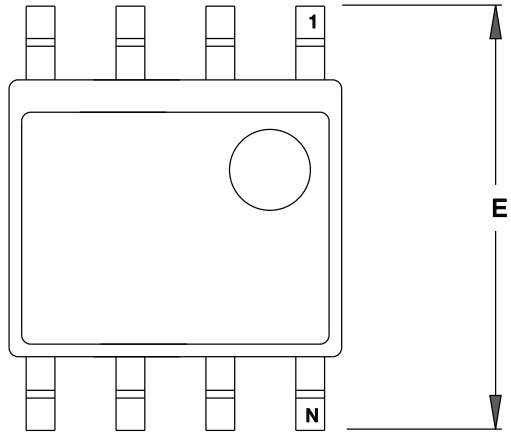

| Ordering Code [1] | Package | Operating Voltage | Max. Freq. (MHz) | Operation Range |
|---|---|---|---|---|
| AT25SF321-SSHD-B | 8S1 | 2.5V to 3.6V | 85MHz | Industrial (-40°C to +85°C) |
| AT25SF321-SSHD-T | | | | |
| AT25SF321-SHD-B | 8S2 | | | |
| AT25SF321-SHD-T | | | | |
| AT25SF321-MHD-T | 8MA1 | | | |
| AT25SF321-DWF [2] | DWF | | | |

1. The shipping carrier option code is not marked on the devices.
2. Contact Adesto for mechanical drawing or Die Sales information.

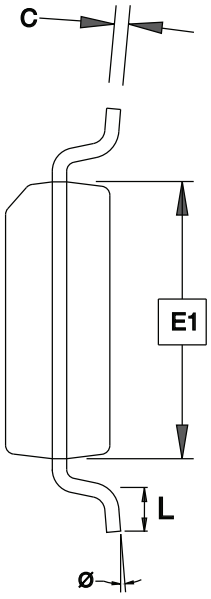
| Package Type | |
|---|---|
| **8S1** | 8-lead, 0.150" Wide, Plastic Gull Wing Small Outline Package (JEDEC SOIC) |
| **8S2** | 8-lead, 0.208" Wide, Plastic Gull Wing Small Outline Package (EIAJ SOIC) |
| **8MA1** | 8-pad (5 x 6 x 0.6mm body), Thermally Enhanced Plastic Ultra Thin Dual Flat No-lead (UDFN) |
| **DWF** | Die in Wafer Form |

# 15. Packaging Information

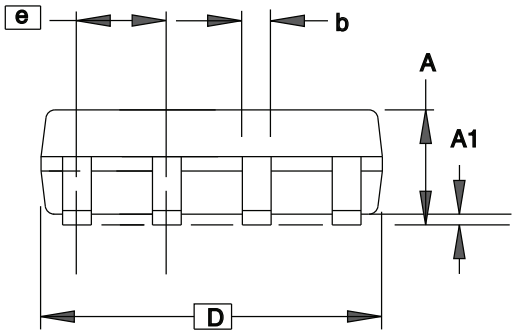
## 15.1 8S1 – 8-lead, .150” JEDEC SOIC




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.

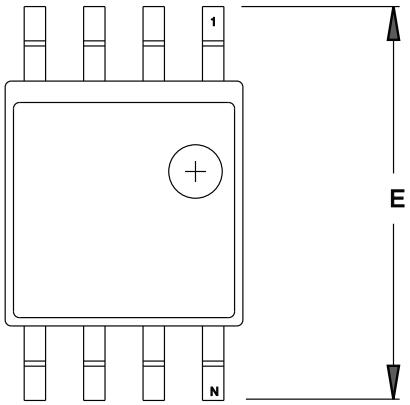
COMMON DIMENSIONS
(Unit of Measure = mm)

| 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° | |

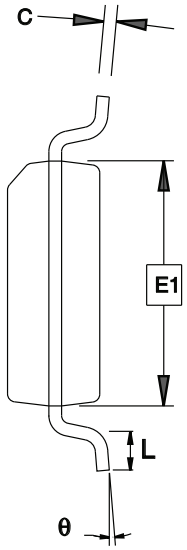
5/19/10

| 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>F |
|---|---|---|---|---|

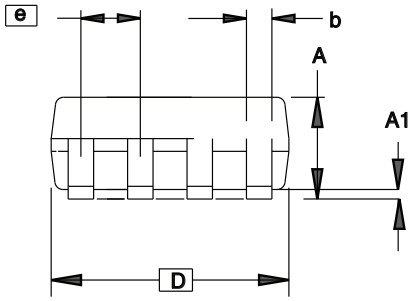
## 15.2 8S2 – 8-lead, .208” EIAJ SOIC




TOP VIEW




END VIEW




SIDE VIEW

COMMON DIMENSIONS
(Unit of Measure = mm)

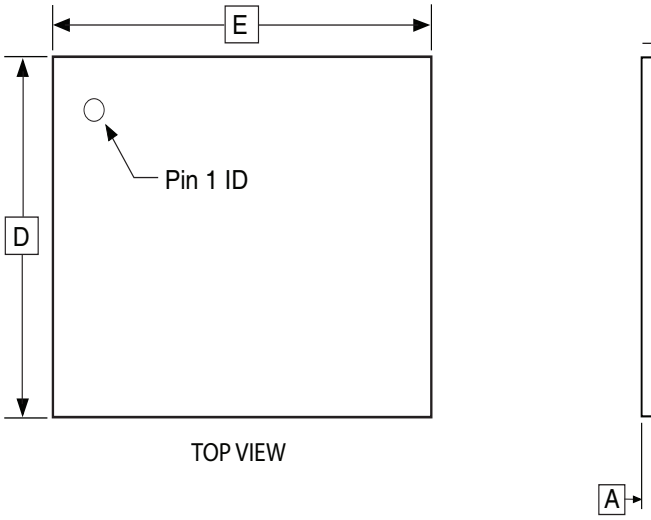
| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 1.70 | | 2.16 | |
| A1 | 0.05 | | 0.25 | |
| b | 0.35 | | 0.48 | 4 |
| C | 0.15 | | 0.35 | 4 |
| D | 5.13 | | 5.35 | |
| E1 | 5.18 | | 5.40 | 2 |
| E | 7.70 | | 8.26 | |
| L | 0.51 | | 0.85 | |
| θ | 0° | | 8° | |
| e | 1.27 BSC | | | 3 |

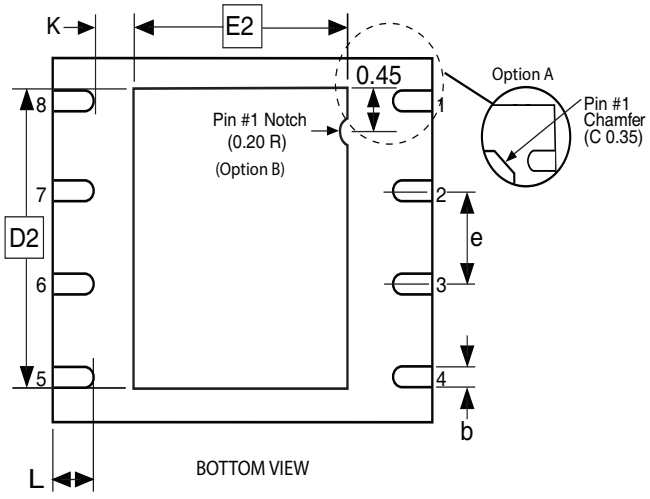
Notes: 1. This drawing is for general information only; refer to EIAJ Drawing EDR-7320 for additional information.
2. Mismatch of the upper and lower dies and resin burrs aren't included.
3. Determines the true geometric position.
4. Values b,C apply to plated terminal. The standard thickness of the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

4/15/08

| adesto® TECHNOLOGIES<br>Package Drawing Contact:<br>contact@adestotech.com | TITLE<br>8S2, 8-lead, 0.208” Body, Plastic Small Outline Package (EIAJ) | GPC<br>STN | DRAWING NO.<br>8S2 | REV.<br>F |
|---|---|---|---|---|

## 15.3 8MA1 – UDFN




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 |

# 16. Revision History

| Revision Level – Release Date | History |
|---|---|
| A – June 2015 | Initial release |
| B – July 2015 | Updated operation description in Section 7.4 (Program/Erase Suspend) and references in Table 7-1. Updated Table 8-2. Updated specifications for $I_{cc3}$ at 85 MHz and Chip Erase Time (max). |
| C – August 2015 | Added Die in Wafer Form ordering option. |
| D – January 2016 | Updated specification for $T_{PP.}$ Changed document status from Preliminary to Complete. |
| E – April 2016 | Updated 90h opcode description (added 3 dummy bytes). Added Suspend bit description. |
| F – August 2017 | Updated corporate address. |

## Corporate Office

California | USA
Adesto Headquarters
3600 Peterson Way
Santa Clara, CA 95054
Phone: (+1) 408.400.0578
Email: contact@adestotech.com



© 2017 Adesto Technologies. All rights reserved. / Rev.: DS-25SF321–047F–8/2017

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.

Стандарт Электрон Связь

Мы молодая и активно развивающаяся компания в области поставок электронных компонентов. Мы поставляем электронные компоненты отечественного и импортного производства напрямую от производителей и с крупнейших складов мира.

Благодаря сотрудничеству с мировыми поставщиками мы осуществляем комплексные и плановые поставки широчайшего спектра электронных компонентов.

Собственная эффективная логистика и склад в обеспечивает надежную поставку продукции в точно указанные сроки по всей России.

Мы осуществляем техническую поддержку нашим клиентам и предпродажную проверку качества продукции. На все поставляемые продукты мы предоставляем гарантию .

Осуществляем поставки продукции под контролем ВП МО РФ на предприятия военно-промышленного комплекса России , а также работаем в рамках 275 ФЗ с открытием отдельных счетов в уполномоченном банке. Система менеджмента качества компании соответствует требованиям ГОСТ ISO 9001.

Минимальные сроки поставки, гибкие цены, неограниченный ассортимент и индивидуальный подход к клиентам являются основой для выстраивания долгосрочного и эффективного сотрудничества с предприятиями радиоэлектронной промышленности, предприятиями ВПК и научно-исследовательскими институтами России.

С нами вы становитесь еще успешнее!

**Наши контакты:**

**Телефон:** +7 812 627 14 35

**Электронная почта:** sales@st-electron.ru

**Адрес:** 198099, Санкт-Петербург,
Промышленная ул, дом № 19, литера Н,
помещение 100-Н Офис 331