Atmel AT25DQ161

16-Mbit, 2.7V Minimum SPI Serial Flash Memory with Dual-I/O and Quad-I/O Support

DATASHEET

Not Recommended for New Designs
Use AT25DQ321

Features

- Single 2.7V - 3.6V supply
- Serial Peripheral Interface (SPI) compatible
  - Supports SPI Modes 0 and 3
  - Supports Atmel® RapidS™ operation
  - Supports Dual- and Quad-Input Program
  - Supports Dual- and Quad-Output Read
- Very high operating frequencies
  - 100MHz for RapidS
  - 85MHz for SPI
  - Clock-to-output time (t_V) of 5ns maximum
- Flexible, optimized erase architecture for code + data storage applications
  - Uniform 4KB, 32KB, and 64KB Block Erase
  - Full Chip Erase
- Individual sector protection with Global Protect/Unprotect feature
  - 32 sectors of 64KB each
- Hardware controlled locking of protected sectors via $\overline{WP}$ pin
- Sector Lockdown
  - Make any combination of 64KB sectors permanently read-only
- 128-byte One-Time Programmable (OTP) Security Register
- Flexible programming
  - Byte/Page Program (1 to 256 bytes)
- Fast Program and Erase times
  - 1.0ms typical Page Program (256 bytes) time
  - 50ms typical 4KB Block Erase time
  - 250ms typical 32KB Block Erase time
  - 400ms typical 64KB Block Erase time
- Program and Erase Suspend/Resume
- Automatic checking and reporting of erase/program failures
- Software controlled reset
- JEDEC Standard Manufacturer and Device ID Read Methodology
- Low power dissipation
  - 5mA Active Read current (typical at 20MHz)
  - 5µA Deep Power-Down 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 (0.150" and 0.208" wide)
  - 8-pad Ultra Thin UDFN (5 x 6 x 0.6mm)

1. Description

The Atmel AT25DQ161 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 AT25DQ161, with its erase granularity as small as 4KB, makes it ideal for data storage as well, eliminating the need for additional data storage EEPROM devices.

The physical sectoring and the erase block sizes of the AT25DQ161 have been optimized to meet the needs of today's code and data storage applications. By optimizing the size of the physical sectors and 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 protected sectors, the wasted and unused memory space that occurs with large sectored and large block erase Flash memory devices can be greatly reduced. This increased memory space efficiency allows additional code routines and data storage segments to be added while still maintaining the same overall device density.

The AT25DQ161 also offers a sophisticated method for protecting individual sectors against erroneous or malicious program and erase operations. By providing the ability to individually protect and unprotect sectors, a system can unprotect a specific sector to modify its contents while keeping the remaining sectors of the memory array securely protected. This is useful in applications where the program code is patched, updated on a subroutine or module basis, or in applications where data storage segments need to be modified without running the risk of errant modifications to the program code segments. In addition to individual sector protection capabilities, the AT25DQ161 incorporates Global Protect and Global Unprotect features that allow the entire memory array to be either protected or unprotected all at once. This reduces overhead during the manufacturing process since sectors do not have to be unprotected one-by-one prior to initial programming.

To take code and data protection to the next level, the AT25DQ161 incorporates a sector lockdown mechanism that allows any combination of individual 64KB sectors to be locked down and become permanently read-only. This addresses the need of certain secure applications that require portions of the Flash memory array to be permanently protected against malicious attempts at altering program code, data modules, security information or encryption/decryption algorithms, keys, and routines. The device also contains a specialized OTP (One-Time Programmable) Security Register that can be used for purposes such as unique device serialization, system-level Electronic Serial Number (ESN) storage, locked key storage, etc.

Specifically designed for use in 3V systems, the AT25DQ161 supports read, program, and erase operations with a supply voltage range of 2.7V to 3.6V. No separate voltage is required for programming and erasing.

2. Pin Descriptions and Pinouts

Table 2-1. Pin Descriptions

Symbol	Name and Function	Asserted State	Type
$\overline{CS}$	**Chip Select:** Asserting the $\overline{CS}$ pin selects the device. When the $\overline{CS}$ pin is deasserted, the device will be deselected and normally be placed in standby mode (not Deep Power-Down mode), and the SO pin will be in a high-impedance state. When the device is deselected, data will not be accepted on the SI pin. 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 or I/O pins is always latched in on the rising edge of SCK, while output data on the SO pin or I/O pins is always clocked out on the falling edge of SCK.	—	Input
SI (I/O_0)	**Serial Input (I/O_0):** 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. With the Dual-Input and Quad-Input Byte/Page Program commands, the SI pin is used as an input pin (I/O_0) in conjunction with other pins to allow two bits (on I/O_{1-0}) or four bits (on I/O_{3-0}) of data to be clocked in on every rising edge of SCK. With the Dual-Output and Quad-Output Read Array commands, the SI pin becomes an output pin (I/O_0) and, along with other pins, allows two bits (on I/O_{1-0}) or four bits (on I/O_{3-0}) of data to be clocked out on every falling edge of SCK. To maintain consistency with SPI nomenclature, the SI (I/O_0) pin will be referenced as SI throughout the document with exception to sections dealing with the Dual-Input and Quad-Input Byte/Page Program commands as well as the Dual-Output and Quad-Output Read Array commands in which it will be referenced as I/O_0. Data present on the SI pin will be ignored whenever the device is deselected ($\overline{CS}$ is deasserted).	—	Input/Output
SO (I/O_1)	**Serial Output (I/O_1):** 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. With the Dual-Input and Quad-Input Byte/Page Program commands, the SO pin becomes an input pin (I/O_1) and, along with other pins, allows two bits (on I/O_{1-0}) or four bits (on I/O_{3-0}) of data to be clocked in on every rising edge of SCK. With the Dual-Output and Quad-Output Read Array commands, the SO pin is used as an output pin (I/O_1) in conjunction with other pins to allow two bits (on I/O_{1-0}) or four bits (on I/O_{3-0}) of data to be clocked out on every falling edge of SCK. To maintain consistency with SPI nomenclature, the SO (I/O_1) pin will be referenced as SO throughout the document with exception to sections dealing with the Dual-Input and Quad-Input Byte/Page Program commands as well as the Dual-Output and Quad-Output Read Array commands in which it will be referenced as I/O_1. The SO pin will be in a high-impedance state whenever the device is deselected ($\overline{CS}$ is deasserted).	—	Input/Output

Table 2-1. Pin Descriptions (Continued)

Symbol	Name and Function	Asserted State	Type
$\overline{WP}$ (I/O_2)	**Write Protect (I/O_2):** The WP# pin controls the hardware locking feature of the device. See "Protection Commands and Features" on page 24 for more details on protection features and the $\overline{WP}$ pin. 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 Array command, the $\overline{WP}$ pin becomes an output pin (I/O_2) and, when used with other pins, allows four bits (on I/O_{3-0}) of data to be clocked out on every falling edge of SCK. The QE bit in the Configuration Register must be set in order for the $\overline{WP}$ pin to be used as an I/O data pin. The $\overline{WP}$ pin must be driven at all times or pulled-high using an external pull-up resistor.	Low	Input
$\overline{HOLD}$ (I/O_3)	**Hold (I/O_3):** 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. 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 affect on internally self-timed operations such as a program or erase cycle. See "Hold" on page 53 for additional details on the Hold operation. 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 Array command, the $\overline{HOLD}$ pin becomes an output pin (I/O_3) and, when used with other pins, allows four bits (on I/O_{3-0}) of data to be clocked out on every falling edge of SCK. The QE bit in the Configuration Register must be set in order for the $\overline{HOLD}$ pin to be used as an I/O data pin. The $\overline{HOLD}$ pin must be driven at all times or pulled-high using an external pull-up resistor.	Low	Input
V_{CC}	**Device Power Supply:** The V_{CC} pin is used to supply the source voltage to the device. 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 2-1. 8-SOIC (Top View)

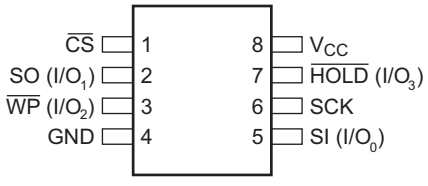
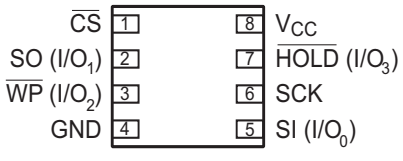
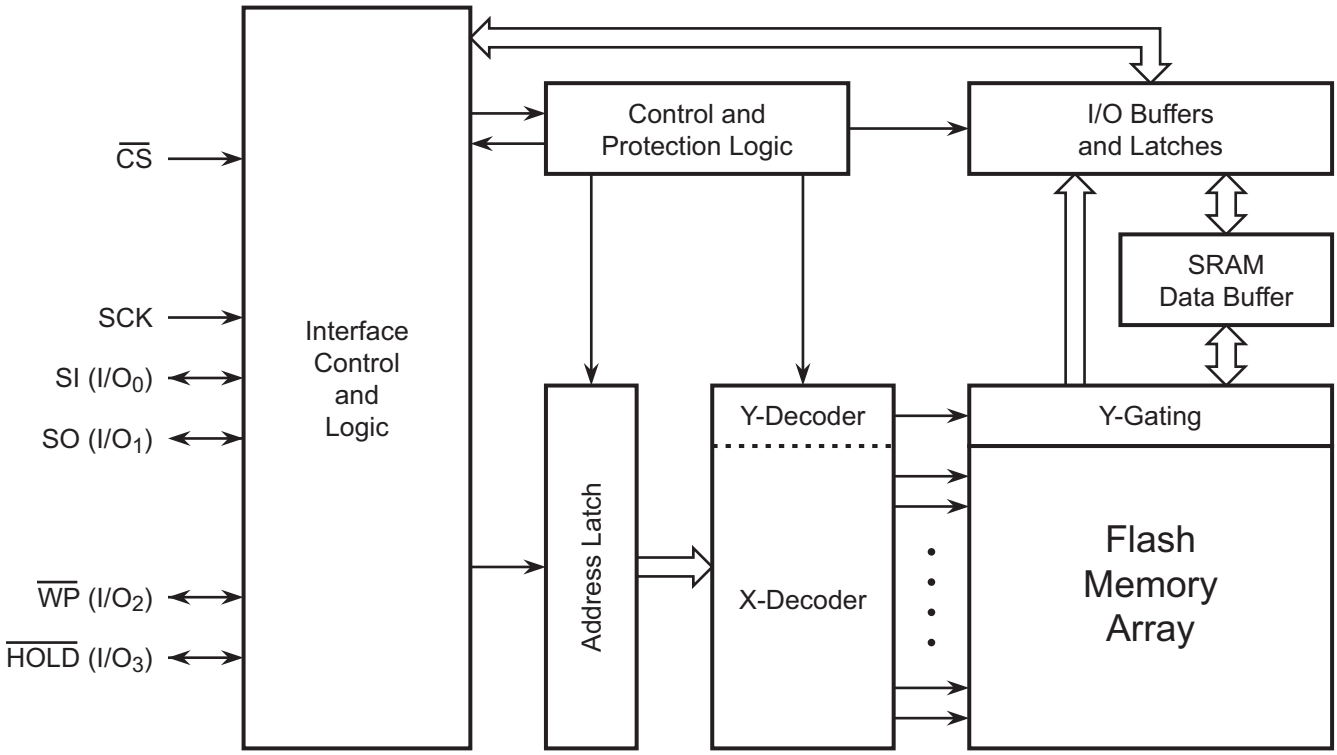


Figure 2-2. 8-UDFN (Top View)



3. Block Diagram

Figure 3-1. Block Diagram



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

4. Memory Array

To provide the greatest flexibility, the memory array of the AT25DQ161 can be erased in four levels of granularity including a full chip erase. In addition, the array has been divided into physical sectors of uniform size, of which each sector can be individually protected from program and erase operations. The size of the physical sectors is optimized for both code and data storage applications, allowing both code and data segments to reside in their own isolated regions. The Memory Architecture Diagram illustrates the breakdown of each erase level as well as the breakdown of each physical sector.

Figure 4-1. Memory Architecture Diagram

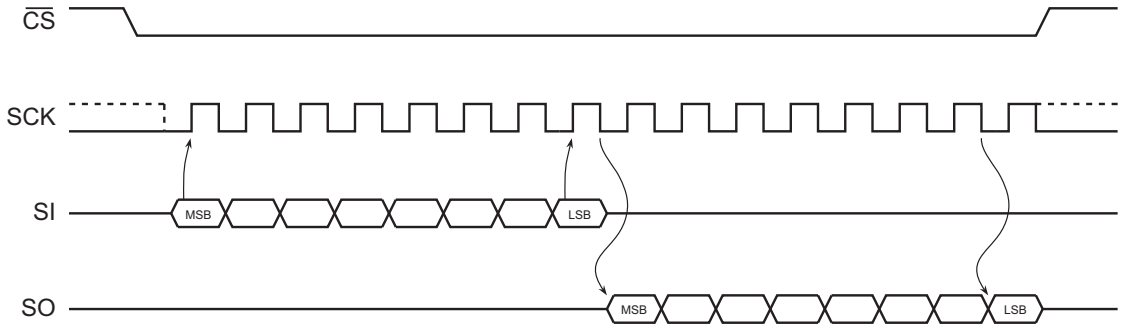
	Block Erase Detail				Page Program Detail	
Internal Sectoring for Protection, Lockdown, and Suspend Functions	**64KB Block Erase (D8h Command)**	**32KB Block Erase (52h Command)**	**4KB Block Erase (20h Command)**	**Block Address Range**	**1-256 Byte Page Program (02h Command)**	**Page Address Range**
64KB (Sector 31)	64KB	32KB	4KB	1FFFFFh – 1FF000h	256 bytes	1FFFFFh – 1FFF00h
			4KB	1FEFFFh – 1FE000h	256 bytes	1FFEFFh – 1FFE00h
			4KB	1FDFFFh – 1FD000h	256 bytes	1FFDFFh – 1FFD00h
			4KB	1FCFFFh – 1FC000h	256 bytes	1FFCFFh – 1FFC00h
			4KB	1FBFFFh – 1FB000h	256 bytes	1FFBFFh – 1FFB00h
			4KB	1FAFFFh – 1FA000h	256 bytes	1FFAFFh – 1FFA00h
			4KB	1F9FFFh – 1F9000h	256 bytes	1FF9FFh – 1FF900h
			4KB	1F8FFFh – 1F8000h	256 bytes	1FF8FFh – 1FF800h
		32KB	4KB	1F7FFFh – 1F7000h	256 bytes	1FF7FFh – 1FF700h
			4KB	1F6FFFh – 1F6000h	256 bytes	1FF6FFh – 1FF600h
			4KB	1F5FFFh – 1F5000h	256 bytes	1FF5FFh – 1FF500h
			4KB	1F4FFFh – 1F4000h	256 bytes	1FF4FFh – 1FF400h
			4KB	1F3FFFh – 1F3000h	256 bytes	1FF3FFh – 1FF300h
			4KB	1F2FFFh – 1F2000h	256 bytes	1FF2FFh – 1FF200h
			4KB	1F1FFFh – 1F1000h	256 bytes	1FF1FFh – 1FF100h
			4KB	1F0FFFh – 1F0000h	256 bytes	1FF0FFh – 1FF000h
64KB (Sector 30)	64KB	32KB	4KB	1EFFFFh – 1EF000h	256 bytes	1FEFFFh – 1FEF00h
			4KB	1EEFFFh – 1EE000h	256 bytes	1FEEFFh – 1FEE00h
			4KB	1EDFFFh – 1ED000h	256 bytes	1FEDFFh – 1FED00h
			4KB	1ECFFFh – 1EC000h	256 bytes	1FECFFh – 1FEC00h
			4KB	1EBFFFh – 1EB000h	256 bytes	1FEBFFh – 1FEB00h
			4KB	1EAFFFh – 1EA000h	256 bytes	1FEAFFh – 1FEA00h
			4KB	1E9FFFh – 1E9000h	256 bytes	1FE9FFh – 1FE900h
			4KB	1E8FFFh – 1E8000h	256 bytes	1FE8FFh – 1FE800h
		32KB	4KB	1E7FFFh – 1E7000h	⋮	
			4KB	1E6FFFh – 1E6000h		
			4KB	1E5FFFh – 1E5000h		
			4KB	1E4FFFh – 1E4000h	256 bytes	0017FFh – 001700h
			4KB	1E3FFFh – 1E3000h	256 bytes	0016FFh – 001600h
			4KB	1E2FFFh – 1E2000h	256 bytes	0015FFh – 001500h
			4KB	1E1FFFh – 1E1000h	256 bytes	0014FFh – 001400h
			4KB	1E0FFFh – 1E0000h	256 bytes	0013FFh – 001300h
⋮	⋮	⋮	⋮		256 bytes	0012FFh – 001200h
					256 bytes	0011FFh – 001100h
					256 bytes	0010FFh – 001000h
64KB (Sector 0)	64KB	32KB	4KB	00FFFFh – 00F000h	256 bytes	000FFFh – 000F00h
			4KB	00EFFFh – 00E000h	256 bytes	000EFFh – 000E00h
			4KB	00DFFFh – 00D000h	256 bytes	000DFFh – 000D00h
			4KB	00CFFFh – 00C000h	256 bytes	000CFFh – 000C00h
			4KB	00BFFFh – 00B000h	256 bytes	000BFFh – 000B00h
			4KB	00AFFFh – 00A000h	256 bytes	000AFFh – 000A00h
			4KB	009FFFh – 009000h	256 bytes	0009FFh – 000900h
			4KB	008FFFh – 008000h	256 bytes	0008FFh – 000800h
		32KB	4KB	007FFFh – 007000h	256 bytes	0007FFh – 000700h
			4KB	006FFFh – 006000h	256 bytes	0006FFh – 000600h
			4KB	005FFFh – 005000h	256 bytes	0005FFh – 000500h
			4KB	004FFFh – 004000h	256 bytes	0004FFh – 000400h
			4KB	003FFFh – 003000h	256 bytes	0003FFh – 000300h
			4KB	002FFFh – 002000h	256 bytes	0002FFh – 000200h
			4KB	001FFFh – 001000h	256 bytes	0001FFh – 000100h
			4KB	000FFFh – 000000h	256 bytes	0000FFh – 000000h

5. Device Operation

The AT25DQ161 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 AT25DQ161 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 AT25DQ161 supports the two most common modes, SPI Modes 0 and 3. The only difference between SPI Modes 0 and 3 is the polarity of the SCK signal when in the inactive state (when the SPI Master is in standby mode and not transferring any data). With SPI Modes 0 and 3, data is always latched in on the rising edge of SCK and always output on the falling edge of SCK.

Figure 5-1. SPI Mode 0 and 3



5.1 Dual-I/O and Quad-I/O Operation

The AT25DQ161 features a Dual-Input Program mode and a Dual-Output Read mode that allows two bits of data to be clocked into or out of the device every clock cycle to improve throughputs. To accomplish this, both the SI and SO pins are utilized as inputs/outputs for the transfer of data bytes. With the Dual-Input Byte/Page Program command, the SO pin becomes an input along with the SI pin. Alternatively, with the Dual-Output Read Array command, the SI pin becomes an output along with the SO pin. For both Dual-I/O commands, the SO pin will be referred to as I/O_1 and the SI pin will be referred to as I/O_0.

The device also supports a Quad-Input Program mode and a Quad-Output Read mode in which the $\overline{WP}$ and $\overline{HOLD}$ pins become data pins for even higher throughputs. For the Quad-Input Byte/Page Program command and for the Quad-Output Read Array command, the $\overline{HOLD}$, $\overline{WP}$, SO, and SI pins are referred to as I/O_{3-0} where $\overline{HOLD}$ becomes I/O_3, $\overline{WP}$ becomes I/O_2, SO becomes I/O_1, and SI becomes I/O_0. The QE bit in the Configuration Register must be set in order for both Quad-I/O commands to be enabled and for the $\overline{HOLD}$ and $\overline{WP}$ pins to be converted to I/O data pins.

6. Commands and Addressing

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

Opcodes not supported by the AT25DQ161 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 AT25DQ161 memory array is 1FFFFFh, address bits A23-A21 are always ignored by the device.

Table 6-1. Command Listing

Command	Opcode		Clock Frequency	Address Bytes	Dummy Bytes	Data Bytes
Read Commands						
Read Array	1Bh	0001 1011	Up to 100MHz	3	2	1+
	0Bh	0000 1011	Up to 85MHz	3	1	1+
	03h	0000 0011	Up to 40MHz	3	0	1+
Dual-Output Read Array	3Bh	0011 1011	Up to 85MHz	3	1	1+
Quad-Output Read Array	6Bh	0110 1011	Up to 85MHz	3	1	1+
Program and Erase Commands						
Block Erase (4KB)	20h	0010 0000	Up to 100MHz	3	0	0
Block Erase (32KB)	52h	0101 0010	Up to 100MHz	3	0	0
Block Erase (64KB)	D8h	1101 1000	Up to 100MHz	3	0	0
Chip Erase	60h	0110 0000	Up to 100MHz	0	0	0
	C7h	1100 0111	Up to 100MHz	0	0	0
Byte/Page Program (1 to 256 bytes)	02h	0000 0010	Up to 100MHz	3	0	1+
Dual-Input Byte/Page Program (1 to 256 bytes)	A2h	1010 0010	Up to 100MHz	3	0	1+
Quad-Input Byte/Page Program (1 to 256 bytes)	32h	0011 0010	Up to 100MHz	3	0	1+
Program/Erase Suspend	B0h	1011 0000	Up to 100MHz	0	0	0
Program/Erase Resume	D0h	1101 0000	Up to 100MHz	0	0	0
Protection Commands						
Write Enable	06h	0000 0110	Up to 100MHz	0	0	0
Write Disable	04h	0000 0100	Up to 100MHz	0	0	0
Protect Sector	36h	0011 0110	Up to 100MHz	3	0	0
Unprotect Sector	39h	0011 1001	Up to 100MHz	3	0	0
Global Protect/Unprotect	Use Write Status Register Byte 1 Command					
Read Sector Protection Registers	3Ch	0011 1100	Up to 100MHz	3	0	1+
Security Commands						
Sector Lockdown	33h	0011 0011	Up to 100MHz	3	0	1
Freeze Sector Lockdown State	34h	0011 0100	Up to 100MHz	3	0	1
Read Sector Lockdown Registers	35h	0011 0101	Up to 100MHz	3	0	1+
Program OTP Security Register	9Bh	1001 1011	Up to 100MHz	3	0	1+
Read OTP Security Register	77h	0111 0111	Up to 100MHz	3	2	1+
Status and Configuration Register Commands						
Read Status Register	05h	0000 0101	Up to 100MHz	0	0	1+
Write Status Register Byte 1	01h	0000 0001	Up to 100MHz	0	0	1
Write Status Register Byte 2	31h	0011 0001	Up to 100MHz	0	0	1
Read Configuration Register	3Fh	0011 1111	Up to 100MHz	0	0	1+
Write Configuration Register	3Eh	0011 1110	Up to 100MHz	0	0	1
Miscellaneous Commands						
Reset	F0h	1111 0000	Up to 100MHz	0	0	1
Read Manufacturer and Device ID	9Fh	1001 1111	Up to 85MHz	0	0	1 to 5
Deep Power-Down	B9h	1011 1001	Up to 100MHz	0	0	0
Resume from Deep Power-Down	ABh	1010 1011	Up to 100MHz	0	0	0

7. Read Commands

7.1 Read Array

The Read Array command can be used to sequentially read a continuous stream of data from the device by simply providing the clock signal once the initial starting address has been specified. The device incorporates an internal address counter that automatically increments on every clock cycle.

Three opcodes (1Bh, 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}. The 1Bh opcode allows the highest read performance possible and can be used at any clock frequency up to the maximum specified by f_{MAX}; however, use of the 1Bh opcode at clock frequencies above f_{CLK} should be reserved to systems employing the RapidS protocol.

To perform the Read Array operation, the $\overline{CS}$ pin must first be asserted and the appropriate opcode (1Bh, 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, additional dummy bytes may need to be clocked into the device depending on which opcode is used for the Read Array operation. If the 1Bh opcode is used, then two dummy bytes must be clocked into the device after the three address bytes. If the 0Bh opcode is used, then a single dummy byte must be clocked in after the address bytes.

After the three address bytes (and the dummy bytes or byte if using opcodes 1Bh or 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 (1FFFFFh) 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 a high-impedance state. The $\overline{CS}$ pin can be deasserted at any time and does not require that a full byte of data be read.

Figure 7-1. Read Array – 1Bh Opcode

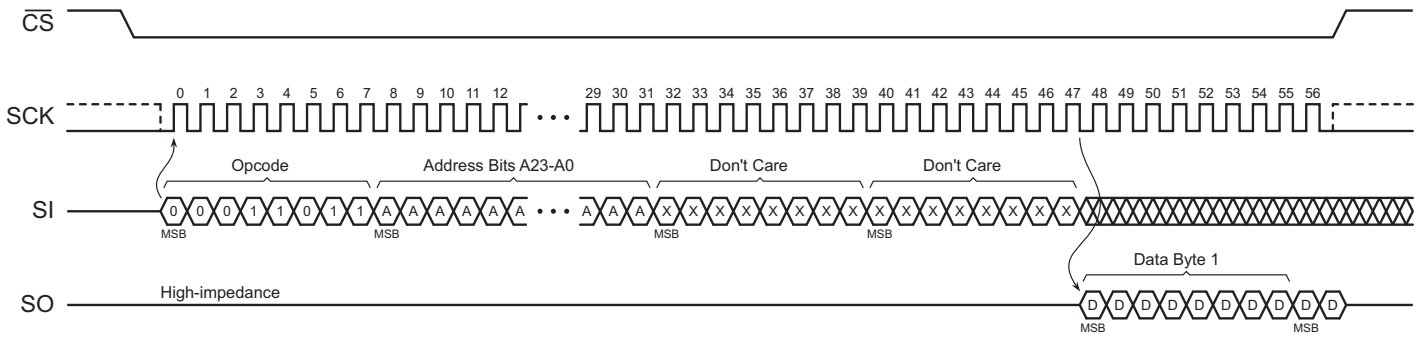


Figure 7-2. Read Array – 0Bh Opcode

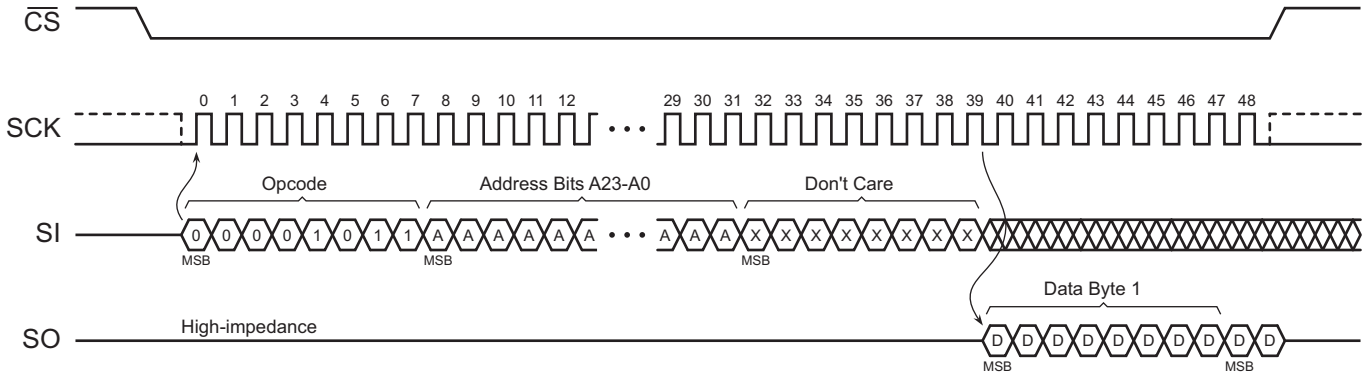
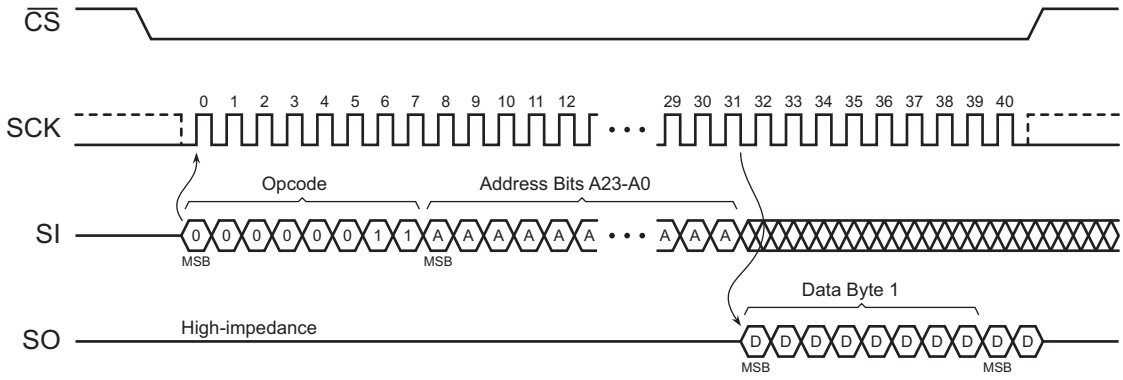


Figure 7-3. Read Array – 03h Opcode



7.2 Dual-Output Read Array

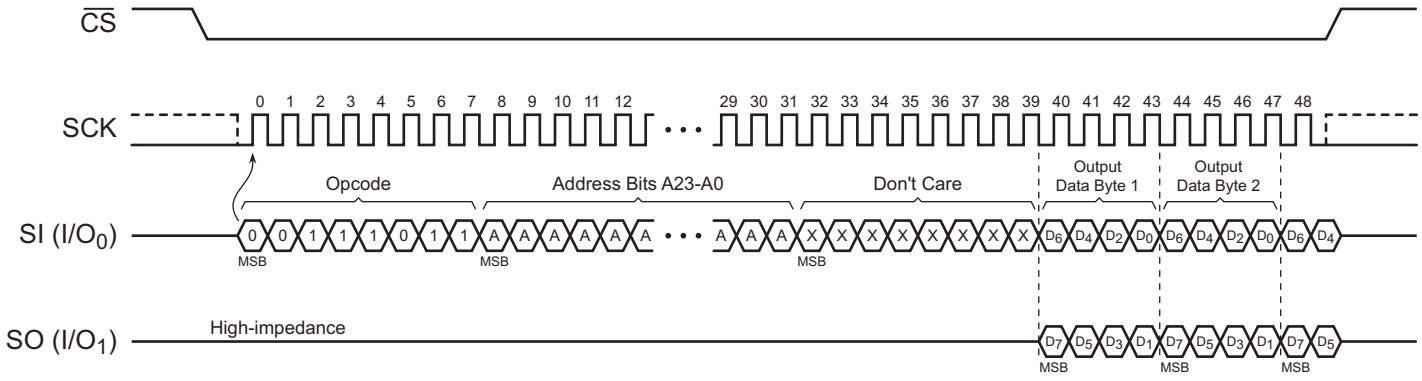
The Dual-Output Read Array command is similar to the standard Read Array command and can be used to sequentially read a continuous stream of data from the device by simply providing the clock signal once the initial starting address has been specified. Unlike the standard Read Array command, however,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

The Dual-Output Read Array command can be used at any clock frequency up to the maximum specified by f_{RDDO}. To perform the Dual-Output Read Array operation, the $\overline{CS}$ pin must first be asserted and the opcode of 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 starting address 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 I/O_1 and I/O_0 pins. The data is always output with the MSB of a byte first, and the MSB is always output on the I/O_1 pin. During the first clock cycle, bit 7 of the first data byte will be output on the I/O_1 pin while bit 6 of the same data byte will be output on the I/O_0 pin. During the next clock cycle, bits 5 and 4 of the first data byte will be output on the I/O_1 and I/O_0 pins, respectively. The sequence continues with each byte of data being output after every four clock cycles. When the last byte (1FFFFFh) 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 I/O_{1-0} 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.

Figure 7-4. Dual-Output Read Array



7.3 Quad-Output Read Array

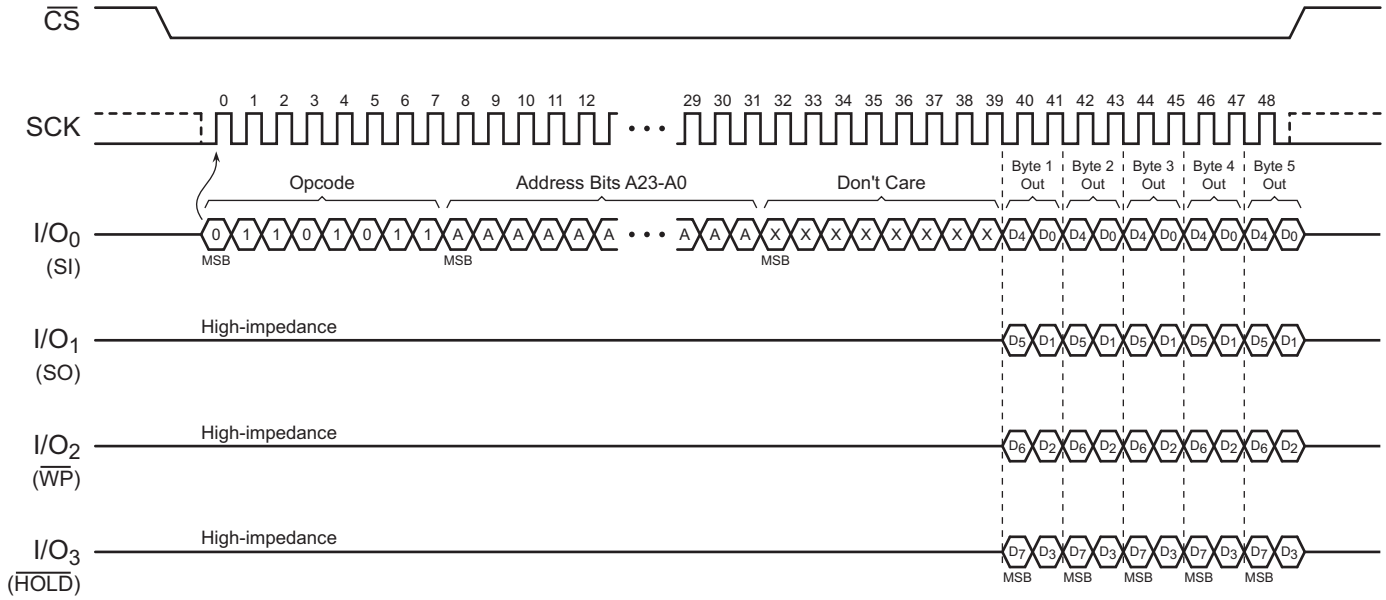
The Quad-Output Read Array command is similar to the Dual-Output Read Array command and can be used to sequentially read a continuous stream of data from the device by simply providing the clock signal once the initial starting address has been specified. Unlike the Dual-Output Read Array command, however,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

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 the opcode of 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 starting address 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 (1FFFFFh) 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 I/O_{3-0} 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.

Figure 7-5. Quad-Output Read Array



8. Program and Erase Commands

8.1 Byte/Page Program

The Byte/Page Program command allows anywhere from a single byte of data to 256 bytes of data to be programmed into previously erased memory locations. An erased memory location is one that has all eight bits set to the Logical 1 state (a byte value of FFh). Before a Byte/Page Program command can be started, the Write Enable command must have been previously issued to the device (see “Write Enable” on page 24) 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.

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 is 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 address specified by A23-A0 points to a memory location within a sector that is in the protected state (see “Protect Sector” on page 26) or locked down (see “Sector Lockdown” on page 32), then the Byte/Page Program command will not be executed, and the device will return to the idle state once the $\overline{CS}$ pin has been deasserted. The WEL bit in the Status Register will be reset back to the Logical 0 state if the program cycle aborts due to an incomplete address being sent, an incomplete byte of data being sent, the $\overline{CS}$ pin being deasserted on uneven byte boundaries, or because the memory location to be programmed is protected or locked down.

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

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

Figure 8-1. Byte Program

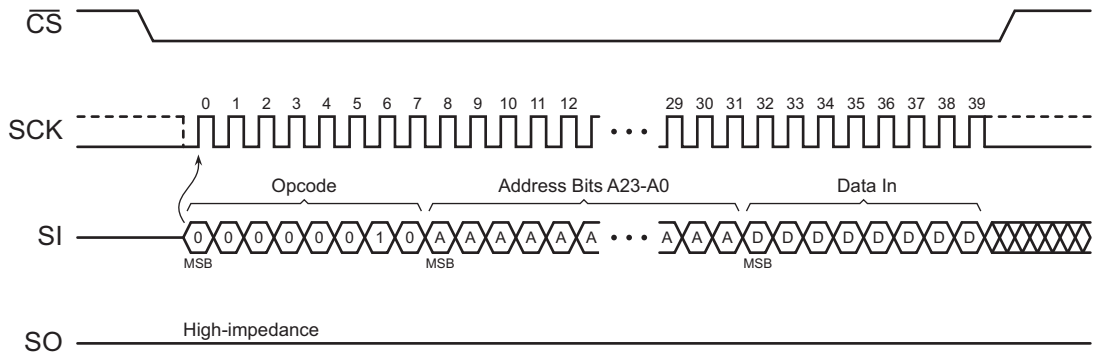
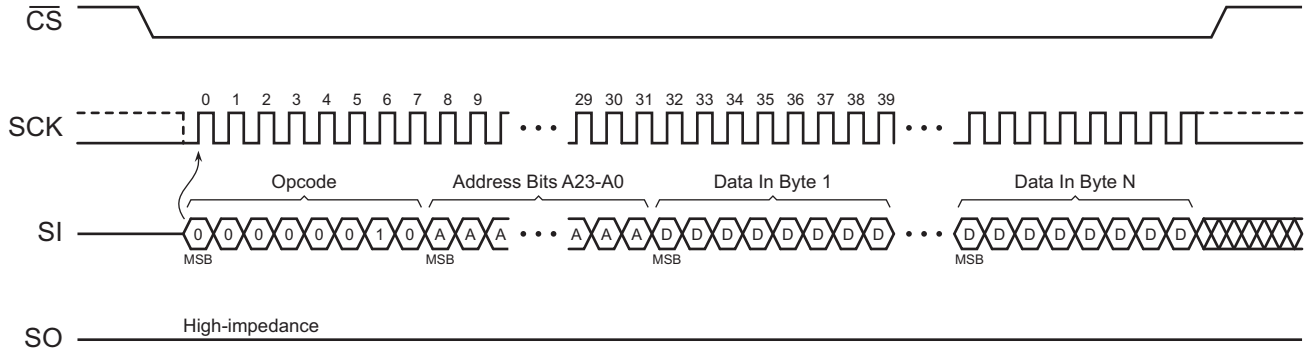


Figure 8-2. Page Program



8.2 Dual-Input Byte/Page Program

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

Before the Dual-Input Byte/Page Program command can be started, the Write Enable command must have been previously issued to the device (see "Write Enable" on page 24) to set the Write Enable Latch (WEL) bit of the Status Register to a Logical 1 state. To perform a Dual-Input Byte/Page Program command, an opcode of A2h 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 two bits at a time on both the I/O_1 and I/O_0 pins.

The data is always input with the MSB of a byte first, and the MSB is always input on the I/O_1 pin. During the first clock cycle, bit 7 of the first data byte would be input on the I/O_1 pin while bit 6 of the same data byte would be input on the I/O_0 pin. During the next clock cycle, bits 5 and 4 of the first data byte would be input on the I/O_1 and I/O_0 pins, respectively. The sequence would continue with each byte of data being input after every four clock cycles. Like the standard Byte/Page Program command, all data clocked into the device is 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.

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 is 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 address specified by A23-A0 points to a memory location within a sector that is in the protected state (see "Protect Sector" on page 26) or locked down (see "Sector Lockdown" on page 32), then the Byte/Page Program command will not be executed, and the device will return to the idle state once the $\overline{CS}$ pin has been deasserted. The WEL bit in the Status Register will be reset back to the Logical 0 state if the program cycle aborts due to an incomplete address being sent, an incomplete byte of data being sent, the $\overline{CS}$ pin being deasserted on uneven byte boundaries, or because the memory location to be programmed is protected or locked down.

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

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

Figure 8-3. Dual-Input Byte Program

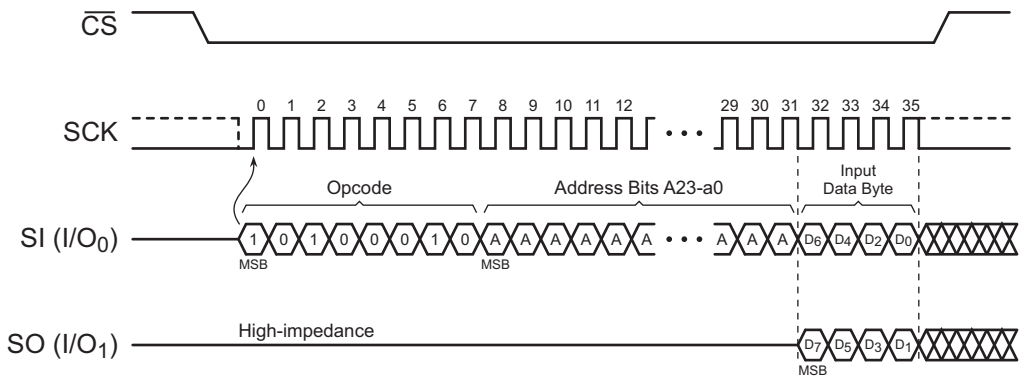
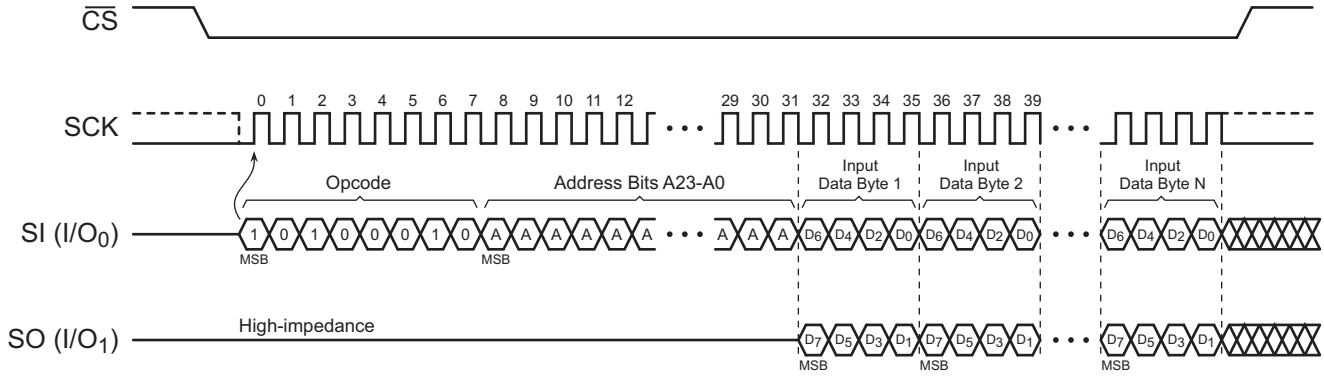


Figure 8-4. Dual-Input Page Program



8.3 Quad-Input Byte/Page Program

The Quad-Input Byte/Page Program command is similar to the Dual-Input Byte/Page Program command and can be used to program anywhere from a single byte of data up to 256 bytes of data into previously erased memory locations. Unlike the Dual-Input Byte/Page Program command, however, the Quad-Input Byte/Page Program command allows four bits of data to be clocked into the device on every clock cycle rather than two.

Before the Quad-Input Byte/Page Program command can be started, the Write Enable command must have been previously issued to the device (See "Write Enable" on page 24) to set the Write Enable Latch (WEL) bit of the Status Register to a Logical 1 state. To perform a Quad-Input Byte/Page Program command, an opcode of 32h 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 four bits at a time on the I/O_{3-0} pins.

The data is always input with the MSB of a byte first, and the MSB is always input on the I/O_3 pin. During the first clock cycle, bit 7 of the first data byte would be input on the I/O_3 pin while bits 6, 5, and 4 of the same data byte would be input 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 would be input on the I/O_3, I/O_2, I/O_1, and I/O_0 pins, respectively. The sequence would continue with each byte of data being input after every two clock cycles. Like the standard Byte/Page Program and Dual-Input Byte/Page Program commands, all data clocked into the device is stored in an internal buffer.

If the starting memory address denoted by A23-A0 does not fall on a 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.

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 is 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 byte boundaries (multiples of eight bits); otherwise, the device will abort the operation and no data will be programmed into the memory array. In addition, if the address specified by A23-A0 points to a memory location within a sector that is in the protected state (See "Protect Sector" on page 26) or locked down (See "Sector Lockdown" on page 32), then the Quad-Input Byte/Page Program command will not be executed, and the device will return to the idle state once the $\overline{CS}$ pin has been deasserted. The WEL bit in the Status Register will be reset back to the Logical 0 state if the program cycle aborts due to an incomplete address being sent, an incomplete byte of data being sent, the $\overline{CS}$ pin being deasserted on uneven byte boundaries, or because the memory location to be programmed is protected or locked down.

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

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

Figure 8-5. Quad-Input Byte Program

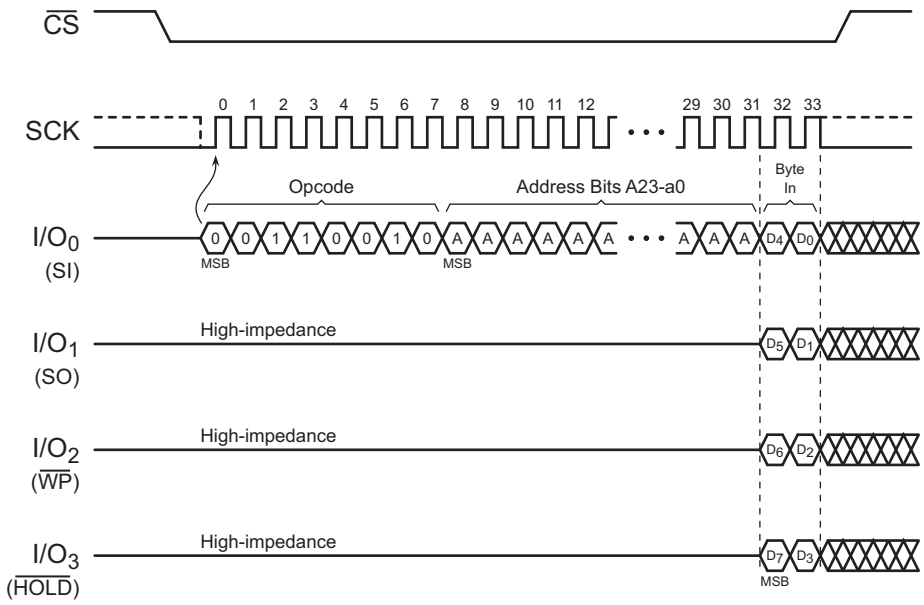
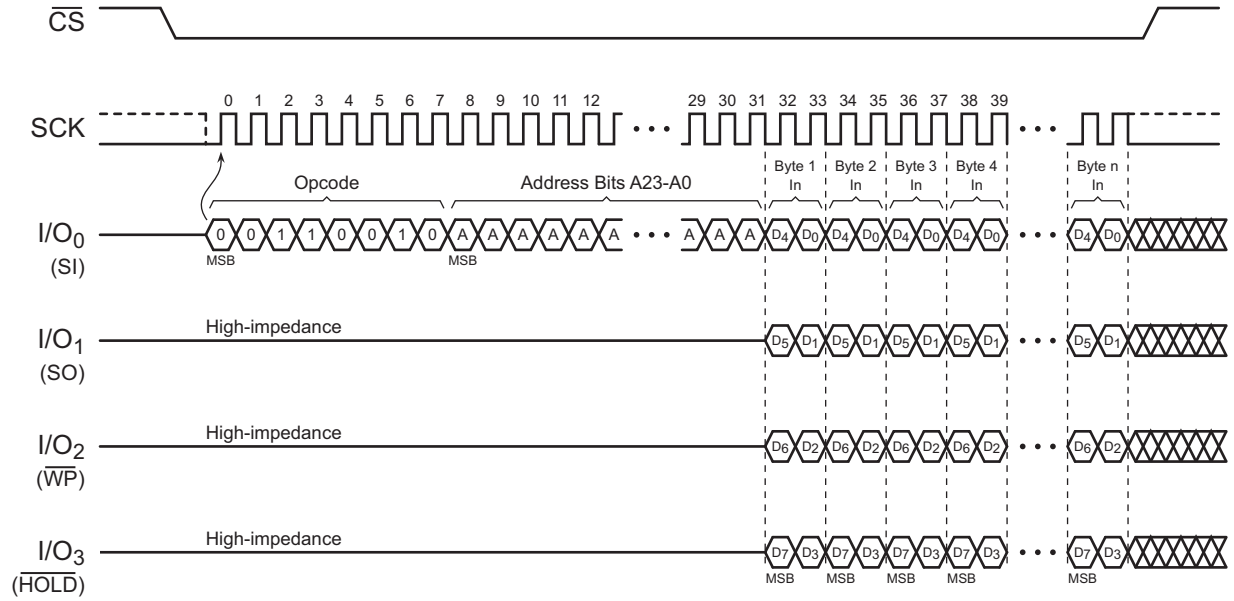


Figure 8-6. Quad-Input Page Program



8.4 Block Erase

A block of 4, 32, or 64KB 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 4KB erase, an opcode of 52h is used for a 32KB erase, and an opcode of D8h is used for a 64KB erase. Before a Block Erase command can be started, the Write Enable command must have been previously issued to the device to set the WEL bit of the Status Register to a Logical 1 state.

To perform a Block Erase, the $\overline{CS}$ pin must first be asserted and the appropriate opcode (20h, 52h, or D8h) must be clocked into the device. After the opcode has been clocked in, the three address bytes specifying an address within the 4, 32, or 64KB 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 4KB erase, address bits A11-A0 will be ignored by the device and their values can be either a Logical 1 or 0. For a 32KB erase, address bits A14-A0 will be ignored, and for a 64KB erase, address bits A15-A0 will be ignored by the device. Despite the lower order address bits not being decoded by the device, the complete three address bytes must still be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on an even byte boundary (multiples of eight bits); otherwise, the device will abort the operation and no erase operation will be performed.

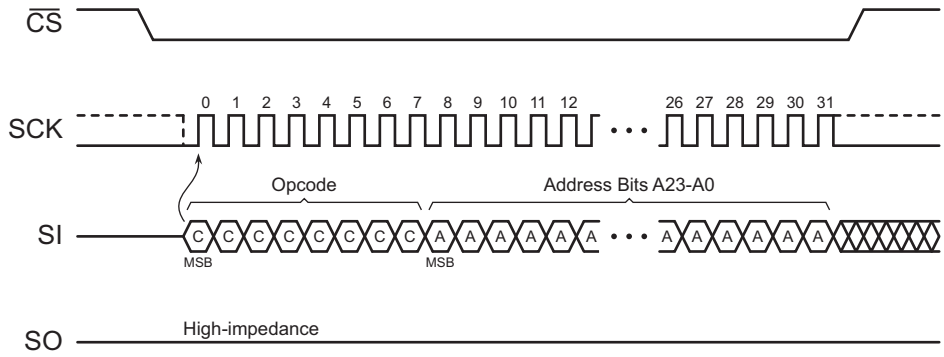
If the address specified by A23-A0 points to a memory location within a sector that is in the protected or locked down state, then the Block Erase command will not be executed, and the device will return to the idle state once the $\overline{CS}$ pin has been deasserted.

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 or locked down.

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.

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

Figure 8-7. Block Erase



8.5 Chip Erase

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

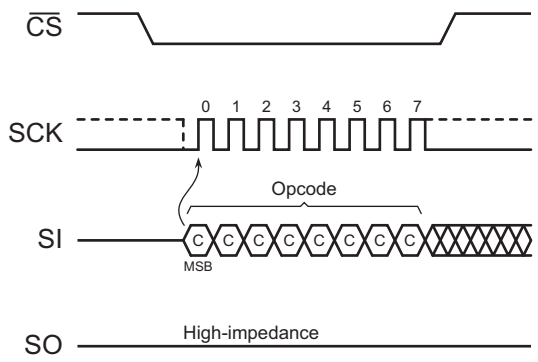
Two opcodes, 60h and C7h, can be used for the Chip Erase command. There is no difference in device functionality when utilizing the two opcodes, so they can be used interchangeably. To perform a Chip Erase, one of the two opcodes (60h or C7h) must be clocked into the device. Since the entire memory array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

The complete opcode must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on an even byte boundary (multiples of eight bits); otherwise, no erase will be performed. In addition, if any sector of the memory array is in the protected or locked down 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 a sector is in the protected or locked down state.

While the device is executing a successful erase cycle, the Status Register can be read and will indicate that the device is busy. For faster throughput, it is recommended that the Status Register be polled rather than waiting the t_{CHPE} time to determine if the device has finished erasing. At some point before the erase cycle completes, the WEL bit in the Status Register will be reset back to the Logical 0 state.

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

Figure 8-8. Chip Erase



8.6 Program/Erase Suspend

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 a particular 64KB sector of the Flash memory array to be suspended so that other device operations can be performed.

Example: By suspending an erase operation to a particular sector, the system can perform functions such as a program or read operation within another 64KB sector in the device. Other device operations, such as a Read Status Register, can also be performed while a program or erase operation is suspended. Table 8-1 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 B0h 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_{SUSP}. The Program Suspend (PS) bit or the Erase Suspend (ES) bit in the Status Register will then be set to the Logical 1 state to indicate that the program or erase operation has been suspended. In addition, the 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 an even byte boundary (multiples of eight bits); otherwise, no suspend operation will be performed.

Read operations are not allowed to a 64KB sector that has had its program or erase operation suspended. If a read is attempted to a suspended sector, then the device will output undefined data. Therefore, when performing a Read Array operation to an unsuspended sector and the device's internal address counter increments and crosses the sector boundary to a suspended sector, the device will then start outputting undefined data continuously until the address counter increments and crosses a sector boundary to an unsuspended sector.

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

During an Erase Suspend, a program operation to a different 64KB sector can be started and subsequently suspended. This results in a simultaneous Erase Suspend/Program Suspend condition and will be indicated by the states of both the ES and PS bits in the Status Register being set to the Logical 1 state.

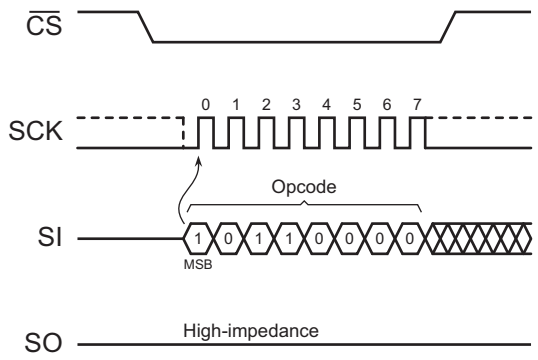
If a Reset operation (see "Reset" on page 48) is performed while a sector is erase suspended, the suspend operation will abort and the contents of the block in the suspended sector will be left in an undefined state. However, if a Reset is performed while a sector is program suspended, the suspend operation will abort but only the contents of the page that was being programmed and subsequently suspended will be undefined. The remaining pages in the 64KB sector will retain their previous contents.

If an attempt is made to perform an operation that is not allowed during a program or erase suspend, such as a Protect Sector 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, as well as the SPRL (Sector Protection Registers Locked), and SLE (Sector Lockdown Enabled) bits, will not be affected.

Table 8-1. Operations Allowed and Not Allowed During a Program or Erase Suspend

Command	Operation During Program Suspend	Operation During Erase Suspend
Read Commands		
Read Array (All Opcodes)	Allowed	Allowed
Program and Erase Commands		
Block Erase	Not Allowed	Not Allowed
Chip Erase	Not Allowed	Not Allowed
Byte/Page Program (All Opcodes)	Not Allowed	Allowed
Program/Erase Suspend	Not Allowed	Allowed
Program/Erase Resume	Allowed	Allowed
Protection Commands		
Write Enable	Not Allowed	Allowed
Write Disable	Not Allowed	Allowed
Protect Sector	Not Allowed	Not Allowed
Unprotect Sector	Not Allowed	Not Allowed
Global Protect/Unprotect	Not Allowed	Not Allowed
Read Sector Protection Registers	Allowed	Allowed
Security Commands		
Sector Lockdown	Not Allowed	Not Allowed
Freeze Sector Lockdown State	Not Allowed	Not Allowed
Read Sector Lockdown Registers	Allowed	Allowed
Program OTP Security Register	Not Allowed	Not Allowed
Read OTP Security Register	Allowed	Allowed
Status and Configuration Register Commands		
Read Status Register	Allowed	Allowed
Write Status Register (All Opcodes)	Not Allowed	Not Allowed
Read Configuration Register	Allowed	Allowed
Write Configuration Register	Not Allowed	Not Allowed
Miscellaneous Commands		
Reset	Allowed	Allowed
Read Manufacturer and Device ID	Allowed	Allowed
Deep Power-Down	Not Allowed	Not Allowed
Resume from Deep Power-Down	Not Allowed	Not Allowed

Figure 8-9. Program/Erase Suspend



8.7 Program/Erase Resume

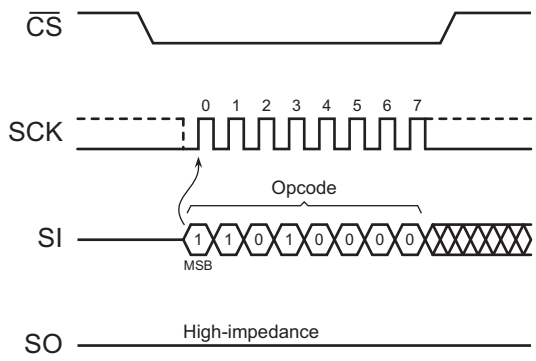
The Program/Erase Resume command allows a suspended program or erase operation to be resumed and continue programming a Flash page or erasing a Flash memory block where it left off. As with the Program/Erase Suspend command, the Write Enable command does not need to be issued prior to the Program/Erase 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 a Program/Erase Resume, the $\overline{CS}$ pin must first be asserted and the opcode of D0h 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 suspended will be resumed within a time of t_{RES}. The PS bit or the ES bit in the Status Register will then be reset back to the Logical 0 state to indicate that 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 $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on an even byte boundary (multiples of eight bits); otherwise, no resume operation will be performed.

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 appropriate PS or ES bit in the Status Register to determine if the previously suspended program or erase operation has resumed.

Figure 8-10. Program/Erase Resume



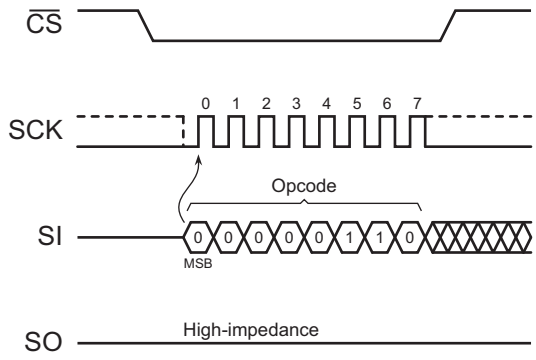
9. Protection Commands and Features

9.1 Write Enable

The Write Enable command is used to set the Write Enable Latch (WEL) bit in the Status Register to a Logical 1 state. The WEL bit must be set before a Byte/Page Program, erase, Protect Sector, Unprotect Sector, Sector Lockdown, Freeze Sector Lockdown State, Program OTP Security Register, Write Status Register, or Write Configuration Register command can be executed. This makes the issuance of these commands a two step process, thereby reducing the chances of a command being accidentally or erroneously executed. If the WEL bit in the Status Register is not set prior to the issuance of one of these commands, then the command will not be executed.

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

Figure 9-1. Write Enable

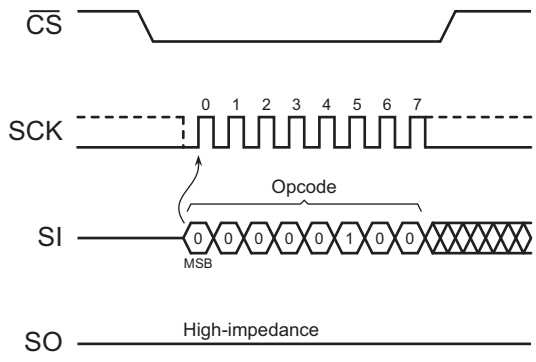


9.2 Write Disable

The Write Disable command is used to reset the Write Enable Latch (WEL) bit in the Status Register to the Logical 0 state. With the WEL bit reset, all Byte/Page Program, erase, Protect Sector, Unprotect Sector, Sector Lockdown, Freeze Sector Lockdown State, Program OTP Security Register, Write Status Register, and Write Configuration Register commands will not be executed. Other conditions can also cause the WEL bit to be reset; for more details, refer to the WEL bit section of the Status Register description.

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

Figure 9-2. Write Disable



9.3 Protect Sector

Every physical 64KB sector of the device has a corresponding single-bit Sector Protection Register that is used to control the software protection of a sector. Upon device power-up, each Sector Protection Register will default to the Logical 1 state indicating that all sectors are protected and cannot be programmed or erased.

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

Table 9-1. Sector Protection Register Values

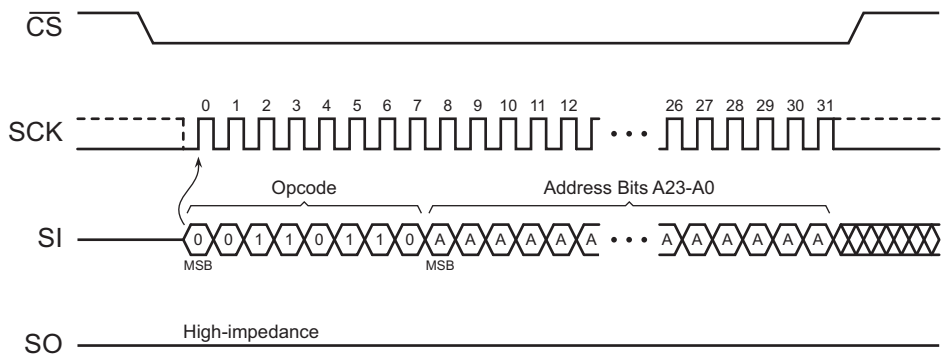
Value	Sector Protection Status
0	Sector is unprotected and can be programmed and erased.
1	Sector is protected and cannot be programmed or erased. This is the default state.

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

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

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

Figure 9-3. Protect Sector



9.4 Unprotect Sector

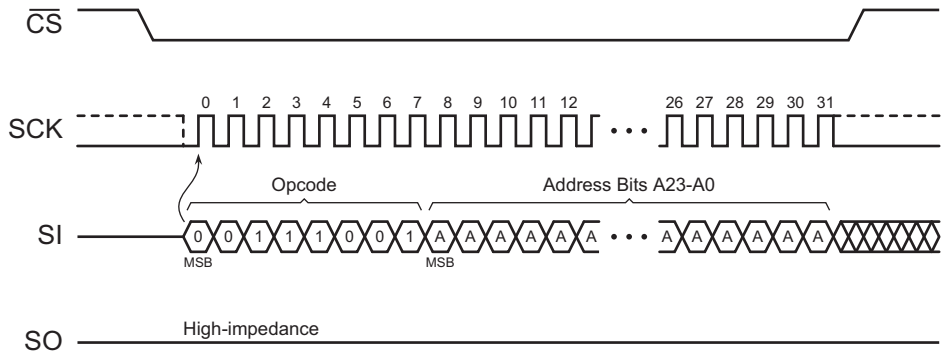
Issuing the Unprotect Sector command to a particular sector address will reset the corresponding Sector Protection Register to the Logical 0 state (see Table 9-1, “Sector Protection Register Values”). Every physical sector of the device has a corresponding single-bit Sector Protection Register that is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

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

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

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

Figure 9-4. Unprotect Sector



9.5 Global Protect/Unprotect

The Global Protect and Global Unprotect features can work in conjunction with the Protect Sector and Unprotect Sector functions.

Example: A system can globally protect the entire memory array and then use the Unprotect Sector command to individually unprotect certain sectors and individually reprotect them later by using the Protect Sector command. Likewise, a system can globally unprotect the entire memory array and then individually protect certain sectors as needed.

Performing a Global Protect or Global Unprotect is accomplished by writing a certain combination of data to the Status Register using the Write Status Register Byte 1 command (see "Write Status Register Byte 1" on page 43 for command execution details). The Write Status Register command is also used to modify the SPRL (Sector Protection Registers Locked) bit to control hardware and software locking.

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

Sectors that have been erase or program suspended must remain in the unprotected state. If a Global Protect operation is attempted while a sector is erase or program suspended, the protection operation will abort, the protection states of all sectors in the Flash memory array will not change, and WEL bit in the Status Register will be reset back to a Logical 0.

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

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

When writing to the first byte of the Status Register, bits 5, 4, 3, and 2 will not actually be modified but will be decoded by the device for the purposes of the Global Protect and Global Unprotect functions. Only bit 7, the SPRL bit, will actually be modified. Therefore, when reading the first byte of the Status Register, bits 5, 4, 3, and 2 will not reflect the values written to them but will instead indicate the status of the $\overline{WP}$ pin and the sector protection status. Please refer to "Read Status Register" on page 39 and Table 11-1, "Status Register Format – Byte 1," on page 39 for details on the Status Register format and what values can be read for bits 5, 4, 3, and 2.

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

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

9.6 Read Sector Protection Registers

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

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

At clock frequencies above f_{CLK}, the first byte of data output will not be valid. Therefore, if operating at clock frequencies above f_{CLK}, at least two bytes of data must be clocked out from the device in order to determine the correct status of the appropriate Sector Protection Register.

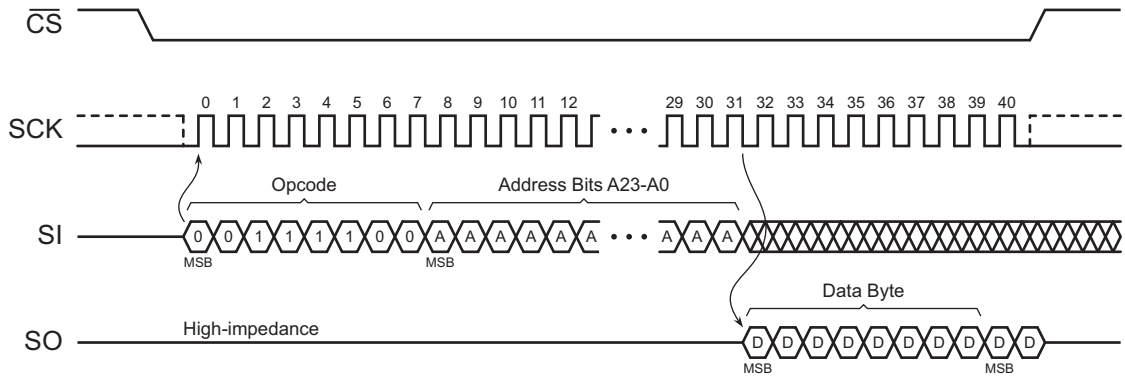
Table 9-3. Read Sector Protection Register – Output Data

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

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

In addition to reading the individual Sector Protection Registers, the Software Protection Status (SWP) bits in the Status Register can be read to determine if all, some, or none of the sectors are software protected (refer to “Read Status Register” on page 39 for more details).

Figure 9-5. Read Sector Protection Register



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

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

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

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

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

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

Table 9-4. Sector Protection Register States

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

Note: 1. “n” represents a sector number

Table 9-5. Hardware and Software Locking

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

10. Security Commands

10.1 Sector Lockdown

Certain applications require that portions of the Flash memory array be permanently protected against malicious attempts at altering program code, data modules, security information or encryption/decryption algorithms, keys, and routines. To address these applications, the device incorporates a sector lockdown mechanism that allows any combination of individual 64KB sectors to be permanently locked so that they become read only. Once a sector is locked down, it can never be erased or programmed again, and it can never be unlocked from the locked down state.

Each 64KB physical sector has a corresponding single-bit Sector Lockdown Register that is used to control the lockdown status of that sector. These registers are nonvolatile and will retain their state even after a device power-cycle or reset operation. The following table outlines the two states of the Sector Lockdown Registers.

Table 10-1. Sector Lockdown Register Values

Value	Sector Lockdown Status
0	Sector is not locked down and can be programmed and erased. This is the default state.
1	Sector is permanently locked down and can never be programmed or erased again.

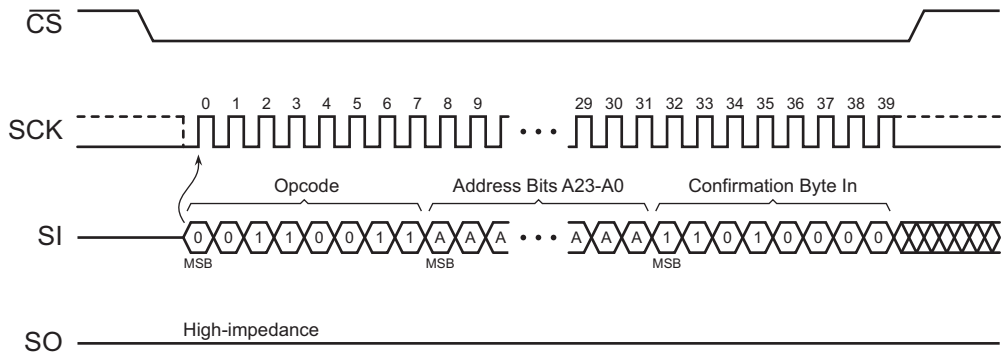
Issuing the Sector Lockdown command to a particular sector address will set the corresponding Sector Lockdown Register to the Logical 1 state. Each Sector Lockdown Register can only be set once; therefore, once set to the Logical 1 state, a Sector Lockdown Register cannot be reset back to the Logical 0 state.

Before the Sector Lockdown 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. In addition, the Sector Lockdown Enabled (SLE) bit in the Status Register must have also been previously set to the Logical 1 state by using the Write Status Register Byte 2 command (see “Write Status Register Byte 2” on page 44). To issue the Sector Lockdown command, the $\overline{CS}$ pin must first be asserted and the opcode of 33h must be clocked into the device followed by three address bytes designating any address within the 64KB sector to be locked down. After the three address bytes have been clocked in, a confirmation byte of D0h must also be clocked in immediately following the three address bytes. Any additional data clocked into the device after the first byte of data will be ignored. When the $\overline{CS}$ pin is deasserted, the Sector Lockdown Register corresponding to the sector addressed by A23-A0 will be set to the Logical 1 state, and the sector itself will then be permanently locked down from program and erase operations within a time of t_{LOCK}. In addition, the WEL bit in the Status Register will be reset back to the Logical 0 state.

The complete three address bytes and the correct confirmation byte value of D0h must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on an even byte boundary (multiples of eight bits); otherwise, the device will abort the operation. When the device aborts the Sector Lockdown operation, the state of the corresponding Sector Lockdown Register as well as the SLE bit in the Status Register will be unchanged; however, the WEL bit in the Status Register will be reset to a Logical 0.

As a safeguard against accidental or erroneous locking down of sectors, the Sector Lockdown command can be enabled and disabled as needed by using the SLE bit in the Status Register. In addition, the current sector lockdown state can be frozen so that no further modifications to the Sector Lockdown Registers can be made (see “Freeze Sector Lockdown State” below). If the Sector Lockdown command is disabled or if the sector lockdown state is frozen, then any attempts to issue the Sector Lockdown command will be ignored, and the device will reset the WEL bit in the Status Register back to a Logical 0 and return to the idle state once the $\overline{CS}$ pin has been deasserted.

Figure 10-1. Sector Lockdown



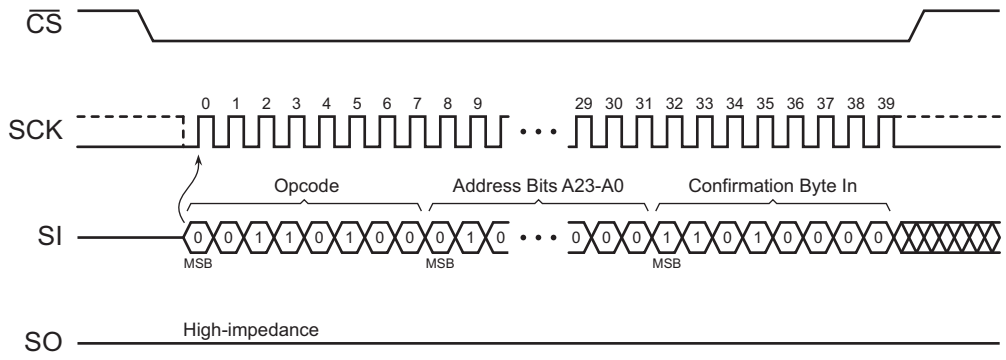
10.2 Freeze Sector Lockdown State

The current sector lockdown state can be permanently frozen so that no further modifications to the Sector Lockdown Registers can be made; therefore, the Sector Lockdown command will be permanently disabled, and no additional sectors can be locked down aside from those already locked down. Any attempts to issue the Sector Lockdown command after the sector lockdown state has been frozen will be ignored.

Before the Freeze Sector Lockdown State 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. In addition, the Sector Lockdown Enabled (SLE) bit in the Status Register must have also been previously set to the Logical 1 state. To issue the Freeze Sector Lockdown State command, the $\overline{CS}$ pin must first be asserted and the opcode of 34h must be clocked into the device followed by three command specific address bytes of 55AA40h. After the three address bytes have been clocked in, a confirmation byte of D0h must be clocked in immediately following the three address bytes. Any additional data clocked into the device will be ignored. When the $\overline{CS}$ pin is deasserted, the current sector lockdown state will be permanently frozen within a time of t_{LOCK}. In addition, the WEL bit in the Status Register will be reset back to the Logical 0 state, and the SLE bit will be permanently reset to a Logical 0 to indicate that the Sector Lockdown command is permanently disabled.

The complete and correct three address bytes and the confirmation byte must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on an even byte boundary (multiples of eight bits); otherwise, the device will abort the operation. When the device aborts the Freeze Sector Lockdown State operation, the WEL bit in the Status Register will be reset to a Logical 0; however, the state of the SLE bit will be unchanged.

Figure 10-2. Freeze Sector Lockdown State



10.3 Read Sector Lockdown Registers

The Sector Lockdown Registers can be read to determine the current lockdown status of each physical 64KB sector. To read the Sector Lockdown Register for a particular 64KB sector, the $\overline{CS}$ pin must first be asserted and the opcode of 35h must be clocked in. Once the opcode has been clocked in, three address bytes designating any address within the 64KB sector must be clocked in. After the address bytes have been clocked in, data will be output on the SO pin during every subsequent clock cycle. The data being output will be a repeating byte of either FFh or 00h to denote the value of the appropriate Sector Lockdown Register.

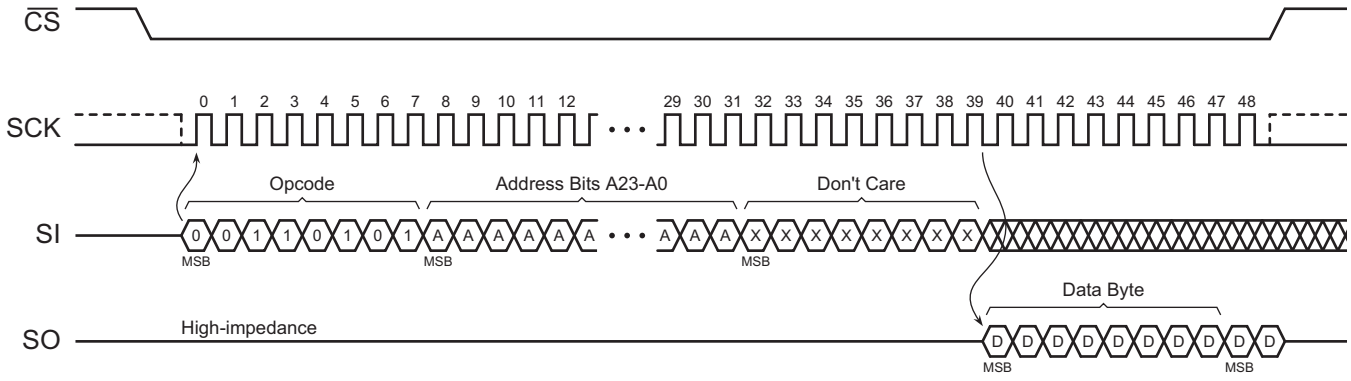
At clock frequencies above f_{CLK}, the first byte of data output will not be valid. Therefore, if operating at clock frequencies above f_{CLK}, at least two bytes of data must be clocked out from the device in order to determine the correct status of the appropriate Sector Lockdown Register.

Table 10-2. Read Sector Lockdown Register – Output Data

Output Data	Sector Lockdown Register Value
00h	Sector Lockdown Register value is 0 (sector is not locked down).
FFh	Sector Lockdown Register value is 1 (sector is permanently locked down).

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

Figure 10-3. Read Sector Lockdown Register



10.4 Program OTP Security Register

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

Table 10-3. OTP Security Register

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

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

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

If the starting memory address denoted by A23-A0 does not start at the beginning of the OTP Security Register memory space (A5-A0 are not all 0), then special circumstances regarding which OTP Security Register locations to be programmed will apply. In this situation, any data that is sent to the device that goes beyond the end of the 64-byte user-programmable space will wrap around back to the beginning of the OTP Security Register.

Example: If the starting address denoted by A23-A0 is 00003Eh, and three bytes of data are sent to the device, then the first two bytes of data will be programmed at OTP Security Register addresses 00003Eh and 00003Fh while the last byte of data will be programmed at address 000000h. The remaining bytes in the OTP Security Register (addresses 000001h through 00003Dh) will not be programmed and will remain in the erased state (FFh). In addition, if more than 64 bytes of data is sent to the device, then only the last 64 bytes sent will be latched into the internal buffer.

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

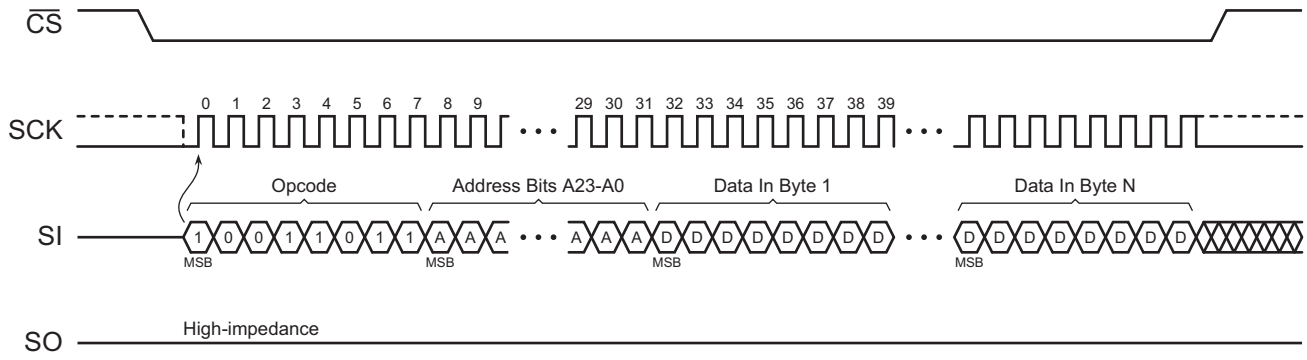
The three address bytes and at least one complete byte of data must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on even byte boundaries (multiples of eight bits); otherwise, the device will abort the operation and the user-programmable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

While the device is programming the OTP Security Register, the Status Register can be read and will indicate that the device is busy. For faster throughput, it is recommended that the Status Register be polled rather than waiting the t_{OTPP} time to determine if the data bytes have finished programming. At some point before the OTP Security Register programming completes, the WEL bit in the Status Register will be reset back to the Logical 0 state.

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

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

Figure 10-4. Program OTP Security Register



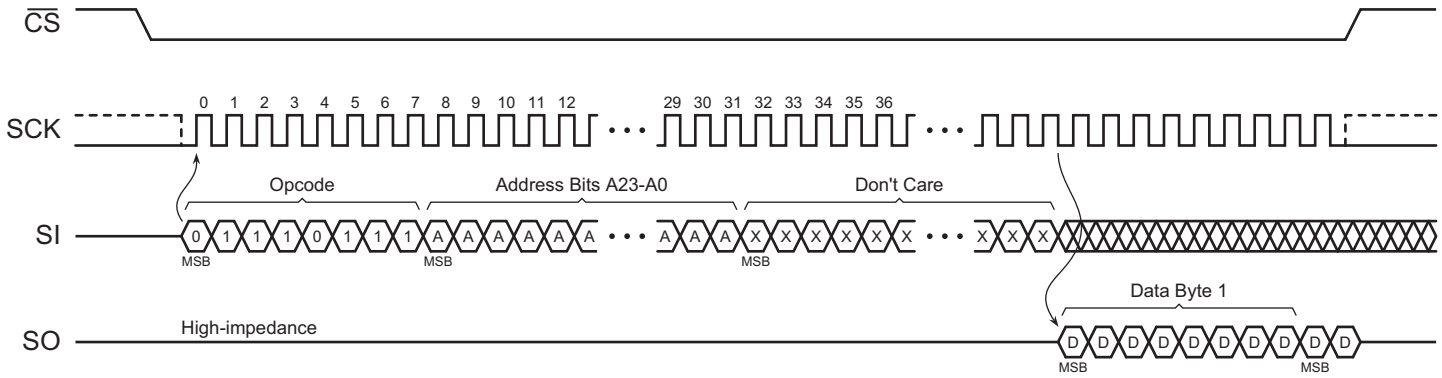
10.5 Read OTP Security Register

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

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

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

Figure 10-5. Read OTP Security Register



11. Status Register Commands

11.1 Read Status Register

The two-byte Status Register can be read to determine the device's ready/busy status, as well as the status of many other functions such as Hardware Locking and Software Protection. The Status Register can be read at any time, including during an internally self-timed program or erase operation.

To read the Status Register, the $\overline{CS}$ pin must first be asserted and the opcode of 05h must be clocked into the device. After the opcode has been clocked in, the device will begin outputting Status Register data on the SO pin during every subsequent clock cycle. After the second byte of the Status Register has been clocked out, the sequence will repeat itself starting again with the first byte of the Status Register 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. The RDY/BSY status is available for both bytes of the Status Register and is updated for each byte.

At clock frequencies above f_{CLK}, the first two bytes of data output from the Status Register will not be valid. Therefore, if operating at clock frequencies above f_{CLK}, at least four bytes of data must be clocked out from the device in order to read the correct values of both bytes of the Status Register.

Deasserting the $\overline{CS}$ pin will terminate the Read Status Register operation and put the SO pin into a high-impedance state. The $\overline{CS}$ pin can be deasserted at any time and does not require that a full byte of data be read.

Table 11-1. Status Register Format – Byte 1

Bit(1)	Name		Type(2)	Description	
7	SPRL	Sector Protection Registers Locked	R/W	0	Sector Protection Registers are unlocked (default)
				1	Sector Protection Registers are locked
6	RES	Reserved for future use	R	0	Reserved for future use
5	EPE	Erase/Program Error	R	0	Erase or program operation was successful
				1	Erase or program error detected
4	WPP	Write Protect ($\overline{WP}$) Pin Status	R	0	$\overline{WP}$ is asserted
				1	$\overline{WP}$ is deasserted
3:2	SWP	Software Protection Status	R	00	All sectors are software unprotected (all Sector Protection Registers are 0)
				01	Some sectors are software protected. Read individual Sector Protection Registers to determine which sectors are protected.
				10	Reserved for future use
				11	All sectors are software protected (all Sector Protection Registers are 1 – default)
1	WEL	Write Enable Latch Status	R	0	Device is not write enabled (default)
				1	Device is write enabled
0	RDY/BSY	Ready/Busy Status	R	0	Device is ready
				1	Device is busy with an internal operation

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

Table 11-2. Status Register Format – Byte 2

Bit[1]	Name		Type[2]		Description
7	RES	Reserved for future use	R	0	Reserved for future use
6	RES	Reserved for future use	R	0	Reserved for future use
5	RES	Reserved for future use	R	0	Reserved for future use
4	RSTE	Reset Enabled	R/W	0	Reset command is disabled (default)
				1	Reset command is enabled
3	SLE	Sector Lockdown Enabled	R/W	0	Sector Lockdown and Freeze Sector Lockdown State commands are disabled (default)
				1	Sector Lockdown and Freeze Sector Lockdown State commands are enabled
2	PS	Program Suspend Status	R	0	No sectors are program suspended (default)
				1	A sector is program suspended
1	ES	Erase Suspend Status	R	0	No sectors are erase suspended (default)
				1	A sector is erase suspended
0	RDY/BSY	Ready/Busy Status	R	0	Device is ready
				1	Device is busy with an internal operation

Notes: 1. Only bits 4 and 3 of Status Register Byte 2 will be modified when using the Write Status Register Byte 2 command.

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

11.1.1 SPRL Bit

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

When the SPRL bit is in the Logical 0 state, all Sector Protection Registers are unlocked and can be modified (the Protect Sector and Unprotect Sector commands, as well as the Global Protect and Global Unprotect features, will be processed as normal). The SPRL bit defaults to the Logical 0 state after device power-up. The Reset command has no effect on the SPRL bit.

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

The SPRL bit is the only bit of Status Register Byte 1 that can be user modified via the Write Status Register Byte 1 command.

11.1.2 EPE Bit

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

11.1.3 WPP Bit

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

11.1.4 SWP Bits

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

11.1.5 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, Protect Sector, Unprotect Sector, Sector Lockdown, Freeze Sector Lockdown State, Program OTP Security Register, Write Status Register, or Write Configuration 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
- Write Configuration Register operation completes successfully or aborts
- Protect Sector operation completes successfully or aborts
- Unprotect Sector operation completes successfully or aborts
- Sector Lockdown operation completes successfully or aborts
- Freeze Sector Lockdown State operation completes successfully or aborts
- Program OTP 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
- Hold condition aborts

If the WEL bit is in the Logical 1 state, it will not be reset to a Logical 0 if an operation aborts due to an incomplete or unrecognized opcode being clocked into the device before the $\overline{CS}$ pin is deasserted. In order for the WEL bit to be reset when an operation aborts prematurely, the entire opcode for a Byte/Page Program, erase, Protect Sector, Unprotect Sector, Sector Lockdown, Freeze Sector Lockdown State, Program OTP Security Register, Write Status Register, or Write Configuration Register command must have been clocked into the device.

11.1.6 RSTE Bit

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

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

11.1.7 SLE Bit

The SLE bit is used to enable and disable the Sector Lockdown and Freeze Sector Lockdown State commands. When the SLE bit is in the Logical 0 state (the default state after power-up), the Sector Lockdown and Freeze Sector Lockdown commands are disabled. If the Sector Lockdown and Freeze Sector Lockdown commands are disabled, then any attempts to issue the commands will be ignored. This provides a safeguard for these commands against accidental or erroneous execution. When the SLE bit is in the Logical 1 state, the Sector Lockdown and Freeze Sector Lockdown State commands are enabled.

Unlike the WEL bit, the SLE bit does not automatically reset after certain device operations. Therefore, once set, the SLE bit will remain in the Logical 1 state until it is modified using the Write Status Register Byte 2 command or until the device has been power cycled. The Reset command has no effect on the SLE bit.

If the Freeze Sector Lockdown State command has been issued, then the SLE bit will be permanently reset in the Logical 0 state to indicate that the Sector Lockdown command has been disabled.

11.1.8 PS Bit

The PS bit indicates whether or not a sector is in the Program Suspend state.

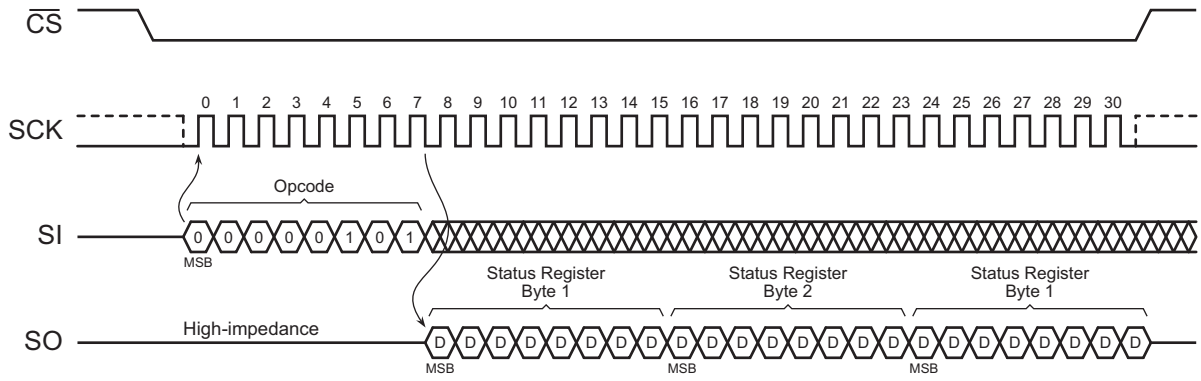
11.1.9 ES Bit

The ES bit indicates whether or not a sector is in the Erase Suspend state.

11.1.10 RDY/BSY Bit

The 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 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 RDY/BSY bit changes from a Logical 1 to a Logical 0.

Figure 11-1. Read Status Register



11.2 Write Status Register Byte 1

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

To issue the Write Status Register Byte 1 command, the $\overline{CS}$ pin must first be asserted and the opcode of 01h must be clocked into the device followed by one byte of data. The one byte of data consists of the SPRL bit value, a don't care bit, four data bits to denote whether a Global Protect or Unprotect should be performed, and two additional don't care bits (see Table 11-3). Any additional data bytes that are sent to the device will be ignored. When the $\overline{CS}$ pin is deasserted, the SPRL bit in the Status Register will be modified, and the WEL bit in the Status Register will be reset back to a Logical 0. The values of bits 5, 4, 3, and 2 and the state of the SPRL bit before the Write Status Register Byte 1 command was executed (the prior state of the SPRL bit) will determine whether or not a Global Protect or Global Unprotect will be performed. Please refer to "Global Protect/Unprotect" on page 28 for more details.

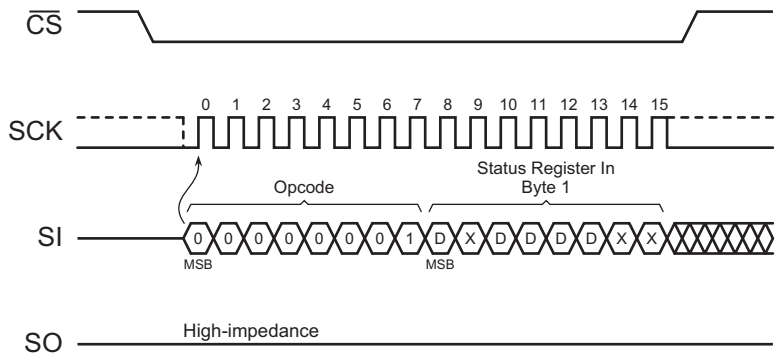
The complete one byte of data must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on even byte boundaries (multiples of eight bits); otherwise, the device will abort the operation, the state of the SPRL bit will not change, no potential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

If the $\overline{WP}$ pin is asserted, then the SPRL bit can only be set to a Logical 1. If an attempt is made to reset the SPRL bit to a Logical 0 while the $\overline{WP}$ pin is asserted, then the Write Status Register Byte 1 command will be ignored, and the WEL bit in the Status Register will be reset back to the Logical 0 state. In order to reset the SPRL bit to a Logical 0, the $\overline{WP}$ pin must be deasserted.

Table 11-3. Write Status Register Byte 1 Format

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

Figure 11-2. Write Status Register Byte 1



11.3 Write Status Register Byte 2

The Write Status Register Byte 2 command is used to modify the RSTE and SLE bits of the Status Register. Using the Write Status Register Byte 2 command is the only way to modify the RSTE and SLE bits in the Status Register during normal device operation, and the SLE bit can only be modified if the sector lockdown state has not been frozen. Before the Write Status Register Byte 2 command can be issued, the Write Enable command must have been previously issued to set the WEL bit in the Status Register to a Logical 1.

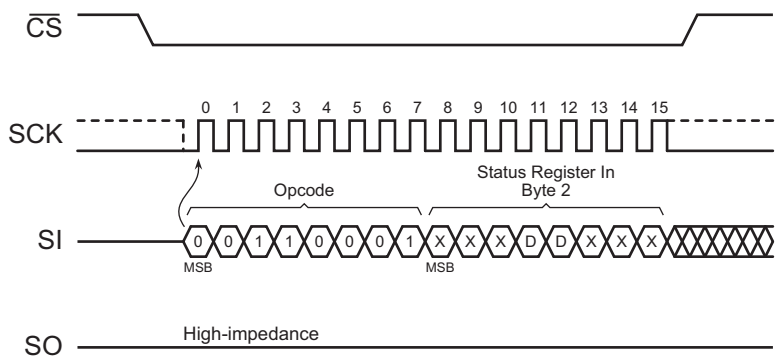
To issue the Write Status Register Byte 2 command, the $\overline{CS}$ pin must first be asserted and the opcode of 31h must be clocked into the device followed by one byte of data. The one byte of data consists of three don't care bits, the RSTE bit value, the SLE bit value, and three additional don't care bits (see Table 11-4). Any additional data bytes that are sent to the device will be ignored. When the $\overline{CS}$ pin is deasserted, the RSTE and SLE bits in the Status Register will be modified, and the WEL bit in the Status Register will be reset back to a Logical 0. The SLE bit will only be modified if the Freeze Sector Lockdown State command has not been previously issued.

The complete one byte of data must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on even byte boundaries (multiples of eight bits); otherwise, the device will abort the operation, the state of the RSTE and SLE bits will not change, and the WEL bit in the Status Register will be reset back to the Logical 0 state.

Table 11-4. Write Status Register Byte 2 Format

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X	X	X	RSTE	SLE	X	X	X

Figure 11-3. Write Status Register Byte 2



11.4 Read Configuration Register

The non-volatile Configuration Register can be read to determine if the Quad-Input Byte/Page Program and Quad-Output Read Array commands have been enabled. Unlike the Status Register, the Configuration Register can only be read when the device is in an idle state (when the RDY/BSY bit of the Status Register indicates that the device is in a ready state).

To read the Configuration Register, the $\overline{CS}$ pin must first be asserted and the opcode of 3Fh must be clocked into the device. After the opcode has been clocked in, the device will begin outputting the one byte of Configuration Register data on the SO pin during subsequent clock cycles. The data being output will be a repeating byte as long as the $\overline{CS}$ pin remains asserted and the clock pin is being pulsed.

At clock frequencies above f_{CLK}, the first byte of data output will not be valid. Therefore, if operating at clock frequencies above f_{CLK}, at least two bytes of data must be clocked out from the device in order to determine the correct value of the Configuration Register.

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

The Configuration Register is a non-volatile register; therefore, the contents of the Configuration Register are not affected by power cycles or power-on reset operations.

Table 11-5. Configuration Register Format

Bit(1)	Name		Type(2)	Description	
7	QE	Quad Enable	R/W	0	Quad-Input/Output commands and operation disabled.
				1	Quad-Input/Output commands and operation enabled ($\overline{WP}$ and $\overline{HOLD}$ disabled).
6:0	RES	Reserved for future use	R	0	Reserved for future use.

Notes: 1. Only bit 7 of the Configuration Register will be modified when using the Write Configuration Register command.

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

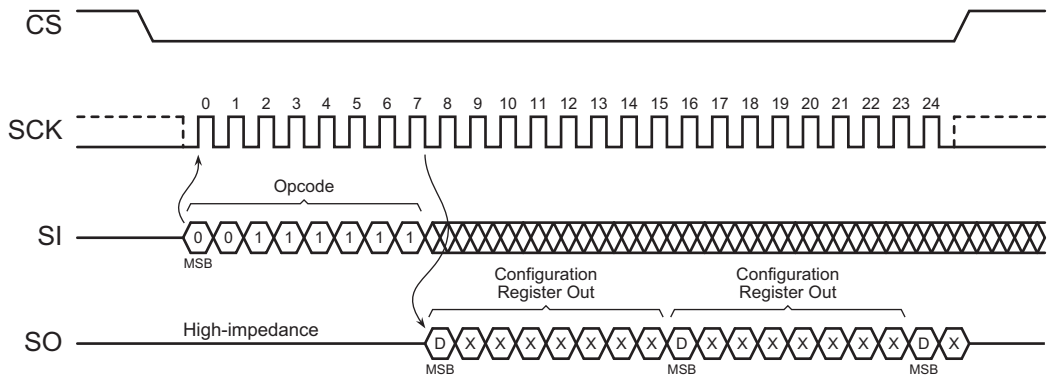
11.4.1 QE Bit

The QE bit is used to control whether the Quad-Input Byte/Page Program and Quad-Output Read Array commands are enabled or disabled. When the QE bit is in the Logical 1 state, the Quad-Input Byte/Page Program and Quad-Output Read Array commands are enabled and will be recognized by the device. In addition, the $\overline{WP}$ and $\overline{HOLD}$ functions are disabled, and the $\overline{WP}$ and $\overline{HOLD}$ pins themselves operate as a bidirectional input/output pins ($\overline{WP}$ is I/O_2 and $\overline{HOLD}$ is I/O_3).

When the QE bit is in the Logical 0 state, the Quad-Input Byte/Page Program and Quad-Output Read Array commands are disabled and will not be recognized by the device as valid commands, and the $\overline{WP}$ and $\overline{HOLD}$ pins function as normal control pins. The $\overline{WP}$ and $\overline{HOLD}$ pins should be externally pulled-high to avoid erroneous or unwanted device operation.

The Reset command has no effect on the QE bit. The QE bit defaults to the Logical 0 state when devices are initially shipped from Atmel.

Figure 11-4. Read Configuration Register



11.5 Write Configuration Register

The Write Configuration Register command is used to modify the QE bit of the non-volatile Configuration Register. Before the Write Configuration 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 Configuration Register command, the $\overline{CS}$ pin must first be asserted and the opcode of 3Eh must be clocked into the device followed by one byte of data. The one byte of data consists of the QE bit value and seven don't care bits (Table 11-6). Any additional data clocked into the device will be ignored. When the $\overline{CS}$ pin is deasserted, the QE bit of the Configuration Register will be modified within a time of t_{WRCR}, and the WEL bit in the Status Register will be reset back to a Logical 0.

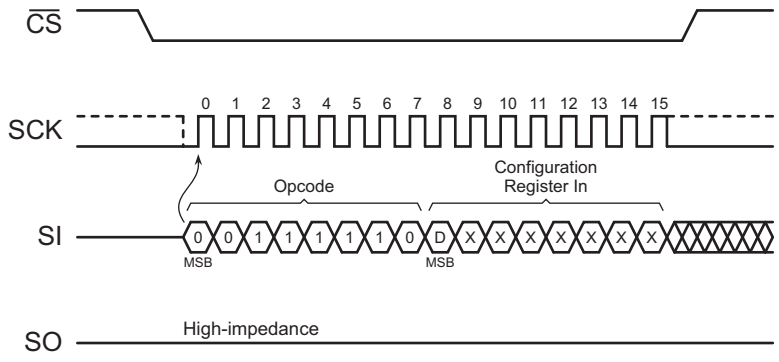
The complete one byte of data must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on byte boundaries (multiples of eight bits); otherwise, the device will abort the operation, the value of the Configuration Register will not change, and the WEL bit in the Status Register will be reset back to the Logical 0 state.

The Configuration Register is a non-volatile register and is subject to the same program/erase endurance characteristics of the main memory array.

Table 11-6. Write Configuration Register Format

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
QE	X	X	X	X	X	X	X

Figure 11-5. Write Configuration Register



12. Other Commands and Functions

12.1 Reset

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

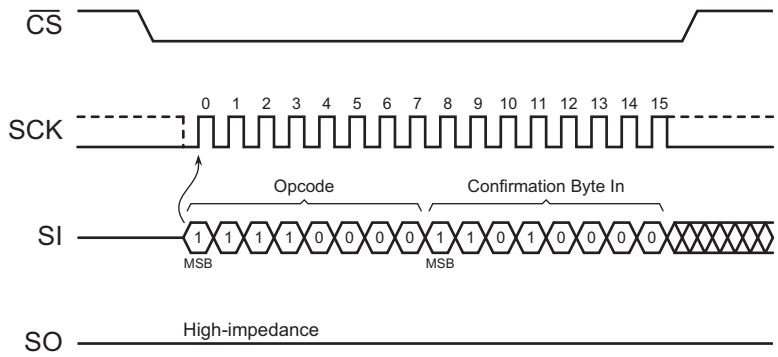
The Reset command can only be executed if the command has been enabled by setting the Reset Enabled (RSTE) bit in the Status Register to a Logical 1. If the Reset command has not been enabled (the RSTE bit is in the Logical 0 state), then any attempts at executing the Reset command will be ignored.

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

The Reset command has no effect on the states of the Sector Protection Registers, the Sector Lockdown Registers, the Configuration Register, or the SPRL, RSTE, and SLE bits in the Status Register. The WEL, PS, and ES bits of the Status Register, however, will be reset back to their default states. If a Reset operation is performed while a sector is erase suspended, the suspend operation will abort, and the contents of the block being erased in the suspended sector will be left in an undefined state. If a Reset is performed while a sector is program suspended, the suspend operation will abort, and the contents of the page that was being programmed and subsequently suspended will be undefined. The remaining pages in the 64KB sector will retain their previous contents.

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

Figure 12-1. Reset



12.2 Read Manufacturer and Device ID

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

The Read Manufacturer and Device ID command is limited to a maximum clock frequency of f_{CLK}. 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 that all devices to be used in the application can be identified properly. Once the identification process is complete, the application can then increase the clock frequency to accommodate specific Flash devices that are capable of operating at the higher clock frequencies.

To read the identification information, the $\overline{CS}$ pin must first be asserted and the opcode of 9Fh must be clocked into the device. After the opcode has been clocked in, the device will begin outputting the identification data on the SO pin during the subsequent clock cycles. The first byte that will be output will be the Manufacturer ID followed by two bytes of Device ID information. The fourth byte output will be the Extended Device Information (EDI) String Length, which will be 01h indicating that one byte of EDI data follows. After the one byte of EDI data is output, the SO pin will go into a high-impedance state; therefore, additional clock cycles will have no affect on the SO pin and no data will be output. As indicated in the JEDEC standard, reading the EDI String Length and any subsequent data is optional.

Deasserting the $\overline{CS}$ pin will terminate the Manufacturer and Device ID read operation and put the SO pin into a high-impedance state. The $\overline{CS}$ pin can be deasserted at any time and does not require that a full byte of data be read.

Table 12-1. Manufacturer and Device ID Information

Byte No.	Data Type	Value
1	Manufacturer ID	1Fh
2	Device ID (Part 1)	86h
3	Device ID (Part 2)	00h
4	[Optional to read] Extended Device Information (EDI) String Length	01h
5	[Optional to read] EDI Byte 1	00h

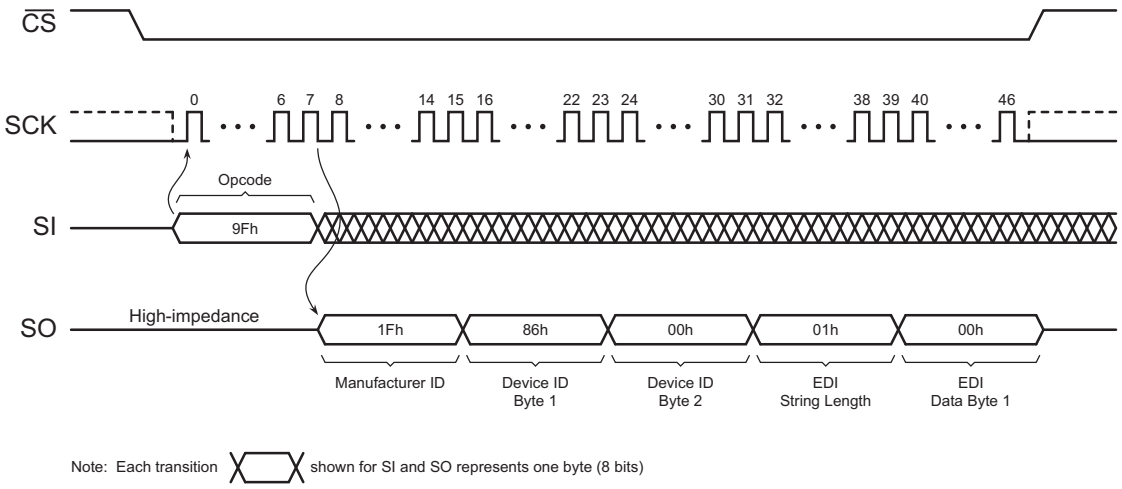
Table 12-2. Manufacturer and Device ID Details

Data Type	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Hex Value	Details
Manufacturer ID	JEDEC Assigned Code								1Fh	JEDEC Code: 0001 1111 (1Fh for Atmel)
	0	0	0	1	1	1	1	1		
Device ID (Part 1)	Family Code			Density Code					86h	Family Code: 100 (Quad-I/O or Rapid4™) Density Code: 00110 (16-Mbit)
	1	0	0	0	0	1	1	0		
Device ID (Part 2)	Sub Code			Product Variant					00h	Sub Code: 000 (Quad-I/O series) Product Variant: 00000 (Standard version)
	0	0	0	0	0	0	0	0		

Table 12-3. EDI Data

Byte Number	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Hex Value	Details
1	RFU			Device Revision					00h	RFU: Reserved for Future Use Density Code: 00000 (Initial version)
	0	0	0	0	0	0	0	0		

Figure 12-2. Read Manufacturer and Device ID



12.3 Deep Power-Down

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

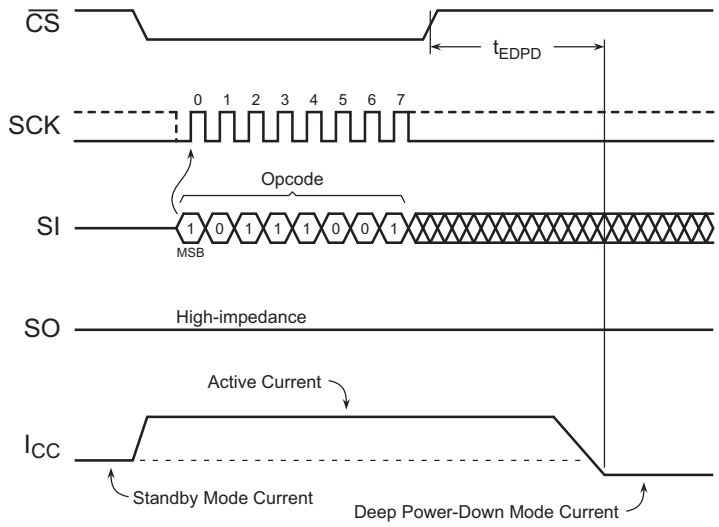
When the device is in the Deep Power-Down mode, all commands including the Read Status Register command will be ignored with the exception of the Resume from Deep Power-Down command. Since all commands will be ignored, the mode can be used as an extra protection mechanism against program and erase operations.

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

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

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

Figure 12-3. Deep Power-Down



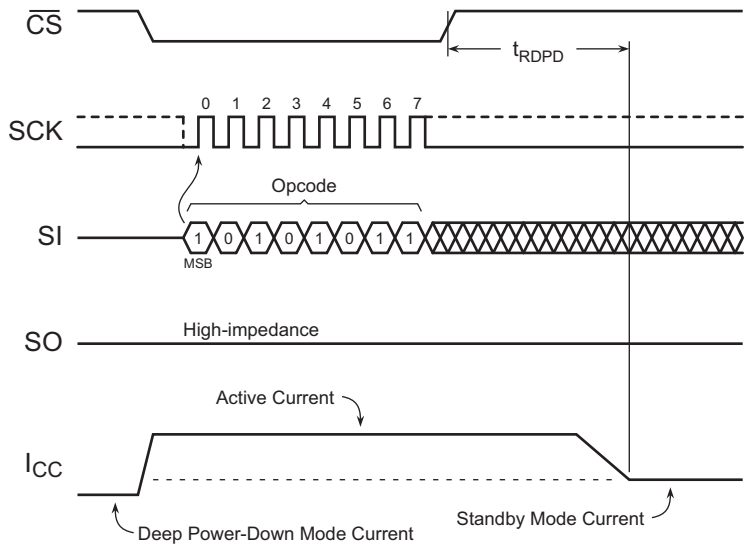
12.4 Resume from Deep Power-Down

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

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

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

Figure 12-4. Resume from Deep Power-Down



12.5 Hold

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

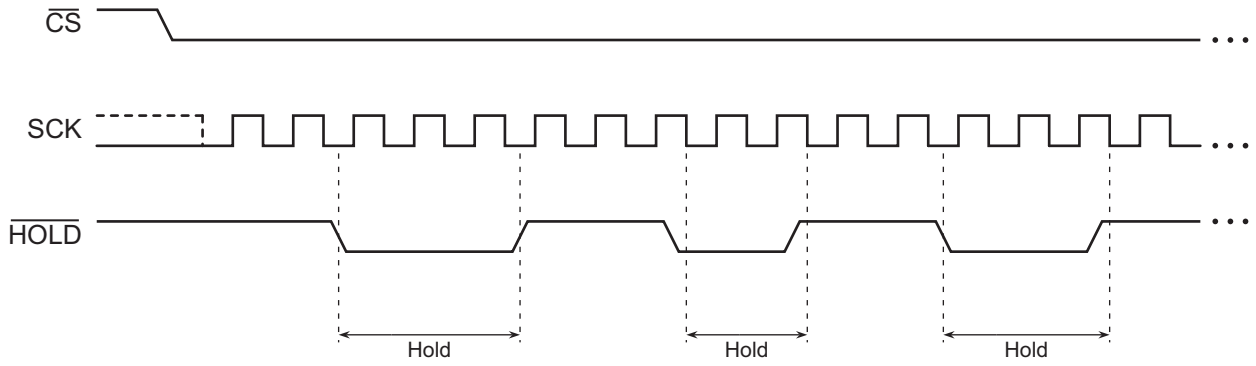
The Hold mode can only be entered while the $\overline{CS}$ pin is asserted. The Hold mode is activated simply by asserting the $\overline{HOLD}$ pin during the SCK low pulse. If the $\overline{HOLD}$ pin is asserted during the SCK high pulse, then the Hold mode won't be started until the beginning of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

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

To end the Hold mode and resume serial communication, the $\overline{HOLD}$ pin must be deasserted during the SCK low pulse. If the $\overline{HOLD}$ 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 $\overline{CS}$ pin is deasserted while the $\overline{HOLD}$ pin is still asserted, then any operation that may have been started will be aborted, and the device will reset the WEL bit in the Status Register back to the Logical 0 state.

Figure 12-5. Hold Mode



13. Atmel RapidS Implementation

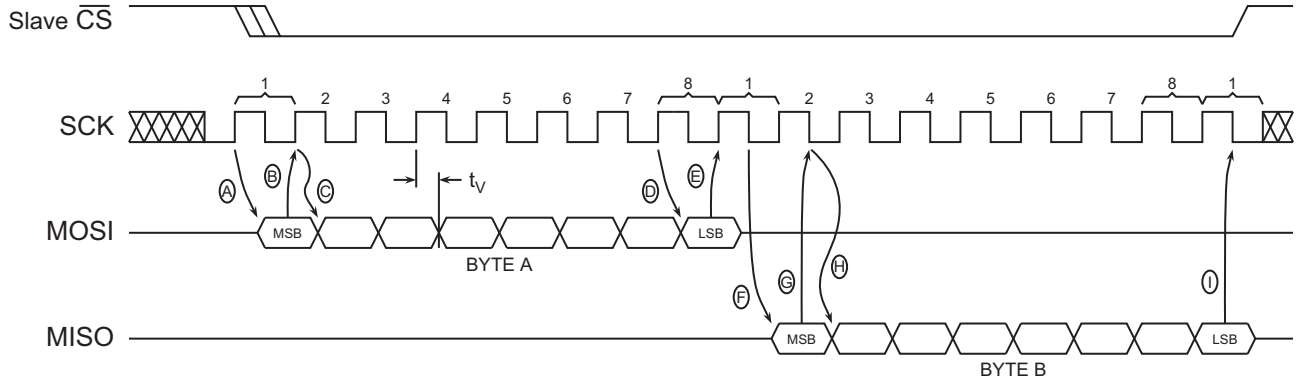
To implement RapidS and operate at clock frequencies higher than what can be achieved in a viable SPI implementation, a full clock cycle can be used to transmit data back and forth across the serial bus. The AT25DQ161 is designed to always clock its data out on the falling edge of the SCK signal and clock data in on the rising edge of SCK.

For full clock cycle operation to be achieved, when the AT25DQ161 is clocking data out on the falling edge of SCK, the host controller should wait until the next falling edge of SCK to latch the data in. Similarly, the host controller should clock its data out on the rising edge of SCK in order to give the AT25DQ161 a full clock cycle to latch the incoming data in on the next rising edge of SCK.

Implementing RapidS allows a system to run at higher clock frequencies since a full clock cycle is used to accommodate a device's clock-to-output time, input setup time, and associated rise/fall times.

Example: If the system clock frequency is 100MHz (10ns cycle time) with a 50% duty cycle, and the host controller has an input setup time of 2ns, then a standard SPI implementation would require that the slave device be capable of outputting its data in less than 3ns to meet the 2ns host controller setup time [(10ns x 50%) – 2ns] not accounting for rise/fall times. In an SPI Mode 0 or 3 implementation, the SPI Master is designed to clock in data on the next immediate rising edge of SCK after the SPI slave has clocked its data out on the preceding falling edge. This essentially makes SPI a half-clock cycle protocol and requires extremely fast clock-to-output times and input setup times in order to run at high clock frequencies. With a RapidS implementation of this example, however, the full 10ns cycle time is available which gives the slave device up to 8ns, not accounting for rise/fall times, to clock its data out. Likewise, with RapidS, the host controller has more time available to output its data to the slave since the slave device would be clocking that data in a full clock cycle later.

Figure 13-1. Atmel RapidS Operation



MOSI = Master Out, Slave In MISO = Master In, Slave Out
The ***Master*** is the ASIC/MCU and the Slave is the memory device.

The ***Master*** always clocks data ***out on the rising edge*** of SCK and always clocks data ***in on the falling edge*** of SCK.
The ***Slave*** always clocks data ***out on the falling edge*** of SCK and always clocks data ***in on the rising edge*** of SCK.

A. ***Master*** clocks ***out*** first bit of BYTE A on the ***rising edge*** of SCK.
B. ***Slave*** clocks ***in*** first bit of BYTE A on the next ***rising edge*** of SCK.
C. Master clocks out second bit of BYTE A on the same rising edge of SCK.
D. Last bit of BYTE A is clocked out from the Master.
E. Last bit of BYTE A is clocked into the slave.
F. Slave clocks out first bit of BYTE B.
G. Master clocks in first bit of BYTE B.
H. Slave clocks out second bit of BYTE B.
I. Master clocks in last bit of BYTE B.

14. Electrical Specifications

14.1 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. Functional operation of the device at these ratings 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. Voltage extremes referenced in the "Absolute Maximum Ratings" are intended to accommodate short duration undershoot/overshoot conditions and does not imply or guarantee functional device operation at these levels for any extended period of time.

14.2 DC and AC Operating Range

		Atmel AT25DQ161
Operating Temperature (Case)	Ind.	-40°C to 85°C
V_{CC} Power Supply		2.7V to 3.6V

14.3 DC Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{SB}	Standby Current	$\overline{CS}$, $\overline{WP}$, $\overline{HOLD}$ = V_{CC}, all inputs at CMOS levels		25	50	µA
I_{DPD}	Deep Power-Down Current	$\overline{CS}$, $\overline{WP}$, $\overline{HOLD}$ = V_{CC}, all inputs at CMOS levels		5	10	µA
I_{CC1}	Active Current, Read Operation	f = 100MHz; I_{OUT} = 0mA; $\overline{CS}$ = V_{IL}, V_{CC} = Max		12	19	mA
		f = 85MHz; I_{OUT} = 0mA; $\overline{CS}$ = V_{IL}, V_{CC} = Max		10	17	
		f = 66MHz; I_{OUT} = 0mA; $\overline{CS}$ = V_{IL}, V_{CC} = Max		8	14	
		f = 50MHz; I_{OUT} = 0mA; $\overline{CS}$ = V_{IL}, V_{CC} =M ax		7	12	
		f = 33MHz; I_{OUT} = 0mA; $\overline{CS}$ = V_{IL}, V_{CC} = Max		6	10	
		f = 20MHz; I_{OUT} = 0mA; $\overline{CS}$ = V_{IL}, V_{CC} = Max		5	8	
I_{CC2}	Active Current, Program Operation	$\overline{CS}$ = V_{CC}, V_{CC} = Max		10	15	mA
I_{CC3}	Active Current, Erase Operation	$\overline{CS}$ = V_{CC}, V_{CC} = Max		12	18	mA
I_{LI}	Input Leakage Current	V_{IN} = CMOS levels			1	µA
I_{LO}	Output Leakage Current	V_{OUT} = CMOS levels			1	µA
V_{IL}	Input Low Voltage				0.3 x V_{CC}	V
V_{IH}	Input High Voltage		0.7 x V_{CC}			V
V_{OL}	Output Low Voltage	I_{OL} = 1.6mA; V_{CC} = Min			0.4	V
V_{OH}	Output High Voltage	I_{OH} = -100 µA; V_{CC} = Min	V_{CC} - 0.2V			V

14.4 AC Characteristics – Maximum Clock Frequencies

Symbol	Parameter	Min	Max	Units
Atmel RapidS and SPI Operation				
f_{MAX}	Maximum Clock Frequency for All Operations – RapidS Operation Only (excluding 03h, 0Bh, 3Bh, 6Bh, and 9Fh opcodes)		100	MHz
f_{CLK}	Maximum Clock Frequency for All Operations (excluding 03h opcode)		85	MHz
f_{RDLF}	Maximum Clock Frequency for 03h Opcode (Read Array – Low Frequency)		50	MHz
f_{RDDO}	Maximum Clock Frequency for 3Bh Opcode (Dual-Output Read)		85	MHz
f_{RDQO}	Maximum Clock Frequency for 6Bh Opcode (Quad-Output Read)		85	MHz

14.5 AC Characteristics – All Other Parameters

Symbol	Parameter	Min	Max	Units
t_{CLKH}	Clock High Time	4.3		ns
t_{CLKL}	Clock Low Time	4.3		ns
t_{CLKR} (1)	Clock Rise Time, Peak-to-Peak (Slew Rate)	0.1		V/ns
t_{CLKF} (1)	Clock Fall Time, Peak-to-Peak (Slew Rate)	0.1		V/ns
t_{CSH}	Chip Select High Time	50		ns
t_{CSLS}	Chip Select Low Setup Time (relative to Clock)	5		ns
t_{CSLH}	Chip Select Low Hold Time (relative to Clock)	5		ns
t_{CSHS}	Chip Select High Setup Time (relative to Clock)	5		ns
t_{CSHH}	Chip Select High Hold Time (relative to Clock)	5		ns
t_{DS}	Data In Setup Time	2		ns
t_{DH}	Data In Hold Time	1		ns
t_{DIS} (1)	Output Disable Time		5	ns
t_{V} (2)	Output Valid Time		5	ns
t_{OH}	Output Hold Time	2		ns
t_{HLS}	$\overline{\text{HOLD}}$ Low Setup Time (relative to Clock)	5		ns
t_{HLH}	$\overline{\text{HOLD}}$ Low Hold Time (relative to Clock)	5		ns
t_{HHS}	$\overline{\text{HOLD}}$ High Setup Time (relative to Clock)	5		ns
t_{HHH}	$\overline{\text{HOLD}}$ High Hold Time (relative to Clock)	5		ns
t_{HLQZ} (1)	$\overline{\text{HOLD}}$ Low to Output High-Z		5	ns
t_{HHQX} (1)	$\overline{\text{HOLD}}$ High to Output Low-Z		5	ns
t_{WPS} (1)(3)	Write Protect Setup Time	20		ns
t_{WPH} (1)(3)	Write Protect Hold Time	100		ns
t_{SECP} (1)	Sector Protect Time (from Chip Select High)		20	ns
t_{SECUP} (1)	Sector Unprotect Time (from Chip Select High)		20	ns
t_{LOCK} (1)	Sector Lockdown and Freeze Sector Lockdown State Time (from Chip Select High)		200	µs
t_{EDPD} (1)	Chip Select High to Deep Power-Down		1	µs
t_{RDPD} (1)	Chip Select High to Standby Mode		30	µs
t_{RST}	Reset Time		30	µs

Notes: 1. Not 100% tested (value guaranteed by design and characterization).
2. 15 pF load at frequencies above 70MHz, 30pF otherwise.
3. Only applicable as a constraint for the Write Status Register Byte 1 command when SPRL = 1.

14.6 Program and Erase Characteristics

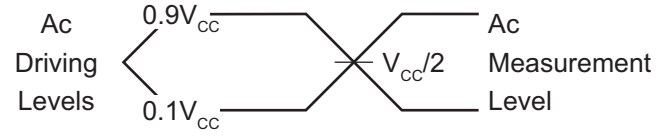
Symbol	Parameter		Min	Typ	Max	Units
t_{PP} (1)	Page Program Time (256 bytes)			1.0	3.0	ms
t_{BP}	Byte Program Time			7		µs
t_{BLKE} (1)	Block Erase Time	4KB		50	200	ms
		32KB		250	600	
		64KB		400	950	
t_{CHPE} (1)(2)	Chip Erase Time			12	28	sec
t_{SUSP}	Suspend Time	Program		10	20	µs
		Erase		25	40	
t_{RES}	Resume Time	Program		10	20	µs
		Erase		12	20	
t_{OTPP} (1)	OTP Security Register Program Time			200	500	µs
t_{WRSR} (2)	Write Status Register Time				200	ns

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

14.7 Power-up Conditions

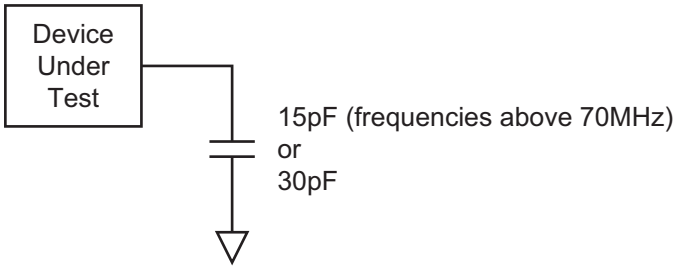
Symbol	Parameter	Min	Max	Units
t_{VCSL}	Minimum V_{CC} to Chip Select Low Time	70		µs
t_{PUW}	Power-up Device Delay Before Program or Erase Allowed		10	ms
V_{POR}	Power-on Reset Voltage	1.5	2.5	V

14.8 Input Test Waveforms and Measurement Levels



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

14.9 Output Test Load



15. AC Waveforms

Figure 15-1. Serial Input Timing

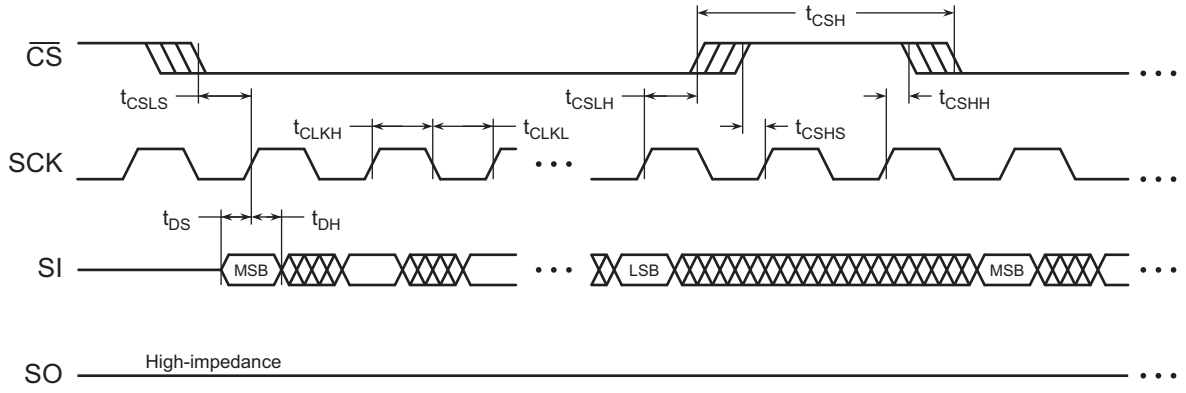


Figure 15-2. Serial Output Timing

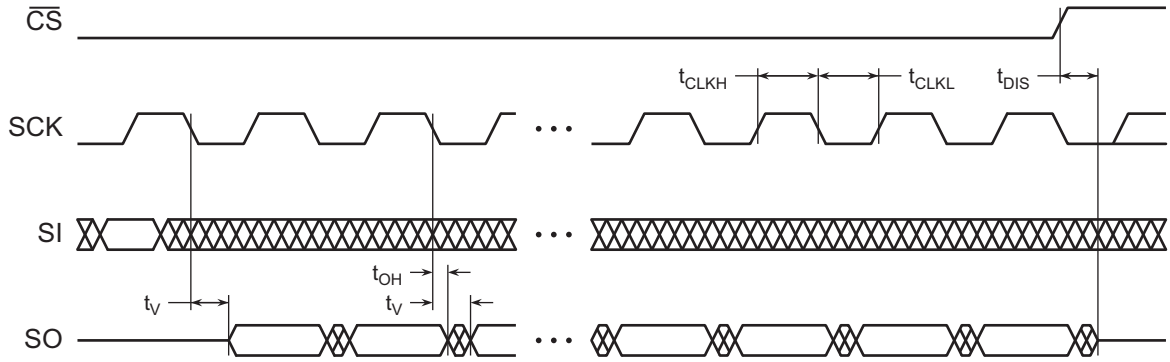


Figure 15-3. $\overline{WP}$ Timing for Write Status Register Byte 1 Command When SPRL = 1

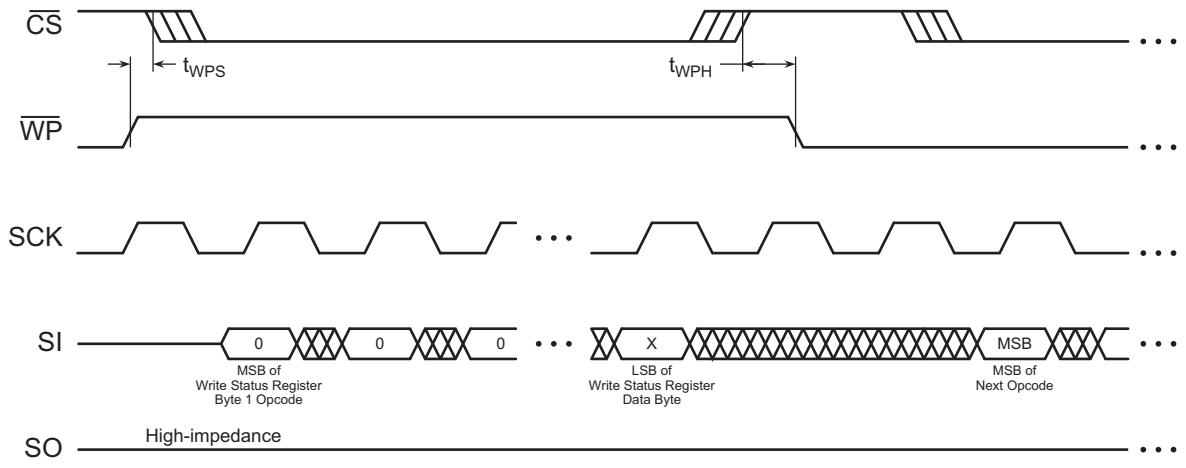


Figure 15-4. $\overline{\text{HOLD}}$ Timing – Serial Input

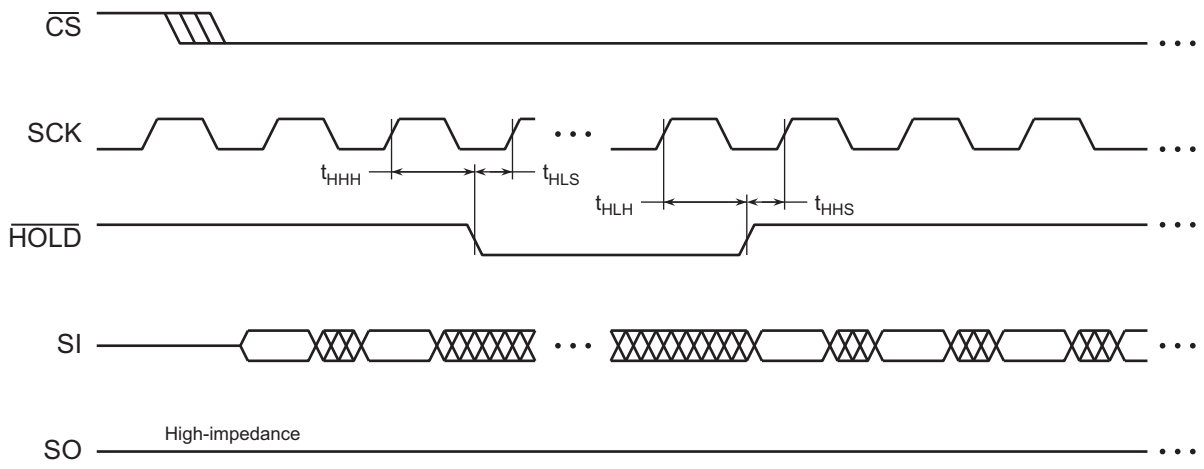
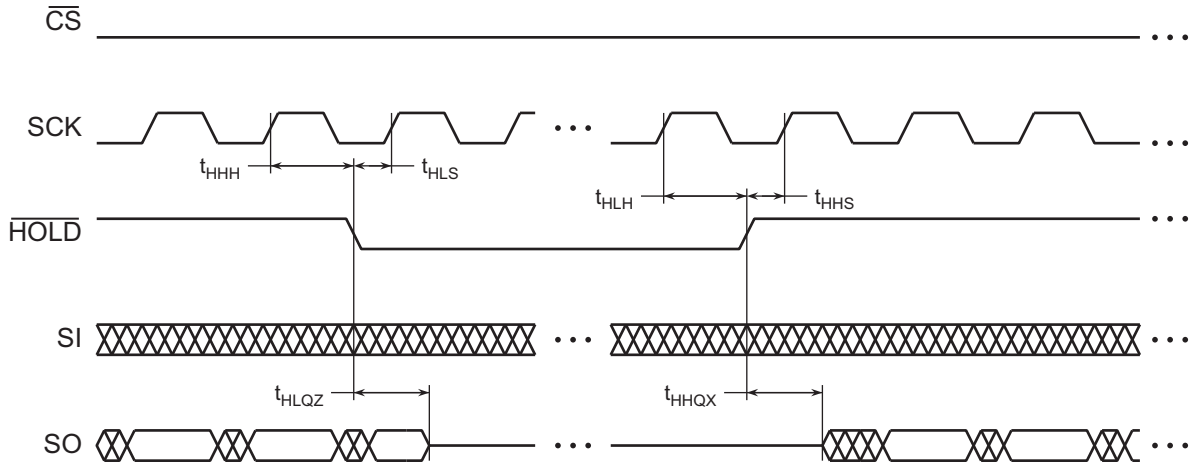
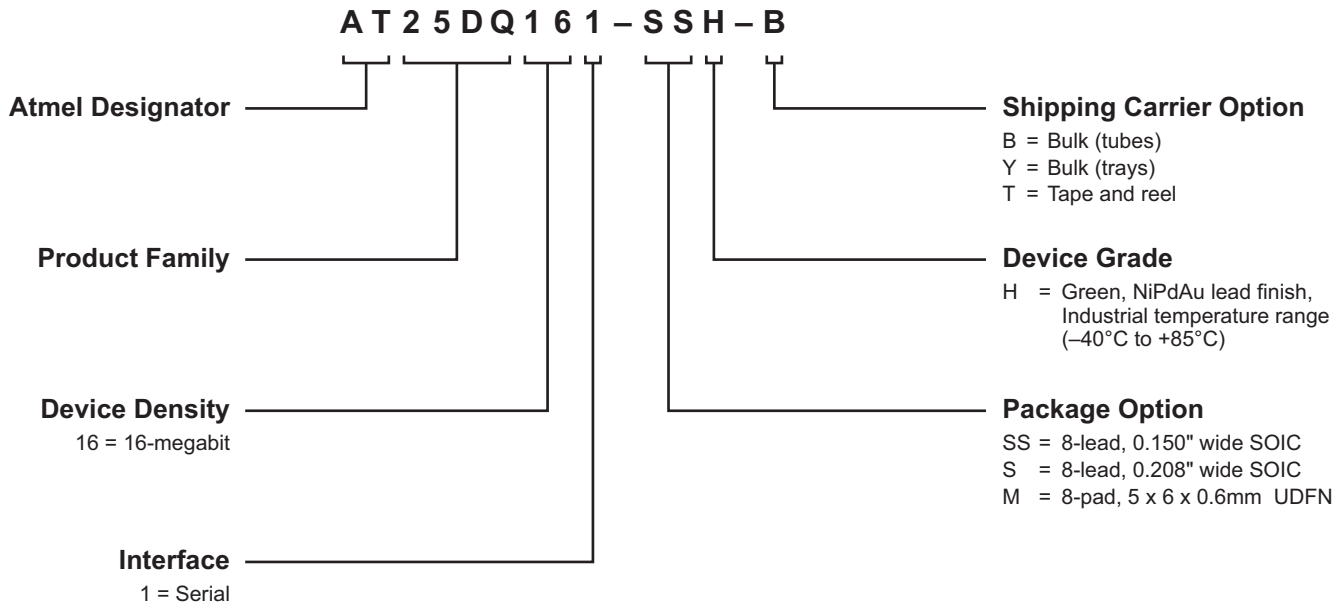


Figure 15-5. $\overline{\text{HOLD}}$ Timing – Serial Output



16. Ordering Information

16.1 Ordering Code Detail



16.2 Green Package Options (Pb/Halide-free/RoHS Compliant)

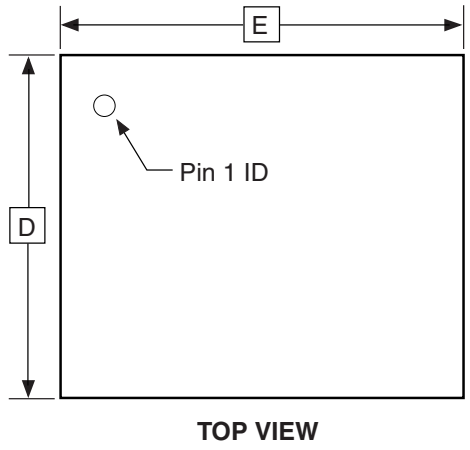
Atmel Ordering Code	Package	Lead (Pad) Finish	Operating Voltage	Max. Freq. (MHz)	Operation Range
AT25DQ161-MH-Y AT25DQ161-MH-T	8MA1	NiPdAu	2.7V to 3.6V	100	Industrial (-40°C to +85°C)
AT25DQ161-SSH-B AT25DQ161-SSH-T	8S1				
AT25DQ161-SH-B AT25DQ161-SH-T	8S2				

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

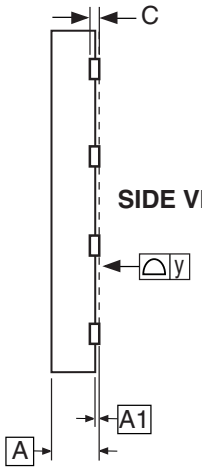
Package Type	
8MA1	8-pad (5 x 6 x 0.6mm body), Thermally Enhanced Plastic Ultra Thin Dual Flat No Lead (UDFN)
8S1	8-lead, 0.150” wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
8S2	8-lead, 0.208” wide, Plastic Gull Wing Small Outline (EIAJ SOIC)

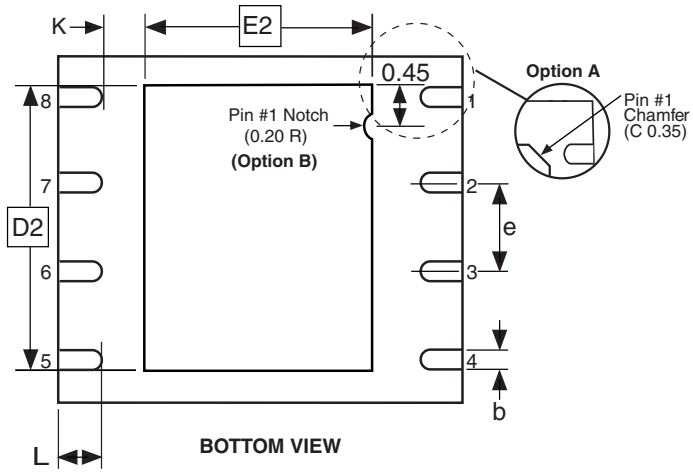
17. Packaging Information

17.1 8MA1 – UDFN



TOP VIEW





BOTTOM VIEW

COMMON DIMENSIONS
(Unit of Measure = mm)

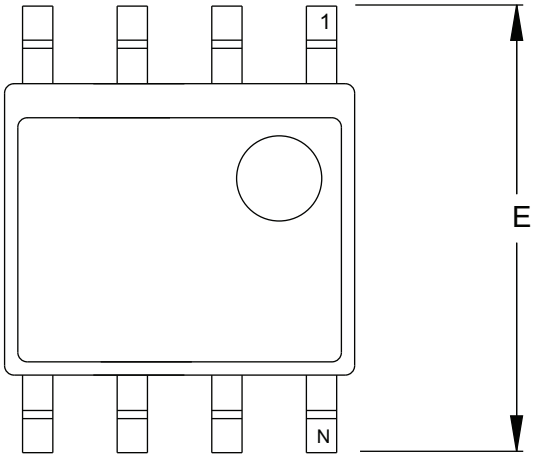
SYMBOL	MIN	NOM	MAX	NOTE
A	0.45	0.55	0.60	
A1	0.00	0.02	0.05	
b	0.35	0.40	0.48	
C		0.152 REF		
D	4.90	5.00	5.10	
D2	3.80	4.00	4.20	
E	5.90	6.00	6.10	
E2	3.20	3.40	3.60	
e		1.27		
L	0.50	0.60	0.75	
y	0.00	–	0.08	
K	0.20	–	–	

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

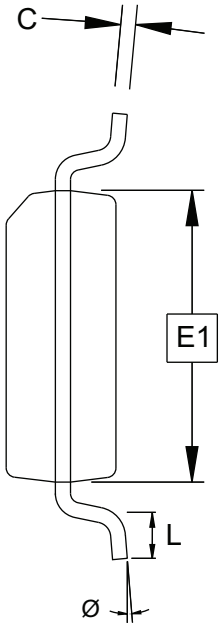
4/15/08

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

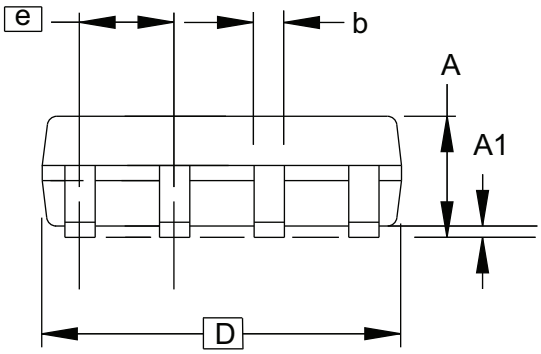
17.2 8S1 – JEDEC SOIC



TOP VIEW



END VIEW



SIDE VIEW

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

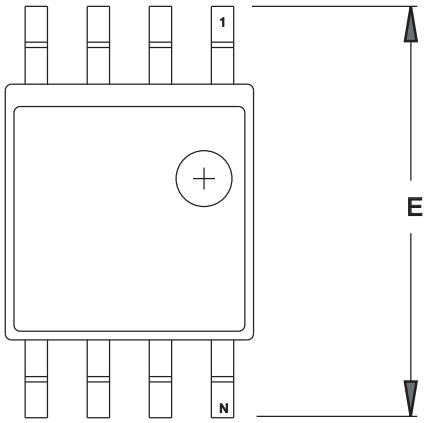
COMMON DIMENSIONS
(Unit of Measure = mm)

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

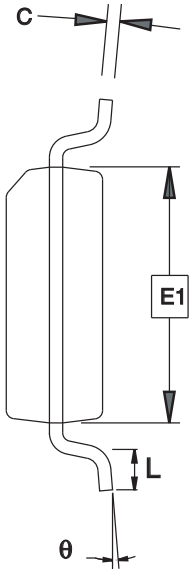
6/22/11

	TITLE	GPC	DRAWING NO.	REV.
Package Drawing Contact: packagedrawings@atmel.com	8S1, 8-lead (0.150” Wide Body), Plastic Gull Wing Small Outline (JEDEC SOIC)	SWB	8S1	G

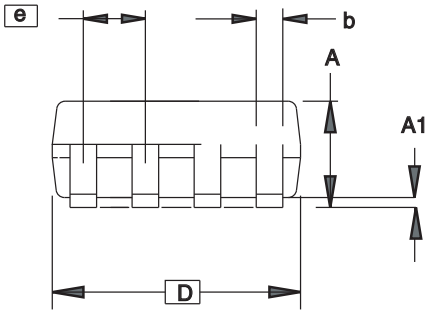
17.3 8S2 – EIAJ SOIC



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

	TITLE	GPC	DRAWING NO.	REV.
Package Drawing Contact: packagedrawings@atmel.com	**8S2,** 8-lead, 0.208” Body, Plastic Small Outline Package (EIAJ)	STN	8S2	F

18. Revision History

Doc. Rev.	Date	Comments
8671B	05/2012	Not recommended for new designs
8671B	03/2012	Update template Update package drawings Remove preliminary status Correct AC Characteristics f_{MAX} excluding 9F to 9Fh opcode
8671A	07/2009	Initial document release

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: (+1) (408) 441-0311
Fax: (+1) (408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Roa
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Bldg
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN
Tel: (+81) (3) 6417-0300
Fax: (+81) (3) 6417-0370

© 2012 Atmel Corporation. All rights reserved. / Rev.: 8671B–DFLASH–3/12

Atmel®, Atmel logo and combinations thereof, RapidS™, Rapid4™, Enabling Unlimited Possibilities®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.