Features

- Single 2.7V - 3.6V Supply
- Serial Peripheral Interface (SPI) Compatible
  - Supports SPI Modes 0 and 3
- 70 MHz Maximum Clock Frequency
- Flexible, Uniform Erase Architecture
  - 4-Kbyte Blocks
  - 32-Kbyte Blocks
  - 64-Kbyte Blocks
  - Full Chip Erase
- Optimized Physical Sectoring for Code Shadowing and Code + Data Storage Applications
  - One 32-Kbyte Top Boot Sector
  - Two 8-Kbyte Sectors
  - One 16-Kbyte Sector
  - Fifteen 64-Kbyte Sectors
- Individual Sector Protection for Program/Erase Protection
- Hardware Controlled Locking of Protected Sectors
- Flexible Programming Options
  - Byte/Page Program (1 to 256 Bytes)
  - Sequential Program Mode Capability
- JEDEC Standard Manufacturer and Device ID Read Methodology
- Low Power Dissipation
  - 7 mA Active Read Current (Typical)
  - 11 µA 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 (150-mil and 200-mil wide)

8-megabit 2.7-volt Only Serial Firmware DataFlash® Memory

AT26DF081A

Preliminary

1. Description

The AT26DF081A 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 AT26DF081A, with its erase granularity as small as 4 Kbytes, 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 AT26DF081A 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.

3600A–DFLASH–11/05

The AT26DF081A 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 program code is patched or 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.

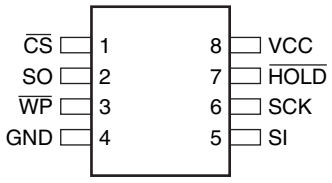
Specifically designed for use in 3-volt systems, the AT26DF081A supports read, program, and erase operations with a supply voltage range of 2.7V to 3.6V. No separate voltage is required for programming and erasing.

2. Pin Descriptions and Pinouts

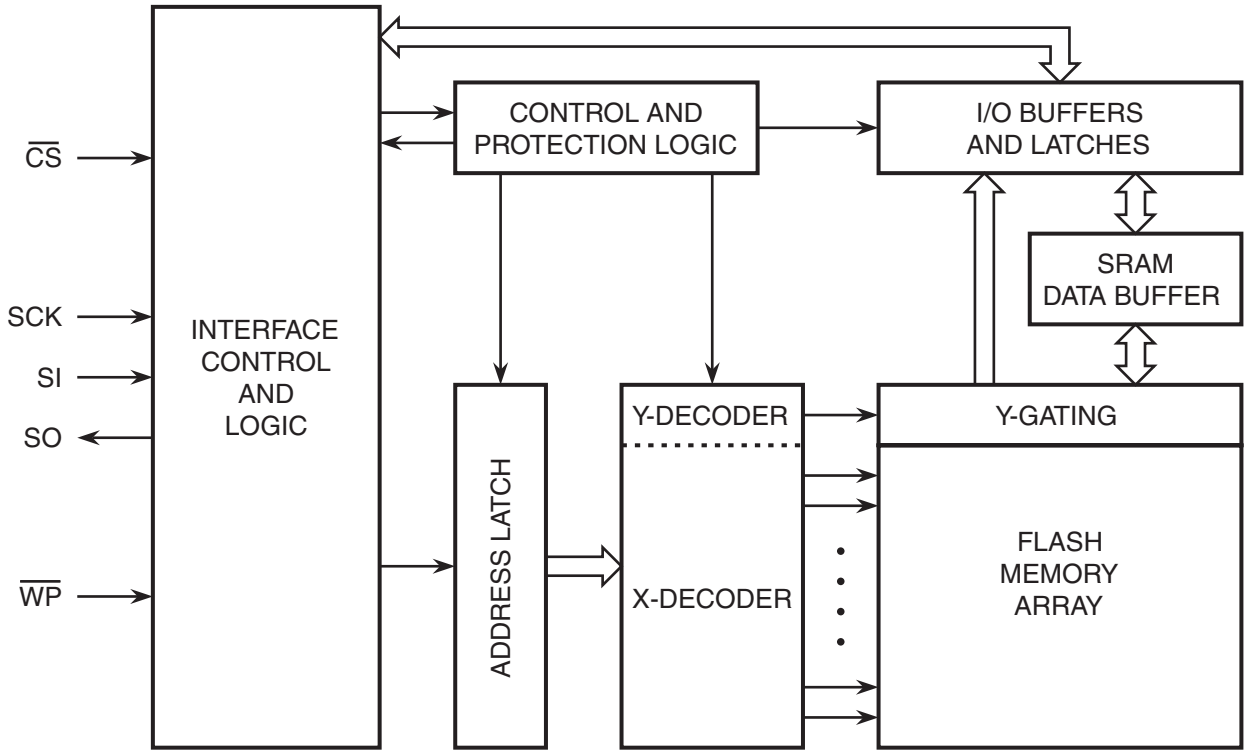
Table 2-1. Pin Descriptions

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 is always latched on the rising edge of SCK, while output data on the SO pin is always clocked out on the falling edge of SCK.		Input
SI	**SERIAL INPUT:** The SI pin is used to shift data into the device. The SI pin is used for all data input including command and address sequences. Data on the SI pin is always latched on the rising edge of SCK.		Input
SO	**SERIAL OUTPUT:** The SO pin is used to shift data out from the device. Data on the SO pin is always clocked out on the falling edge of SCK.		Output
$\overline{WP}$	**WRITE PROTECT:** The $\overline{WP}$ pin controls the hardware locking feature of the device. Please refer to section "Protection Commands and Features" on page 14 for more details on protection features and the $\overline{WP}$ pin. The $\overline{WP}$ pin is internally pulled-high and may be left floating if hardware-controlled protection will not be used. However, it is recommended that the $\overline{WP}$ pin also be externally connected to V_{CC} whenever possible.	Low	Input
$\overline{HOLD}$	**HOLD:** The $\overline{HOLD}$ pin is used to temporarily pause serial communication without deselecting or resetting the device. While the $\overline{HOLD}$ pin is asserted, transitions on the SCK pin and data on the SI pin will be ignored, and the SO pin will be in a high-impedance state. The $\overline{CS}$ pin must be asserted, and the SCK pin must be in the low state in order for a Hold condition to start. A Hold condition pauses serial communication only and does not have an effect on internally self-timed operations such as a program or erase cycle. Please refer to section "Hold" on page 26 for additional details on the Hold operation. The $\overline{HOLD}$ pin is internally pulled-high and may be left floating if the Hold function will not be used. However, it is recommended that the $\overline{HOLD}$ pin also be externally connected to V_{CC} whenever possible.	Low	Input
V_{CC}	**DEVICE POWER SUPPLY:** The V_{CC} pin is used to supply the source voltage to the device. 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



3. Block Diagram



4. Memory Array

To provide the greatest flexibility, the memory array of the AT26DF081A can be erased in four levels of granularity including a full chip erase. In addition, the array has been divided into physical sectors of various sizes, of which each sector can be individually protected from program and erase operations. The sizes of the physical sectors are optimized for both code and data storage applications, allowing both code and data segments to reside in their own isolated regions. The Figure 4-1 on page 4 illustrates the breakdown of each erase level as well as the breakdown of each physical sector.

Figure 4-1. Memory Architecture Diagram

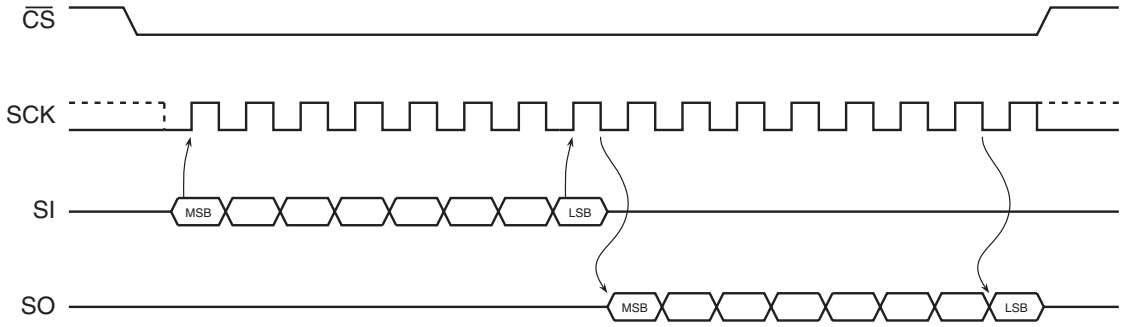
	Block Erase Detail				Page Program Detail	
Internal Sectoring for Sector Protection Function	64KB Block Erase (D8h Command)	32KB Block Erase (52h Command)	4KB Block Erase (20h Command)	Block Address Range	1-256 Byte Page Program (02h Command)	Page Address Range
32KB (Sector 18)	64KB	32KB	4KB	0FFFFFh – 0FF000h	256 Bytes	0FFFFFh – 0FFF00h
			4KB	0FEFFFh – 0FE000h	256 Bytes	0FFEFFh – 0FFE00h
			4KB	0FDFFFh – 0FD000h	256 Bytes	0FFDFFh – 0FFD00h
			4KB	0FCFFFh – 0FC000h	256 Bytes	0FFCFFh – 0FFC00h
			4KB	0FBFFFh – 0FB000h	256 Bytes	0FFBFFh – 0FFB00h
			4KB	0FAFFFh – 0FA000h	256 Bytes	0FFAFFh – 0FFA00h
			4KB	0F9FFFh – 0F9000h	256 Bytes	0FF9FFh – 0FF900h
			4KB	0F8FFFh – 0F8000h	256 Bytes	0FF8FFh – 0FF800h
8KB (Sector 17)		32KB	4KB	0F7FFFh – 0F7000h	256 Bytes	0FF7FFh – 0FF700h
			4KB	0F6FFFh – 0F6000h	256 Bytes	0FF6FFh – 0FF600h
8KB (Sector 16)			4KB	0F5FFFh – 0F5000h	256 Bytes	0FF5FFh – 0FF500h
			4KB	0F4FFFh – 0F4000h	256 Bytes	0FF4FFh – 0FF400h
16KB (Sector 15)			4KB	0F3FFFh – 0F3000h	256 Bytes	0FF3FFh – 0FF300h
			4KB	0F2FFFh – 0F2000h	256 Bytes	0FF2FFh – 0FF200h
			4KB	0F1FFFh – 0F1000h	256 Bytes	0FF1FFh – 0FF100h
			4KB	0F0FFFh – 0F0000h	256 Bytes	0FF0FFh – 0FF000h
64KB (Sector 14)	64KB	32KB	4KB	0EFFFFh – 0EF000h	256 Bytes	0FEFFFh – 0FEF00h
			4KB	0EEFFFh – 0EE000h	256 Bytes	0FEEFFh – 0FEE00h
			4KB	0EDFFFh – 0ED000h	256 Bytes	0FEDFFh – 0FED00h
			4KB	0ECFFFh – 0EC000h	256 Bytes	0FECFFh – 0FEC00h
			4KB	0EBFFFh – 0EB000h	256 Bytes	0FEBFFh – 0FEB00h
			4KB	0EAFFFh – 0EA000h	256 Bytes	0FEAFFh – 0FEA00h
			4KB	0E9FFFh – 0E9000h	256 Bytes	0FE9FFh – 0FE900h
			4KB	0E8FFFh – 0E8000h	256 Bytes	0FE8FFh – 0FE800h
		32KB	4KB	0E7FFFh – 0E7000h	⋮	
			4KB	0E6FFFh – 0E6000h		
			4KB	0E5FFFh – 0E5000h		
			4KB	0E4FFFh – 0E4000h	256 Bytes	0017FFh – 001700h
			4KB	0E3FFFh – 0E3000h	256 Bytes	0016FFh – 001600h
			4KB	0E2FFFh – 0E2000h	256 Bytes	0015FFh – 001500h
			4KB	0E1FFFh – 0E1000h	256 Bytes	0014FFh – 001400h
			4KB	0E0FFFh – 0E0000h	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 AT26DF081A 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 AT26DF081A 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 AT26DF081A 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



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 SPI Master 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 SPI Master. 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 AT26DF081A 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 AT26DF081A memory array is 0FFFFFh, address bits A23 - A20 are always ignored by the device.

Table 6-1. Command Listing

Command	Opcode		Address Bytes	Dummy Bytes	Data Bytes
Read Commands					
Read Array	0Bh	0000 1011	3	1	1+
Read Array (Low Frequency)	03h	0000 0011	3	0	1+
Program and Erase Commands					
Block Erase (4 Kbytes)	20h	0010 0000	3	0	0
Block Erase (32 Kbytes)	52h	0101 0010	3	0	0
Block Erase (64 Kbytes)	D8h	1101 1000	3	0	0
Chip Erase	60h	0110 0000	0	0	0
	C7h	1100 0111	0	0	0
Byte/Page Program (1 to 256 Bytes)	02h	0000 0010	3	0	1+
Sequential Program Mode	ADh	1010 1101	3, 0[1]	0	1
	AFh	1010 1111	3, 0[1]	0	1
Protection Commands					
Write Enable	06h	0000 0110	0	0	0
Write Disable	04h	0000 0100	0	0	0
Protect Sector	36h	0011 0110	3	0	0
Unprotect Sector	39h	0011 1001	3	0	0
Read Sector Protection Registers	3Ch	0011 1100	3	0	1+
Status Register Commands					
Read Status Register	05h	0000 0101	0	0	1+
Write Status Register	01h	0000 0001	0	0	1
Miscellaneous Commands					
Read Manufacturer and Device ID	9Fh	1001 1111	0	0	1 to 4
Deep Power-down	B9h	1011 1001	0	0	0
Resume from Deep Power-down	ABh	1010 1011	0	0	0

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

7. Read Commands

7.1 Read Array

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

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

To perform the Read Array operation, the $\overline{CS}$ pin must first be asserted and the appropriate opcode (0Bh or 03h) must be clocked into the device. After the opcode has been clocked in, the three address bytes must be clocked in to specify the starting address location of the first byte to read within the memory array. If the 0Bh opcode is used, then one don't care byte must also be clocked in after the three address bytes.

After the three address bytes (and the one don't care byte if using opcode 0Bh) have been clocked in, additional clock cycles will result in serial data being output on the SO pin. The data is always output with the MSB of a byte first. When the last byte (0FFFFFh) 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 – 0Bh Opcode

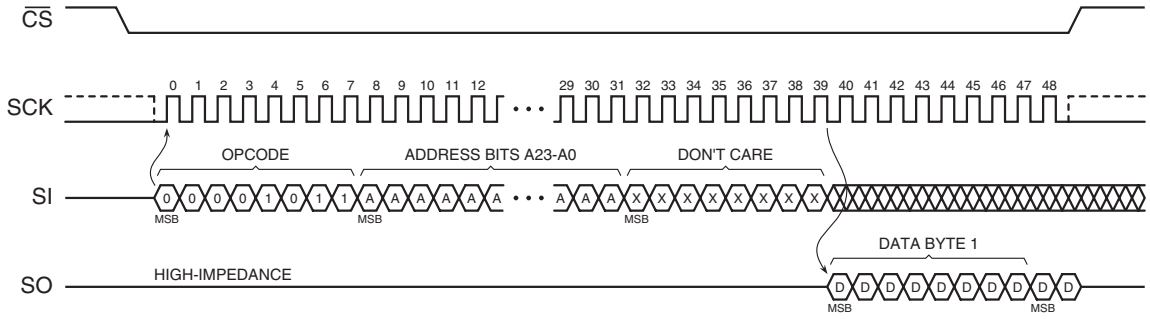
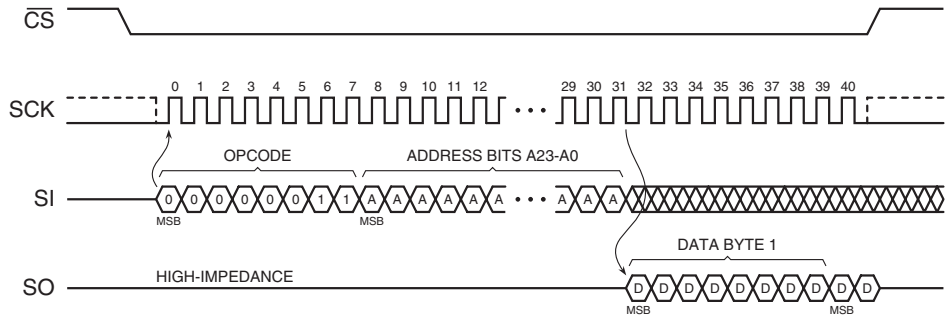


Figure 7-2. Read Array – 03h Opcode



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 14 command description) 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’s), then special circumstances regarding which memory locations will be programmed will apply. In this situation, any data that is sent to the device that goes beyond the end of the page will wrap around back to the beginning of the same page. For example, if the starting address denoted by A23 - A0 is 0000FEh, and three bytes of data are sent to the device, then the first two bytes of data will be programmed at addresses 0000FEh and 0000FFh while the last byte of data will be programmed at address 000000h. The remaining bytes in the page (addresses 000001h through 0000FDh) will be unaffected and will not change. In addition, if more than 256 bytes of data are sent to the device, then only the last 256 bytes sent will be latched into the internal buffer.

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

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 section “Protect Sector” on page 15), 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, or because the memory location to be programmed is protected.

While the device is programming, the Status Register can be read and will indicate that the device is busy. For faster throughput, it is recommended that the Status Register be polled rather than waiting the t_{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.

The Byte/Page Program mode is the default programming mode after the device powers-up or resumes from a device reset.

Figure 8-1. Byte Program

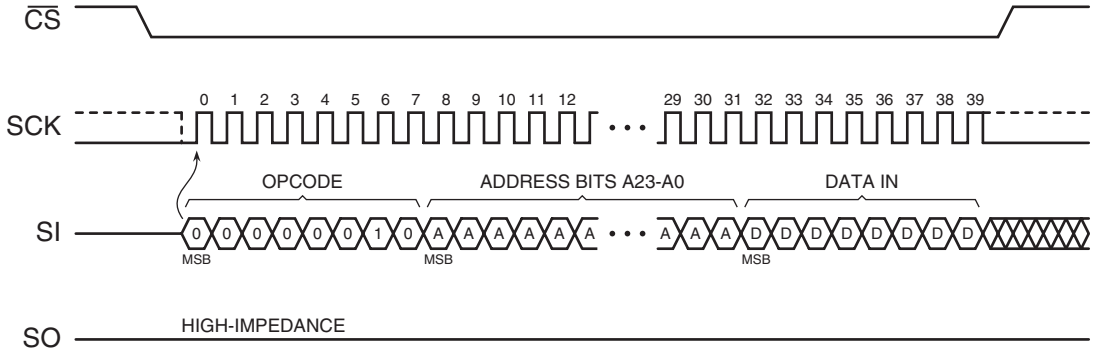
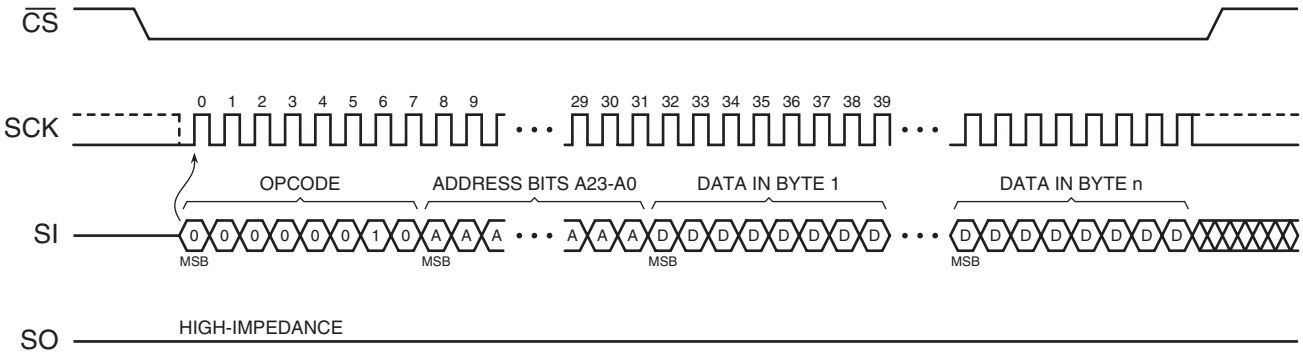


Figure 8-2. Page Program



8.2 Sequential Program Mode

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

The Sequential Programming Mode helps reduce system overhead and bus traffic by incorporating an internal address counter that keeps track of the byte location to program, thereby eliminating the need to supply an address sequence to the device for every byte to program. When using the Sequential Program mode, all address locations to be programmed must be in the erased state. Before the Sequential Program mode can first be entered, the Write Enable command must have been previously issued to the device to set the WEL bit of the Status Register to a logical “1” state.

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

After the first byte has been successfully programmed, a second byte can be programmed by simply reasserting the $\overline{CS}$ pin, clocking in the ADh or AFh opcode, and then clocking in the next byte of data. When the $\overline{CS}$ pin is deasserted, the second byte of data will be programmed into the next sequential memory location. The process would be repeated for any additional bytes. There is no need to reissue the Write Enable command once the Sequential Program Mode has been entered.

When the last desired byte has been programmed into the memory array, the Sequential Program Mode operation can be terminated by reasserting the $\overline{CS}$ pin and sending the Write Disable command to the device to reset the WEL bit in the Status Register back to the logical “0” state.

If more than one byte of data is ever clocked in during each program cycle, then only the last byte of data sent on the SI pin will be stored in the internal latches. The programming of each byte is internally self-timed and should take place in a time of t_{BP}. For each program cycle, a complete byte of data must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on even byte boundaries (multiples of eight bits); otherwise, the device will abort the operation, the byte of data will not be programmed into the memory array, and the WEL bit in the Status Register will be reset back to the logical “0” state.

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

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

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

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

Figure 8-3. Sequential Program Mode – Status Register Polling

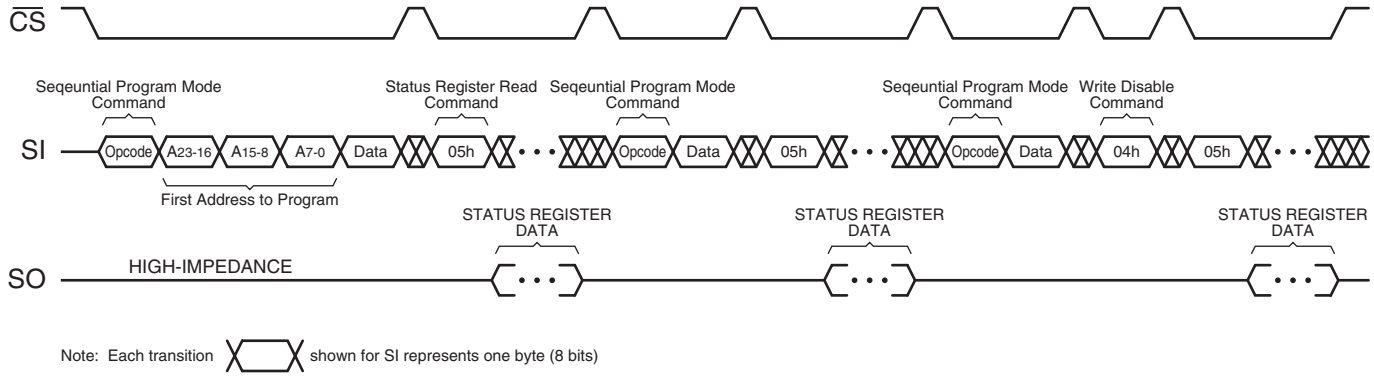
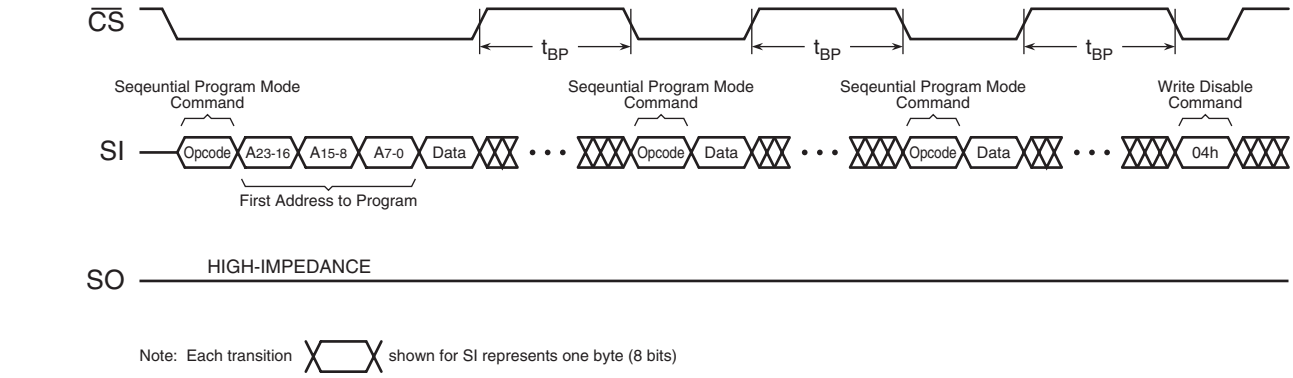


Figure 8-4. Sequential Program Mode – Waiting Maximum Byte Program Time



8.3 Block Erase

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

To perform a Block Erase, the $\overline{CS}$ pin must first be asserted and the appropriate opcode (20h, 52h or D8h) must be clocked into the device. After the opcode has been clocked in, the three address bytes specifying an address within the 4-, 32-, or 64-Kbyte block to be erased must be clocked in. Any additional data clocked into the device will be ignored. When the $\overline{CS}$ pin is deasserted, the device will erase the appropriate block. The erasing of the block is internally self-timed and should take place in a time of t_{BLKE}.

Since the Block Erase command erases a region of bytes, the lower order address bits do not need to be decoded by the device. Therefore, for a 4-Kbyte erase, address bits A11 - A0 will be ignored by the device and their values can be either a logical “1” or “0”. For a 32-Kbyte erase, address bits A14 - A0 will be ignored, and for a 64-Kbyte erase, address bits A15 - A0 will be ignored by the device. Despite the lower order address bits not being decoded by the device, the complete three address bytes must still be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on an even byte boundary (multiples of eight bits); otherwise, the device will abort the operation and no erase operation will be performed.

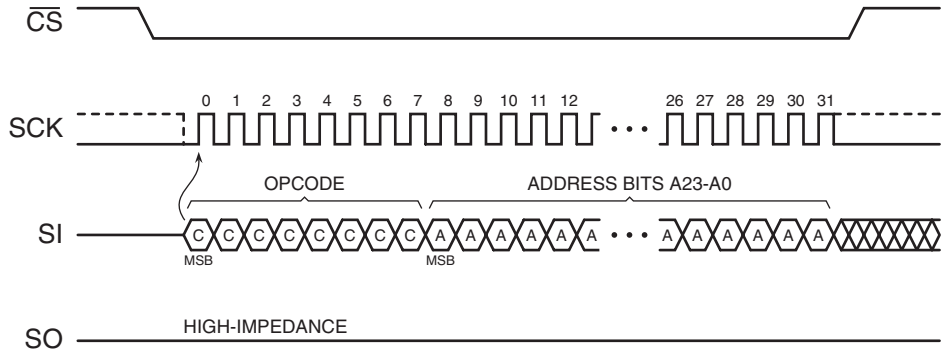
If the address specified by A23 - A0 points to a memory location within a sector that is in the protected state, then the Block Erase command will not be executed, and the device will return to the idle state once the $\overline{CS}$ pin has been deasserted. In addition, with the larger Block Erase sizes of 32K and 64 Kbytes, more than one physical sector may be erased (e.g. sectors 18 through 15) at one time. Therefore, in order to erase a larger block that may span more than one sector, all of the sectors in the span must be in the unprotected state. If one of the physical sectors within the span is in the protected state, then the device will ignore the Block Erase command and will return to the idle state once the $\overline{CS}$ pin is 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 or because a memory location within the region to be erased is protected.

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

The device also incorporates an intelligent erasing 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 8-5. Block Erase



8.4 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.

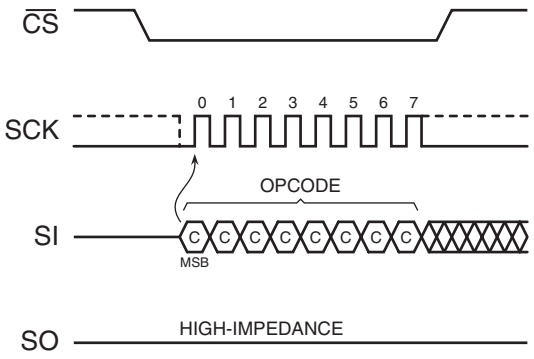
Two opcodes, 60h and C7h, can be used for the Chip Erase command. There is no difference in device functionality when utilizing the two opcodes, so they can be used interchangeably. To perform a Chip Erase, one of the two opcodes (60h or C7h) must be clocked into the device. Since the entire memory array 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 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 a sector is in the protected state.

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

The device also incorporates an intelligent erasing 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 8-6. Chip Erase



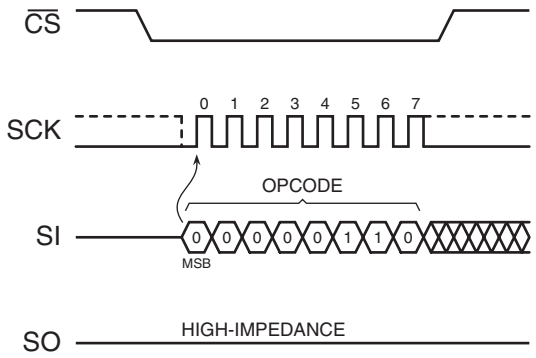
9. Protection Commands and Features

9.1 Write Enable

The Write Enable command is used to set the Write Enable Latch (WEL) bit in the Status Register to a logical “1” state. The WEL bit must be set before a program, erase, Protect Sector, Unprotect Sector, or Write Status Register command can be executed. This makes the issuance of these commands a two step process, thereby reducing the chances of a command being accidentally or erroneously executed. If the WEL bit in the Status Register is not set prior to the issuance of one of these commands, then the command will not be executed.

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

Figure 9-1. Write Enable

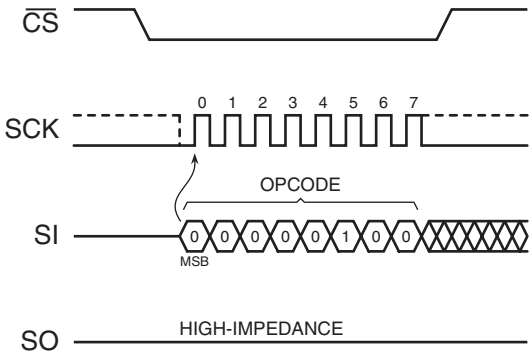


9.2 Write Disable

The Write Disable command is used to reset the Write Enable Latch (WEL) bit in the Status Register to the logical “0” state. With the WEL bit reset, all program, erase, Protect Sector, Unprotect Sector, and Write Status Register commands will not be executed. The Write Disable command is also used to exit the Sequential Program Mode. Other conditions can also cause the WEL bit to be reset; for more details, refer to the WEL bit section of the Status Register description.

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

Figure 9-2. Write Disable



9.3 Protect Sector

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

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

Table 9-1. Sector Protection Register Values

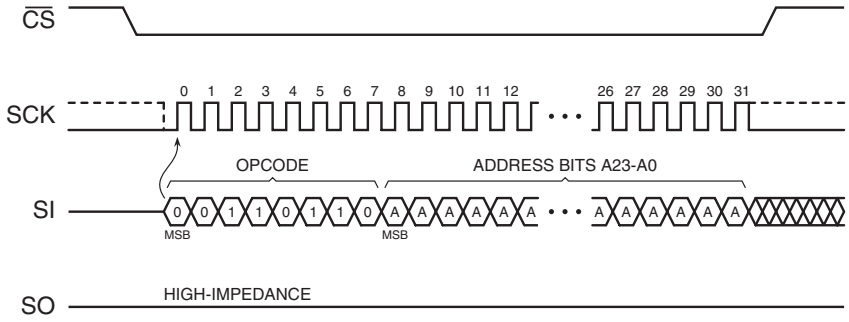
Value	Sector Protection Status
0	Sector is unprotected and can be programmed and erased.
1	Sector is protected and cannot be programmed or erased. This is the default state.

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

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

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

Figure 9-3. Protect Sector



9.4 Unprotect Sector

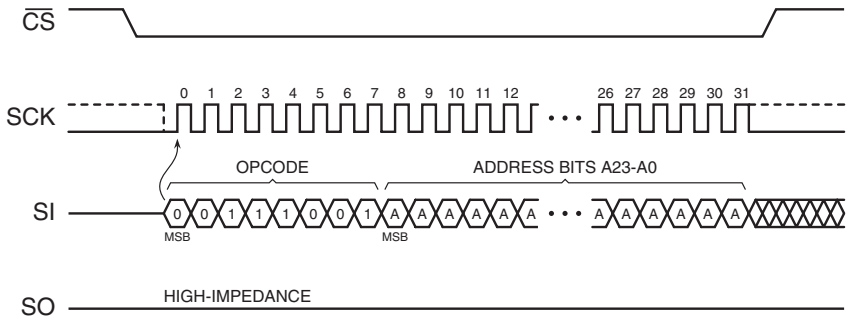
Issuing the Unprotect Sector command to a particular sector address will reset the corresponding Sector Protection Register to the logical “0” state (see Table 9-1 for Sector Protection Register values). Every physical sector of the device has a corresponding single-bit Sector Protection Register that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

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

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

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

Figure 9-4. Unprotect Sector



9.5 Read Sector Protection Registers

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

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

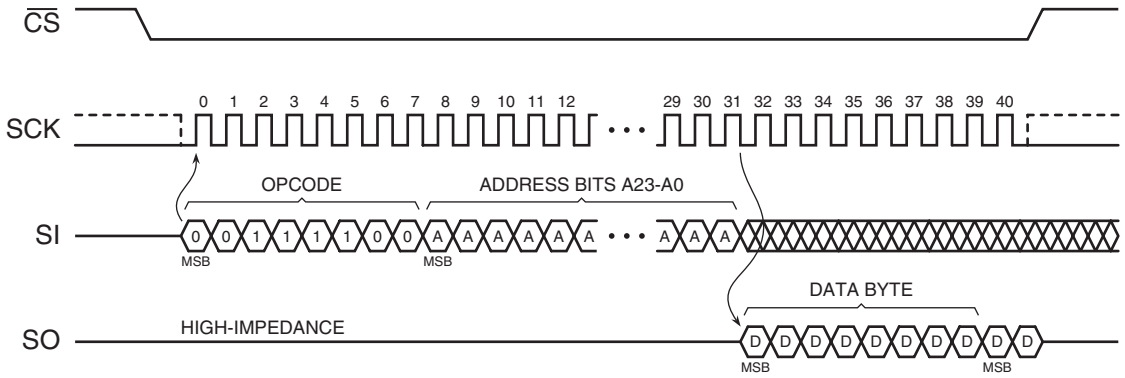
Table 9-2. Read Sector Protection Register – Output Data

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

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

In addition to reading the individual Sector Protection Registers, the Software Protection Status (SWP) bit in the Status Register can be read to determine if all, some, or none of the sectors are software protected (refer to the "Status Register Commands" on page 19 for more details).

Figure 9-5. Read Sector Protection Register



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

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

When hardware locking is active, the Sector Protection Registers are locked and the SPRL bit itself is also locked. Therefore, sectors that are protected will be locked in the protected state, and sectors that are unprotected will be locked in the unprotected state. These states cannot be changed as long as hardware locking is active, so the Protect Sector, Unprotect Sector, and Write Status Register commands will be ignored. In order to modify the protection status of a sector, the $\overline{WP}$ pin must first be deasserted, and the SPRL bit in the Status Register must be reset back to the logical "0" state.

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

When the $\overline{WP}$ pin is deasserted, or if the $\overline{WP}$ pin is permanently connected to VCC, the SPRL bit in the Status Register can still be set to a logical "1" to lock the Sector Protection Registers. This provides a software locking ability to prevent erroneous Protect Sector or Unprotect Sector commands from being processed.

Tables 9-3 and 9-4 detail the various protection and locking states of the device.

Table 9-3. Software Protection

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

Note: 1. "n" represents a sector number

Table 9-4. Hardware and Software Locking

$\overline{WP}$	SPRL	Locking	SPRL	Sector Protection Registers
0	0		Can be modified from 0 to 1	Unlocked and modifiable using the Protect and Unprotect Sector commands
0	1	Hardware Locked	Locked	Locked in current state. Protect and Unprotect Sector commands will be ignored.
1	0		Can be modified from 0 to 1	Unlocked and modifiable using the Protect and Unprotect Sector commands
1	1	Software Locked	Can be modified from 1 to 0	Locked in current state. Protect and Unprotect Sector commands will be ignored.

10. Status Register Commands

10.1 Read Status Register

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

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 last bit of the opcode has been clocked in, the device will begin outputting Status Register data on the SO pin during every subsequent clock cycle. After the last bit (bit 0) of the Status Register has been clocked out, the sequence will repeat itself starting again with bit 7 as long as the $\overline{CS}$ pin remains asserted and the SCK pin is being pulsed. The data in the Status Register is constantly being updated, so each repeating sequence will output new data.

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

Table 10-1. Status Register Format

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

Notes: 1. Bit 7 of the Status Register is the only bit that can be user modified.

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

10.1.1 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). 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 will be processed as normal). The SPRL bit defaults to the logical “0” state after a power-up or a device reset.

The SPRL bit can be modified freely whenever the $\overline{WP}$ 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 command, the $\overline{WP}$ pin will have to first be deasserted.

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

10.1.2 SPM Bit

The SPM bit indicates whether the device is in the Byte/Page Program mode or the Sequential Program Mode. The default state after power-up or device reset is the Byte/Page Program mode.

10.1.3 EPE Bit

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

10.1.4 WPP Bit

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

10.1.5 SWP Bits

The SWP bits provide feedback on the software protection status for the device. There are three possible combinations of the SWP bits that indicate whether none, some, or all of the sectors have been protected using the Protect Sector command. 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.

10.1.6 WEL Bit

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

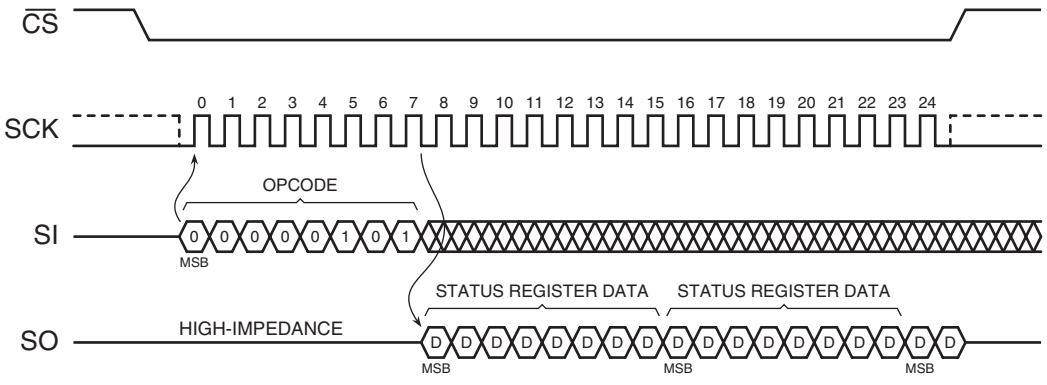
- Write Disable operation completes successfully
- Write Status Register operation completes successfully or aborts
- Protect Sector operation completes successfully or aborts
- Unprotect Sector operation completes successfully or aborts
- Byte/Page Program operation completes successfully or aborts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

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

10.1.7 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 10-1. Read Status Register



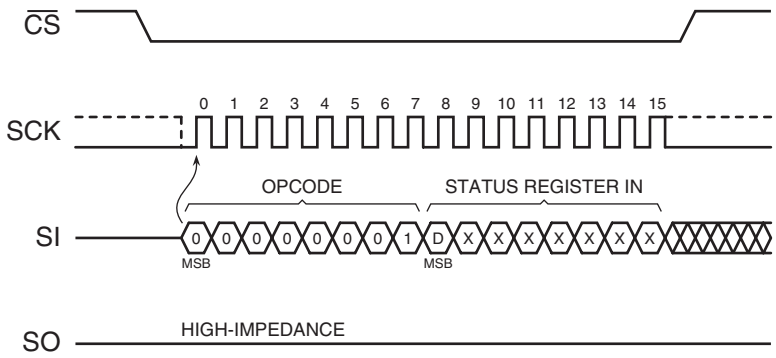
10.2 Write Status Register

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

To issue the Write Status Register command, the $\overline{CS}$ pin must first be asserted and the opcode of 01h must be clocked into the device. After the opcode has been clocked in, one byte of data comprised of the SPRL bit value and seven don't care bits must be clocked in. 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 complete one byte of data must be clocked into the device before the $\overline{CS}$ pin is deasserted; otherwise, the device will abort the operation, the state of the SPRL bit will not change, and the WEL bit in the Status Register will be reset back to the logical “0” state.

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

Figure 10-2. Write Status Register



11. Other Commands and Functions

11.1 Read Manufacturer and Device ID

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

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

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

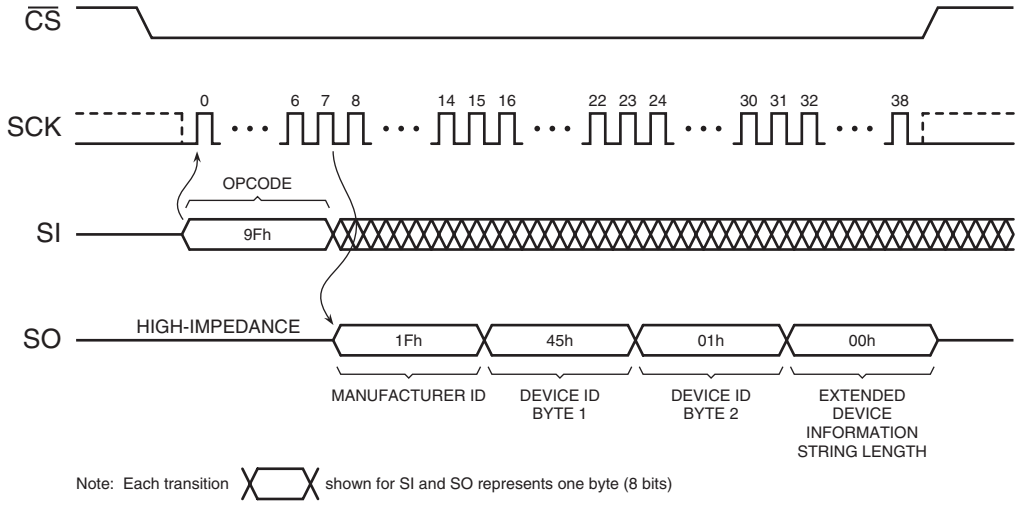
Table 11-1. Manufacturer and Device ID Information

Byte No.	Data Type	Value
1	Manufacturer ID	1Fh
2	Device ID (Part 1)	45h
3	Device ID (Part 2)	01h
4	Extended Device Information String Length	00h

Table 11-2. Manufacturer and Device ID Details

Data Type	Bit 7	Bit 6	Bit 5	Bit 5	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					45h	Family Code: 010 (AT26DFxxx series) Density Code: 00101 (8-Mbit)
	0	1	0	0	0	1	0	1		
Device ID (Part 2)	MLC Code			Product Version Code					01h	MLC Code: 000 (1-bit/cell technology) Product Version: 00001 (First major revision)
	0	0	0	0	0	0	0	1		

Figure 11-1. Read Manufacturer and Device ID



11.2 Deep Power-down

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

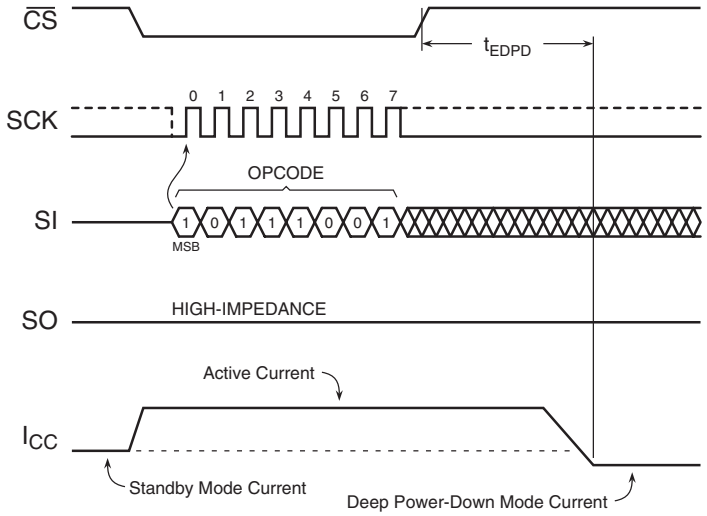
When the device is in the Deep Power-down mode, all commands including the Read Status Register command will be ignored with the exception of the Resume from Deep Power-down command. Since all commands will be ignored, the mode can be used as an extra protection mechanism against program and erase operations.

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

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

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

Figure 11-2. Deep Power-down



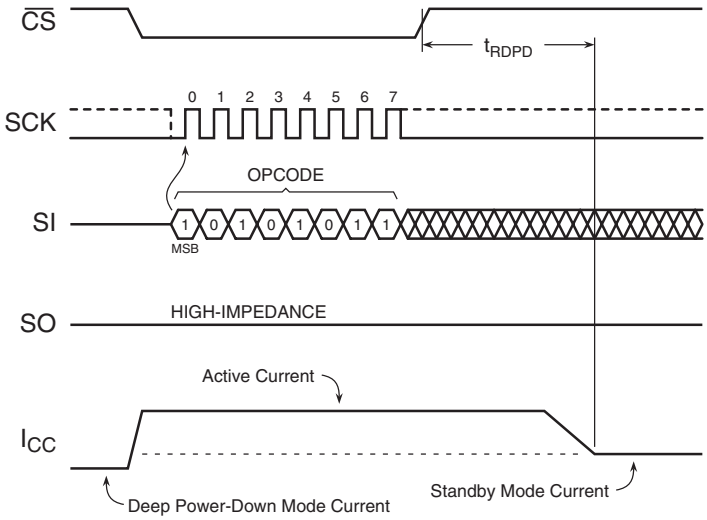
11.3 Resume from Deep Power-down

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

To resume from the Deep Power-down mode, the $\overline{CS}$ pin must first be asserted and 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 11-3. Resume from Deep Power-down



11.4 Hold

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

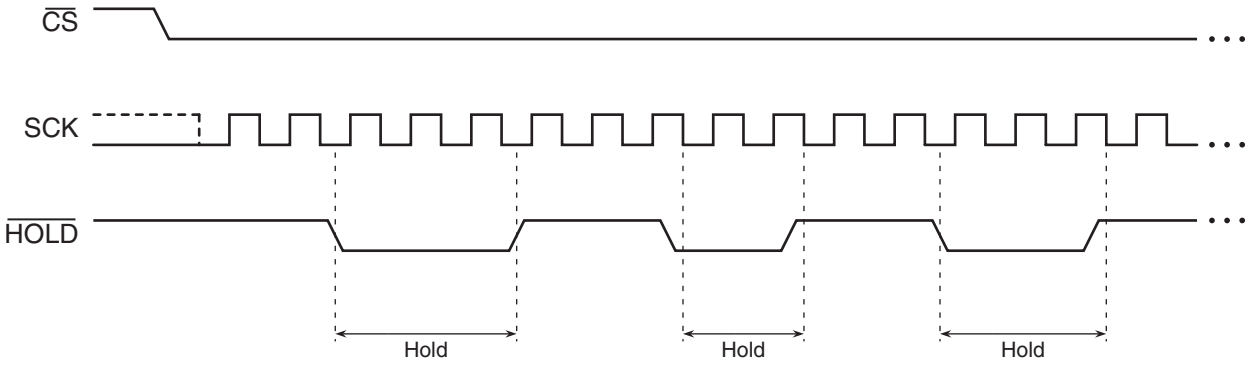
The Hold mode can only be entered while the $\overline{\text{CS}}$ pin is asserted. The Hold mode is activated simply by asserting the $\overline{\text{HOLD}}$ pin during the SCK low pulse. If the $\overline{\text{HOLD}}$ pin is asserted during the SCK high pulse, then the Hold mode won't be started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

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

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

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

Figure 11-4. Hold Mode



12. Electrical Specifications

12.1 Absolute Maximum Ratings*

Temperature under Bias	-55° C to +125° C
Storage Temperature	-65° C to +150° C
All Input Voltages (including NC Pins) with Respect to Ground	-0.6V to +4.1V
All Output Voltages with Respect to Ground	-0.6V to V_{CC} + 0.5V

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

12.2 DC and AC Operating Range

		AT26DF081A
Operating Temperature (Case)	Ind.	-40° C to 85° C
V_{CC} Power Supply		2.7V to 3.6V

12.3 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	35	µA
I_{DPD}	Deep Power-down Current	$\overline{CS}$, $\overline{WP}$, $\overline{HOLD}$ = V_{CC}, all inputs at CMOS levels		11	15	µA
I_{CC1}	Active Current, Read Operation	f = 70 MHz; I_{OUT} = 0 mA; $\overline{CS}$ = V_{IL}, V_{CC} = Max		12	16	mA
		f = 66 MHz; I_{OUT} = 0 mA; $\overline{CS}$ = V_{IL}, V_{CC} = Max		11	15	
		f = 50 MHz; I_{OUT} = 0 mA; $\overline{CS}$ = V_{IL}, V_{CC} = Max		10	14	
		f = 33 MHz; I_{OUT} = 0 mA; $\overline{CS}$ = V_{IL}, V_{CC} = Max		8	12	
		f = 20 MHz; I_{OUT} = 0 mA; $\overline{CS}$ = V_{IL}, V_{CC} = Max		7	10	
I_{CC2}	Active Current, Program Operation	$\overline{CS}$ = V_{CC}, V_{CC} = Max		12	18	mA
I_{CC3}	Active Current, Erase Operation	$\overline{CS}$ = V_{CC}, V_{CC} = Max		14	20	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} = 1.6 mA; V_{CC} = Min			0.4	V
V_{OH}	Output High Voltage	I_{OH} = -100 µA	V_{CC} - 0.2V			V

12.4 AC Characteristics

Symbol	Parameter	Min	Max	Units
f_{SCK}	Serial Clock (SCK) Frequency		70	MHz
f_{RDLF}	SCK Frequency for Read Array (Low Frequency - 03h opcode)		33	MHz
t_{SCKH}	SCK High Time	6.4		ns
t_{SCKL}	SCK Low Time	6.4		ns
t_{SCKR} (1)	SCK Rise Time, Peak-to-Peak (Slew Rate)	0.1		V/ns
t_{SCKF} (1)	SCK 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 SCK)	5		ns
t_{CSLH}	Chip Select Low Hold Time (relative to SCK)	5		ns
t_{CSHS}	Chip Select High Setup Time (relative to SCK)	5		ns
t_{CSHH}	Chip Select High Hold Time (relative to SCK)	5		ns
t_{DS}	Data In Setup Time	2		ns
t_{DH}	Data In Hold Time	3		ns
t_{DIS} (1)	Output Disable Time		6	ns
t_{V} (2)	Output Valid Time		6	ns
t_{OH}	Output Hold Time	0		ns
t_{HLS}	$\overline{\text{HOLD}}$ Low Setup Time (relative to SCK)	5		ns
t_{HLH}	$\overline{\text{HOLD}}$ Low Hold Time (relative to SCK)	5		ns
t_{HHS}	$\overline{\text{HOLD}}$ High Setup Time (relative to SCK)	5		ns
t_{HHH}	$\overline{\text{HOLD}}$ High Hold Time (relative to SCK)	5		ns
t_{HLQZ} (1)	$\overline{\text{HOLD}}$ Low to Output High-Z		6	ns
t_{HHQX} (1)	$\overline{\text{HOLD}}$ High to Output Low-Z		6	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_{EDPD} (1)	Chip Select High to Deep Power-down		3	µs
t_{RDPD} (1)	Chip Select High to Standby Mode		3	µs

Notes: 1. Not 100% tested (value guaranteed by design and characterization).
2. 15 pF load at 70 MHz, 30 pF load at 66 MHz.
3. Only applicable as a constraint for the Write Status Register command when SPRL = 1

12.5 Program and Erase Characteristics

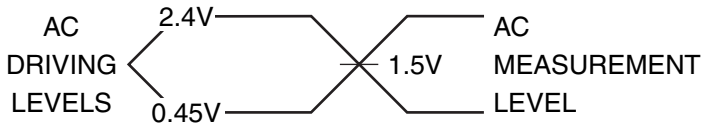
Symbol	Parameter		Min	Typ	Max	Units
t_{PP}	Page Program Time (256 Bytes)			1.5	3.0	ms
t_{BP}	Byte Program Time			6		µs
t_{BLKE}	Block Erase Time	4 Kbytes		0.05	0.2	sec
		32 Kbytes		0.35	0.6	
		64 Kbytes		0.7	1.0	
t_{CHPE}	Chip Erase Time			10	14	sec
t_{WRSR}[1]	Write Status Register Time				200	ns

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

12.6 Power-up Conditions

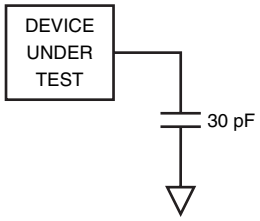
Parameter	Min	Max	Units
Minimum V_{CC} to Chip Select Low Time	50		µs
Power-up Device Delay Before Program or Erase Allowed		10	ms
Power-on Reset Voltage	1.5	2.5	V

12.7 Input Test Waveforms and Measurement Levels



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

12.8 Output Test Load



13. Waveforms

Figure 13-1. Serial Input Timing

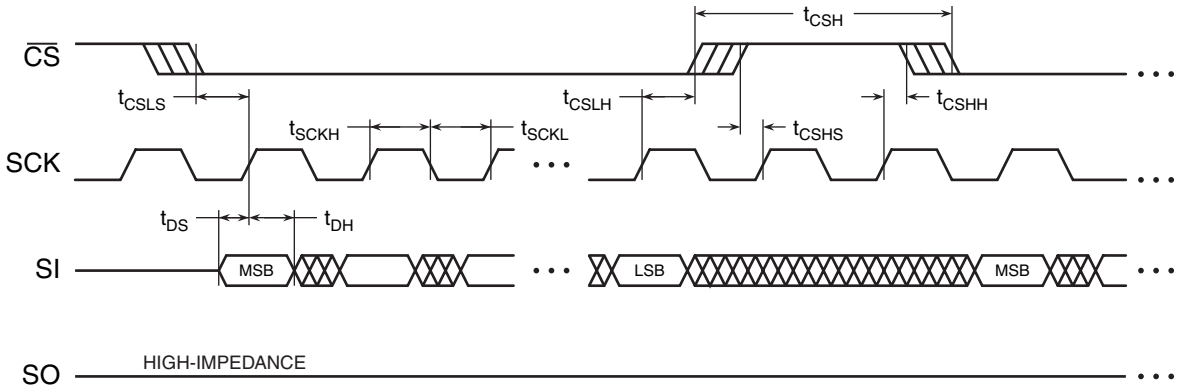


Figure 13-2. Serial Output Timing

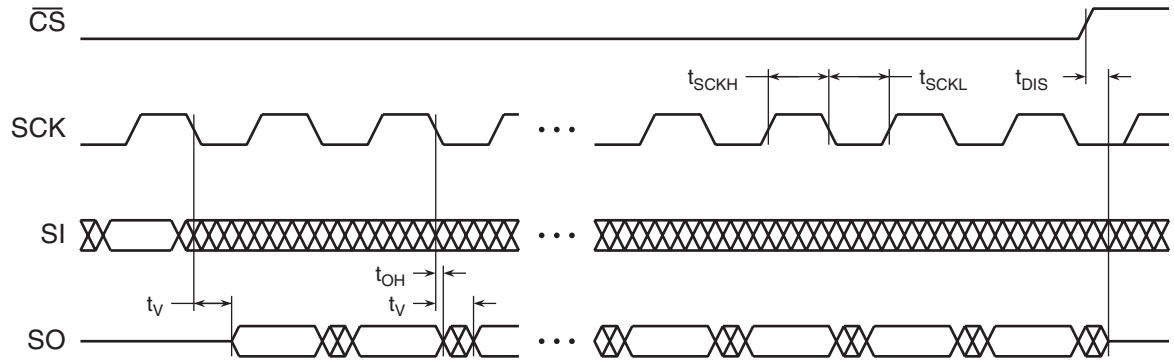


Figure 13-3. $\overline{HOLD}$ Timing – Serial Input

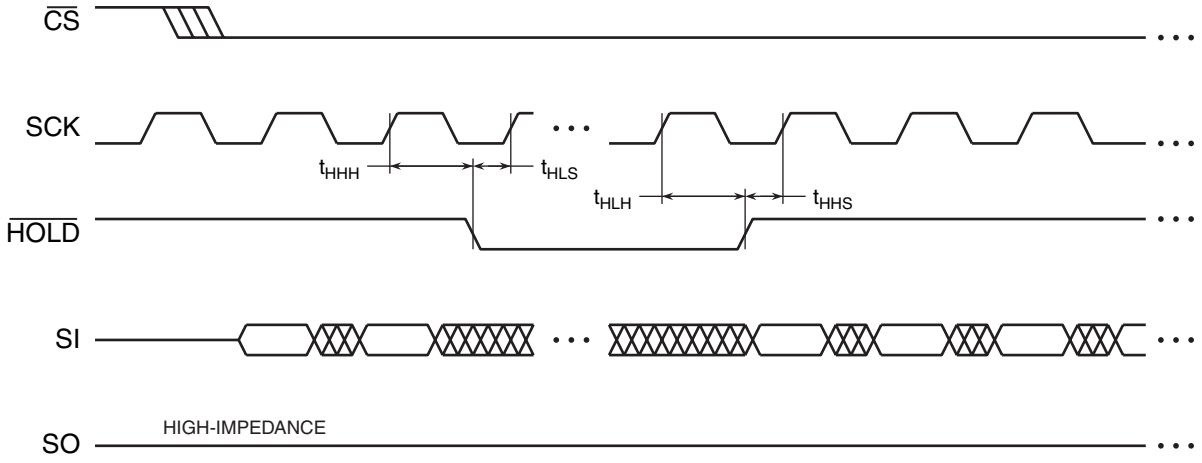


Figure 13-4. $\overline{\text{HOLD}}$ Timing – Serial Output

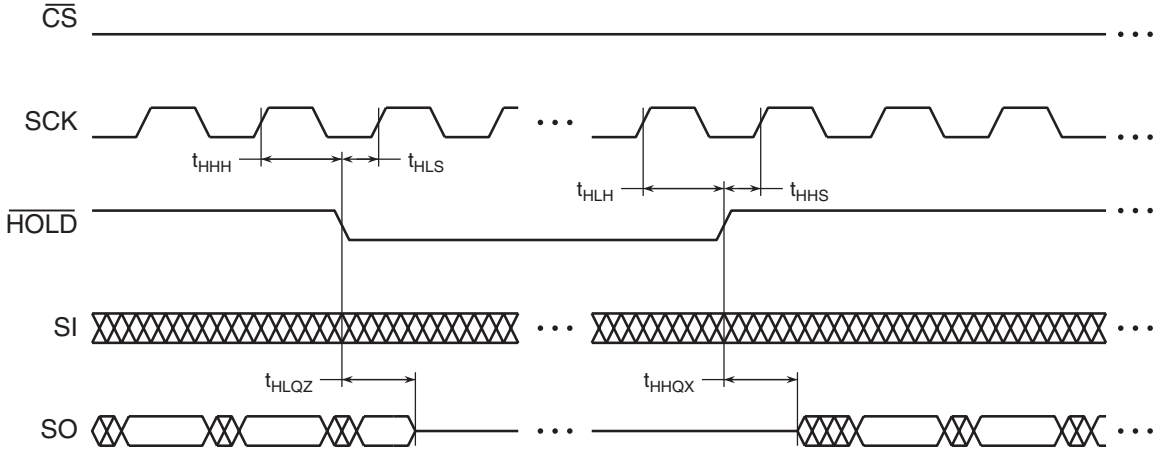
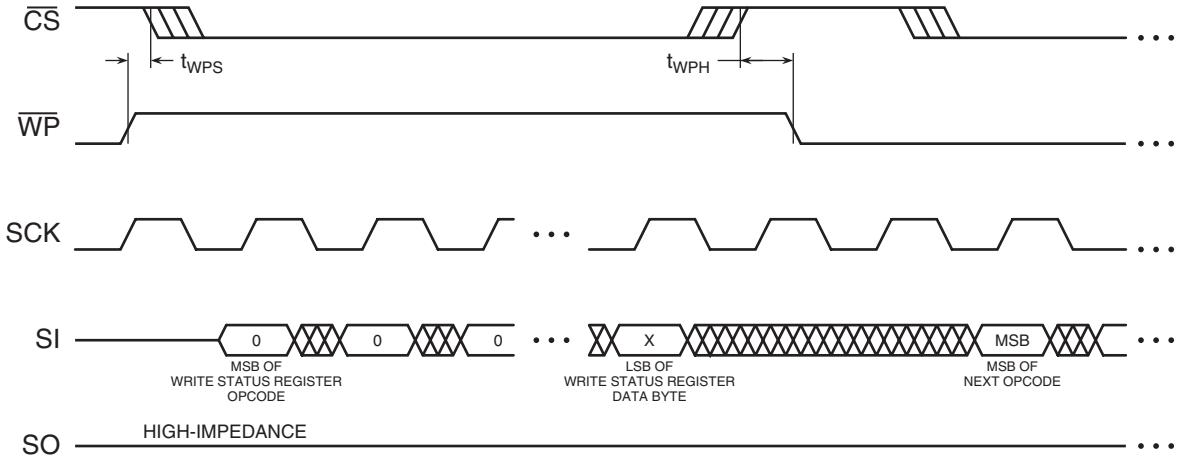


Figure 13-5. $\overline{\text{WP}}$ Timing for Write Status Register Command When SPRL = 1



14. Ordering Information

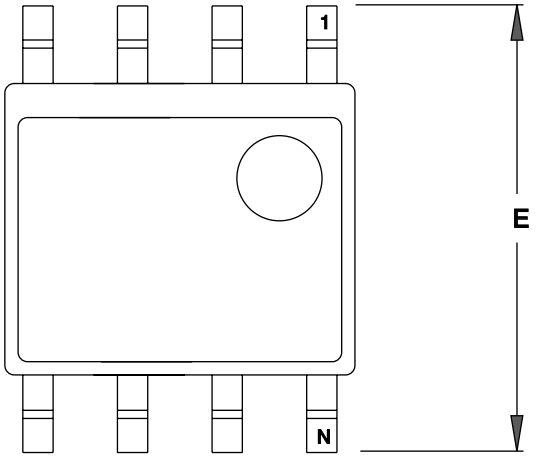
14.1 Green Package Options (Pb/Halide-free/RoHS Compliant)

f_{SCK} (MHz)	Ordering Code	Package	Operation Range
70	AT26DF081A-SSU	8S1	Industrial (-40° C to 85° C)
	AT26DF081A-SU	8S2	

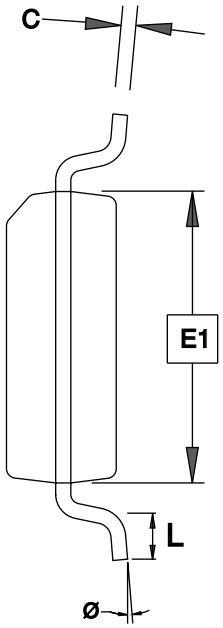
Package Type	
8S1	8-lead, 0.150” Wide, Plastic Gull Wing Small Outline Package (JEDEC SOIC)
8S2	8-lead, 0.209” Wide, Plastic Gull Wing Small Outline Package (EIAJ SOIC)

15. Packaging Information

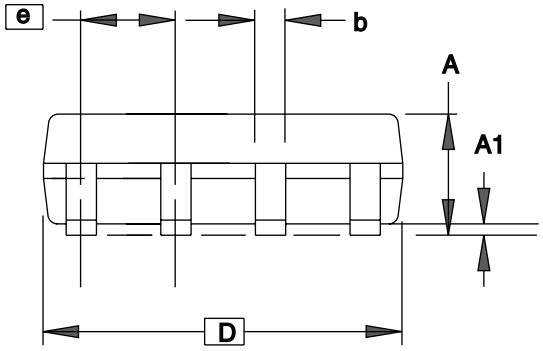
15.1 8S1 – EIAJ SOIC



TOP VIEW



END VIEW



SIDE VIEW

COMMON DIMENSIONS
(Unit of Measure = mm)

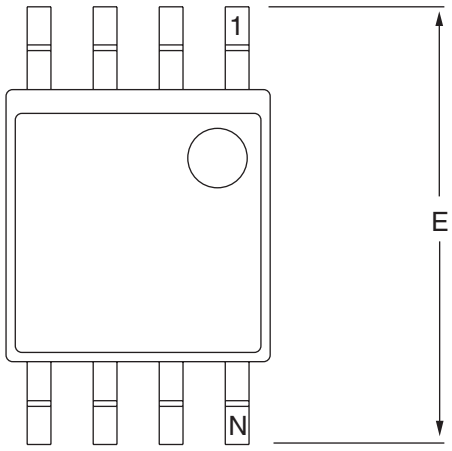
SYMBOL	MIN	NOM	MAX	NOTE
A	1.35	–	1.75	
A1	0.10	–	0.25	
b	0.31	–	0.51	
C	0.17	–	0.25	
D	4.80	–	5.05	
E1	3.81	–	3.99	
E	5.79	–	6.20	
e	1.27 BSC			
L	0.40	–	1.27	
θ	0°	–	8°	

Note: These drawings are for general information only. Refer to JEDEC Drawing MS-012, Variation AA for proper dimensions, tolerances, datums, etc.

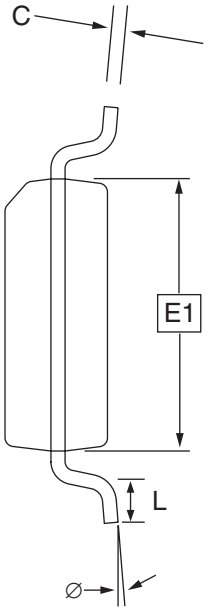
3/17/05

	TITLE	DRAWING NO.	REV.
1150 E. Cheyenne Mtn. Blvd. Colorado Springs, CO 80906	**8S1**, 8-lead (0.150" Wide Body), Plastic Gull Wing Small Outline (JEDEC SOIC)	8S1	C

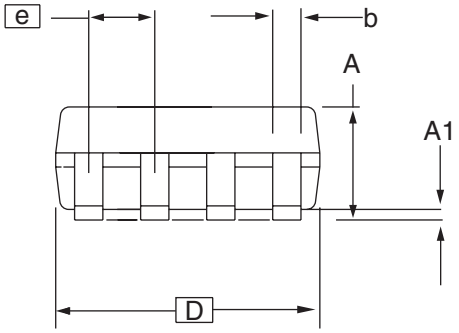
15.2 8S2 – EIAJ SOIC



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	5
C	0.15		0.35	5
D	5.13		5.35	
E1	5.18		5.40	2, 3
E	7.70		8.26	
L	0.51		0.85	
∅	0°		8°	
e	1.27 BSC			4

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 are not included.
3. It is recommended that upper and lower cavities be equal. If they are different, the larger dimension shall be regarded.
4. Determines the true geometric position.
5. Values b and C apply to pb/Sn solder plated terminal. The standard thickness of the solder layer shall be 0.010 +0.010/–0.005 mm.

10/7/03

2325 Orchard Parkway San Jose, CA 95131	**TITLE** **8S2**, 8-lead, 0.209" Body, Plastic Small Outline Package (EIAJ)	**DRAWING NO.** 8S2	**REV.** C

16. Revision History

Version No./Release Date	History
Revision A – November 2005	• Initial Release

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

3600A–DFLASH–11/05