Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

AT45DB021E

2-Mbit DataFlash (with Extra 64-Kbits), 1.65V Minimum SPI Serial Flash Memory

Features

- Single 1.65V - 3.6V supply
- Serial Peripheral Interface (SPI) compatible
  - Supports SPI modes 0 and 3
  - Supports RapidS™ operation
- Continuous read capability through entire array
  - Up to 85MHz
  - Low-power read option up to 15MHz
  - Clock-to-output time (t_V) of 6ns maximum
- User configurable page size
  - 256 bytes per page
  - 264 bytes per page (default)
  - Page size can be factory pre-configured for 256 bytes
- One SRAM data buffer (256/264 bytes)
- Flexible programming options
  - Byte/Page Program (1 to 256/264 bytes) directly into main memory
  - Buffer Write
  - Buffer to Main Memory Page Program
- Flexible erase options
  - Page Erase (256/264 bytes)
  - Block Erase (2KB)
  - Sector Erase (32KB)
  - Chip Erase (2-Mbits)
- Advanced hardware and software data protection features
  - Individual sector protection
  - Individual sector lockdown to make any sector permanently read-only
- 128-byte, One-Time Programmable (OTP) Security Register
  - 64 bytes factory programmed with a unique identifier
  - 64 bytes user programmable
- Hardware and software controlled reset options
- JEDEC Standard Manufacturer and Device ID Read
- Low-power dissipation
  - 200nA Ultra-Deep Power-Down current (typical)
  - 3µA Deep Power-Down current (typical)
  - 25µA Standby current (typical at 20MHz)
  - 4.5mA Active Read current (typical)
- Endurance: 100,000 program/erase cycles per page minimum
- Data retention: 20 years
- Complies with full industrial temperature range
- Green (Pb/Halide-free/RoHS compliant) packaging options
  - 8-lead SOIC (0.150" wide and 0.208" wide)
  - 8-pad Ultra-thin DFN (5 x 6 x 0.6mm)
  - 8-ball (2 x4 Array) Wafer Level Chip Scale Package
  - Die in Wafer Form

Description

The Adesto® AT45DB021E is a 1.65V minimum, serial-interface sequential access Flash memory is ideally suited for a wide variety of digital voice, image, program code, and data storage applications. The AT45DB021E also supports the RapidS serial interface for applications requiring very high speed operation. Its 2,162,688 bits of memory are organized as 1,024 pages of 256 bytes or 264 bytes each. In addition to the main memory, AT45DB021E also contains one SRAM buffer of 256/264 bytes. The Buffer can be used as additional system scratch memory, and E^2PROM emulation (bit or byte alterability) can be easily handled with a self-contained three step read-modify-write operation.

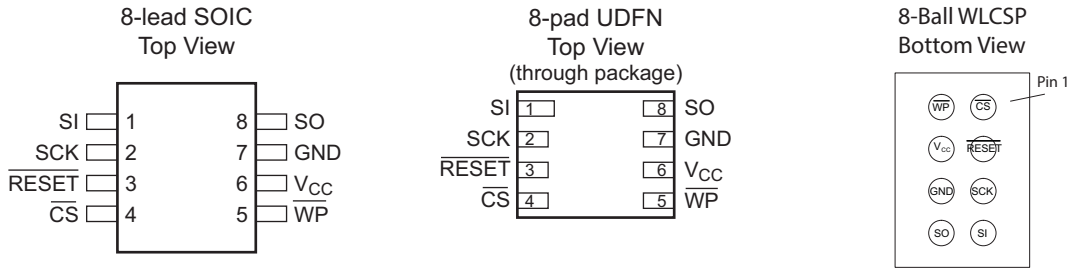
Unlike conventional Flash memories that are accessed randomly with multiple address lines and a parallel interface, the Adesto DataFlash® uses a serial interface to sequentially access its data. The simple sequential access dramatically reduces active pin count, facilitates simplified hardware layout, increases system reliability, minimizes switching noise, and reduces package size. The device is optimized for use in many commercial and industrial applications where high-density, low-pin count, low-voltage, and low-power are essential.

To allow for simple in-system re-programmability, AT45DB021E does not require high input voltages for programming. The device operates from a single 1.65V to 3.6V power supply for the erase and program and read operations. The AT45DB021E is enabled through the Chip Select pin ($\overline{CS}$) and accessed via a 3-wire interface consisting of the Serial Input (SI), Serial Output (SO), and the Serial Clock (SCK).

All programming and erase cycles are self-timed.

1. Pin Configurations and Pinouts

Figure 1-1. Pinouts



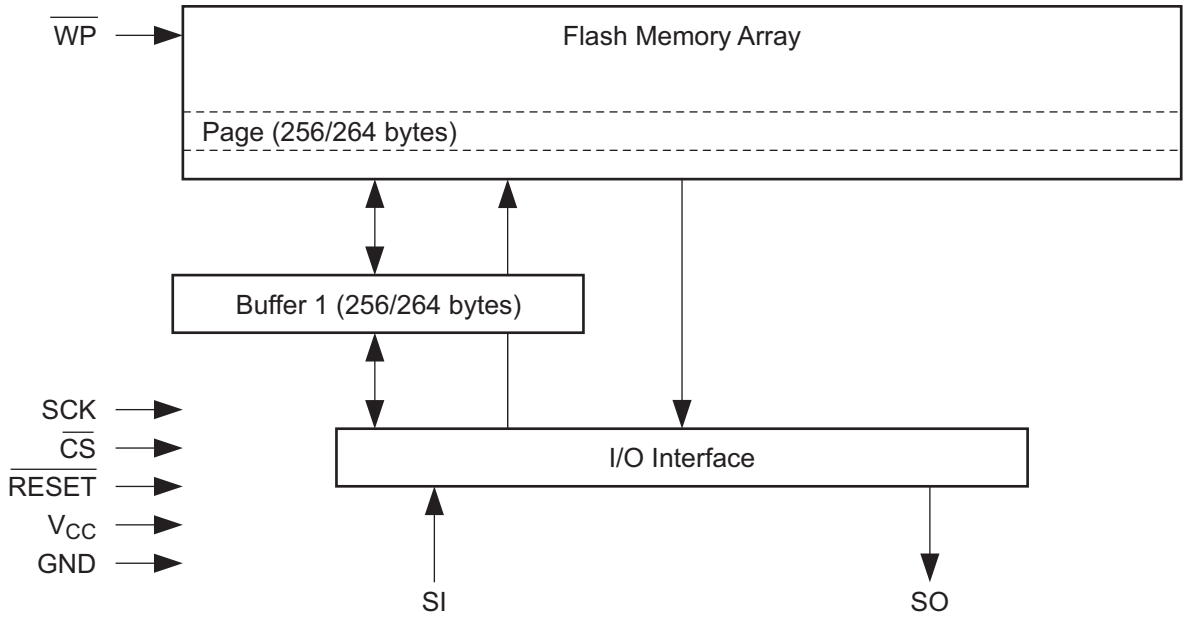
Note: 1. The metal pad on the bottom of the UDFN package is not internally connected to a voltage potential. This pad can be a “no connect” or connected to GND.

Table 1-1. Pin Configurations

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 the standby mode (not Deep Power-Down mode) and the output pin (SO) will be in a high-impedance state. When the device is deselected, data will not be accepted on the input pin (SI). 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. Data present on the SI pin will be ignored whenever the device is deselected ($\overline{CS}$ is deasserted).	—	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. The SO pin will be in a high-impedance state whenever the device is deselected ($\overline{CS}$ is deasserted).	—	Output
$\overline{WP}$	**Write Protect:** When the $\overline{WP}$ pin is asserted, all sectors specified for protection by the Sector Protection Register will be protected against program and erase operations regardless of whether the Enable Sector Protection command has been issued or not. The $\overline{WP}$ pin functions independently of the software controlled protection method. After the $\overline{WP}$ pin goes low, the contents of the Sector Protection Register cannot be modified. If a program or erase command is issued to the device while the $\overline{WP}$ pin is asserted, the device will simply ignore the command and perform no operation. The device will return to the idle state once the $\overline{CS}$ pin has been deasserted. The Enable Sector Protection command and the Sector Lockdown command will be recognized by the device when the $\overline{WP}$ pin is asserted. 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
RESET	**Reset:** A low state on the reset pin ($\overline{RESET}$) will terminate the operation in progress and reset the internal state machine to an idle state. The device will remain in the reset condition as long as a low level is present on the $\overline{RESET}$ pin. Normal operation can resume once the $\overline{RESET}$ pin is brought back to a high level. The device incorporates an internal power-on reset circuit, so there are no restrictions on the $\overline{RESET}$ pin during power-on sequences. If this pin and feature is not utilized, then it is recommended that the $\overline{RESET}$ pin be driven high externally.	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.	—	Ground

2. Block Diagram

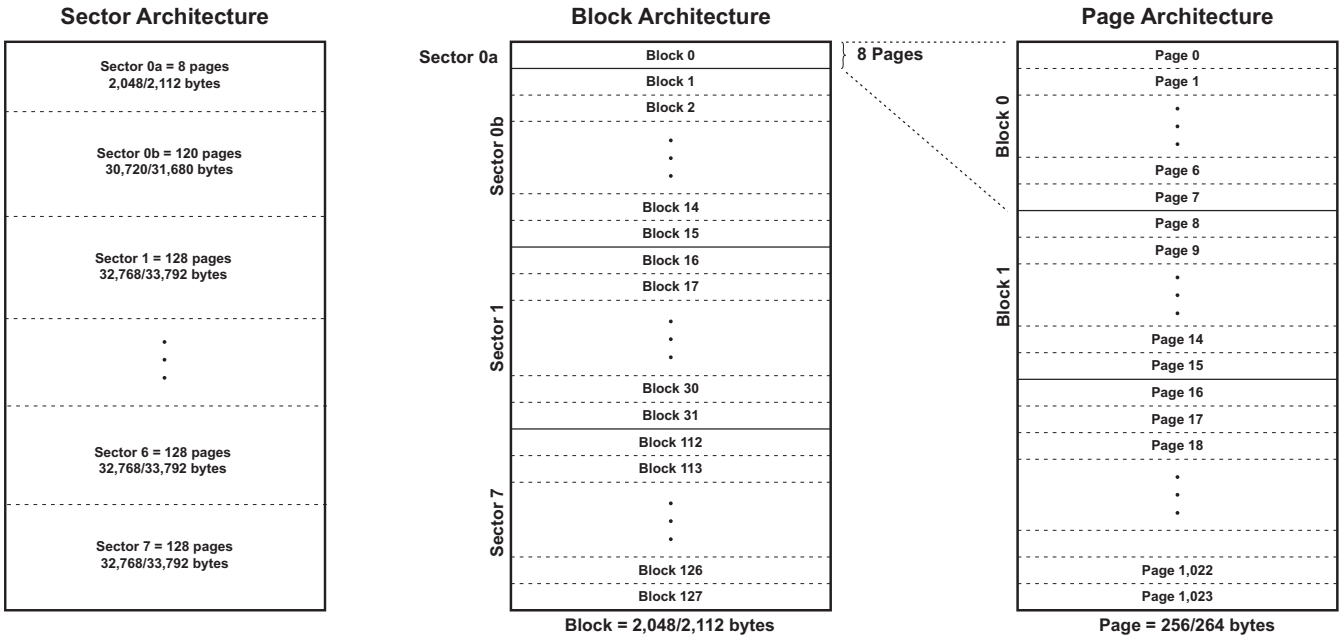
Figure 2-1. Block Diagram



3. Memory Array

To provide optimal flexibility, the AT45DB021E memory array is divided into three levels of granularity comprising of sectors, blocks, and pages. Figure 3-1, Memory Architecture Diagram illustrates the breakdown of each level and details the number of pages per sector and block. Program operations to the DataFlash can be done at the full page level or at the byte level (a variable number of bytes). The erase operations can be performed at the chip, sector, block, or page level.

Figure 3-1. Memory Architecture Diagram



4. Device Operation

The device operation is controlled by instructions from the host processor. The list of instructions and their associated opcodes are contained in Table 15-1 on page 38 through Table 15-4 on page 39. A valid instruction starts with the falling edge of $\overline{CS}$ followed by the appropriate 8-bit opcode and the Buffer or main memory address location. While the $\overline{CS}$ pin is low, toggling the SCK pin controls the loading of the opcode and the Buffer or main memory address location through the SI (Serial Input) pin. All instructions, addresses, and data are transferred with the Most Significant Bit (MSB) first.

Three address bytes are used to address memory locations in either the main memory array or in the Buffer. The three address bytes will be comprised of a number of dummy bits and a number of actual device address bits, with the number of dummy bits varying depending on the operation being performed and the selected device page size. Buffer addressing for the standard DataFlash page size (264 bytes) is referenced in the datasheet using the terminology BFA8 - BFA0 to denote the nine address bits required to designate a byte address within the Buffer. The main memory addressing is referenced using the terminology PA9 - PA0 and BA8 - BA0, where PA9 - PA0 denotes the 10 address bits required to designate a page address, and BA8 - BA0 denotes the nine address bits required to designate a byte address within the page. Therefore, when using the standard DataFlash page size, a total of 22 address bits are used.

For the “power of 2” binary page size (256 bytes), the Buffer addressing is referenced in the datasheet using the conventional terminology BFA7 - BFA0 to denote the eight address bits required to designate a byte address within the Buffer. Main memory addressing is referenced using the terminology A17 - A0, where A17 - A8 denotes the 10 address bits required to designate a page address, and A7 - A0 denotes the eight address bits required to designate a byte address within a page. Therefore, when using the binary page size, a total of 21 address bits are used.

5. Read Commands

By specifying the appropriate opcode, data can be read from the main memory or from the data buffer. The DataFlash supports RapidS protocols for Mode 0 and Mode 3. Please see Section 25., "Detailed Bit-level Read Waveforms: RapidS Mode 0/Mode 3" on page 55 for diagrams detailing the clock cycle sequences for each mode.

5.1 Continuous Array Read (Legacy Command: E8h)

By supplying an initial starting address for the main memory array, the Continuous Array Read command can be utilized to sequentially read a continuous stream of data from the device by simply providing a clock signal; no additional addressing information or control signals need to be provided. The DataFlash incorporates an internal address counter that will automatically increment on every clock cycle, allowing one continuous read from memory to be performed without the need for additional address sequences. To perform a Continuous Array Read using the standard DataFlash page size (264-bytes), an opcode of E8h must be clocked into the device followed by three address bytes (which comprise the 19-bit page and byte address sequence) and four dummy bytes. The first 10 bits (PA9 - PA0) of the 19-bit address sequence specify which page of the main memory array to read and the last nine bits (BA8 - BA0) of the 19-bit address sequence specify the starting byte address within the page. To perform a Continuous Array Read using the binary page size (256 bytes), the opcode E8h must be clocked into the device followed by three address bytes (A17 - A0) and four dummy bytes. The dummy bytes that follow the address bytes are needed to initialize the read operation. Following the dummy bytes, additional clock pulses on the SCK pin will result in data being output on the SO (Serial Output) pin.

The $\overline{CS}$ pin must remain low during the loading of the opcode, the address bytes, the dummy bytes and the reading of data. When the end of a page in main memory is reached during a Continuous Array Read, the device will continue reading at the beginning of the next page with no delays incurred during the page boundary crossover (the crossover from the end of one page to the beginning of the next page). When the last bit in the main memory array has been read, the device will continue reading back at the beginning of the first page of memory. As with crossing over page boundaries, no delays will be incurred when wrapping around from the end of the array to the beginning of the array.

A low-to-high transition on the $\overline{CS}$ pin will terminate the read operation and tri-state the output pin (SO). The maximum SCK frequency allowable for the Continuous Array Read is defined by the f_{CAR1} specification. The Continuous Array Read bypasses the data buffer and leaves the contents of the Buffer unchanged.

Note: This command is not recommended for new designs.

5.2 Continuous Array Read (High Frequency Mode: 0Bh Opcode)

This command can be used to read the main memory array sequentially at the highest possible operating clock frequency up to the maximum specified by f_{CAR1}. To perform a Continuous Array Read using the standard DataFlash page size (264 bytes), the $\overline{CS}$ pin must first be asserted, and then an opcode of 0Bh must be clocked into the device followed by three address bytes and one dummy byte. The first 10 bits (PA9 - PA0) of the 19-bit address sequence specify which page of the main memory array to read and the last nine bits (BA8 - BA0) of the 19-bit address sequence specify the starting byte address within the page. To perform a Continuous Array Read using the binary page size (256 bytes), the opcode 0Bh must be clocked into the device followed by three address bytes (A17 - A0) and one dummy byte. Following the dummy byte, additional clock pulses on the SCK pin will result in data being output on the SO pin.

The $\overline{CS}$ pin must remain low during the loading of the opcode, the address bytes, the dummy byte, and the reading of data. When the end of a page in the main memory is reached during a Continuous Array Read, the device will continue reading at the beginning of the next page with no delays incurred during the page boundary crossover (the crossover from the end of one page to the beginning of the next page). When the last bit in the main memory array has been read, the device will continue reading back at the beginning of the first page of memory. As with crossing over page boundaries, no delays will be incurred when wrapping around from the end of the array to the beginning of the array.

A low-to-high transition on the $\overline{CS}$ pin will terminate the read operation and tri-state the output pin (SO). The maximum SCK frequency allowable for the Continuous Array Read is defined by the f_{CAR1} specification. The Continuous Array Read bypasses the data buffer and leaves the contents of the Buffer unchanged.

5.3 Continuous Array Read (Low Frequency Mode: 03h Opcode)

This command can be used to read the main memory array sequentially at lower clock frequencies up to maximum specified by f_{CAR2}. Unlike the previously described read commands, this Continuous Array Read command for lower clock frequencies does not require the clocking in of dummy bytes after the address byte sequence. To perform a Continuous Array Read using the standard DataFlash page size (264 bytes), the $\overline{CS}$ pin must first be asserted, and then an opcode of 03h must be clocked into the device followed by three address bytes (which comprise the 24-bit page and byte address sequence). The first 10 bits (PA9 - PA0) of the 19-bit address sequence specify which page of the main memory array to read, and the last nine bits (BA8 - BA0) of the 19-bit address sequence specify the starting byte address within the page. To perform a Continuous Array Read using the binary page size (256 bytes), the opcode 03h must be clocked into the device followed by three address bytes (A17 - A0). Following the address bytes, additional clock pulses on the SCK pin will result in data being output on the SO pin.

The $\overline{CS}$ pin must remain low during the loading of the opcode, the address bytes, and the reading of data. When the end of a page in the main memory is reached during a Continuous Array Read, the device will continue reading at the beginning of the next page with no delays incurred during the page boundary crossover (the crossover from the end of one page to the beginning of the next page). When the last bit in the main memory array has been read, the device will continue reading back at the beginning of the first page of memory. As with crossing over page boundaries, no delays will be incurred when wrapping around from the end of the array to the beginning of the array.

A low-to-high transition on the $\overline{CS}$ pin will terminate the read operation and tri-state the output pin (SO). The maximum SCK frequency allowable for the Continuous Array Read is defined by the f_{CAR2} specification. The Continuous Array Read bypasses the data buffer and leaves the contents of the Buffer unchanged.

5.4 Continuous Array Read (Low Power Mode: 01h Opcode)

This command is ideal for applications that want to minimize power consumption and do not need to read the memory array at high frequencies. Like the 03h opcode, this Continuous Array Read command allows reading the main memory array sequentially without the need for dummy bytes to be clocked in after the address byte sequence. The memory can be read at clock frequencies up to maximum specified by f_{CAR3}. To perform a Continuous Array Read using the standard DataFlash page size (264 bytes), the $\overline{CS}$ pin must first be asserted, and then an opcode of 01h must be clocked into the device followed by three address bytes (which comprise the 24-bit page and byte address sequence). The first 10 bits (PA9 - PA0) of the 19-bit address sequence specify which page of the main memory array to read and the last nine bits (BA8 - BA0) of the 19-bit address sequence specify the starting byte address within the page. To perform a Continuous Array Read using the binary page size (256 bytes), the opcode 01h must be clocked into the device followed by three address bytes (A17 - A0). Following the address bytes, additional clock pulses on the SCK pin will result in data being output on the SO pin.

The $\overline{CS}$ pin must remain low during the loading of the opcode, the address bytes, and the reading of data. When the end of a page in the main memory is reached during a Continuous Array Read, the device will continue reading at the beginning of the next page with no delays incurred during the page boundary crossover (the crossover from the end of one page to the beginning of the next page). When the last bit in the main memory array has been read, the device will continue reading back at the beginning of the first page of memory. As with crossing over page boundaries, no delays will be incurred when wrapping around from the end of the array to the beginning of the array.

A low-to-high transition on the $\overline{CS}$ pin will terminate the read operation and tri-state the output pin (SO). The maximum SCK frequency allowable for the Continuous Array Read is defined by the f_{CAR3} specification. The Continuous Array Read bypasses the data buffer and leaves the contents of the Buffer unchanged.

5.5 Main Memory Page Read

A Main Memory Page Read allows the user to read data directly from any one of the 1,024 pages in the main memory, bypassing the data buffer and leaving the contents of the Buffer unchanged. To start a Main Memory Page Read using the standard DataFlash page size (264 bytes), the $\overline{CS}$ pin must first be asserted then an opcode of D2h must be clocked into the device followed by three address bytes (which comprise the 24-bit page and byte address sequence) and four dummy bytes. The first 10 bits (PA9 - PA0) of the 19-bit address sequence specify which page of the main memory array to read, and the last nine bits (BA8 - BA0) of the 19-bit address sequence specify the starting byte address within the page. To perform a Main Memory Page Read with the binary page size (256 bytes), the opcode D2h must be clocked into the device followed by three address bytes (A17 - A0) and four dummy bytes. The first 10 bits (A17 - A8) of the 18-bit address sequence specify which page of the main memory array to read, and the last eight bits (A7 - A0) of the 18-bit address sequence specify the starting byte address within that page. The dummy bytes that follow the address bytes are sent to initialize the read operation. Following the dummy bytes, the additional pulses on SCK result in data being output on the SO (Serial Output) pin.

The $\overline{CS}$ pin must remain low during the loading of the opcode, the address bytes, the dummy bytes, and the reading of data. Unlike the Continuous Array Read command, when the end of a page in main memory is reached, the device will continue reading back at the beginning of the same page rather than the beginning of the next page.

A low-to-high transition on the $\overline{CS}$ pin will terminate the read operation and tri-state the output pin (SO). The maximum SCK frequency allowable for the Main Memory Page Read is defined by the f_{SCK} specification. The Main Memory Page Read bypasses the data buffer and leaves the contents of the Buffer unchanged.

5.6 Buffer Read

The data buffer can be accessed independently from the main memory array, and utilizing the Buffer Read command allows data to be sequentially read directly from the Buffer. Two opcodes, D4h or D1h, can be used for the Buffer Read command. The use of each opcode depends on the maximum SCK frequency that will be used to read data from the Buffer. The D4h opcode can be used at any SCK frequency up to the maximum specified by f_{CAR} while the D1h opcode can be used for lower frequency read operations up to the maximum specified by f_{CAR2}.

To perform a Buffer Read using the standard DataFlash buffer size (264 bytes), the opcode must be clocked into the device followed by three address bytes comprised of 15 dummy bits and nine buffer address bits (BFA8 -BFA0). To perform a Buffer Read using the binary buffer size (256 bytes), the opcode must be clocked into the device followed by three address bytes comprised of 16 dummy bits and eight address bits (A7 - A0). Following the address bytes, one dummy byte must be clocked into the device to initialize the read operation if using opcode D4h. The $\overline{CS}$ must remain low during the loading of the opcode, the address bytes, the dummy byte (for opcode D4h only), and the reading of data. When the end of a buffer is reached, the device will continue reading back at the beginning of the Buffer. A low-to-high transition on the $\overline{CS}$ pin will terminate the read operation and tri-state the output pin (SO).

6. Program and Erase Commands

6.1 Buffer Write

Utilizing the Buffer Write command allows data clocked in from the SI pin to be written directly into the data buffer.

To load data into the Buffer using the standard DataFlash buffer size (264 bytes), an opcode of 84h must be clocked into the device followed by three address bytes comprised of 15 dummy bits and nine buffer address bits (BFA8 - BFA0). The nine buffer address bits specify the first byte in the Buffer to be written.

To load data into the Buffer using the binary buffer size (256 bytes), an opcode of 84h must be clocked into the device followed by 16 dummy bits and eight address bits (A7 - A0). The eight address bits specify the first byte in the Buffer to be written.

After the last address byte has been clocked into the device, data can then be clocked in on subsequent clock cycles. If the end of the data buffer is reached, the device will wrap around back to the beginning of the Buffer. Data will continue to be loaded into the Buffer until a low-to-high transition is detected on the $\overline{CS}$ pin.

6.2 Buffer to Main Memory Page Program with Built-In Erase

The Buffer to Main Memory Page Program with Built-In Erase command allows data that is stored in the Buffer to be written into an erased or programmed page in the main memory array. It is not necessary to pre-erase the page in main memory to be written because this command will automatically erase the selected page prior to the program cycle.

To perform a Buffer to Main Memory Page Program with Built-In Erase using the standard DataFlash page size (264 bytes), an opcode of 83h must be clocked into the device followed by three address bytes comprised of five dummy bits,10 page address bits (PA9 - PA0) that specify the page in the main memory to be written, and nine dummy bits.

To perform a Buffer to Main Memory Page Program with Built-In Erase using the binary page size (256 bytes), an opcode of 83h must be clocked into the device followed by three address bytes comprised of six dummy bits, 10 page address bits (A17 - A8) that specify the page in the main memory to be written, and eight dummy bits.

When a low-to-high transition occurs on the $\overline{CS}$ pin, the device will first erase the selected page in main memory (the erased state is a Logic 1) and then program the data stored in the Buffer into that same page in main memory. Both the erasing and the programming of the page are internally self-timed and should take place in a maximum time of t_{EP}. During this time, the RDY/$\overline{BUSY}$ bit in the Status Register will indicate that the device is busy.

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

6.3 Buffer to Main Memory Page Program without Built-In Erase

The Buffer to Main Memory Page Program without Built-In Erase command allows data that is stored in the Buffer to be written into a pre-erased page in the main memory array. It is necessary that the page in main memory to be written be previously erased in order to avoid programming errors.

To perform a Buffer to Main Memory Page Program without Built-In Erase using the standard DataFlash page size (264 bytes), an opcode of 88h must be clocked into the device followed by three address bytes comprised of five dummy bits,10 page address bits (PA9 - PA0) that specify the page in the main memory to be written, and nine dummy bits.

To perform a Buffer to Main Memory Page Program using the binary page size (256 bytes), an opcode 88h must be clocked into the device followed by three address bytes comprised of six dummy bits, 10 page address bits (A17 - A8) that specify the page in the main memory to be written, and eight dummy bits.

When a low-to-high transition occurs on the $\overline{CS}$ pin, the device will program the data stored in the Buffer into the specified page in the main memory. The page in main memory that is being programmed *must* have been previously erased using one of the erase commands (Page Erase, Block Erase, Sector Erase, or Chip Erase). The programming of the page is internally self-timed and should take place in a maximum time of t_P. During this time, the RDY/$\overline{BUSY}$ bit in the Status Register will indicate that the device is busy.

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.

6.4 Main Memory Page Program through Buffer with Built-In Erase

The Main Memory Page Program through Buffer with Built-In Erase command combines the Buffer Write and Buffer to Main Memory Page Program with Built-In Erase operations into a single operation to help simplify application firmware development. With the Main Memory Page Program through Buffer with Built-In Erase command, data is first clocked into the Buffer, the addressed page in memory is then automatically erased, and then the contents of the Buffer are programmed into the just-erased main memory page.

To perform a Main Memory Page Program through Buffer using the standard DataFlash page size (264 bytes), an opcode of 82h must first be clocked into the device followed by three address bytes comprised of five dummy bits, 10 page address bits (PA9 - PA0) that specify the page in the main memory to be written, and nine buffer address bits (BFA8 - BFA0) that select the first byte in the Buffer to be written.

To perform a Main Memory Page Program through Buffer using the binary page size (256 bytes), an opcode of 82h must first be clocked into the device followed by three address bytes comprised of six dummy bits, 10 page address bits (A17 - A8) that specify the page in the main memory to be written, and eight address bits (A7 - A0) that selects the first byte in the Buffer to be written.

After all address bytes have been clocked in, the device will take data from the input pin (SI) and store it in the Buffer. If the end of the Buffer is reached, the device will wrap around back to the beginning of the Buffer. When there is a low-to-high transition on the $\overline{CS}$ pin, the device will first erase the selected page in main memory (the erased state is a Logic 1) and then program the data stored in the Buffer into that main memory page. Both the erasing and the programming of the page are internally self-timed and should take place in a maximum time of t_{EP}. During this time, the RDY/$\overline{BUSY}$ bit in the Status Register will indicate that the device is busy.

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

6.5 Main Memory Byte/Page Program through Buffer without Built-In Erase

The Main Memory Byte/Page Program through the Buffer without Built-In Erase combines both the Buffer Write and Buffer to Main Memory Program without Built-In Erase operations to allow any number of bytes (1 to 256/264 bytes) to be programmed directly into previously erased locations in the main memory array. With the Main Memory Byte/Page Program through Buffer without Built-In Erase command, data is first clocked into Buffer, and then only the bytes clocked into the Buffer are programmed into the pre-erased byte locations in main memory. Multiple bytes up to the page size can be entered with one command sequence.

To perform a Main Memory Byte/Page Program through the Buffer using the standard DataFlash page size (264 bytes), an opcode of 02h must first be clocked into the device followed by three address bytes comprised of five dummy bits, 10 page address bits (PA9 - PA0) that specify the page in the main memory to be written, and nine buffer address bits (BFA8 - BFA0) that select the first byte in the Buffer to be written. After all address bytes are clocked in, the device will take data from the input pin (SI) and store it in the Buffer. Any number of bytes (1 to 264) can be entered. If the end of the Buffer is reached, then the device will wrap around back to the beginning of the Buffer.

To perform a Main Memory Byte/Page Program through the Buffer using the binary page size (256 bytes), an opcode of 02h must first be clocked into the device followed by three address bytes comprised of six dummy bits, 10 page address bits (PA9 - PA0) that specify the page in the main memory to be written, and eight address bits (A7 - A0) that selects the first byte in the Buffer to be written. After all address bytes are clocked in, the device will take data from the input pin (SI) and store it in the Buffer. Any number of bytes (1 to 256) can be entered. If the end of the Buffer is reached, then the device will wrap around back to the beginning of the Buffer. When using the binary page size, the page and buffer address bits correspond to an 18-bit logical address (A17-A0) in the main memory.

After all data bytes have been clocked into the device, a low-to-high transition on the $\overline{CS}$ pin will start the program operation in which the device will program the data stored in the Buffer into the main memory array. Only the data bytes that were clocked into the device will be programmed into the main memory.

Example: If only two data bytes were clocked into the device, then only two bytes will be programmed into main memory and the remaining bytes in the memory page will remain in their previous state.

The $\overline{CS}$ pin must be deasserted on a byte boundary (multiples of eight bits); otherwise the operation will be aborted and no data will be programmed. The programming of the data bytes is internally self-timed and should take place in a maximum time of t_P (the program time will be a multiple of the t_{BP} time depending on the number of bytes being programmed). During this time, the RDY/$\overline{BUSY}$ bit in the Status Register will indicate that the device is busy.

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.

6.6 Page Erase

The Page Erase command can be used to individually erase any page in the main memory array allowing the Buffer to Main Memory Page Program without Built-In Erase command or the Main Memory Byte/Page Program through Buffer command to be utilized at a later time.

To perform a Page Erase with the standard DataFlash page size (264 bytes), an opcode of 81h must be clocked into the device followed by three address bytes comprised of five dummy bits, 10 page address bits (PA9 - PA0) that specify the page in the main memory to be erased, and nine dummy bits.

To perform a Page Erase with the binary page size (256 bytes), an opcode of 81h must be clocked into the device followed by three address bytes comprised of six dummy bits, 10 page address bits (A17 - A8) that specify the page in the main memory to be erased, and eight dummy bits.

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

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

6.7 Block Erase

The Block Erase command can be used to erase a block of eight pages at one time. This command is useful when needing to pre-erase larger amounts of memory and is more efficient than issuing eight separate Page Erase commands.

To perform a Block Erase with the standard DataFlash page size (264 bytes), an opcode of 50h must be clocked into the device followed by three address bytes comprised of five dummy bits, seven page address bits (PA9 - PA3), and 12 dummy bits. The seven page address bits are used to specify which block of eight pages is to be erased.

To perform a Block Erase with the binary page size (256 bytes), an opcode of 50h must be clocked into the device followed by three address bytes comprised of six dummy bits, seven page address bits (A17 - A11), and 11 dummy bits. The seven page address bits are used to specify which block of eight pages is to be erased.

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

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

Table 6-1. Block Erase Addressing

PA9/A17	PA8/A16	PA7/A15	PA6/A14	PA5/A13	PA4/A12	PA3/A11	PA2/A10	PA1/A9	PA0/A8	Block
0	0	0	0	0	0	0	X	X	X	0
0	0	0	0	0	0	1	X	X	X	1
0	0	0	0	0	1	0	X	X	X	2
0	0	0	0	0	1	1	X	X	X	3
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	0	0	X	X	X	124
1	1	1	1	1	0	1	X	X	X	125
1	1	1	1	1	1	0	X	X	X	126
1	1	1	1	1	1	1	X	X	X	127

6.8 Sector Erase

The Sector Erase command can be used to individually erase any sector in the main memory.

The main memory array is comprised of nine sectors, and only one sector can be erased at a time. To perform an erase of Sector 0a or Sector 0b with the standard DataFlash page size (264 bytes), an opcode of 7Ch must be clocked into the device followed by three address bytes comprised of five dummy bits, seven page address bits (PA9 - PA3), and 12 dummy bits. To perform a Sector 1-7 erase, an opcode of 7Ch must be clocked into the device followed by three address bytes comprised of five dummy bits, three page address bits (PA9 - PA7), and 16 dummy bits.

To perform a Sector 0a or Sector 0b erase with the binary page size (256 bytes), an opcode of 7Ch must be clocked into the device followed by three address bytes comprised of six dummy bits, seven page address bits (A17 - A11), and 11 dummy bits. To perform a Sector 1-7 erase, an opcode of 7Ch must be clocked into the device followed by six dummy bits, three page address bits (A17 - A15), and 15 dummy bits.

The page address bits are used to specify any valid address location within the sector is to be erased. When a low-to-high transition occurs on the $\overline{CS}$ pin, the device will erase the selected sector. The erase operation is internally self-timed and should take place in a maximum time of t_{SE}. During this time, the RDY/$\overline{BUSY}$ bit in the Status Register will indicate that the device is busy.

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

Table 6-2. Sector Erase Addressing

PA9/A17	PA8/A16	PA7/A15	PA6/A14	PA5/A13	PA4/A12	PA3/A11	PA2/A10	PA1/A9	PA0/A8	Sector
0	0	0	0	0	0	0	X	X	X	0a
0	0	0	0	0	0	1	X	X	X	0b
0	0	1	X	X	X	X	X	X	X	1
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	0	1	X	X	X	X	X	X	X	5
1	1	0	X	X	X	X	X	X	X	6
1	1	1	X	X	X	X	X	X	X	7

6.9 Chip Erase

The Chip Erase command allows the entire main memory array to be erased at one time.

To execute the Chip Erase command, a 4-byte command sequence of C7h, 94h, 80h, and 9Ah 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. After the last bit of the opcode sequence has been clocked in, the $\overline{CS}$ pin must be deasserted to start the erase process. The erase operation is internally self-timed and should take place in a time of t_{CE}. During this time, the RDY/$\overline{BUSY}$ bit in the Status Register will indicate that the device is busy.

The Chip Erase command will not affect sectors that are protected or locked down; the contents of those sectors will remain unchanged. Only those sectors that are not protected or locked down will be erased.

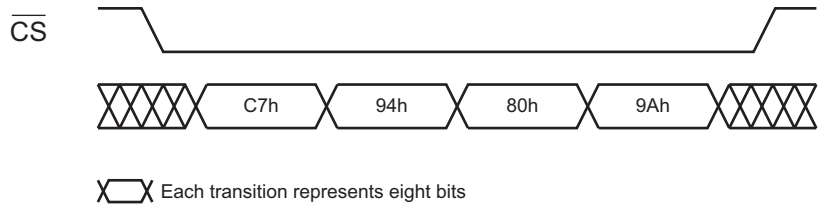
The $\overline{WP}$ pin can be asserted while the device is erasing, but protection will not be activated until the internal erase cycle completes.

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

Table 6-3. Chip Erase Command

Command	Byte 1	Byte 2	Byte 3	Byte 4
Chip Erase	C7h	94h	80h	9Ah

Figure 6-1. Chip Erase



6.10 Read-Modify-Write

A completely self-contained read-modify-write operation can be performed to reprogram any number of sequential bytes in a page in the main memory array without affecting the rest of the bytes in the same page. This command allows the device to easily emulate an EEPROM by providing a method to modify a single byte or more in the main memory in a single operation, without the need for pre-erasing the memory or the need for any external RAM buffers. The Read-Modify-Write command is essentially a combination of the Main Memory Page to Buffer Transfer, Buffer Write, and Buffer to Main Memory Page Program with Built-in Erase commands.

To perform a Read-Modify-Write using the standard DataFlash page size (264 bytes), an opcode of 58h for Buffer 1 must be clocked into the device followed by three address bytes comprised of five dummy bits, 10 page address bits (PA9 - PA0) that specify the page in the main memory to be written and nine byte address bits (BA8-BA0) that designate the starting byte address within the page to reprogram.

To perform a Read-Modify-Write using the binary page size (256 bytes), an opcode of 58h for Buffer 1 must be clocked into the device followed by three address bytes comprised of six dummy bits, 10 page address bits (A17 - A8) that specify the page in the main memory to be written and eight byte address bits (A7-A0) that designate the starting byte address within the page to reprogram.

After the address bytes have been clocked in, any number of sequential data bytes from one to 256/264 bytes can be clocked into the device. If the end of the buffer is reached when clocking in the data, then the device will wrap around back to the beginning of the buffer. After all data bytes have been clocked into the device, a low-to-high transition on the CS pin will start the self-contained, internal read-modify-write operation. Only the data bytes that were clocked into the device will be reprogrammed in the main memory.

Example: If only one data byte was clocked into the device, then only one byte in main memory will be reprogrammed and the remaining bytes in the main memory page will remain in their previous state.

The $\overline{CS}$ pin must be deasserted on a byte boundary (multiples of eight bits); otherwise, the operation will be aborted and no data will be programmed. The reprogramming of the data bytes is internally self-timed and should take place in a maximum time of t_P. During this time, the RDY/$\overline{BUSY}$ bit in the Status Register will indicate that the device is busy.

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

Note: The Read-Modify-Write command uses the same opcodes as the Auto Page Rewrite command. If no data bytes are clocked into the device, then the device will perform an Auto Page Rewrite operation. See the Auto Page Rewrite command description on page 25 for more details.

7. Sector Protection

Two protection methods, hardware and software controlled, are provided for protection against inadvertent or erroneous program and erase cycles. The software controlled method relies on the use of software commands to enable and disable sector protection while the hardware controlled method employs the use of the Write Protect ($\overline{WP}$) pin. The selection of which sectors that are to be protected or unprotected against program and erase operations is specified in the nonvolatile Sector Protection Register. The status of whether or not sector protection has been enabled or disabled by either the software or the hardware controlled methods can be determined by checking the Status Register.

7.1 Software Sector Protection

Software controlled protection is useful in applications in which the $\overline{WP}$ pin is not or cannot be controlled by a host processor. In such instances, the $\overline{WP}$ pin may be left floating (the $\overline{WP}$ pin is internally pulled high) and sector protection can be controlled using the Enable Sector Protection and Disable Sector Protection commands.

If the device is power cycled, then the software controlled protection will be disabled. Once the device is powered up, the Enable Sector Protection command should be reissued if sector protection is desired and if the $\overline{WP}$ pin is not used.

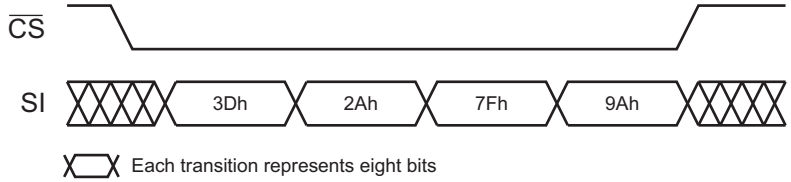
7.1.1 Enable Sector Protection

Sectors specified for protection in the Sector Protection Register can be protected from program and erase operations by issuing the Enable Sector Protection command. To enable the sector protection, a 4-byte command sequence of 3Dh, 2Ah, 7Fh, and A9h must be clocked into the device. After the last bit of the opcode sequence has been clocked in, the $\overline{CS}$ pin must be deasserted to enable the Sector Protection.

Table 7-1. Enable Sector Protection Command

Command	Byte 1	Byte 2	Byte 3	Byte 4
Enable Sector Protection	3Dh	2Ah	7Fh	A9h

Figure 7-1. Enable Sector Protection



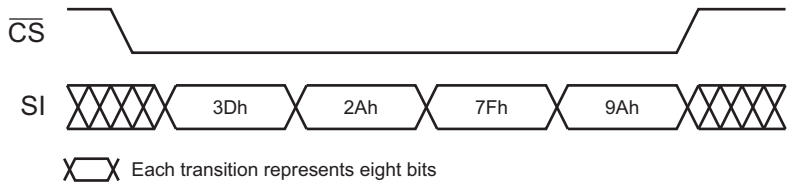
7.1.2 Disable Sector Protection

To disable the sector protection, a 4-byte command sequence of 3Dh, 2Ah, 7Fh, and 9Ah must be clocked into the device. After the last bit of the opcode sequence has been clocked in, the $\overline{CS}$ pin must be deasserted to disable the sector protection.

Table 7-2. Disable Sector Protection Command

Command	Byte 1	Byte 2	Byte 3	Byte 4
Disable Sector Protection	3Dh	2Ah	7Fh	9Ah

Figure 7-2. Disable Sector Protection



7.2 Hardware Controlled Protection

Sectors specified for protection in the Sector Protection Register and the Sector Protection Register itself can be protected from program and erase operations by asserting the $\overline{WP}$ pin and keeping the pin in its asserted state. The Sector Protection Register and any sector specified for protection cannot be erased or programmed as long as the $\overline{WP}$ pin is asserted. In order to modify the Sector Protection Register, the $\overline{WP}$ pin must be deasserted. If the $\overline{WP}$ pin is permanently connected to GND, then the contents of the Sector Protection Register cannot be changed. If the $\overline{WP}$ pin is deasserted or permanently connected to V_{CC}, then the contents of the Sector Protection Register can be modified.

The $\overline{WP}$ pin will override the software controlled protection method but only for protecting the sectors.

Example: If the sectors were not previously protected by the Enable Sector Protection command, then simply asserting the $\overline{WP}$ pin would enable the sector protection within the maximum specified t_{WPE} time. When the $\overline{WP}$ pin is deasserted, however, the sector protection would no longer be enabled (after the maximum specified t_{WPD} time) as long as the Enable Sector Protection command was not issued while the $\overline{WP}$ pin was asserted. If the Enable Sector Protection command was issued before or while the $\overline{WP}$ pin was asserted, then simply deasserting the $\overline{WP}$ pin would not disable the sector protection. In this case, the Disable Sector Protection command would need to be issued while the $\overline{WP}$ pin is deasserted to disable the sector protection. The Disable Sector Protection command is also ignored whenever the $\overline{WP}$ pin is asserted.

A noise filter is incorporated to help protect against spurious noise that my inadvertently assert or deassert the $\overline{WP}$ pin.

Figure 7-3 and Table 7-3 detail the sector protection status for various scenarios of the $\overline{WP}$ pin, the Enable Sector Protection command, and the Disable Sector Protection command.

Figure 7-3. $\overline{WP}$ Pin and Protection Status

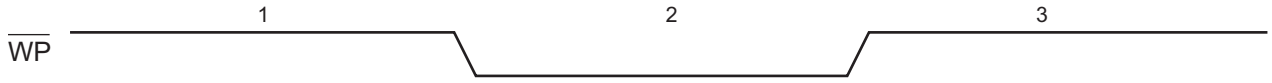


Table 7-3. $\overline{WP}$ Pin and Protection Status

Time Period	$\overline{WP}$ Pin	Enable Sector Protection Command	Disable Sector Protection Command	Sector Protection Status	Sector Protection Register
1	High	Command Not Issued Previously	X	Disabled	Read/Write
		—	Issue Command	Disabled	Read/Write
		Issue Command	—	Enabled	Read/Write
2	Low	X	X	Enabled	Read
3	High	Command Issued During Period 1 or 2	Not Issued Yet	Enabled	Read/Write
		—	Issue Command	Disabled	Read/Write
		Issue Command	—	Enabled	Read/Write

7.3 Sector Protection Register

The nonvolatile Sector Protection Register specifies which sectors are to be protected or unprotected with either the software or hardware controlled protection methods. The Sector Protection Register contains eight bytes of data, of which byte locations 0 through 7 contain values that specify whether Sectors 0 through 7 will be protected or unprotected. The Sector Protection Register is user modifiable and must be erased before it can be reprogrammed. Table 7-4 illustrates the format of the Sector Protection Register.

Table 7-4. Sector Protection Register

Sector Number	0 (0a, 0b)	1 to 7
Protected	See Table 7-5	FFh
Unprotected		00h

Note: 1. The default values for bytes 0 through 7 are 00h when shipped from Adesto.

Table 7-5. Sector 0 (0a, 0b) Sector Protection Register Byte Value

	Bit 7:6	Bit 5:4	Bit 3:2	Bit 1:0	
	Sector 0a (Page 0-7)	Sector 0b (Page 8-127)	N/A	N/A	Data Value
Sectors 0a and 0b Unprotected	00	00	XX	XX	0Xh
Protect Sector 0a	11	00	XX	XX	CXh
Protect Sector 0b	00	11	XX	XX	3Xh
Protect Sectors 0a and 0b	11	11	XX	XX	FXh

Note: 1. X = Don't care

7.3.1 Erase Sector Protection Register

In order to modify and change the values of the Sector Protection Register, it must first be erased using the Erase Sector Protection Register command.

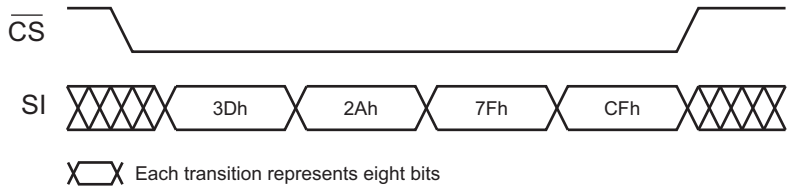
To erase the Sector Protection Register, a 4-byte command sequence of 3Dh, 2Ah, 7Fh, and CFh must be clocked into the device. After the last bit of the opcode sequence has been clocked in, the $\overline{CS}$ pin must be deasserted to initiate the internally self-timed erase cycle. The erasing of the Sector Protection Register should take place in a maximum time of t_{PE}. During this time, the RDY/$\overline{BUSY}$ bit in the Status Register will indicate that the device is busy. If the device is powered-down before the completion of the erase cycle, then the contents of the Sector Protection Register cannot be guaranteed.

The Sector Protection Register can be erased with sector protection enabled or disabled. Since the erased state (FFh) of each byte in the Sector Protection Register is used to indicate that a sector is specified for protection, leaving the sector protection enabled during the erasing of the register allows the protection scheme to be more effective in the prevention of accidental programming or erasing of the device. If for some reason an erroneous program or erase command is sent to the device immediately after erasing the Sector Protection Register and before the register can be reprogrammed, then the erroneous program or erase command will not be processed because all sectors would be protected.

Table 7-6. Erase Sector Protection Register Command

Command	Byte 1	Byte 2	Byte 3	Byte 4
Erase Sector Protection Register	3Dh	2Ah	7Fh	CFh

Figure 7-4. Erase Sector Protection Register



7.3.2 Program Sector Protection Register

Once the Sector Protection Register has been erased, it can be reprogrammed using the Program Sector Protection Register command.

To program the Sector Protection Register, a 4-byte command sequence of 3Dh, 2Ah, 7Fh, and FCh must be clocked into the device followed by eight bytes of data corresponding to Sectors 0 through 7. After the last bit of the opcode sequence and data have been clocked in, the $\overline{CS}$ pin must be deasserted to initiate the internally self-timed program cycle. The programming of the Sector Protection Register should take place in a maximum time of t_P. During this time, the RDY/$\overline{BUSY}$ bit in the Status Register will indicate that the device is busy. If the device is powered-down before the completion of the erase cycle, then the contents of the Sector Protection Register cannot be guaranteed.

If the proper number of data bytes is not clocked in before the $\overline{CS}$ pin is deasserted, then the protection status of the sectors corresponding to the bytes not clocked in cannot be guaranteed.

Example: If only the first two bytes are clocked in instead of the complete eight bytes, then the protection status of the last six sectors cannot be guaranteed. Furthermore, if more than eight bytes of data is clocked into the device, then the data will wrap back around to the beginning of the register. For instance, if nine bytes of data are clocked in, then the ninth byte will be stored at byte location 0 of the Sector Protection Register.

The data bytes clocked into the Sector Protection Register need to be valid values (0Xh, 3Xh, CXh, and FXh for Sector 0a or Sector 0b, and 00h or FFh for other sectors) in order for the protection to function correctly. If a non-valid value is clocked into a byte location of the Sector Protection Register, then the protection status of the sector corresponding to that byte location cannot be guaranteed.

Example: If a value of 17h is clocked into byte location 2 of the Sector Protection Register, then the protection status of Sector 2 cannot be guaranteed.

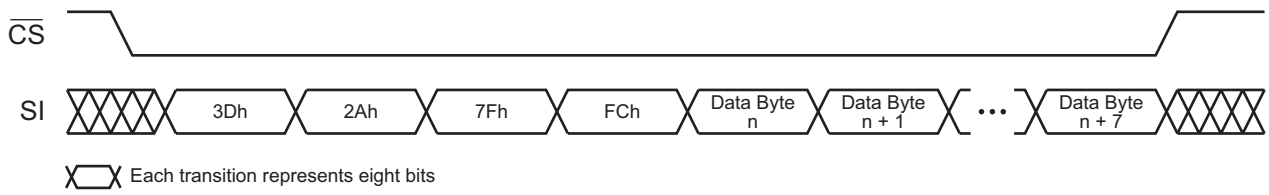
The Sector Protection Register can be reprogrammed while the sector protection is enabled or disabled. Being able to reprogram the Sector Protection Register with the sector protection enabled allows the user to temporarily disable the sector protection to an individual sector rather than disabling sector protection completely.

The Program Sector Protection Register command utilizes the internal buffer for processing. Therefore, the contents of the Buffer will be altered from its previous state when this command is issued.

Table 7-7. Program Sector Protection Register Command

Command	Byte 1	Byte 2	Byte 3	Byte 4
Program Sector Protection Register	3Dh	2Ah	7Fh	FCh

Figure 7-5. Program Sector Protection Register



7.3.3 Read Sector Protection Register

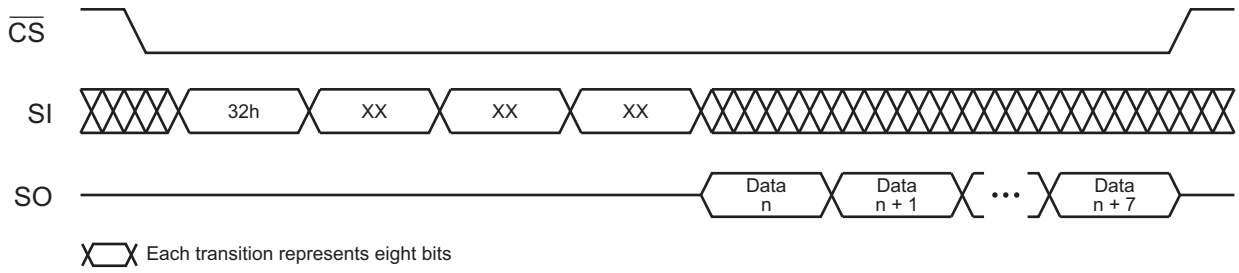
To read the Sector Protection Register, an opcode of 32h and three dummy bytes must be clocked into the device. After the last bit of the opcode and dummy bytes have been clocked in, any additional clock pulses on the SCK pin will result in the Sector Protection Register contents being output on the SO pin. The first byte (byte location 0) corresponds to Sector 0 (0a and 0b), the second byte corresponds to Sector 1, and the last byte (byte location 7) corresponds to Sector 7. Once the last byte of the Sector Protection Register has been clocked out, any additional clock pulses will result in undefined data being output on the SO pin. The $\overline{CS}$ pin must be deasserted to terminate the Read Sector Protection Register operation and put the output into a high-impedance state.

Table 7-8. Read Sector Protection Register Command

Command	Byte 1	Byte 2	Byte 3	Byte 4
Read Sector Protection Register	32h	XXh	XXh	XXh

Note: 1. XX = Dummy byte

Figure 7-6. Read Sector Protection Register



7.3.4 About the Sector Protection Register

The Sector Protection Register is subject to a limit of 10,000 erase/program cycles. Users are encouraged to carefully evaluate the number of times the Sector Protection Register will be modified during the course of the application's life cycle. If the application requires that the Security Protection Register be modified more than the specified limit of 10,000 cycles because the application needs to temporarily unprotect individual sectors (sector protection remains enabled while the Sector Protection Register is reprogrammed), then the application will need to limit this practice. Instead, a combination of temporarily unprotecting individual sectors along with disabling sector protection completely will need to be implemented by the application to ensure that the limit of 10,000 cycles is not exceeded.

8. Security Features

8.1 Sector Lockdown

The device incorporates a sector lockdown mechanism that allows each individual sector to be permanently locked so that it becomes read-only (ROM). This is useful for applications that require the ability to permanently protect a number of sectors against malicious attempts at altering program code or security information.

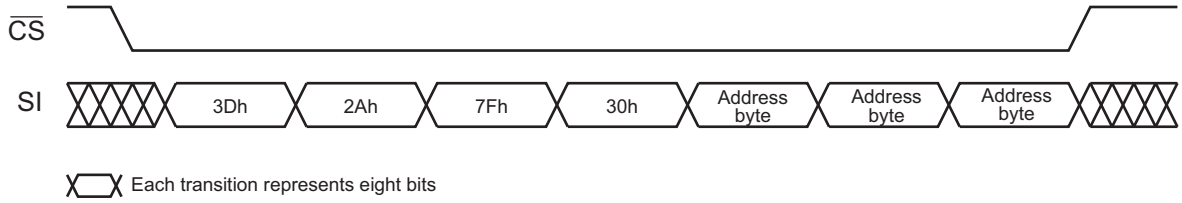
Warning: Once a sector is locked down, it can never be erased or programmed, and it can never be unlocked.

To issue the sector lockdown command, a 4-byte command sequence of 3Dh, 2Ah, 7Fh, and 30h must be clocked into the device followed by three address bytes specifying any address within the sector to be locked down. After the last address bit has been clocked in, the $\overline{CS}$ pin must be deasserted to initiate the internally self-timed lockdown sequence. The lockdown sequence should take place in a maximum time of t_P. During this time, the RDY/$\overline{BUSY}$ bit in the Status Register will indicate that the device is busy. If the device is powered-down before the completion of the lockdown sequence, then the lockdown status of the sector cannot be guaranteed. In this case, it is recommended that the user read the Sector Lockdown Register to determine the status of the appropriate sector lockdown bits or bytes and re-issue the Sector Lockdown command if necessary.

Table 8-1. Sector Lockdown Command

Command	Byte 1	Byte 2	Byte 3	Byte 4
Sector Lockdown	3Dh	2Ah	7Fh	30h

Figure 8-1. Sector Lockdown



8.1.1 Read Sector Lockdown Register

The nonvolatile Sector Lockdown Register specifies which sectors in the main memory are currently unlocked or have been permanently locked down. The Sector Lockdown Register is a read-only register and contains eight bytes of data which correspond to Sectors 0 through 7. To read the Sector Lockdown Register, an opcode of 35h must be clocked into the device followed by three dummy bytes. After the last bit of the opcode and dummy bytes have been clocked in, the data for the contents of the Sector Lockdown Register will be clocked out on the SO pin. The first byte (byte location 0) corresponds to Sector 0 (0a and 0b), the second byte corresponds to Sector 1, and the last byte (byte location 7) corresponds to Sector 7. After the last byte of the Sector Lockdown Register has been read, additional pulses on the SCK pin will result in undefined data being output on the SO pin.

Deasserting the $\overline{CS}$ pin will terminate the Read Sector Lockdown Register operation and put the SO pin into a high-impedance state. Table 8-2 details the format the Sector Lockdown Register.

Table 8-2. Sector Lockdown Register

Sector Number	0 (0a, 0b)	1 to 7
Locked	See Table 8-3	FFh
Unlocked		00h

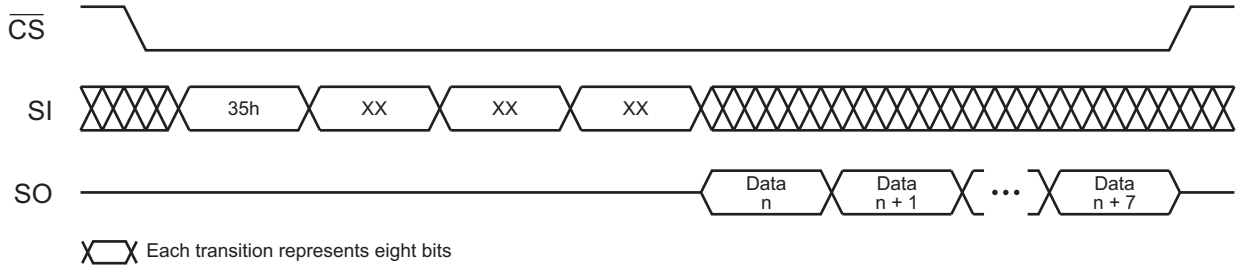
Table 8-3. Sector 0 (0a and 0b) Sector Lockdown Register Byte Value

	Bit 7:6	Bit 5:4	Bit 3:2	Bit 1:0	
	Sector 0a (Page 0-7)	Sector 0b (Page 8-127)	N/A	N/A	Data Value
Sectors 0a and 0b Unlocked	00	00	00	00	00h
Sector 0a Locked	11	00	00	00	C0h
Sector 0b Locked	00	11	00	00	30h
Sectors 0a and 0b Locked	11	11	00	00	F0h

Table 8-4. Read Sector Lockdown Register Command

Command	Byte 1	Byte 2	Byte 3	Byte 4
Read Sector Lockdown Register	35h	XXh	XXh	XXh

Figure 8-2. Read Sector Lockdown Register



8.1.2 Freeze Sector Lockdown

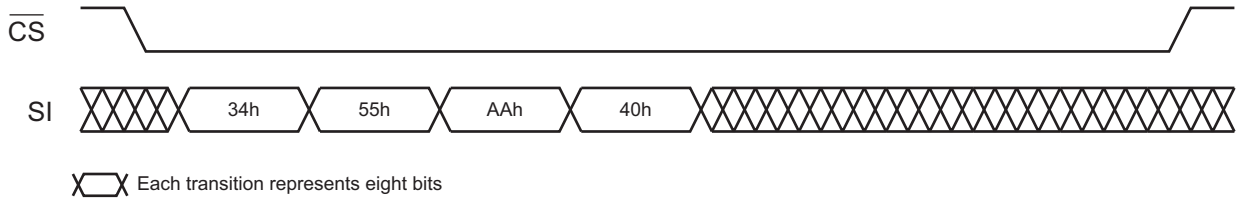
The Sector Lockdown command can be permanently disabled, and the current sector lockdown state can be permanently frozen so that no additional sectors can be locked down aside from those already locked down. Any attempts to issue the Sector Lockdown command after the Sector Lockdown State has been frozen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

To issue the Freeze Sector Lockdown command, the $\overline{CS}$ pin must be asserted and the opcode sequence of 34h, 55h, AAh, and 40h must be clocked into the device. 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 SLE bit in the Status Register will be permanently reset to a Logic 0 to indicate that the Sector Lockdown command is permanently disabled.

Table 8-5. Freeze Sector Lockdown

Command	Byte 1	Byte 2	Byte 3	Byte 4
Freeze Sector Lockdown	34h	55h	AAh	40h

Figure 8-3. Freeze Sector Lockdown



8.2 Security Register

The device contains a specialized Security Register that can be used for purposes such as unique device serialization or locked key storage. The register is comprised of a total of 128 bytes that is divided into two portions. The first 64 bytes (byte locations 0 through 63) of the Security Register are allocated as an One-Time Programmable space. Once these 64 bytes have been programmed, they cannot be erased or reprogrammed. The remaining 64 bytes of the register (byte locations 64 through 127) are factory programmed by Adesto and will contain a unique value for each device. The factory programmed data is fixed and cannot be changed.

Table 8-6. Security Register

	Security Register Byte Number							
	0	1	· · ·	63	64	65	· · ·	127
Data Type	One-Time User Programmable				Factory Programmed by Adesto			

8.2.1 Programming the Security Register

The user programmable portion of the Security Register does not need to be erased before it is programmed.

To program the Security Register, a 4-byte opcode sequence of 9Bh, 00h, 00h, and 00h must be clocked into the device. After the last bit of the opcode sequence has been clocked into the device, the data for the contents of the 64-byte user programmable portion of the Security Register must be clocked in.

After the last data byte has been clocked in, the $\overline{CS}$ pin must be deasserted to initiate the internally self-timed program cycle. The programming of the Security Register should take place in a time of t_P, during which time the RDY/$\overline{BUSY}$ bit in the Status Register will indicate that the device is busy. If the device is powered-down during the program cycle, then the contents of the 64-byte user programmable portion of the Security Register cannot be guaranteed.

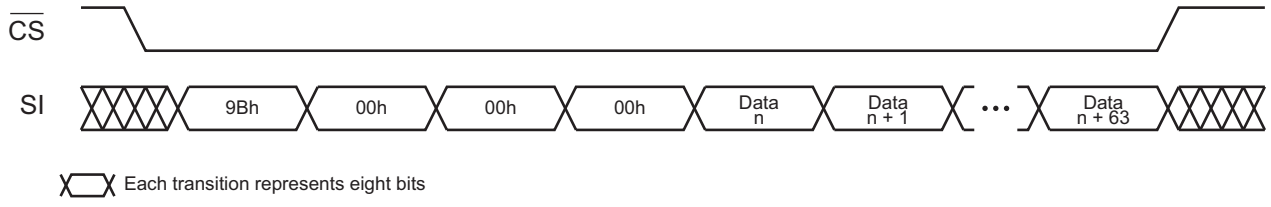
If the full 64 bytes of data are not clocked in before the $\overline{CS}$ pin is deasserted, then the values of the byte locations not clocked in cannot be guaranteed.

Example: If only the first two bytes are clocked in instead of the complete 64 bytes, then the remaining 62 bytes of the user programmable portion of the Security Register cannot be guaranteed. Furthermore, if more than 64 bytes of data is clocked into the device, then the data will wrap back around to the beginning of the register. For example, if 65 bytes of data are clocked in, then the 65th byte will be stored at byte location 0 of the Security Register.

Warning: **The user programmable portion of the Security Register can only be programmed one time.** Therefore, it is not possible, for example, to only program the first two bytes of the register and then program the remaining 62 bytes at a later time.

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

Figure 8-4. Program Security Register

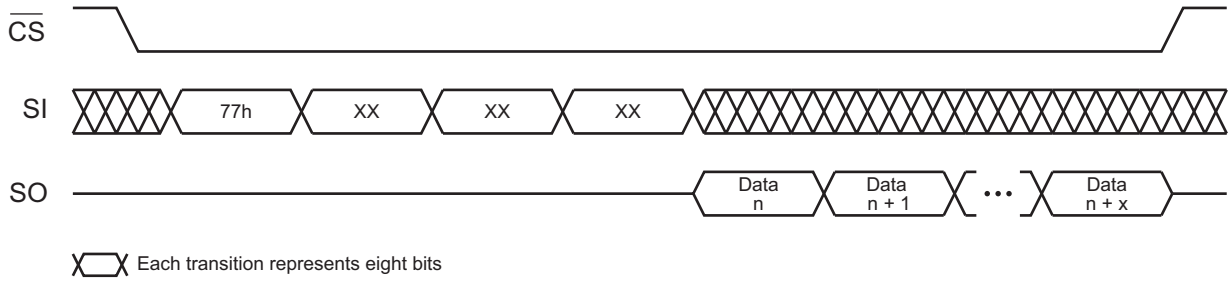


8.2.2 Reading the Security Register

To read the Security Register, an opcode of 77h and three dummy bytes must be clocked into the device. After the last dummy bit has been clocked in, the contents of the Security Register can be clocked out on the SO pin. After the last byte of the Security Register has been read, additional pulses on the SCK pin will result in undefined data being output on the SO pin.

Deasserting the $\overline{CS}$ pin will terminate the Read Security Register operation and put the SO pin into a high-impedance state.

Figure 8-5. Read Security Register



9. Additional Commands

9.1 Main Memory Page to Buffer Transfer

A page of data can be transferred from the main memory to the Buffer. To transfer a page of data using the standard DataFlash page size (264 bytes), an opcode of 53h must be clocked into the device followed by three address bytes comprised of five dummy bits, 10 page address bits (PA9 - PA0) which specify the page in main memory to be transferred, and nine dummy bits. To transfer a page of data using the binary page size (256 bytes), an opcode of 53h must be clocked into the device followed by three address bytes comprised of six dummy bits, 10 page address bits (A17 - A8) which specify the page in the main memory to be transferred, and eight dummy bits.

The $\overline{CS}$ pin must be low while toggling the SCK pin to load the opcode and the three address bytes from the input pin (SI). The transfer of the page of data from the main memory to the Buffer will begin when the $\overline{CS}$ pin transitions from a low to a high state. During the page transfer time (t_{XFR}), the RDY/$\overline{BUSY}$ bit in the Status Register can be read to determine whether or not the transfer has been completed.

9.2 Main Memory Page to Buffer Compare

A page of data in main memory can be compared to the data in the Buffer as a method to ensure that data was successfully programmed after a Buffer to Main Memory Page Program command. To compare a page of data with the standard DataFlash page size (264 bytes), an opcode of 60h must be clocked into the device followed by three address bytes comprised of five dummy bits, 10 page address bits (PA9 - PA0) which specify the page in the main memory to be compared to the Buffer, and nine dummy bits. To compare a page of data with the binary page size (256 bytes), an opcode of 60h must be clocked into the device followed by three address bytes comprised of six dummy bits, 10 page address bits (A17 - A8) which specify the page in the main memory to be compared to the Buffer, and eight dummy bits.

The $\overline{CS}$ pin must be low while toggling the SCK pin to load the opcode and the address bytes from the input pin (SI). On the low-to-high transition of the $\overline{CS}$ pin, the data bytes in the selected Main Memory Page will be compared with the data bytes in the Buffer. During the compare time (t_{COMP}), the RDY/$\overline{BUSY}$ bit in the Status Register will indicate that the part is busy. On completion of the compare operation, bit 6 of the Status Register will be updated with the result of the compare.

9.3 Auto Page Rewrite

This command only needs to be used if the possibility exists that static (non-changing) data may be stored in a page or pages of a sector and the other pages of the same sector are erased and programmed a large number of times. Applications that modify data in a random fashion within a sector may fall into this category. To preserve data integrity of a sector, each page within a sector must be updated/rewritten at least once within every 50,000 cumulative page erase/program operations within that sector. The Auto Page Rewrite command provides a simple and efficient method to “refresh” a page in the main memory array in a single operation.

The Auto Page Rewrite command is a combination of the Main Memory Page to Buffer Transfer and Buffer to Main Memory Page Program with Built-In Erase commands. With the Auto Page Rewrite command, a page of data is first transferred from the main memory to the Buffer and then the same data is programmed back into the same page of main memory, essentially “refreshing” the contents of that page. To start the Auto Page Rewrite operation with the standard DataFlash page size (264 bytes), a 1-byte opcode, 58h must be clocked into the device followed by three address bytes comprised of five dummy bits, 10 page address bits (PA9-PA0) that specify the page in main memory to be rewritten, and nine dummy bits.

To initiate an Auto Page Rewrite with the a binary page size (256 bytes), the opcode 58h must be clocked into the device followed by three address bytes consisting of six dummy bits, 10 page address bits (A17 - A8) that specify the page in the main memory that is to be rewritten, and eight dummy bits. When a low-to-high transition occurs on the $\overline{CS}$ pin, the part will first transfer data from the page in main memory to the Buffer and then program the data from the Buffer back into same page of main memory. The operation is internally self-timed and should take place in a maximum time of t_{EP}. During this time, the RDY/$\overline{BUSY}$ Status Register will indicate that the part is busy.