ADVANCED DATASHEET

ATXP064B

EcoXiP™ High Performance Low-Power Octal Flash

Features

- Optimized for eXecute-in-Place (XiP) operations
  - Instant on
  - Reduces average latency for improving CPU performance
  - Enables 40% higher CPU performance than the basic octal SPI protocol
- High throughput
  - Octal xSPI mode (8-8-8) with dual data rate (DDR)
  - 133 MHz maximum operating frequency in octal DDR mode
  - Up to 266 MBytes per second data rate in octal DDR mode
  - Full JESD251, and JESD216D compatibility
  - Supports SPI modes 0 and 3
- Low power dissipation
  - 200 nA ultra-deep power-down current (typical)
  - 4 µA deep power-down current (typical)
  - 20 µA standby current (typical, for SPI Mode)
  - 35 µA standby current (typical, for octal mode)
  - 10 mA active read current (typical, @ 66 MHz, for SPI Mode@ 1 pF load)
  - 34 mA active read current (typical, @ 133 MHz, for octal mode dual data rate @ 1 pF load)
- Concurrent Read and Write
  - Simultaneous execution of Read and Write operations
  - No additional delay executing Read commands issued during Program or Erase
  - Flexible boundary between Data Storage Area and Read While Write Area
- Single voltage operation with range of 1.7V to 1.95V
- Flexible, optimized erase architecture for code + data storage applications
  - Uniform 4-Kbyte block erase
  - Uniform 32-Kbyte block erase
  - Uniform 64-Kbyte block erase
  - Full chip erase
- Hardware controlled locking of protected sectors via $\overline{WP}$ pin
- 256-byte, One-Time Programmable (OTP) security register
  - 128 bytes factory programmed with unique identifier
  - 128 bytes user programmable
- Flexible programming
  - Byte/page program (1 to 256 bytes)
  - Single and octal-input byte/page program (1 to 256 bytes)
  - Write-to-buffer and write-buffer-to-memory commands
  - Active status interrupt when program or erase operation has finished
- Program and erase suspend/resume
- Power optimized program and erase control
  - Automatic deep power-down or ultra-deep power-down upon the completion of program or erase operation
- Automatic checking and reporting of program/erase failures
- Software controlled reset
- Hardware reset pin
- JEDEC standard (JESD252) hardware reset
- JEDEC standard (JEP106AX) manufacturer and device ID read methodology
- Support for Serial Flash Discoverable Parameters (SFDP)
- Programmable I/O drive strength
- Endurance: 100,000 program/erase cycles
- Data Retention: 20 years
- Complies with full industrial temperature range
  - -40°C - 85°C for packaged parts
  - -40°C - 105°C for Known Good Die (KGD)
- Industry standard green (Pb/halide-free/RoHS compliant) package options
  - 24-ball BGA
  - 46-ball WLCSP
  - Known Good Die [KGD]

Adesto™

1. Description

The Adesto® EcoXiP™ ATXP064B is a high-performance, low-power octal Flash memory device designed for use in IoT, smart devices, and embedded processors that demand high performance and instant-on capabilities while keeping power consumption down. eXecute in Place (XiP) technology is well suited to meet these needs. Specifically designed to work with cache controllers, EcoXiP dramatically reduces latency for cache misses. Unlike other octal Flash solutions that sacrifice power consumption for high performance, EcoXiP maintains low power operation by utilizing Adesto's proprietary technology.

The ATXP064B allows writing to the Flash array at the same time as code is being fetched from a different part of the array. This Read-While-Write capability enables firmware updates and data logging without the need for additional data storage devices in the system.

The ATXP064B has high data throughput supporting octal xSPI with dual data rate up to 266 Mbytes per second and is fully JESD251 and JESD216D compliant. For faster transfer of data from the device, the ATXP064B provides a Data Strobe (DS) output signal. DS serves as a source-synchronous clock to the output data. This enables much faster clock rates for both DDR and SDR modes than can be achieved by using SCK as the clock signal for incoming data.

The ATXP064B is optimized for low power system operation and support programmable strength I/O drivers that can be matched to the required operating capacitive load. The ATXP064B supports 3 low-power operation modes and an option to automatically switch to low power mode upon completion of a program or erase operation

The erase block sizes of the ATXP064B have been optimized to meet the needs of today's code and data storage applications. By optimizing the size of the erase blocks, the memory space can be used much more efficiently. Because certain code modules and data storage segments must reside by themselves in their own erase regions, the wasted and unused memory space that occurs with large sectored and large block erase Flash memory devices can be greatly reduced. This increased memory space efficiency allows additional code routines and data storage segments to be added while still maintaining the same overall device density.

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

2. Pin Descriptions and Pinouts

All I/O pins and Data Strobe (DS) are in the tri-state mode when not actively driven. To reduce power consumption, it is recommended to not leave pins floating, but rather to have internal pull-down resistors in the host controller to ensure that all pins have a valid logic level at all times.

Table 2-1. Pin Descriptions

Symbol	Description	Asserted State	Type
$\overline{CS}$	**CHIP SELECT:** Asserting the $\overline{CS}$ pin selects the device. When the $\overline{CS}$ pin is deasserted, the device is deselected and normally placed in standby mode (not deep power-down mode), and the output pins are placed in a high-impedance state. When the device is deselected, data is not 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 does not enter the standby mode until the completion of the operation.	Low	Input

Table 2-1. Pin Descriptions (Continued)

Symbol	Description	Asserted State	Type
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. In Single Data Rate (SDR) modes, command, address, and input data present on the I/O pins are always latched in on the rising edge of SCK, while output data on the I/O pins is always clocked out on the falling edge of SCK. In the Double Data Rate (DDR) modes, address and input data present on the I/O pins data are latched on both clock edges. For more accurate operation at high speeds, SCK is returned as DS synchronous to output data.	-	Input
SI (I/O_0)	**SERIAL INPUT:** In SPI mode, 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. In SDR modes, command, address, and input data present on the SI pin is always latched in on the rising edge of SCK. In the DDR modes, address and input data present on the SI pin is latched on both edges of SCK. In octal mode, the SI Pin becomes an I/O pin (I/O_0) in conjunction with other pins. In SDR modes this allows eight bits of command, address, or input data on I/O_{7-0} to be clocked in on the rising edge of SCK, or eight bits clocked out on the falling edge of SCK. In Double Data Rate (DDR) mode this allows eight bits of address or input data on I/O_{7-0} to be clocked in on every edge of SCK, or eight bits clocked out on every edge of SCK. Commands are clocked on the rising edge of SCK, requiring a whole clock cycle also in the DDR modes. To maintain consistency with the SPI nomenclature, the SI (I/O_0) pin is referenced as the SI pin, unless specifically addressing the multi-I/O modes, in which case it is referenced as I/O_0. Data present on the SI pin is ignored whenever the device is deselected ($\overline{CS}$ is deasserted).	-	Input/ Output
SO (I/O_1)	**SERIAL OUTPUT:** The SO pin is used to shift data out from the device. In the Single Data Rate (SDR) mode, data on the SO pin is always clocked out on the falling edge of SCK. In the Double Data Rate (DDR) mode, data on the SO pin is clocked out on both edges of SCK. In octal mode, the SO Pin becomes an I/O pin (I/O_1) in conjunction with other pins. In SDR modes this allows eight bits of command, address, or input data on I/O_{7-0} to be clocked in on the rising edge of SCK, or eight bits clocked out on the falling edge of SCK. In DDR mode this allows eight bits of address or input data on I/O_{7-0} to be clocked in on every edge of SCK, or eight bits clocked out on every edge of SCK. Commands are clocked on the rising edge of SCK, requiring a whole clock cycle also in the DDR mode. To maintain consistency with the SPI nomenclature, the SO (I/O_1) pin is referenced as the SO pin throughout this document, unless specifically addressing the Multi-I/O modes, in which case it is referenced as I/O_1. The SO pin is placed in the high-impedance state whenever the device is deselected ($\overline{CS}$ is deasserted).	-	Input/ Output

Table 2-1. Pin Descriptions (Continued)

Symbol	Description	Asserted State	Type
$\overline{WP}$ (I/O_2)	**WRITE PROTECT:** The $\overline{WP}$ pin controls the hardware locking feature of the device. Please refer to "Protection Commands and Features" on page 40 for more details on protection features and the $\overline{WP}$ pin. In octal mode, the I/O_2 pin is used together with I/O_{7-0} as a bidirectional I/O pin. In these modes, the I/O_2 pin is placed in the high-impedance state whenever the device is deselected ($\overline{CS}$ is deasserted). The $\overline{WP}$ pin is internally set to a logic 1 state at power up. The $\overline{WP}$ pin is used for write protection control in standard SPI mode (1-1-1). The WPP bit in Status/Control register byte 3 reflects the current state of $\overline{WP}$ pin. If the ATXP064B is installed in a system which supports only standard SPI mode, and write protection control is not needed, the $\overline{WP}$ pin can be left floating. In this case the value of the WPP bit is undefined. Once the $\overline{WP}$ pin is used in standard SPI mode, it should be always driven. If the ATXP064B is installed in a system which also supports octal mode, the $\overline{WP}$ pin functions as write protection control only in standard SPI mode. In this case, if write protection control is required, the $\overline{WP}$ pin must be driven externally when the device is switched from OPI to standard SPI mode. In octal mode the WPP bit value is undefined and write protection control logic behaves as if the $\overline{WP}$ pin is connected to a logic 1 state.	Low	Input/ Output
I/O_3	**I/O_3 :** In octal mode, I/O_3 is used together with I/O_{7-0} as a bidirectional I/O pin. In these modes, the I/O_3 pin is placed in the high-impedance state whenever the device is deselected ($\overline{CS}$ is deasserted). The I/O_3 pin should be kept in high-Z state or pulled high while in SPI mode.	-	Input/ Output
DS	**DATA STROBE:** DS is the return of the SCK clock, synchronized to the return data. It is available in all modes, and makes it easier to achieve high clock speeds in a system. DS is required to achieve maximum clock speeds. DS is driven low as soon as $\overline{CS}$ is driven low by the host controller, and is driven until $\overline{CS}$ is pulled high again. DS is only toggled while the device is transmitting data. DS remains low while the device is receiving commands, address and data. In Single Data Rate (SDR) mode, DS is driven high in the first half of the data output cycle, and low in the second half. In this mode, DS is basically the inverse of SCK, but with a delay. If SCK is low at the end of an operation, DS will be high. This is the recommended mode of operation in SDR mode. However, if the host controller requires DS to end low to be able to use this signal correctly, it is also possible to add half of an SCK pulse and end with SCK high and DS low. In DDR mode, DS changes value at the edge of each data bit. In this mode, DS is basically the same value as SCK, but with a delay. Achieving high clock rates in systems without DS requires a short signal path between the SPI master and the memory device, and careful layout of all signal lines to minimize signal delays.	-	Output
I/O_7, I/O_6, I/O_5, I/O_4	**SERIAL I/O:** In octal Mode, I/O_{7-4} are used together with I/O_{3-0} as bidirectional I/O pins. In this mode, the I/O_{7-4} pins (as well as the I/O_{3-0} pins) are placed in the high-impedance state whenever the device is deselected ($\overline{CS}$ is deasserted). In standard SPI mode, the I/O_{7-4} pins are always in a high-impedance state.	-	Input/ Output

Table 2-1. Pin Descriptions (Continued)

Symbol	Description	Asserted State	Type
V_{CC}, V_{CC} I/O	**DEVICE POWER SUPPLY:** The V_{CC} and V_{CC} I/O pins are used to supply the source voltage to the device. The V_{CC} and V_{CC} I/O pins have to be connected to the same supply voltage. Each V_{CC} and V_{CC} I/O pin requires a separate decoupling capacitor to GND. 1 µF ceramic capacitors are recommended. Operations at invalid V_{CC} voltages may produce spurious results and should not be attempted.	-	Power
GND, GND I/O	**GROUND:** The ground reference for the power supply. GND and GND I/O should be connected to the system ground.	-	Power
$\overline{\text{RESET}}$	**RESET:** A low state on the reset pin ($\overline{\text{RESET}}$) terminates the operation in progress and resets the internal state machine to an idle state. The device remains in the reset condition as long as a low level is present on the $\overline{\text{RESET}}$ pin. Normal operation can resume once the $\overline{\text{RESET}}$ pin is brought back to a high level. See Section 12.11 for details about the device operation when $\overline{\text{RESET}}$ pin is engaged. The device incorporates an internal power-on reset circuit, so there are no restrictions on the $\overline{\text{RESET}}$ pin during power-on sequences. If this pin and feature is not utilized, then it is recommended that the $\overline{\text{RESET}}$ pin is driven high externally. It has an internal pull-up, and may alternatively be left open. The $\overline{\text{RESET}}$ pin is not required for operation of the device. The JEDEC Standard Hardware Reset (In-Band Reset) function described in Section 12.10 provides the same functions without requiring a dedicated pin. The $\overline{\text{RESET}}$ pin is included for compatibility with older systems. For new designs, the JEDEC Standard Hardware Reset (In-Band Reset) is recommended. The $\overline{\text{RESET}}$ pin may not be included in all package options.	Low	Input

Figure 2-1. 24-ball 6x8 mm BGA Pinout

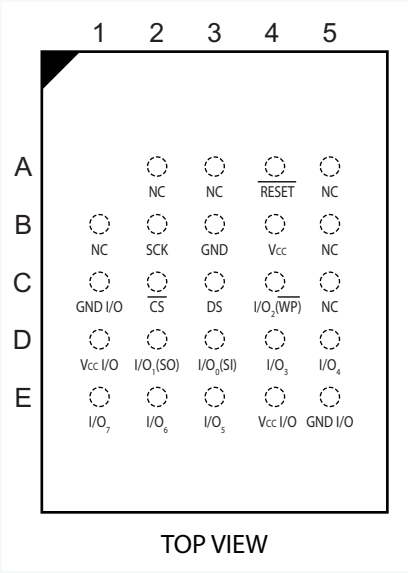


Figure 2-2. 46-ball WLCSP Pinout

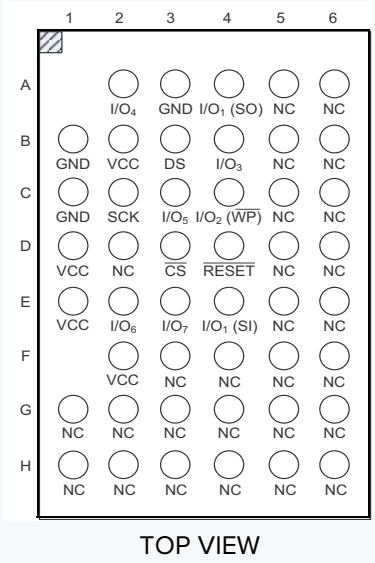
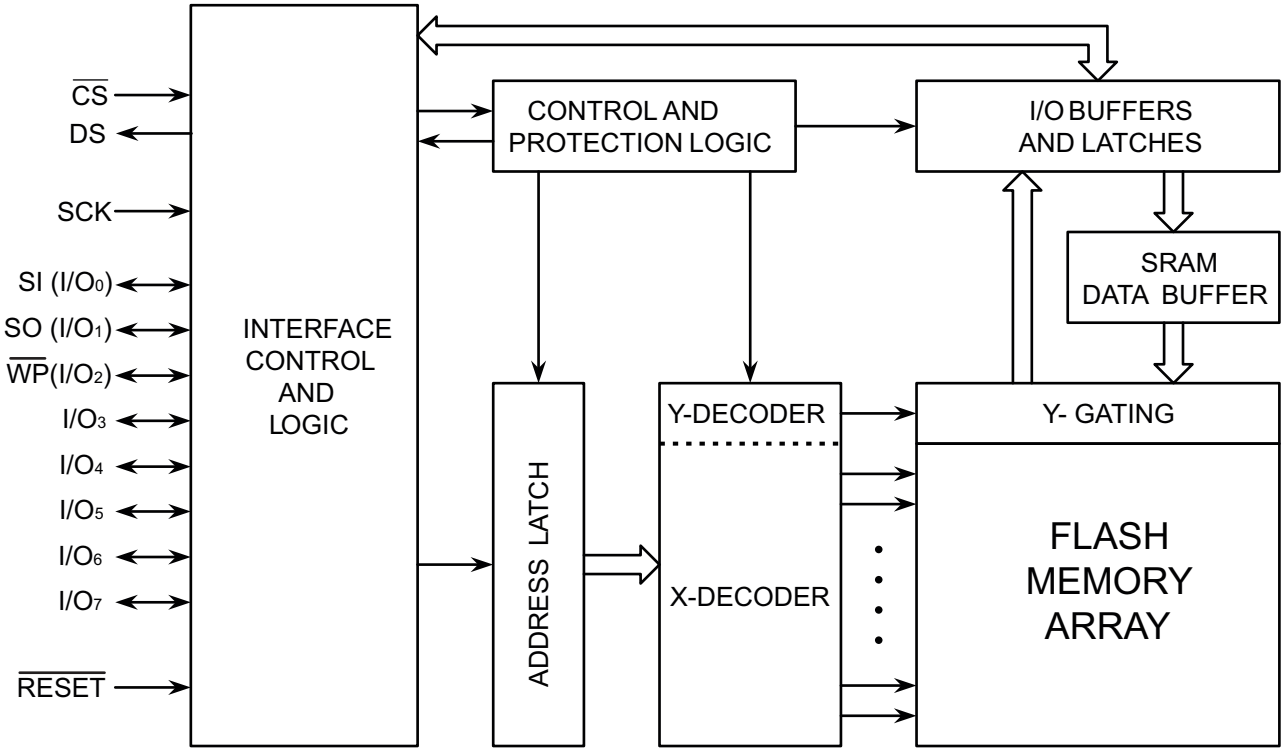


Figure 2-3. QFN Pinout

Contact Adesto for pinout and availability

3. Block Diagram

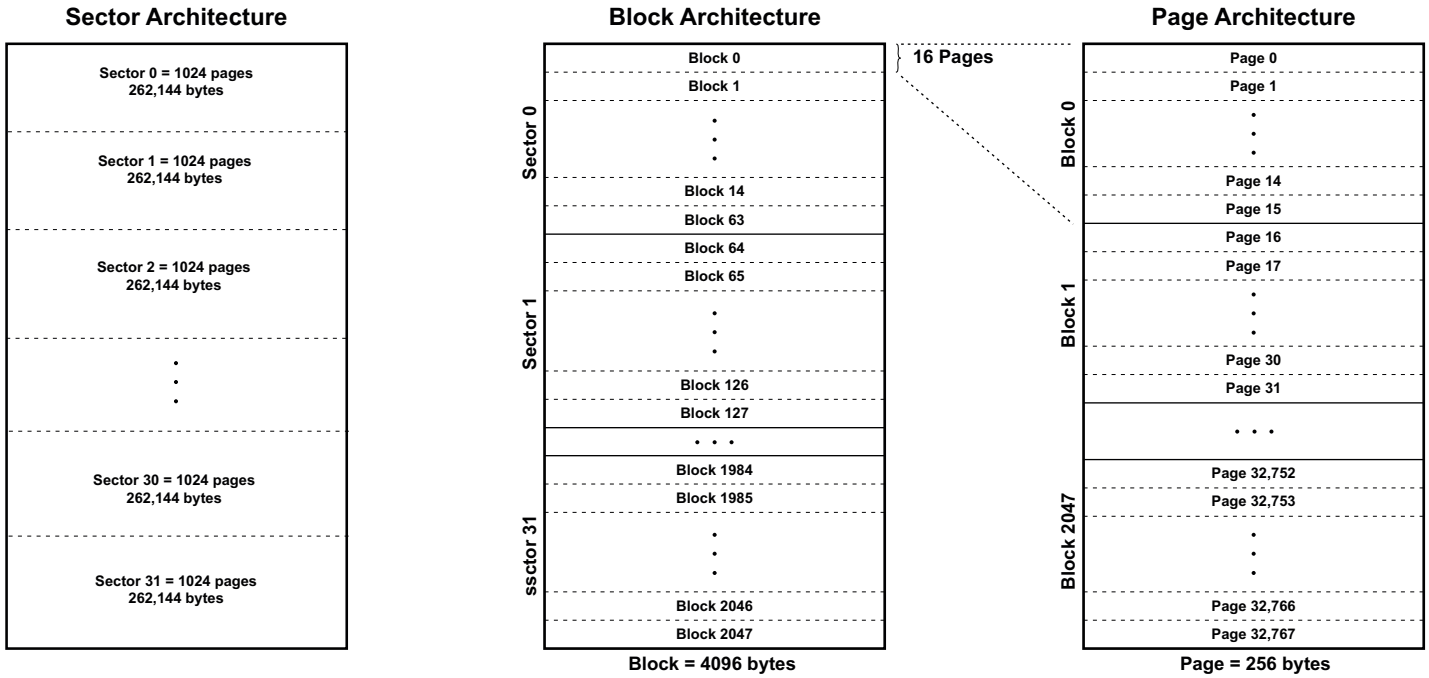
Figure 3-1. Block Diagram



4. Memory Array

To provide the greatest flexibility, the memory array of the ATXP064B is divided into three levels of granularity comprising of sectors, blocks, and pages. The size of the erase blocks is optimized for both code and data storage applications, allowing both code and data segments to reside in their own erase regions. Figure 4-1, Memory Architecture Diagram, illustrates the breakdown of each level and details the number of pages per sector and block. Program operations to the memory array 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 level or at 3 different block size levels.

Figure 4-1. Memory Architecture Diagram

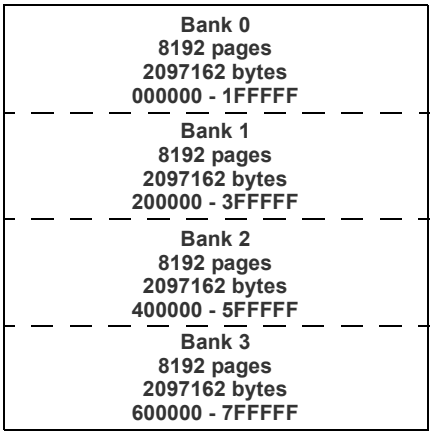


4.1 Read-While-Write Memory Banks

For Read-While-Write operations, the memory array is divided into 4 memory banks of 16 Mbits (or 2 MBytes) each as shown in Figure 4-2, Read-While-Write Memory Banks. While an erase or program operation is taking place in one bank, a read operation can take place in any of the others.

See Section 7.3, Read-While-Write, for more details about using Read-While-Write operations.

Figure 4-2. Read-While-Write Memory Banks



5. Device Operation

5.1 Standard SPI Mode

The ATXP064B 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 ATXP064B 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 ATXP064B supports the two most common modes, SPI modes 0 and 3 for the standard SPI (1-1-1). For octal mode, only SPI mode 0 is supported. 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.

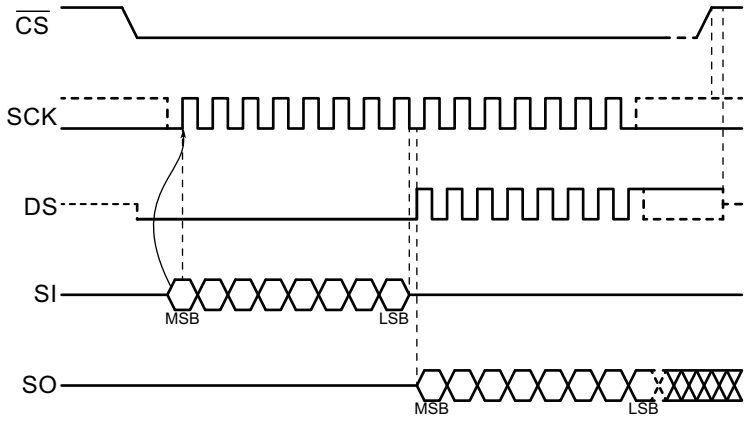
The default SPI mode on power up is the traditional (1-1-1) SPI mode.

Note that the device always wakes up in (1-1-1) SPI mode when coming out of ultra-deep power-down.

The Enter Octal Mode (E8h) command, or directly writing the bit in Status/Control Register Byte 2, is used to enter octal mode. The Return to Standard SPI Mode (FFh) instruction, or directly writing the bit in Status/Control Register Byte 2, can be used to switch the device back to standard SPI mode.

The maximum clock speed f_{CK} supported for standard SPI mode is lower than the maximum clock speed f_{CK} supported for octal mode. See Section 13.4, AC Characteristics - Maximum Clock Frequencies for details.

Figure 5-1. SPI Mode 0 and 3



5.2 Octal Mode

The ATXP064B features an octal mode to further improve throughput. This allows the command byte to be clocked in only 1 clock cycle, and for every clock cycle thereafter, eight address bits can be clocked into the device or eight bits of data can be clocked into or out of the device. Between the address bits and data bits, 4 (or more) dummy cycles are required as shown in Figures 5-3 and 5-5. An additional half dummy cycle is required for octal DDR as described in Figure 5-5. The number of dummy cycles required for each command is described in Tables 6-1 and 6-3, and in the individual command descriptions. For octal mode, only SPI mode 0 is supported.

The ATXP064B supports both Single Data Rate (SDR) and Dual Data Rate (DDR) operations in octal mode. See Section 5.3 for details about DDR mode.

The octal mode is an (8-8-8) mode, using I/O_{7-0} for command, address and data. It does require a mode change, once in this mode the device expects all commands to use this mode. standard SPI mode and octal mode are exclusive. Only one mode can be active at any given time.

Enter Octal Mode (E8h) and Return to Standard SPI Mode (FFh) commands are used to switch between these two modes. Upon power-up the default state of the device is standard SPI mode. To enable octal mode, the volatile Octal Mode Enable (OME) bit in Status/Control register byte 2 is required to be set to 1. The *Write Status / Control Register* commands may also be used to change modes.

The device remains in the same mode while in deep power-down, while it will always wake up in SPI mode when coming out of ultra-deep power-down. Note that when using the Auto Ultra-Deep Power-Down Mode after a Program or Erase Operation command, the device switches from octal mode to standard SPI mode after the program/erase operation has been received. (t_{AUDPD} after $\overline{CS}$ goes high.)

Note: If octal mode is enabled by mistake in a system that does not use this mode, the device can be brought back to standard SPI mode by power cycling the device or by using the *JEDEC Hardware Reset* command described in Section 12.10.

5.2.1 Octal Interface Communication Lines

The octal interface has eleven communication lines, as shown in Figure 5-2:

- CHIP SELECT ($\overline{CS}$): A host-to-device signal. $\overline{CS}$ operates in push-pull mode. Asserting the $\overline{CS}$ pin selects the device. When the $\overline{CS}$ pin is deasserted, the device is deselected and normally be placed in standby mode (not deep power-down mode), and the output pins will be in a high-impedance state. When the device is deselected, data is not accepted on the I/O pins.
  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 does not enter the standby mode until the completion of the operation.
- SCK: This is a host-to-device signal. SCK operates in push-pull mode.
- Data Strobe (DS): This is a device-to-host signal and is output only. DS operates in push-pull mode and is the return of the clock signal.
  In SDR mode, DS is driven high in the first half of the data output cycle, and low in the second half. In this mode, DS is basically be the inverse of SCK, but with a delay. If SCK is low at the end of an operation, DS is high. This is the recommended mode of operation in SDR mode. However, if the host controller requires DS to end low to be able to use this signal correctly, it is also possible to add half of an SCK pulse and end with SCK high and DS low.
  In DDR mode, DS changes value at the edge of each data bit. In this mode, DS is basically be the same value as SCK, but with a delay. As SCK is driven low by the master at the end of a command sequence, DS remains low until $\overline{CS}$ is driven high again. DS is primarily required for fast operation in DDR mode, but is available in other modes as well.
- I/O_{7-0}: Data lines are bidirectional signals. Host and device drivers are operating in push-pull mode. In octal mode, I/O_{7-4} are used together with I/O_{3-0} as bidirectional I/O pins. In these modes, the I/O pins is placed in the high-impedance (high-Z) state whenever the device is deselected ($\overline{CS}$ is deasserted).

Figure 5-2. Octal Interface Bus Circuitry Diagram

5.2.2 Programmable Device Output Driver

The bus capacitance of each line of the octal interface bus is the sum of the bus master capacitance, the bus capacitance itself and the capacitance of each inserted device. The sum of host and bus capacitance are fixed for one application, but may vary between different applications. The device load may vary in one application with each of the inserted devices.

The IOD2:0 bits in the I/O Pin Drive Strength Control Register are used to configure the output driver strength.

Figure 5-3. Single Data Rate (SDR) Operation in Octal Mode — Read Operations

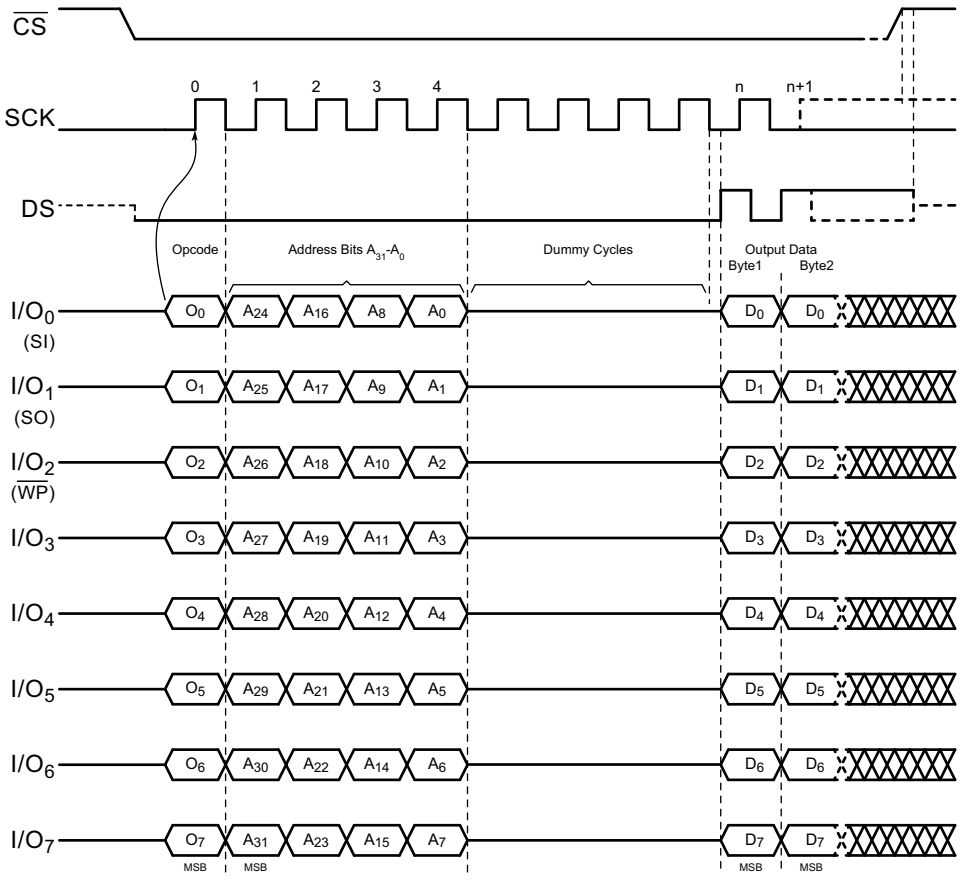
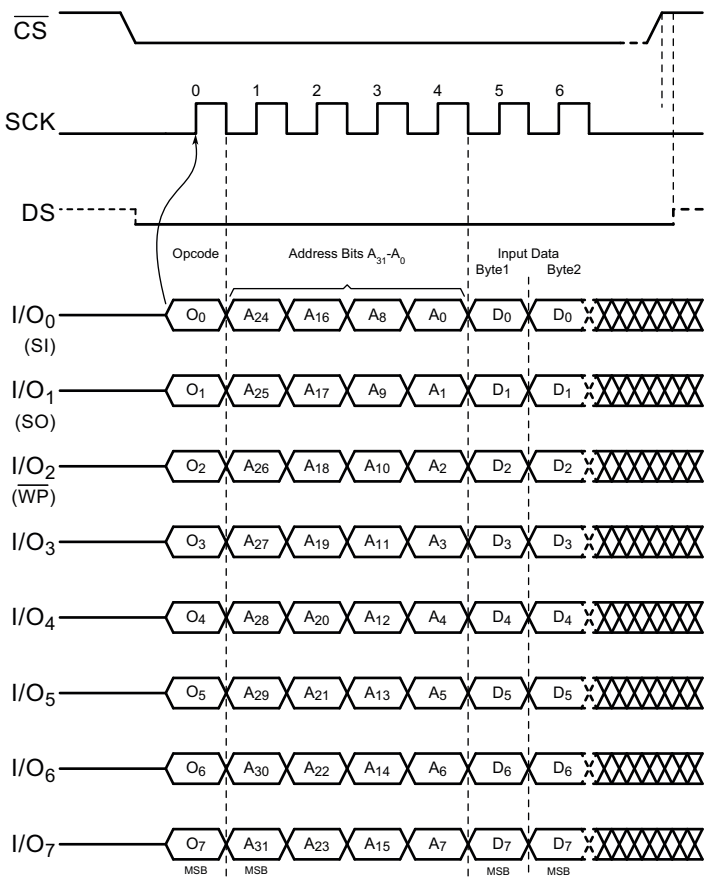


Figure 5-4. Single Data Rate (SDR) Operation in Octal Mode — Write Operations



5.3 Dual Data Rate Operation

The ATXP064B allows Dual Data Rate (DDR) operation in octal mode for additional increase of throughput speed. Dual data rate operation is enabled by setting the $\overline{SDR}$/DDR bit in Status/Control Register Byte 2.

Operation using DDR mode is identical to using Single Data Rate (SDR) mode, except that both edges of the clock are used for clocking address and data. Commands are clocked on the rising edge of SCK, requiring a whole clock cycle also in the DDR mode. In addition, for DDR mode 4 (or more) dummy cycles are required as shown in Figures 5-5 and 5-6. An additional half dummy cycle is required for Octal DDR as described in Figure 5-5.The number of dummy cycles required for each command is described in Tables 6-1 and 6-3, and in the individual command descriptions. For read commands to the Flash memory array, the number of dummy cycles is dependent on the clock frequency used. See Section 11.3, Status/Control Register Byte 3, for details.

Note that for DDR operation in octal mode, all read and write operations operate on an even number of bytes. The minimum number of SCK pulses for clocking data is one, and as data is clocked on both edges, two bytes are read or written. Address bit A_0 has to be always 0 in this mode to ensure correct alignment of the two bytes read or written. If address bit A_0 is set to 1, it is ignored and treated as A_0 = 0 by the device. For register operations that operate on a single 8-bit register, a read operation outputs the same register value on the first clock edge and the data on the second clock edge is unpredictable. For write operations on a single 8-bit register read operation, only the value on the first clock edge is written and the data on the second clock edge is ignored.

For high speed operation, the signal delays in the system need to be taken into account. The signal path going from the SPI master to the memory device and back may cause so much skew between the SCK and the data that the data read from the memory device is no longer synchronized with the outgoing SCK signal as seen by the SPI master.

To counter this problem, the SCK signal is returned as DS for read operations, providing a clock signal to the master that is synchronous to the data coming from the memory device.

The DS signal is also available in all other modes.

Figure 5-5. Dual Data Rate (DDR) Operation in Octal Mode — Read Operations

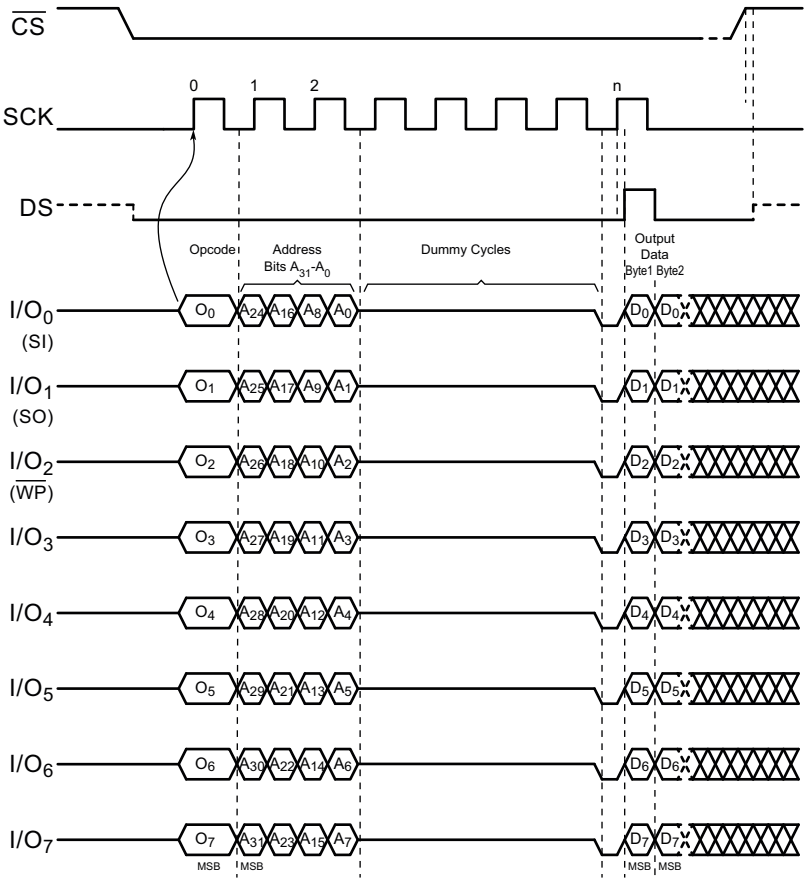
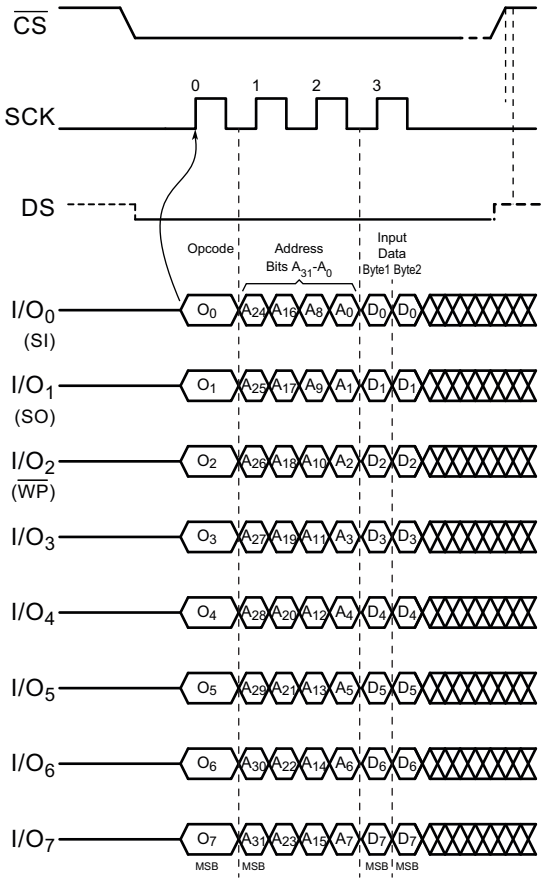


Figure 5-6. Dual Data Rate (DDR) Operation in Octal Mode — Write Operations



6. Commands and Addressing

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

Device behavior is not defined for commands not supported by the ATXP064B. Commands not described in this document should therefore be avoided.

If the $\overline{CS}$ pin is deasserted before complete command and address information is sent to the device, then no operation will be performed and the device simply returns to the idle state and waits for the next operation.

Addressing of the device requires a total of four bytes of information to be sent, representing address bits A_{31} - A_0. Since the upper address limit of the ATXP064B memory array is 7FFFFFh, address bits A_{31} - A_{23} must be zero for proper operation.

Table 6-1. Command Listing 1 — Commands Used in All Modes

Command	Command	Max Clock Freq in SPI Mode	Max Clock Freq. in Octal Modes	Address Bytes	Dummy Bytes in SPI Mode	Dummy Cycles in Octal Mode		Data Bytes	Section Link
						SDR	DDR		
Read Commands									
Read Array	0Bh	0000 1011	66 MHz	133 MHz[1]	4	1	8+ [1]	8+[1]	1+	7.1
Program and Erase Commands									
Block Erase (4 Kbytes)	20h	0010 0000	66 MHz	133 MHz	4	0	0	0	0	8.4
Block Erase (32 Kbytes)	52h	0101 0010	66 MHz	133 MHz	4	0	0	0	0	8.4
Block Erase (64 Kbytes)	D8h	1101 1000	66 MHz	133 MHz	4	0	0	0	0	8.4
Chip Erase	60h	0110 0000	66 MHz	133 MHz	0	0	0	0	0	8.5
	C7h	1100 0111	66 MHz	133 MHz	0	0	0	0	0	8.5
Byte/Page Program (1 - 256 Bytes)	02h	0000 0010	66 MHz	133 MHz	4	0	0	0	1+	8.1
Buffer Write	84h	1000 0100	66 MHz	133 MHz	4	0	0	0	1+	8.2
Buffer to Main Memory Page Program without Built-In Erase	88h	1000 1000	66 MHz	133 MHz	4	0	0	0	0	8.3
Program/Erase Suspend	B0h	1011 0000	66 MHz	133 MHz	0	0	0	0	0	8.6
Program/Erase Resume	D0h	1101 0000	66 MHz	133 MHz	0	0	0	0	0	8.7
Protection Commands									
Write Enable	06h	0000 0110	66 MHz	133 MHz	0	0	0	0	0	9.1
Write Disable	04h	0000 0100	66 MHz	133 MHz	0	0	0	0	0	9.2
Protect Sector	36h	0011 0110	66 MHz	133 MHz	4	0	0	0	0	9.3
Unprotect Sector	39h	0011 1001	66 MHz	133 MHz	4	0	0	0	0	9.4
Read Sector Protection Registers	3Ch	0011 1100	66 MHz	133 MHz	4	0	4	4	1+	9.6
Security Commands									
Program OTP Security Register	9Bh	1001 1011	66 MHz	133 MHz	4	0	0	0	1+	10.1
Read OTP Security Register	77h	0111 0111	66 MHz	133 MHz	4	1	8+[1]	8+[1]	1+	10.2
Status/Control Register Commands									
Read Status/Control Registers	65h	0110 0101	66 MHz	133 MHz	1	1	4	3 [2]	1+	11.5
Read Status Register Byte 1	05h	0000 0101	66 MHz	133 MHz	0	0	4	4	1+	11.6

Table 6-1. Command Listing 1 — Commands Used in All Modes (Continued)

Command	Command		Max Clock Freq in SPI Mode	Max Clock Freq. in Octal Modes	Address Bytes	Dummy Bytes in SPI Mode	Dummy Cycles in Octal Mode		Data Bytes	Section Link
							SDR	DDR		
Active Status Interrupt	25h	0010 0101	66 MHz	133 MHz	0	1	4	4	0	11.10
Write Status/Control Registers	71h	0111 0001	66 MHz	133 MHz	1	0	0	0	1+	11.7
Write Status Register Byte 1	01h	0000 0001	66 MHz	133 MHz	0	0	0	0	1	11.8
Write Status Register Byte 2	31h	0011 0001	66 MHz	133 MHz	0	0	0	0	1	11.9
Miscellaneous Commands										
Terminate Operation	F0h	1111 0000	66 MHz	133 MHz	0	0	0	0	1 (D0h)	12.8
Reset Enable	66h	0110 0110	66 MHz	133 MHz	0	0	0	0	0	12.9
Reset	99h	1001 1001	66 MHz	133 MHz	0	0	0	0	0	12.9
Deep Power-Down	B9h	1011 1001	66 MHz	133 MHz	0	0	0	0	0	12.2
Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh)	ABh	1010 1011	66 MHz	133 MHz	0	0	0	0	0	12.4
Ultra-Deep Power-Down	79h	0111 1001	66 MHz	133 MHz	0	0	0	0	0	12.5
Read SFDP	5Ah	0101 1010	50 MHz	50 MHz	3	1	8	8(1)	1+	12.17

1. See Table 11-8, Dummy Clock Cycles and Maximum Operating Frequency
2. Octal DDR mode adds an additional ½ dummy cycle to align the first byte of data to the rising edge of SCK/DS

Table 6-2. Command Listing 2 — Commands Only Used in Standard SPI Mode

Command	Command		Max Clock Frequency (MHz)	Address Bytes	Dummy Bytes in SPI Mode	Data Bytes	Section Link
Read Commands							
Read Array (1-1-1)	03h	0000 0011	50	3	0	1+	7.1
Read Array (1-1-1)	13h	0001 0011	50 MHz	4	0	1+	7.1
Buffer Read	D4h	1101 0100	50	4	1	1+	7.4
Miscellaneous Commands							
Read Manufacturer and Device ID	9Fh	1001 1111	66	0	0	12	12.1
Enter Octal Mode (1-1-1 to 8-8-8) (1)	E8h	1110 1000	66	0	0	0	12.15

1. The command is entered in (1-1-1) mode, the device will change to (8-8-8) mode afterwards.

Table 6-3. Command Listing 3 — Commands Only Used in Octal Mode (8-8-8)

Command	Command		Maximum Clock Frequency (MHz) Octal Mode	Address Bytes	Dummy Cycles	Data Bytes	Section Link
Read Commands							
Burst Read with Wrap	0Ch	0000 1100	133(1)	4	8+ (1)	1+	7.2
Miscellaneous Commands							
Echo	AAh	1010 1010	150	1 Value Byte	4 (2)	1	12.13
Echo with inversion	A5h	1010 0101	150	1 Value Byte	4(2)	1	12.12
Return to Standard SPI Mode	FFh	1111 1111	150	0	0	0	12.16

1. For more information, refer to Table 11-8, Dummy Clock cycles and Maximum Operating Frequency.
2. Octal DDR mode adds an additional 1/2 dummy cycle to align the first byte of data to the rising edge of SCK/DS.

7. Read Commands

7.1 Read Array (0Bh, 13h, and 03h)

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

Three commands (0Bh, 13h and 03h) can be used for the Read Array command. The use of each command depends on the maximum clock frequency that will be used to read data from the device and the mode it operates in. The 0Bh command can be used for any mode at any clock frequency up to the maximum specified by f_{CLK}. The 13h and 03h commands can only be used in SPI mode, and can be used for lower frequency read operations up to the maximum specified by f_{RDLF}. To be compatible with older host controllers, the 03h command is using 3 address bytes, while the 13h command is using 4. This allows the host controller to read data from the flash memory without prior knowledge about what type of flash device it is connected to.

To perform the read array operation with 3-byte addressing using the 03h command, the CS pin must first be asserted and the 03h command clocked into the device. After the command has been clocked in, the 3 address bytes must be clocked in to specify the starting address location of the first byte to read within the memory array.

To perform the Read Array operation with 4-byte addressing using the 13h command, the $\overline{CS}$ pin must first be asserted and the 13h command clocked into the device. After the command has been clocked in, the 4 address bytes must be clocked in to specify the starting address location of the first byte to read within the memory array.

To perform the fast read array operation with 4-byte addressing using the 0Bh command, the $\overline{CS}$ pin must first be asserted and the command 0Bh clocked into the device. After the command has been clocked in, the 4 address bytes must be clocked in to specify the starting address location of the first byte to be read within the memory array. After the command, one or more additional dummy bytes need to be clocked into the device. For the SPI mode, one dummy byte is used. For

octal mode, 8 or more dummy cycles are used as shown in Table 6-1 and Table 11-8. Note that the table refers to full clock cycles, also for DDR modes. Half cycles are not used for dummy cycles, the settings are a multiple of 2 full cycles.

Note that for DDR operation in octal mode, all read operations operate on an even number of bytes. The minimum number of SCK pulses for clocking data is one, and as data is clocked on both edges, two bytes are read. Address bit A_0 has to be always 0 in this mode to ensure correct alignment of the two bytes read or written. If address bit A_0 is set to 1, it is ignored and treated as A_0 = 0 by the device.

After the address bytes (and the dummy bytes if using command 0Bh) have been clocked in, additional clock cycles result in data being output on the I/O pin(s). The data is always output with the MSB of a byte first. When the last byte (7FFFFFh) of the memory array has been read, the device continues reading back at the beginning of the array (000000h). No delays are incurred when wrapping around from the end of the array to the beginning of the array.

Deasserting the $\overline{CS}$ pin terminates the read operation and put the I/O pins into high-impedance state. The $\overline{CS}$ pin can be deasserted at any time and does not require a full byte of data be read.

Figure 7-1. Read Array with 3-byte Address — 03h Command

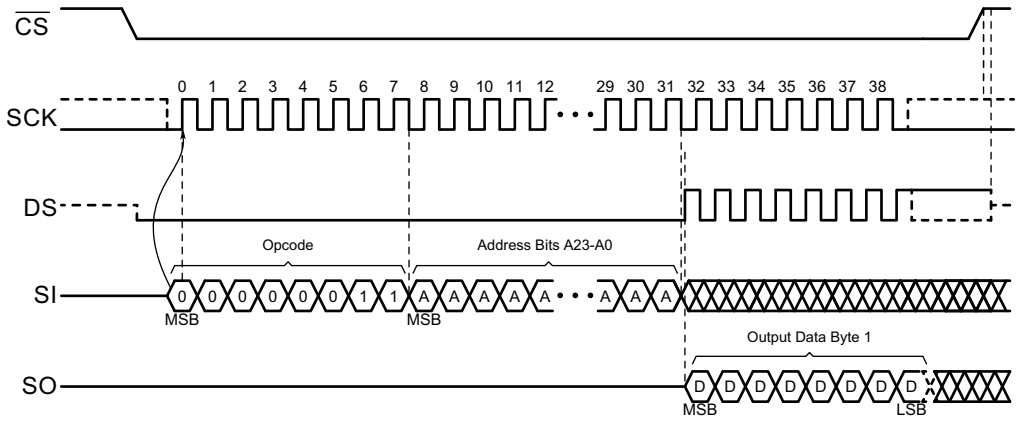


Figure 7-2. Read Array with 4-byte address - 13h Command

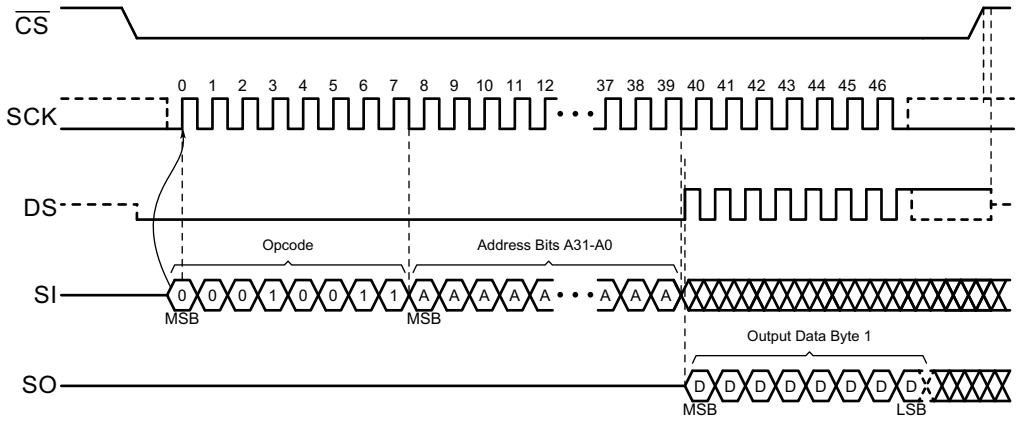


Figure 7-3. Read Array — 0Bh Command

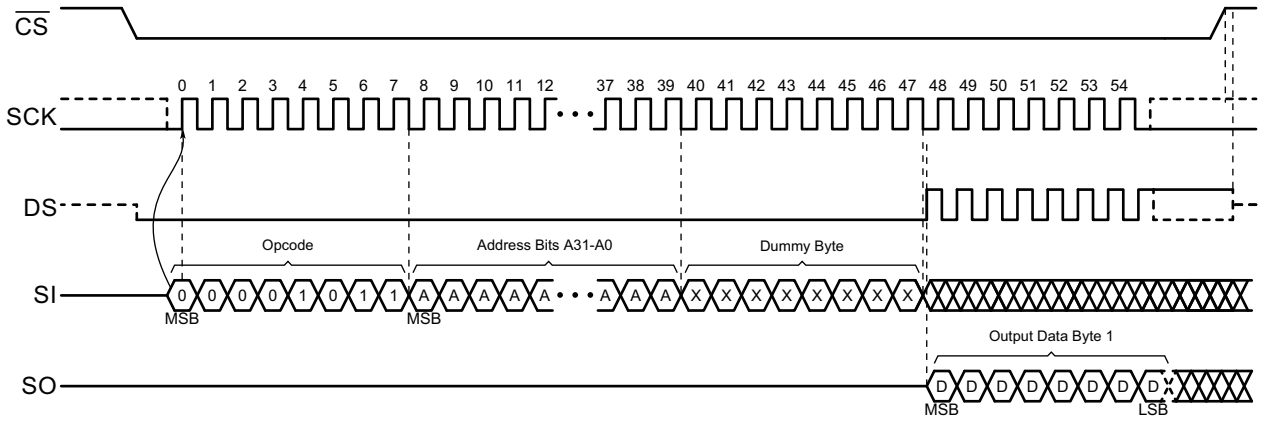


Figure 7-4. Read Array — 0Bh Command in Octal Mode SDR

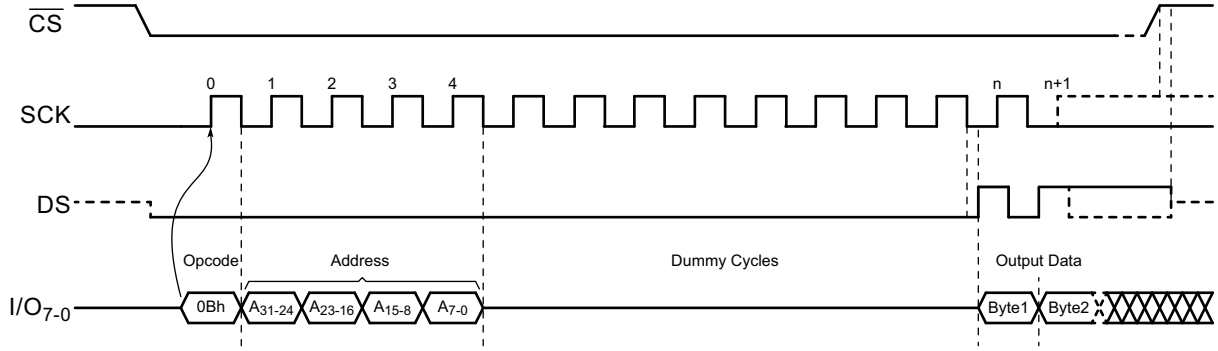
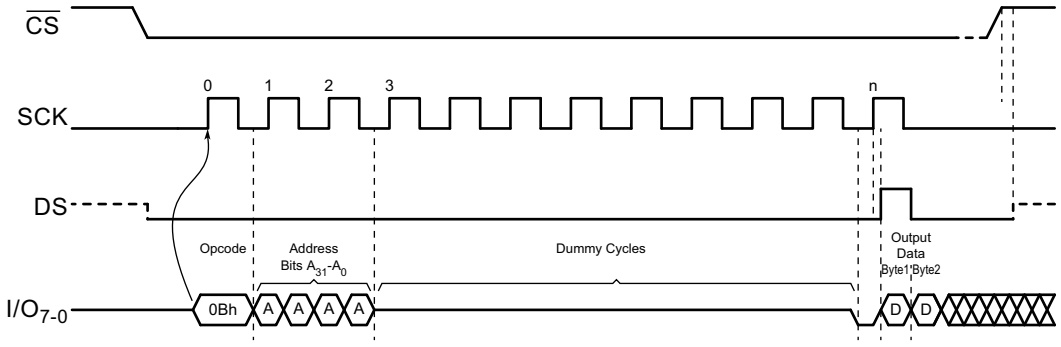


Figure 7-5. Read Array — 0Bh Command in Octal Mode DDR



7.2 Burst Read with Wrap (0Ch)

In Octal Mode, the “Burst Read with Wrap (0Ch)” instruction is used to perform the read operation with “Wrap Around” feature.

Microcontroller units (MCU’s) with cache benefit from this feature for eXecute-in-Place (XiP) operations, as it guarantees an efficient way of reading a whole cache line in one burst regardless of which byte in the cache line the read starts from. This improves code execution performance in the MCU system, as the MCU first receives the command or data it requires at that instant, and then the remainder of the cache line, without requiring additional commands or addresses to be sent.

The continuous mode operation further improves the MCU performance. This mode allows the MCU to directly load the following cache line if this is required, again without requiring additional commands or addresses to be sent. This improves the performance of a typical MCU system by 40% or more without increasing system clock speed.

As an example; The timing for fetching the next consecutive 16-byte cache line on a Flash device with 14 dummy SPI cycles for eXecute-in-Place (XiP) operations.

In octal SDR mode, the number of cycles will be reduced from 35 to 16:

Command (byte):	1 clock cycle (eliminated for the next consecutive line)
Address (4 bytes):	4 clock cycles (eliminated for the next consecutive line)
Dummy cycles:	14 clock cycles (eliminated for the next consecutive line)
Data (16 Bytes):	16 clock cycles

In octal DDR mode, the number of cycles will be reduced from 25 to 8:

Command (byte):	0.5 clock cycles (eliminated for the next consecutive line)
Address (4 bytes):	2 clock cycles (eliminated for the next consecutive line)
Dummy cycles:	14.5 clock cycles (eliminated for the next consecutive line)
Data (16 Bytes):	8 clock cycles

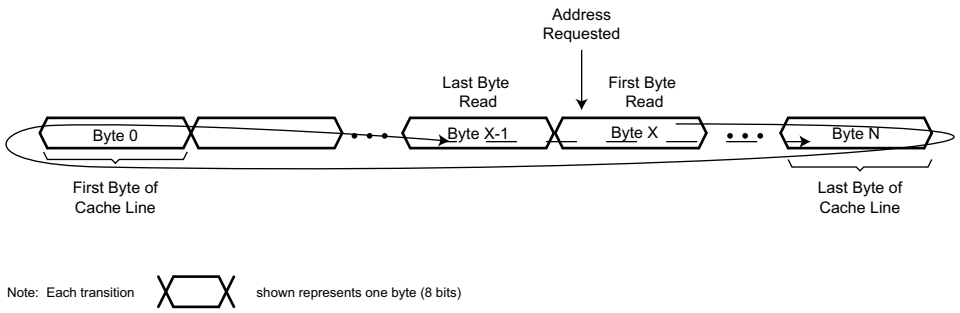
The behavior of the “Burst Read with Wrap (0Ch)” instruction is controlled by the W7-W5 bits in as described in Table 11-6, Status/Control Register Byte 3.

The wrap length and wrap mode parameters set by the W7 - W5 bits in one SPI mode is still valid in the other SPI mode and can also be re-configured by the “Write Status Registers (71h)” instruction. See Table 11-7 for details on wrap bit settings.

Once W7 - W5 are set, all the following “Burst Read with Wrap (0Ch)” instructions use the W7 - W5 setting to access the 8/16/32-byte section within any page.

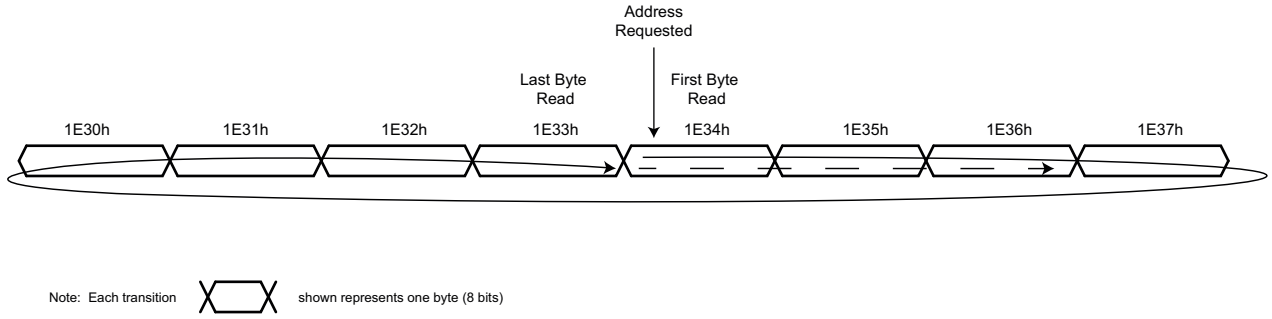
If W7 is cleared, the device operates in wrap-around mode. The “Burst Read with Wrap (0Ch)” instruction will read to the end of the fixed length section (cache line), wrap around to the beginning of the section and continue reading the same cache line continuously for as long as additional clock pulses are sent to SCK. There is no delay caused by the wrap around, the first byte of the cache line follows immediately after the last. This operation is shown in Figure 7-6. This mode of operation is included for compatibility with other XiP devices, and for MCUs that do not support the continuous mode.

Figure 7-6. Burst Read with Wrap, Wrap Around Mode (W7 = 0)



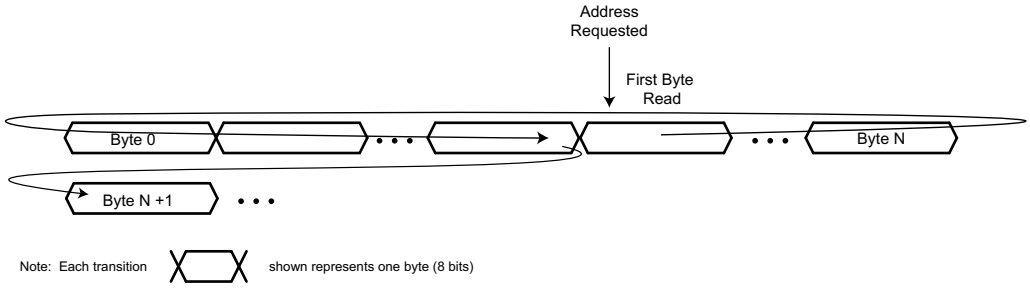
An example of this operation in wrap-around mode is shown in Figure 7-7. Here the wrap length is set to 8 for simplicity. The requested address is 1E34h. The device therefore reads 1E34h first, followed by 1E35h, 1E36h, and 1E37h. This is the end of the 8-byte cache line. The next address to be read out is 1E30h, followed by 1E31h, 1E32h, and 1E33h. If the read operation continues, the device outputs 1E34h again, and this loop continues for as long as the device is clocked.

Figure 7-7. Burst Read with Wrap Example, Wrap Around Mode (W7 = 0), Wrap Length 8, Requested Address = 1E34h



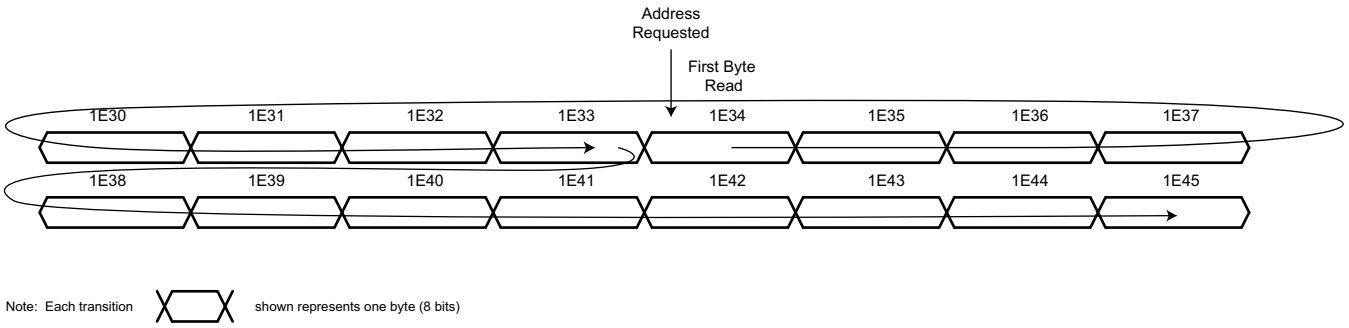
If W7 is set, the device operates in continuous mode. This is the preferred mode for maximum XiP performance for cached MCUs. For the first cache line, the continuous mode operates the same way as the wrap-around mode. The “Burst Read with Wrap (0Ch)” instruction reads to the end of the line, then wraps around to the beginning and continues reading the same line until all bytes of the first line have been read out *once*. In the next clock cycle, the device starts reading at the beginning of the *next* line as shown in Section 7-8. The device reads continuously as long as SCK is clocked. There is no delay caused by the wrap-around, the first byte of the cache line follows immediately after the last. There is also no additional delay caused by the jump to the next cache line, the first byte of the following cache line follows immediately after the last byte read from the previous one, independent of where in the cache line the jump is coming from.

Figure 7-8. Burst Read with Wrap, Continuous Mode (W7 = 1)



An example of this operation in continuous mode is shown in Figure 7-9. As in the above example, the wrap length is set to 8, and the requested address is 1E34h. The device reads 1E34h first, followed by 1E35h, 1E36h, and 1E37h. It then wraps around, and the next address to be read out is 1E30h, followed by 1E31h, 1E32h, and 1E33h. If the read operation continues, the device will then output 1E38h, followed by 1E39h and so on. The device reads continuously as long as the device is clocked.

Figure 7-9. Burst Read with Wrap Example, Continuous Mode (W7 = 1), Wrap Length 8, Requested Address = 1E34h



If the MCU does not immediately need or cannot immediately handle the next cache line after the first one has finished, it can stop the clock while keeping the $\overline{CS}$ line low. It can resume clocking once it is ready to receive the next cache line.This of course assumes the MCU but does not need to issue any other commands to the memory device between receiving the two cache lines.

The value of the I/O lines while the clock is stopped depend on the value of the clock:

- If the clock is stopped high in SDR mode, the I/O lines output the value of the last byte of the previous cache line. The entire byte is output in octal mode.
- If the clock is stopped low in SDR mode, the I/O lines output the value of the first byte of the next cache line. The entire byte is output in octal mode.
- If the clock is stopped high in DDR mode, the I/O lines output the value of the first byte of the next cache line. The entire byte is output in octal mode.
- If the clock is stopped low in DDR mode, the I/O lines output the value of the last byte of the previous cache line. The entire byte is output in octal mode.

Figure 7-10. Burst Read with Wrap in Octal Mode — SDR

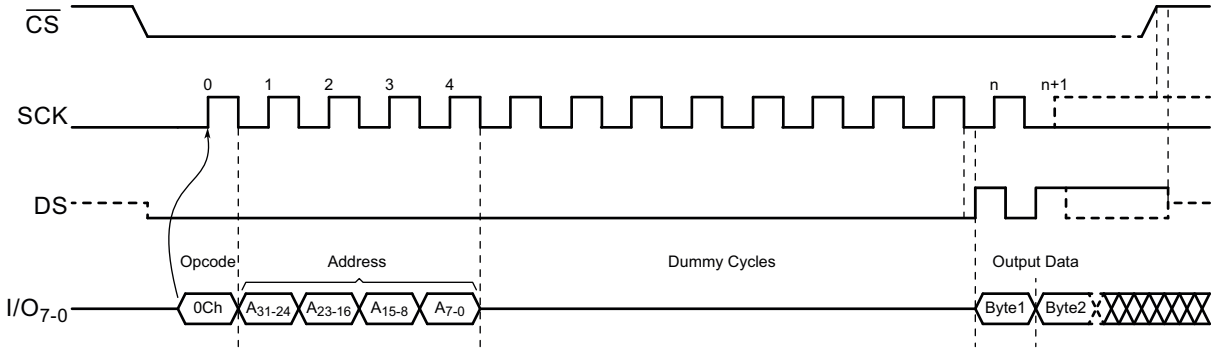
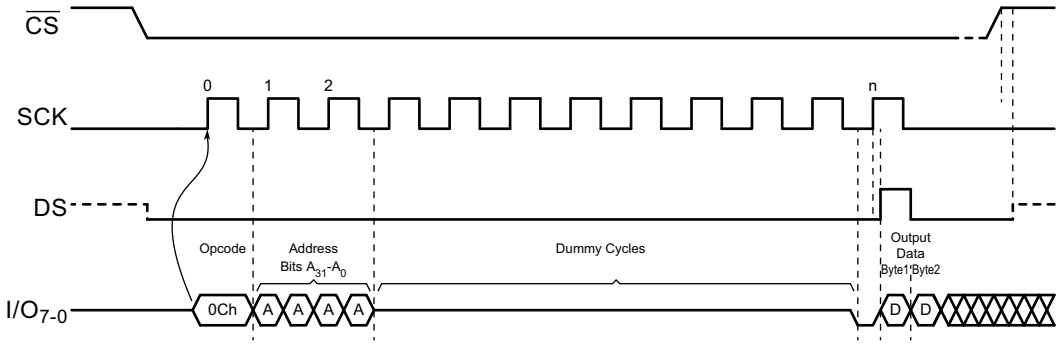
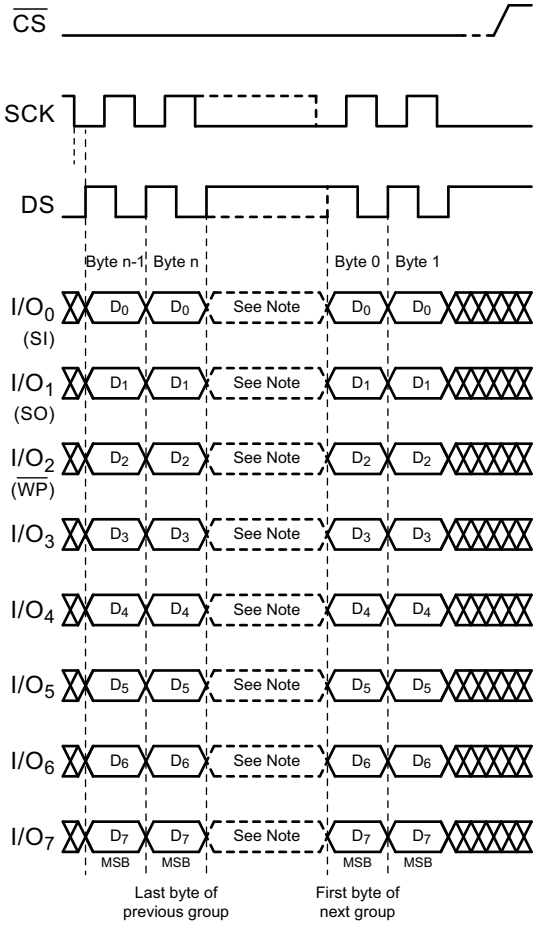


Figure 7-11. Burst Read with Wrap in Octal Mode — DDR



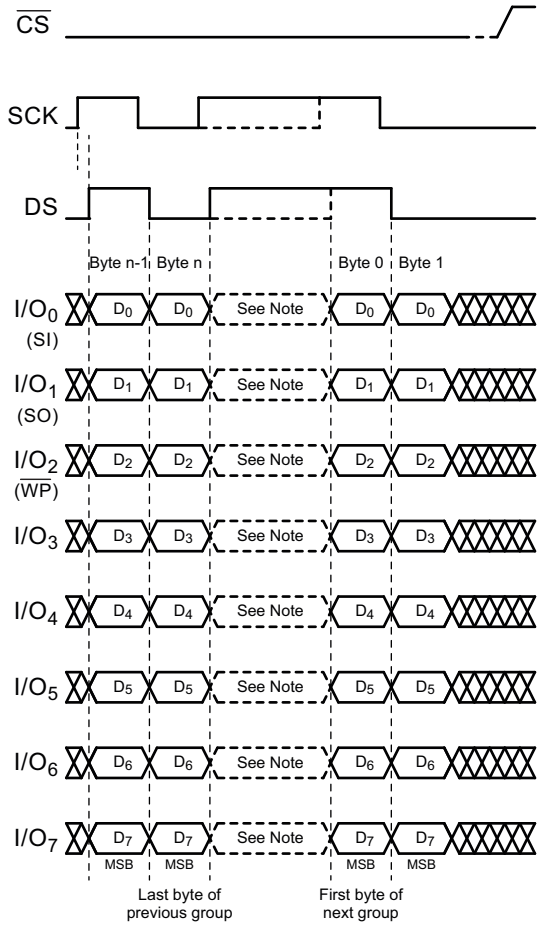
See note on page 22 regarding the value of the I/O lines while the clock is stopped.

Figure 7-12. Timing Diagram for Two Consecutive Accesses in Octal Mode — SDR



See note on page 22 regarding the value of the I/O lines while the clock is stopped.

Figure 7-13. Timing Diagram for Two Consecutive Accesses in Octal Mode — DDR



See note on page 22 regarding the value of the I/O lines while the clock is stopped.

7.3 Read-While-Write

For (optional) Read-While-Write operations, the memory array is divided into 4 memory banks of 16 Mbits (or 2 MBytes) each as shown in Figure 4-2, Read-While-Write Memory Banks. While an Erase or Program operation is taking place in one bank, a Read operation can take place in any of the other. No configuration settings are required for the Read-While-Write operations.

A complete list of which commands can be issued while a Program or Erase operation is in progress is found in Table 8-1.

If a read command is issued that attempts to read from the same bank where a write operation is in progress, the result of the read command is unpredictable.

7.4 Buffer Read (D4h)

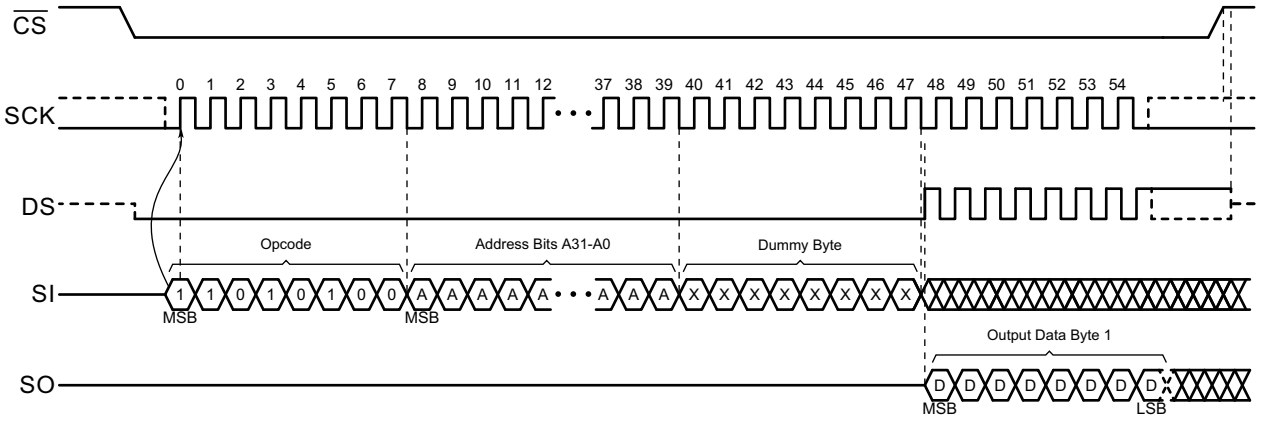
The SRAM data buffers can be accessed independently from the main memory array, and utilizing the Buffer Read command allows data to be sequentially read directly from the buffer.

To perform a Buffer Read, the command D4h must be clocked into the device followed by four address bytes comprised of 24 dummy bits and 8 buffer address bits (BFA_7 - BFA_0). Following the address bytes, one dummy byte must be clocked into the device to initialize the read operation. The $\overline{CS}$ pin must remain low during the loading of the command, the address bytes, the dummy byte and the reading of data. When the end of a buffer is reached, the device continues reading back at

the beginning of the buffer. A low-to-high transition on the $\overline{CS}$ pin terminates the read operation and tri-state the output pin (SO).

Note that the Buffer Read command is only allowed in Standard SPI Mode.

Figure 7-14. Buffer Read



8. Program and Erase Commands

8.1 Byte/Page Program (02h)

The *Byte/Page Program* command allows anywhere from a single byte of data to 256 bytes of data to be programmed into previously erased memory locations. An erased memory location is one that has all eight bits set to the logic "1" state (a byte value of FFh). Before a *Byte/Page Program* command can be started, the *Write Enable* command must have been previously issued to the device (see "Write Enable (06h)" on page 40) to set the Write Enable Latch (WEL) bit of the Status/Control register to a logic "1" state.

To perform a Byte/Page Program command, an command of 02h must be clocked into the device followed by the four 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 is stored in an internal buffer.

If the starting memory address denoted by A_{31} - A_0 does not fall on an even 256-byte page boundary (A_7 - A_0 are not all 0), then special circumstances regarding which memory locations to be programmed apply. In this situation, any data that is sent to the device that goes beyond the end of the page wraps around back to the beginning of the same page. For example, if the starting address denoted by A_{31} - A_0 is 0000FEh, and three bytes of data are sent to the device, then the first two bytes of data is programmed at addresses 0000FEh and 0000FFh while the last byte of data is programmed at address 000000h. The remaining bytes in the page (addresses 000001h through 0000FDh) are not programmed and remain in the erased state (FFh). In addition, if more than 256 bytes of data are sent to the device, then only the last 256 bytes sent is latched into the internal buffer.

When the $\overline{CS}$ pin is deasserted, the device takes the data stored in the internal buffer and programs it into the appropriate memory array locations based on the starting address specified by A_{31}-A_0 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 are not programmed and remain in the erased state (FFh). The programming of the data bytes is internally self-timed and should take place in a time of t_{PP} or t_{BP} if only programming a single byte.

The four 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 whole byte boundaries (multiples of eight bits); otherwise, the device aborts the operation and no data is programmed into the memory array. In addition, if the memory is in the protected state, then the *Byte/Page Program* command is not executed, and the device returns to the idle state once the $\overline{CS}$ pin has been

deasserted. The WEL bit in the Status/Control register is reset back to the logic “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/Control register can be read and indicates that the device is busy. For faster throughput, it is recommended that the Status/Control register be polled rather than waiting the t_{BP} or t_{PP} time to determine if the data bytes have finished programming. For fastest throughput and least power consumption, it is recommended that the *Active Status Interrupt* command 25h be used. After the initial 16 clocks, no more clocks are required. Once the BUSY cycle is done, SO will be driven low immediately to signal the device has finished programming. At some point before the program cycle completes, the WEL bit in the Status/Control register is reset back to the logic “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 is indicated by the EPE bit in the Status/Control register.

Note that in octal DDR mode, the byte program operation programs two bytes instead of one, and the page program operation programs an even number of bytes. Address bit A_0 has to be 0 in this case.

Figure 8-1. Byte Program

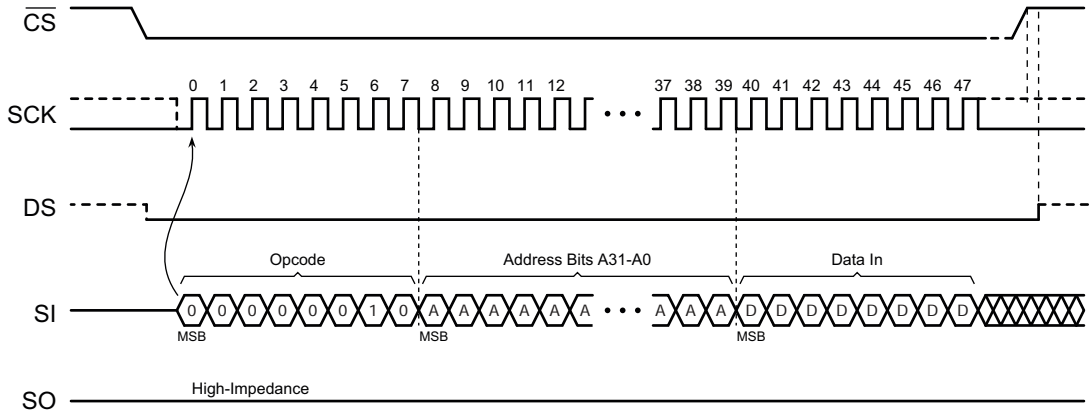


Figure 8-2. Byte Program in Octal Mode — SDR

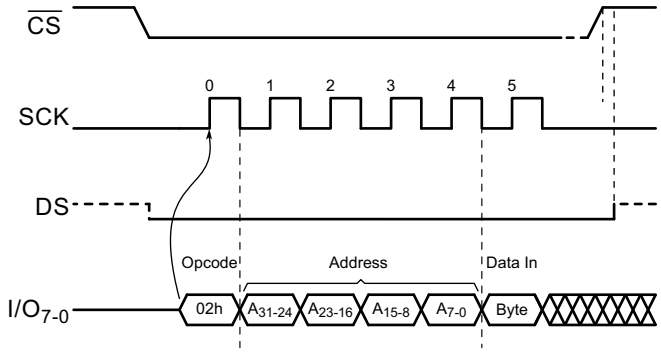
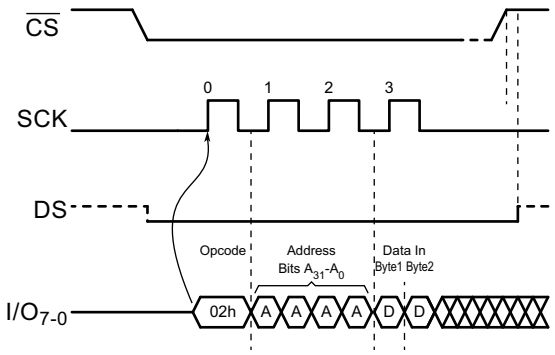


Figure 8-3. Byte Program in Octal Mode — DDR



Note that 2 bytes are programmed when using the *Byte Program* command in octal mode - DDR.

Figure 8-4. Page Program

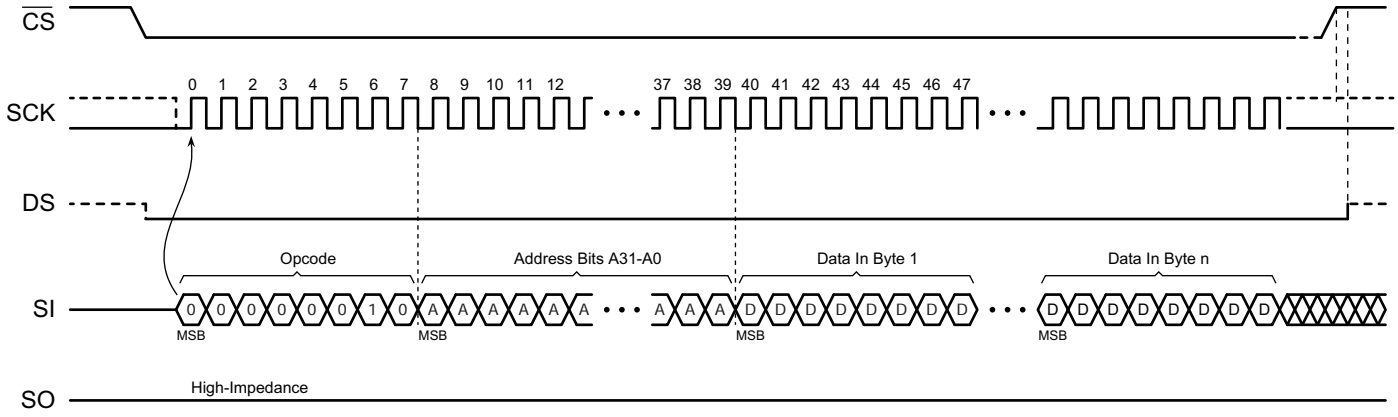


Figure 8-5. Page Program in Octal Mode — SDR

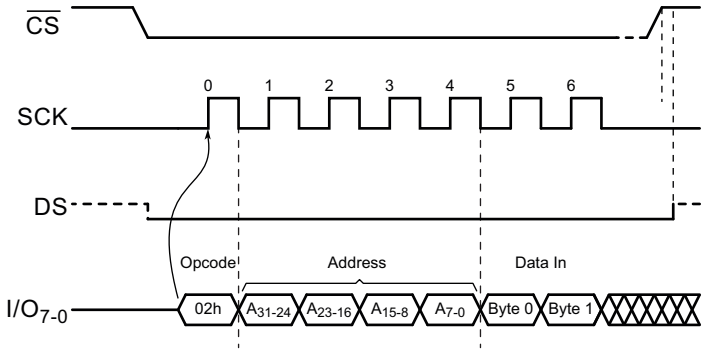
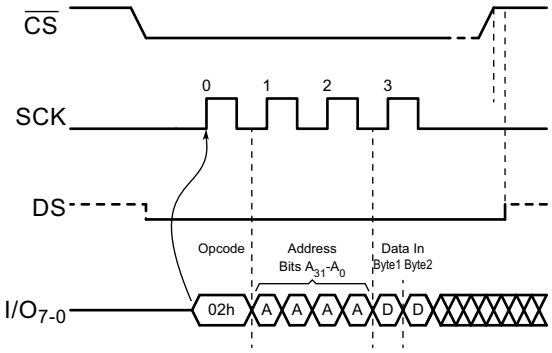


Figure 8-6. Page Program in Octal Mode — DDR



Note that an even number of bytes are programmed when using the *Page Program* command in octal mode - DDR.

8.2 Buffer Write (84h)

Utilizing the Buffer Write (84h) command allows data clocked in from the I/O pins to be written directly into the internal write buffer without starting a write to the Flash array at the same time. This write operation is faster than a regular byte or page write. It is typically used to write small amounts of data at the time to the buffer, and then starting a write operation to the Flash memory when the buffer is full. The Buffer to Main Memory Page Program without Built-In Erase (88h) command is then used to write the buffer to the Flash memory.

To load data into a buffer, the 84h command must be clocked into the device followed by 4 address bytes comprised of 24 dummy bits and 8 buffer address bits (BFA_7 - BFA_0). The eight buffer 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 buffer is reached, the device wraps around back to the beginning of the buffer. Data continues to be loaded into the buffer until a low-to-high transition is detected on the $\overline{CS}$ pin.

Note that the buffer is *not* automatically cleared prior to starting a *Buffer Write* command. This allows multiple *Buffer Write* commands to be performed to update different parts of the buffer before writing the buffer to the Flash memory, or to update the same buffer locations multiple times before writing to Flash memory. Buffer locations which are not updated will contain whatever data is left in the buffer from the previous write operation. The Buffer to Main Memory Page Program without Built-In Erase (88h) command writes the full 256-byte page, even if less than 256 bytes were written to the buffer.

Note also that the Buffer to Main Memory Page Program without Built-In Erase (88h) command should be executed before any Byte/Page Program (02h) command is issued. These commands share the same write buffer, so a Byte/Page Program (02h) command overwrites whatever data is written by the Buffer Write (84h) command.

Figure 8-7. Buffer Write

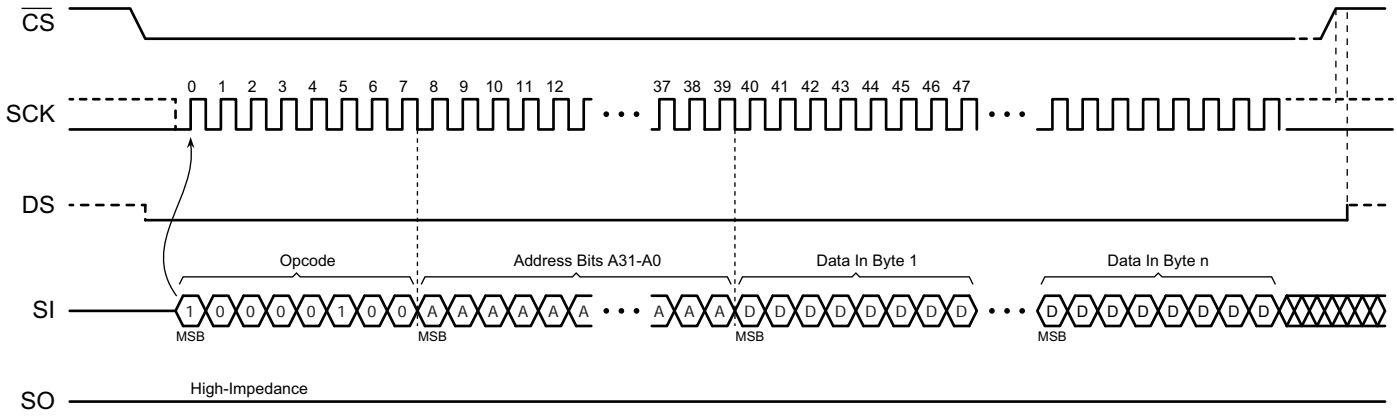


Figure 8-8. Buffer Write in Octal Mode — SDR

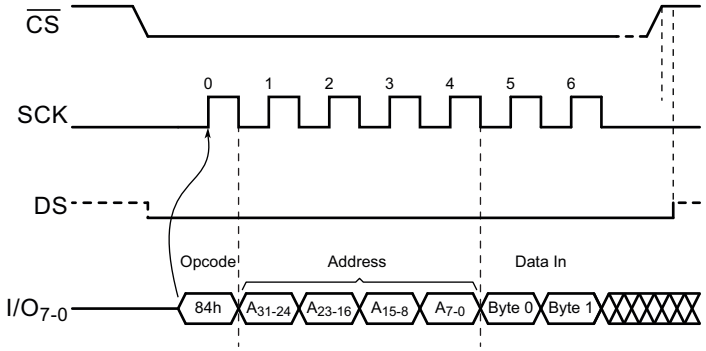
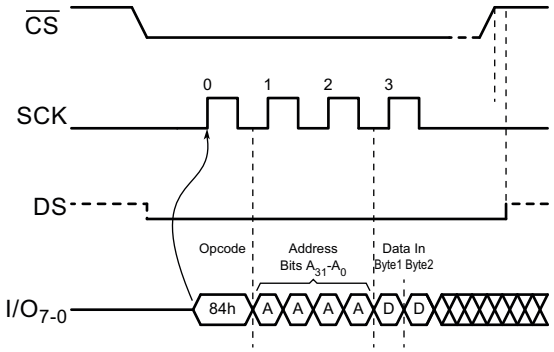


Figure 8-9. Buffer Write in Octal Mode — DDR



8.3 Buffer to Main Memory Page Program without Built-In Erase (88h)

The *Buffer to Main Memory Page Program without Built-In Erase* command allows data that is stored in the internal buffer to be written into a pre-erased page in the main memory array. The data in the buffer will be data written by the Buffer Write (84h) command or any data left in the buffer by the previous *Byte Program* or *Page Program* command. It is necessary that the page in main memory to be written has been previously erased in order to avoid programming errors. The *Buffer to Main Memory Page Program without Built-In Erase* command writes the full 256-byte page, even if less than 256 bytes were written to the buffer.

Before an 88h command can be started, the *Write Enable* command must have been previously issued to the device (see “Write Enable (06h)” on page 40) to set the Write Enable Latch (WEL) bit of the Status/Control register to a logic “1” state.

To perform a *Buffer to Main Memory Page Program without Built-In Erase*, the 88h command must be clocked into the device followed by four address bytes comprised of 10 dummy bits, 14 page address bits (PA_{13} - PA_0) that specify the page in the main memory to be written, and 8 dummy bits.

Or described differently: The four address bytes point to a byte within the page to be written, but the lower 8 bits A_7 - A_0 will be set to 0, so the page program starts at the beginning of the page boundary.

When a low-to-high transition occurs on the $\overline{CS}$ pin, the device programs 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. The programming of the page is internally self-timed and should take place in a maximum time of t_{PP}. During this time, the RDY/BUSY bit in the Status/Control register indicates that the device is busy. At some point before the program cycle completes, the WEL bit in the Status/Control Register is reset back to the logic “0” state.

In addition, if the memory is in the protected state, then the *Buffer to Main Memory Page Program without Built-In Erase Program* command is not executed, and the device returns to the idle state once the $\overline{CS}$ pin has been deasserted. The WEL bit in the Status/Control Register is reset back to the logic “0” state if the program cycle aborts due to an incomplete address being sent, or because the memory location to be programmed is protected.

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 is indicated by the EPE bit in the Status/Control register.

Figure 8-10. Buffer to Main Memory Page Program without Built-In Erase

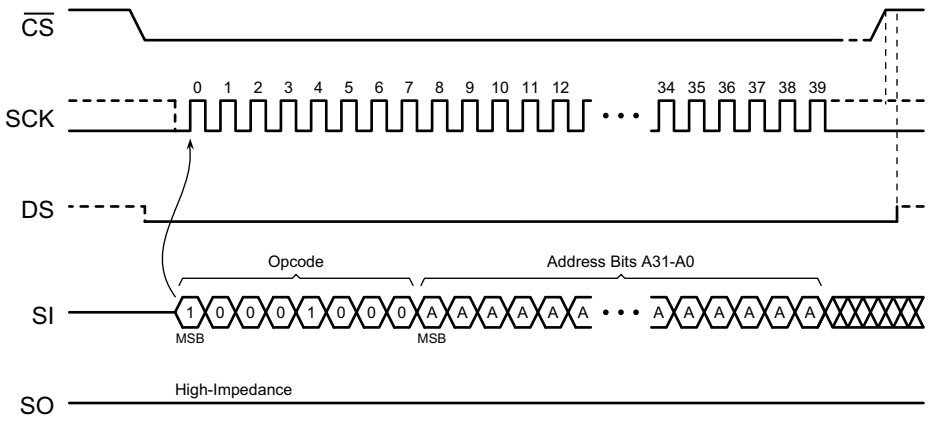


Figure 8-11. Buffer to Main Memory Page Program without Built-In Erase in Octal Mode — SDR

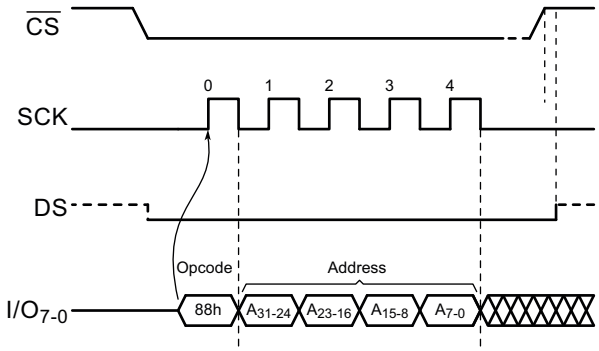
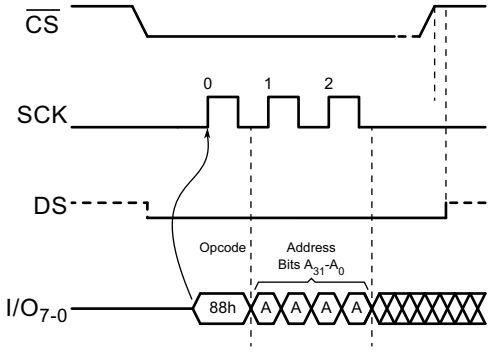


Figure 8-12. Buffer to Main Memory Page Program without Built-In Erase in Octal Mode — DDR



8.4 Block Erase

A block of 4, 32, or 64 Kbytes can be erased (all bits set to the logic “1” state) in a single operation by using one of three different commands for the *Block Erase* command. The 20h command is used for a 4-Kbyte erase, the 52h command for a 32-Kbyte erase, and the D8h command 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/Control register to a logic “1” state.

To perform a block erase operation, the $\overline{CS}$ pin must first be asserted and the appropriate command (20h, 52h, or D8h) must be clocked into the device. After the command has been clocked in, the four 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 is ignored. When the $\overline{CS}$ pin is deasserted, the device erases 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 A_{11} - A_0 are ignored by the device and their values can be either a logic “1” or “0”. For a 32-Kbyte erase, address bits A_{14} - A_0 are ignored by the device. For a 64-Kbyte erase, address bits A_{15} - A_0 are ignored by the device. Despite the lower order address bits not being decoded by the device, the complete four 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 a whole byte boundary (multiples of eight bits); otherwise, the device aborts the operation and no erase operation is performed.

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

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

While the device is executing a successful erase cycle, the Status/Control register can be read and indicates that the device is busy. For faster throughput, it is recommended that the Status/Control register be polled rather than waiting for 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/Control register is reset back to the logic “0” state.

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

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

Figure 8-13. Block Erase

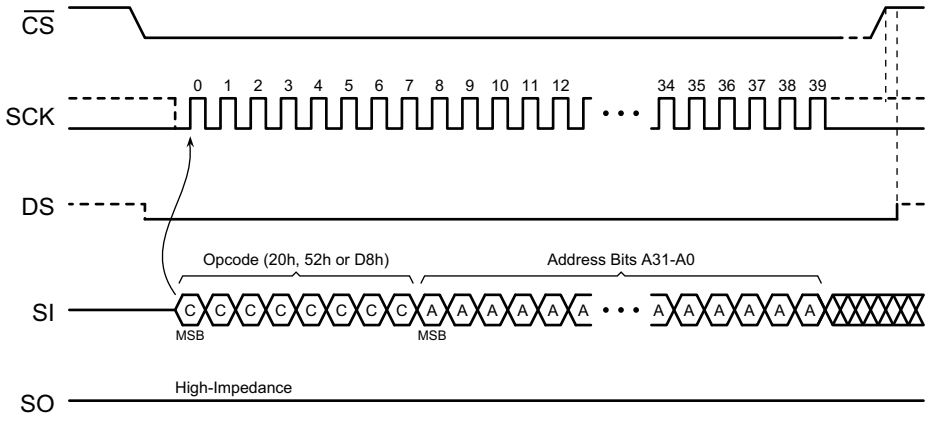


Figure 8-14. Block Erase in Octal Mode —SDR

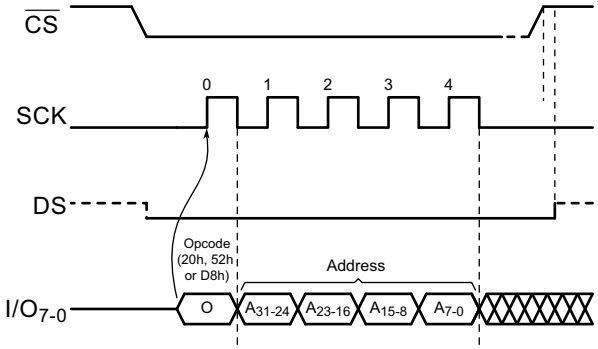
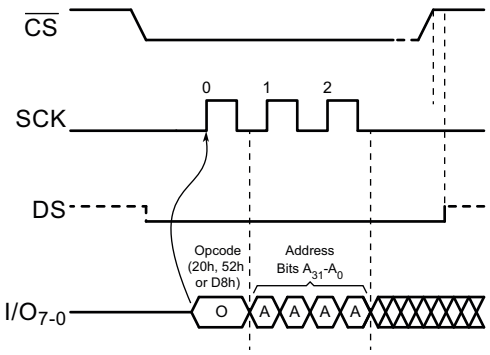


Figure 8-15. Block Erase in Octal Mode — DDR



8.5 Chip Erase (60h or C7h)

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

Two commands (60h and C7h) can be used for the *Chip Erase* command. There is no difference in device functionality when utilizing the two commands, so they can be used interchangeably. To perform a *Chip Erase*, one of the two commands 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 command is ignored. When the $\overline{CS}$ pin is deasserted, the device erases 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 command must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on a whole byte boundary (multiples of eight bits); otherwise, no erase is performed. In addition, if any sector in the memory array is in the protected state, then the *Chip Erase* command is not executed, and the device returns to the idle state once the $\overline{CS}$ pin has been deasserted. The WEL bit in the Status/Control register is reset back to the logic “0” state if the $\overline{CS}$ pin is deasserted before a whole byte is finished, or if the memory is in the protected state.

While the device is executing a successful erase cycle, the Status/Control register can be read and indicates that the device is busy. For faster throughput, it is recommended that the Status/Control 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/Control register is reset back to the logic “0” state.

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

Figure 8-16. Chip Erase

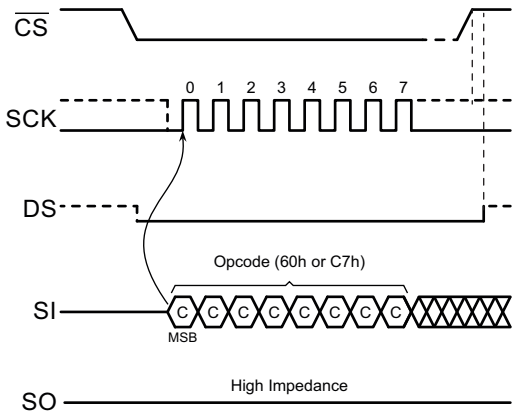


Figure 8-17. Chip Erase in Octal Mode — SDR

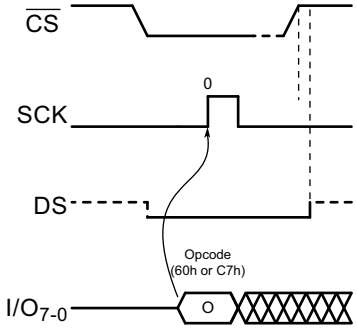
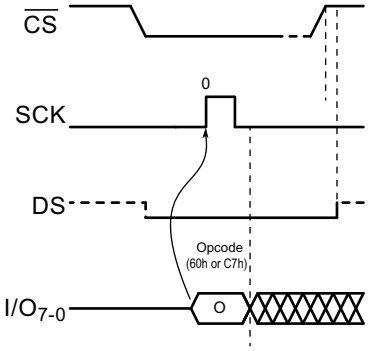


Figure 8-18. Chip Erase in Octal Mode — DDR



8.6 Program/Erase Suspend (B0h)

The Program/Erase Suspend commands can be used in addition to or instead of the Read-While-Write (RWW) option described in Section 4.1. While the RWW operation is limited to only allow read operations during program or erase, the Program/Erase Suspend command also allows a program operation to be started while a Block Erase operation is in progress.

The *Program/Erase Suspend* command allows a program operation to be started while a block erase operation is in progress. Once an on-going block erase operation has been suspended, a program operation (in a different 2-Mbit Sector than the one suspended) may be started. Once the program operation has finished, the erase operation can be resumed using the Program/Erase Resume (D0h) operation. Suspending a program operation does not allow another program operation to be started. The Program/Erase Suspend command also allows operations within the same memory bank as long as the operations are in different 2-Mbit sectors, while Read-While-Write (RWW) operations require the read and write operations to be in different memory banks.

In some code plus data storage applications, it is often necessary to process certain high-level system interrupts that require relatively immediate reading or writing of code or data from the Flash memory. In such an instance, it may not be possible for the system to wait the microseconds or milliseconds required for the Flash memory to complete a program or erase cycle. The *Program/Erase Suspend* command allows a program or erase operation in progress to a particular 2-Mbit sector of the Flash memory array to be suspended so that other device operations can be performed. For example, by suspending an erase operation to a particular block in a 2-Mbit sector, the system can perform functions such as a program or read operation within another 2-Mbit sector in the device. The two sectors *can* reside within the same memory bank. Program/Erase Suspend can therefore allow access to sectors that cannot be accessed by Read-While-Write operations.

Only *Block Erase* commands may be suspended. A *Chip Erase* command cannot be suspended. The *Program/Erase Suspend* command will be ignored if it is issued during a chip erase.

Other device operations can also be performed while a program or erase operation is suspended. Table 8-1 outlines the operations that are allowed and not allowed during a program or erase suspend.

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

To perform a program/erase suspend operation, the $\overline{CS}$ pin must first be asserted and the B0h command must be clocked into the device. No address bytes need to be clocked into the device, and any data clocked in after the command is ignored. When the $\overline{CS}$ pin is deasserted, the program or erase operation currently in progress is suspended within a time of t_{SUSP}. The Program Suspend (PS) or Erase Suspend (ES) bit in Status/Control Register Byte 2 is set to the logic “1” state to indicate that the program or erase operation has been suspended. In addition, the $\overline{RDY}$/BSY bit in the Status/Control register indicates that the device is ready for another operation. The complete command must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on a byte boundary (multiples of eight bits). Otherwise, no suspend operation is performed.

Read operations are not allowed to a 2-Mbit sector that has had its program or erase operation suspended. If a read is attempted from a suspended block, the device outputs undefined data. Therefore, when performing a read array operation to an unsuspended block and the device's internal address counter increments and crosses the block boundary to a suspended block, the device starts outputting undefined data continuously until the address counter increments and crosses a block boundary to an unsuspended block.

A program operation is not allowed to a 2-Mbit sector that has been erase suspended. If a program operation is attempted to an erase suspended block, then the program operation aborts and the WEL bit in the Status/Control register is reset back to the logic “0” state. During an erase/suspend operation, a program operation to a different 2-Mbit sector can be started and subsequently suspended. This results in a simultaneous erase suspend/program suspend condition and is indicated by the states of both the ES and PS bits in the Status/Control register being set to the logic “1” state. Another erase operation cannot be started while in the erase suspend state.

Neither program nor erase commands are allowed during a program suspend operation. If attempted, the operation is aborted.

If an attempt is made to perform an operation that is not allowed during a program or erase suspend operation, such as executing a *Write Status Register* command, then the device simply ignores the command and no operation is performed. The state of the WEL bit in the Status/Control register is not affected.

Note: *Program Suspend* and *Erase Suspend* commands cannot be used if the ADPD or AUDPD bits in Status/Control register byte 2 are set.

Table 8-1 shows which operations are allowed and not allowed during a program or erase suspend operation.

Table 8-1. Commands Allowed/Not Allowed During a Program/Erase Operation or During a Program/Erase Suspend Operation.

Command	During Program or Erase	During Program Suspend	During Erase Suspend
Read Commands			
Read Array, 4 byte address (0Bh, 0Ch, 13h)	Not Allowed	Allowed (1)	Allowed(1)
Read Array, 3 byte address (03h)	Not Allowed	Allowed(1)	Allowed(1)
Buffer Read (D4h)	Not Allowed	Allowed	Allowed
Program and Erase Commands			
Block Erase (20h, 52h, D8h)	Not Allowed	Not Allowed	Not Allowed
Chip Erase (C7h, 60h)	Not Allowed	Not Allowed	Not Allowed
Byte/Page Program (02h)	Not Allowed	Not Allowed	Allowed(1)
Buffer Write (84h)	Not Allowed	Not Allowed	Allowed
Buffer to Main Memory Page Program without Built-In Erase	Not Allowed	Not Allowed	Allowed
Program/Erase Suspend (B0h)	Allowed	N/A (2)	Program Suspend Allowed
Program/Erase Resume (D0h)	Allowed	Allowed	Allowed
Protection Commands			
Write Enable (06h)	Not Allowed	Allowed	Allowed
Write Disable (04h)	Not Allowed	Allowed (3)	Allowed(3)
Protect Sector (36h)	Not Allowed	Not Allowed	Not Allowed
Unprotect Sector (39h)	Not Allowed	Not Allowed	Not Allowed
Read Sector Protection Registers (3Ch)	Not Allowed	Allowed	Allowed
Security Commands			
Program OTP Security Register (9Bh)	Not Allowed	Not Allowed	Not Allowed
Read OTP Security Register (77h)	Not Allowed	Allowed	Allowed
Status Register Commands			
Read Status Registers (65h)	Allowed	Allowed	Allowed
Read Status Register Byte 1 (05h)	Allowed	Allowed	Allowed
Active Status Interrupt (25h)	Allowed	Allowed	Allowed

Table 8-1. Commands Allowed/Not Allowed During a Program/Erase Operation or During a Program/Erase Suspend Operation.

Command	During Program or Erase	During Program Suspend	During Erase Suspend
Write Status Register (01h, 31h)	Not Allowed	Not Allowed	Not Allowed
Write Status Registers (71h)	Not Allowed	Not Allowed	Not Allowed
Miscellaneous Commands			
Terminate Operation (F0h)	Allowed [4]	Allowed[4]	Allowed[4]
Reset Enable (66h)	Allowed	Allowed	Allowed
Reset (99h)	Allowed	Allowed	Allowed
Read Manufacturer and Device ID (9Fh)	Allowed	Allowed	Allowed
Deep Power-Down (B9h)	Not Allowed	Not Allowed	Not Allowed
Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh)	N/A	N/A	N/A
Ultra-Deep Power-Down (79h)	Not Allowed	Not Allowed	Not Allowed
Read SFDP (5Ah)	Not Allowed	Allowed	Allowed
Enter Octal Mode (E8h)	Not Allowed	Not Allowed	Not Allowed
Return to Standard SPI Mode (FFh)	Not Allowed	Not Allowed	Not Allowed
Echo (AAh)	Allowed	Allowed	Allowed
Echo with inversion (A5h)	Allowed	Allowed	Allowed

1. Allowed for all 2 Mbit sectors other than the one currently suspended.
2. Another program or erase operation cannot be started during a program suspend, so an additional program or erase suspend is not applicable.
3. Current write in progress will finish. To stop a write in progress, use the Terminate command.
4. This operation will terminate any Program or Erase operation in progress. Since the program or erase operation may not complete before the device is reset, the contents of the page being programmed or erased cannot be guaranteed to be valid.

Figure 8-19. Program/Erase Suspend

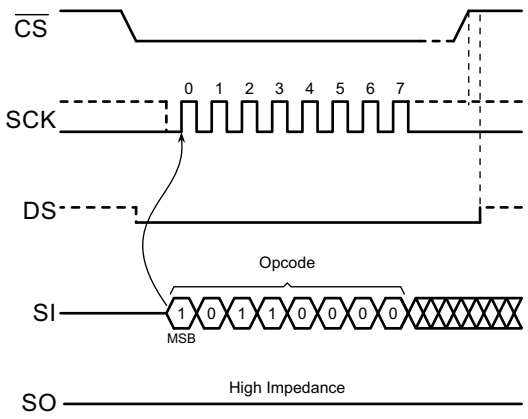


Figure 8-20. Program/Erase Suspend in Octal Mode — SDR

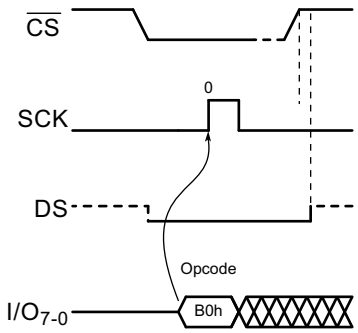
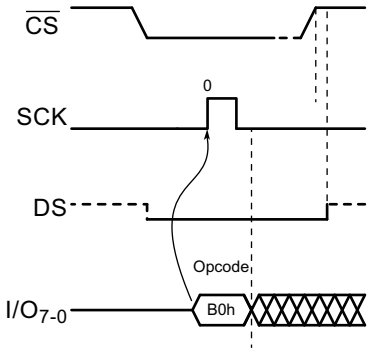


Figure 8-21. Program/Erase Suspend in Octal Mode — DDR



8.7 Program/Erase Resume (D0h)

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

To perform *Program/Erase Resume* command, the $\overline{\text{CS}}$ pin must first be asserted and the D0h command must be clocked into the device.

No address bytes need to be clocked into the device, and any data clocked in after the command is ignored. When the $\overline{\text{CS}}$ pin is deasserted, the program or erase operation currently suspended resumes within a time of t_{RES}. The SUS bit in the Status/Control register is then reset back to the logic “0” state to indicate the program or erase operation is no longer suspended. In addition, the $\overline{\text{RDY}}$/BSY bit in the Status/Control register indicates that the device is busy performing a program or erase operation. The complete command must be clocked into the device before the $\overline{\text{CS}}$ pin is deasserted, and the $\overline{\text{CS}}$ pin must be deasserted on a byte boundary (multiples of eight bits). Otherwise, no resume operation is performed.

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

While the device is busy resuming a program or erase operation, any attempts at issuing the *Program/Erase Suspend* command is ignored. Therefore, if a resumed program or erase operation needs to be subsequently suspended again, the

system must either wait the entire t_{RES} time before issuing the *Program/Erase Suspend* command, or it must check the status of the $\overline{RDY}$/BSY bit or the SUS bit in the Status/Control register to determine if the previously suspended program or erase operation has resumed.

Figure 8-22. Program/Erase Resume

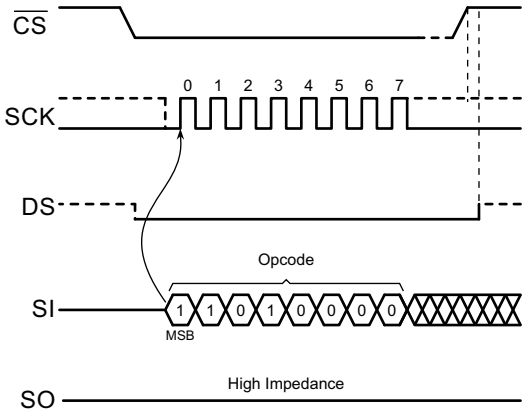


Figure 8-23. Program/Erase Resume in Octal Mode — SDR

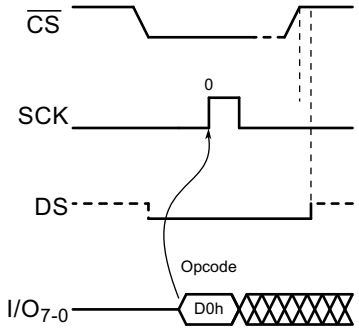
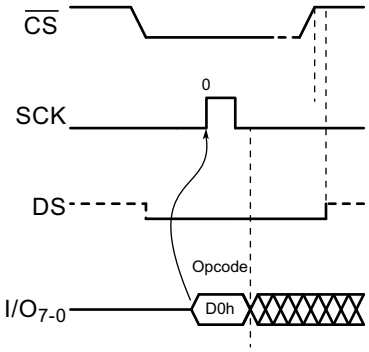


Figure 8-24. Program/Erase Resume in Octal Mode — DDR



9. Protection Commands and Features

9.1 Write Enable (06h)

The *Write Enable* command is used to set the Write Enable Latch (WEL) bit in the Status/Control register to a logic “1” state. The WEL bit must be set before a *Byte/Page Program*, *Erase*, *Program OTP Security Register*, or *Write Status Register* command can be executed. This makes the issuance of these commands a two step process, thereby reducing the chances of a command being accidentally or erroneously executed. If the WEL bit in the Status/Control register is not set prior to the issuance of one of these commands, then the command is not executed.

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

Figure 9-1. Write Enable

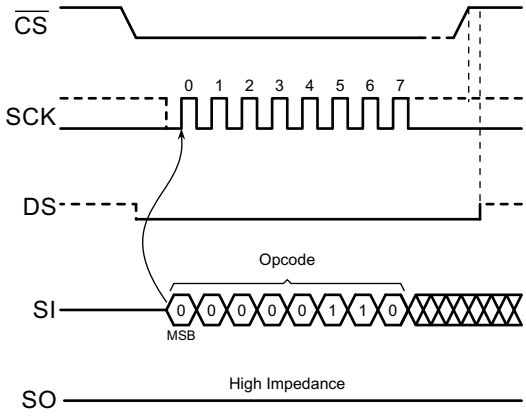


Figure 9-2. Write Enable in Octal Mode — SDR

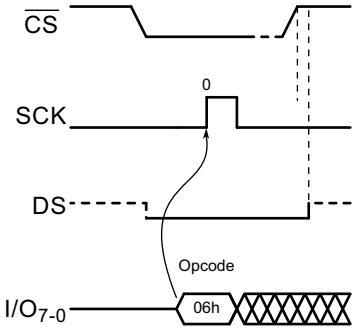
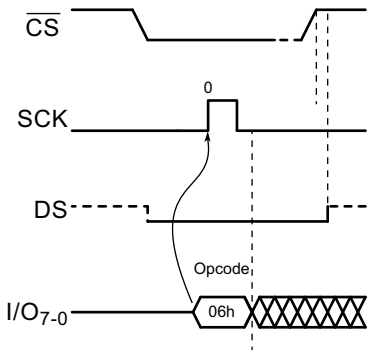


Figure 9-3. Write Enable in Octal Mode — DDR



9.2 Write Disable (04h)

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

The *Write Disable* command is ignored if a program or erase operation is already in progress. To halt an operation already in progress, use the *Reset* command.

To issue the *Write Disable* command, the $\overline{CS}$ pin must first be asserted and the 04h command must be clocked into the device. No address bytes need to be clocked into the device, and any data clocked in after the command is ignored. When the $\overline{CS}$ pin is deasserted, the WEL bit in the Status/Control register is reset to a logic "0". The complete command must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on a whole byte boundary (multiples of eight bits); otherwise, the device aborts the operation and the state of the WEL bit does not change.

Figure 9-4. Write Disable

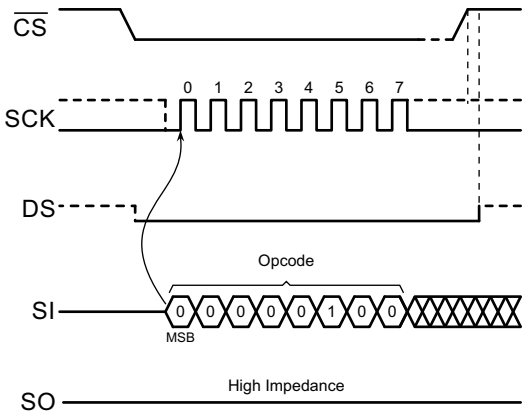


Figure 9-5. Write Disable in Octal Mode — SDR

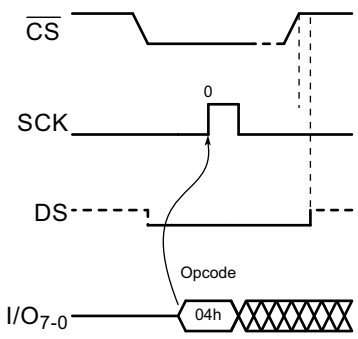
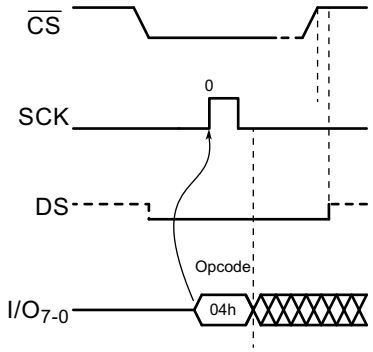


Figure 9-6. Write Disable in Octal Mode — DDR



9.3 Protect Sector (36h)

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 defaults to the logic "1" state indicating that all sectors are protected and cannot be programmed or erased.

Issuing the *Protect Sector* command to a particular sector address sets the corresponding sector protection register to the logic "1" state. The following table outlines the two states of the sector protection registers. Note that the sector protection register reverts to the default state after waking up from the ultra-deep power-down state, or after a JEDEC Standard Hardware Reset (In-Band Reset).

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/Control register to a logic "1". To issue the *Protect Sector* command, the $\overline{CS}$ pin must first be asserted and the 36h command must be clocked into the device followed by four address bytes designating any address within the sector to be protected. Any additional data clocked into the device is ignored. When the $\overline{CS}$ pin is deasserted, the Sector Protection register corresponding to the physical sector addressed by A_{31} - A_0 is set to the logic "1" state, and the sector itself

will then be protected from program and erase operations. In addition, the WEL bit in the Status/Control register is reset back to the logic “0” state.

The complete four address bytes must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on a whole byte boundary (multiples of eight bits); otherwise, the device aborts the operation, the state of the Sector Protection Register is unchanged, and the WEL bit in the Status/Control register is reset to a logic “0”.

As a safeguard against accidental or erroneous sector protection or unprotection, the Sector Protection Registers can themselves be locked from updates by using the SPRL (Sector Protection Registers Locked) bit of the Status/Control register. For more information, refer to Section 11.1, Status/Control Register Byte 1. If the Sector Protection registers are locked, then any attempts to issue the *Protect Sector* command are ignored, and the device resets the WEL bit in the Status/Control register back to a logic “0” and return to the idle state once the $\overline{CS}$ pin has been deasserted.

Figure 9-7. Protect Sector

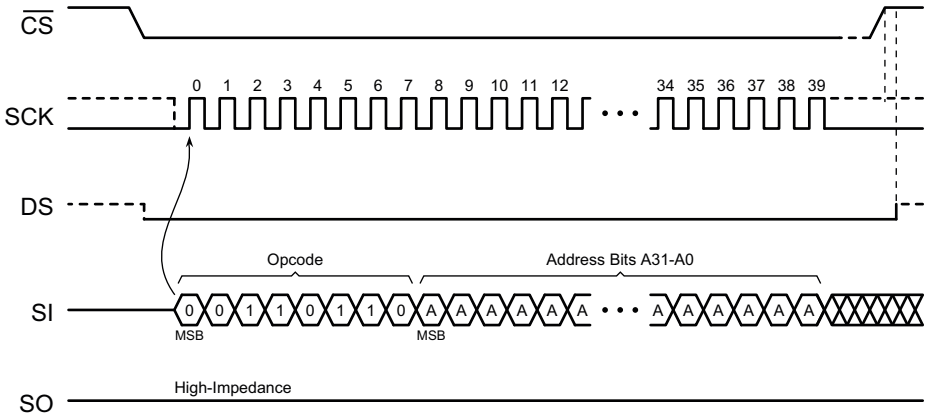


Figure 9-8. Protect Sector in Octal Mode — SDR

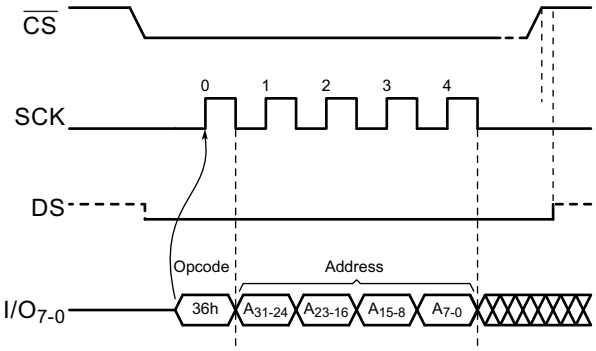
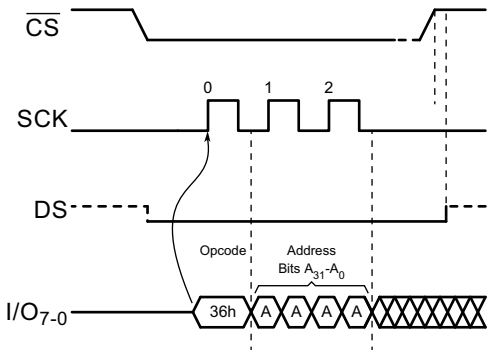


Figure 9-9. Protect Sector in Octal Mode — DDR



9.4 Unprotect Sector (39h)

Issuing the *Unprotect Sector* command to a particular sector address resets the corresponding Sector Protection register to the logic “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/Control register to a logic “1”. To issue the *Unprotect Sector* command, the $\overline{CS}$ pin must first be asserted and the 39h command must be clocked into the device. After the command has been clocked in, the four 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 is ignored. When the $\overline{CS}$ pin is deasserted, the Sector Protection register corresponding to the sector addressed by A_{31} - A_0 is reset to the logic “0” state, and the sector itself is left unprotected. In addition, the WEL bit in the Status/Control register is reset back to the logic “0” state.

The complete four address bytes must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on a whole byte boundary (multiples of eight bits); otherwise, the device aborts the operation, the state of the Sector Protection register is unchanged, and the WEL bit in the Status/Control register is reset to a logic “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 bit of the Status/Control register. For more information, refer to Section 11.1, Status/Control Register Byte 1. If the Sector Protection registers are locked, then any attempts to issue the *Unprotect Sector* command is ignored, and the device resets the WEL bit in the Status/Control register back to a logic “0” and returns to the idle state once the $\overline{CS}$ pin has been deasserted.

Figure 9-10. Unprotect Sector

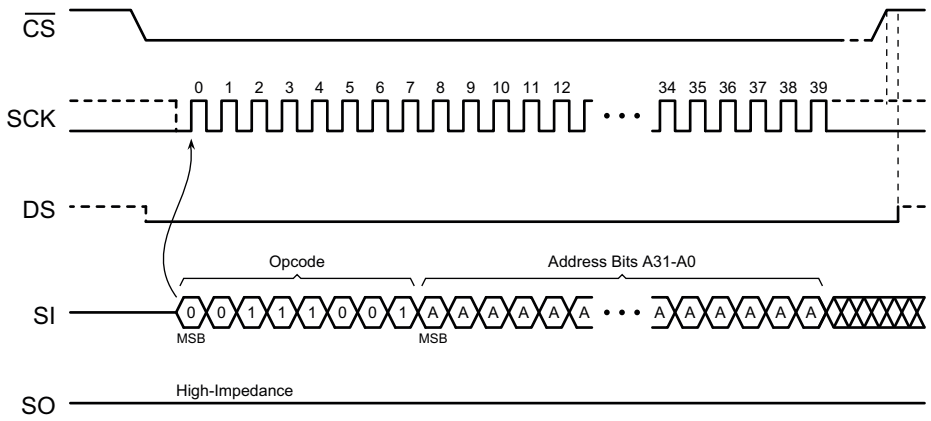


Figure 9-11. Unprotect Sector in Octal Mode — SDR

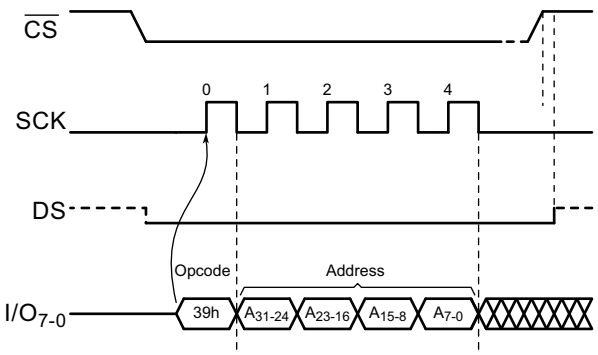
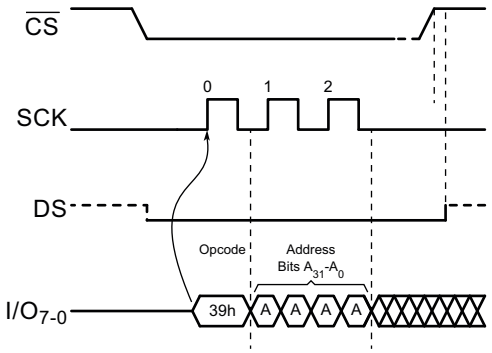


Figure 9-12. Unprotect Sector in Octal Mode — DDR



9.5 Global Protect/Unprotect

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

Performing a global protect or global unprotect operation is accomplished by writing a certain combination of data to the Status/Control register using the *Write Status Register* command. (See "Write Status Registers (71h)" on page 65 or "Write Status Register Byte 1 (01h)" on page 67 for command execution details) The *Write Status Register* command is also used to modify the SPRL bit to control hardware and software locking.

To perform a global protect operation, the appropriate $\overline{WP}$ pin and SPRL conditions must be met, and the system must write a logic "1" to bits 5, 4, 3, and 2 of the Status/Control register. Conversely, to perform a global unprotect operation, the same $\overline{WP}$ and SPRL conditions must be met but the system must write a logic "0" to bits 5, 4, 3, and 2 of the Status/Control register. Table 9-2 details the conditions necessary for a global protect or global unprotect operation to be performed.

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

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

Essentially, if the SPRL bit of the Status/Control register is in the logic “0” state (Sector Protection registers are not locked), then writing a 00h to the Status/Control register performs a global unprotect without changing the state of the SPRL bit. Similarly, writing a 7Fh to the Status/Control register performs a global protect and keeps the SPRL bit in the logic “0” state. The SPRL bit can, of course, be changed to a logic “1” by writing an FFh if software-locking or hardware-locking is desired along with the global protect.

If the desire is to only change the SPRL bit without performing a global protect or global unprotect, then the system can simply write a 0Fh to the Status/Control register to change the SPRL bit from a logic “1” to a logic “0” provided the $\overline{WP}$ pin is deasserted. Likewise, the system can write an F0h to change the SPRL bit from a logic “0” to a logic “1” without affecting the current sector protection status (no changes will be made to the Sector Protection registers).

When writing to the Status/Control register, bits 5, 4, 3, and 2 are not actually modified, but are decoded by the device for the purposes of the Global Protect and Global Unprotect functions. Only bit 7, the SPRL bit, is actually modified. Therefore, when reading the Status/Control register, bits 5, 4, 3, and 2 do not reflect the values written to them but instead indicate the status of the $\overline{WP}$ pin and the sector protection status. Please refer to the Read Status Register Byte 1 (05h) section and Table 11-3 on page 54 for details on the Status/Control register format and what values can be read for bits 5, 4, 3, and 2.

9.6 Read Sector Protection Registers (3Ch)

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

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

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

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

Deasserting the $\overline{CS}$ pin terminates 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/Control register can be read to determine if all, some, or none of the sectors are software protected. (See “Write Status Registers (71h)” on page 65 for more details.)

Figure 9-13. Read Sector Protection Register

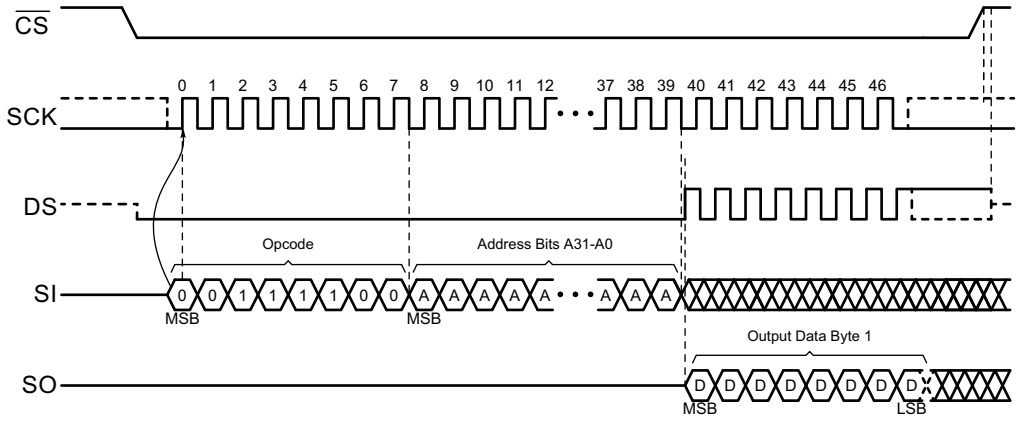


Figure 9-14. Read Sector Protection Register in Octal Mode —SDR

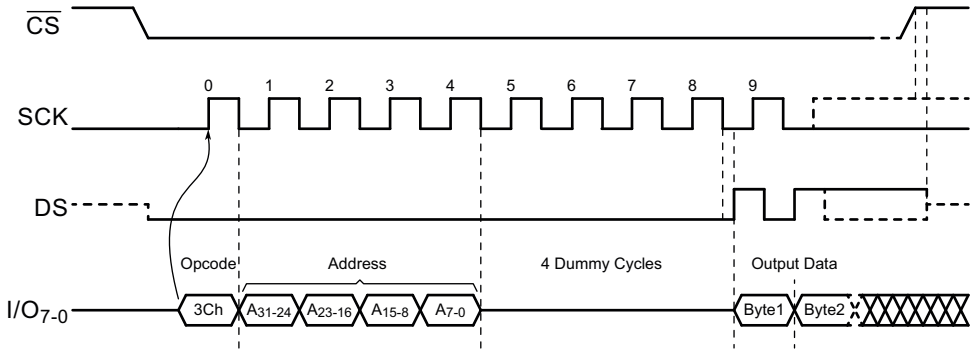
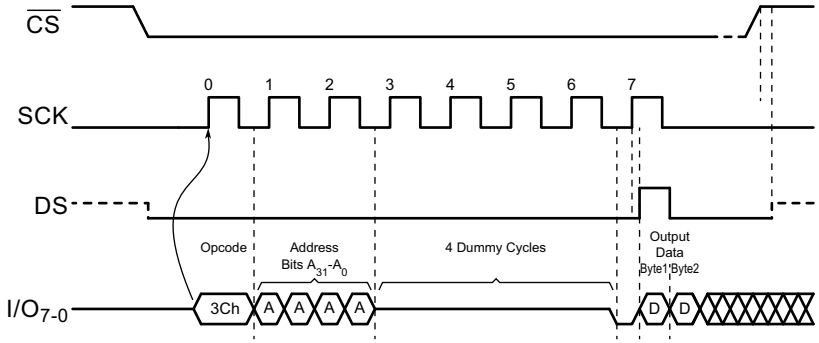


Figure 9-15. Read Sector Protection Register in Octal Mode — DDR



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

The $\overline{WP}$ pin is not linked to the memory array itself and has no direct effect on the protection status of the memory array. Instead, the $\overline{WP}$ pin, in conjunction with the SPRL (Sector Protection Registers Locked) bit in the Status/Control 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 logic "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 are locked in the protected state, and sectors that are unprotected are 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 are 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/Control register must be reset back to the logic "0" state using the *Write Status Register* command. When resetting the SPRL bit back to a logic "0", it is not possible to perform a global protect or global unprotect at the same time since the Sector Protection registers remain soft-locked until after the *Write Status Register* command has been executed.

If the $\overline{WP}$ pin is permanently connected to GND, then once the SPRL bit is set to a logic "1", the only way to reset the bit back to the logic "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/Control register.

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

In octal mode, the $\overline{WP}$ pin is used as I/O_2. The $\overline{WP}$ pin feature is disabled, and the device behaves as if the $\overline{WP}$ pin is connected to a logic '1' state.

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

Table 9-4. Sector Protection Register States

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

1. "n" represents a sector number

Table 9-5. Hardware and Software Locking

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

10. Security Commands

10.1 Program OTP Security Register (9Bh)

The device contains a specialized OTP (One-Time Programmable) security register that can be used for purposes such as unique device serialization, system-level Electronic Serial Number (ESN) storage, locked key storage, etc. The OTP security register is independent of the main Flash memory array and is comprised of a total of 256 bytes of memory divided into two portions. The first 128 bytes (byte locations 0 through 127) can be programmed (but not erased) in any order as long as byte 127 is not programmed. Once byte 127 is programmed to any value (including 0xFF), the OTP register is locked, and no further programming operations are allowed. The remaining 128 bytes of the OTP security register (byte locations 128 through 255) are factory programmed by Adesto and contains a unique value for each device. The factory programmed data is fixed and cannot be changed.

Table 10-1. OTP Security Register

Security Register Byte Number									
0	1	. . .	126	127	128	129	. . .	254	255
One-Time User Programmable					Factory Programmed by Adesto				

The user-programmable portion of the OTP security register does not need to be erased before it is programmed.

Before the *Program OTP Security Register* command can be issued, the *Write Enable* command must have been previously issued to set the WEL bit in the Status/Control register to a logic “1”. To program the OTP Security Register, the $\overline{CS}$ pin must first be asserted and the 9Bh command must be clocked into the device followed by the four address bytes denoting the first byte location of the OTP security register to begin programming at. Since the size of the user-programmable portion of the OTP security register is 128 bytes, the upper order address bits do not need to be decoded by the device. Therefore, address bits A_{31} - A_7 are ignored by the device and their values can be either a logic “1” or “0”. After the address bytes have been clocked in, data can then be clocked into the device and are stored in the internal buffer. If byte number 127 (7Fh) is among the bytes written, the OTP Security register is locked upon completion of the command execution, and no subsequent writes to the OTP Security registers are executed.

Note that any data that is sent to the device that goes beyond the end of the 128-byte user-programmable space wraps around back to the beginning of the OTP Security register. For example, if the starting address denoted by A_{31} - A_0 is 0000007Eh, and three bytes of data are sent to the device, then the first two bytes of data are programmed at OTP security register addresses 7Eh and 7Fh while the last byte of data is programmed at address 00h. As location 7Fh is written by this operation, the device is locked after these bytes have been written. 00h is still written in this case. The remaining bytes in the OTP Security register (addresses 01h through 7Dh) is not programmed by this operation, and remains in the state they previously had. If they have not been programmed previously, they remain in the erased state (FFh). In addition, if more than 128 bytes of data are sent to the device, then only the last 128 bytes sent are latched into the internal buffer.

When the $\overline{CS}$ pin is deasserted, the device takes the data stored in the internal buffer and program it into the appropriate OTP Security register locations based on the starting address specified by A_{31} - A_0 and the number of data bytes sent to the device. If less than 128 bytes of data were sent to the device, then the remaining bytes within the OTP Security register is not programmed and remains in the erased state (FFh). The programming of the data bytes is internally self-timed and should take place in a time of t_{OTPP}.

The four 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 whole byte boundaries (multiples of eight bits); otherwise, the device aborts the operation and the user-programmable portion of the OTP Security register is not programmed. The WEL bit in the Status/Control register is reset back to the logic “0” state if the OTP Security register program cycle aborts due to an

incomplete address being sent, an incomplete byte of data being sent, the $\overline{CS}$ pin being deasserted before a whole byte is finished, or because the user attempts to write a portion of the OTP Security register that was previously programmed.

Warning: Each byte of the user programmable portion of the security register can only be programmed one time.

The behavior of the device if a user tries to overwrite an already written byte cannot be guaranteed.

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

If the device is powered-down during the OTP Security register program cycle, then the contents of the 128-byte user programmable portion of the OTP Security register cannot be guaranteed and cannot be programmed again.

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

Figure 10-1. Program OTP Security Register

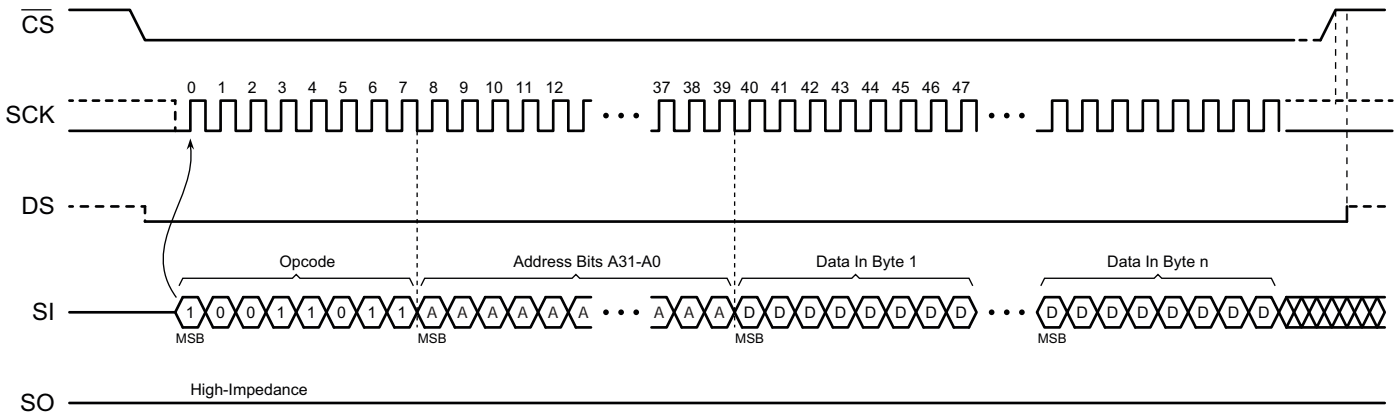


Figure 10-2. Program OTP Security Register in Octal Mode — SDR

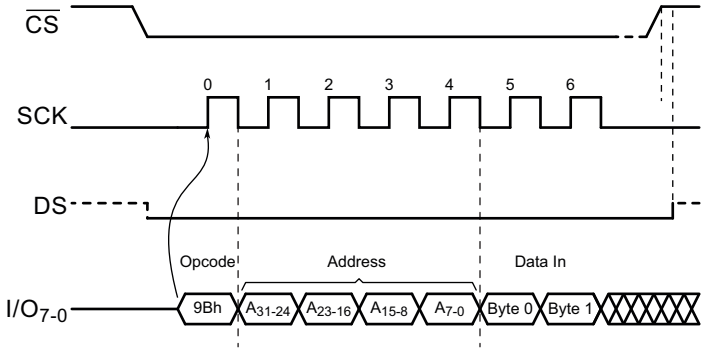
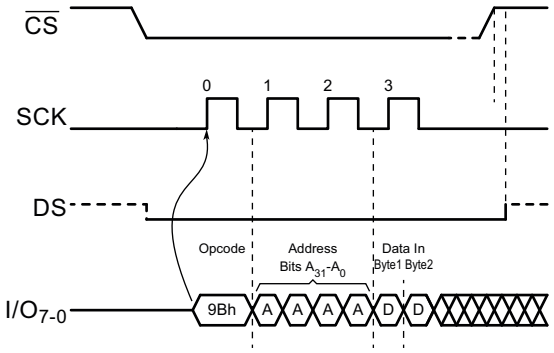


Figure 10-3. Program OTP Security Register in Octal Mode — DDR



10.2 Read OTP Security Register (77h)

The OTP Security register can be sequentially read in a similar fashion to the read array operation up to the maximum clock frequency specified by f_{CLK}. To read the OTP Security register, the $\overline{CS}$ pin must first be asserted and the 77h command must be clocked into the device. After the command has been clocked in, the four address bytes must be clocked in to specify the starting address location of the first byte to read within the OTP Security register.

After the four address bytes have been clocked in, one or more additional dummy bytes need to be clocked into the device. For the SPI Mode, one dummy byte is used. For octal mode, 8 or more dummy cycles are used as shown in Table 11-8, Dummy Clock Cycles and Maximum Operating Frequency. Note that the table refers to full clock cycles, also for DDR modes. Half cycles are not used for dummy cycles, the settings are a multiple of 2 full cycles.

After the four address bytes and the dummy cycles have been clocked in, additional clock cycles result in data being output on the I/O pin(s). The data is always output with the MSB of a byte first. When the last byte (0000FFh) of the OTP Security register has been read, the device continues reading back at the beginning of the register (000000h). No delays are incurred when wrapping around from the end of the register to the beginning of the register.

Deasserting the $\overline{CS}$ pin terminates the read operation and put the I/O pin(s) used into a high-impedance state. The $\overline{CS}$ pin can be deasserted at any time and does not require that a full byte of data be read.

Figure 10-4. Read OTP Security Register

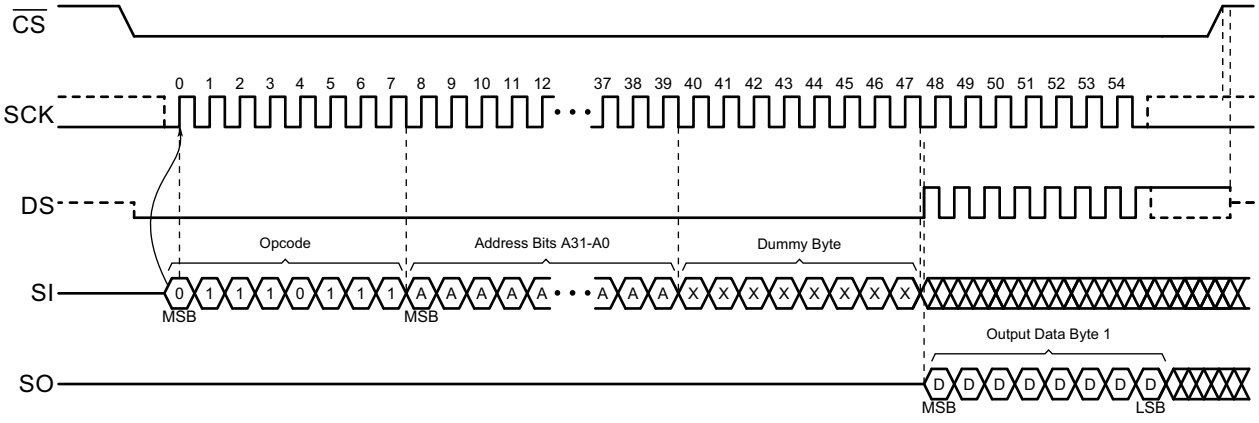


Figure 10-5. Read OTP Security Register in Octal Mode — SDR

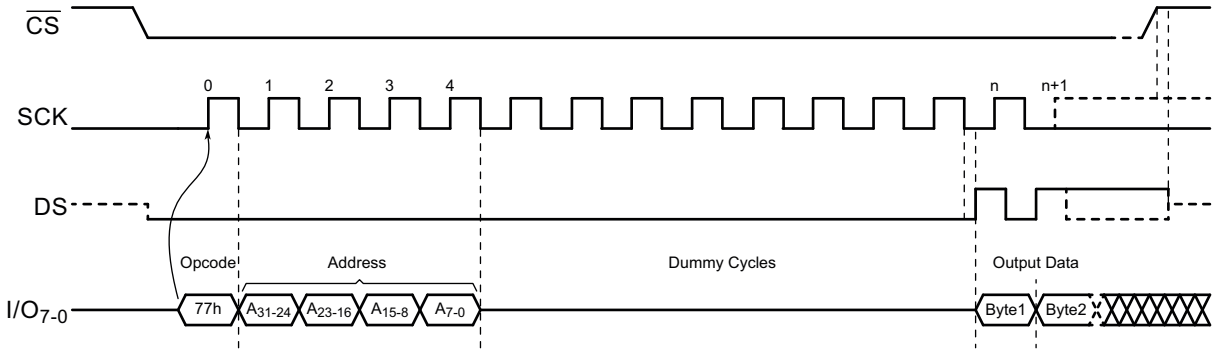
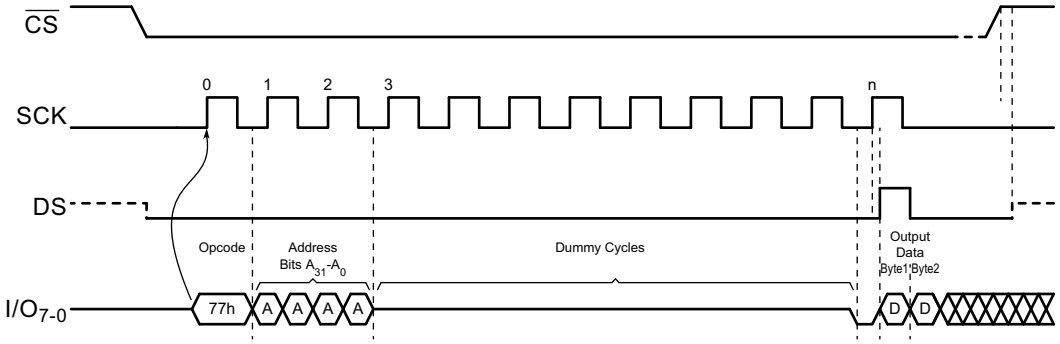


Figure 10-6. Read OTP Security Register in Octal Mode — DDR



11. Status and Control Registers - Descriptions and Commands

The ATXP064B has three Volatile Status and Control registers as listed in Table 11-1, and two Non-Volatile Status and Control registers as listed in Table 11-2. All Status and Control registers can be read and written by the generic Read Status Registers (65h) and Write Status Registers (71h).

For backward compatibility, the specific Read Status Register Byte 1 (05h) command is also supported for Status/Control Register Byte 1, and the specific Write Status Register Byte 1 (01h) and Write Status Register Byte 2 (31h) commands are supported for Status/Control register 1 and 2. Note that the behavior of the Read Status Register Byte 1 (05h) command is slightly different from many other Adesto products, as it only reads Status/Control Register Byte 1.

When writing to registers, bits that are read only or reserved for future use should be written as 0. All Volatile or all Non-Volatile registers may be written in a single operation by using the Write Status Registers (71h) command. It is not recommended to try to write both the nolatile and the non-volatile registers at the same time. Non-volatile registers are only written if the Write Status Registers (71h) command specifies the address of a non-volatile register.

Non-volatile registers are typically only written once, at the initial configuration of the device. Volatile registers have to be re-written every time the device is powered up or wakes up from ultra-deep power down.

Table 11-1. Volatile Register Address Map

Address	Register Name
0	Reserved - Reads as 0
1	Status/Control Register Byte 1
2	Status/Control Register Byte 2
3	Status/Control Register Byte 3

Table 11-2. Non-Volatile Register Address Map

Address	Register Name
128	Reserved - Reads as 0
129	I/O Pin Drive Strength Control Register

11.1 Status/Control Register Byte 1

Status/Control Register Byte 1 contains the fields listed in Table 11-3. Detailed description of each bit follows below. Note that when reading Status/Control register byte 1 while in the ultra-deep power-down state, all bits are read as 1.

Table 11-3. Status/Control Register Format - Byte 1

Bit [1]	Name		Type [2]		Description
7	SPRL	Sector Protection Registers Locked	R/W	0	Sector Protection Registers are unlocked (default)
				1	Sector Protection Registers are locked.
6	DPDS	Deep Power-Down Status	R	0	Device is in Active or Standby Mode
				1	Device is in Deep Power-Down Mode [3]
5	EPE [4]	Program/Erase Error	R	0	Erase or program operation was successful.
				1	Erase or program error detected.
4	UDPDS	Ultra-Deep Power-Down Status	R	0	Device is in Active, Standby or DPD Mode
				1	Device is in Ultra-Deep Power-Down Mode
3:2	SWP	Software Protection Status	R	00	All sectors are software unprotected (all Sector Protection Registers are 0).
				01	Some sectors are software protected. Read individual Sector Protection Registers to determine which sectors are protected.
				10	Reserved for future use.
				11	All sectors are software protected (all Sector Protection Registers are 1 – default).

Table 11-3. Status/Control Register Format - Byte 1

Bit [1]	Name		Type [2]		Description
1	WEL	Write Enable Latch Status	R	0	Device is not write enabled (default).
				1	Device is write enabled.
0	$\overline{\text{RDY}}$/BSY	Ready/Busy Status	R	0	Device is ready.
				1	Device is busy with an internal operation.

1. Only bit 7 of the Status/Control register will be modified when using the Write Status Register command.
2. R/W = Readable and writable
   R = Readable only
3. In Ultra-Deep Power-Down Mode, all bits, including the DPDS bit, reads as 1
4. Even though EPE is a volatile bit, it will not be cleared when the device enters Auto Ultra-Deep Power-Down. In this case, it is *not* be erased by a JEDEC Hardware Reset or by exercising the RESET pin. However, it is erased on a full power cycle.

11.1.1 SPRL Bit

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

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

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

11.1.2 DPDS Bit

The DPDS bit is used to detect if the device is in deep power-down mode or not. This bit reads 1 if in DPD or UDPD, 0 in any other mode. To differentiate between DPD and UDPD, check the value of the UDPDS Bit.

11.1.3 EPE Bit

The EPE bit indicates whether the last erase or program operation completed successfully or not. If at least one byte during the erase or program operation did not erase or program properly, then the EPE bit is set to the logic “1” state. The EPE bit is not 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 is updated after every erase and program operation.

The EPE bit reflects the correct state after an erase or program operation if read upon P/E completion, before executing another erase or program command or an Ultra-Deep Power-Down (79h) command.

The EPE bit reflects the correct state after erase or program with Auto Deep Power-Down Mode after a Program or Erase Operation, if exit from UDPD is done with JEDEC Standard Hardware Reset (In-Band Reset) or Hardware Reset and the EPE bit is read before executing another erase or program command or an Ultra-Deep Power-Down (79h) command.

The EPE bit state is undefined after Ultra-Deep Power-Down (79h) command, if exit from UDPD is done with JEDEC Standard Hardware Reset (In-Band Reset) or Hardware Reset.

The EPE bit is always cleared when power is cycled.

The effect on the EPE bit as a result of various operations is summarized in Table 11-4 below.

Note: Even though EPE is a volatile bit, it is *not* cleared when the device is entering Auto Ultra-Deep Power-Down. This to ensure that the error can still be detected after an Auto Ultra-Deep Power-Down Program or Erase operation.In this case, it will *not* be erased by a JEDEC Hardware Reset or by exercising the $\overline{\text{RESET}}$ pin. However, it will be erased by a full power cycle.

Note also that the EPE bit is *not* set if a hardware reset ($\overline{\text{RESET}}$ pin or JEDEC hardware reset) occurs or a Terminate (F0h) command is issued during a program or erase operation, even though the operation is terminated and it is likely that the program or erase operation will not be successful.

If a *Reset* (Terminate Operation, F0h) command is issued while a program or erase operation is actually in progress, the EPE bit is not updated.

Upon recovery from ultra-deep power-down, all internal volatile registers default to their power-on default state, except for the EPE bit in Status/Control Register Byte 1 in the following case: Even though EPE is a volatile bit, it is *not* cleared when the device is entering auto-ultra-deep power-down. This is to ensure that an error can still be detected after an auto ultra-deep power-down program or erase operation. In this case, the EPE bit is *not* erased by a JEDEC hardware reset or by exercising the $\overline{\text{RESET}}$ pin. However, it is erased by a full power cycle, or if the Ultra-Deep Power-Down (79h) command was used to enter UDPD.

Table 11-4. Effect on the EPE bit as a Result of Various Operations

Action	Effect on EPE bit
Power Cycle	Cleared
After Program/Erase	Valid value
JEDEC Standard Hardware Reset (In-Band Reset) or Hardware Reset while in Auto Deep Power-Down Mode after a Program or Erase Operation	Valid value
JEDEC Standard Hardware Reset (In-Band Reset) or Hardware Reset while in Ultra-Deep Power-Down mode (after Ultra-Deep Power-Down (79h) command)	Undefined (1)
JEDEC Standard Hardware Reset (In-Band Reset), Hardware Reset (not in AUDPD or UDPD) or Reset Enable (66h) and Reset (99h)	Undefined (2)
Terminate (F0h) while a Program or Erase operation is in progress	Undefined (3)

1. If the AUDPD feature is not used, the EPE bit will be cleared upon exit from UDPD
2. If any reset sequence or command is used to interrupt/terminate a Program or Erase operation in progress, the EPE bit will not be set
3. If a Reset (Terminate Operation, F0h) command is issued while a Program or Erase operation is actually in progress, the EPE bit will not be updated.

11.1.4 UDPDS Bit

The UDPDS bit is used to detect if the device is in ultra-deep dower-down mode or not. This bit reads 1 if in UDPD, 0 in any other mode.

11.1.5 SWP Bits

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

11.1.6 WEL Bit

The WEL bit indicates the current status of the internal write enable latch. When the WEL bit is in the logic "0" state, the device does not accept any *Program*, *Erase*, *Protect Sector*, *Unprotect Sector*, or *Write Status Register* commands. The WEL bit defaults to the logic "0" state after a device power-up or reset. In addition, the WEL bit is reset to the logic "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
- Block erase operation completes successfully or aborts
- Chip erase operation completes successfully or aborts

If the WEL bit is in the logic "1" state, it is not reset to a logic "0" if an operation aborts due to an incomplete or unrecognized command 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 command for a *Program*, *Erase*, *Protect Sector*, *Unprotect Sector*, or *Write Status Register* command must have been clocked into the device.

11.1.7 $\overline{RDY}$/BSY Bit

The $\overline{RDY}$/BSY bit is used to determine whether or not an internal operation, such as a program or erase, is in progress. To poll the $\overline{RDY}$/BSY bit to detect the completion of a program or erase cycle, new Status/Control register data must be continually clocked out of the device until the state of the $\overline{RDY}$/BSY bit changes from a logic "1" to a logic "0". See also Section 11.10, Active Status Interrupt (25h).

11.2 Status/Control Register Byte 2

Status/Control register byte 2 contains the fields listed inTable 11-5. Detailed description of each bit follows below. Note that when reading Status/Control register byte 2 while in ultra-deep power-down, all bits are read as 1.

Table 11-5. Status/Control Register – Byte 2

Bit	Acronym	Name	Type (1)	Default	Description
7	$\overline{SDR}$/DDR	Select SDR or DDR mode	R/W	0	SDR - Single data rate mode (Default)
				1	DDR - Dual data rate mode
6	AUDPD(2)	Auto Ultra-Deep Power-Down enable	R/W	11	Illegal combination - Reserved for future use
				10	AUDPD Set: Go to UDPD after Program/Erase
5	ADPD (2)	Auto Deep Power-Down enable		01	ADPD Set: Go to DPD after Program/Erase
				00	Normal mode - Go to Standby after Program/Erase

Table 11-5. Status/Control Register – Byte 2

Bit	Acronym	Name	Type [1]	Default	Description
4	TERE	Terminate enabled	R/W	0	Terminate command is disabled (default)
				1	Terminate command is enabled
3	OME	Octal mode enable	R/W	1	Octal mode enabled
				0	SPI mode enabled
2	R	Reserved for future use.	R/W	0	Reserved. Must be written as zero.
1	PS	Program Suspend status	R	0	No program operation has been suspended.
				1	A sector is program suspended.
0	ES	Erase Suspend status	R	0	No sectors are erase suspended.
				1	A sector is erase suspended.

1. R/W = Readable and writable
   R = Readable only
2. Do not use Program/Erase Suspend command if AUDPD or ADPD bits are set.

11.2.1 $\overline{\text{SDR}}$/DDR Bit

The $\overline{\text{SDR}}$/DDR bit is used to determine if the device is running in SDR or DDR mode. The default mode after power up is SDR. See Section 5.3 for details about Dual Data Rate Operation.

11.2.2 AUDPD and ADPD Bits

The AUDPD and ADPD bits are used to enable the Ultra-deep Power-down (UDPD) mode and Deep Power-down (DPD) mode respectively. The default setting after power up is always 00, the device does not use any of these modes unless they are specifically enabled. Only one of the AUDPD and ADPD bits can be active at any given time, attempting to write both to 1 at the same time is an illegal combination.

See Section 12.3, Auto Deep Power-Down Mode after a Program or Erase Operation, and Section 12.6, Auto Ultra-Deep Power-Down Mode after a Program or Erase Operation, for details about these operations.

The AUDPD bit is cleared every time the device goes into UDPD mode, so it has to be set again if another program or erase operation followed by auto ultra-deep power-down is wanted. When using the AUDPD bit, the device switches to Standard SPI Mode within t_{AUDPD} after the *Program* or *Erase* command is initiated. All Status read operations, both while the program or erase operation is still in progress and after the device has entered UDPD, can therefore be done using Standard SPI mode.

The ADPD bit is cleared every time the device goes into DPD. The device remains in the same communication mode when entering and exiting from DPD.

Note: *Program Suspend* and *Erase Suspend* commands cannot be used if the ADPD or AUDPD bits are set.

11.2.3 TERE Bit

The TERE bit is used to enable or disable the Terminate Operation command. When the TERE bit is in the logic 0 state (the default state after power-up), the Terminate Operation command is disabled and any attempts to reset the device using the *Terminate* command is ignored. When the TERE bit is in the logic 1 state, the *Terminate* command is enabled.

The TERE bit retains its state as long as power is applied to the device. Once set to the logic 1 state, the TERE bit remains in that state until it is modified using the *Write Status Register Byte 2 (31h)* command or until the device has been power cycled. The *Terminate* command itself will not change the state of the TERE bit.

11.2.4 OME Bit

The OME bit is used to enable the octal mode. The default setting after power up is always 0, so the device always wakes up in standard SPI mode. See Section 5.2 for details about Octal Mode.

11.2.5 PS (Program Suspend Status)

The PS bit indicates if a program operation has been suspended. If the PS bit is a Logic 1, then a program operation has been suspended. See Section 8.6 for details about Program/Erase Suspend (B0h) and what commands are allowed when the device is in this state.

11.2.6 ES (Erase Suspend Status)

The ES bit indicates whether or not an erase operation has been suspended. If the ES bit is a Logic 1, then an erase operation (block, sector, or chip) has been suspended. See Section 8.6 for details about Program/Erase Suspend (B0h) and what commands are allowed when the device is in this state.

11.3 Status/Control Register Byte 3

In octal mode, the "Burst Read with Wrap (0Ch)" instruction is used to perform the read operation with the "Wrap Around" feature.

11.3.1 Wrap Length Bits

The wrap length (W) bits are stored in the W6 and W5 bits of *Status/Control Register Byte 3*. The wrap-around mode is set by W7. The wrap length and wrap mode set in either SPI or octal mode is still valid in any other mode. The wrap bits do not have to be set again after changing mode.

If W7 is cleared, the "Burst Read with Wrap (0Ch)" instruction operates in continuous mode. It then reads to the end of the line, wrap around to the beginning and continue reading the same line continuously for as long as additional clock pulses are sent to SCK.

If W7 is set, the *Burst Read with Wrap* (0Ch) command reads to the end of the line, wrap around to the beginning and continue reading the same line until all bytes of the first line have been read out once. In the next clock cycle, the device starts reading at the beginning of the next line. This speeds up the loading of multiple cache lines in cached MCUs significantly, and can increase the system throughput by 40% or more without increasing system clock speeds.

See Table 11-7 below for wrap length settings.

11.3.2 WPP Bit

The WPP bit can be read to determine if the $\overline{WP}$ pin has been asserted or not. If the $\overline{WP}$ pin is floating, the WPP bit as an undefined value. This bit is only used in single SPI mode. In octal mode, this bit is undefined.

11.3.3 Dummy Clock Bits

The P3 - P0 bits are used to set the number of dummy clock cycles used for the 0Bh and 0Ch read commands.The default value after power up is 22, as this value always works independent of frequency. A lower value may be set for lower operating frequencies as described in Table 11-8, Dummy Clock Cycles and Maximum Operating Frequency.

In octal mode, to accommodate a wide range of applications with different needs for either maximum read frequency or minimum data access latency, byte 3 of the *Status/Control register* can be used to configure the number of dummy clocks for

the *Fast Read* (0Bh), *Read OTP Security Register* (77h) and *Burst Read with Wrap* (0Ch) commands, and to configure the number of bytes of 'wrap length' for the *Burst Read with Wrap* (0Ch) command.

The dummy clocks for other *Fast Read* commands in Standard SPI mode are fixed, please refer toTable 6-1 for details.

The default "wrap length" after a power up or a *Reset* command is 8 bytes, continuous mode. The default number of dummy clocks is 22. The number of dummy clocks is only programmable for *Fast Read* (0Bh), *Read OTP Security Register* (77h) and "*Burst Read with Wrap* (0Ch)" command in octal mode.

The number of dummy clocks in Table 11-8 below refers to full clock cycles on the SCK pin, both for SDR and DDR modes.

Refer to section Section 7.1, Read Array (0Bh, 13h, and 03h) and Section 7.2, Burst Read with Wrap (0Ch) for additional details.

Table 11-6. Status/Control Register Byte 3

Bit	Acronym	Name	Type	Default	Description
7	W7	Wrap Length	R/W	0	Used to set wrap length. See Table 11-7, Wrap Bit Functions — W[7:5]. Default value is 000.
6	W6		R/W	0	
5	W5		R/W	0	
4	WPP	Write Protect ($\overline{WP}$) Pin Status	R	0	$\overline{WP}$ is asserted.
				1	$\overline{WP}$ is deasserted.
3	P3	Dummy Clocks	R/W	0	Used to set number of dummy clock cycles. See Table 11-8, Dummy Clock Cycles and Maximum Operating Frequency. Default value is 0x7.
2	P2		R/W	1	
1	P1		R/W	1	
0	P0		R/W	1	

Table 11-7. Wrap Bit Functions — W[7:5]

W7	W6	W5	Wrap Length	Wrap Around
0	0	0	8-byte	Continuous on same 8-byte line (Default)
0	0	1	16-byte	Continuous on same 16-byte line
0	1	0	32-byte	Continuous on same 32-byte line
0	1	1	64-byte	Continuous on same 64-byte line
1	0	0	8-byte	Once, then continue at the beginning of the next 8-byte line
1	0	1	16-byte	Once, then continue at the beginning of the next 16-byte line
1	1	0	32-byte	Once, then continue at the beginning of the next 32-byte line
1	1	1	64-byte	Once, then continue at the beginning of the next 64-byte line

Table 11-8. Dummy Clock Cycles and Maximum Operating Frequency

					Maximum Frequency			
					0Bh Command		0Ch Command	
P3	P2	P1	P0	Dummy Clocks	Octal Mode SDR	Octal Mode DDR	Octal Mode SDR	Octal Mode DDR
0	0	0	0	8	75	50	60	45
0	0	0	1	10	95	85	80	75
0	0	1	0	12	95	85	80	75
0	0	1	1	14	95	115	80	95
0	1	0	0	16	95	115	80	95
0	1	0	1	18	95	133	80	133
0	1	1	0	20	95	133	80	133
0	1	1	1	22 (Default)	95	133	80	133
1	x	x	x	*Reserved for future use*	--	--	--	--

11.4 I/O Pin Drive Strength Control Register

The I/O Pin Strength Control register is used to control the driver strength and the nominal impedance of the I/O pins. Choice of driver strength depends on host design and on system speed and power requirements. Lower impedance typically allows higher speed operation and/or higher capacitive load, while higher impedance reduces overall power consumption and switching noise.

Table 11-9. I/O Pin Strength Control Register

Bit	Acronym	Name	Type [1]	Default	Description
7	R	*Reserved for future use*	R	0	*Reserved for future use*
6	R	*Reserved for future use*	R	0	*Reserved for future use*
5	R	*Reserved for future use*	R	0	*Reserved for future use*
4	R	*Reserved for future use*	R	0	*Reserved for future use*
3	R	*Reserved for future use*	R	0	*Reserved for future use*
2:0	IOD2	I/O Drive Select 2	R/W	See Table 11-10	
	IOD1	I/O Drive Select 1	R/W		
	IOD0	I/O Drive Select 0	R/W		

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

Table 11-10. I/O Driver Strength Types — IOD[2:0]

IOD2	IOD1	IOD0	Driver Type Value	Nominal Impedance	Approximate driving capability compared to Type-0	Remarks
0	0	0	0x00	50 Ω	x1	Default
0	0	1	0x01	33 Ω	x1.5	Supports highest load / speed
0	1	0	0x02	66 Ω	x0.75	Supports lower energy consumption
0	1	1	0x03	100 Ω	x0.5	Supports lowest energy consumption
1	0	0	0x04	40 Ω	x1.2	Supports higher load / speed
1	0	1	N/A	N/A	N/A	N/A
1	1	0	N/A	N/A	N/A	N/A
1	1	1	N/A	N/A	N/A	N/A

11.5 Read Status Registers (65h)

All Status/Control registers may be read out by using the generic *Read Status Registers* command. This command allows for a readout of all Status/Control registers in one operation.The Status/Control registers can be read at any time, including during an internally self-timed program or erase operation.

The Status/Control registers can be sequentially read in a similar fashion to the read array operation up to the maximum clock frequency specified by f_{CLK}. To read the Status/Control registers, the $\overline{CS}$ pin must first be asserted and the command of 65h must be clocked into the device. After the command has been clocked in, one address byte must be clocked in to specify which register to read. Following the address byte, one or more additional dummy bytes need to be clocked into the device. For the SPI mode, one dummy byte (8 cycles) is used. For octal SDR mode, 4 dummy cycles are used (3.5 dummy cycles for octal DDR).

After the address byte and the dummy cycles have been clocked in, additional clock cycles result in data being output on the I/O pin(s). The data is always output with the MSB of a byte first. If Status/Control Register Byte 1 is read first, it is directly followed by Status/Control register byte 2 and so on. The output values for addresses higher than 3 are not defined in this device.

Note that for octal DDR mode, Status/Control register bytes are output on both edges of the SCK pulse. When reading just a single register (on the rising SCK edge), the next register location is also read out (on the falling SCK edge). When reading all three registers, the following register location is also read out (on the falling edge of the 2nd SCK clock). If the extra register location read out is not with address 1 - 3, the value read out is undefined.

Deasserting the $\overline{CS}$ pin terminates the read operation and put the I/O pin(s) used into a high-impedance state. The $\overline{CS}$ pin can be deasserted at any time and does not require that a full byte of data be read.

Figure 11-1. Read Status/Control Registers

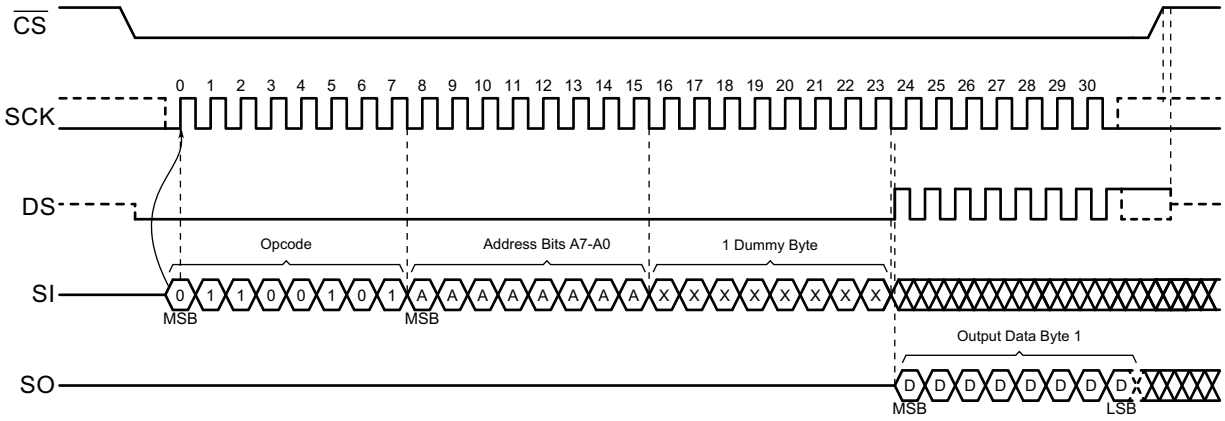


Figure 11-2. Read Status/Control Registers in Octal Mode — SDR

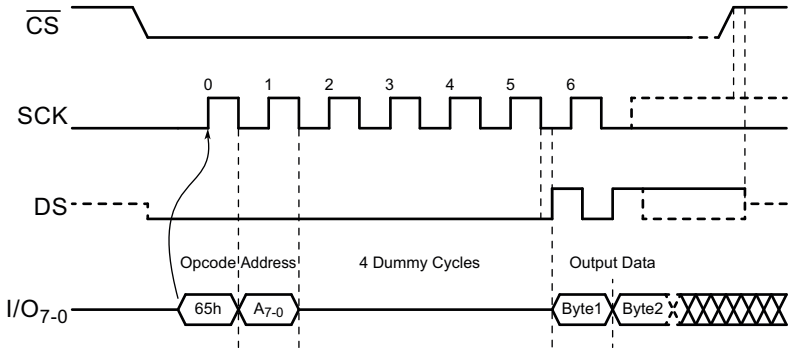
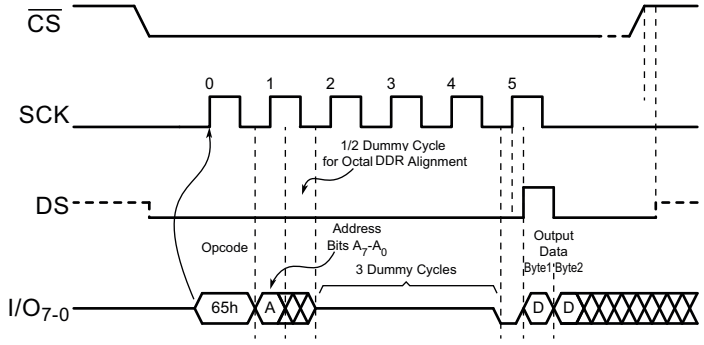


Figure 11-3. Read Status/Control Registers in Octal Mode — DDR



11.6 Read Status Register Byte 1 (05h)

Status/Control Register Byte 1 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. Status/Control Register Byte 1 can be read at any time, including during an internally self-timed program or erase operation.

To read Status/Control Register Byte 1, the $\overline{CS}$ pin must first be asserted and the 05h command must be clocked into the device. In Octal mode, 4 dummy cycles have to be clocked into the device. After the command and dummy cycles have been

clocked in, the device begins outputting Status/Control register data on the I/O pins during every subsequent clock cycle. After the last bit (0) of Status/Control Register Byte 1 has been clocked out, the sequence repeats itself, starting again with bit 7 of Status/Control Register Byte 1, as long as the $\overline{CS}$ pin remains asserted and the clock pin is being pulsed. The data in the Status/Control register is constantly being updated, so each repeating sequence may output new data.

Deasserting the $\overline{CS}$ pin terminates the Read Status Register Byte 1 (05h) operation and put the I/O pins into a high-impedance state. The $\overline{CS}$ pin can be deasserted at any time and does not require that a full byte of data be read.

11.6.1 Reading Status Register Byte 1 while in Deep Power-Down

The Read Status Register Byte 1 (05h) command also works while in Deep Power-Down. The DPDS bit in Status/Control register 1 reads as 1 in this case, while the $\overline{RDY}$/BSY and the UDPDS bits reads as 0. This indicates that the device is in deep power-down.

11.6.2 Reading Status Register Byte 1 while in Ultra-Deep Power-Down

In ultra-deep power-down mode, the Read Status Register Byte 1 (05h) command is only supported in standard SPI mode. All bits are read as 1 in this mode. Specifically, the UDPDS bit reads as 1 in this mode, and only in this mode, and can therefore be used as a clear indication that the device is in ultra-deep power-down.

Figure 11-4. Read Status Register Byte 1

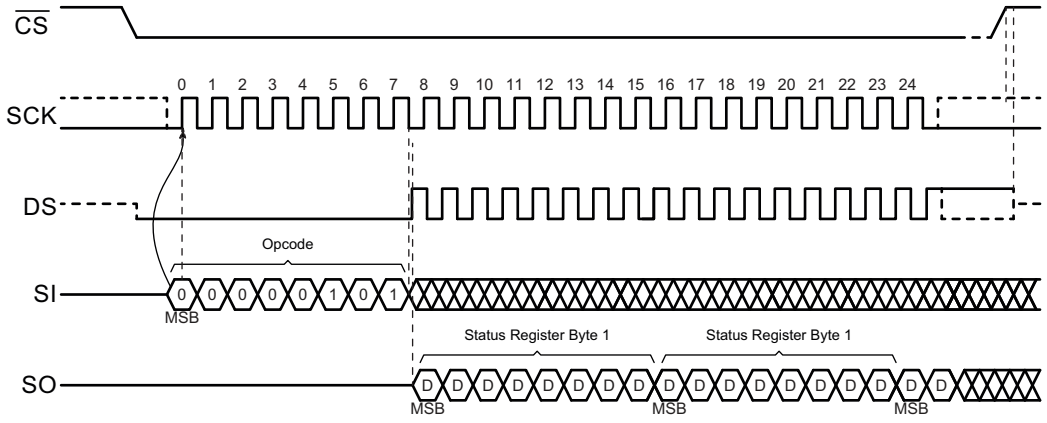


Figure 11-5. Read Status Register Byte 1 in Octal Mode — SDR

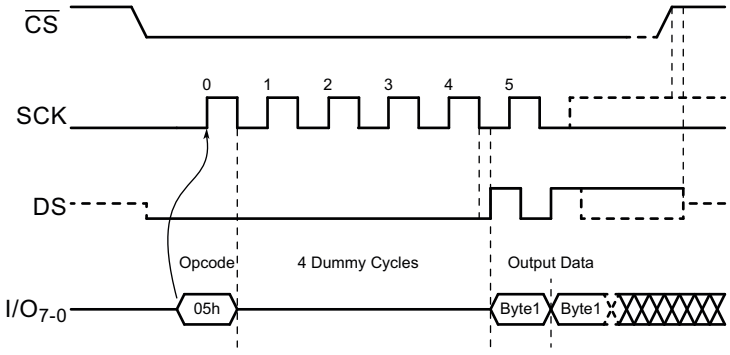
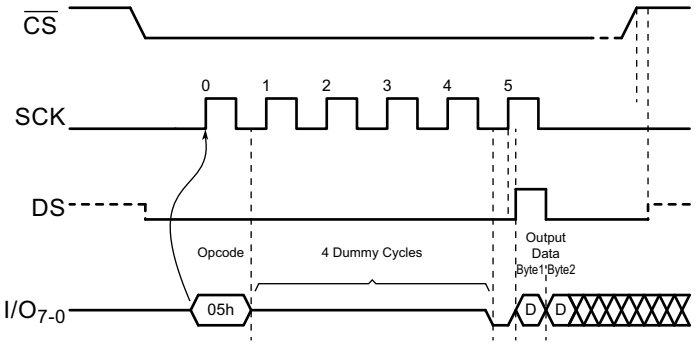


Figure 11-6. Read Status Register Byte 1 in Octal Mode — DDR



11.7 Write Status Registers (71h)

All Status/Control registers may be written by using the generic *Write Status Register* command.

This command allows writing to all volatile Status/Control registers in a single operation for standard SPI mode and octal SDR mode. However, in octal DDR mode, volatile Status/Control registers can only be written one at the time. They cannot all be written in one operation as in octal SDR mode. Similarly, non-volatile Status/Control registers must be written one at the time in any mode. They cannot all be written in one operation.

It is not possible to write both the volatile and the non-volatile registers at the same time. Non-volatile registers are only written if the command specifies the address of a non-volatile register.

Writing to other addresses than the ones listed in Table 11-1 or Table 11-2 is not supported.

There is no wrap around on the address field. When writing multiple volatile Status/Control registers in a single operation, the registers have to be clocked in ascending order.

When writing to registers, bits that are read only or reserved for future use should be written as 0.

Before the *Write Status Registers* command can be issued, the *Write Enable* command must have been previously issued to set the WEL bit in the Status/Control register to a logic “1”.

To issue the *Write Status Registers* command, the $\overline{CS}$ pin must first be asserted and the 71h command must be clocked into the device. After the command has been clocked in, one address byte must be clocked in to specify at which register to start the write operation, followed by one or more bytes of data. See Table 11-1 to Table 11-6 for details about register values.

When the $\overline{CS}$ pin is deasserted, the rewritable bits in the registers are modified, and the WEL bit in the Status/Control register is reset back to a logic “0”.

The address byte 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 whole byte boundaries (multiples of eight bits); otherwise, the device aborts the operation and no data is programmed into the registers, and the WEL bit in the Status/Control register is reset back to the logic “0” state.

When writing volatile Status/Control registers in octal DDR mode, the address given has to be for the single register to be written. Only that register is written. To write additional registers in this mode, additional commands have to be issued.

When writing non-volatile Status/Control registers in Octal DDR mode, the address given can be odd or even, and a single register is written.

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

Status/Control register is reset back to the logic “0” state. In order to reset the SPRL bit to a logic “0”, the $\overline{WP}$ pin must be deasserted.

Note: No other commands should be issued while this operation is in progress. As changes to the Status/Control registers will change the configuration of I/O pins, users should wait t_{WRSR} (for volatile registers) or t_{WRSRNV} (for non-volatile registers) before issuing further commands.

Figure 11-7. Write Status Registers

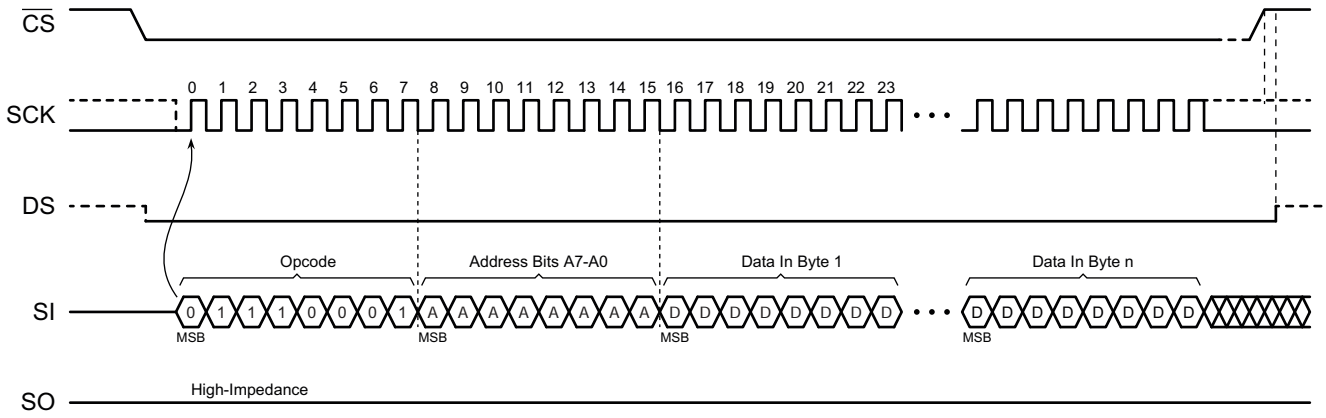


Figure 11-8. Write Status Registers in Octal Mode — SDR

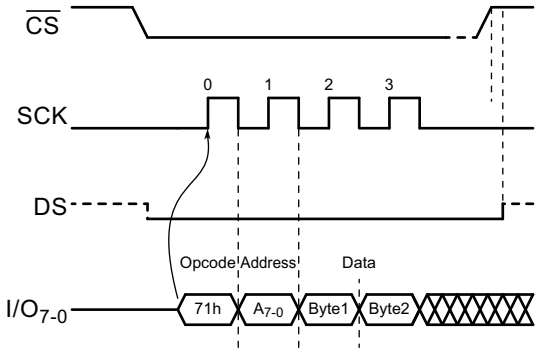
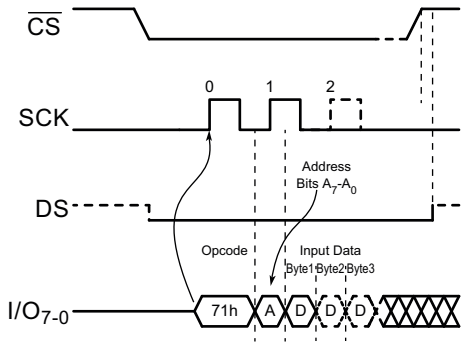


Figure 11-9. Write Status Registers in Octal Mode — DDR



11.8 Write Status Register Byte 1 (01h)

The Write Status Register Byte 1 (01h) command is used to modify the SPRL bit of the Status/Control register and/or to perform a global protect or global unprotect operation. Before the *Write Status Register* command can be issued, the *Write Enable* command must have been previously issued to set the WEL bit in the Status/Control register to a logic “1”.

To issue the *Write Status Register* command, the $\overline{CS}$ pin must first be asserted and the command of 01h must be clocked into the device followed by one byte of data. The one byte of data consists of the SPRL bit value, a don’t care bit, four data bits to denote whether a global protect or unprotect operation should be performed, and two additional don’t care bits (see Table 11-11). Any additional data bytes that are sent to the device are ignored. When the $\overline{CS}$ pin is deasserted, the SPRL bit in the Status/Control register is modified, and the WEL bit in the Status/Control register is reset back to a logic “0”. The values of bits 5, 4, 3, and 2 and the state of the SPRL bit before the *Write Status Register* command was executed (the prior state of the SPRL bit) determines whether or not a global protect or global unprotect operation is performed. See Section 9.5, Global Protect/Unprotect on page 45 for more details.

The complete one byte of data must be clocked into the device before the $\overline{CS}$ pin is deasserted; otherwise, the device aborts the operation, the state of the SPRL bit does not change, no potential global protect or unprotect is performed, and the WEL bit in the Status/Control register is reset back to the logic “0” state.

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

Table 11-11. Write Status Register Format

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

Figure 11-10. Write Status Register

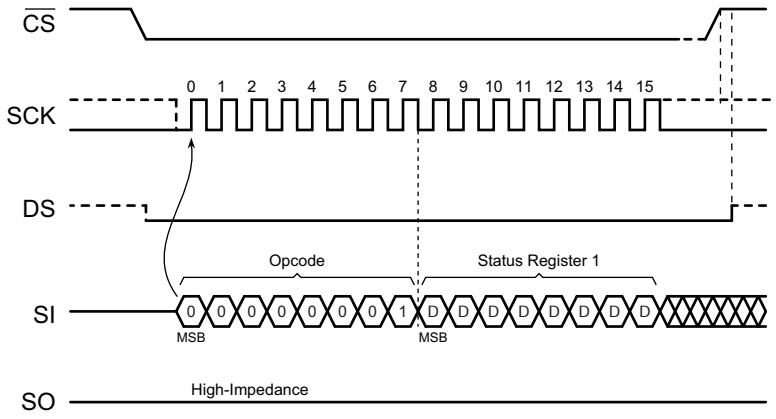


Figure 11-11. Write Status Register in Octal Mode — SDR

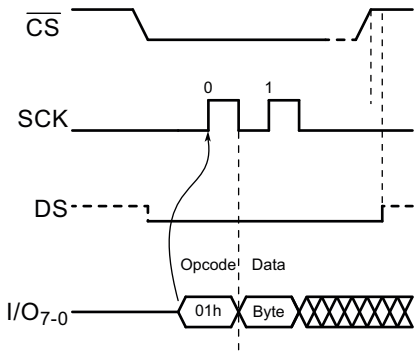
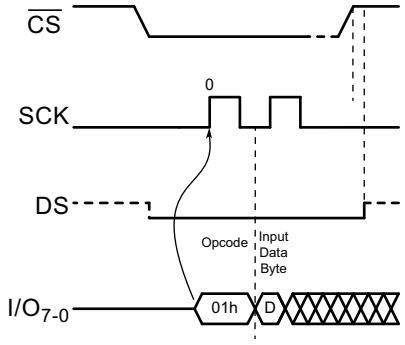


Figure 11-12. Write Status Register in Octal Mode — DDR



11.9 Write Status Register Byte 2 (31h)

The *Write Status Register Byte 2* command is used to modify the Status/Control register byte 2 bits. Using the *Write Status Register Byte 2* command is the only way to modify the TERE and $\overline{\text{SDR}}$/DDR bits in Status/Control register byte 2 during normal device operation. The standard and octal select bits may also be changed by using the Enter Octal Mode (E8h) and Return to Standard SPI Mode (FFh) commands.

Before the *Write Status Register Byte 2* command can be issued, the *Write Enable* command must have been previously issued to set the WEL bit in the Status/Control register to a logic 1.

To issue the *Write Status Register Byte 2* command, the $\overline{\text{CS}}$ pin must first be asserted and then the command 31h must be clocked into the device followed by one byte of data. The one byte of data consists of the bits described in Table 11-5. Any additional data bytes sent to the device are ignored. When the $\overline{\text{CS}}$ pin is deasserted, the TERE bit in the Status/Control register is modified, and the WEL bit in the Status/Control register is reset back to a logic 0.

The complete one byte of data must be clocked into the device before the $\overline{\text{CS}}$ pin is deasserted, and the $\overline{\text{CS}}$ pin must be deasserted on whole byte boundaries (multiples of eight bits); otherwise, the device aborts the operation, the state of the TERE bit does not change, and the WEL bit in the Status/Control register is reset back to the logic 0 state.

Figure 11-13. Write Status Register Byte 2

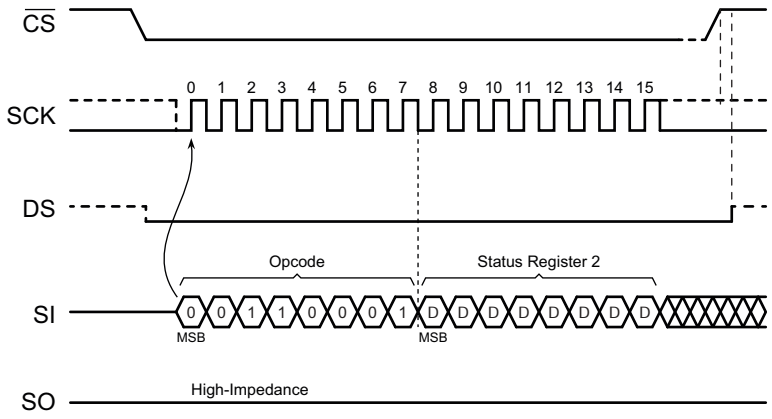


Figure 11-14. Write Status Register Byte 2 in Octal Mode — SDR

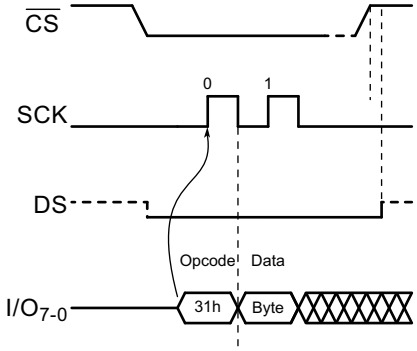
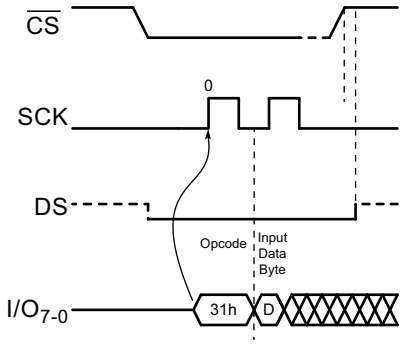


Figure 11-15. Write Status Register Byte 2 in Octal Mode — DDR



11.10 Active Status Interrupt (25h)

To simplify the readout of the $\overline{\text{RDY}}$/BSY bit, the *Active Status Interrupt* command (25h) may be used. It is then not necessary to continuously read the Status/Control register, it is sufficient to monitor the value of the SO line. If the SO line is connected to an interrupt line on the host controller, the host controller may be in sleep mode until the SO line indicates that the ATXP064B is ready for the next command.

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

To enable the *Active Status Interrupt* command, the $\overline{\text{CS}}$ pin must first be asserted and the command of 25h must be clocked into the device. For SPI Mode 3, at least one dummy bit has to be clocked into the device after the last bit of the command has been clocked in. (In most cases, this is most easily done by sending a dummy byte to the device.) The value of the SI line after the command is clocked in is of no significance to the operation. For SPI Mode 0, this dummy bit (dummy byte) is not required. For octal mode, 4 dummy cycles are required.

The value of $\overline{\text{RDY}}$/BSY is then output on the SO line, and is continuously updated by the device for as long as the $\overline{\text{CS}}$ pin remains asserted. All other output pins remain in a high-impedance state.

Additional clocks on the SCK pin are not required. If the $\overline{\text{RDY}}$/BSY bit changes from 1 to 0 while the $\overline{\text{CS}}$ pin is asserted, the SO line changes from 1 to 0. The $\overline{\text{RDY}}$/BSY bit cannot change from 0 to 1 during an operation, so if the SO line already is 0, it will not change.

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

Note: When operating in octal DDR mode at high speed, it is recommended to have the interrupt signal connected to the SO line internally in the host controller. Connecting an external interrupt pin to the SO line adds additional load on that one line, adding a delay on this line compared to the others.

Note: The *Active Status Interrupt* command is intended to be used when the host controller is not actively reading from the device. When the host controller is actively reading from the device, it can read the $\overline{\text{RDY}}$/BSY bit by polling Status/Control Register Byte 1 from time to time.

Figure 11-16. Active Status Interrupt

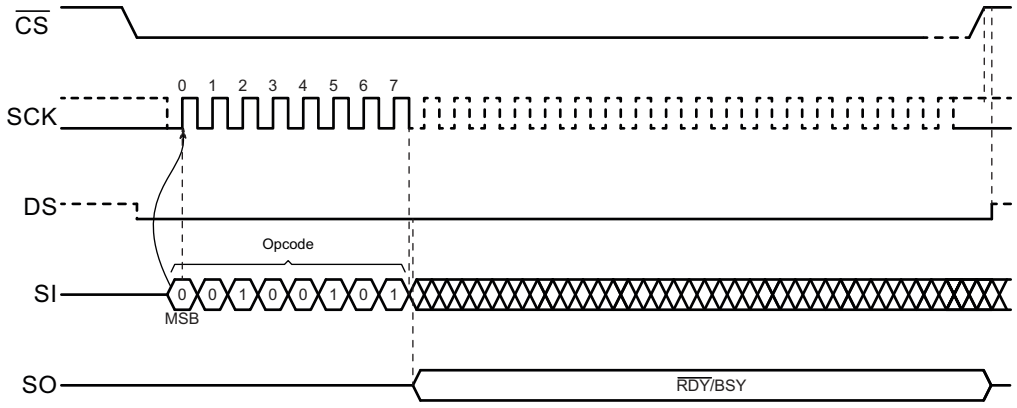


Figure 11-17. Active Status Interrupt in Octal Mode — SDR

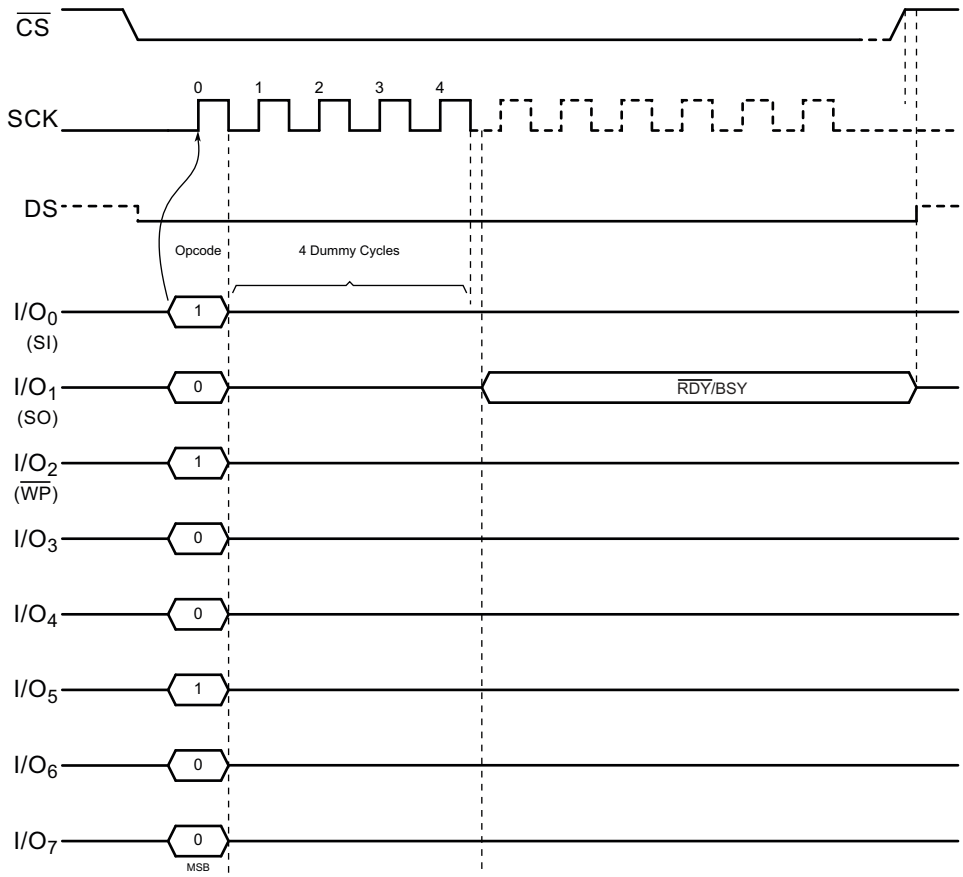
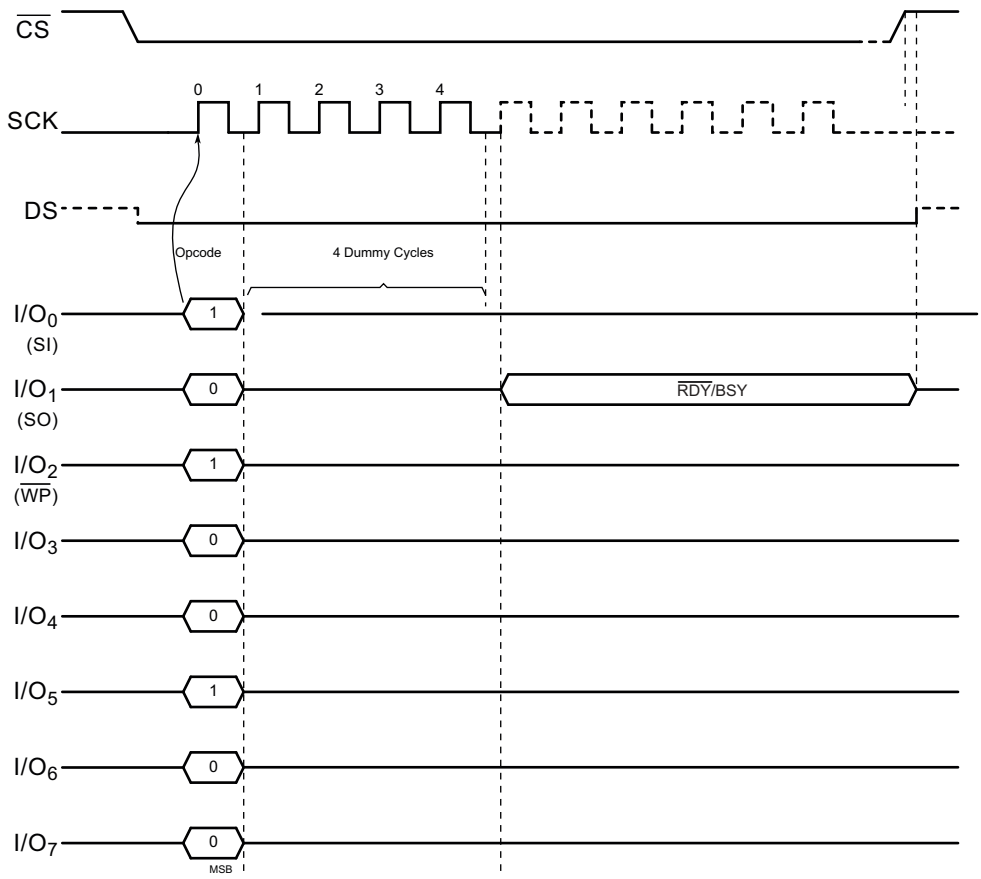


Figure 11-18. Active Status Interrupt in Octal Mode — DDR



12. Other Commands and Functions

12.1 Read Manufacturer and Device ID (9Fh)

Identification information can be read from the device to enable systems to electronically query and identify the device while it is in the system. The identification method and the command comply with the JEDEC standard for “manufacturer and device ID read methodology for SPI-compatible serial interface memory devices”. The type of information that can be read from the device includes the JEDEC-defined manufacturer ID, the vendor-specific device ID, and the vendor-specific extended device information.

The *Read Manufacturer and Device ID* command is limited to a maximum clock frequency of f_{CLK}. Since not all Flash devices are capable of operating at very high clock frequencies, applications should be designed to read the identification information from the devices at a reasonably low clock frequency to ensure all devices used in the application can be identified properly. Once the identification process is complete, the application can increase the clock frequency to accommodate specific Flash devices that are capable of operating at the higher clock frequencies.

To read the identification information, the $\overline{CS}$ pin must first be asserted and then the command of 9Fh must be clocked into the device. After the command has been clocked in, the device begins outputting the identification data on the SO pin during the subsequent clock cycles. The first seven bytes output are continuation codes (7Fh) followed by the manufacturer ID and two bytes of device ID information. The eleventh byte output is the extended device information string length, which is 01h, indicating that one byte of extended device information follows. After the extended device information byte 1 is output, the SO pin enters the high-impedance state; therefore, additional clock cycles have no affect on the SO pin and no

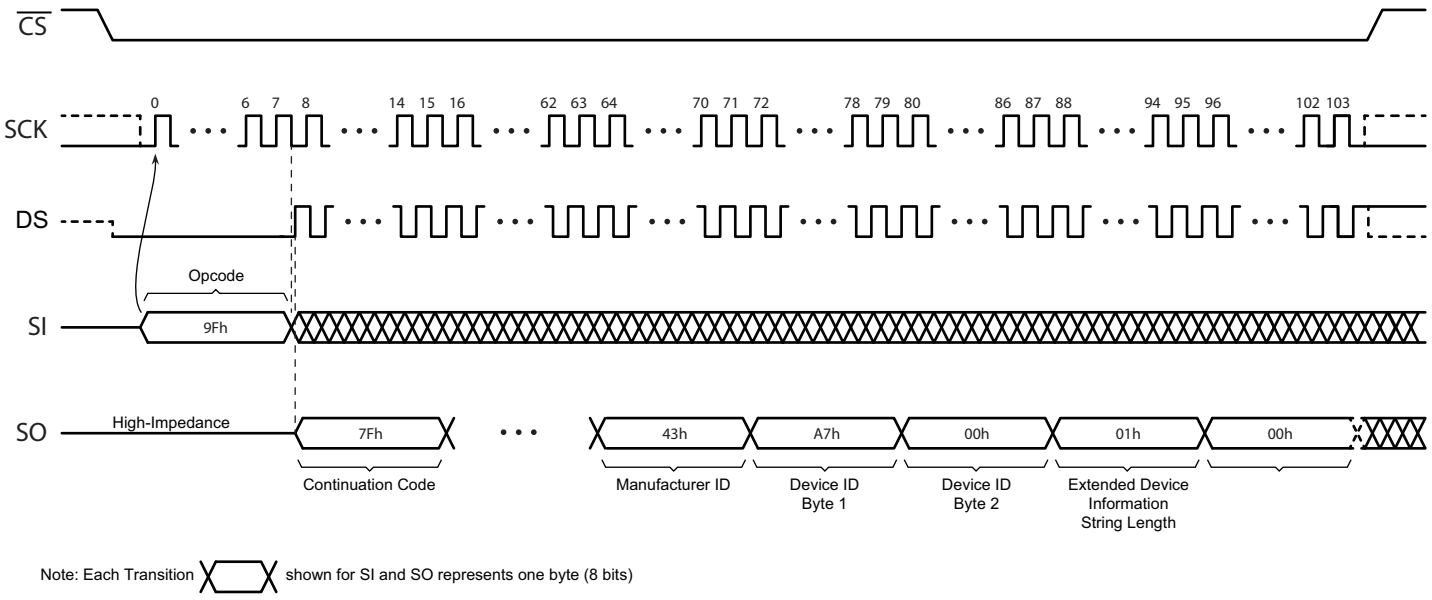
data is output. As indicated in the JEDEC standard, reading the extended device information string length and any subsequent data is optional.

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

Table 12-1. Manufacturer and Device ID Details

Data Type	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Hex Value	Details
Manufacturer ID	JEDEC Assigned Code								1Fh	JEDEC code: 0001 1111 (1Fh for Adesto)
	0	0	0	1	1	1	1	1		
Device ID (Byte 1)	Family Code			Density Code					A8h	Family code: 101 (ATXPxxxB series) Density code: 01000 (64-Mbit)
	1	0	1	0	1	0	0	0		
Device ID (Byte 2)	Sub Code			Product Version Code					01h	Sub code: 000 (Standard Series) Product variant: 00001
	0	0	0	0	0	0	0	1		

Figure 12-1. Read Manufacturer and Device ID



12.2 Deep Power-Down (B9h)

During normal operation, the device is 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 are ignored with the exception of the Read Status Register Byte 1 (05h) command and the Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh) command. Since all write commands are 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 command of B9h, and then deasserting the $\overline{CS}$ pin. Any additional data clocked into the device after the command is ignored. When the $\overline{CS}$ pin is deasserted, the device enters the deep power-down mode within the maximum time of t_{EDPD}.

The complete command must be clocked in before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on a whole byte boundary (multiples of eight bits); otherwise, the device aborts the operation and returns to the standby mode once the $\overline{CS}$ pin is deasserted. In addition, the device defaults to the standby mode after a power-cycle.

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

Figure 12-2. Deep Power-Down

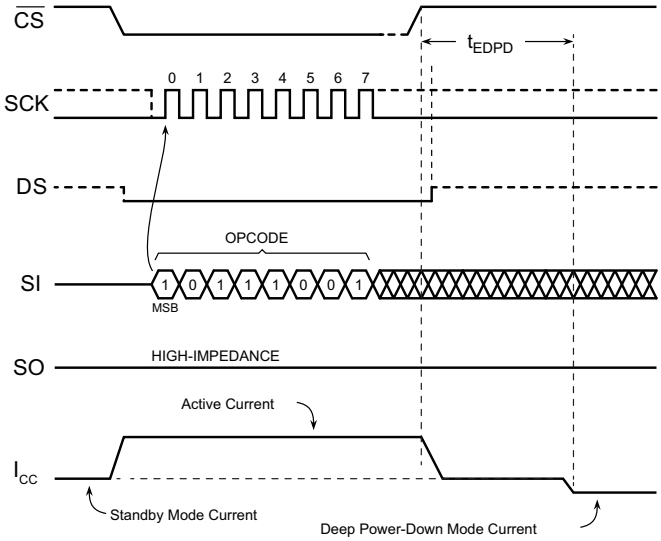


Figure 12-3. Deep Power-Down in Octal Mode — SDR

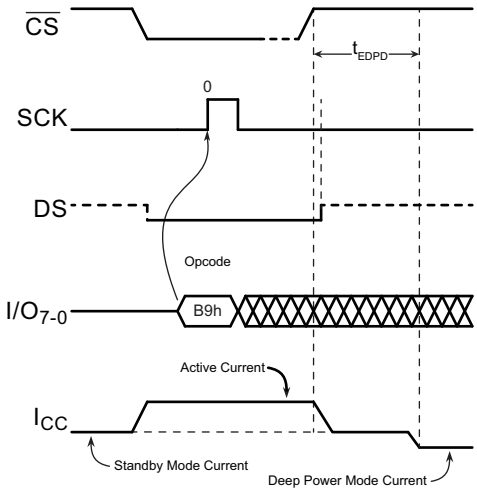
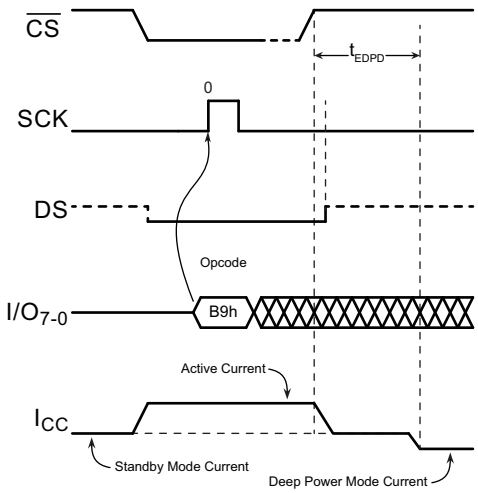


Figure 12-4. Deep Power-Down in Octal Mode — DDR



12.3 Auto Deep Power-Down Mode after a Program or Erase Operation

The auto deep power-down mode after a program or erase operation allows the device to further reduce its energy consumption by automatically entering the deep power-down mode after completing an internally timed program or erase operation. The program or erase operation can be any one of the *Block Erase* or *Chip Erase* commands, *Byte/Page Program* command or *Buffer to Main Memory Page Program without Built-In Erase* commands. Note that the *Buffer Write* command or any of the *Register Write* commands do not cause the device to go into deep power-down mode.

The Auto Deep Power-Down Mode after Program/Erase Operation is enabled by setting the Auto Deep Power-Down Enable bit in Status/Control Register Byte 2. The device remains in the same communication mode when entering and exiting from DPD.

ADPD will be cleared every time the device goes into DPD, so it has to be set again if another Program or Erase operation followed by Auto Deep Power-Down is wanted.

12.4 Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh)

In order to exit the deep power-down mode and resume normal device operation, the *Resume from Deep Power-Down* command must be issued. The *Resume from Deep Power Down* or *Ultra-Deep Power Down* (ABh) command and the *Read Status Register Byte 1* (05h) command are the only commands that the device is recognized while in the deep power-down mode.

To resume from the deep power-down mode, the $\overline{CS}$ pin must first be asserted and command of ABh must be clocked into the device. Any additional data clocked into the device after the command is ignored. When the $\overline{CS}$ pin is deasserted, the device exits deep power-down mode within the maximum time of t_{RDPD} and return to the standby mode. After issuing the ABh command, the user can check if the device is already out of DPD by checking the value of DPDS bit in Status/Control register 1.

The Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh) command can also be used to bring the device out of ultra-deep power-down mode. See Section 12.7, Exit Ultra-Deep Power-Down for details about exiting ultra-deep power-down mode.

To resume from the Ultra-Deep Power-Down mode, the $\overline{CS}$ pin must first be asserted and command of ABh must be clocked into the device. Any additional data clocked into the device after the command is ignored. When the $\overline{CS}$ pin is deasserted, the device will exit the Ultra-Deep Power-Down mode within the maximum time of t_{XUDPD} and return to the standby mode. After issuing the ABh command, the user can check if the device is already out of UDPD by checking the value of UDPDS bit in Status/Control register 1.

After the device has returned to the standby mode, normal command operations such as read array can be resumed.

If the complete command is not clocked in before the $\overline{CS}$ pin is deasserted, or if the $\overline{CS}$ pin is not deasserted on a whole byte boundary (multiples of eight bits), then the device aborts the operation and return to the deep power-down mode. The device remains in the same mode (SPI or octal Mode) while in deep power-down and wakes up in the same mode it had before entering deep power-down.

Figure 12-5. Resume from Deep Power-Down or Ultra-Deep Power-Down

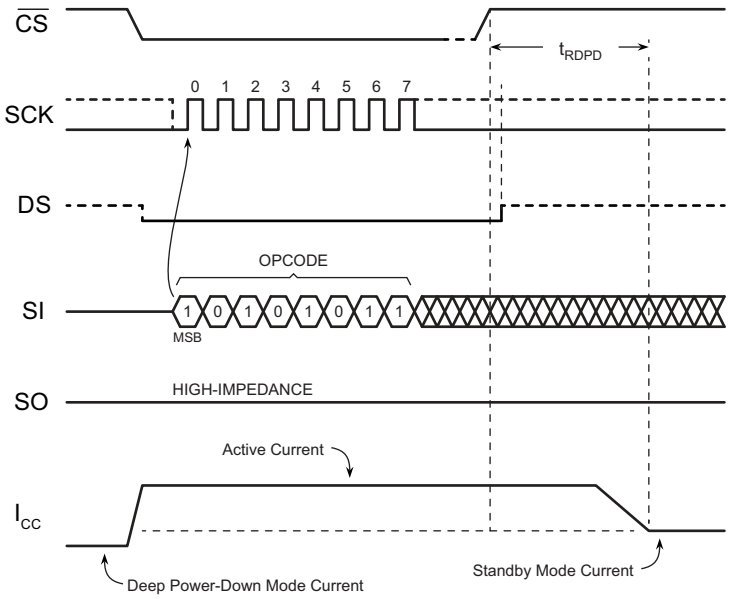


Figure 12-6. Resume from Deep Power-Down or Ultra-Deep Power-Down in Octal Mode — SDR

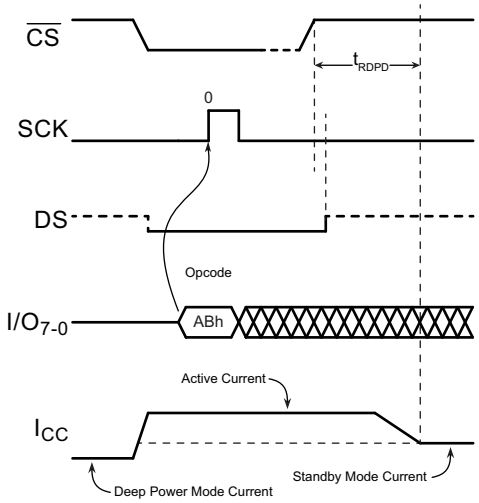
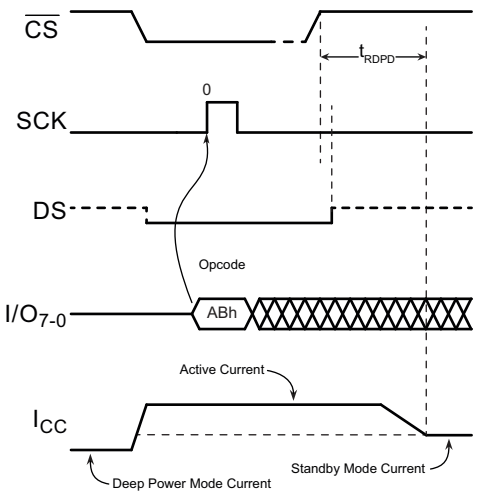


Figure 12-7. Resume from Deep Power-Down or Ultra-Deep Power-Down in Octal Mode — DDR



12.5 Ultra-Deep Power-Down (79h)

The ultra-deep power-down mode allows the device to further reduce its energy consumption compared to the existing standby and deep power-down modes by shutting down additional internal circuitry. When the device is in the ultra-deep power-down mode, the Read Status Register Byte 1 (05h) command is the only command that the device recognizes. When reading the Status/Control registers in this mode, all bits are read as “1”, indicating that the device is in UDPD mode. All other commands, including the *Resume from Deep Power-Down* and *Resume from Ultra-Deep Power-Down* (ABh) commands, are ignored.

Since all write commands are ignored, the mode can be used as an extra protection mechanism against inadvertent or unintentional program and erase operations. Entering the ultra-deep power-down mode is accomplished by simply asserting the $\overline{CS}$ pin, clocking in the command 79h, and then deasserting the $\overline{CS}$ pin. Any additional data clocked into the device after the command is ignored. When the $\overline{CS}$ pin is deasserted, the device enters the ultra-deep power-down mode within the maximum time of t_{EUDPD}.

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

All input pins have to be at valid CMOS levels to minimize power consumption in ultra-deep power-down mode.

Upon recovery from ultra-deep power-down mode, all internal registers except the EPE bit in Status/Control Register Byte 1 are in the power-on default state. See the description in Section 11.1.3, EPE Bit for details of what values EPE may have upon recovery from ultra-deep power-down.

The device wakes up in SPI mode even if it was in octal mode when it entered the ultra-deep power-down state.

Figure 12-8. Ultra-Deep Power-Down

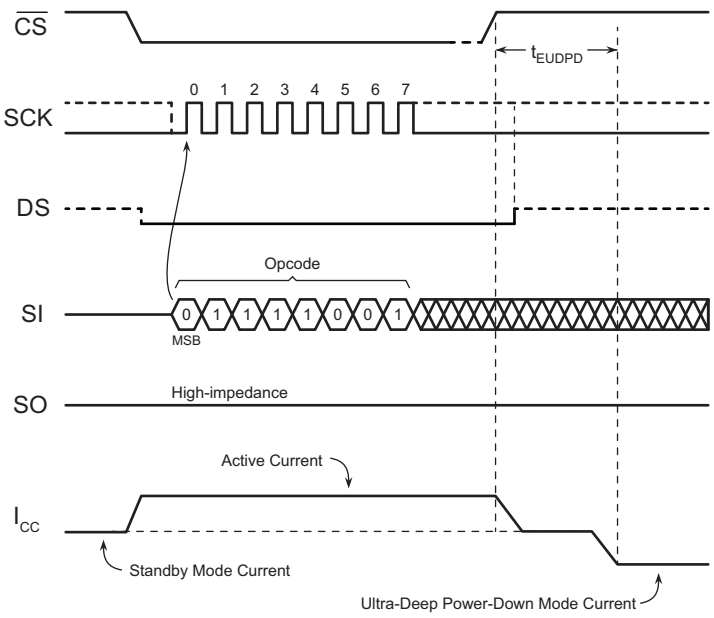


Figure 12-9. Ultra-Deep Power-Down in Octal Mode — SDR

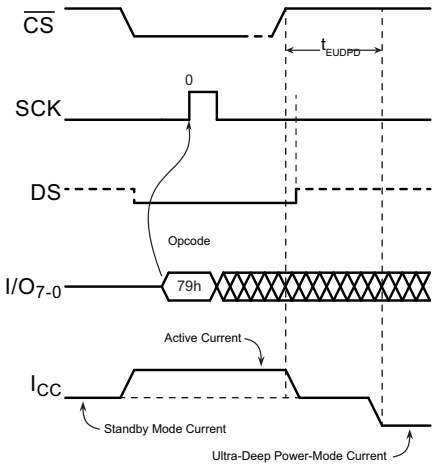
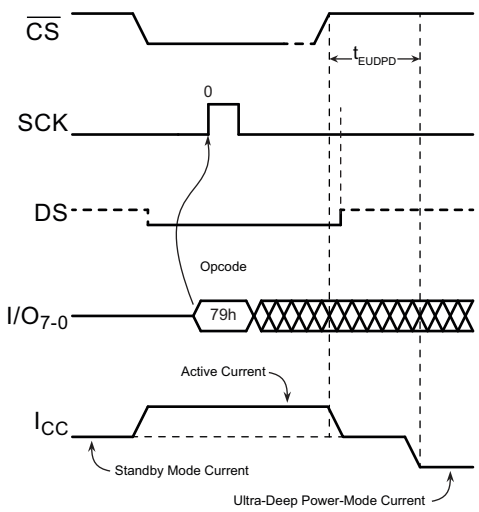


Figure 12-10. Ultra-Deep Power-Down in Octal Mode — DDR



12.6 Auto Ultra-Deep Power-Down Mode after a Program or Erase Operation

The auto Ultra-deep Power-down (UDPD) mode after a program or erase operation allows the device to further reduce its energy consumption by automatically entering the ultra-deep power-down mode after completing an internally timed program or erase operation. The program or erase operation can be any one of the *Block Erase* or *Chip Erase* commands, *Byte/Page Program* command or *Buffer to Main Memory Page Program without Built-In Erase* commands. Note that the *Buffer Write* command or any of the *Register Write* commands do not cause the device to go into ultra-deep power-down mode.

The auto UDPD mode after a program/erase operation is enabled by setting the Auto Ultra-Deep Power-Down Enable (AUDPD) bit in Status/Control Register Byte 2. The AUDPD is cleared every time the device goes into UDPD mode, so it has to be set again if another program or erase operation followed by auto ultra-deep power-down is wanted.

When using auto ultra-deep power-down mode after a program/erase operation, the device switches to standard SPI mode within t_{AUDPD} after the program or erase command is initiated. All Status read operations, both while the program or erase operation is still in progress and after the device has entered UDPD, can therefore be done using standard SPI mode.

12.7 Exit Ultra-Deep Power-Down

To exit from the UDPD mode, a JEDEC hardware reset, exercising the hardware $\overline{\text{RESET}}$ pin, or a power cycling of the device can be performed.

Upon recovery from ultra-deep power-down, all internal registers except the EPE bit in Status/Control Register Byte 1 are at the power-on default state. See the description in Section 11.1.3, EPE Bit for details of what values EPE may have upon recovery from UDPD.

The device wake up in SPI mode even if it was in octal mode when it entered the UDPD state. The system must wait for the device to return to the standby mode before normal command operations can be resumed.

[Note: ***"Chip Select Toggle"*** and ***"Chip Select Low"***, which are the methods used on many other Adesto devices to exit UDPD mode, are not implemented on the ATXP064B, as this would make status read while in UDPD mode impossible without waking up the device.

12.7.1 Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh) command

The Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh) command sequence can be used to wake-up the device from Ultra-Deep Power-Down. See Section 12.4, Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh) for details about this command.

12.7.2 JEDEC Hardware Reset

The *Exit Ultra-Deep Power-Down / JEDEC Hardware Reset* command sequence can be used to wake-up the device from ultra-deep power-down. This sequence can also be used to reset the device to its power on state without cycling power. See Section 12.10 for details on JEDEC Standard Hardware Reset (In-Band Reset). See also Section 13.5, AC Characteristics – All Other Parameters, for timing details.

12.7.3 Hardware Reset

The hardware $\overline{\text{RESET}}$ pin can be used to wake up the device from UDPD mode. This option can also be used to reset the device to a state similar to the power-on state without cycling power. See Section 12.11 for details on Hardware Reset. See also Section 13.5, AC Characteristics – All Other Parameters, for timing details.

12.7.4 Power Cycling

The device can also exit the UDPD mode by power cycling the device. Note that the time t_{PUW} power-up device delay before program or erase allowed applies in this case. See Section 14.1, Power-Up/Power-Down Voltage and Timing Requirements, for details.

12.8 Terminate (F0h)

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

The *Terminate* command can be executed only if the command has been enabled by setting the Terminate Enabled (TERE) bit in the Status/Control register to a logic 1 using the Write Status Register Byte 2 (31h) command. This command should be entered before a program command is entered. If the terminate operation has not been enabled (the TERE bit is in the logic 0 state), then any attempts at executing the *Terminate* command are ignored.

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

The *Terminate* command has no effect on the states of the Configuration Register or TERE bit in the Status/Control register. The WEL bit is reset back to its default state.

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

Figure 12-11. Reset (Terminate Operation)

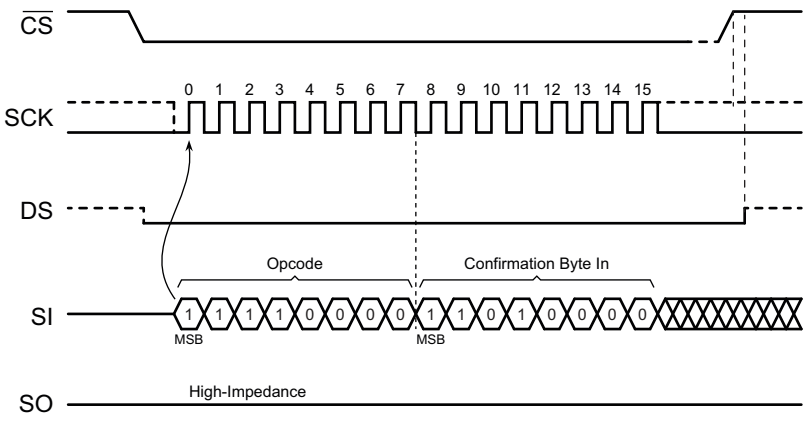


Figure 12-12. Reset (Terminate Operation) in Octal Mode —SDR

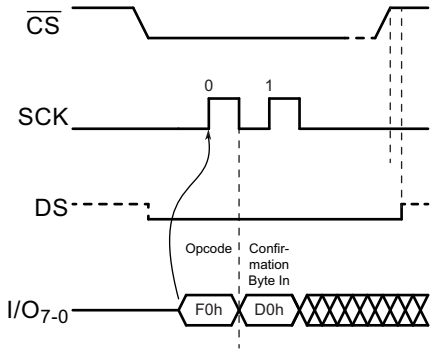
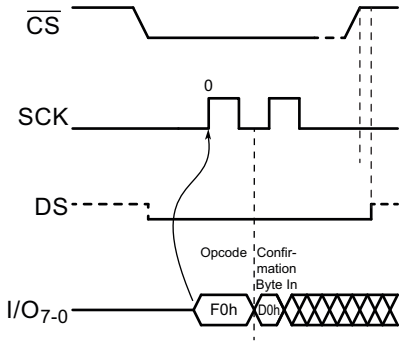


Figure 12-13. Reset (Terminate Operation) in Octal Mode — DDR



12.9 Reset Enable (66h) and Reset (99h)

The ATXP064B provides a software Reset instruction as an alternative to the dedicated RESET pin.

Once the Reset instruction is accepted, any on-going internal operations will be terminated and the device returns to its default power-on state and lose all the current volatile settings. All volatile Status/Control registers, including the Sector Protection Bits, will be reset to their default values. All non-volatile Status/Control registers keep the value they had prior to reset, with the following exception: If the Reset sequence is initiated during a write to a non-volatile Status/Control register, the value of that register may be corrupted.

The device will always revert back to Standard SPI Mode after a Reset.

“Reset Enable (66h)” and “Reset (99h)” instructions can be issued in any mode. To avoid an accidental reset, both instructions must be issued in sequence. Any other instructions other than “Reset (99h)” after the “Reset Enable (66h)” instruction will disable the “Reset Enable” state. A new sequence of “Reset Enable (66h)” and “Reset (99h)” is needed to reset the device. Once the Reset instruction is accepted by the ATXP064B, the device takes approximately t_{SWRES} to reset. During this period, no instructions are accepted.

Data corruption may happen if there is an on-going or suspended internal Erase or Program operation in progress when a Reset instruction sequence is accepted by the device. It is recommended to check the $\overline{RDY}$/BSY bit in Status/Control Register Byte 1 and the PS and ES bits in Status/Control Register Byte 2 before issuing the Reset instruction sequence.

Figure 12-14. Reset Enable and Reset

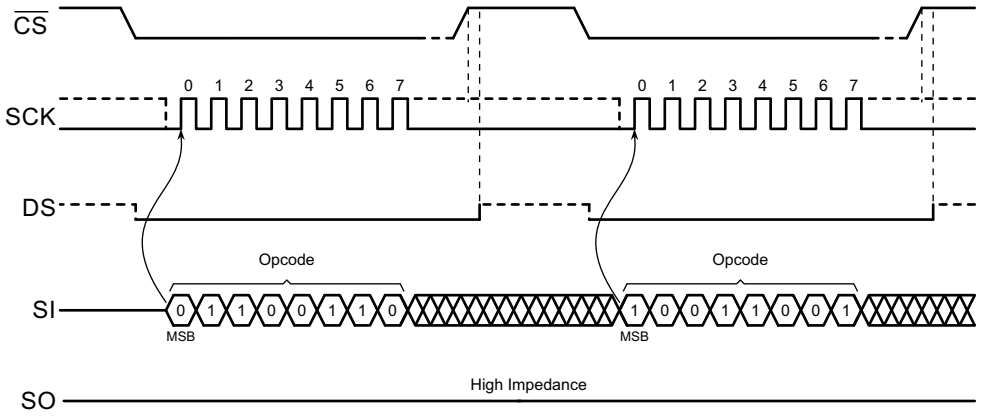


Figure 12-15. Reset Enable and Reset in Octal Mode - SDR

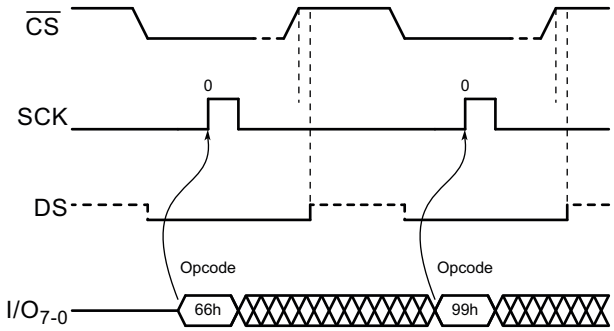
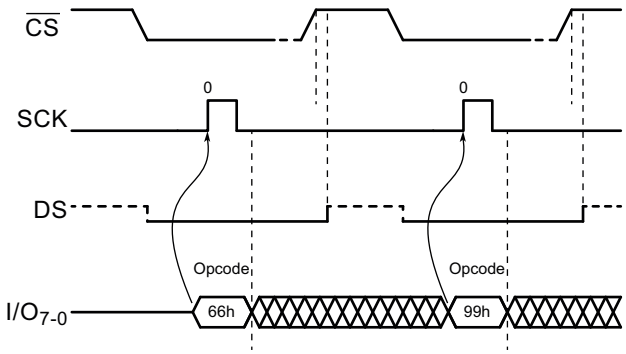


Figure 12-16. Reset Enable and Reset in Octal Mode - DDR



12.10 JEDEC Standard Hardware Reset (In-Band Reset)

The *Exit Ultra-Deep Power-Down / JEDEC Hardware Reset* command sequence can be used to wake up the device from Ultra-Deep Power-Down (UDPD) mode. This sequence can also be used to reset the device to a state similar to the power on state without cycling power. The differences between the power on state and the reset state are described below.

The reset sequence does not use the SCK pin. The SCK pin has to be held low (mode 0) or high (mode 3) through the entire reset sequence. This prevents any confusion with a command, as no command bits are transferred (clocked).

A reset is commanded when the data on the SI pin is 0101 on four consecutive positive edges of the $\overline{CS}$ pin with no edge on the SCK pin throughout. This is a sequence where:

1. $\overline{CS}$ is driven active low to select the device.
2. Clock (SCK) remains stable in either a high or low state.
3. SI is driven low by the bus master, simultaneously with $\overline{CS}$ going active low. No SPI bus slave drives SI during $\overline{CS}$ low before a transition of SCK i.e.: slave streaming output active is not allowed until after the first edge of SCK.
4. $\overline{CS}$ is driven inactive. The slave captures the state of SI on the rising edge of $\overline{CS}$.

The above steps are repeated 4 times, each time alternating the state of SI.

After the fourth $\overline{CS}$ pulse, the slave triggers its internal reset. SI is low on the first $\overline{CS}$, high on the second, low on the third, high on the fourth. This provides a value of 5h, unlike random noise. Any activity on SCK during this time will halt the sequence and a Reset will not be generated.

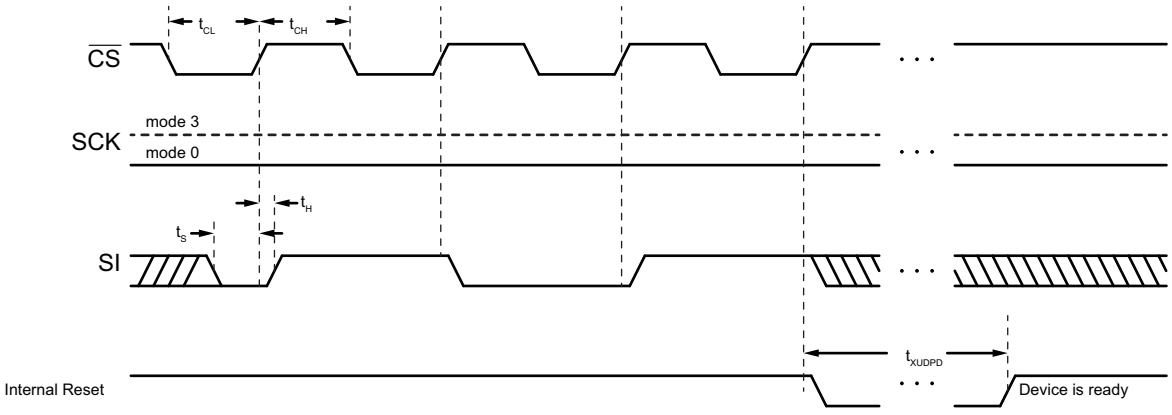
After a JEDEC hardware reset while the device is in UDPD mode, the SRAM buffer is reset to an undefined value. All volatile Status/Control registers, including the Sector Protection Bits, are reset to their default values, except in the following case: If the AUDP bit in Status/Control register byte 2 was set prior to the last *Program* or *Erase* command, the device entered the UDPD mode after a *Program* or *Erase* command finished, and the JEDEC hardware reset is used to wake up the device from UDPD mode, then the EPE bit is not reset. In this case the EPE bit still shows the correct status after the latest *Program* or *Erase* command. All non-volatile Status/Control registers keep the value they had prior to reset.

After a JEDEC hardware reset while the device is in any other mode than UDPD mode, the SRAM buffer keeps the values it had prior to reset, with the following exception: If the reset sequence is initiated during an update of the SRAM buffer, the contents of the SRAM buffer may be corrupted. All volatile Status/Control registers, including the sector protection bits, are reset to their default values. All non-volatile Status/Control registers keep the value they had prior to reset, with the following exception: If the reset sequence is initiated during a write to a non-volatile Status/Control register, the value of that register may be corrupted.

The device always reverts back to standard SPI mode after JEDEC hardware reset.

Figure 12-17 below illustrates the timing for the JEDEC hardware reset operation.

Figure 12-17. JEDEC Standard Hardware Reset



12.11 Hardware Reset

The hardware reset pin ($\overline{\text{RESET}}$) can be used to wake up the device from Ultra-Deep Power-Down (UDPD) mode. This sequence can also be used to reset the device to a state similar to the power-on state without cycling power. The differences between the power-on state and the reset state are described below.

The reset sequence in this case does not use any other pins. A low state on the reset pin ($\overline{\text{RESET}}$) terminates the operation in progress and resets the internal state machine to an idle state. The device remains in the reset condition as long as a low level is present on the $\overline{\text{RESET}}$ pin. Normal operation can resume once the $\overline{\text{RESET}}$ pin is brought back to a high level. The device incorporates an internal power-on reset circuit, so there are no restrictions on the $\overline{\text{RESET}}$ pin during power-on sequences.

After a hardware reset initiated by the $\overline{\text{RESET}}$ pin while the device is in UDPD mode, the SRAM buffer resets to an undefined value. All volatile Status/Control registers, including the sector protection bits, are reset to their default values, except in the following case:

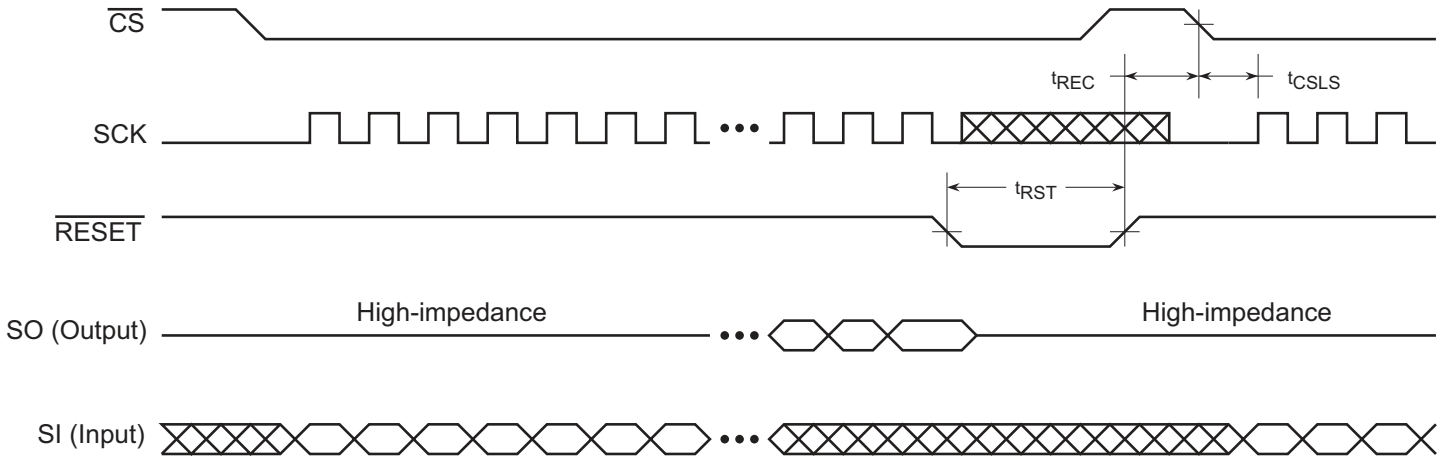
If the AUDP bit in Status/Control register byte 2 was set prior to the last *Program* or *Erase* command, the device enters the UDPD mode after a *Program* or *Erase* command finished, and $\overline{\text{RESET}}$ is used to wake up the device from UDPD mode, then the EPE bit is not reset. In this case the EPE still shows the correct status after the latest *Program* or *Erase* command. All non-volatile Status/Control registers keep the value they had prior to reset.

After a Hardware Reset initiated by the $\overline{\text{RESET}}$ pin while the device is in any other mode than Ultra-Deep Power-Down mode, the SRAM buffer will keep the values it had prior to Reset, with the following exception: If the Reset sequence is initiated during an update of the SRAM buffer, the contents of the SRAM buffer may be corrupted. All volatile Status/Control registers are reset to their default values, except the sector protection bits. The sector protection bits remain unchanged in this case. All non-volatile Status/Control registers keep the value they had prior to reset, with the following exception: If the reset sequence is initiated during a write to a non-volatile Status/Control register, the value of that register may be corrupted.

The device always reverts back to standard SPI mode after $\overline{\text{RESET}}$.

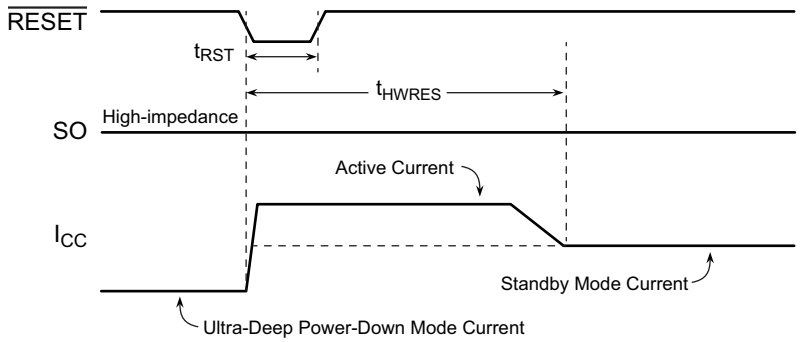
Figures 12-18 and 12-19 below illustrates the timing for the hardware $\overline{\text{RESET}}$ sequence. See AC Characteristics – All Other Parameters for details on reset timing.

Figure 12-18. Hardware Reset Sequence



Note: The $\overline{CS}$ signal should be in the high state before the $\overline{RESET}$ signal is deasserted.

Figure 12-19. Hardware Reset Sequence from UDPD



12.12 Reset Summary

The following table summarizes the effects of the various reset functions:

Table 12-2. Effects of Reset Functions

Action	Volatile Registers	Non-Volatile Registers (1)	Sector Protection Registers	EPE bit
Power Cycle	Default state	Unchanged	Default state (protected)	See Table 11-4
JEDEC Standard Hardware Reset (In-Band Reset) (2)	Default state	Unchanged	Default state (protected)	See Table 11-4
Hardware Reset (not in UDPD) (3)	Default state	Unchanged	Unchanged	See Table 11-4

Table 12-2. Effects of Reset Functions (Continued)

Action	Volatile Registers	Non-Volatile Registers [1]	Sector Protection Registers	EPE bit
Exit Ultra-Deep Power-Down (Power cycle, JEDEC Standard Hardware Reset (In-Band Reset), Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh))	Default state	Unchanged	Default state (Protected)	See Table 11-4
Reset Enable (66h) and Reset (99h)	Default state	Unchanged	Unchanged	See Table 11-4
Terminate (F0h)	WEL and $\overline{RDY}$/BUSY bits in are cleared in Status Register Byte 1	Unchanged	Unchanged	See Table 11-4

1. Non-volatile registers are affected only by programming, not by any Reset sequences or power cycling.
2. JEDEC Reset simulates POR cycle
3. Hardware Reset (when not in UDPD) resets only volatile registers and internal state machines.

12.13 Echo (AA)

The *Echo* command is used for testing the signal integrity and delays on I/O lines. It continues to return the same data byte value as long as $\overline{CS}$ is held low and SCK is being clocked. This command is not supported in SPI mode. In octal mode, the *Echo* command returns the same value on all lines continuously. To create alternating waveforms in octal mode, use the *Echo with Inversion* command described in the following subsection.

Figure 12-20. Echo in Octal Mode — SDR

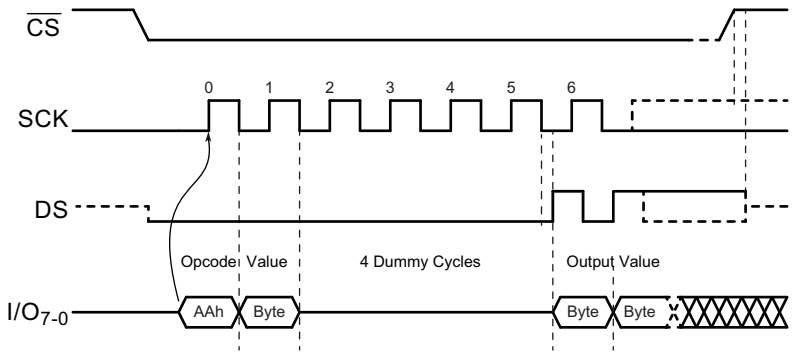
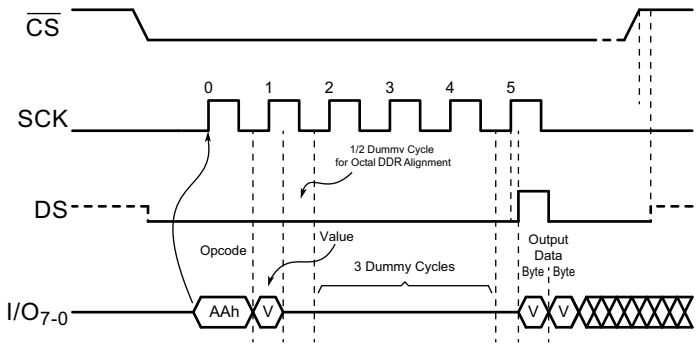


Figure 12-21. Echo in Octal Mode — DDR



12.14 Echo with Inversion (A5)

The *Echo with Inversion* command is used for testing the signal integrity and delays on I/O lines. It continues to output data as long as $\overline{CS}$ is held low and SCK is being clocked. It alternates between transmitting the input data value it received and the inverse value. It is mostly intended for use in octal mode, but can also be used in the other modes for generating specific waveforms.

In octal mode, any value sent creates an alternating pulse train. The value entered with the command determines the value of the first byte returned. The next byte is inverted, and so on. So a value of 00h or FFh causes all the I/O lines to have the same value and alternate between 00h and FFh, while a value of AAh or 55h alternates between AAh and 55h.

This command is not supported in SPI mode.

Figure 12-22. Echo with Inversion in Octal Mode — SDR

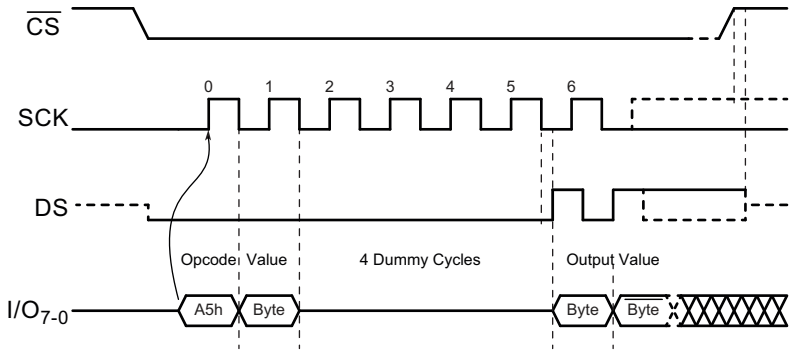
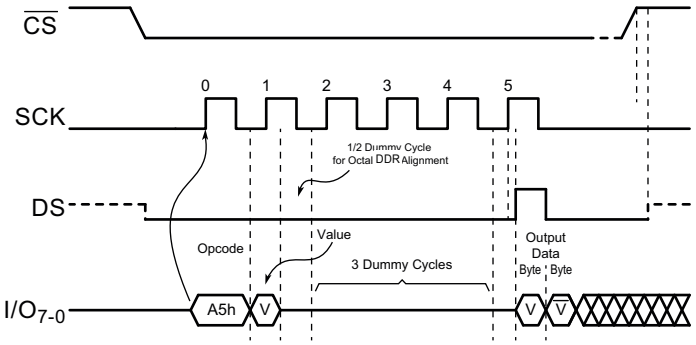


Figure 12-23. Echo with Inversion in Octal Mode — DDR

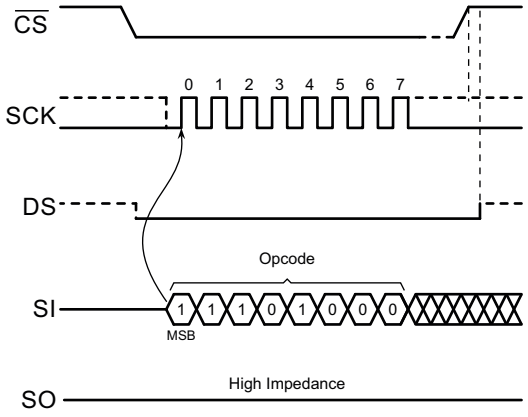


12.15 Enter Octal Mode (E8h)

The *Enter Octal Mode* command can only be entered from Standard SPI Mode. Before the *Enter Octal Mode* command can be issued, the Write Enable (06h) command must have been previously issued to set the WEL bit in the Status/Control register to a logic "1".

To issue the *Enter Octal Mode* command, the $\overline{CS}$ pin must first be asserted and the command of E8h must be clocked into the device.

Figure 12-24. Enter Octal Mode



12.16 Return to Standard SPI Mode (FFh)

In order to exit the octal mode and return to the standard SPI mode, a *Return to Standard SPI Mode* command must be issued.

Before the *Return to Standard SPI Mode* command can be issued, the Write Enable (06h) command must have been previously issued to set the WEL bit in the Status/Control register to a logic 1.

To issue the *Return to Standard SPI Mode* command, the $\overline{CS}$ pin must first be asserted and the command of FFh must be clocked into the device.

The WEL bit in Status/Control register byte 1 is reset back to the logic '0' state. The $\overline{SDR}$/DDR and OME bits in Status/Control register byte 2 are all set to 0. The device returns to standard SPI mode, SDR, independent of which mode it was in prior to issuing this command.

Figure 12-25. Return to Standard SPI Mode from Octal Mode — SDR

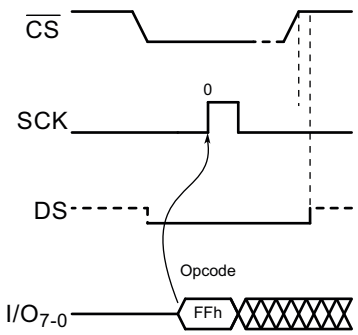
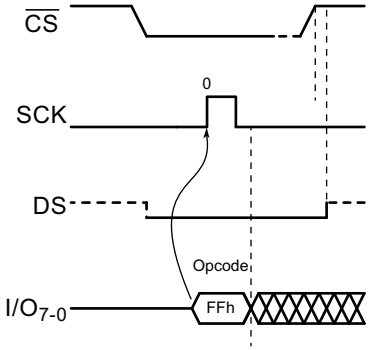


Figure 12-26. Return to Standard SPI Mode from Octal Mode — DDR



12.17 Read SFDP (5Ah)

The ATXP064B contains a 256-byte Serial Flash Discoverable Parameter (SFDP) register.

The SFDP register can be sequentially read in a similar fashion to the read array operation up to the maximum clock frequency specified by f_{CLK}. To read the SFDP security register, the $\overline{CS}$ pin must first be asserted and the command of 5Ah must be clocked into the device. After the command 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 SFDP security register. Following the three address bytes, eight additional dummy cycles (8 dummy cycles = 1 dummy byte in SPI mode) need to be clocked into the device. Note that this means eight complete clock cycles, also for DDR modes. Octal DDR mode has an additional 1/2 dummy cycle, due to clock alignment.

Note that even though this device otherwise is using 4-byte addressing, SFDP is always using 3-byte addressing.

Note also that in DDR mode, the SFDP command can only be used for addressing data on a 32-bit (4-byte) boundary. (Address bits A_1 and A_0 both have to be 0).

After the three address bytes and the dummy cycles have been clocked in, additional clock cycles will result in data being output on the I/O pin(s). The data is always output with the MSB of a byte first. When the last byte (0000FFh) of the SFDP security register has been read, the device continues reading back at the beginning of the register (000000h). No delays are incurred when wrapping around from the end of the register to the beginning of the register.

Deasserting the $\overline{CS}$ pin terminates the read operation and places the I/O pin(s) into a high-impedance state. The $\overline{CS}$ pin can be deasserted at any time and does not require that a full byte of data be read. The format of the SFDP register follows the format provided in JEDEC Standard No. 216 Rev D.

Figure 12-27. Read SFDP

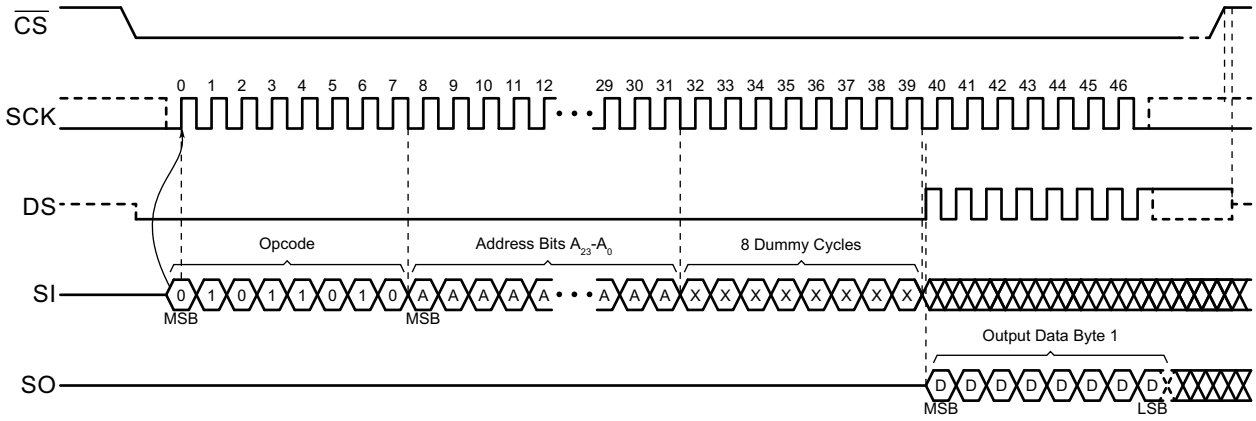


Figure 12-28. Read SFDP in Octal Mode — SDR

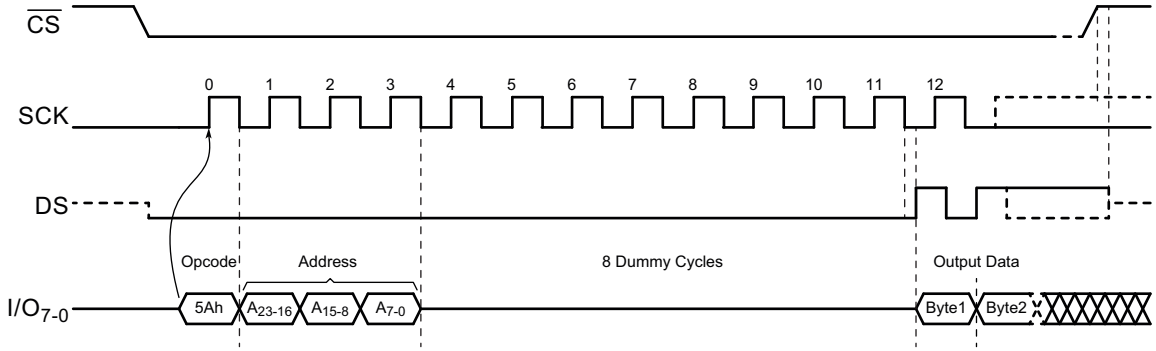
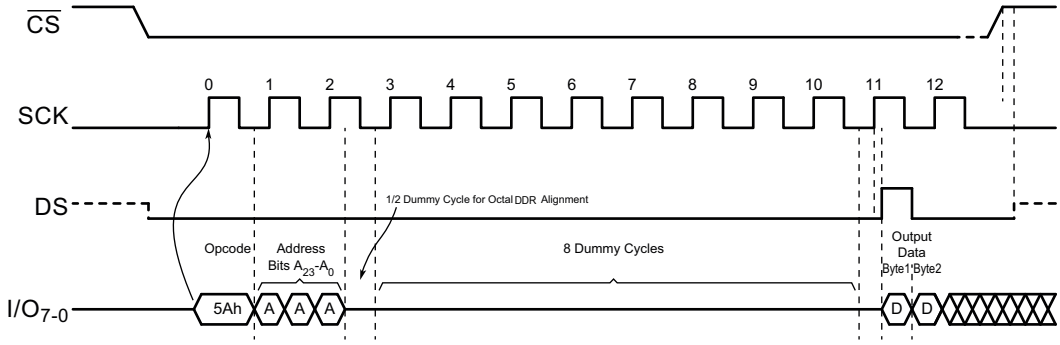


Figure 12-29. Read SFDP in Octal Mode — DDR



12.17.1 SFDP Parameters

For more information on the SFDP parameters for the EcoXiP ATXP064B device, please contact Adesto.

13. Electrical Specifications

13.1 Absolute Maximum Ratings*

Temperature under bias . . . . . -55°C to +125°C
Storage temperature . . . . . -65°C to +150°C
All input voltages (including NC Pins) with respect to ground . . . . . -0.6V to + V_{CC} + 0.5V
All output voltages with respect to ground . . . . . .-0.6V to V_{CC} + 0.5V

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

13.2 DC and AC Operating Range

Parameter	Condition	Range
Operating temperature (case, packaged parts)	Industrial	-40°C to 85°C
Operating temperature (KGD)	Extended	-40°C to 105°C
V_{CC} power supply		1.70 - 1.95V

13.3 DC Characteristics

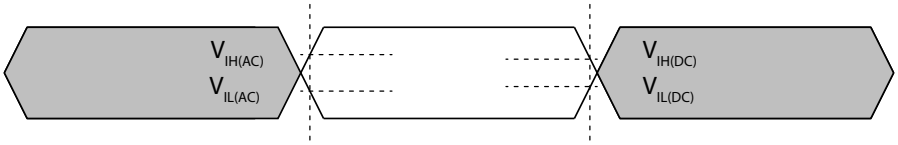
Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{UDPD}	Ultra-Deep Power-Down Current	$\overline{CS}$ = $V_{CC.}$ All other inputs at 0V or V_{CC}		0.2	1	µA
I_{DPD}	Deep Power-Down Current	$\overline{CS}$ = $V_{CC.}$ All other inputs at 0V or V_{CC}		4	10	µA
I_{SB}	Standby Current, SPI Mode	$\overline{CS}$ = $V_{CC.}$ All other inputs at 0V or V_{CC}		20	35	µA
I_{SB}	Standby Current, Octal Mode	$\overline{CS}$ = $V_{CC.}$ All other inputs at 0V or V_{CC}		35	50	µA
I_{CC2}	Active current, read operation, packaged part (QFN24/BGA24/ WLCSP)	1-pin SDR (Read @ 1pf load)	66 MHz	10	11	mA
		8-pin DDR (Read @ 1pf load)	66 MHz	22		mA
			100 MHz	28		mA
			133 MHz	34	37	mA
		8-pin SDR (Read @ 1pf load)	66 MHz			
			100 MHz			
			133 MHz			

13.3 DC Characteristics (Continued)

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{CC3}	Active Current, Program Operation	$\overline{CS}$ = V_{CC}		11.5	13	mA
I_{CC4}	Active Current, Erase Operation	$\overline{CS}$ = V_{CC}		10.5	12	mA
I_{LI}	Input Load Current	All inputs at CMOS levels (0V to V_{CC} x 0.3 or V_{CC} x 0.7 to V_{CC})			±2 (1)	µA
I_{LO}	Output Leakage Current	All inputs at CMOS levels (0V to V_{CC}x 0.3 or V_{CC}x 0.7 to V_{CC})			1	µA
V_{IL}	Input Low Voltage		-0.3		V_{CC} x 0.35	V
V_{IH}	Input High Voltage		V_{CC} x 0.65		V_{CC} + 0.3	V
V_{OL}	Output Low Voltage	I_{OL} = 2mA			0.45	V
V_{OH}	Output High Voltage	I_{OH} = -2mA	V_{CC} - 0.45			V
V_T	Clock Reference Point for Timing Measurements			V_{CC} x 0.5		V
Z_0	Nominal line impedance	Driver type 0x00, x1		50		Ω
		Driver type 0x02, x0.75		66		Ω
		Driver type 0x03, x0.5		100		Ω
T_d	Transmission line delay	DDR, f = 133 MHz			0.5	ns
C_{Load}	Capacitive load	DDR, f = 133 MHz, Driver type 0x00			6	pF
		DDR, f = 133 MHz, Driver type 0x02			4.5	
		DDR, f = 133 MHz, Driver type 0x03			3	
C_{in}	Input Capacitance	Packaged parts, WLCSP		4		pF
		KGD		3		

1. The RESET pin has an internal pull-up resistor. Its input load current is +2 / -17 µA maximum.

Figure 13-1. Input Levels and AC Timing Definition



Note:
AC Level is guaranteed transition point.
DC Level is hysteresis.

= Don't Care

13.4 AC Characteristics - Maximum Clock Frequencies

Symbol	Parameter	Mode	Min	Typ	Max	Units
f_{CLK}	Maximum Clock Frequency for All Operations (excluding 03h, 13h, 5Ah and D4h commands)	SPI mode			66	MHz
		Octal Mode			133	
f_{RDLF}	Maximum Clock Frequency for 03h, 13h, 5Ah and D4h commands (Read Array – Low Frequency, Read SFDP, and Buffer Read)				50	MHz

13.5 AC Characteristics – All Other Parameters

Note: All all parameters are specified for industrial temperature range (-40°C to 85°C). For extended range (KGD) parameters, contact Adesto.
All parameters are specified for max frequency of 133 MHz

Symbol[4][1]	Parameter		Min	Typ	Max	Units
t_{CLKH}	Input SCK High Time	SPI mode (excluding 03h and 13h commands)	6.75			ns
		SPI mode (03h and 13h commands)	9			
		Octal mode, SDR	3			
t_{CLKL}	Input SCK Low Time	SPI mode (excluding 03h and 13h commands)	6.75			ns
		SPI mode (03h and 13h commands)	9			
		Octal mode, SDR	3			
t_{CKMPW}	Input SCK: Minimum pulse width	Octal mode, DDR	3.375			ns
t_{CLKR}	Input SCK Rise Time, Peak-to-Peak (Slew Rate)				1	ns
t_{CLKF}	Input SCK Fall Time, Peak-to-Peak (Slew Rate)				1	ns
t_{PERIOD}	Input SCK, Data Strobe: Cycle time data transfer mode		7.5			ns
SR_{CLK}	Input SCK: Slew rate		0.75			V/ns

13.5 AC Characteristics – All Other Parameters (Continued)

Note: All all parameters are specified for industrial temperature range (-40°C to 85°C). For extended range (KGD) parameters, contact Adesto.
All parameters are specified for max frequency of 133 MHz

Symbol[4][1]	Parameter		Min	Typ	Max	Units
t_{CKDCD}	Input SCK: Duty cycle distortion		0		0.375	ns
SR_{DS}	Data Strobe: Slew rate		0.75			V/ns
t_{DSDCD}	Data Strobe: Duty cycle distortion[3]		0		0.3	ns
t_{DSMPW}	Data Strobe: Minimum pulse width		3.075			ns
t_{RPRE}	Output Data: Read pre-amble		0.4			t_{PERIOD}
t_{ISU}	Input set-up time (referenced to SCK):		0.8			ns
t_{IH}	Input hold time (referenced to SCK):		0.8			ns
SR_{IN}	Input Slew rate		0.75			V/ns
t_{R0},t_{F0}	Rise/Fall Time, Driver type 0x00, x1; C_L = 15pF			1.3	2.0	ns
	Rise/Fall Time, Driver type 0x02, x0.75; C_L = 10pF					ns
	Rise/Fall Time, Driver type 0x03, x0.5; C_L = 7.5pF					ns
R_{FR}	Ratio of fall time to rise time			1	1.4	
t_{CSH}	Chip Select High Time	After a Read command reading 16 bytes or more	15			ns
t_{CSLCKH} (t_{CSLS})[4]	Chip Select Low Setup Time (relative to SCK)	Up to 100 MHz	5			ns
		100 - 133 MHz	6			
t_{CKLCSH} (t_{CSLH})[4]	Chip Select Low Hold Time (relative to SCK)	Up to 100 MHz	5			ns
		100 - 133 MHz	6			
t_{CSHCKH} (t_{CSHS})[4]	Chip Select High Setup Time (relative to SCK)	Up to 100 MHz	5			ns
		100 - 133 MHz	6			
t_{CKLCSL} (t_{CSHH})[4]	Chip Select High Hold Time (relative to SCK)	Up to 100 MHz	5			ns
		100 - 133 MHz	6			
t_{CSLDS}	CS low to DS low				20	ns
t_{DSLCSH}	DS low to CS high		80% of t_{PERIOD}			ns
t_{CSHDST}	CS high to DS tri-state				80% of t_{PERIOD}	ns
t_{DS}[2]	Data In Setup Time		2			ns
t_{DH}[2]	Data In Hold Time		0.4			ns
t_{DIS}	Output Disable Time				7.5	ns
t_V	Output Valid Time (SCK to DS or I/O out); C_L =10pF				6	ns

13.5 AC Characteristics – All Other Parameters (Continued)

Note: All all parameters are specified for industrial temperature range (-40°C to 85°C). For extended range (KGD) parameters, contact Adesto.
All parameters are specified for max frequency of 133 MHz

Symbol[4][1]	Parameter	Min	Typ	Max	Units
t_{OH}	Output Hold Time	0			ns
t_{RQ}	Output Skew (referenced to DS)			0.55	ns
t_{RQH}	Output Hold Skew (referenced to DS)			0.55	ns
SR_{OUT}	Output Slew Rate	0.75			V/ns
t_{WPS}	Write Protect Setup Time	20			ns
t_{WPH}	Write Protect Hold Time	100			ns
t_{EDPD}	Chip Select High to Deep Power-Down			0.5	µs
t_{EUDPD}	Chip Select High to Ultra-Deep Power-Down			0.5	µs
t_{AUDPD}	Chip Select High to Standard SPI Mode enabled (AUDPD enabled)			0	µs
t_{SWRST}	Software Reset Time			40	µs
t_{CL}	JEDEC Hardware Reset - Chip Select low time	50			ns
t_{CH}	JEDEC Hardware Reset - Chip Select high time	50			ns
t_S	JEDEC Hardware Reset - SI Setup time	5			ns
t_H	JEDEC Hardware Reset - SI hold time	5			ns
t_{RST}	$\overline{\text{RESET}}$ Pulse Width (Hardware $\overline{\text{RESET}}$ Pin)	10			µs
t_{HWRES}	Hardware Reset Recovery Time			70	µs
t_{SWRES}	Software Reset Recovery Time			30	µs
t_{XUDPD}	JEDEC Reset Recovery Time		50	70	µs
t_{RDPD}	Chip Select High to Standby Mode (Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh) Time)		5	8	µs

Notes:
1. Not 100% tested (value guaranteed by design and characterization).
2. Only applicable for Standard Serial Mode.
3. The Data Strobe Duty Cycle Distortion will be in addition to whatever distortion there is on the incoming SCK clock. There is no clock recovery circuit that will attempt to clean up the incoming clock signal before DS is created.
4. Symbols used match JEDEC xSPI spec (JESD251) where applicable. (Symbols in parenthesis are the symbols used for the same parameters in other Adesto datasheets.)

13.6 Program and Erase Characteristics

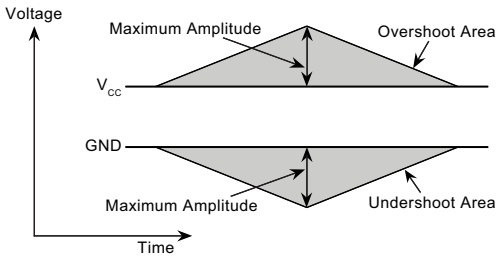
Symbol	Parameter		1.7V - 1.95V Min	Typ[4]	Max	Units
t_{PP}[1][2]	Page Program Time (256 Bytes)			4	12	ms
t_{BP}[2]	Byte Program Time			25		µs
t_{BLKE}[1][2]	Block Erase Time	4 Kbytes		70	250	ms
		32 Kbytes		500	1000	
		64 Kbytes		1000	1600	
t_{CHPE}[1][2]	Chip Erase Time			60	80	sec
t_{SUSP}	Suspend Time	Program		10	20	µs
		Erase		25	40	
t_{RES}	Resume Time	Program		10	20	µs
		Erase		12	20	
t_{OTPP}	OTP Security Register Program Time			2	6	ms
t_{WRSR}	Write Status/Control register Time - Volatile Registers (all commands)				200	ns
t_{WRSRNV}	Write Status/Control register Time - Non-Volatile Registers (71h command)			20	40	ms

Note:
1. Maximum values indicate worst-case performance after 100,000 Program/Erase cycles.
2. Programming and erase times for concurrent read and write will be longer than for regular programming. The increase in programming and erase times will depend on the frequency of read commands. Worst case programming times may be as much as 8 times longer than for regular programming, while worst case erase times may be as much as 2 times longer than for regular erase.
3. Not 100% tested (value guaranteed by design and characterization).
4. Typical values indicate performance for beginning of life for new devices at room temperature and nominal voltage (1.8V) or higher.

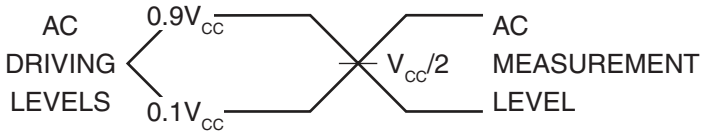
13.7 Overshoot/Undershoot Specification

Parameter	1.70V - 1.95V Min	Max	Units
Maximum peak amplitude allowed for overshoot area		0.3	V
Maximum peak amplitude allowed for undershoot area		0.3	V
Maximum area above V_{CC}		1.2	V*ns
Maximum area below GND		1.2	V*ns

Figure 13-2. Overshoot / Undershoot Definition

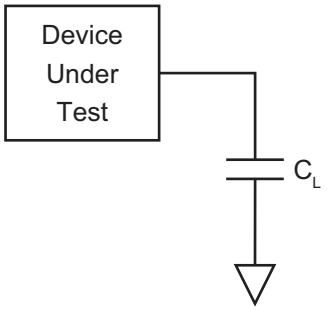


13.8 Input Test Waveforms and Measurement Levels



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

13.9 Output Test Load



14. Power-On / Reset State

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

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

As the device initializes, there is a transient current demand. The system needs to be capable of providing this current to ensure correct initialization. During power-up, the device must not be READ for at least the minimum t_{VCSL} time after the supply voltage reaches the minimum V_{CC} level (V_{CC} min). While the device is being powered-up, the internal Power-on Reset (POR) circuitry keeps the device in a reset mode until the supply voltage rises above the minimum V_{CC}. During this time, all operations are disabled and the device does not respond to any commands. The internal reset level, V_{POR}, is lower than V_{CC} min. However, the t_{VCSL} time is measured from when V_{CC} reaches V_{CC} min.

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

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

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

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

Figure 14-1. Power-Up Timing

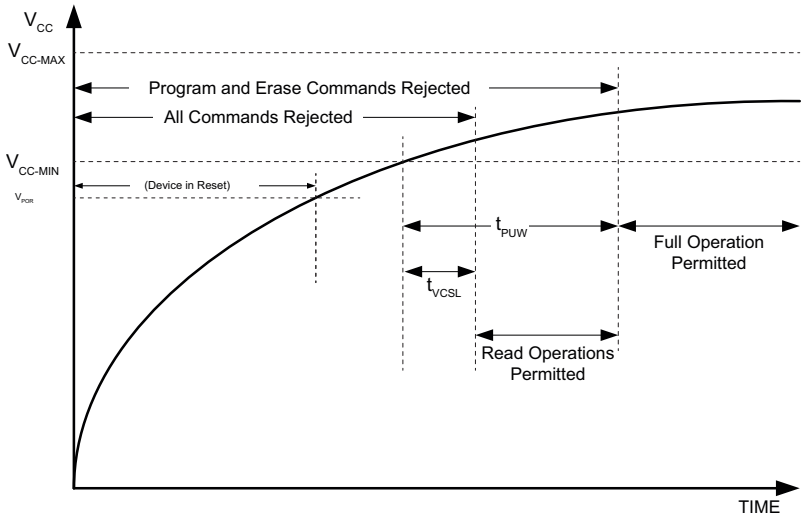
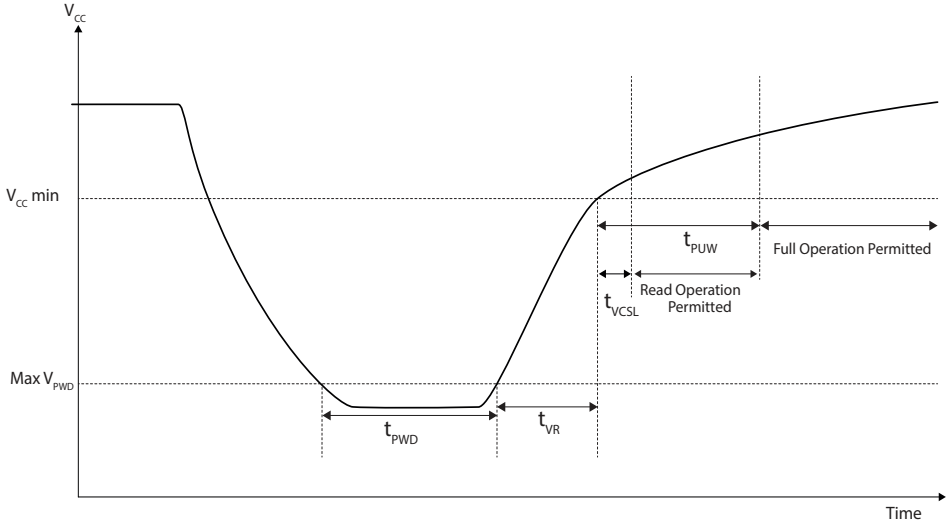


Figure 14-2. Power-Up After Brown-Out Timing



15. AC Waveforms

Figure 15-1. Serial Input Timing

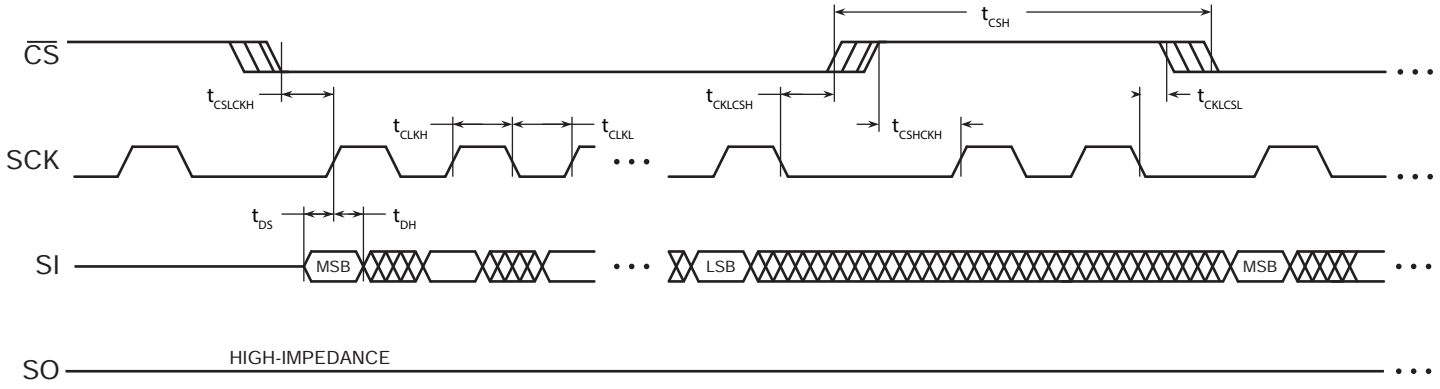


Figure 15-2. Serial Output Timing

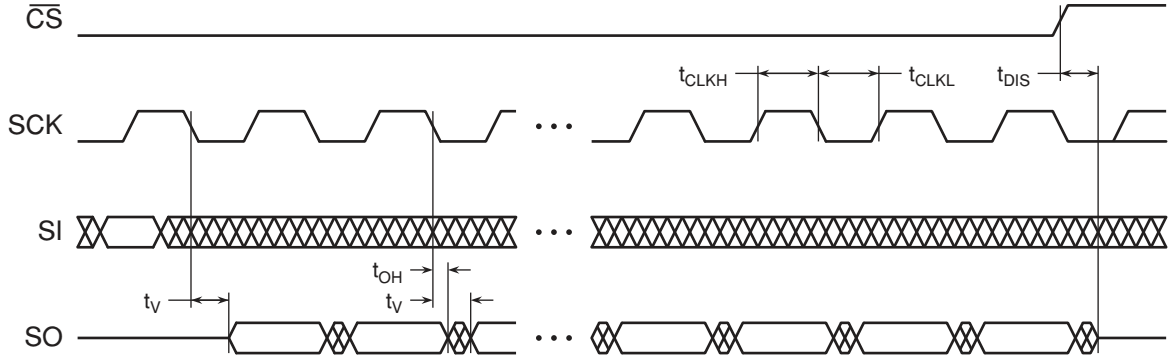


Figure 15-3. $\overline{WP}$ Timing for Write Status Register Command

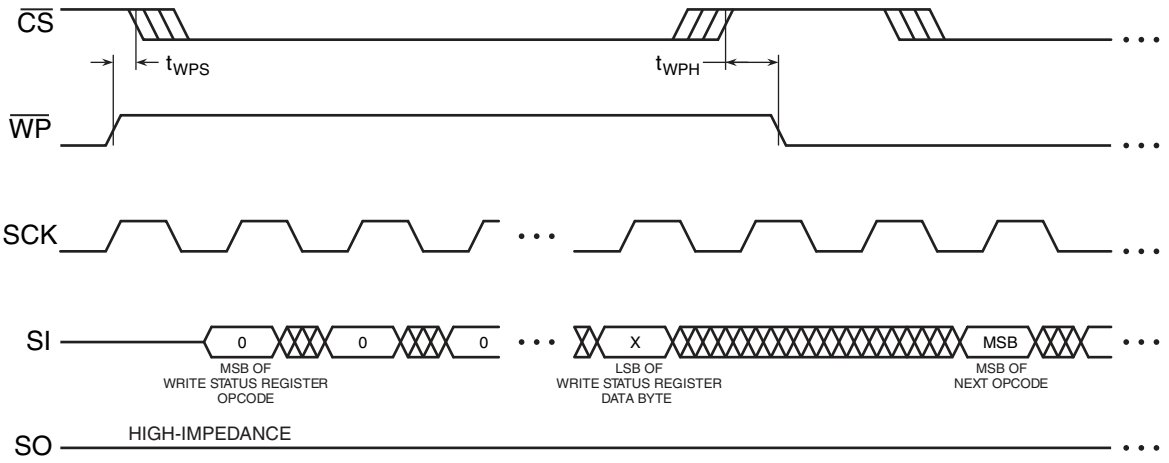


Figure 15-4. Device Input Timing DDR

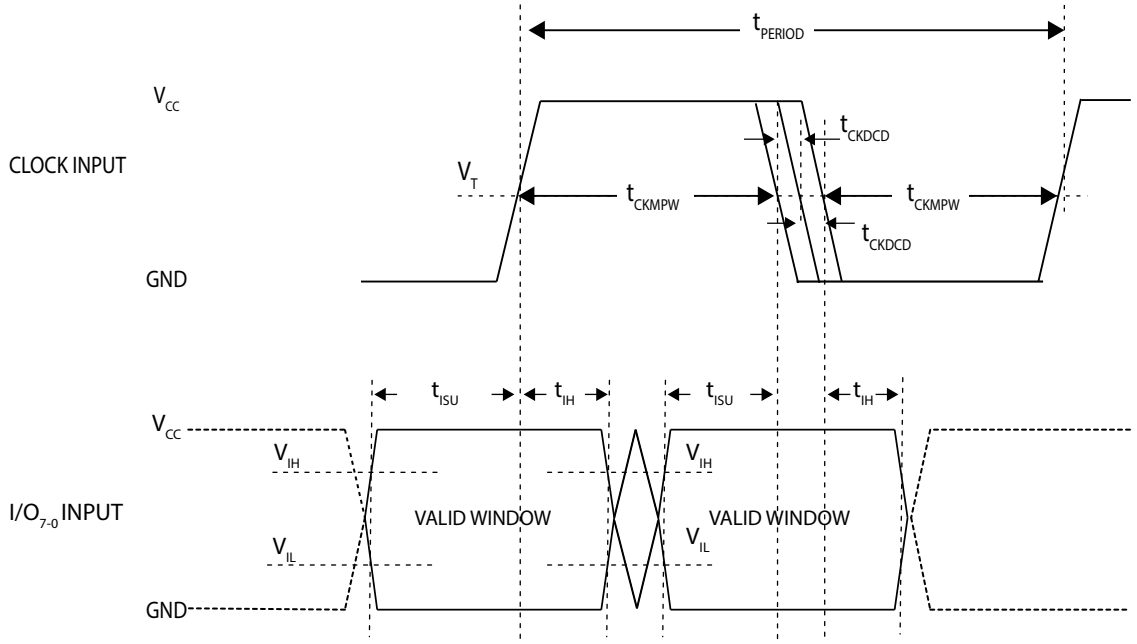
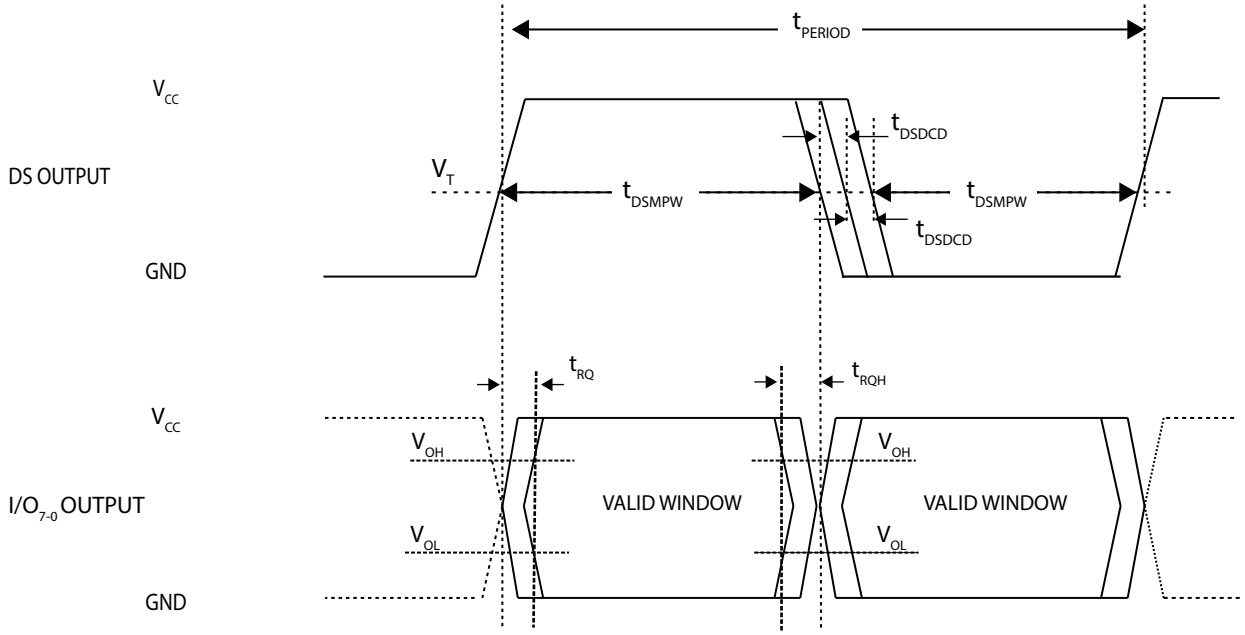


Figure 15-5. Device Output Timing DDR



16. Ordering Information

16.1 Ordering Code Detail

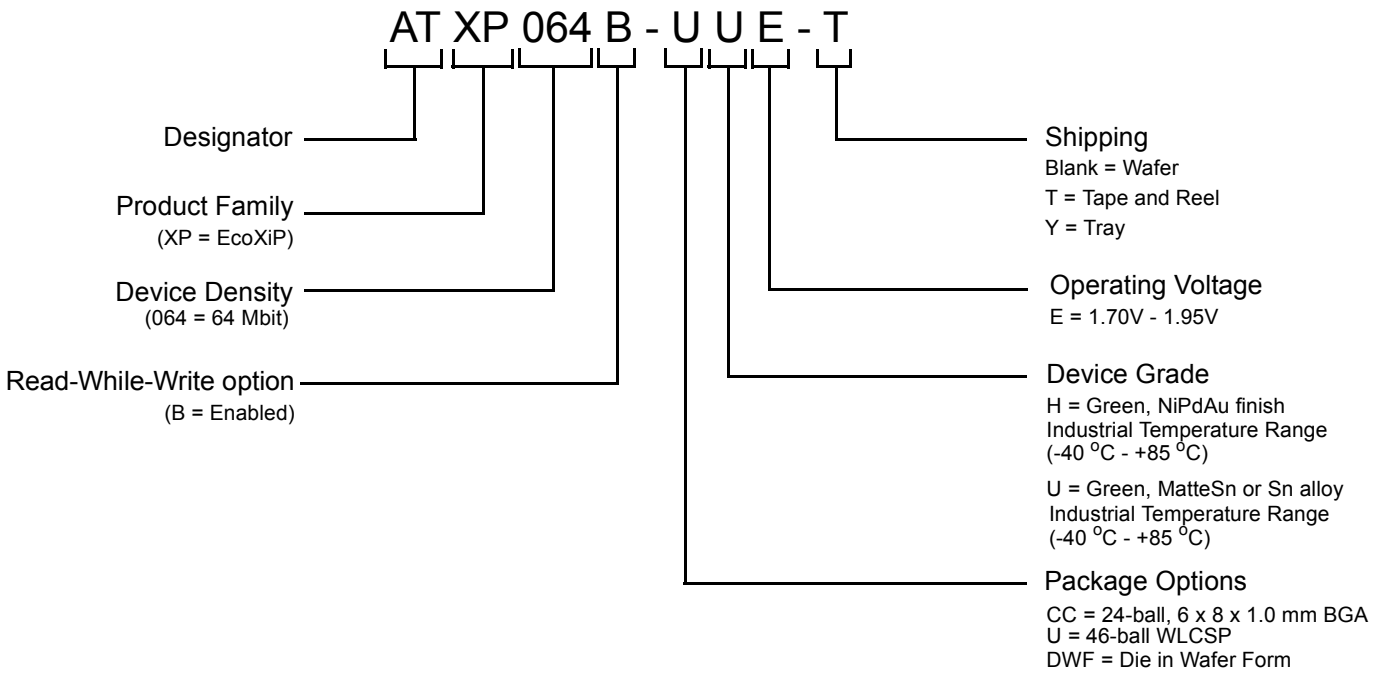


Table 16-1. ATXP064B Ordering Codes

Ordering Code [1]	Package	Lead Finish	Operating Voltage	Maximum Frequency (MHz)	Operating Range [2]
ATXP064B-CCUE-Y	24CBGA	SnAgCu	1.70V to 1.95V	133 [3]	Industrial (-40°C to +85°C)
ATXP064B-CCUE-T					
ATXP064B-UUE-T [4]	46WLCSP				
ATXP064B-DWF[4]	DWF	N/A			Extended (-40°C to +105°C)

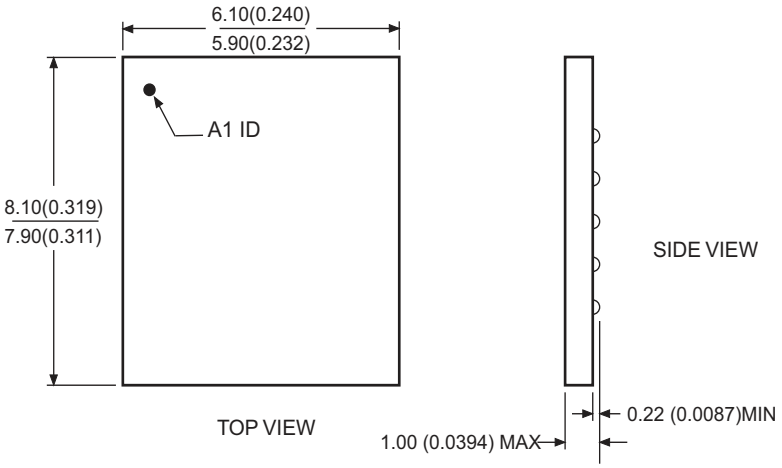
1. The shipping carrier option code is not marked on the device.
2. Contact Adesto for extended temperature devices. (-40°C to +105°C).
3. See Section 13.4, AC Characteristics - Maximum Clock Frequencies on page 93 for maximum operating frequencies for different commands.
4. Contact Adesto for mechanical drawing or die sales information.

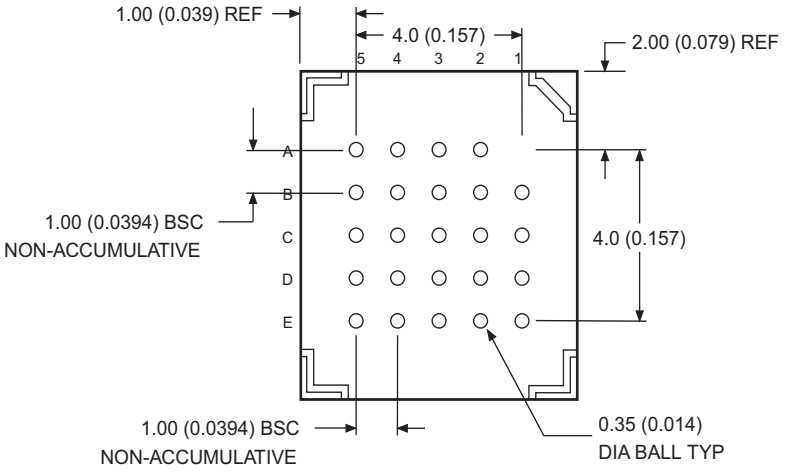
Package Name	Package Type
24CBGA	24C2, 24-ball (5 x 5 Array), 6 x 8 x 1.0mm Body, 1.0 mm Ball Pitch Chip-scale Ball Grid Array Package (CBGA)
46WLCSP	46-ball Wafer Level Chip Scale Package (WLCSP) die Ball Grid Array (dBGA)
DWF	Die in Wafer Form

17. Packaging Information

17.1 24C2 - 24-ball CBGA

Dimensions in Millimeters and (Inches).
Controlling dimension: Millimeters.

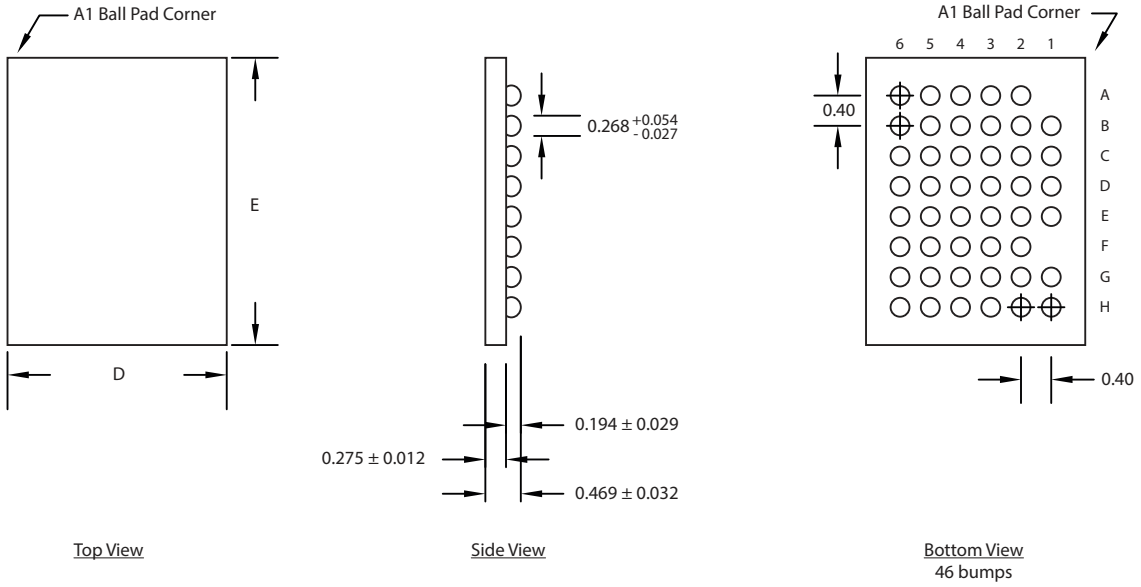




06/2016

Package Drawing Contact : contact@adestotech.com	TITLE 24C2, 24-ball (5 x 5 Array), 6 x 8 x 1.0 mm Body, 1.0 mm Ball Pitch Chip-scale Ball Grid Array Package (CBGA)	DRAWING NO. 24C2	REV. B

17.2 CS46 - 46-ball WLCSP



Top View

Side View

Bottom View
46 bumps

D and E dimensions: Contact Adesto

Dimensions in Millimeters

2019/Feb/19

	TITLE	GPC	DRAWING NO.	REV.
Package Drawing Contact: contact@adestotech.com	46-WLCSP, 46-ball, modified 6 x 8 array 0.4mm pitch, die Ball Grid Array (dBGA), WLCSP	PTI	CS46-002	0A

18. Revision History

Revision	Date	Change History
A	4/2019	Initial release.
B	10/2019	Removed references to ATXP064R. Removed references to Quad Mode. Removed references to QPI mode. Updated legal page. Updated Command Listing tables 6-1, 6-2, and 6-3. Removed all Quad and QPI related commands. Updated to new Adesto template. Removed SFDP Parameter tables in Section 12.16.1. Updated ICC2 values in AC Characteristics table. Changed Vmin from 1.65V to 1.7V throughout document. Updated size of OTP register to 256 bytes. Removed comment on sequential programming. Changed I_{LI} parameter in DC Characteristics table from 1 to 2 μA. Updated notes in Section 13.5, AC Characteristics. Updated notes in Section 13.6, Program and Erase Characteristics. Updated Device ID Byte 1 Product Version Code in Table 12-1 from 08h to 01h.

Corporate Office

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

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Adesto Technologies:

ATXP064B-CCUE-T ATXP064B-UUE-T