ATXP064 / ATXP064R

266 Mbyte/s High-Performance 64Mbit XiP System Accelerating Memory

ADVANCE DATASHEET

Features

- Optimized for eXecute-in-Place (XiP) operations
  - Reduces average latency for improving CPU performance
  - Enables 40% higher CPU performance than the basic Octal SPI protocol
- 133MHz Maximum Operating Frequency
  - Up to 266-Mbytes per second data transfer in Octal DTR mode
- Concurrent Read and Write *(*ATXP064R *only)*
  - 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
  - Each area can have any size from 0 to 32 Mbits in 4 Mbit steps
- Single 1.8V Supply
- Serial Peripheral Interface (SPI) Compatible
  - Supports SPI Modes 0 and 3 (1-1-1)
  - Supports QPI Mode (4-4-4)
  - Supports Octal Mode (8-8-8)
  - Supports Dual Transfer Rate (DTR) for QPI and Octal modes
- 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 QPI and Octal mode)
  - 1.0 mA + 30 µA/MHz Active Read Current (KGD, Typical, for SPI Mode@ 1pF load)
  - 1.0 mA + 65 µA/MHz Active Read Current (KGD, Typical, for QPI Single Transfer Rate @ 1pF load)
  - 1.0 mA + 91 µA/MHz Active Read Current (KGD, Typical, for QPI Dual Transfer Rate @ 1pF load)
  - 1.0 mA + 142 µA/MHz Active Read Current (KGD, Typical, for Octal Mode Dual Transfer Rate @ 1pF load)
- 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 a unique identifier
  - 128 bytes user programmable
- Flexible Programming
  - Byte/Page Program (1 to 256 Bytes)
  - Single, Quad 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 Hardware Reset
- JEDEC Standard 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
  - WLCSP
  - Known Good Die [KGD]

1. Description

The Adesto® ATXP064 and ATXP064R are high speed serial interface Flash memory devices designed for use in a wide variety of high-volume consumer based applications in which program code is executed directly from Flash memory (XiP) or shadowed from Flash memory into embedded or external RAM for execution. The ATXP064R 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.

Except for the Read-While-Write capability, the ATXP064 and ATXP064R are identical. Except when there is a specific difference between the two, the term “ATXP064” is used to cover both in this datasheet.

The ATXP064 is specifically optimized for eXecute-in-Place (XiP) operations. While being backwards compatible with existing XiP protocols, the ATXP064 includes additional improvements that reduce significantly the latency of fetching the next cache line(s). The improved command protocol may enable more than 40% faster execution than the standard XiP protocol running at the same clock frequency.

In addition to standard SPI (1-1-1) Operation, the ATXP064 supports QPI Mode (4-4-4) and Octal Mode (8-8-8).

For even higher data throughput, the ATXP064 supports Dual Transfer Rate (DTR) for QPI and Octal modes.

For faster transfer of data from the device, the ATXP064 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 DTR and STR modes than can be achieved by using SCK as the clock signal for incoming data.

The devices are optimized for low power system operation. In addition to the inherently low power consumption of the devices, they support programmable strength IO drivers that can be matched to the required operating capacitive load. The ATXP064 / ATXP064R 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 ATXP064 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.

Specifically designed for use in many different systems, the ATXP064 supports read, program, and erase operations with a single 1.8V supply voltage. No separate voltage is required for programming and erasing.

2. Pin Descriptions and Pinouts

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

Table 2-1. Pin Descriptions

Symbol	Name and Function	Asserted State	Type
$\overline{CS}$	**CHIP SELECT:** Asserting the $\overline{CS}$ pin selects the device. When the $\overline{CS}$ pin is deasserted, the device will be deselected and normally be placed in standby mode (not Deep Power-Down mode), and the output pins will be in a high-impedance state. When the device is deselected, data will not be accepted on the SI pin. A high-to-low transition on the $\overline{CS}$ pin is required to start an operation, and a low-to-high transition is required to end an operation. When ending an internally self-timed operation such as a program or erase cycle, the device will not enter the standby mode until the completion of the operation.	Low	Input
SCK	**SERIAL CLOCK:** This pin is used to provide a clock to the device and is used to control the flow of data to and from the device. In Single Transfer Rate 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 Transfer Rate 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 Single Transfer Rate modes, command, address, and input data present on the SI pin is always latched in on the rising edge of SCK. In the Double Transfer Rate Modes, address and input data present on the SI pin is latched on both edges of SCK. In QPI and Octal modes, the SI Pin becomes an I/O pin (I/O_0) in conjunction with other pins. In Single Transfer Rate modes this allows four or eight bits of command, address, or input data on I/O_{3-0} or I/O_{7-0} to be clocked in on the rising edge of SCK, or four or eight bits clocked out on the falling edge of SCK. In Double Transfer Rate modes this allows four or eight bits of address or input data on I/O_{3-0} or I/O_{7-0} to be clocked in on every edge of SCK, or four 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 Double Transfer Rate modes. To maintain consistency with the SPI nomenclature, the SI (I/O_0) pin will be referenced as the SI pin unless specifically addressing the Multi-I/O modes in which case it will be referenced as I/O_0. Data present on the SI pin will be ignored whenever the device is deselected ($\overline{CS}$ is deasserted).	-	Input/ Output

Table 2-1. Pin Descriptions (Continued)

Symbol	Name and Function	Asserted State	Type
SO (I/O_1)	**SERIAL OUTPUT:** The SO pin is used to shift data out from the device. In the Single Transfer Rate modes, Data on the SO pin is always clocked out on the falling edge of SCK. In the Double Data Rate modes, Data on the SO pin is clocked out on both edges of SCK In QPI and Octal modes, the SO Pin becomes an I/O pin (I/O_1) in conjunction with other pins. In Single Transfer Rate modes this allows four or eight bits of command, address, or input data on I/O_{3-0} or I/O_{7-0} to be clocked in on the rising edge of SCK, or four or eight bits clocked out on the falling edge of SCK. In Double Transfer Rate modes this allows four or eight bits of address or input data on I/O_{3-0} or I/O_{7-0} to be clocked in on every edge of SCK, or four 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 Double Transfer Rate modes. To maintain consistency with the SPI nomenclature, the SO (I/O_1) pin will be referenced as the SO pin unless specifically addressing the Multi-I/O modes in which case it is referenced as I/O_1. The SO pin will be in a high-impedance state whenever the device is deselected ($\overline{CS}$ is deasserted).	-	Input/ Output
$\overline{WP}$ (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 48 for more details on protection features and the $\overline{WP}$ pin. In QPI and Octal modes, I/O_2 is used together with I/O_{3-0} or I/O_{7-0} as a bidirectional I/O pin. pin. In these modes, the I/O_2 pin will be in a high-impedance state whenever the device is deselected ($\overline{CS}$ is deasserted). The WP pin is internally set to a logic 1 state at power up. The 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 WP pin. If the ATXP064 is installed in a system which supports only Standard SPI Mode, and write protection control is not needed, the WP pin can be left floating. In this case the value of the WPP bit is undefined. Once the WP pin is used in Standard SPI Mode, it should be always driven. If the ATXP064 is installed in a system which also supports QPI and/or Octal modes, the WP pin functions as write protection control only in Standard SPI Mode. In this case, if write protection control is required, the WP pin must be driven externally when the device is switched from QPI/OPI to Standard SPI Mode. In QPI and Octal modes the WPP bit value is undefined and write protection control logic behaves as if the WP pin is connected to a logical 1 state.	Low	Input/ Output
I/O_3	**I/O_3 :** In QPI and Octal modes, I/O_3 is used together with I/O_{2-0} or I/O_{7-0} as a bidirectional I/O pin. In these modes, the I/O_3 pin will be in a 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

Table 2-1. Pin Descriptions (Continued)

Symbol	Name and Function	Asserted State	Type
DS	**DATA STROBE:** DS is the return of the SCK clock, synchronized to the return data. It is available in all modes, and will make 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 CS is driven low by the host controller, and will be driven until 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 Transfer Rate mode, DS is driven high in the first half of the data output cycle, and low in the second half. In this mode, DS will basically be 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 STR 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 Dual Transfer Rate mode, DS changes value at the edge of each data bit. In this mode, DS will basically be the same value as SCK, but with a delay. Achieving high clock rates in systems without DS will require 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 these modes, the I/O_{7-4} pins (as well as the I/O_{3-0} pins) will be in a high-impedance state whenever the device is deselected ($\overline{CS}$ is deasserted). In other modes, the I/O_{7-4} pins are always in a high-impedance state. These I/O lines are not available in 8-pin packages.	-	Input/ Output
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
$\overline{RESET}$	**RESET:** A low state on the reset pin ($\overline{RESET}$) will terminate the operation in progress and reset the internal state machine to an idle state. The device will remain in the reset condition as long as a low level is present on the $\overline{RESET}$ pin. Normal operation can resume once the $\overline{RESET}$ pin is brought back to a high level. See Section 12.11 for details about the device operation when $\overline{RESET}$ pin is engaged. The device incorporates an internal power-on reset circuit, so there are no restrictions on the $\overline{RESET}$ pin during power-on sequences. If this pin and feature are not utilized, then it is recommended that the $\overline{RESET}$ pin is driven high externally. It has an internal pull-up, and may alternatively be left open. The $\overline{RESET}$ pin is not required for operation of the device. The JEDEC Standard Hardware Reset function described in Section 12.10 provides the same functions without requiring a dedicated pin. The $\overline{RESET}$ pin is included for compatibility with older systems. For new designs, the JEDEC Standard Hardware Reset is recommended. The $\overline{RESET}$ pin may not be included in all package options.	Low	Input

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

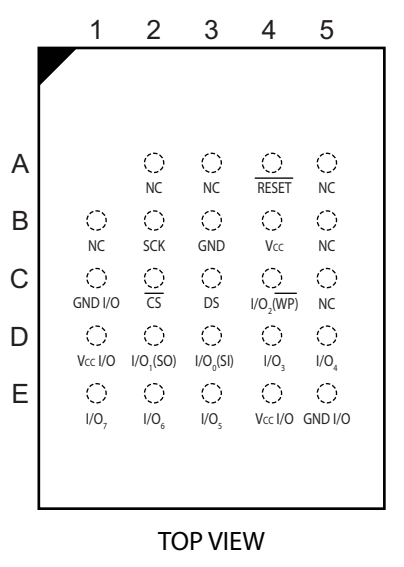


Figure 2-2. 46-ball WLCSP Pinout

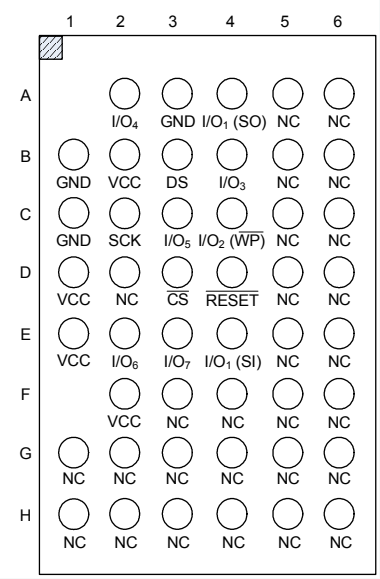
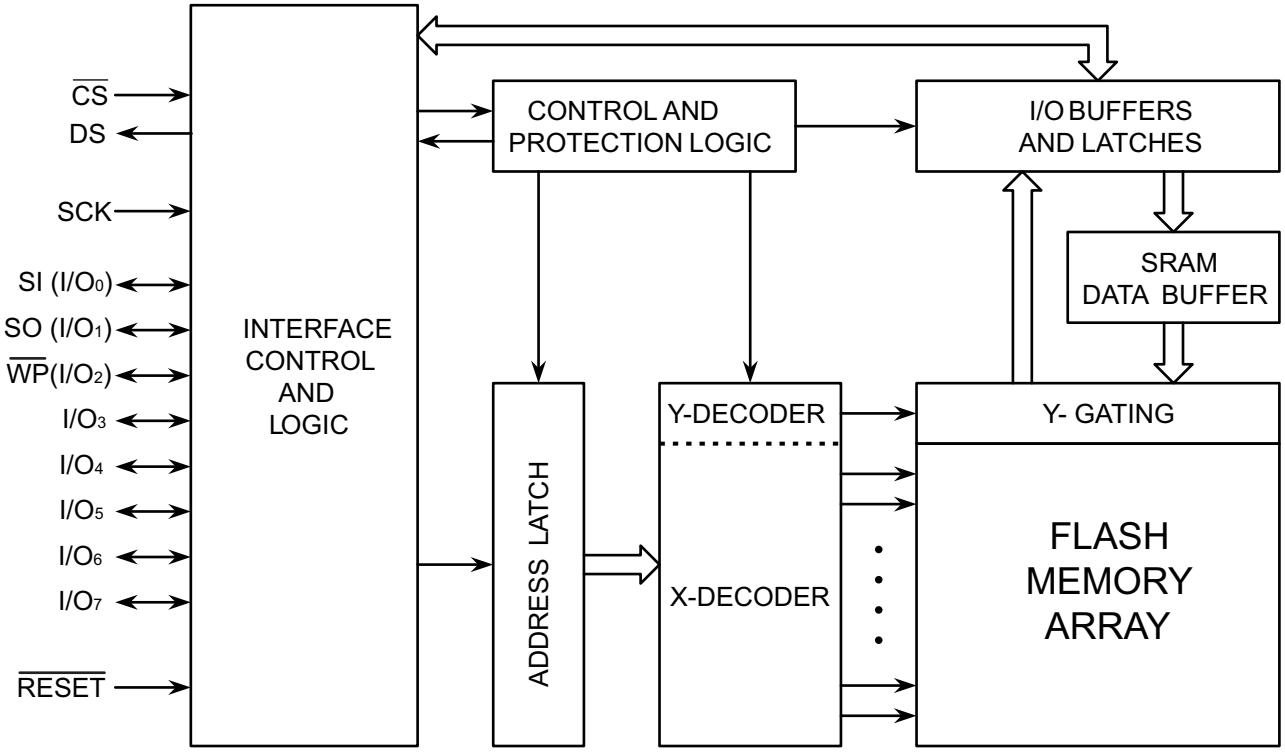


Figure 2-3. QFN Pinout

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 ATXP064 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.

4.1 Read-While-Write Memory Banks (ATXP064R *only*)

For Read-While-Write operations, the memory array is divided into 8 memory banks of 16 Mbits each as shown in Figure 4-2, Read-While-Write Memory Banks (ATXP064R only). 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 (ATXP064R only), for more details about using Read-While-Write operations.

Figure 4-1. Memory Architecture Diagram

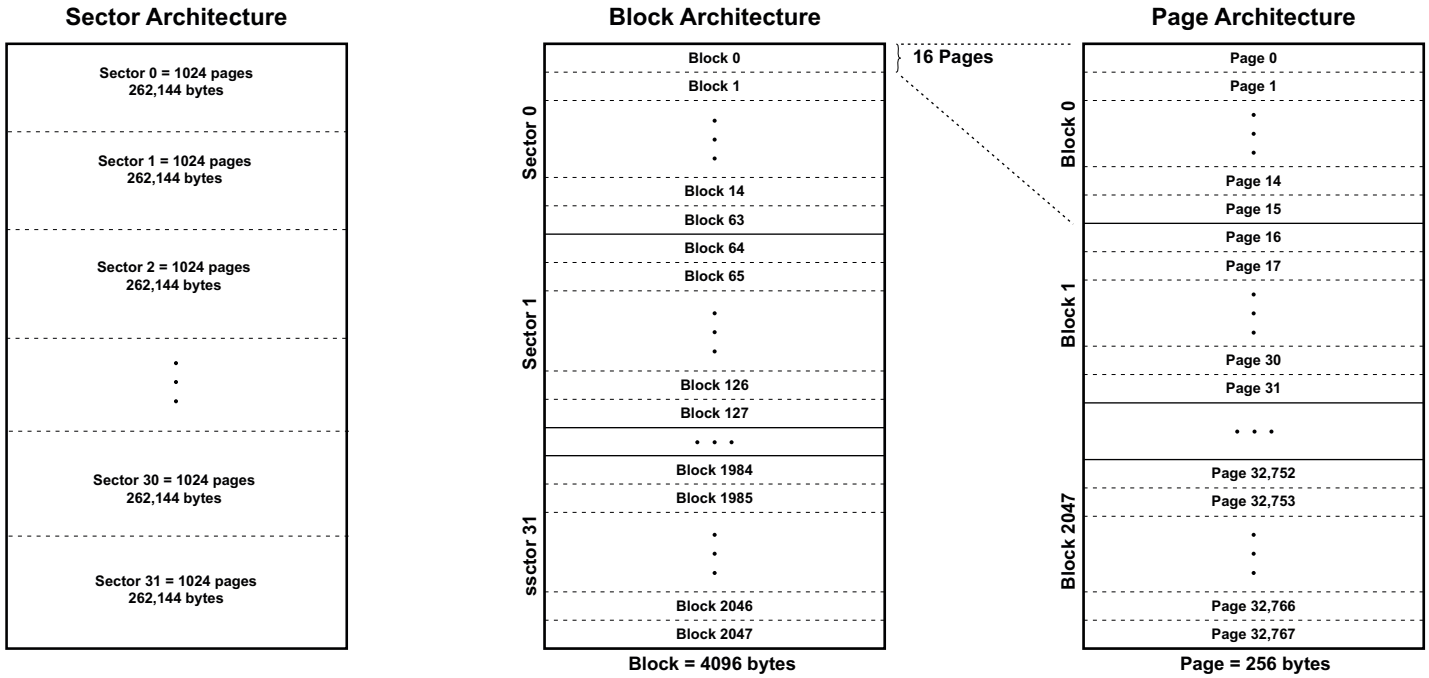
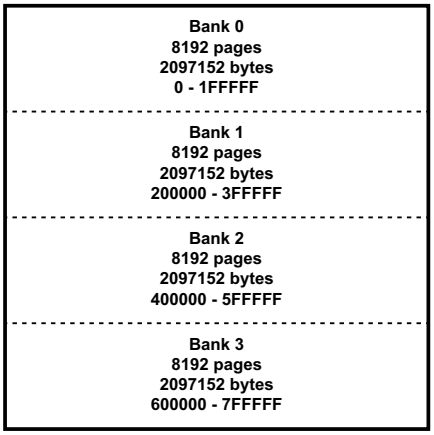


Figure 4-2. Read-While-Write Memory Banks (ATXP064R only)



5. Device Operation

5.1 Standard SPI Mode

The ATXP064 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 ATXP064 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 ATXP064 supports the two most common modes, SPI Modes 0 and 3 for the standard SPI (1-1-1). For QPI and Octal modes, 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.

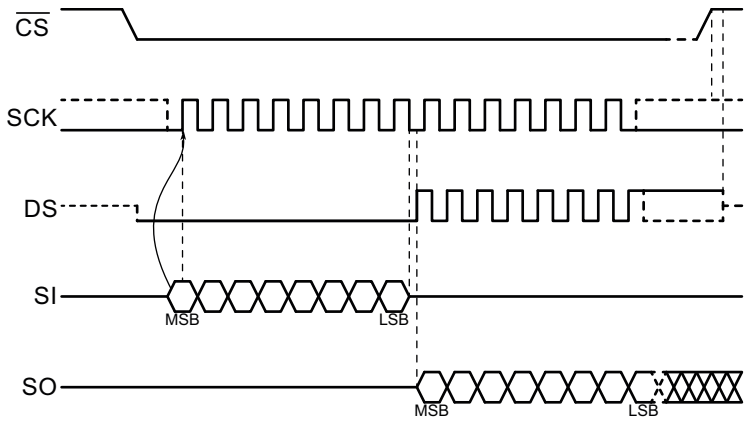
The default SPI Mode on power up is the traditional (1-1-1) SPI Mode.

Note that the device will always wake up in (1-1-1) SPI Mode when coming out of Ultra-Deep Power-Down.

The Enter QPI Mode (38h) and Enter Octal Mode (E8h) commands, or directly writing the corresponding bits in Status Register Byte 2, are used to change to a different mode. The Return to Standard SPI Mode (FFh) instruction, or directly writing the corresponding bits in Status 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 QPI or Octal modes. See Section 13.4, AC Characteristics - Maximum Clock Frequencies for details.

Figure 5-1. SPI Mode 0 and 3



5.2 QPI Mode

The ATXP064 features a QPI Mode to further improve throughput. QPI mode allows the command byte to be clocked in only 1 or 2 clock cycles, and four address bits to be clocked into the device or four bits of data to be clocked into or out of the device every full or half clock cycle. Between the address bits and data bits, 4 (or more) dummy cycles are required as shown in Figures 5-2 and 5-7. 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 QPI Mode, only SPI Mode 0 is supported.

The ATXP064 supports both Single Transfer Mode and Dual Transfer Mode for QPI Mode. See Section 5.4 for details about Dual Transfer Mode.

The QPI Mode is a (4-4-4) mode, using I/O_{3-0} 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, QPI Mode and Octal Mode are exclusive. Only one mode can be active at any given time.

Upon power-up, the default state of the device is Standard SPI Mode. To enable QPI Mode, the Quad Enable bit (QPIE) in Status Register Byte 2 is required to be set to 1. Enter QPI Mode (38h) and Return to Standard SPI Mode (FFh) instructions are used to switch between these two modes. The Write Status / Control Register commands may also be used to change modes.

The device will remain 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, the device will switch from QPI Mode to Standard SPI Mode after the Program/Erase Operation has been received. (t_{AUDPD} after $\overline{CS}$ goes high.)

Note: If QPI 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.

Figure 5-2. Single Transfer Rate (STR) operation in QPI Mode - Read operations

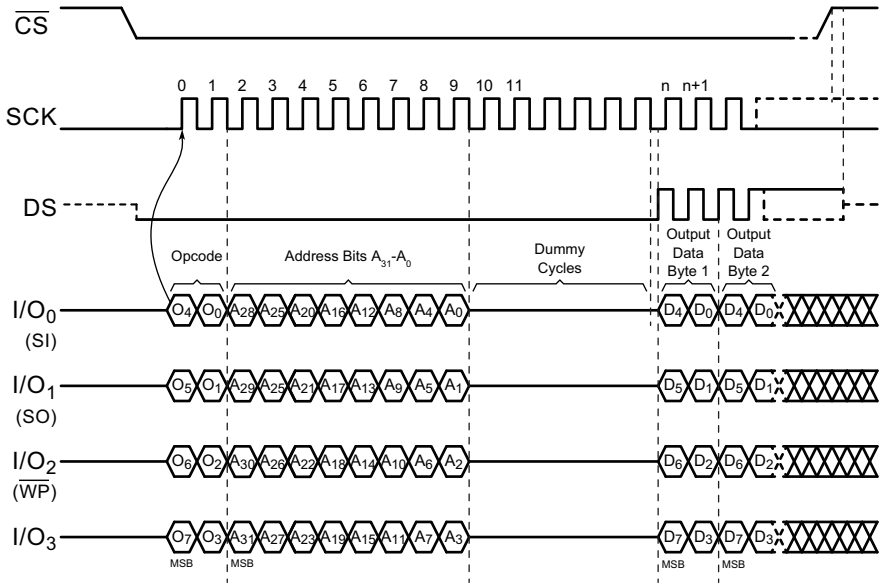
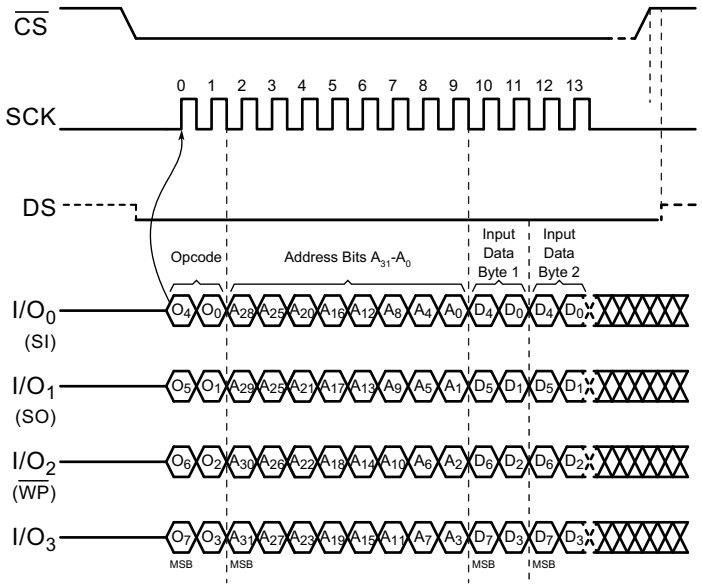


Figure 5-3. Single Transfer Rate (STR) operation in QPI Mode - Write operations



5.3 Octal Mode

The ATXP064 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 5-5 and 5-9. An additional half dummy cycle is required for Octal DTR as described in Figure 5-9. 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 ATXP064 supports both Single Transfer Mode and Dual Transfer Mode for Octal Mode. See Section 5.4 for details about Dual Transfer 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, QPI 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) instructions 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 bit (OME) in Status 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 will remain 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, the device will switch 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.

5.3.1 Octal Interface communication lines:

The Octal Interface has eleven communication lines, as shown in Figure I:

- CHIP SELECT ($\overline{CS}$): $\overline{CS}$ is 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 will be 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 will not be 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 will not enter the standby mode until the completion of the operation.
- SCK: Serial Clock is a Host to Device signal. SCK operates in push-pull mode.
- Data Strobe (DS): Data Strobe is a Device to Host signal and is output only. Data Strobe operates in push-pull mode. DS is the return of the clock signal.
  In Single Transfer Rate mode, DS is driven high in the first half of the data output cycle, and low in the second half. In this mode, DS will basically be 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 STR 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 Dual Transfer Rate mode, DS changes value at the edge of each data bit. In this mode, DS will 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 will remain low until $\overline{CS}$ is driven high again. DS is primarily required for fast operation in DTR 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 will be in a high-impedance (high-Z) state whenever the device is deselected ($\overline{CS}$ is deasserted).

Figure 5-4. Octal Interface Bus circuitry diagram

5.3.2 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 5-5. Single Transfer Rate (STR) operation in Octal Mode - Read operations

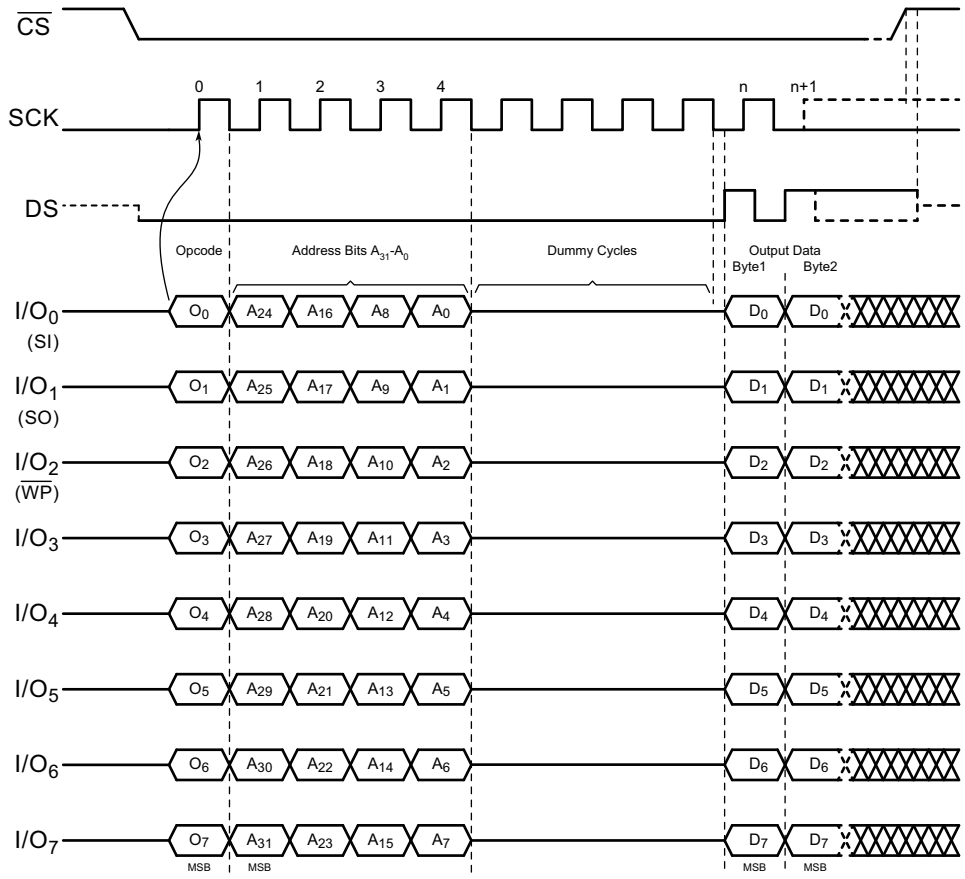
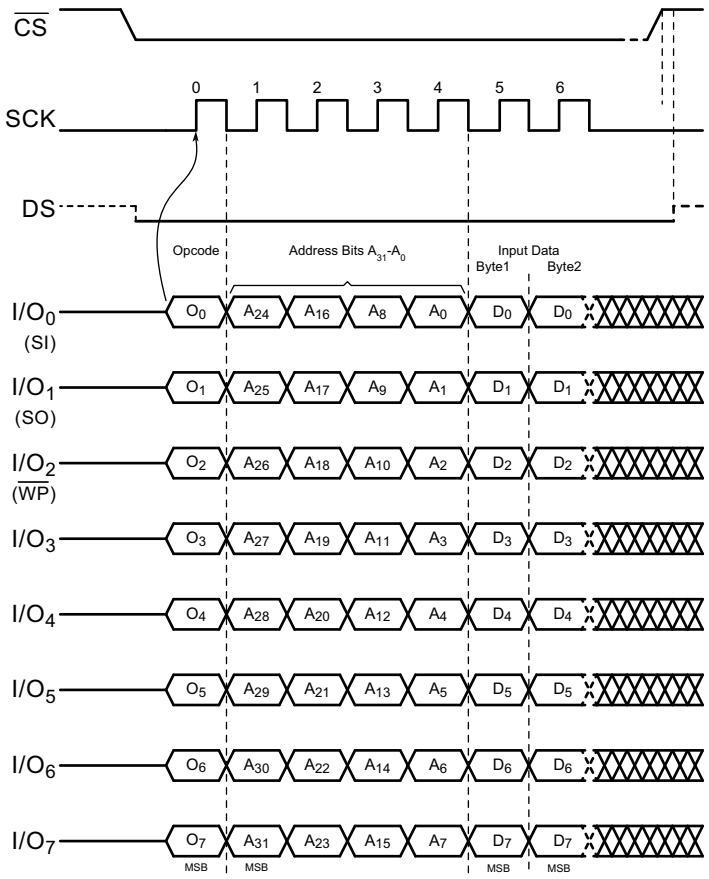


Figure 5-6. Single Transfer Rate (STR) operation in Octal Mode - Write operations



5.4 Dual Transfer Rate Operation

The ATXP064 allows Dual Transfer Rate (DTR) operation in QPI and Octal Modes for additional increase of throughput speed. Dual Transfer Rate Operation is enabled by setting the $\overline{STR}$/DTR bit in Status Register Byte 2.

Operation using DTR mode is identical to using Single Transfer Rate (STR) 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 Double Transfer Rate modes. In addition, for DTR mode 4 (or more) dummy cycles are required as shown in Figures 5-7 and 5-9. An additional half dummy cycle is required for Octal DTR as described in Figure 5-9.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 Dual Transfer Rate (DTR) operation in Octal Mode, all read and write operations will operate on an even number of bytes. The minimum number of SCK pulses for clocking data is one, and as data will be clocked on both edges, two bytes will be 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 will be ignored and treated as a_0 = 0 by the device. For register operations that operate on a single 8-bit register, a read operation will output 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 will be 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 5-7. Dual Transfer Rate (DTR) operation in QPI Mode - Read operations

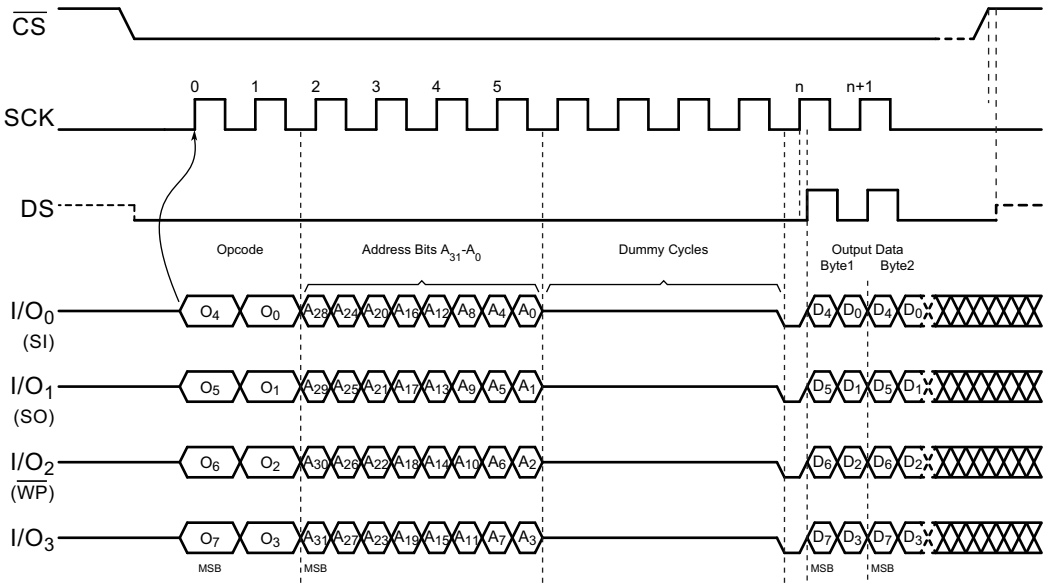


Figure 5-8. Dual Transfer Rate (DTR) operation in QPI Mode - Write operations

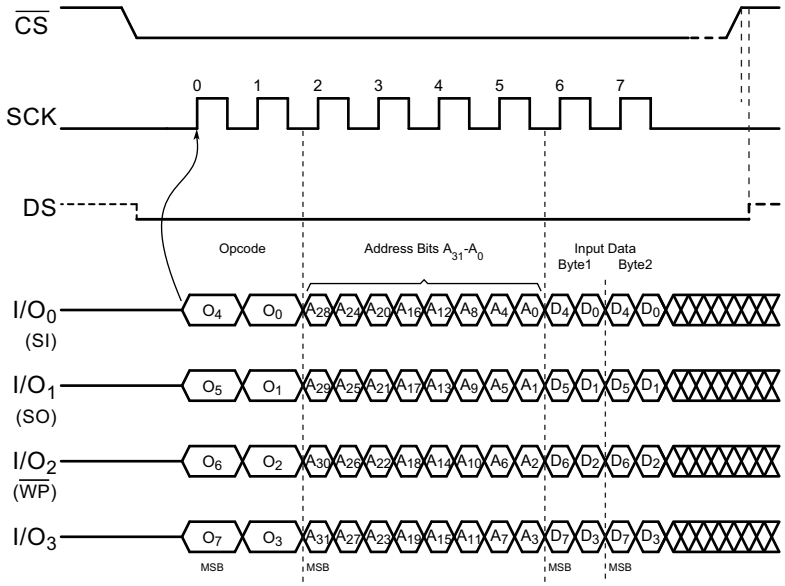


Figure 5-9. Dual Transfer Rate (DTR) operation in Octal Mode - Read operations

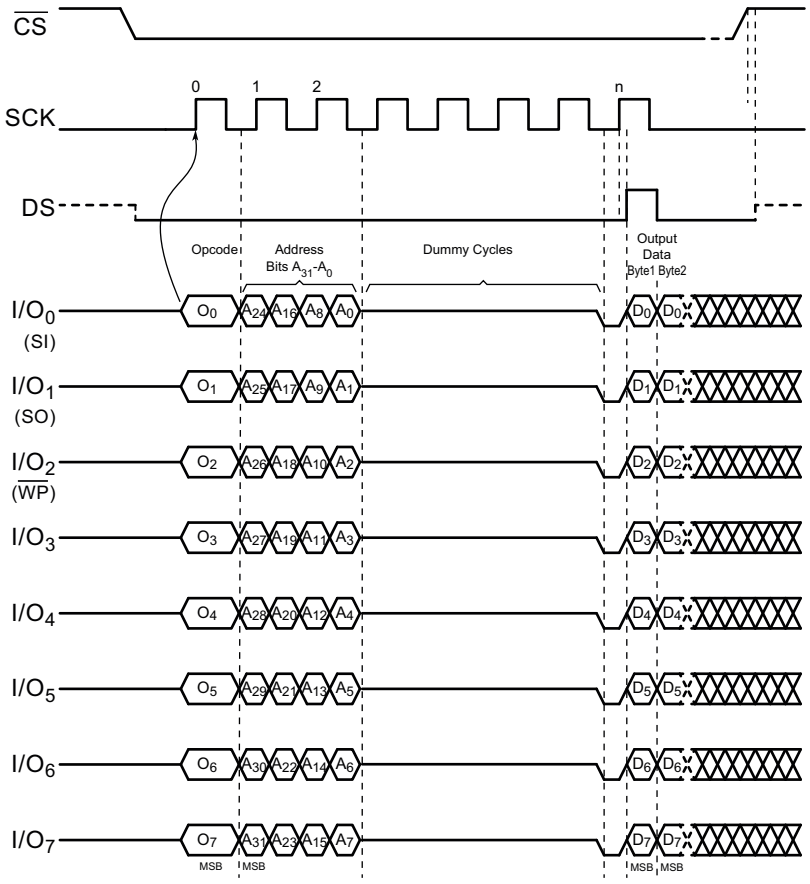
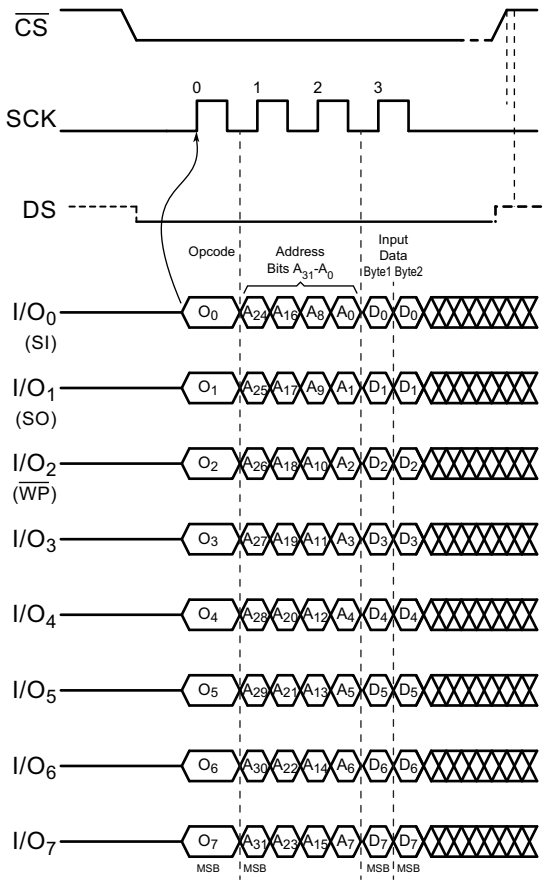


Figure 5-10. Dual Transfer Rate (DTR) operation in Octal Mode - Write operations



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 opcode on the SPI bus. Following the opcode, instruction dependent information such as address and data bytes would then be clocked out by the host controller. All opcode, address, and data bytes are transferred with the most-significant bit (MSB) first. An operation is ended by deasserting the $\overline{CS}$ pin.

Device behavior is not defined for opcodes not supported by the ATXP064. Opcodes not described in this document should therefore be avoided.

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

Addressing of the device requires a total of four bytes of information to be sent, representing address bits A31-A0. Since the upper address limit of the ATXP064 memory array is 7FFFFFh, address bits A31-A23 must be zero for proper operation.

Table 6-1. Command Listing 1 - Commands used in all modes.

Command	Opcode		Max Clock Frequency in SPI Mode	Max Clock Freq. in QPI and Octal Modes	Add-ress Bytes	Dummy Bytes in SPI Mode	Dummy Cycles in QPI and Octal Modes		Data Bytes	Section Link
							STR	DTR		
Read Commands										
Read Array	0Bh	0000 1011	66 MHz	133 MHz	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	QPI:4 OPI:3(2)	1+	11.5
Read Status Register Byte 1	05h	0000 0101	66 MHz	133 MHz	0	0	4	4	1+	11.6
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

Table 6-1. Command Listing 1 - Commands used in all modes.

Command	Opcode		Max Clock Frequency in SPI Mode	Max Clock Freq. in QPI and Octal Modes	Add-ress Bytes	Dummy Bytes in SPI Mode	Dummy Cycles in QPI and Octal Modes		Data Bytes	Section Link
							STR	DTR		
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 (2)	1+	12.18

1. See Table 11-8, Dummy Clock cycles
2. Octal DTR 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	Opcode		Max Clock Frequency	Address Bytes	Dummy Bytes in SPI Mode	Data Bytes	Section Link
Read Commands							
Read Array (1-1-1)	03h	0000 0011	50 MHz	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 MHz	4	1	1+	7.4
Miscellaneous Commands							
Read Manufacturer and Device ID	9Fh	1001 1111	66 MHz	0	0	12	12.1
Enter QPI Mode (1-1-1 to 4-4-4) (1)	38h	0011 1000	66 MHz	0	0	0	12.15
Enter Octal Mode (1-1-1 to 8-8-8) (2)	E8h	1110 1000	66 MHz	0	0	0	12.16

1. The command is entered in (1-1-1) mode, the device will change to (4-4-4) mode afterwards
2. 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 QPI (4-4-4) and Octal Mode (8-8-8).

Command	Opcode		Max Clock Frequency	Address Bytes	Dummy Cycles in QPI and Octal Modes	Data Bytes	Section Link
Read Commands							
Burst Read with Wrap	0Ch	0000 1100	133 MHz	4	8+ (1)	1+	7.2
Miscellaneous Commands							
Echo	AAh	1010 1010	133 MHz	1 Value Byte	4 (2)	1	12.13
Echo with inversion	A5h	1010 0101	133 MHz	1 Value Byte	4(2)	1	12.14
Return to Standard SPI Mode	FFh	1111 1111	133 MHz	0	0	0	12.17

1. See Table 11-8, Dummy Clock cycles
2. Octal DTR mode adds an additional ½ dummy cycle to align the first byte of data to the rising edge of SCK/DS See the Octal DTR figures included in the individual command description for details.

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 opcodes (0Bh 13h and 03h) can be used for the Read Array command. The use of each opcode depends on the maximum clock frequency that will be used to read data from the device and the mode it operates in. The 0Bh opcode can be used for any mode at any clock frequency up to the maximum specified by f_{CLK}. The 13h and 03h opcodes 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 opcode is using 3 address bytes, while the 13h opcode 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 opcode, the $\overline{CS}$ pin must first be asserted and the opcode 03h clocked into the device. After the opcode 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 opcode, the $\overline{CS}$ pin must first be asserted and the opcode 13h clocked into the device. After the opcode 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 opcode, the $\overline{CS}$ pin must first be asserted and the opcode 0Bh clocked into the device. After the opcode 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. After the opcode, one or more additional dummy bytes need to be clocked into the device. For the SPI Mode, one dummy byte is used. For the QPI and Octal modes, 8 or more dummy cycles are used as shown in Table 11-8, Dummy Clock cycles. Note that the table refers to full clock cycles, also for DTR modes. Half cycles are not used for dummy cycles, the settings are a multiple of 2 full cycles.

Note that for Dual Transfer Rate (DTR) operation in Octal Mode, all read operations will operate on an even number of bytes. The minimum number of SCK pulses for clocking data is one, and as data will be clocked on both edges, two bytes will be 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 will be ignored and treated as a_0 = 0 by the device.

After the address bytes (and the dummy bytes if using opcode 0Bh) 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 (7FFFFFh) of the memory array has been read, the device will continue reading back at the beginning of the array (000000h). No delays will be incurred when wrapping around from the end of the array to the beginning of the array.

Deasserting the $\overline{CS}$ pin will terminate the read operation and put the I/O 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 Opcode

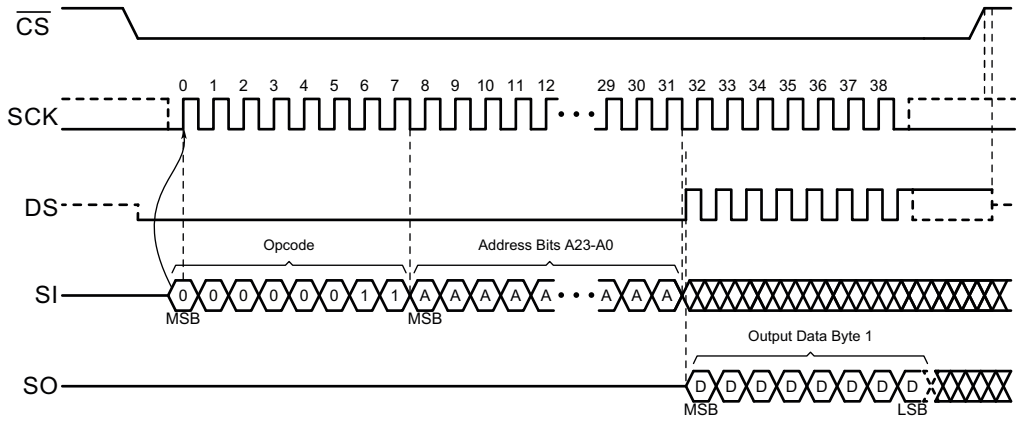


Figure 7-2. Read Array with 4-byte address - 13h Opcode

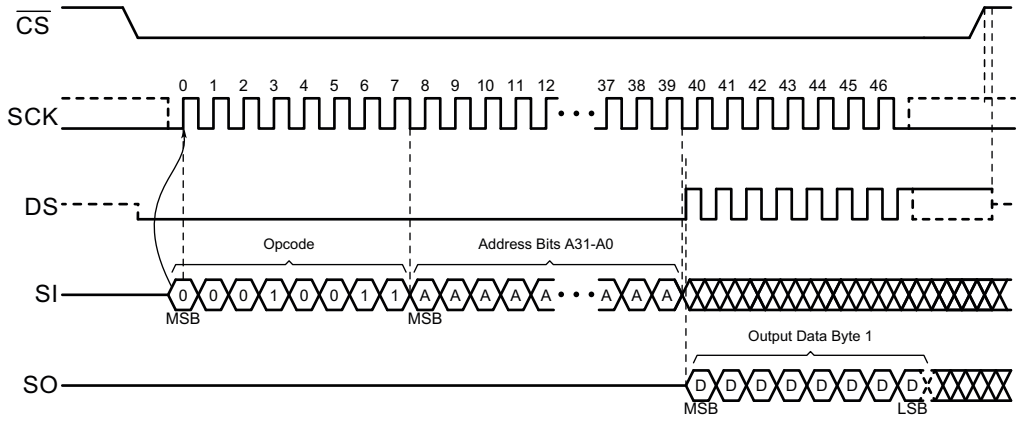


Figure 7-3. Read Array - 0Bh Opcode

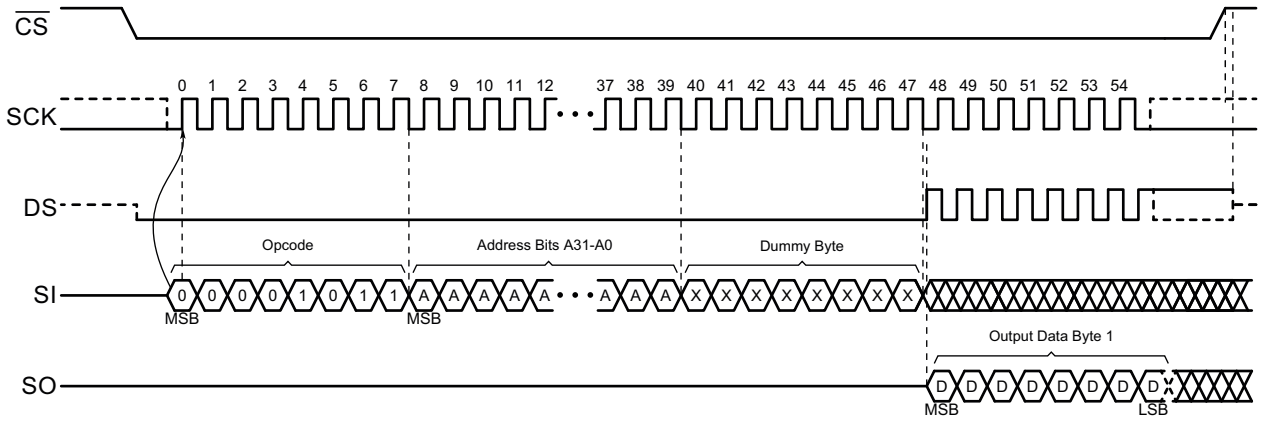


Figure 7-4. Read Array - 0Bh Opcode in QPI Mode STR

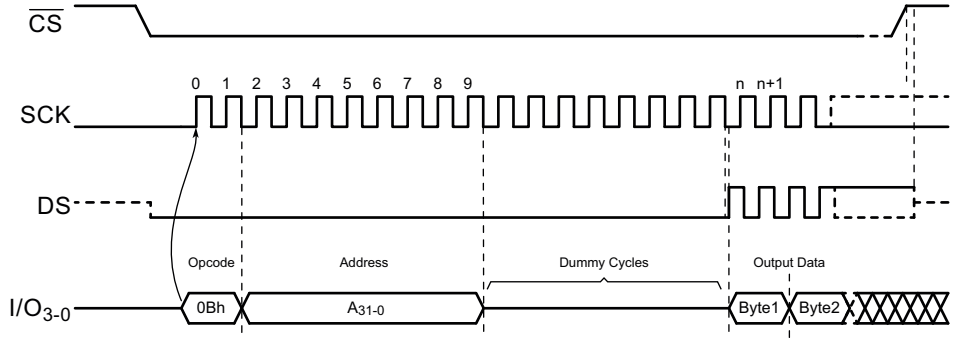


Figure 7-5. Read Array - 0Bh Opcode in Octal Mode STR

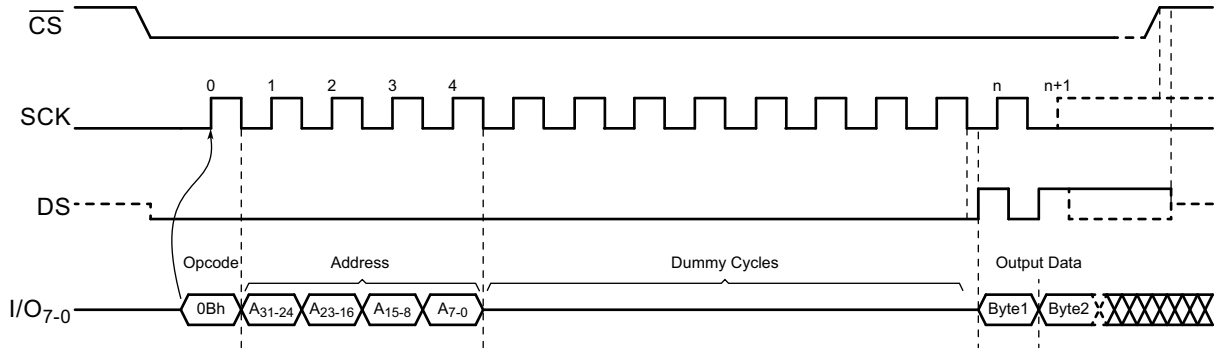


Figure 7-6. Read Array - 0Bh Opcode in QPI Mode DTR

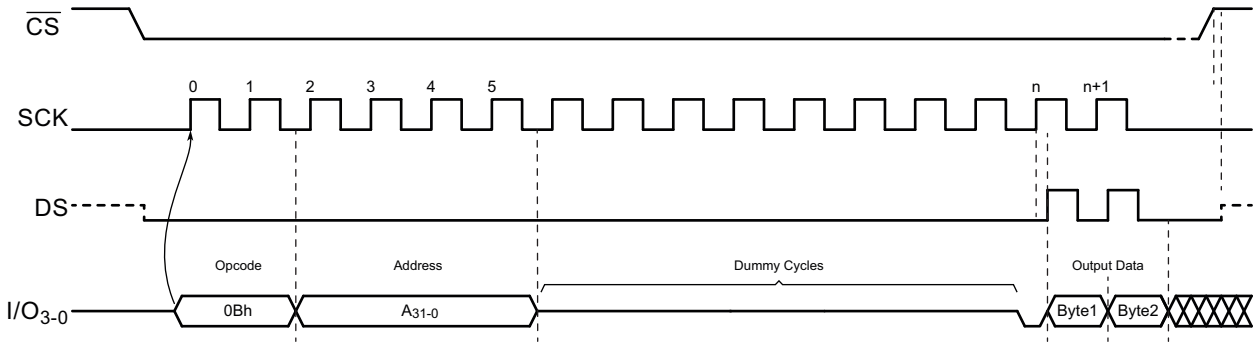
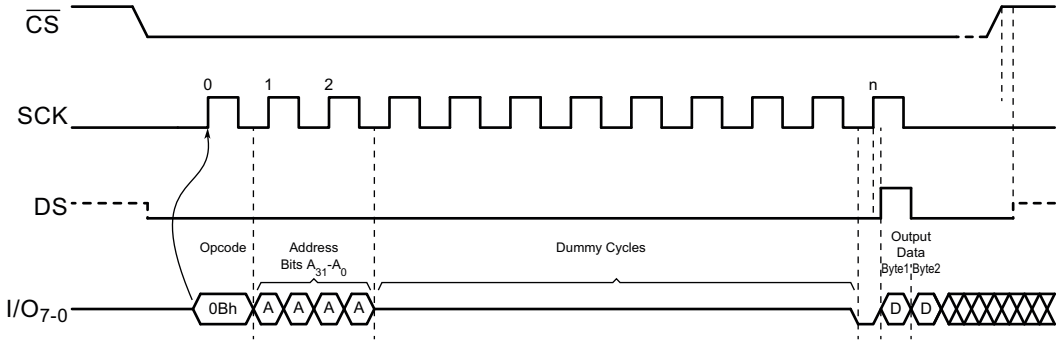


Figure 7-7. Read Array - 0Bh Opcode in Octal Mode DTR



7.2 Burst Read with Wrap (0Ch)

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

MCUs with cache benefit from this feature for eXecute-in-Place 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 will improve code execution performance in the MCU system, as the MCU will first receive 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:

In QPI STR Mode, the number of cycles will be reduced from 56 to 32:

Command (byte):	2 clock cycles (eliminated for the next consecutive line)
Address (4 bytes):	8 clock cycles (eliminated for the next consecutive line)
Dummy cycles:	14 clock cycles (eliminated for the next consecutive line)
Data (16 Bytes):	32 clock cycles

In QPI DTR Mode, the number of cycles will be reduced from 35 to 16:

Command (byte):	1 clock cycles (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 DTR 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

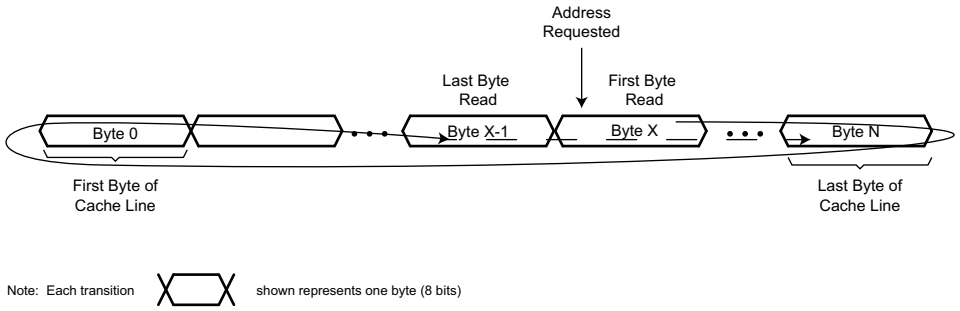
The behavior of the “Burst Read with Wrap (0Ch)” instruction is controlled by the W7-W5 bits in as described in Table 11-6, Status/Control Register Byte 3.

The Wrap Length and Wrap Mode set by W7-W5 in either SPI mode is still valid in any other SPI mode and can also be re-configured by the “Write Status/Control Registers (71h)” instruction. See Table 11-7 for details on Wrap Bit settings.

Once W7-5 are set, all the following “Burst Read with Wrap (0Ch)” instructions will use the W7-5 setting to access the 8/16/32/64-byte section within any page.

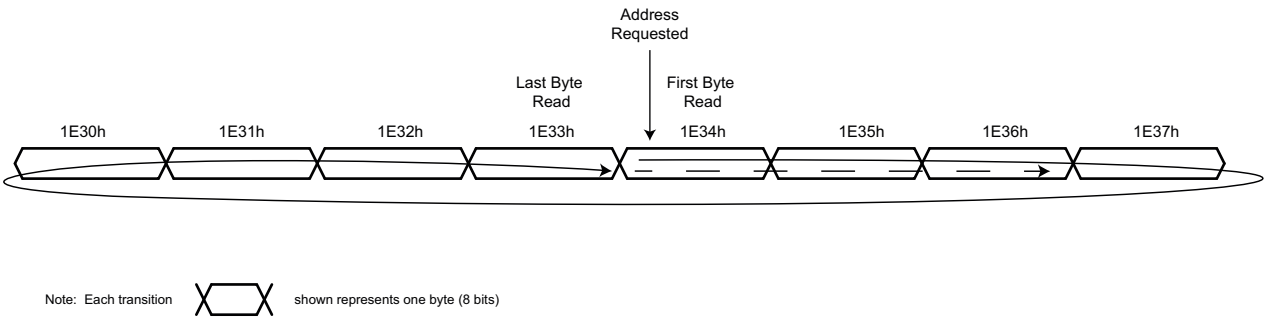
If W7 is cleared, the device will operate in Wrap Around Mode. The “Burst Read with Wrap (0Ch)” instruction will read to the end of the fixed length section (cache line), wrap around to the beginning of the section and continue reading the same cache line continuously for as long as additional clock 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 7-8. This mode of operation is included for compatibility with other XiP devices, and for MCUs that do not support the Continuous Mode.

Figure 7-8. Burst Read with Wrap, Wrap Around Mode (W7=0)



An example of this operation in Wrap Around Mode is shown in Figure 7-9. Here the Wrap Length is set to 8 for simplicity. The requested address is 1E34h. The device will therefore read 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 will be 1E30h, followed by 1E31h,1E32h and 1E33h. If the read operation continues, the device will then output 1E34h again, and this loop will continue for as long as the device is clocked.

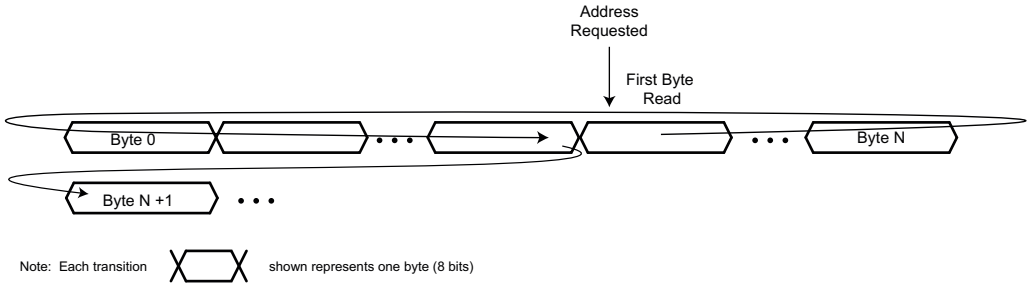
Figure 7-9. Burst Read with Wrap Example, Wrap Around Mode (W7=0), Wrap Length 8, Requested Address = 1E34h



If W7 is set, the device will operate 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 will read 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 will then start reading at the beginning of the *next* line as shown in Section 7-10. The device will read continuously for as long as SCK is clocked. There is no delay caused by

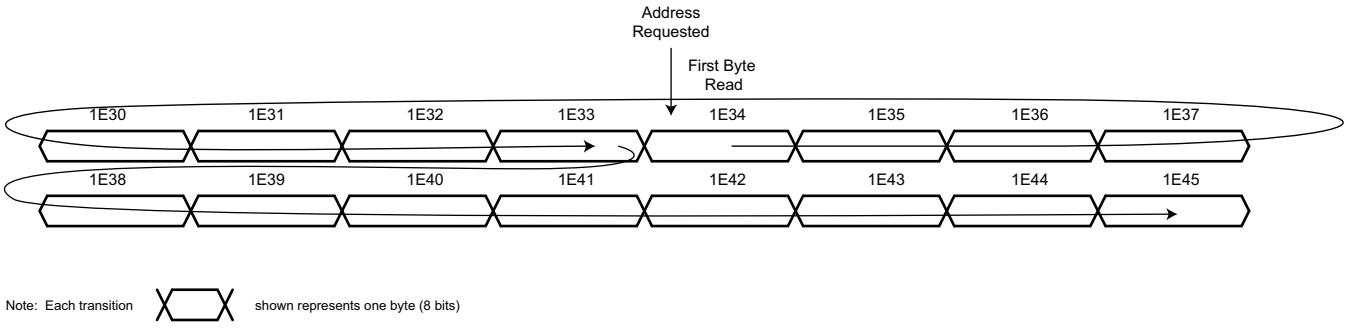
the wrap around, the first byte of the cache line follows immediately after the last. There is also no additional delay caused by the jump to the next cache line, the first byte of the following cache line follows immediately after the last byte read from the previous one, independent of where in the cache line the jump is coming from.

Figure 7-10. Burst Read with Wrap, Continuous Mode (W7=1)



An example of this operation in Continuous Mode is shown in Figure 7-11. As in the above example, the Wrap Length is set to 8, and the requested address is 1E34h. The device will read 1E34h first, followed by 1E35h, 1E36h, and 1E37h. It then wraps around, and the next address to be read out will be 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 will read continuously for as long as the device is clocked.

Figure 7-11. 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 as shown in Figure 7-16 and Figure 7-17. 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.

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

If the clock is stopped high in STR mode, the I/O lines will output the value of the last byte of the previous cache line. (The entire byte is output in Octal Mode, lower 4 bits in Quad and QPI modes.) If the clock is stopped low in STR mode, the I/O lines will output the value of the first byte of the next cache line. (The entire byte is output in Octal Mode, upper 4 bits in Quad and QPI modes.)

If the clock is stopped high in DTR mode, the I/O lines will output the value of the first byte of the next cache line. (The entire byte is output in Octal Mode, upper 4 bits in Quad and QPI modes.) If the clock is stopped low in DTR mode, the I/O lines will output the value of the last byte of the previous cache line. (The entire byte is output in Octal Mode, lower 4 bits in Quad and QPI modes.)

Figure 7-12. Burst Read with Wrap in QPI Mode - STR

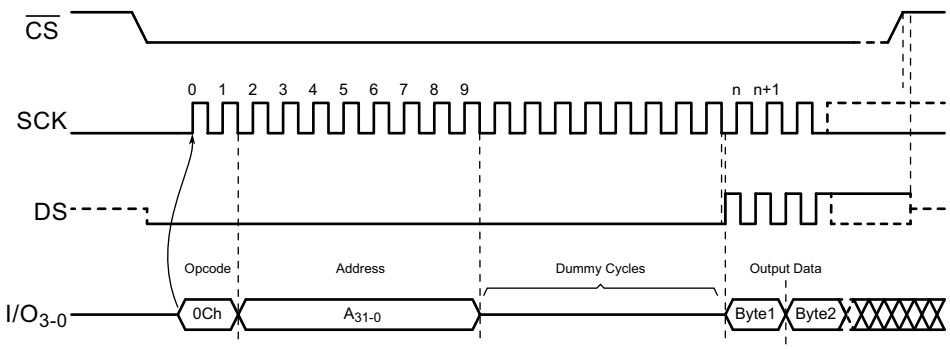


Figure 7-13. Burst Read with Wrap in Octal Mode - STR

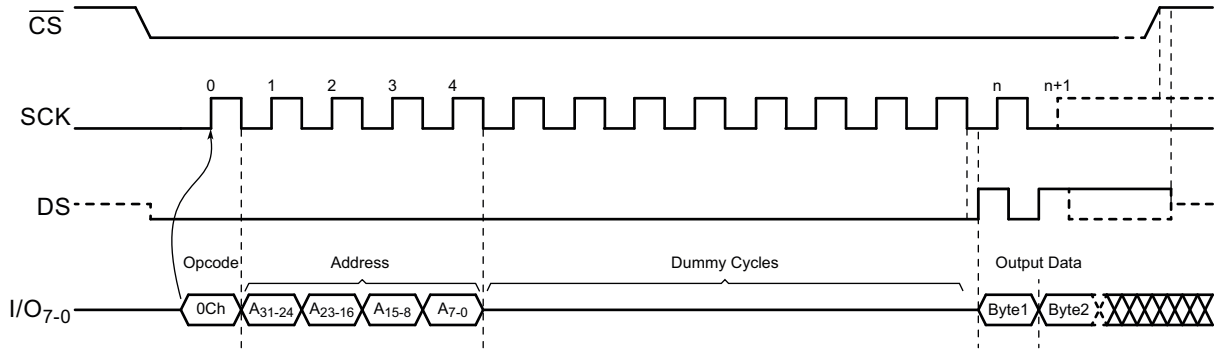


Figure 7-14. Burst Read with Wrap in QPI Mode - DTR

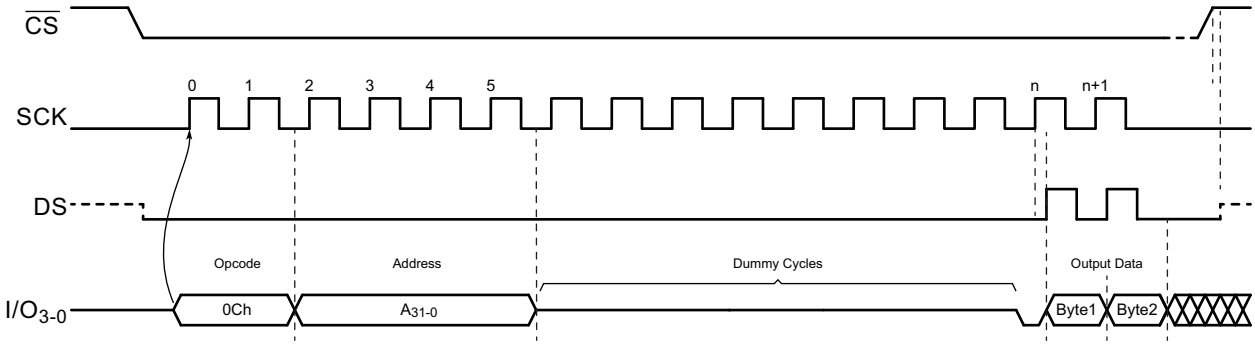


Figure 7-15. Burst Read with Wrap in Octal Mode - DTR

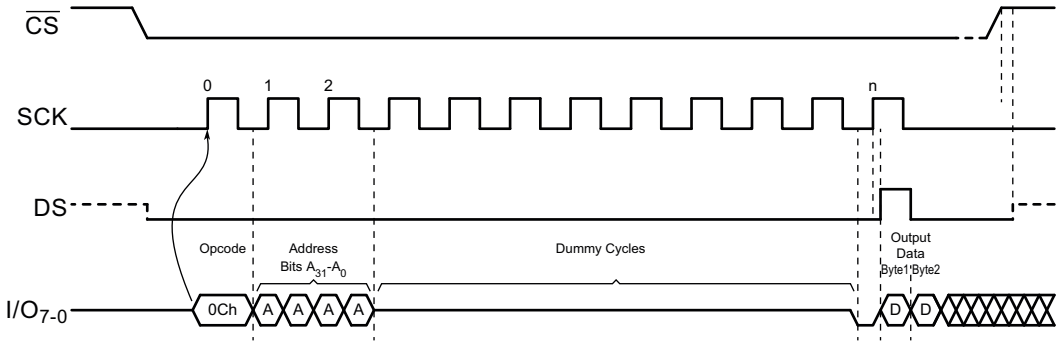
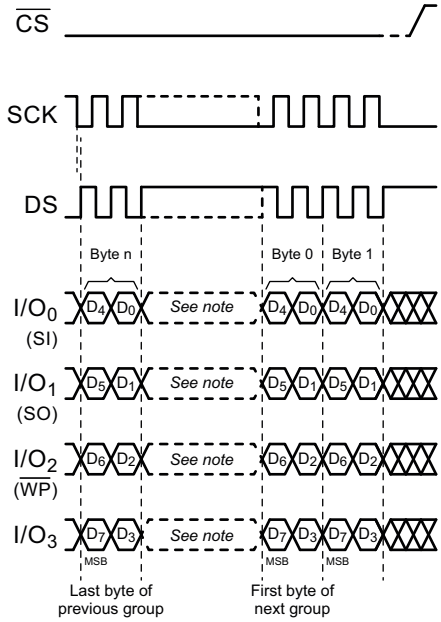
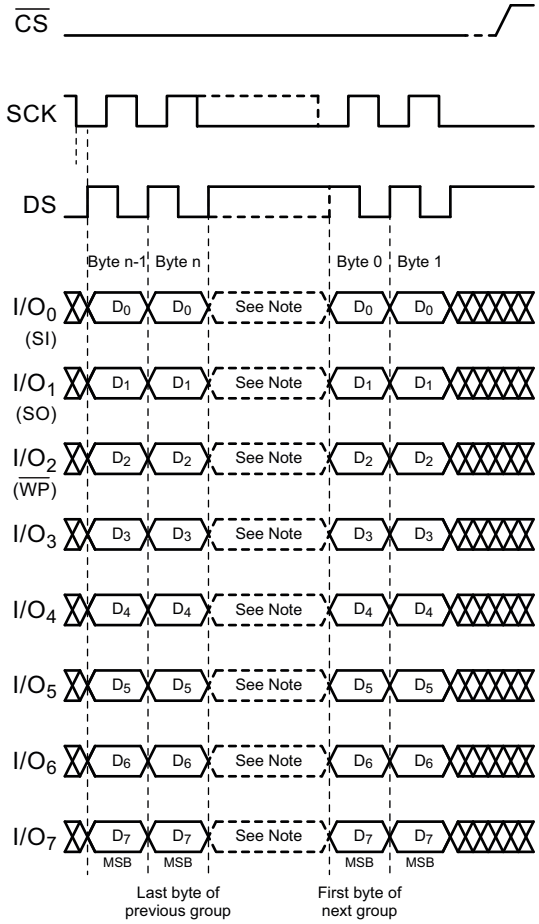


Figure 7-16. Timing diagram for two consecutive accesses in QPI Mode - STR).



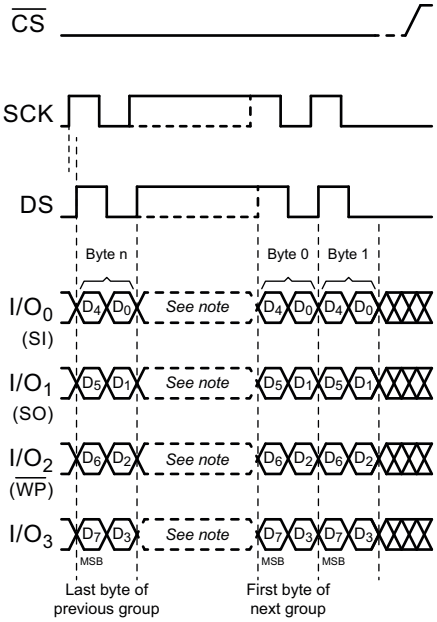
See note on page 24 regarding the value of the I/O lines while the clock is stopped.

Figure 7-17. Timing diagram for two consecutive accesses in Octal Mode - STR)



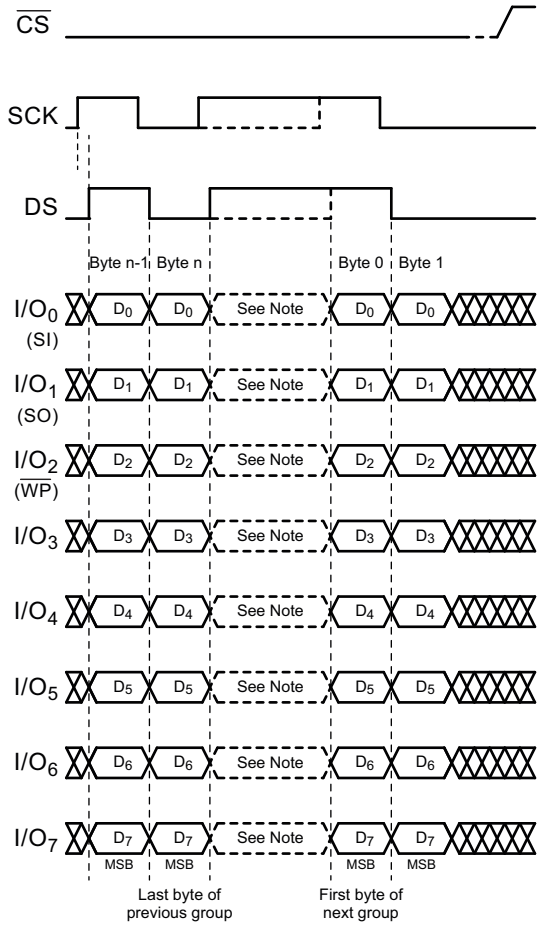
See note on page 24 regarding the value of the I/O lines while the clock is stopped.

Figure 7-18. Timing diagram for two consecutive accesses in QPI Mode - DTR



See note on page 24 regarding the value of the I/O lines while the clock is stopped.

Figure 7-19. Timing diagram for two consecutive accesses in Octal Mode - DTR



See note on page 24 regarding the value of the I/O lines while the clock is stopped.

7.3 Read-While-Write (ATXP064R only)

For Read-While-Write operations, the memory array is divided into 8 memory banks of 16 Mbits each as shown in Figure 4-2, Read-While-Write Memory Banks (ATXP064R only). 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 in ATXP064R.

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.

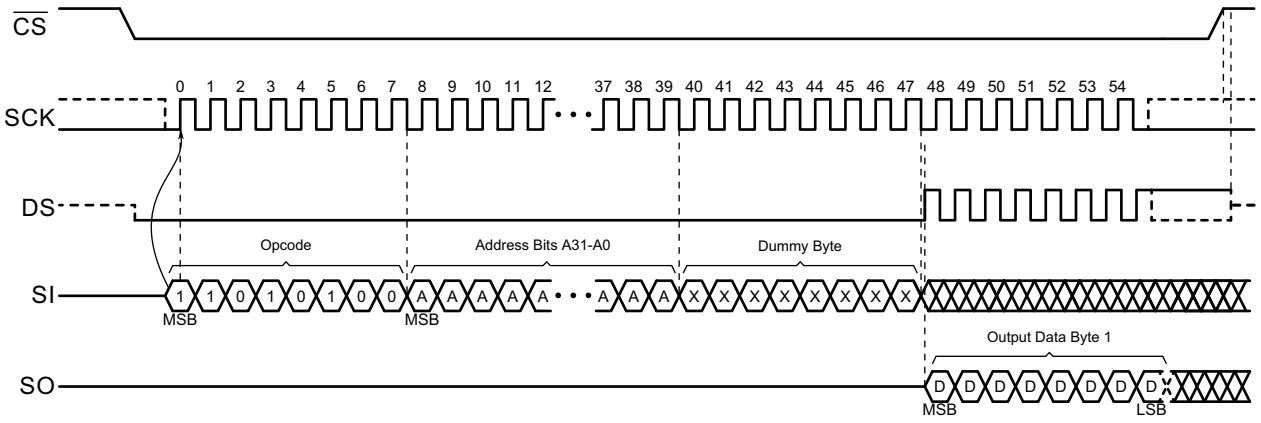
7.4 Buffer Read (D4h)

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 opcode D4h must be clocked into the device followed by four address bytes comprised of 24 dummy bits and 8 buffer address bits (BFA7 - BFA0). 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 opcode, the address bytes, the dummy byte and the reading of data. When the end of a buffer is reached, the device will continue reading back at the beginning of the buffer. A low-to-high transition on the $\overline{CS}$ pin will terminate the read operation and tri-state the output pin (SO).

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

Figure 7-20. 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 logical “1” state (a byte value of FFh). Before a Byte/Page Program command can be started, the Write Enable command must have been previously issued to the device (see “Write Enable (06h)” on page 48) to set the Write Enable Latch (WEL) bit of the Status Register to a logical “1” state.

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

If the starting memory address denoted by A31-A0 does not fall on an even 256-byte page boundary (A7-A0 are not all 0), then special circumstances regarding which memory locations to be programmed will apply. In this situation, any data that is sent to the device that goes beyond the end of the page will wrap around back to the beginning of the same page. For example, if the starting address denoted by A31-A0 is 0000FEh, and three bytes of data are sent to the device, then the first two bytes of data will be programmed at addresses 0000FEh and 0000FFh while the last byte of data will be programmed at address 000000h. The remaining bytes in the page (addresses 000001h through 0000FDh) will not be programmed and will remain in the erased state (FFh). In addition, if more than 256 bytes of data are sent to the device, then only the last 256 bytes sent will be latched into the internal buffer.

When the $\overline{CS}$ pin is deasserted, the device will take the data stored in the internal buffer and program it into the appropriate memory array locations based on the starting address specified by A31-A0 and the number of data bytes sent to the device. If less than 256 bytes of data were sent to the device, then the remaining bytes within the page will not be programmed and will remain in the erased state (FFh). The programming of the data bytes is internally self-timed and should take place in a time of t_{PP} or t_{BP} if only programming a single byte.

The 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 will abort the operation and no data will be programmed into the memory array. In addition, if the memory is in the protected state (see “Protect Sector” on page 19), then the Byte/Page Program command will not be executed, and the device will return to the idle state once the $\overline{CS}$ pin has been deasserted. The WEL bit in the Status Register will be reset back to the logical “0” state if the program cycle aborts due to an incomplete address being sent, an incomplete byte of data being sent, or because the memory location to be programmed is protected.

While the device is programming, the Status Register can be read and will indicate that the device is busy. For faster throughput, it is recommended that the Status Register be polled rather than waiting the t_{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 Register will be reset back to the logical “0” state.

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

Note that in Octal DTR mode, the Byte Program operation will program 2 bytes instead of one, and the Page Program operation will program an even number of bytes. Address bit a_0 has to be 0 in this case.

Figure 8-1. Byte Program

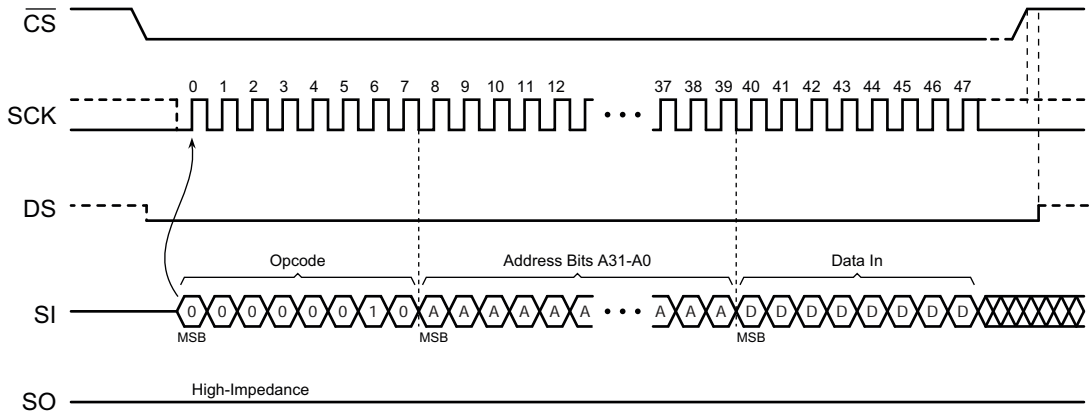


Figure 8-2. Byte Program in QPI Mode - STR

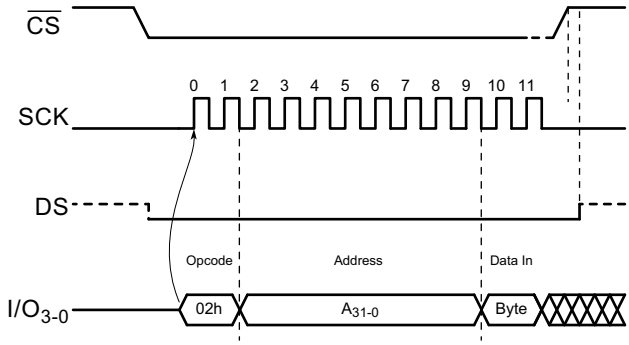


Figure 8-3. Byte Program in Octal Mode - STR

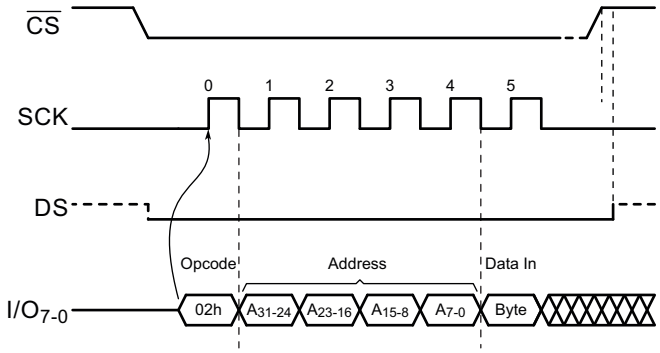


Figure 8-4. Byte Program in QPI Mode - DTR

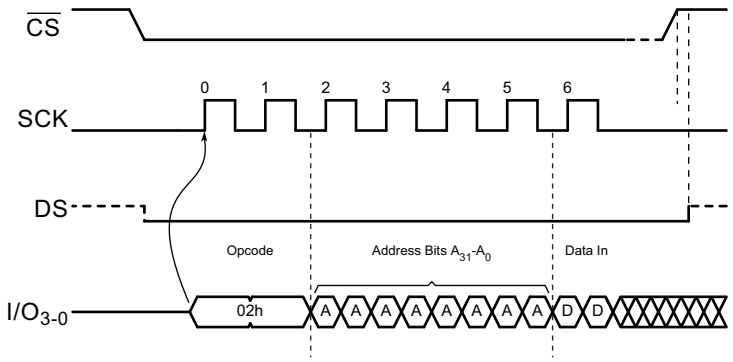
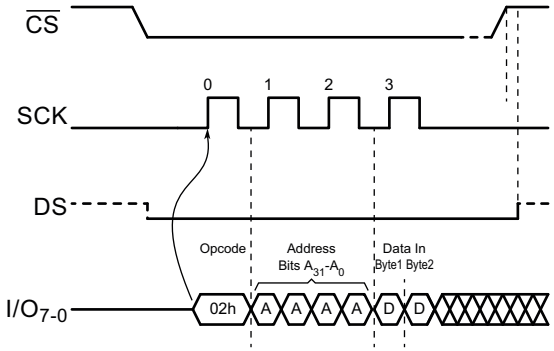


Figure 8-5. Byte Program in Octal Mode - DTR



Note that 2 bytes will be programmed when using Byte Program in Octal mode - DTR.

Figure 8-6. Page Program

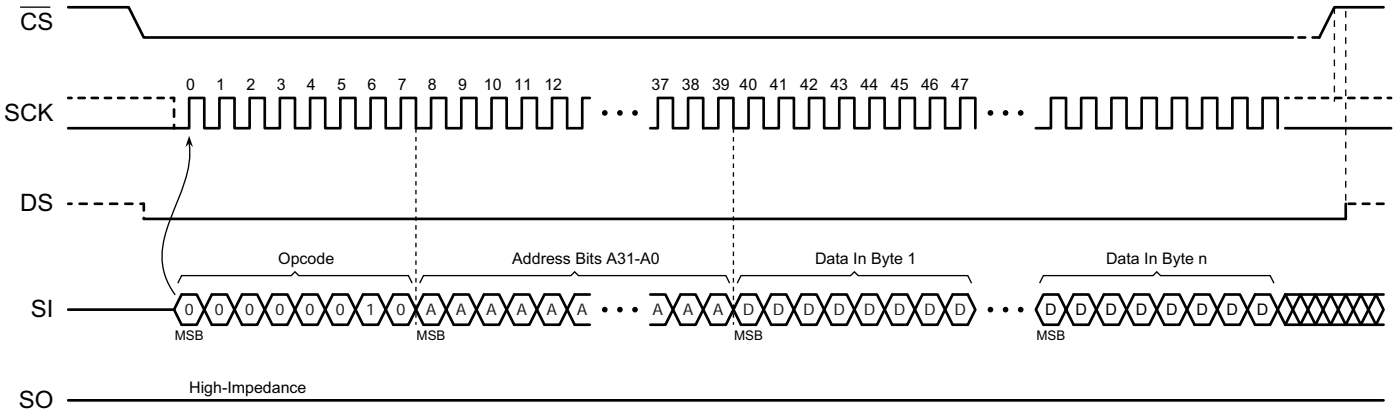


Figure 8-7. Page Program in QPI Mode - STR

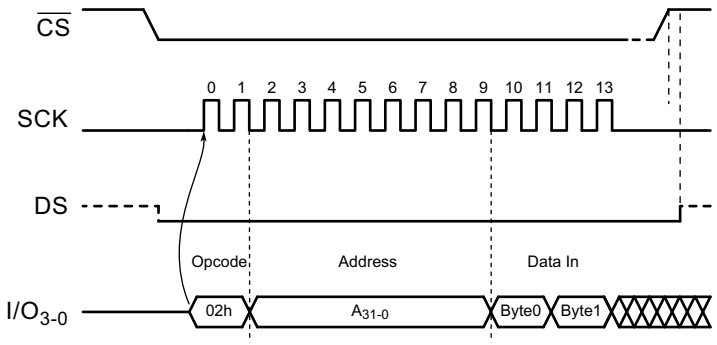


Figure 8-8. Page Program in Octal Mode - STR

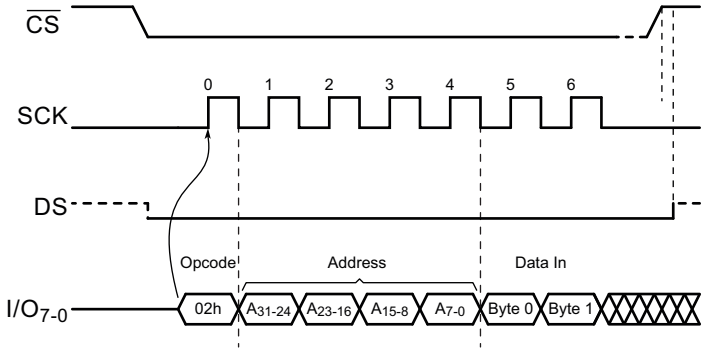


Figure 8-9. Page Program in QPI Mode - DTR

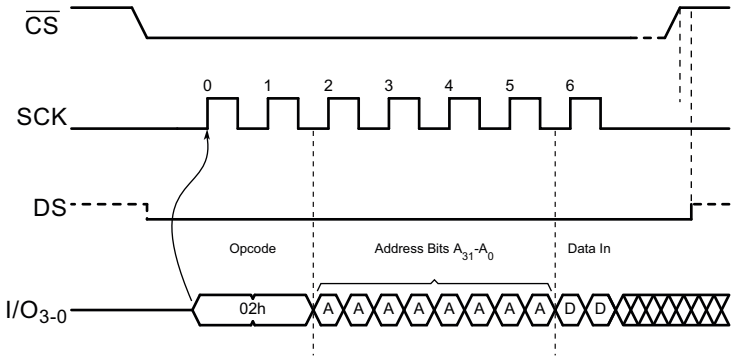
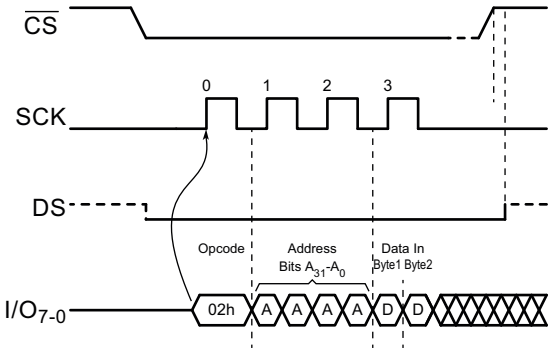


Figure 8-10. Page Program in Octal Mode - DTR



Note that an even number of bytes will be programmed when using Page Program in Octal mode - DTR.

8.2 Buffer Write (84h)

Utilizing the Buffer Write command allows data clocked in from the I/O pins to be written directly into the internal buffer without starting a write to the flash array at the same time. This write operation is faster than a 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, an opcode of 84h must be clocked into the device followed by 4 address bytes comprised of 24 dummy bits and 8 buffer address bits (BFA7 - BFA0). The 8 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 will wrap around back to the beginning of the buffer. Data will continue to be loaded into the buffer until a low-to-high transition is detected on the $\overline{CS}$ pin.

Note that the buffer will not be 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) will write the full 256-byte page, even if less than 256 bytes were written to the buffer.

Figure 8-11. Buffer Write

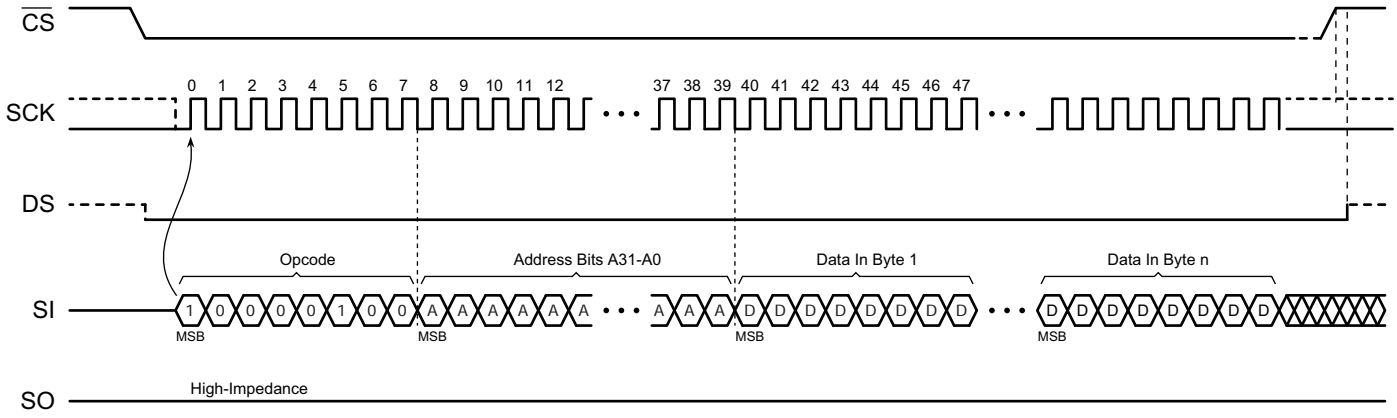


Figure 8-12. Buffer Write in QPI Mode - STR

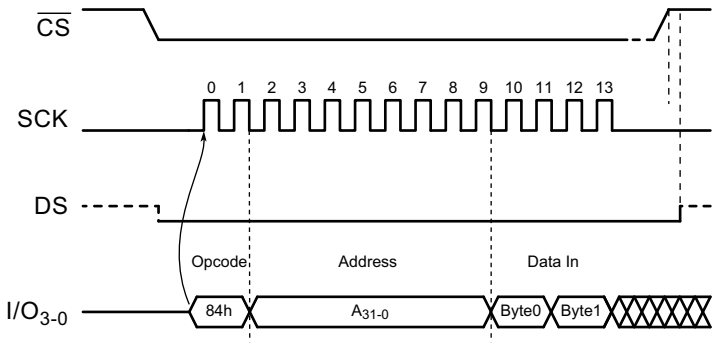


Figure 8-13. Buffer Write in Octal Mode - STR

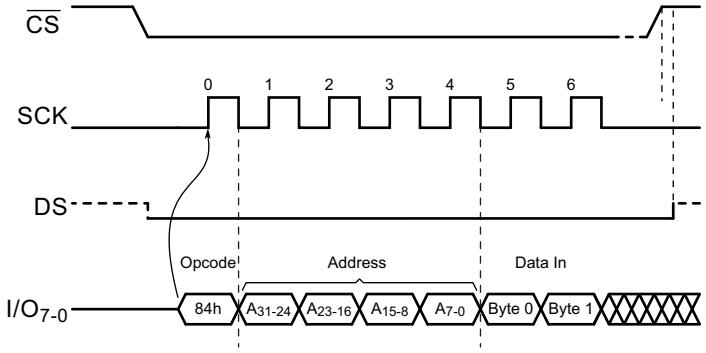


Figure 8-14. Buffer Write in QPI Mode - DTR

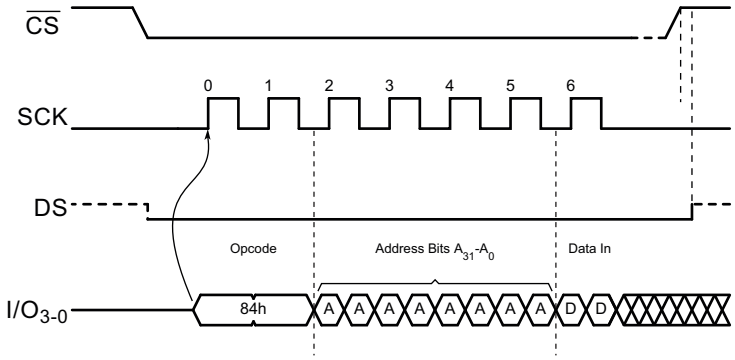
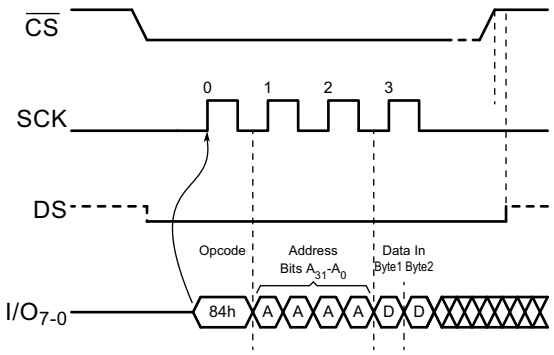


Figure 8-15. Buffer Write in Octal Mode - DTR



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 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 will write the full 256-byte page, even if less than 256 bytes were written to the buffer.

Before a Buffer to Main Memory Page Program without Built-In Erase command can be started, the Write Enable command must have been previously issued to the device (see “Write Enable (06h)” on page 48) to set the Write Enable Latch (WEL) bit of the Status Register to a logical “1” state.

To perform a Buffer to Main Memory Page Program without Built-In Erase, an opcode of 88h must be clocked into the device followed by four address bytes comprised of 10 dummy bits, 14 page address bits (PA13 - PA0) that specify the page in the main memory to be written, and 8 dummy bits.

Or described differently: The four address bytes points to a byte within the page to be written, but the lower 8 bits A7 - A0 will be set to 0, so the page program will start at the beginning of the page boundary.

When a low-to-high transition occurs on the $\overline{CS}$ pin, the device will program the data stored in the buffer into the specified page in the main memory. The page in main memory that is being programmed must have been previously erased using one of the erase commands. 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 Register will indicate that the device is busy. At some point before the program cycle completes, the WEL bit in the Status Register will be reset back to the logical “0” state.

In addition, if the memory is in the protected state (see “Protect Sector” on page 19), then the Buffer to Main Memory Page Program without Built-In Erase Program command will not be executed, and the device will return to the idle state once the $\overline{CS}$ pin has been deasserted. The WEL bit in the Status Register will be reset back to the logical “0” state if the program cycle aborts due to an incomplete address being sent, 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 will be indicated by the EPE bit in the Status Register.

Figure 8-16. Buffer to Main Memory Page Program without Built-In Erase

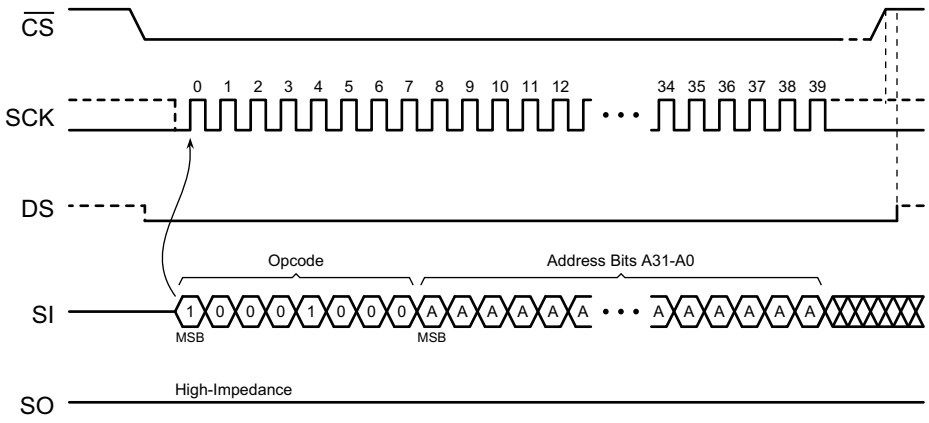


Figure 8-17. Buffer to Main Memory Page Program without Built-In Erase in QPI Mode - STR

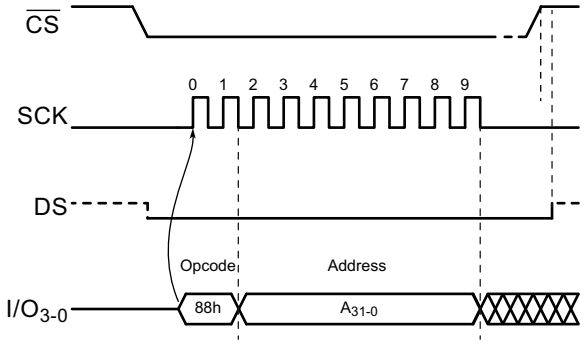


Figure 8-18. Buffer to Main Memory Page Program without Built-In Erase in Octal Mode - STR

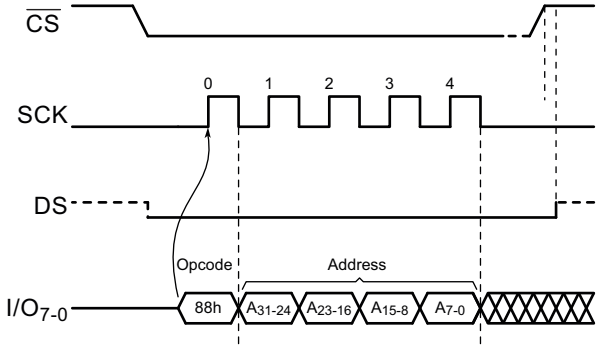


Figure 8-19. Buffer to Main Memory Page Program without Built-In Erase in QPI Mode - DTR

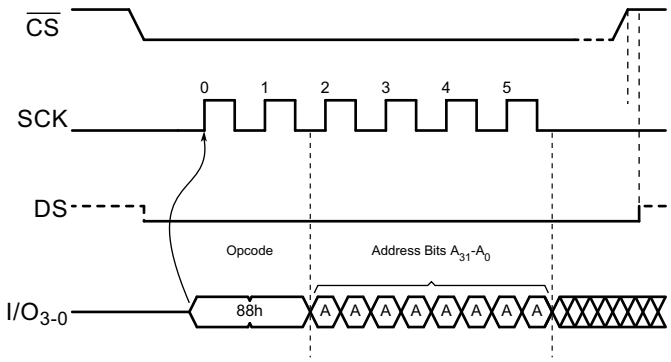
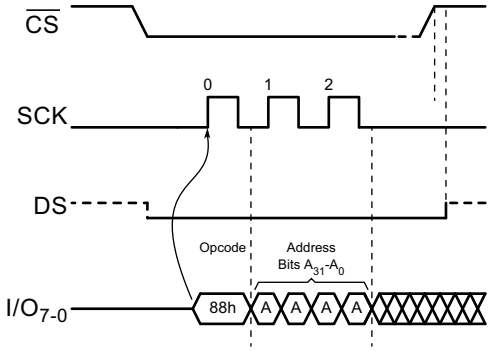


Figure 8-20. Buffer to Main Memory Page Program without Built-In Erase in Octal Mode - DTR



8.4 Block Erase

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

To perform a Block Erase, the $\overline{CS}$ pin must first be asserted and the appropriate opcode (20h, 52h, or D8h) must be clocked into the device. After the opcode has been clocked in, the 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 will be ignored. When the $\overline{CS}$ pin is deasserted, the device will erase the appropriate block. The erasing of the block is internally self-timed and should take place in a time of t_{BLKE}.

Since the Block Erase command erases a region of bytes, the lower order address bits do not need to be decoded by the device. Therefore, for a 4-Kbyte erase, address bits A11-A0 will be ignored by the device and their values can be either a logical “1” or “0”. For a 32-Kbyte erase, address bits A14-A0 will be ignored by the device. For a 64-Kbyte erase, address bits A15-A0 will be ignored by the device. Despite the lower order address bits not being decoded by the device, the complete 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 will abort the operation and no erase operation will be performed.

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

The WEL bit in the Status Register will be reset back to the logical “0” state if the erase cycle aborts due to an incomplete address being sent, the $\overline{CS}$ pin being deasserted 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 Register can be read and will indicate that the device is busy. For faster throughput, it is recommended that the Status Register be polled rather than waiting the t_{BLKE} time to determine if the

device has finished erasing. At some point before the erase cycle completes, the WEL bit in the Status Register will be reset back to the logical “0” state.

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 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 will be indicated by the EPE bit in the Status Register.

Figure 8-21. Block Erase

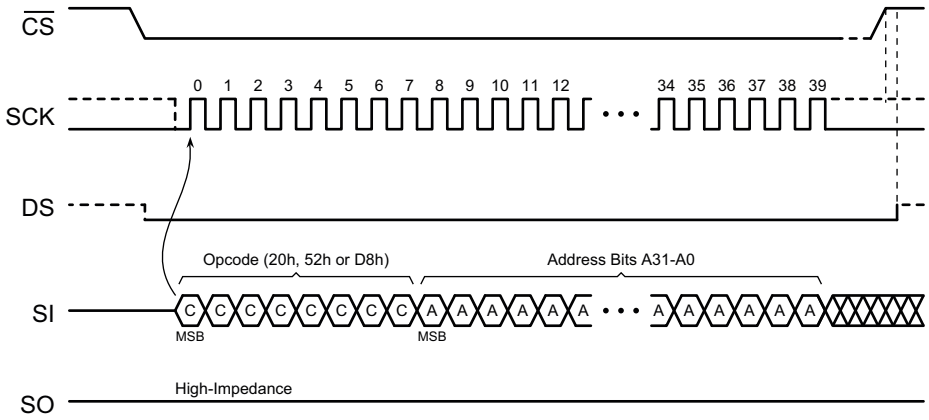


Figure 8-22. Block Erase in QPI Mode - STR

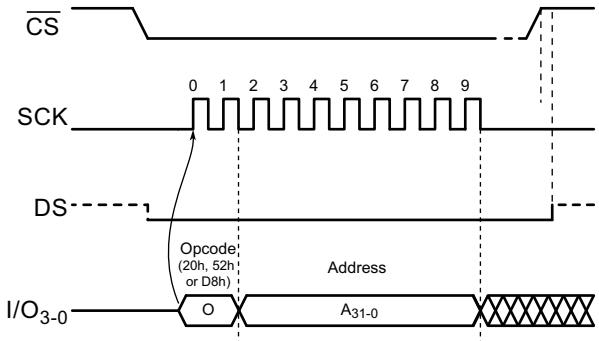


Figure 8-23. Block Erase in Octal Mode - STR

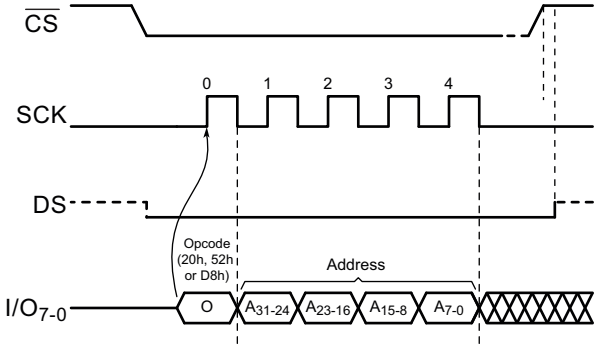


Figure 8-24. Block Erase in QPI Mode - DTR

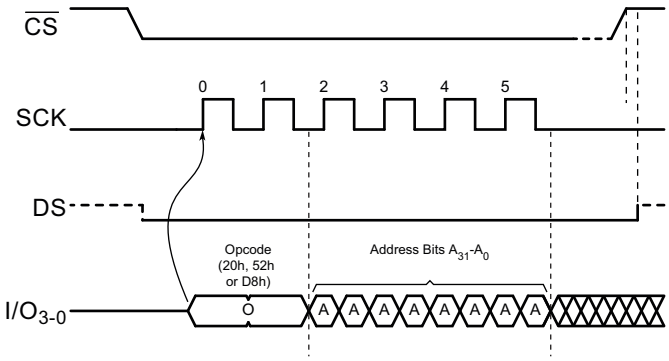
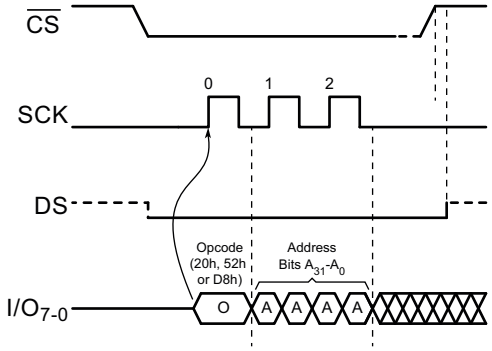


Figure 8-25. Block Erase in Octal Mode - DTR



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 Register to a logical “1” state.

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

The complete opcode must be clocked into the device before the $\overline{CS}$ pin is deasserted, and the $\overline{CS}$ pin must be deasserted on a whole byte boundary (multiples of eight bits); otherwise, no erase will be performed. In addition, if any sector in the memory array is in the protected state, then the Chip Erase command will not be executed, and the device will return to the idle state once the $\overline{CS}$ pin has been deasserted. The WEL bit in the Status Register will be reset back to the logical “0” state if the $\overline{CS}$ pin is deasserted 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 Register can be read and will indicate that the device is busy. For faster throughput, it is recommended that the Status Register be polled rather than waiting the t_{CHPE} time to determine if the device has finished erasing. At some point before the erase cycle completes, the WEL bit in the Status Register will be reset back to the logical “0” state.

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

Figure 8-26. Chip Erase

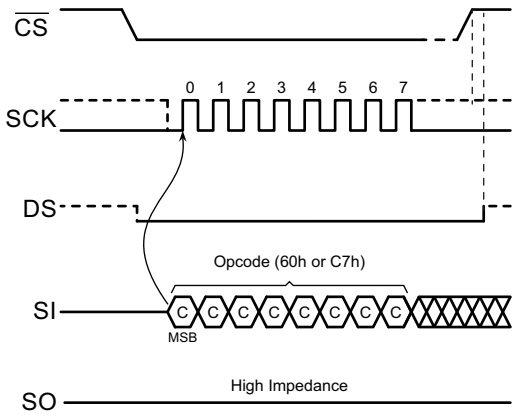


Figure 8-27. Chip Erase in QPI Mode - STR

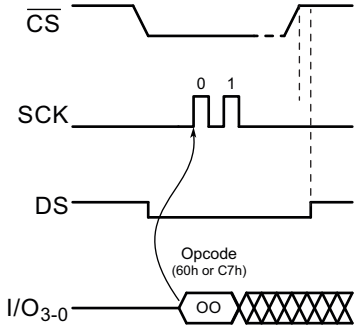


Figure 8-28. Chip Erase in Octal Mode - STR

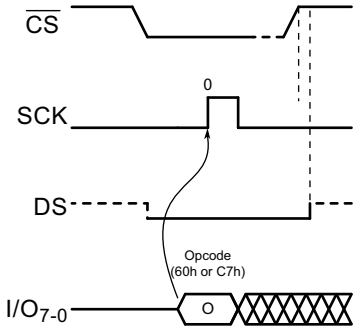


Figure 8-29. Chip Erase in QPI Mode - DTR

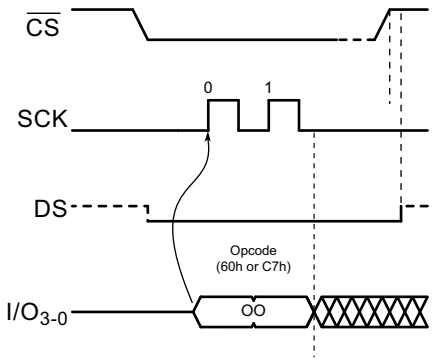
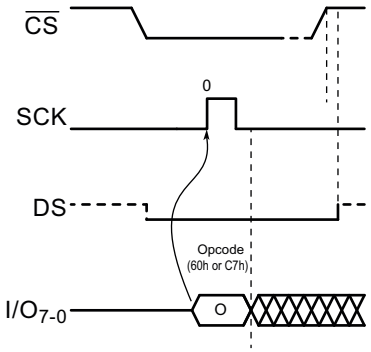


Figure 8-30. Chip Erase in Octal Mode - DTR



8.6 Program/Erase Suspend (B0h)

The ATXP064R has Read-While-Write as described in Section 4.1, while the ATXP064 does not include this feature.

The Program/Erase Suspend commands can be used in addition to or instead of the Read-While-Write option. While Read-While-Write is limited to only allow read operations during program or erase, the Program/Erase Suspend command will also allow 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 will 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 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, Chip Erase 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 Register.

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

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 will output 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 will then start 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 will abort and the WEL bit in the Status Register will be reset back to the logical “0” state. During an Erase Suspend, 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 will be indicated by the states of both the ES and PS bits in the Status Register being set to the logical “1” state. Another erase operation cannot be started while in Erase Suspend.

Neither program nor erase commands are allowed during a Program Suspend. If attempted, the operation will be aborted.

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

Note: Program Suspend and Erase Suspend commands cannot be used if ADPD or AUDPD in Status 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
	ATXP064	ATXP064R		
Read Commands				
Read Array, 4 byte address (0Bh, 0Ch, 13h)	Not Allowed	Allowed (1)	Allowed(1) (2)	Allowed(1)(2)
Read Array, 3 byte address (03h)	Not Allowed	Allowed(1)	Allowed	Allowed
Buffer Read (D4h)	Not Allowed	Not Allowed	Allowed	Allowed
Program and Erase Commands				
Block Erase (20h, 52h, D8h)	Not Allowed	Not Allowed	Not Allowed	Not Allowed
Chip Erase (C7h, 60h)	Not Allowed	Not Allowed	Not Allowed	Not Allowed
Byte/Page Program (02h)	Not Allowed	Not Allowed	Not Allowed	Allowed(2)
Buffer Write (84h)	Not Allowed	Not Allowed	Not Allowed	Allowed
Buffer to Main Memory Page Program without Built-In Erase	Not Allowed	Not Allowed	Not Allowed	Allowed
Program/Erase Suspend (B0h)	Allowed	Allowed	N/A (3)	Program Suspend Allowed
Program/Erase Resume (D0h)	Allowed	Allowed	Allowed	Allowed
Protection Commands				
Write Enable (06h)	Not Allowed	Not Allowed	Allowed	Allowed
Write Disable (04h)	Not Allowed	Not Allowed	Allowed (4)	Allowed(4)
Protect Sector (36h)	Not Allowed	Not Allowed	Not Allowed	Not Allowed
Unprotect Sector (39h)	Not Allowed	Not Allowed	Not Allowed	Not Allowed
Read Sector Protection Registers (3Ch)	Not Allowed	Not Allowed	Allowed	Allowed
Security Commands				
Program OTP Security Register (9Bh)	Not Allowed	Not Allowed	Not Allowed	Not Allowed
Read OTP Security Register (77h)	Not Allowed	Not Allowed	Allowed	Allowed
Status Register Commands				
Read Status/Control Registers (65h)	Allowed	Allowed	Allowed	Allowed
Read Status Register Byte 1 (05h)	Allowed	Allowed	Allowed	Allowed
Active Status Interrupt (25h)	Allowed	Allowed	Allowed	Allowed
Write Status Register (01h, 31h)	Not Allowed	Not Allowed	Not Allowed	Not Allowed
Write Status/Control Registers (71h)	Not Allowed	Not Allowed	Not Allowed	Not Allowed
Miscellaneous Commands				
Terminate Operation (F0h)	Allowed (5)	Allowed(5)	Allowed(5)	Allowed(5)
Reset Enable (66h)	Allowed	Allowed	Allowed	Allowed
Reset (99h)	Allowed(5) (6) (7)	Allowed(5)(6)(7)	Allowed(5)(6)(7)	Allowed(5)(6)(7)
Read Manufacturer and Device ID (9Fh)	Allowed	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
	ATXP064	ATXP064R		
Deep Power-Down (B9h)	Not Allowed	Not Allowed	Not Allowed	Not Allowed
Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh)	N/A	N/A	N/A	N/A
Ultra-Deep Power-Down (79h)	Not Allowed	Not Allowed	Not Allowed	Not Allowed
Read SFDP (5Ah)	Not Allowed	Not Allowed	Allowed	Allowed
Enter QPI Mode (38h)	Not Allowed	Not Allowed	Not Allowed	Not Allowed
Enter Octal Mode (E8h)	Not Allowed	Not Allowed	Not Allowed	Not Allowed
Return to Standard SPI Mode (FFh)	Not Allowed	Not Allowed	Not Allowed	Not Allowed
Echo (AAh)	Allowed	Allowed	Allowed	Allowed
Echo with inversion (A5h)	Allowed	Allowed	Allowed	Allowed

1. This command is only allowed for an alternate memory bank, *not* for the memory bank where the Program or Erase command is in progress.
2. Allowed for all 2 Mbit sectors other than the one currently suspended.
3. Another program or erase operation cannot be started during a program suspend, so an additional program or erase suspend is not applicable.
4. Current write in progress will finish. To stop a write in progress, use the Terminate Operation or Reset command.
5. 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.
6. Reset is only allowed directly after a Reset Enable.
7. Reset will be ignored during OTP programming and while device is executing the Terminate Operation (F0h) command.

Figure 8-31. Program/Erase Suspend

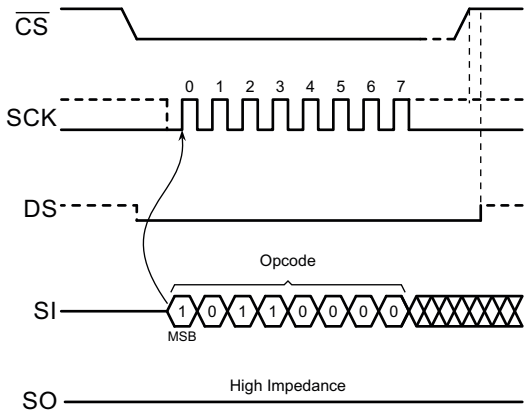


Figure 8-32. Program/Erase Suspend in QPI Mode - STR

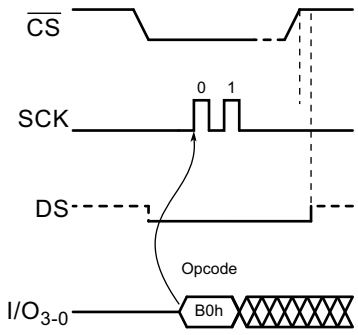


Figure 8-33. Program/Erase Suspend in Octal Mode - STR

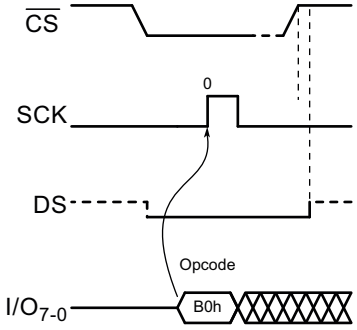


Figure 8-34. Program/Erase Suspend in QPI Mode - DTR

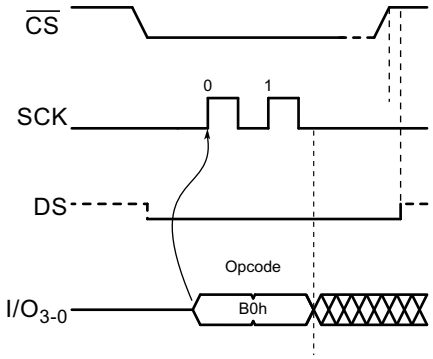
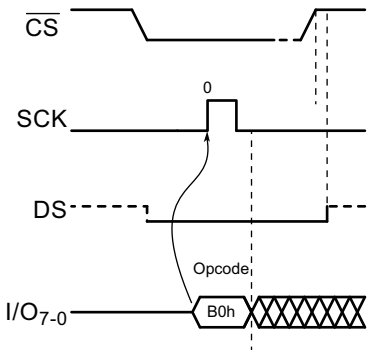


Figure 8-35. Program/Erase Suspend in Octal Mode - DTR



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 instruction will be accepted by the device only if the SUS bit in the Status Register equals 1 and the RDY/BSY bit equals 0. If the SUS bit equals 0 or the RDY/BSY bit equals to 1, the Program/Erase Resume command will be ignored by the device. As with the Program/Erase Suspend command, the Write Enable command does not need to be issued prior to the Program/Erase Resume command being issued. Therefore, the Program/Erase Resume command operates independently of the state of the WEL bit in the Status Register.

To perform Program/Erase Resume, the CS pin must first be asserted and opcode D0h must be clocked into the device.

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

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

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

Figure 8-36. Program/Erase Resume

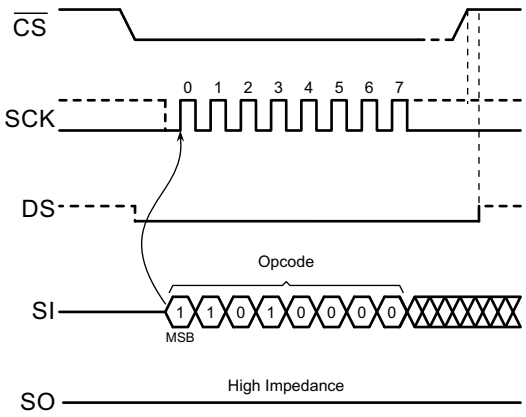


Figure 8-37. Program/Erase Resume in QPI Mode - STR

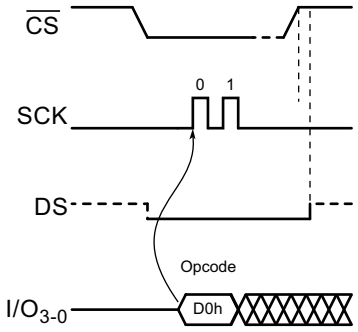


Figure 8-38. Program/Erase Resume in Octal Mode - STR

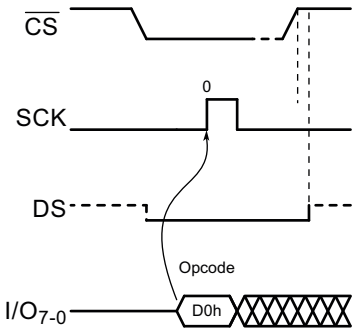


Figure 8-39. Program/Erase Resume in QPI Mode - DTR

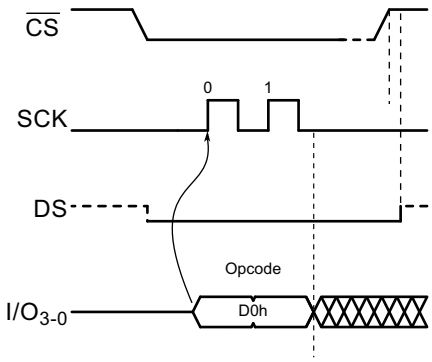
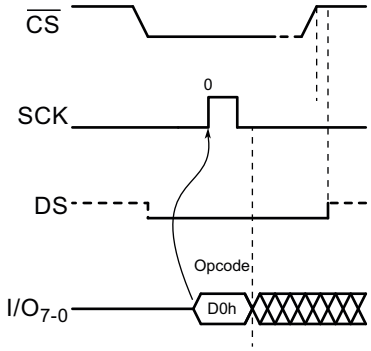


Figure 8-40. Program/Erase Resume in Octal Mode - DTR



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 Register to a logical “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 Register is not set prior to the issuance of one of these commands, then the command will not be executed.

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

Figure 9-1. Write Enable

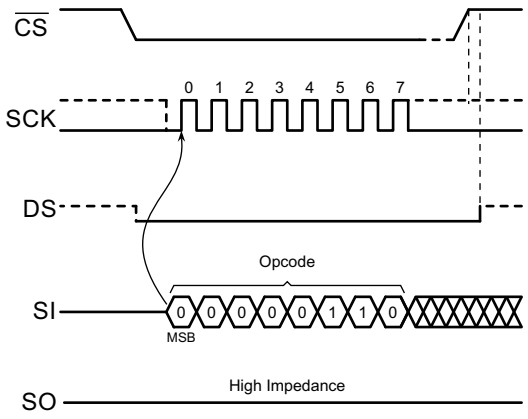


Figure 9-2. Write Enable in QPI Mode - STR

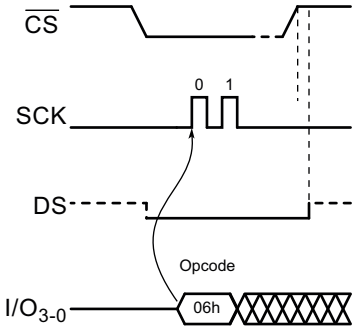


Figure 9-3. Write Enable in Octal Mode - STR

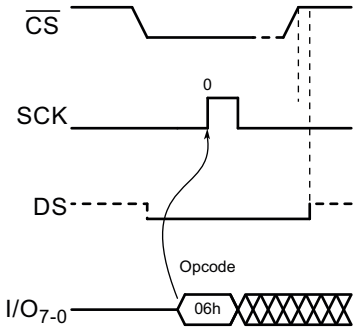


Figure 9-4. Write Enable in QPI Mode - DTR

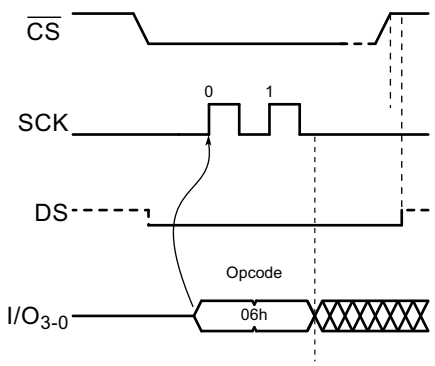
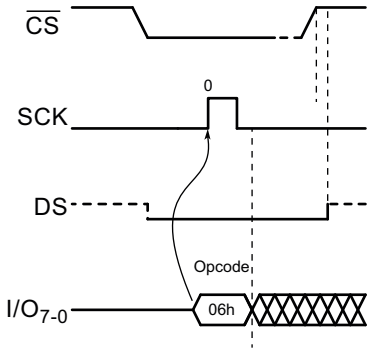


Figure 9-5. Write Enable in Octal Mode - DTR



9.2 Write Disable (04h)

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

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

Figure 9-6. Write Disable

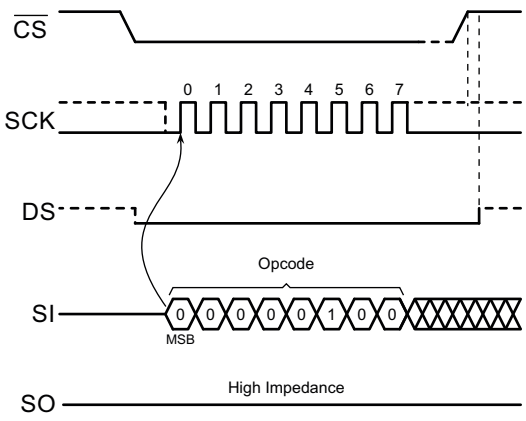


Figure 9-7. Write Disable in QPI Mode - STR

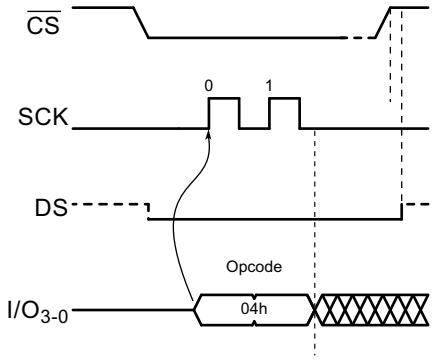


Figure 9-8. Write Disable in Octal Mode - STR

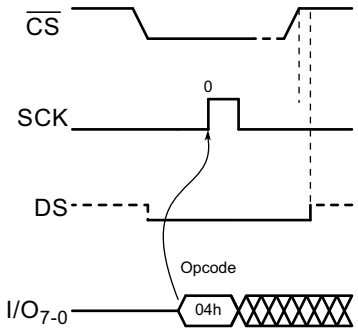


Figure 9-9. Write Disable in QPI Mode - DTR

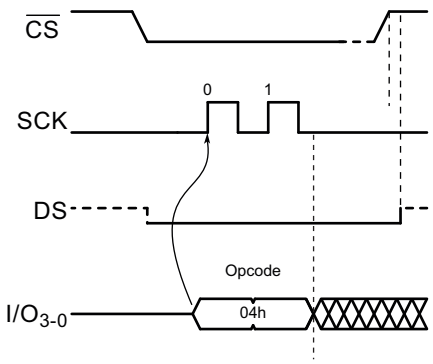
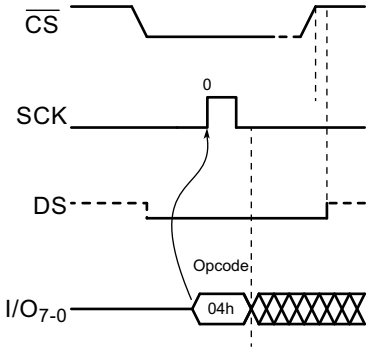


Figure 9-10. Write Disable in Octal Mode - DTR



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 will default to the logical “1” state indicating that all sectors are protected and cannot be programmed or erased.

Issuing the Protect Sector command to a particular sector address will set the corresponding Sector Protection Register to the logical “1” state. The following table outlines the two states of the Sector Protection Registers. Note that the Sector Protection Register will be revert to the default state after waking up from Ultra-Deep Power-Down of after a JEDEC Standard Hardware 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 Register to a logical “1”. To issue the Protect Sector command, the $\overline{CS}$ pin must first be asserted and the opcode of 36h must be clocked into the device followed by four address bytes designating any address within the sector to be protected. Any additional data clocked into the device will be ignored. When the $\overline{CS}$ pin is deasserted, the Sector Protection Register corresponding to the physical sector addressed by A31 - A0 will be set to the logical “1” state, and the sector itself will then be

protected from program and erase operations. In addition, the WEL bit in the Status Register will be reset back to the logical “0” state.

The complete 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 will abort the operation, the state of the Sector Protection Register will be unchanged, and the WEL bit in the Status Register will be reset to a logical “0”.

As a safeguard against accidental or erroneous protecting or unprotecting of sectors, the Sector Protection Registers can themselves be locked from updates by using the SPRL (Sector Protection Registers Locked) bit of the Status Register (please refer to the Status Register Byte 1 description for more details). If the Sector Protection Registers are locked, then any attempts to issue the Protect Sector command will be ignored, and the device will reset the WEL bit in the Status Register back to a logical “0” and return to the idle state once the $\overline{CS}$ pin has been deasserted.

Figure 9-11. Protect Sector

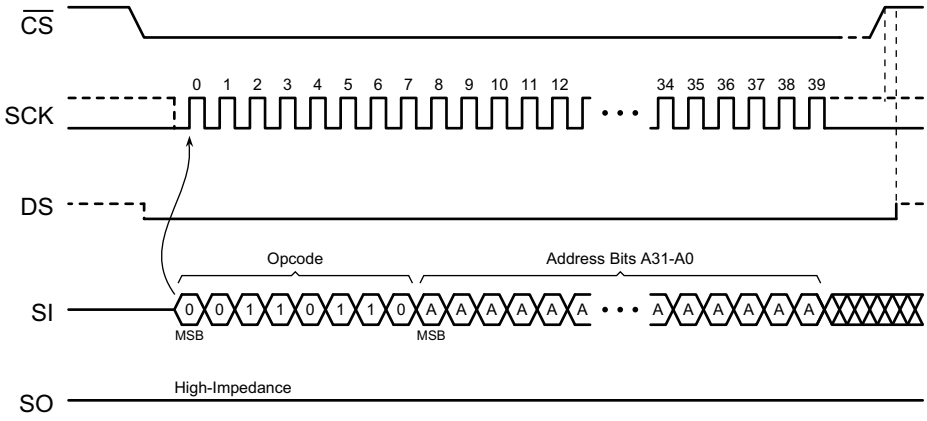


Figure 9-12. Protect Sector in QPI Mode - STR

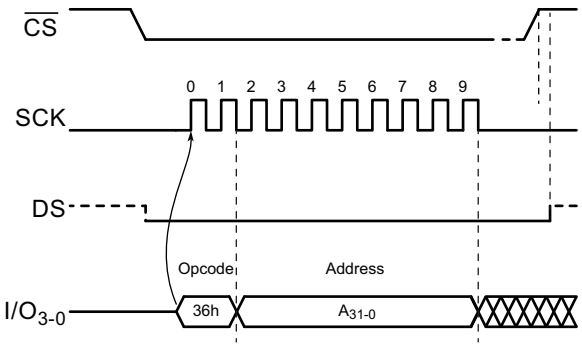


Figure 9-13. Protect Sector in Octal Mode - STR

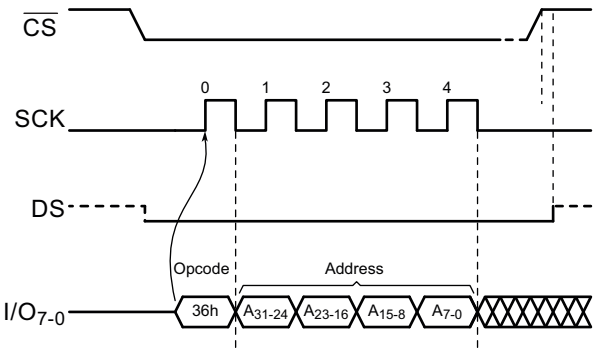


Figure 9-14. Protect Sector in QPI Mode - DTR

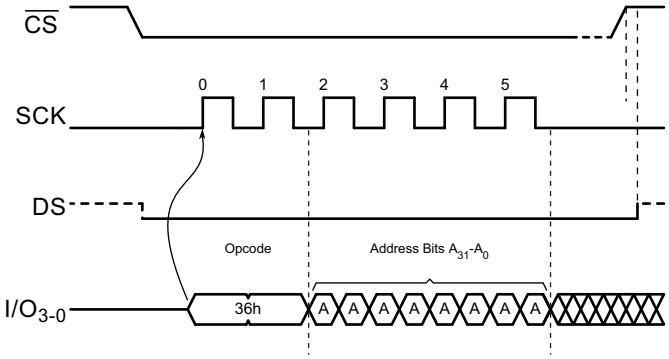
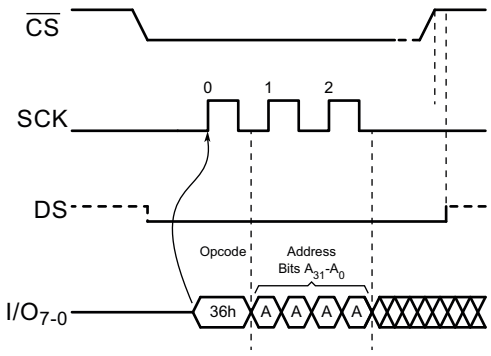


Figure 9-15. Protect Sector in Octal Mode - DTR



9.4 Unprotect Sector (39h)

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

Before the Unprotect Sector command can be issued, the Write Enable command must have been previously issued to set the WEL bit in the Status Register to a logical “1”. To issue the Unprotect Sector command, the $\overline{CS}$ pin must first be asserted and the opcode of 39h must be clocked into the device. After the opcode has been clocked in, the 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 will be ignored. When the $\overline{CS}$ pin is deasserted, the Sector Protection Register corresponding to the sector addressed by A31 - A0

will be reset to the logical “0” state, and the sector itself will be unprotected. In addition, the WEL bit in the Status Register will be reset back to the logical “0” state.

The complete 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 will abort the operation, the state of the Sector Protection Register will be unchanged, and the WEL bit in the Status Register will be reset to a logical “0”.

As a safeguard against accidental or erroneous locking or unlocking of sectors, the Sector Protection Registers can themselves be locked from updates by using the SPRL (Sector Protection Registers Locked) bit of the Status Register (please refer to the Status Register Byte 1 description for more details). If the Sector Protection Registers are locked, then any attempts to issue the Unprotect Sector command will be ignored, and the device will reset the WEL bit in the Status Register back to a logical “0” and return to the idle state once the $\overline{CS}$ pin has been deasserted.

Figure 9-16. Unprotect Sector

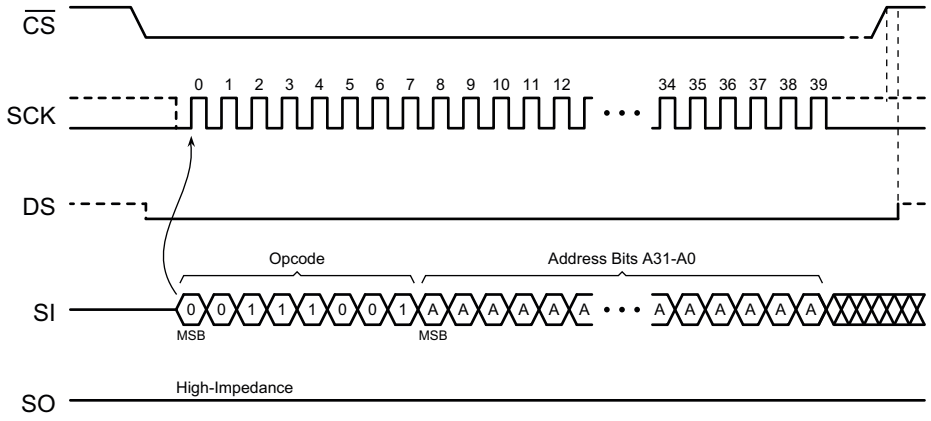


Figure 9-17. Unprotect Sector in QPI Mode - STR

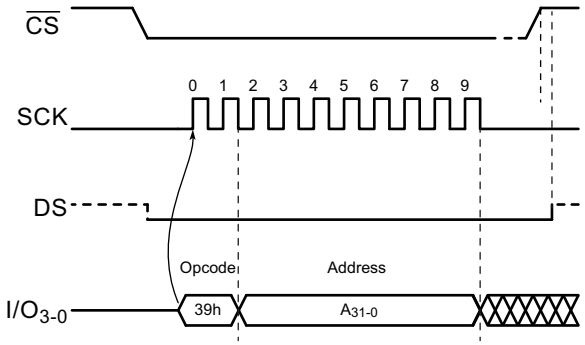


Figure 9-18. Unprotect Sector in Octal Mode - STR

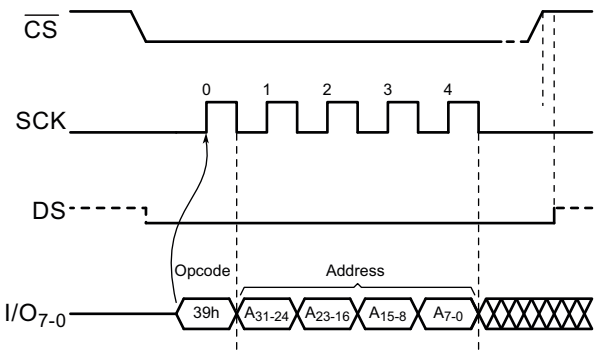


Figure 9-19. Unprotect Sector in QPI Mode - DTR

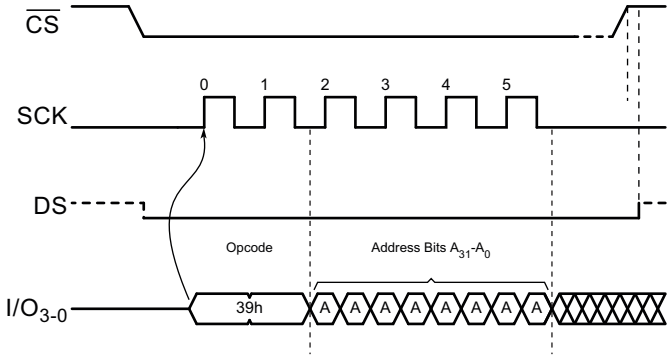
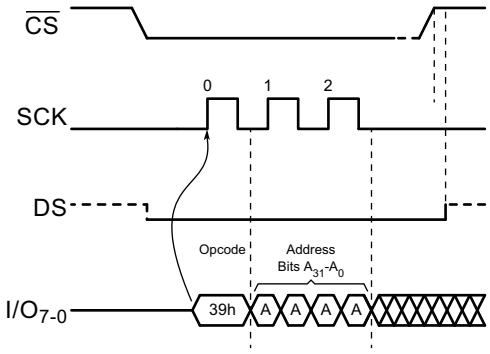


Figure 9-20. Unprotect Sector in Octal Mode - DTR



9.5 Global Protect/Unprotect

The Global Protect and Global Unprotect features can work in conjunction with the Protect Sector and Unprotect Sector functions. 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 is accomplished by writing a certain combination of data to the Status Register using the Write Status Register command. (See “Write Status/Control Registers (71h)” on page 78 or “Write Status Register Byte 1 (01h)” on page 80 for command execution details) The Write Status Register command is also used to modify the SPRL (Sector Protection Registers Locked) bit to control hardware and software locking.

To perform a Global Protect, the appropriate $\overline{WP}$ pin and SPRL conditions must be met, and the system must write a logical “1” to bits 5, 4, 3, and 2 of the Status Register. Conversely, to perform a Global Unprotect, the same $\overline{WP}$ and SPRL conditions must be met but the system must write a logical “0” to bits 5, 4, 3, and 2 of the Status Register. Table 9-2 details the conditions necessary for a Global Protect or Global Unprotect 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 Register is in the logical “0” state (Sector Protection Registers are not locked), then writing a 00h to the Status Register will perform a Global Unprotect without changing the state of the SPRL bit. Similarly, writing a

7Fh to the Status Register will perform a Global Protect and keep the SPRL bit in the logical “0” state. The SPRL bit can, of course, be changed to a logical “1” by writing an FFh if software-locking or hardware-locking is desired along with the Global Protect.

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

When writing to the Status Register, bits 5, 4, 3, and 2 will not actually be modified but will be decoded by the device for the purposes of the Global Protect and Global Unprotect functions. Only bit 7, the SPRL bit, will actually be modified. Therefore, when reading the Status Register, bits 5, 4, 3, and 2 will not reflect the values written to them but will instead indicate the status of the $\overline{WP}$ pin and the sector protection status. Please refer to the Read Status Register Byte 1 (05h) section and Table 11-3 on page 67 for details on the Status 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, will not determine the status of the $\overline{WP}$ pin.

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

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

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

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

In addition to reading the individual Sector Protection Registers, the Software Protection Status (SWP) bit in the Status Register can be read to determine if all, some, or none of the sectors are software protected. (See “Write Status/Control Registers (71h)” on page 78 for more details.)

Figure 9-21. Read Sector Protection Register

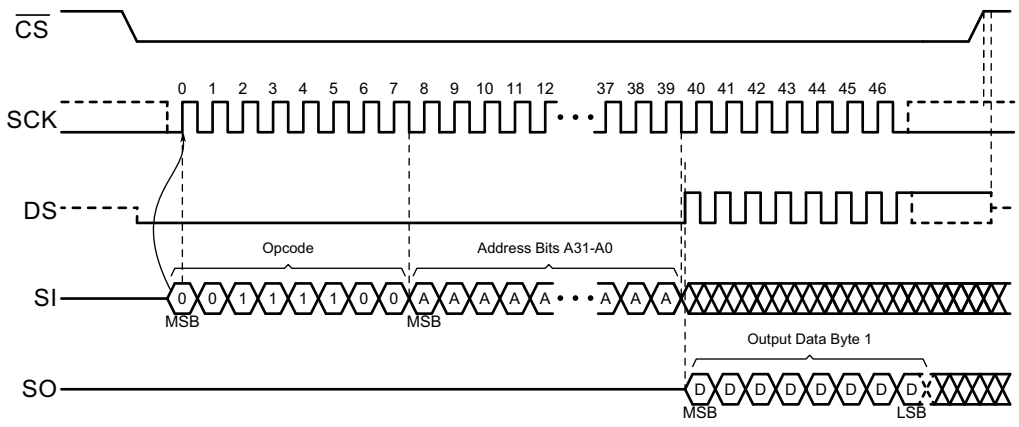


Figure 9-22. Read Sector Protection Register in QPI Mode - STR

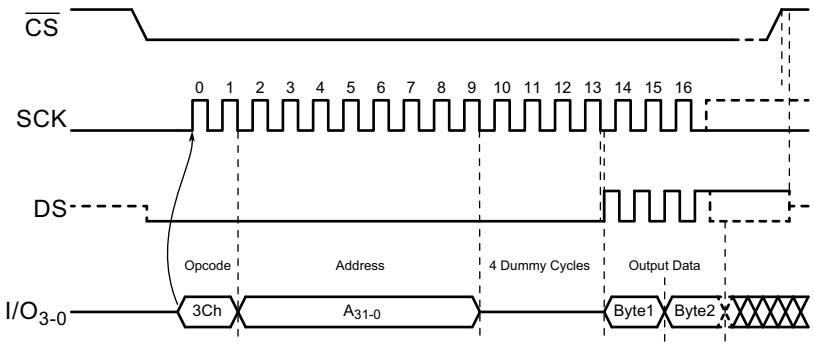


Figure 9-23. Read Sector Protection Register in Octal Mode - STR

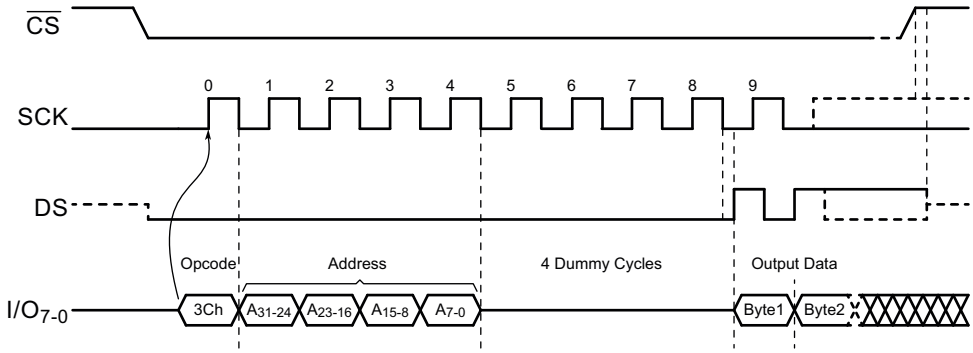


Figure 9-24. Read Sector Protection Register in QPI Mode - DTR

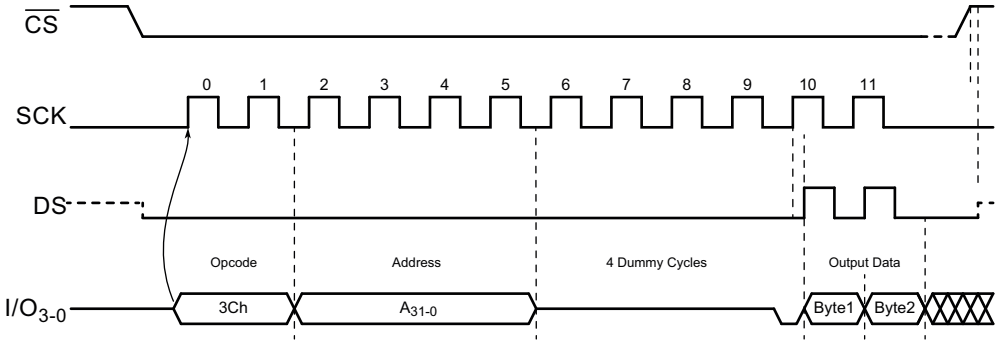
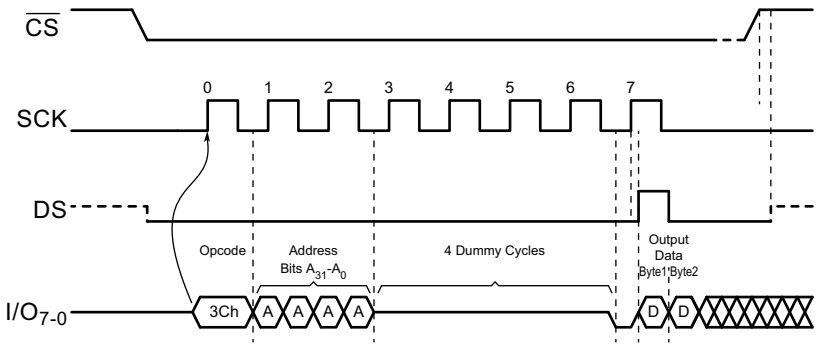


Figure 9-25. Read Sector Protection Register in Octal Mode - DTR



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 Register, is used to control the hardware locking mechanism of the device. For hardware locking to be active, two conditions must be met – the $\overline{WP}$ pin must be asserted and the SPRL bit must be in the logical “1” state.

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

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

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

In QPI and OPI modes of operation, 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 logical 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 will be ignored. Global Protect and Unprotect cannot be performed.
1	0		Can be modified from 0 to 1	Unlocked and modifiable using the Protect and Unprotect Sector commands. Global Protect and Unprotect can also be performed.
1	1	Software Locked	Can be modified from 1 to 0	Locked in current state. Protect and Unprotect Sector commands will be ignored. Global Protect and Unprotect cannot be performed.

10. Security Commands

10.1 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 will contain 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 Register to a logical “1”. To program the OTP Security Register, the $\overline{CS}$ pin must first be asserted and an opcode of 9Bh must be clocked into the device followed by the 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 A31-A7 will be ignored by the device and their values can be either a logical “1” or “0”. After the address bytes have been clocked in, data can then be clocked into the device and will be stored in the internal buffer. If byte number 127 (7Fh) is among the bytes written, the OTP Security Register will be locked upon completion of the command execution, and no subsequent Program OTP Security Register writes will be executed.

Note that any data that is sent to the device that goes beyond the end of the 128-byte user-programmable space will wrap around back to the beginning of the OTP Security Register. For example, if the starting address denoted by A31-A0 is 0000007Eh, and three bytes of data are sent to the device, then the first two bytes of data will be programmed at OTP Security Register addresses 7Eh and 7Fh while the last byte of data will be programmed at address 00h. As location 7Fh is written by this operation, the device is locked after these bytes have been written. 00h will still be written in this case. The remaining bytes in the OTP Security Register (addresses 01h through 7Dh) will not be programmed by this operation, and will remain in the state they previously had. If they have not been programmed previously, they will 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 will be latched into the internal buffer.

When the $\overline{CS}$ pin is deasserted, the device will take the data stored in the internal buffer and program it into the appropriate OTP Security Register locations based on the starting address specified by A31-A0 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 will not be programmed and will remain in the erased state (FFh). The programming of the data bytes is internally self-timed and should take place in a time of t_{OTPP}.

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 will abort the operation and the user-programmable portion of the OTP Security Register will not be programmed. The WEL bit in the Status Register will be reset back to the logical “0” state if the OTP Security Register program cycle aborts due to an incomplete address being sent, an incomplete byte of data being sent, the $\overline{CS}$ pin being deasserted 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 Register can be read and will indicate that the device is busy. For faster throughput, it is recommended that the Status Register be polled rather than waiting the t_{OTPP} time to determine if the data bytes have finished programming. At some point before the OTP Security Register programming completes, the WEL bit in the Status Register will be reset back to the logical “0” state.

If the device is powered-down during the OTP Security Register program cycle, then the contents of the 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 will be altered from its previous state when this command is issued.

Figure 10-1. Program OTP Security Register

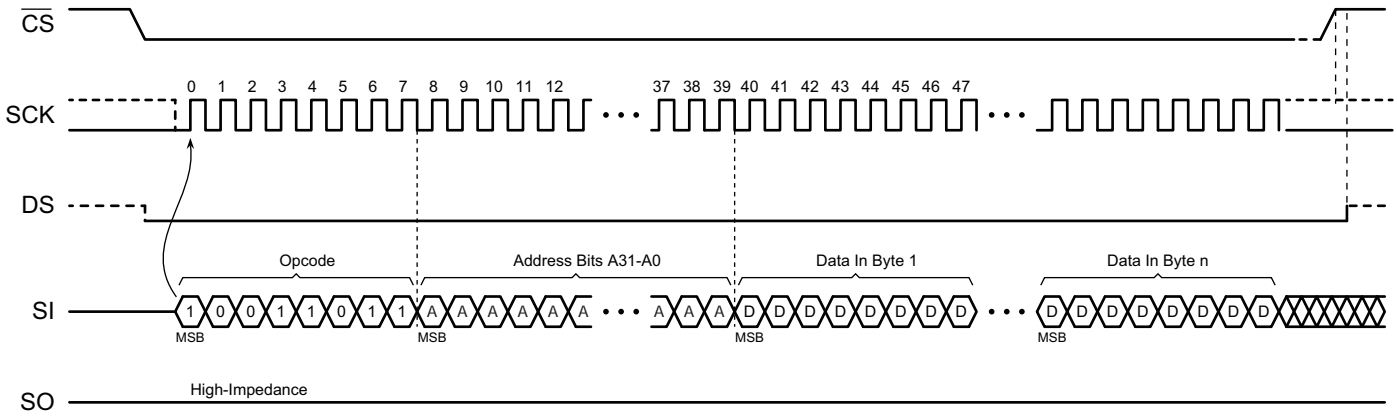


Figure 10-2. Program OTP Security Register in QPI Mode - STR

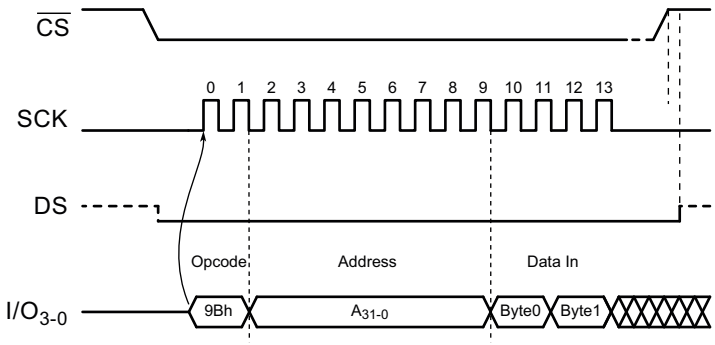


Figure 10-3. Program OTP Security Register in Octal Mode - STR

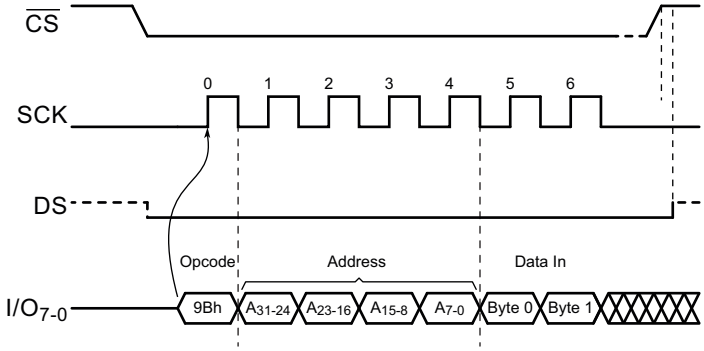


Figure 10-4. Program OTP Security Register in QPI Mode - DTR

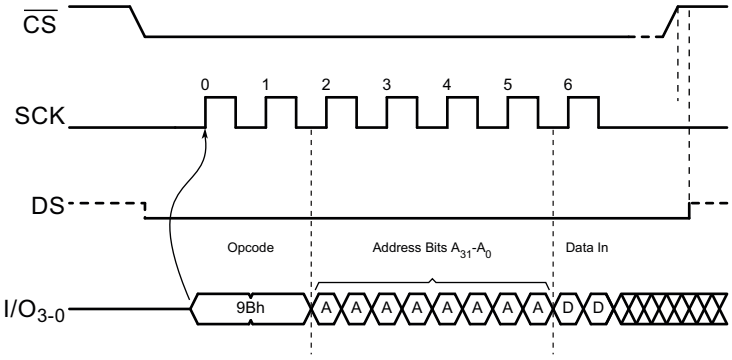
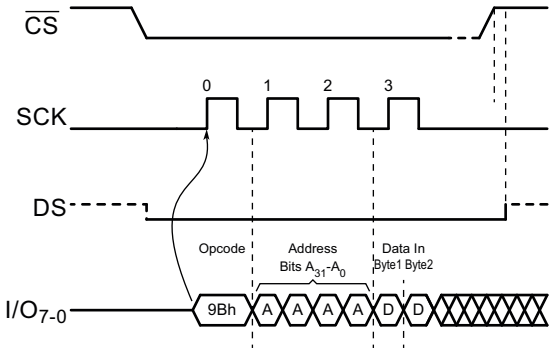


Figure 10-5. Program OTP Security Register in Octal Mode - DTR



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 opcode of 77h must be clocked into the device. After the opcode 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 the QPI and Octal modes, 8 or more dummy cycles are used as shown in Table 11-8, Dummy Clock cycles. Note that the table refers to full clock cycles, also for DTR 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 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 OTP Security Register has been read, the device will continue reading back at the beginning of the register (000000h). No delays will be incurred when wrapping around from the end of the register to the beginning of the register.

Deasserting the $\overline{CS}$ pin will terminate the read operation and put the 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 10-6. Read OTP Security Register

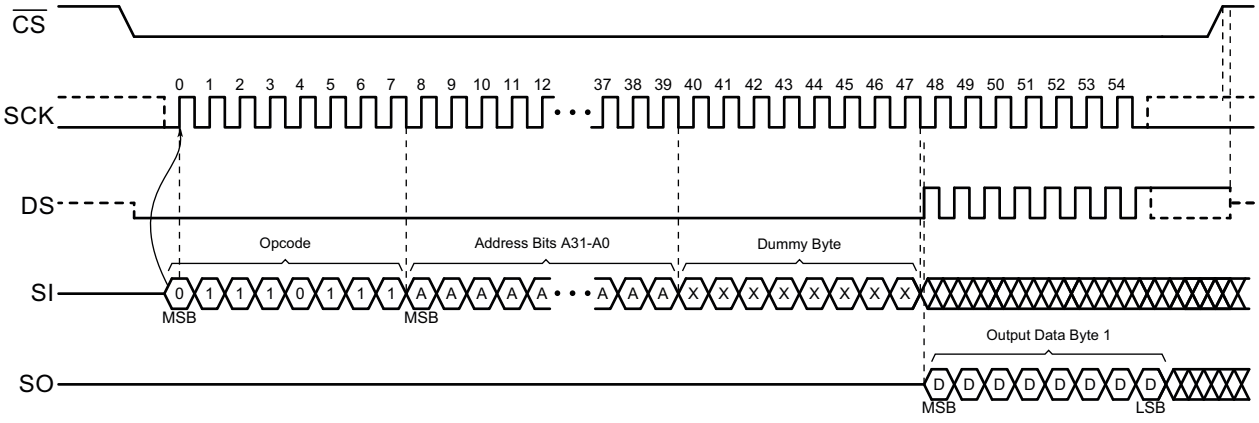


Figure 10-7. Read OTP Security Register in QPI Mode - STR

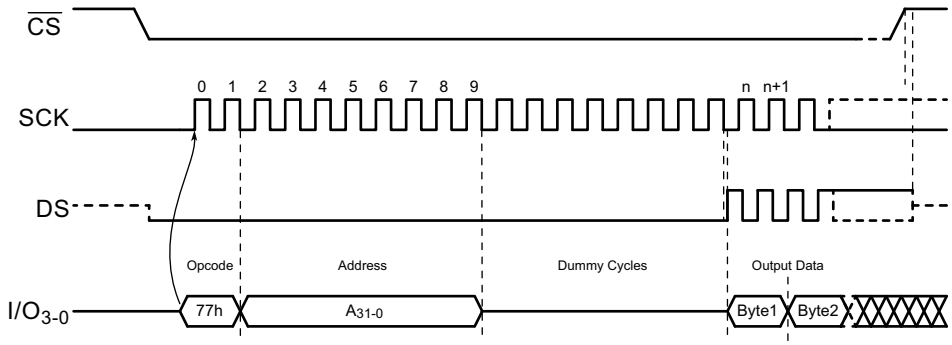


Figure 10-8. Read OTP Security Register in Octal Mode - STR

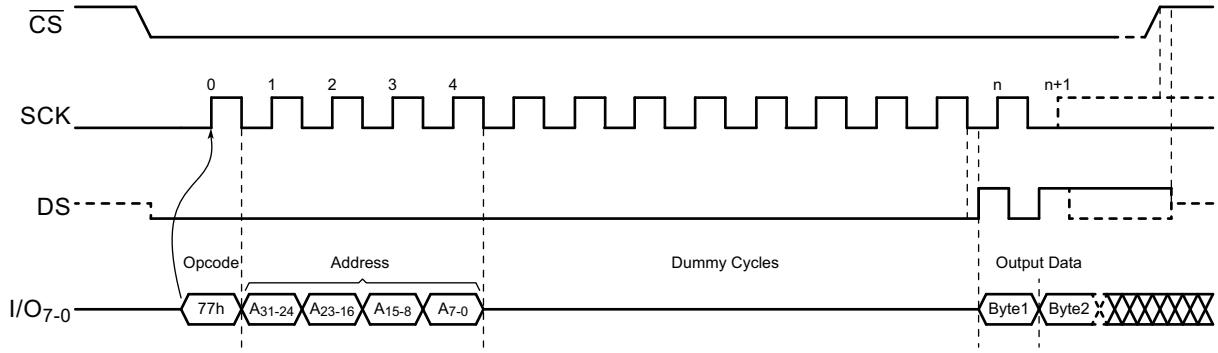


Figure 10-9. Read OTP Security Register in QPI Mode - DTR

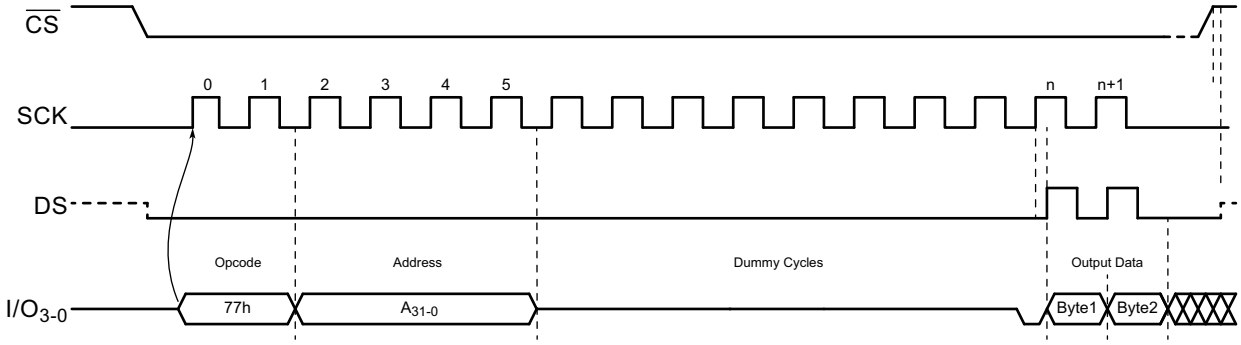
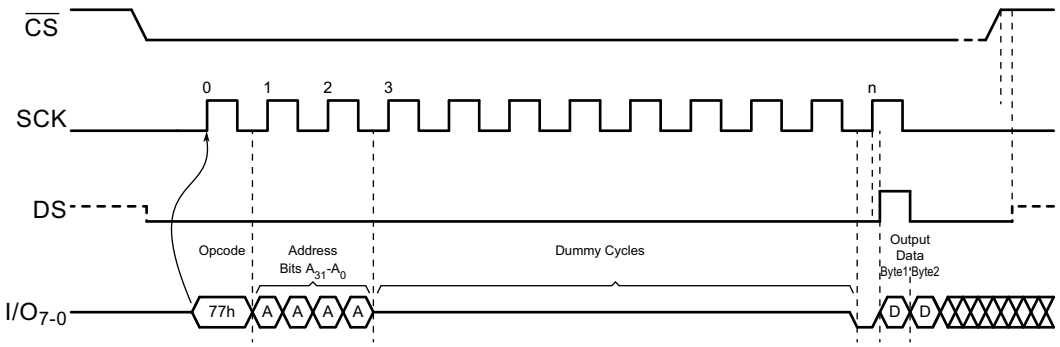


Figure 10-10.Read OTP Security Register in Octal Mode - DTR



11. Status and Control Registers - Descriptions and Commands

The ATXP064 / ATXP064R has 3 Volatile Status and Control registers as listed in Table 11-1, and 1 Non-Volatile Status and Control register as listed in Table 11-2. All Status and Control Registers can be read and written by the generic Read Status/Control Registers (65h) and Write Status/Control Registers (71h). For backwards compatibility, the specific Read Status Register Byte 1 (05h) command is also supported for Status Register Byte 1, and the specific Write Status Register Byte 1 (01h) and Write Status Register Byte 2 (31h) commands are supported for Status 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 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/Control Registers (71h) command. It is not recommended to try to write both the Volatile and the Non-volatile registers at the same time. Non-Volatile registers will only be written if the Write Status/Control 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 will 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 Register Byte 1
2	Status Register Byte 2
3	Status/Control Register Byte 3
4	Status/Control Register Byte 4

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 Register Byte 1

Status Register Byte 1 contains the Status and Control Registers listed inTable 11-3. Detailed description of each bit follows below. Note that when reading Status Register Byte 1 while in Ultra-Deep Power-Down, all bits will read as 1.

Table 11-3. Status Register Format - Byte 1

Bit [1]	Name		Type [2]	Description	
7	SPRL	Sector Protection Registers Locked	R/W	0	Sector Protection Registers are unlocked (default)
				1	Sector Protection Registers are locked.
6	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).
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 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, will read 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 will *not* be erased by a JEDEC Hardware Reset or by exercising the RESET pin. However, it will be erased by 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 logical “1” state, all Sector Protection Registers are locked and cannot be modified with the Protect Sector and Unprotect Sector commands (the device will ignore these commands). In addition, the Global Protect and Global Unprotect features cannot be performed. Any sectors that are presently protected will remain protected, and any sectors that are presently unprotected will remain unprotected.

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

The SPRL bit can be modified freely whenever the $\overline{WP}$ pin is deasserted. However, if the $\overline{WP}$ pin is asserted, then the SPRL bit may only be changed from a logical “0” (Sector Protection Registers are unlocked) to a logical “1” (Sector Protection Registers are locked). In order to reset the SPRL bit back to a logical “0” using the Write Status Register command, the $\overline{WP}$ pin will have to first be deasserted.The SPRL bit is the only bit of the Status Register 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 will be set to the logical “1” state. The EPE bit will not be set if an erase or program operation aborts for any reason such as an attempt to erase or program a protected region or if the WEL bit is not set prior to an erase or program operation. The EPE bit will be updated after every erase and program operation.

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 UDPDN is done with JEDEC Standard Hardware Reset or Hardware Reset (RESET Pin) 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 UDPDN is done with JEDEC Standard Hardware Reset or Hardware Reset (RESET Pin).

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 will *not* be 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 $\overline{RESET}$ pin. However, it will be erased by a full power cycle.

Note also that the EPE bit will *not* be set if a HW Reset (Reset pin or Jedec Hardware Reset) occurs or a Terminate Operation, (F0h) or Reset Enable (66h) and Reset (99h) command sequence is issued *during* a Program or Erase operation, even though the operation will be terminated and it is likely that the Program or Erase operation will not be successful.

If a Terminate Operation, (F0h) or Reset Enable (66h) and Reset (99h) command sequence is issued while a Program or Erase operation is actually in progress, the EPE bit will not be updated.

Upon recovery from Ultra-Deep Power-Down, all internal volatile registers will be at their Power-On default state, except for the EPE bit in Status Register Byte 1 in the following case: Even though EPE is a volatile bit, it will *not* be 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 will *not* be erased by a JEDEC Hardware Reset or by exercising the $\overline{RESET}$ pin. However, it will be 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 or Hardware Reset (RESET Pin) while in Auto Deep Power-Down Mode after a Program or Erase Operation	Valid value
JEDEC Standard Hardware Reset or Hardware Reset (RESET Pin) while in Ultra-Deep Power-Down mode (after Ultra-Deep Power-Down (79h) command)	Undefined (1)
JEDEC Standard Hardware Reset or Hardware Reset (RESET Pin) (not in AUDPD or UDPD) or Reset Enable (66h) and Reset (99h)	Undefined (2)
Terminate Operation, (F0h) while a Program or Erase operation is in progress	Undefined (2) (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 logical “0” state, the device will not accept any program, erase, Protect Sector, Unprotect Sector, or Write Status Register commands. The WEL bit defaults to the logical “0” state after a device power-up or reset. In addition, the WEL bit will be reset to the logical “0” state automatically under the following conditions:

- Write Disable operation completes successfully
- Write Status Register operation completes successfully or aborts
- Protect Sector operation completes successfully or aborts
- Unprotect Sector operation completes successfully or aborts
- Byte/Page Program operation completes successfully or aborts
- Block Erase operation completes successfully or aborts
- Chip Erase operation completes successfully or aborts

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

11.1.7 $\overline{\text{RDY}}$/BSY Bit

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

11.2 Status Register Byte 2

Status Register Byte 2 contains the Status and Control Registers listed inTable 11-5. Detailed description of each bit follows below. Note that when reading Status Register Byte 2 while in Ultra-Deep Power-Down, all bits will read as 1.

Table 11-5. Status Register – Byte 2

Bit	Name		Type (1)		Description
7	$\overline{\text{STR}}$/DTR	Select STR or DTR mode	R/W	0	STR - Single Transfer Mode (Default)
				1	DTR - Dual Transfer 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*
4	TERE	Terminate Enabled	R/W	0	Terminate Operation command is disabled (default)
				1	Terminate Operation command is enabled
3	OME	Octal Mode Enable	R/W	11	Illegal combination - Reserved for future use
				10	Octal Mode Enabled
2	QPIE	QPI Mode Enable		01	QPI mode Enabled
				00	Standard SPI Mode Enabled (Default).
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 $\overline{\text{STR}}$/DTR Bit

The $\overline{\text{STR}}$/DTR Bit is used to determine if the device is running in Single Transfer Mode (STR) or Dual Transfer Mode (DTR). The default mode after power up is STR mode. See Section 5.4 for details about Dual Transfer Rate Operation.

11.2.2 AUDPD and ADPD Bits

The AUDPD and ADPD bits are used to enable the Auto Ultra-Deep Power-Down Mode and Auto Deep Power-Down Mode respectively. The default setting after power up is always 00, the device will 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.

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

ADPD will be cleared every time the device goes into DPD. The device will remain in the same communication mode when entering and exiting from DPD.

Note: Program Suspend and Erase Suspend commands cannot be used if 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 Logical 0 state (the default state after power-up), the Terminate Operation command is disabled and any attempts to reset the device using the Terminate Operation command will be ignored. When the TERE bit is in the Logical 1 state, the Terminate Operation command is enabled.

The TERE bit will retain its state as long as power is applied to the device. Once set to the Logical 1 state, the TERE bit will remain 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 Operation command itself will not change the state of the TERE bit.

11.2.4 OME and QPIE Bits

The OME and QPIE bits are used to enable the Octal Mode and QPI Mode respectively. The default setting after power up is always 00, so the device always wakes up in Standard SPI Mode. Only one of the OME and QPIE bits can be active at any given time, attempting to write both to 1 at the same time is an illegal combination.

See Section 5.2 for details about QPI Mode, and Section 5.3 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 QPI Mode and Octal Mode, the “Burst Read with Wrap (0Ch)” instruction is used to perform the Read operation with “Wrap Around” feature.

11.3.1 Wrap Length Bits

The Wrap Length is set by W6 and W5 in the Status/Control Register Byte 3. The Wrap Around mode is set by W7. The Wrap Length and Wrap mode set in either SPI, QPI 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 will operate in Continuous Mode. It will then read 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)” instruction will read 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 will then start 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 QPI or Octal modes, this bit is undefined.

11.3.3 Dummy Clock Bits

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

In QPI and Octal modes, to accommodate a wide range of applications with different needs for either maximum read frequency or minimum data access latency, the Status/Control Register Byte 3 can be used to configure the number of dummy clocks for “Fast Read (0Bh)”, “Read OTP Security Register (77h)” and “Burst Read with Wrap (0Ch)” instructions, and to configure the number of bytes of “Wrap Length” for the “Burst Read with Wrap (0Ch)” instruction.

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

The default “Wrap Length” after a power up or a Reset instruction 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)” instructions in the QPI or Octal modes.

The number of Dummy Clocks in Table 11-8 below refers to full clock cycles on the SCK pin, both for SDR and DTR 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(1)		Name	Type(2)		Description
7	W7		R/W		Used to set wrap length. See Table 11-7, Wrap bit functions. Default value is 000.
6	W6		R/W		
5	W5		R/W		
4	WPP	Write Protect ($\overline{WP}$) Pin Status	R	0	$\overline{WP}$ is asserted.
				1	$\overline{WP}$ is deasserted.
3	P3		R/W		Used to set number of dummy clock cycles. See Table 11-8, Dummy Clock cycles. Default value is 0111.
2	P2		R/W		
1	P1		R/W		
0	P0		R/W		

Table 11-7. Wrap bit functions

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

Table 11-7. Wrap bit functions

W7	W6	W5	Wrap Length	Wrap Around
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

					Maximum Frequency
P3	P2	P1	P0	Dummy Clocks	Any mode
0	0	0	0	8	66
0	0	0	1	10	80
0	0	1	0	12	100
0	0	1	1	14	120
0	1	0	0	16	133
0	1	0	1	18	133
0	1	1	0	20	133
0	1	1	1	22 (Default)	133
1	x	x	x	*Reserved for future use*	

11.4 I/O Pin Drive Strength Control Register

The Non-Volatile I/O pin Drive Strength Control Register is used to control the driver strength and the nominal impedance of the I/O pins. Choice of driver strength will depend 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 Drive Strength Control Register

Bit	Name		Type [1]	Description	
7	RES	*Reserved for future use*	R	0	*Reserved for future use*
6	RES	*Reserved for future use*	R	0	*Reserved for future use*
5	RES	*Reserved for future use*	R	0	*Reserved for future use*
4	RES	*Reserved for future use*	R	0	*Reserved for future use*
3	RES	*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

IOD2	IOD1	IOD0	Driver Type Value	Nominal Impedance	Approximated 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/Control Registers (65h)

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

The Status and 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 and Control Registers, the $\overline{CS}$ pin must first be asserted and the opcode of 65h must be clocked into the device. After the opcode 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 the QPI and Octal modes, 4 dummy cycles are used. (3.5 dummy cycles for Octal DTR.)

After the address byte 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. If Status Register Byte 1 is read first, it will be directly followed by Status Register Byte 2 and so on. The output values for addresses higher than 3 are not defined in this device.

Note that for Octal DTR mode, Status Register Bytes will be output on both edges of the SCK pulse. When reading just a single register (on the rising SCK edge), the next register location will also be read out (on the falling SCK edge). When reading three registers, the following register location will also be 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 will be undefined.

Deasserting the $\overline{CS}$ pin will terminate 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

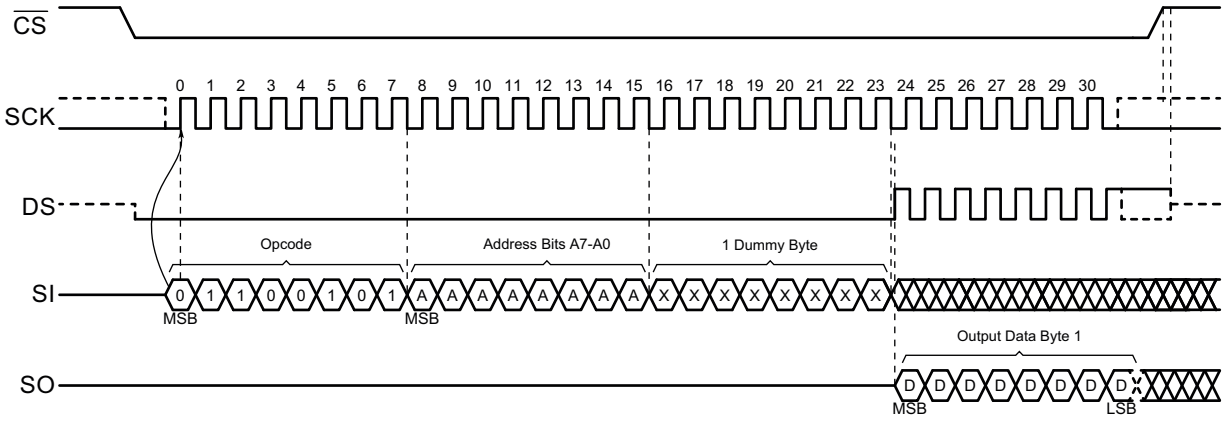


Figure 11-2. Read Status/Control Registers in QPI Mode - STR

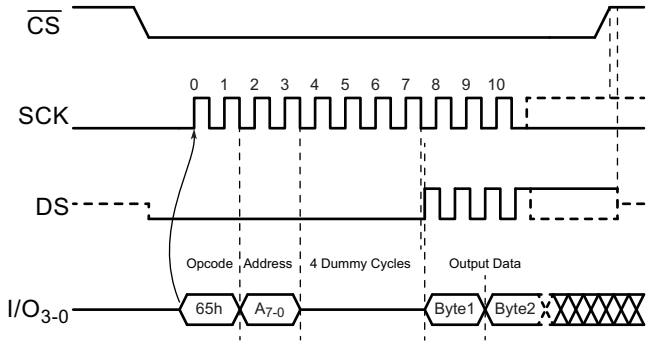


Figure 11-3. Read Status/Control Registers in Octal Mode - STR

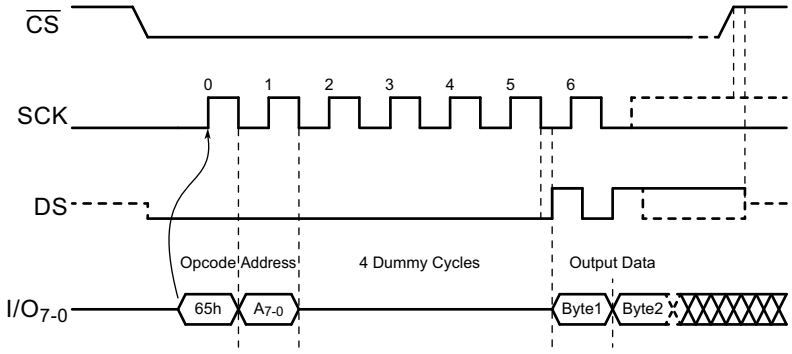


Figure 11-4. Read Status/Control Registers in QPI Mode - DTR

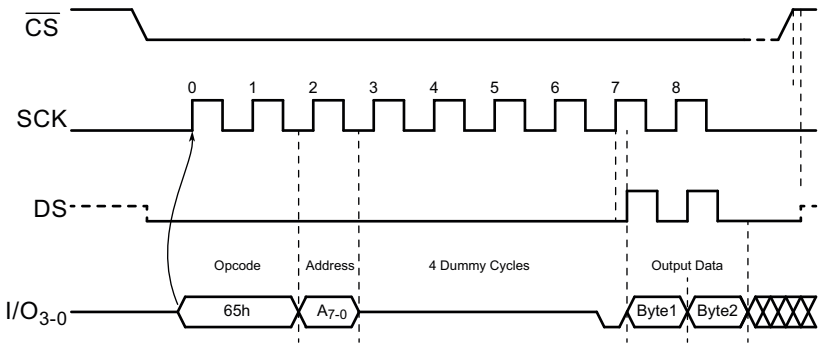
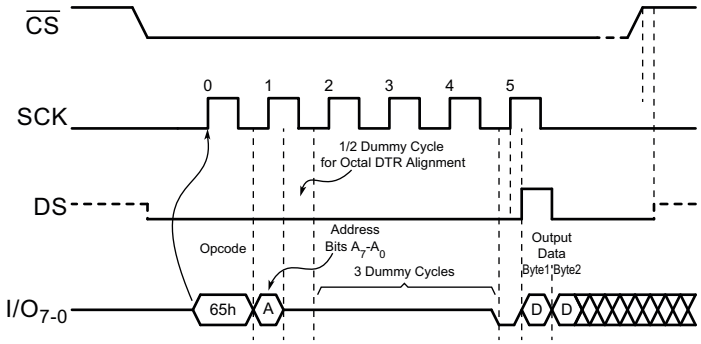


Figure 11-5. Read Status/Control Registers in Octal Mode - DTR



11.6 Read Status Register Byte 1 (05h)

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

To read Status Register Byte 1, the $\overline{CS}$ pin must first be asserted and the opcode of 05h must be clocked into the device. In QPI and Octal mode, 4 dummy cycles have to be clocked into the device. After the opcode and dummy cycles have been clocked in, the device will begin outputting Status Register data on the I/O pins during every subsequent clock cycle. After the last bit (bit 0) of Status Register Byte 1 has been clocked out, the sequence will repeat itself, starting again with bit 7 of Status Register Byte 1, as long as the $\overline{CS}$ pin remains asserted and the clock pin is being pulsed. The data in the Status Register is constantly being updated, so each repeating sequence may output new data.

Deasserting the $\overline{CS}$ pin will terminate 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 will also work while in Deep Power-Down. The DPDS bit in Status Register 1 will read as 1 in this case, while the $\overline{RDY}$/BSY and the UDPDS bits will read as 0. This will indicate 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, the Read Read Status Register Byte 1 (05h) command is only supported in Standard SPI Mode. All bits will read as 1 in this mode. Specifically, the UDPDS bit will read 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 11-6. Read Status Register Byte 1

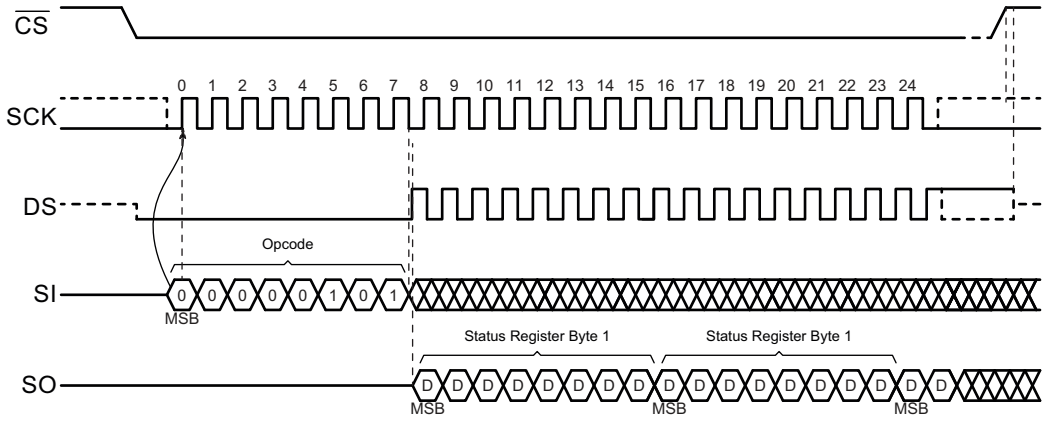


Figure 11-7. Read Status Register Byte 1 in QPI Mode - STR

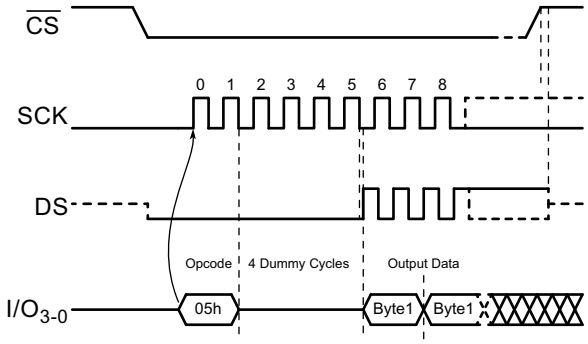


Figure 11-8. Read Status Register Byte 1 in Octal Mode - STR

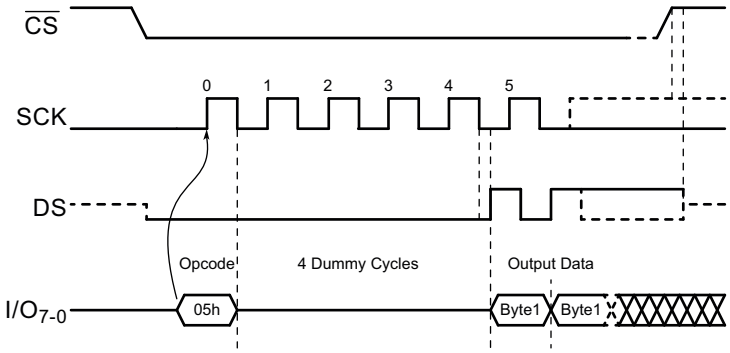


Figure 11-9. Read Status Register Byte 1 in QPI Mode - DTR

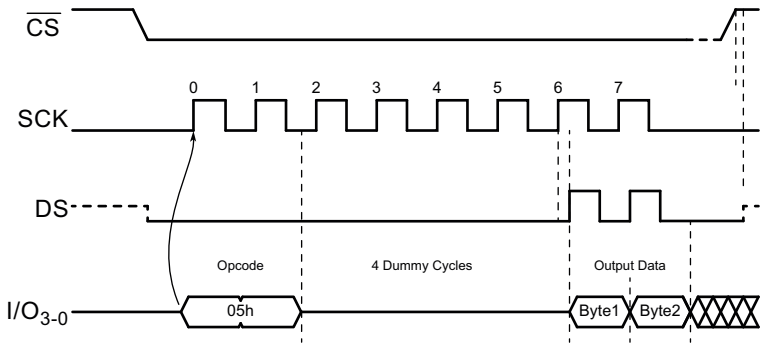
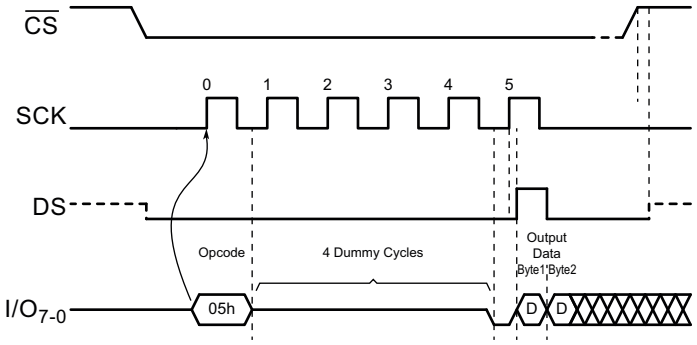


Figure 11-10.Read Status Register Byte 1 in Octal Mode - DTR



11.7 Write Status/Control Registers (71h)

All status and control registers may be written by using the generic Write Status/Control Registers command.

This command allows writing to all Volatile Status and Control Registers in a single operation for Standard SPI Mode, QPI STR mode, QPI DTR mode, and for Octal STR mode.

When writing Volatile Status and Control Registers in Octal DTR mode, Volatile Status and Control Registers have to be written one at the time, they cannot all be written in one operation in this mode.

Non-Volatile Status and Control Registers have to 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 will only be 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 and 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/Control Registers command can be issued, the Write Enable command must have been previously issued to set the WEL bit in the Status Register to a logical “1”.

To issue the Write Status/Control Registers command, the $\overline{CS}$ pin must first be asserted and the opcode of 71h must be clocked into the device. After the opcode 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 will be modified, and the WEL bit in the Status Register will be reset back to a logical “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 will abort the operation and no data will be programmed into the registers, and the WEL bit in the Status Register will be reset back to the logical “0” state.

When writing Volatile Status and Control Registers in Octal DTR mode, the address given has to be an even number (Address bit A0=0), and an even number of registers will be written.

When writing Non-Volatile Status and Control Registers in Octal DTR mode, the address given can be odd or even, and a single register will be written.

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

Note: No other commands should be issued while this operation is in progress. As changes to the Status and 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 11-11.Write Status/Control Registers

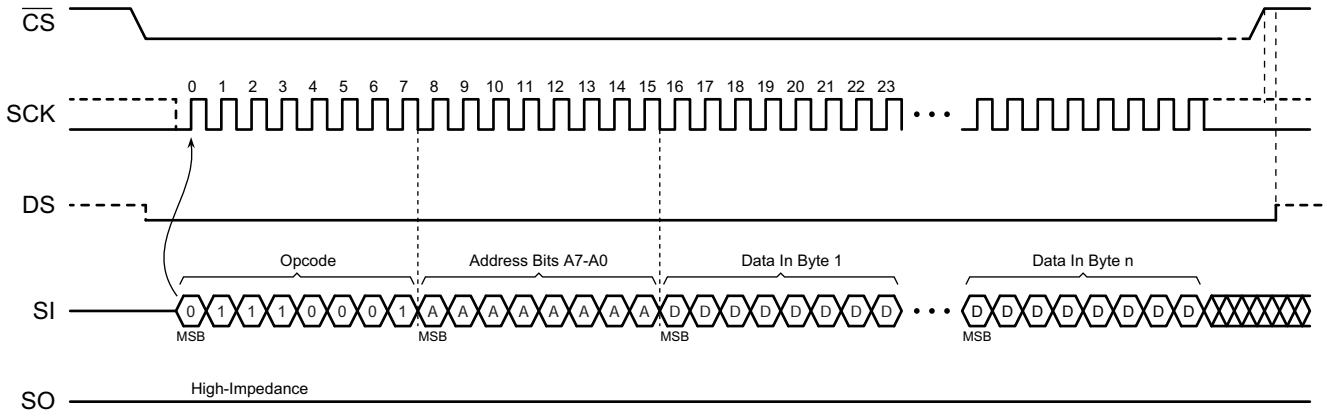


Figure 11-12.Write Status/Control Registers in QPI Mode - STR

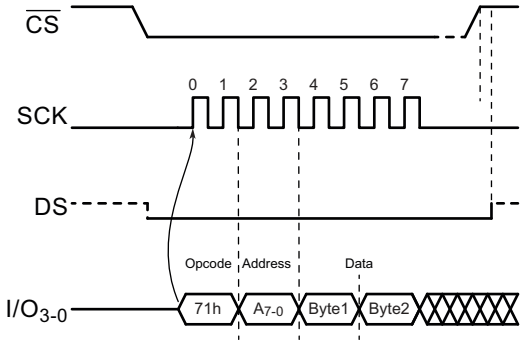


Figure 11-13.Write Status/Control Registers in Octal Mode - STR

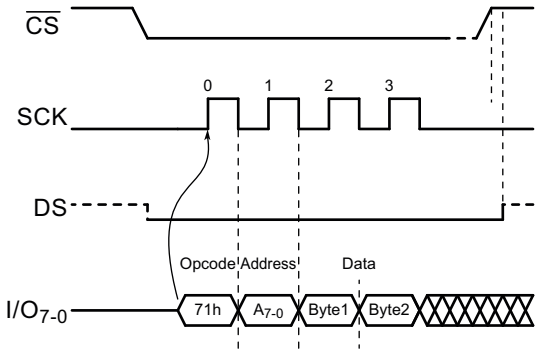


Figure 11-14.Write Status/Control Registers in QPI Mode - DTR

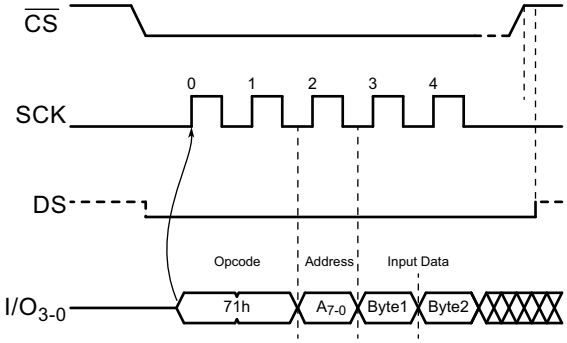
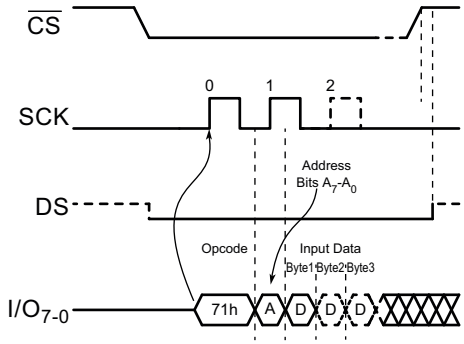


Figure 11-15.Write Status/Control Registers in Octal Mode - DTR



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 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 Register to a logical “1”.

To issue the Write Status Register command, the $\overline{CS}$ pin must first be asserted and the opcode of 01h must be clocked into the device followed by one byte of data. The one byte of data consists of the SPRL bit value, a don’t care bit, four data bits to denote whether a Global Protect or Unprotect should be performed, and two additional don’t care bits (see Table 11-11). Any additional data bytes that are sent to the device will be ignored. When the $\overline{CS}$ pin is deasserted, the SPRL bit in the Status Register will be modified, and the WEL bit in the Status Register will be reset back to a logical “0”. The values of bits 5, 4, 3, and 2 and the state

of the SPRL bit before the Write Status Register command was executed (the prior state of the SPRL bit) will determine whether or not a Global Protect or Global Unprotect will be performed. See section 9.5, Global Protect/Unprotect on page 56 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 will abort the operation, the state of the SPRL bit will not change, no potential Global Protect or Unprotect will be performed, and the WEL bit in the Status Register will be reset back to the logical “0” state.

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

Table 11-11. 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 11-16.Write Status Register

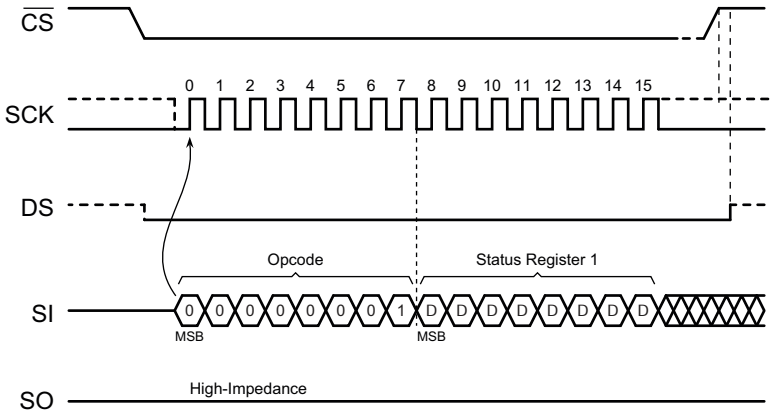


Figure 11-17.Write Status Register in QPI Mode - STR

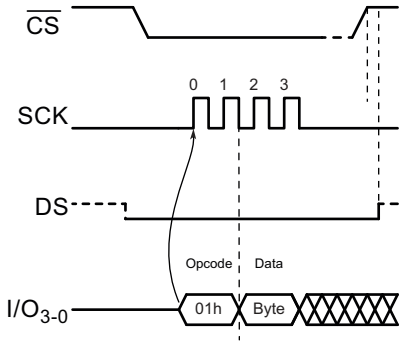


Figure 11-18.Write Status Register in Octal Mode - STR

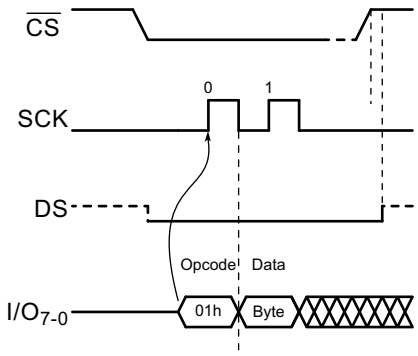


Figure 11-19.Write Status Register in QPI Mode - DTR

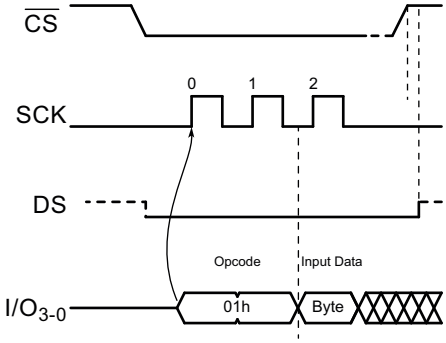
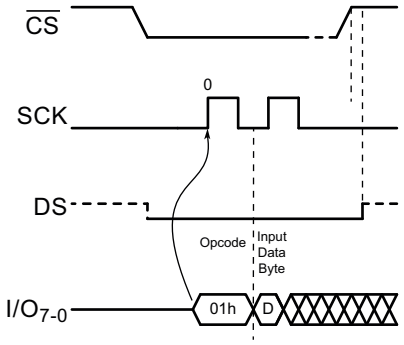


Figure 11-20.Write Status Register in Octal Mode - DTR



11.9 Write Status Register Byte 2 (31h)

The Write Status Register Byte 2 command is used to modify the Status Register Byte 2 bits. Using the Write Status Register Byte 2 command is the only way to modify the TERE and $\overline{\text{STR}}$/DTR bits in Status Register Byte 2 during normal device operation. The Standard/QPI/Octal Select bits may also be changed by using the Enter QPI Mode (38h), 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 Register to a Logical 1.

To issue the Write Status Register Byte 2 command, the $\overline{\text{CS}}$ pin must first be asserted and then the opcode 31h must be clocked into the device followed by one byte of data. The one byte of data consists of the bits described inTable 11-5. Any additional data

bytes sent to the device will be ignored. When the $\overline{CS}$ pin is deasserted, the TERE bit in the Status Register will be modified, and the WEL bit in the Status Register will be reset back to a Logical 0.

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

Figure 11-21.Write Status Register Byte 2

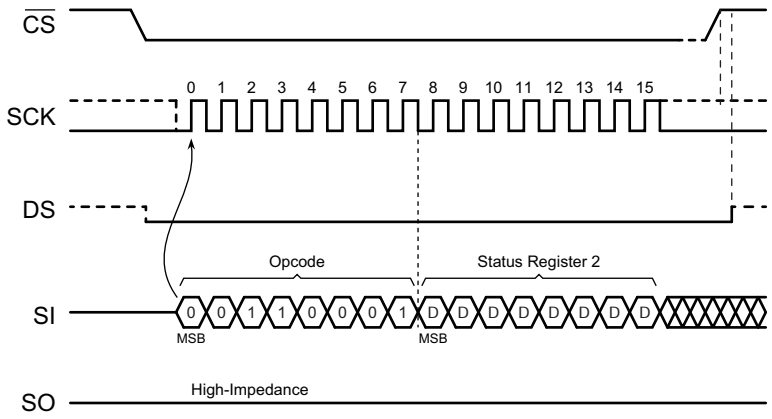


Figure 11-22.Write Status Register Byte 2 in QPI Mode - STR

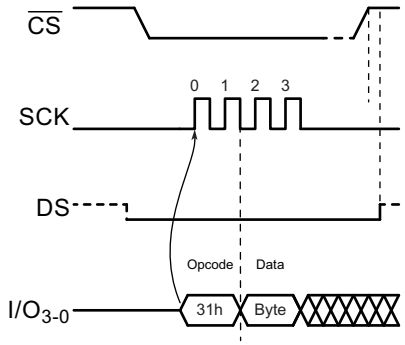


Figure 11-23.Write Status Register Byte 2 in Octal Mode - STR

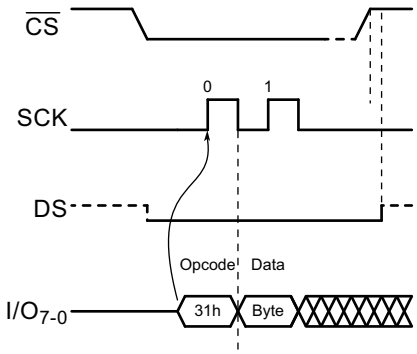


Figure 11-24.Write Status Register Byte 2 in QPI Mode - DTR

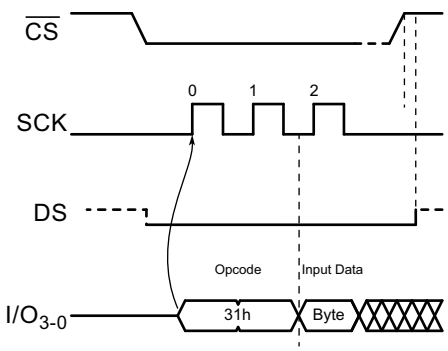
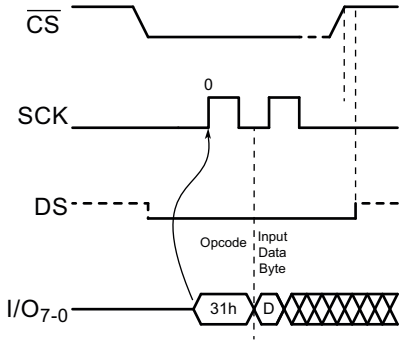


Figure 11-25.Write Status Register Byte 2 in Octal Mode - DTR



11.10 Active Status Interrupt (25h)

To simplify the readout of the RDY/BSY bit, the Active Status Interrupt command (25h) may be used. It is then not necessary to continuously read the status 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 ATXP064 is ready for the next command.

The 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 CS pin must first be asserted and the opcode of 25h must be clocked into the device. For SPI Mode3, at least one dummy bit has to be clocked into the device after the last bit of the opcode 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 opcode is clocked in is of no significance to the operation. For SPI Mode 0, this dummy bit (dummy byte) is not required. For QPI and Octal modes, 4 dummy cycles are required.

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

Additional clocks on the SCK pin are not required. If the RDY/BSY bit changes from 1 to 0 while the CS pin is asserted, the SO line will change from 1 to 0. (The 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 CS pin will terminate the Active Status Interrupt operation and put the SO pin into a high-impedance state. The CS pin can be deasserted at any time and does not require that a full byte of data be read.

Note: When operating in QPI or Octal DTR 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 will add additional load on that one line, adding a delay on this line compared to the others.

Note: Active Status Interrupt 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 RDY/BSY bit by polling Status Register Byte 1 from time to time.

Figure 11-26.Active Status Interrupt

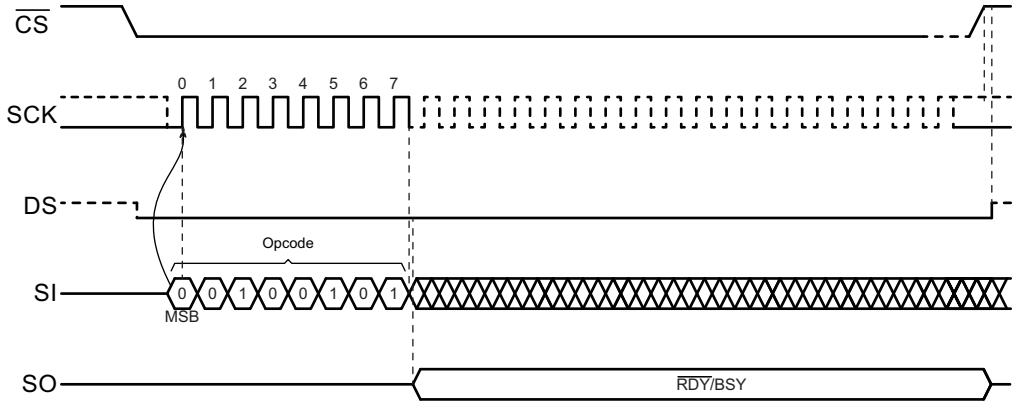


Figure 11-27.Active Status Interrupt in QPI Mode - STR

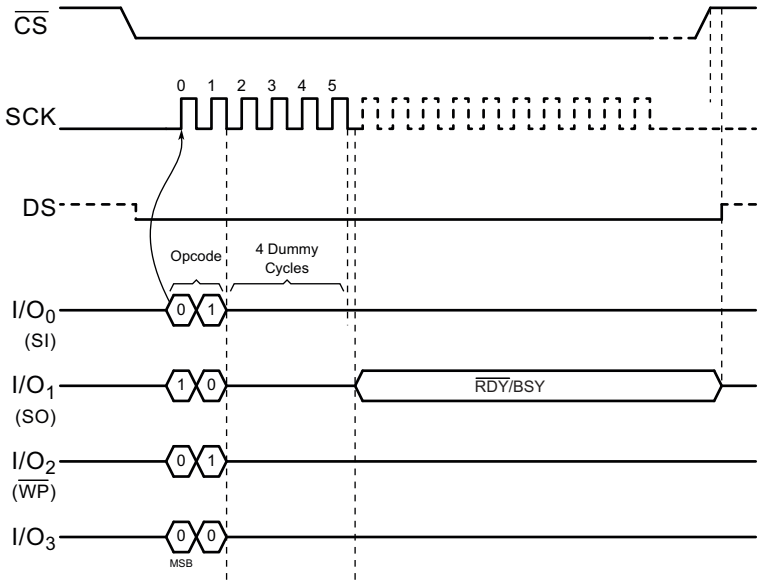


Figure 11-28.Active Status Interrupt in Octal Mode - STR

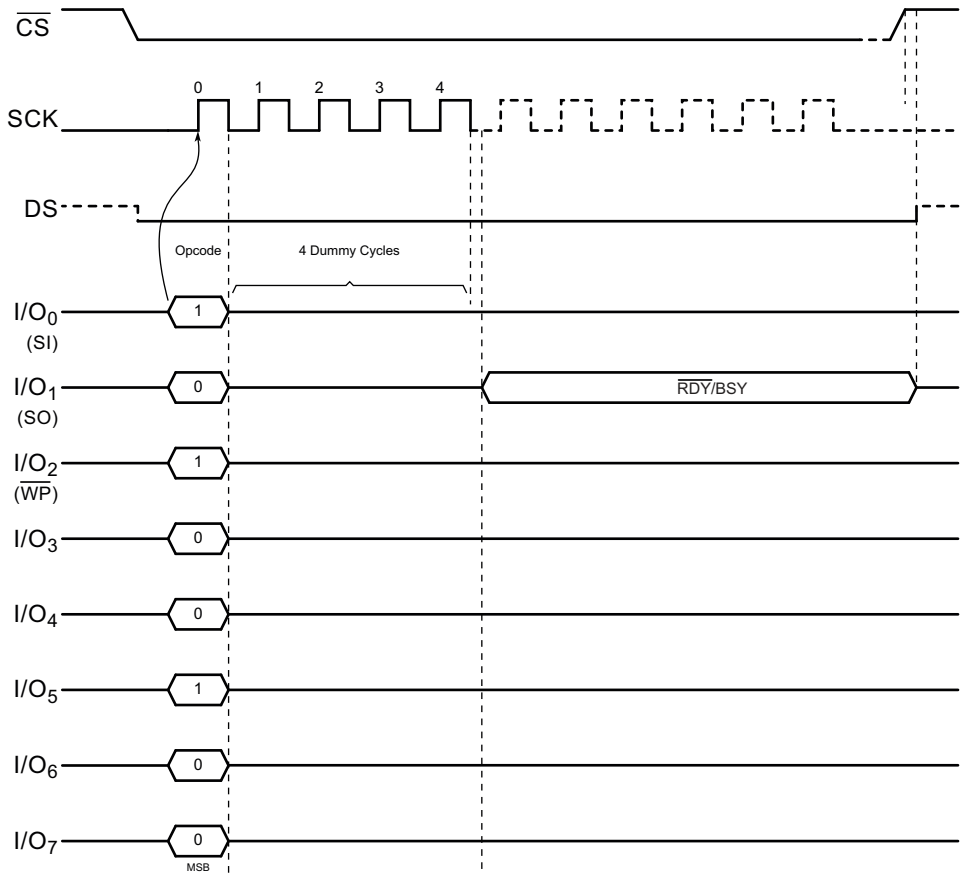


Figure 11-29.Active Status Interrupt in QPI Mode - DTR

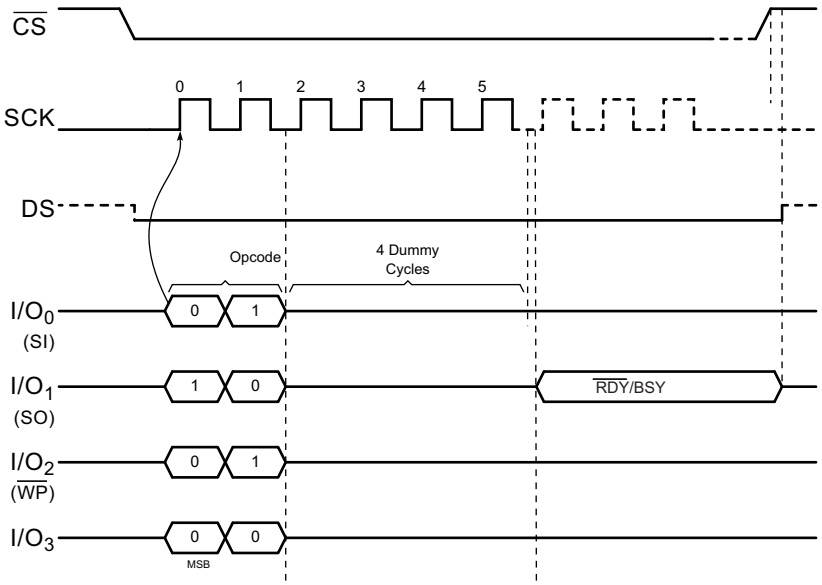
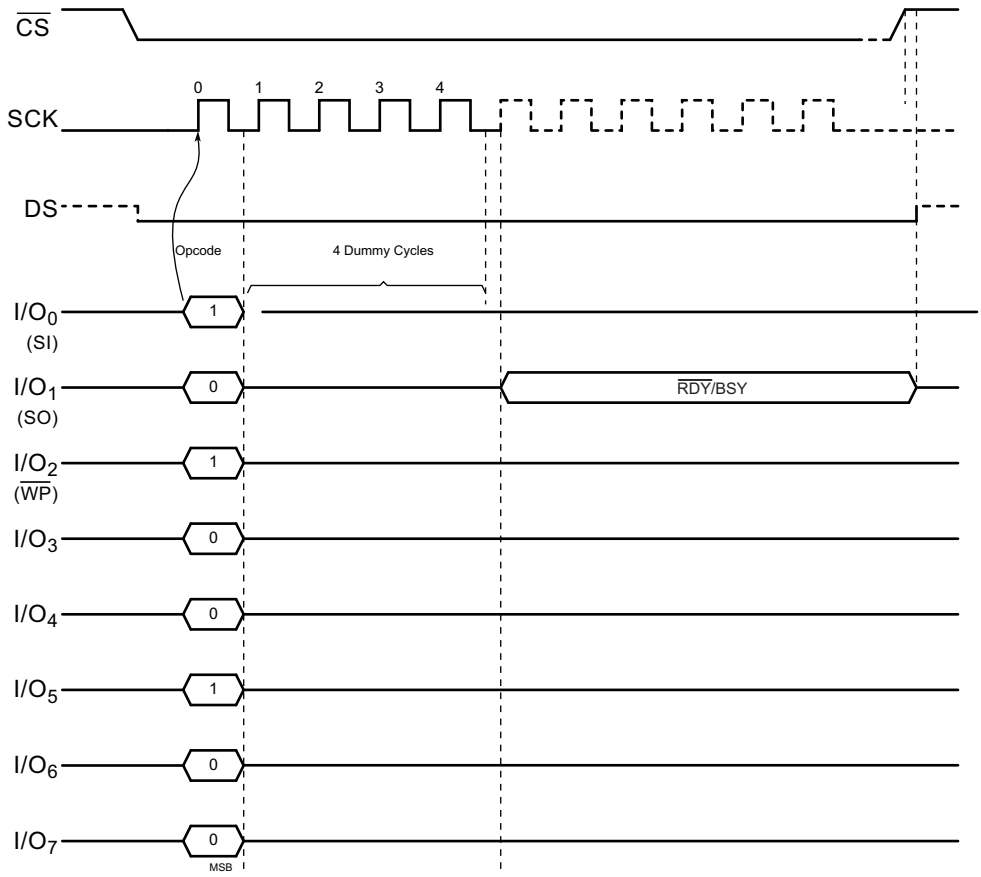


Figure 11-30.Active Status Interrupt in Octal Mode - DTR



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 opcode comply with the JEDEC standard for “Manufacturer and Device ID Read Methodology for SPI Compatible Serial Interface Memory Devices”. The type of information that can be read from the device includes the JEDEC-defined Manufacturer ID, the vendor-specific Device ID, and the vendor-specific Extended Device Information.

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

To read the identification information, the $\overline{CS}$ pin must first be asserted and then the opcode 9Fh must be clocked into the device. After the opcode has been clocked in, the device begins outputting the identification data on the SO pin during the subsequent clock cycles. The first byte to be output is the Manufacturer ID, followed by two bytes of the Device ID information. The fourth byte output is the Extended Device Information (EDI) String Length, which is 01h, indicating that one byte of EDI data follows. After the one byte of EDI data is output, the SO pin is placed into 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 EDI String Length and any subsequent data is optional.

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

Table 12-1. Manufacturer and Device ID Information

Byte No.	Data Type	Value
1	Manufacturer ID	1Fh
2	Device ID (Byte 1)	A9h
3	Device ID (Byte 2)	00h
4	Extended Device Information (EDI) String Length	01h
5	[Optional to Read] EDI Byte 1	00h

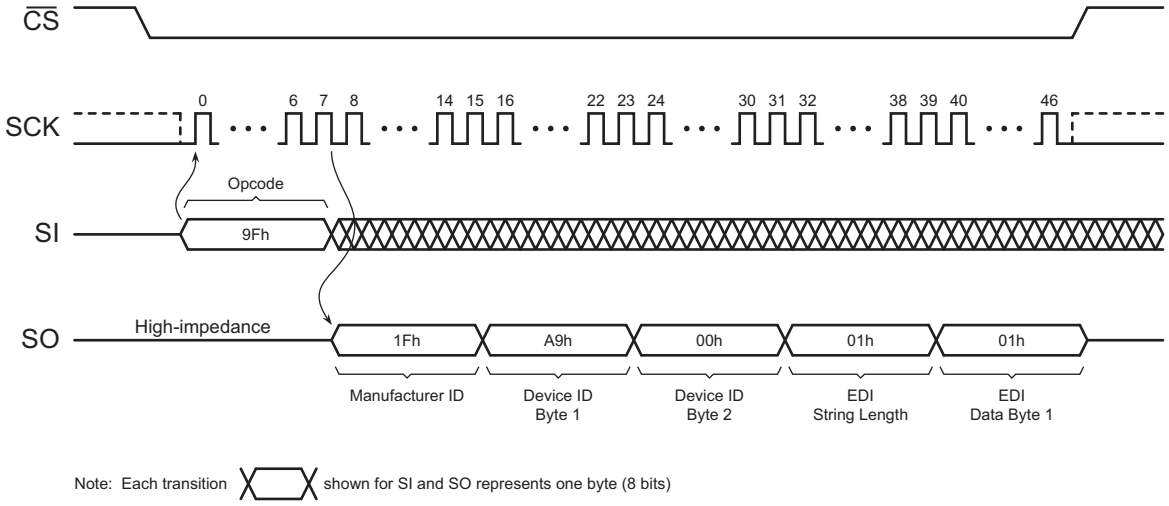
Table 12-2. Manufacturer and Device ID Details

Data Type	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Hex Value	Details
Manufacturer ID	JEDEC Assigned Code								1Fh	JEDEC code: 0001 1111 (1Fh for Adesto)
	0	0	0	1	1	1	1	1		
Device ID (Byte 1)	Family Code			Density Code					A8h	Family code: 001 (ATXPxxx series) Density code: 01000 (64-Mbit)
	1	0	1	0	1	0	0	0		
Device ID (Byte 2)	Sub Code			Product Version Code					00h	Sub code: 000 (Standard Series) Product variant: 00000
	0	0	0	0	0	0	0	0		

Table 12-3. EDI Data

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

Figure 12-1. Read Manufacturer and Device ID



12.2 Deep Power-Down (B9h)

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

When the device is in the Deep Power-Down mode, all commands will be 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 will be ignored, the mode can be used as an extra protection mechanism against program and erase operations.

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

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

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

Figure 12-2. Deep Power-Down

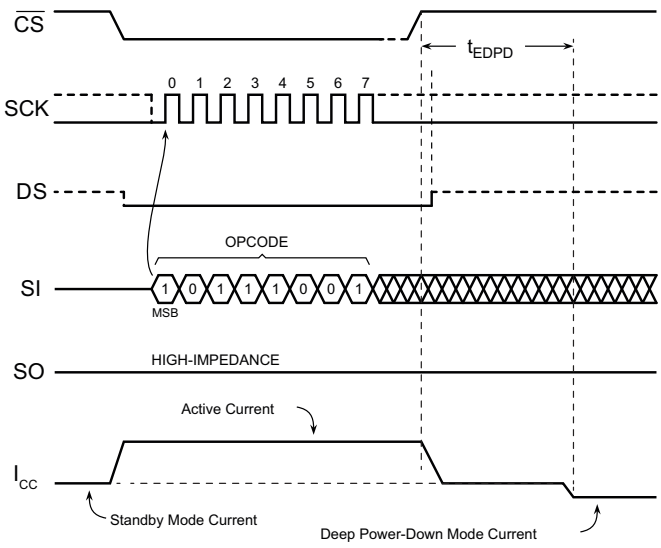


Figure 12-3. Deep Power-Down in QPI Mode - STR

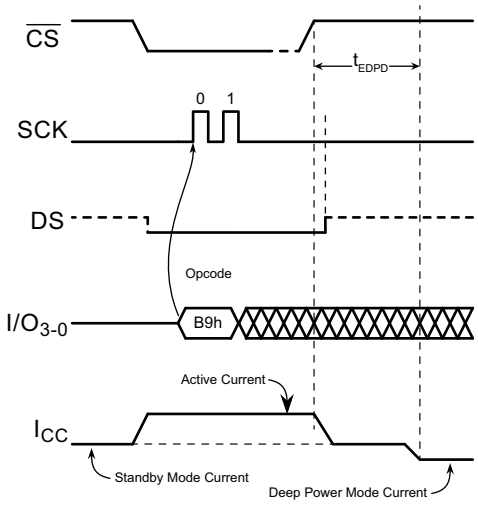


Figure 12-4. Deep Power-Down in Octal Mode - STR

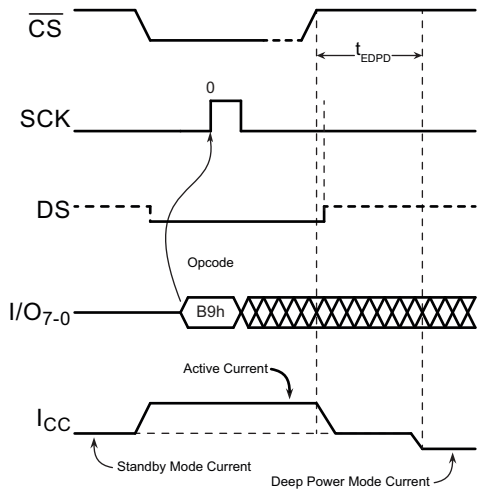


Figure 12-5. Deep Power-Down in QPI Mode - DTR

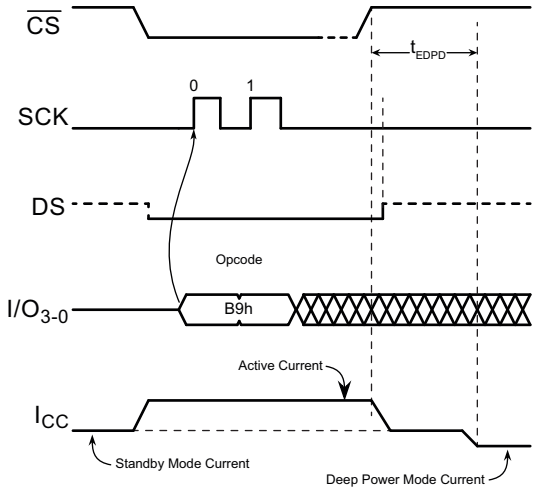
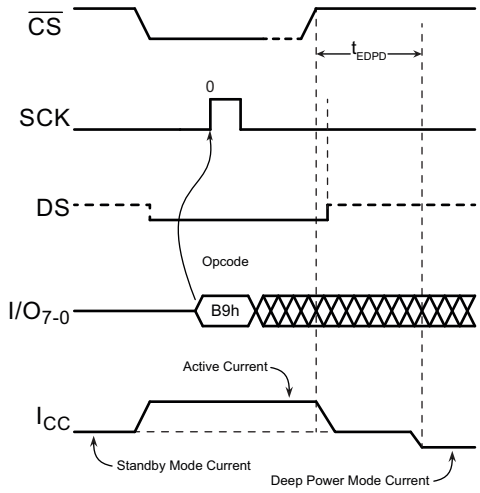


Figure 12-6. Deep Power-Down in Octal Mode - DTR



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- 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 Register Byte 2. The device will remain 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 or Ultra-Deep Power-Down (ABh) 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 will recognized while in the Deep Power-Down mode.

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 Deep Power-Down mode, the $\overline{CS}$ pin must first be asserted and opcode of ABh must be clocked into the device. Any additional data clocked into the device after the opcode will be ignored. When the $\overline{CS}$ pin is deasserted, the device will exit the Deep Power-Down mode within the maximum time of t_{RDPD} and return to the standby mode. After 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 Register 1.

To resume from the Ultra-Deep Power-Down mode, the $\overline{CS}$ pin must first be asserted and opcode of ABh must be clocked into the device. Any additional data clocked into the device after the opcode will be ignored. When the $\overline{CS}$ pin is deasserted, the device will exit the 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 Register 1.

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

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

The device will remain in the same mode (SPI, QPI or Octal Mode) while in Deep Power-Down and will wake up in the same mode it had before entering Deep Power-Down.

Figure 12-7. Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh)

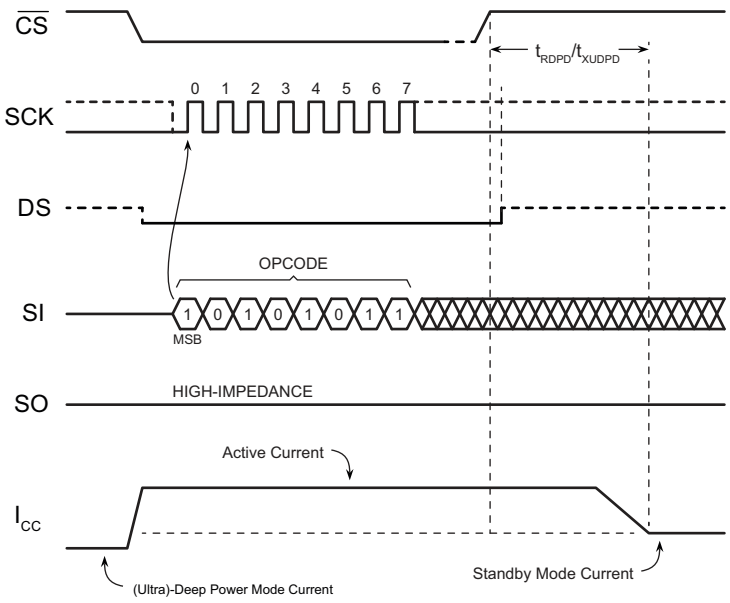


Figure 12-8. Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh) in QPI Mode - STR

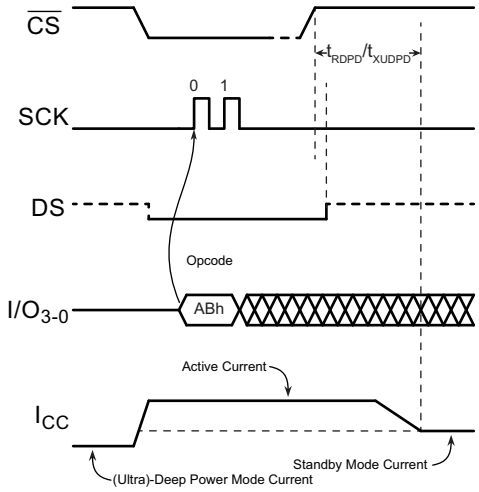


Figure 12-9. Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh) in Octal Mode - STR

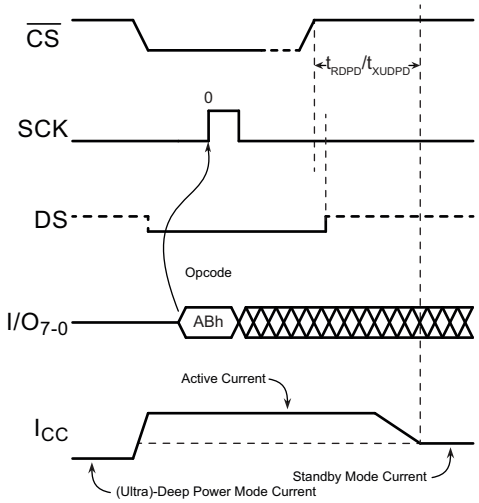


Figure 12-10.Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh) in QPI Mode - DTR

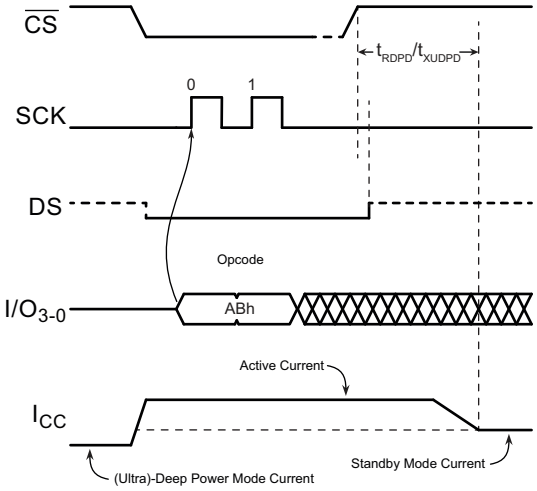
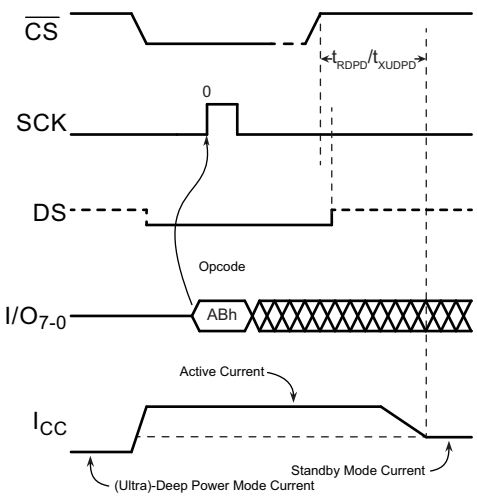


Figure 12-11.Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh) in Octal Mode - DTR



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 will recognize. When reading the Status Registers in this mode, all bits will read as “1”, indicating that the device is in UDPD mode. All other commands, including the Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh) command, will be ignored. Since all write commands will be 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 opcode 79h, and then deasserting the $\overline{CS}$ pin. Any additional data clocked into the device after the opcode will be ignored. When the $\overline{CS}$ pin is deasserted, the device will enter the Ultra-Deep Power-Down mode within the maximum time of t_{EUDPD}

The complete opcode must be clocked in before the $\overline{CS}$ pin is deasserted; otherwise, the device will abort the operation and return to the standby mode once the $\overline{CS}$ pin is deasserted. In addition, the device will default to the standby mode after a power cycle. The Ultra-Deep Power-Down command will be 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.

Upon recovery from Ultra-Deep Power-Down, all internal registers except the EPE bit in Status Register Byte 1 will be 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 Ultra-Deep Power-Down.

The device will wake up in SPI Mode even if it was in QPI or Octal Mode when it entered Ultra-Deep Power-Down.

Figure 12-12.Ultra-Deep Power-Down

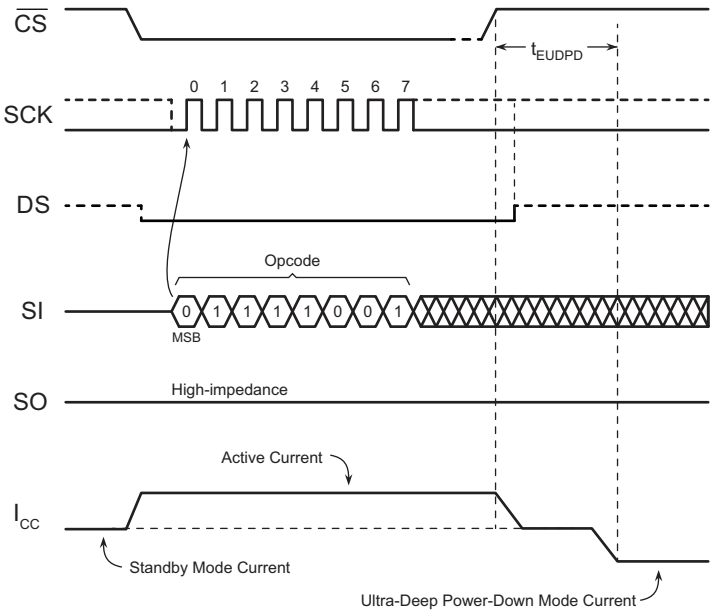


Figure 12-13.Ultra-Deep Power-Down in QPI Mode - STR

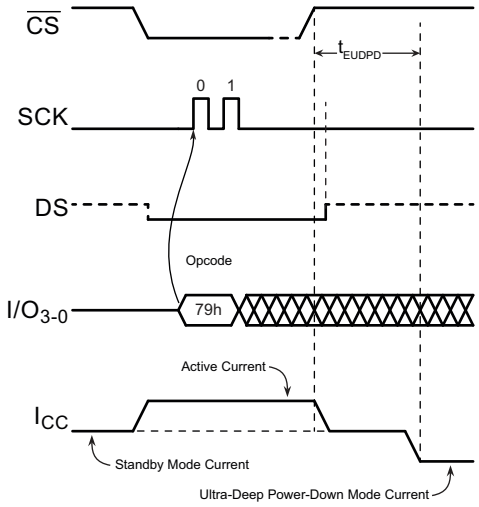


Figure 12-14.Ultra-Deep Power-Down in Octal Mode - STR

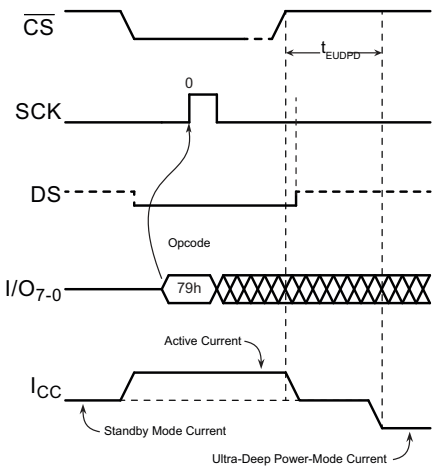


Figure 12-15.Ultra-Deep Power-Down in QPI Mode - DTR

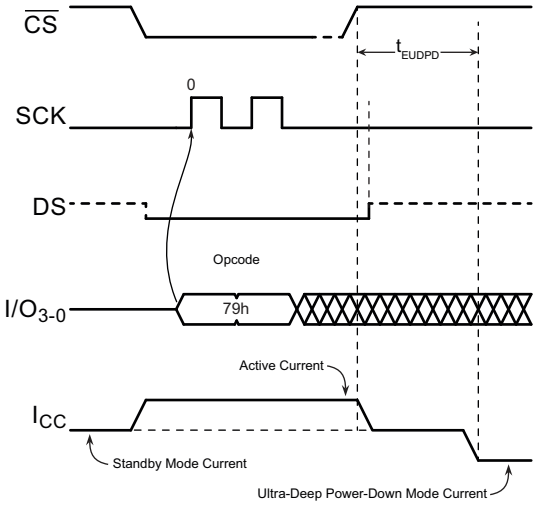
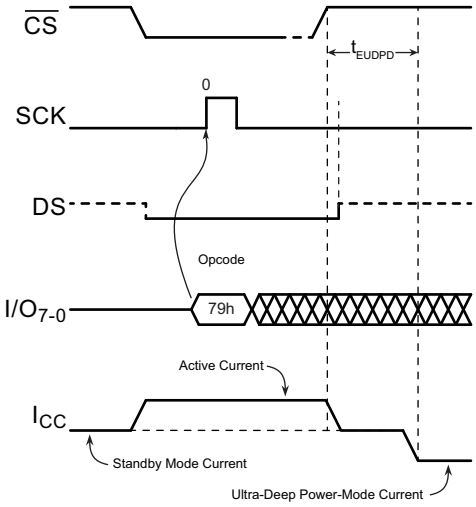


Figure 12-16.Ultra-Deep Power-Down in Octal Mode - DTR



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

The Auto Ultra-Deep Power-Down 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- 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 Ultra-Deep Power-Down Mode after Program/Erase Operation is enabled by setting the Auto Ultra-Deep Power-Down Enable bit in Status Register Byte 2. AUDPD will be cleared every time the device goes into UDPD, 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 Program/Erase Operation, the device will switch 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 Ultra-Deep Power-Down mode, the Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh) command, 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 Register Byte 1 will be 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 Ultra-Deep Power-Down.

The device will wake up in SPI Mode even if it was in QPI or Octal Mode when it entered Ultra-Deep Power-Down. 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 Ultra-Deep Power-Down, are not implemented on the ATXP064, as this would make Status Read while in Ultra-Deep Power-Down 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 wakeup 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 wakeup 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. See also See 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 Ultra-Deep Power-Down. 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 (RESET Pin). See also See Section 13.5, AC Characteristics – All Other Parameters, for timing details.

12.7.4 Power Cycling

The device can also exit the Ultra-Deep Power-Down Mode by power cycling the device. Note that the 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 Operation, (F0h)

In some applications, it may be necessary to prematurely terminate a program or erase operation rather than wait the hundreds of microseconds or milliseconds necessary for the program or erase operation to complete normally. The Terminate Operation 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 terminate the device is immediate, the Write Enable command does not need to be issued prior to the Terminate command. Therefore, the Terminate Operation command operates independently of the state of the WEL bit in the Status Register.

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

To perform a Terminate Operation, the $\overline{CS}$ pin must first be asserted, and then the opcode F0h must be clocked into the device. No address bytes need to be clocked in, but a confirmation byte of D0h must be clocked into the device immediately after the opcode. Any additional data clocked into the device after the confirmation byte will be ignored. When the $\overline{CS}$ pin is deasserted, the program operation currently in progress will be terminated within a time of t_{SWRST}. Since the program or erase operation may not complete before the device is Terminate, 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 Register. Apart from Sequential Programming, the WEL bit will be reset back to its default state.

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

Figure 12-17.Terminate Operation

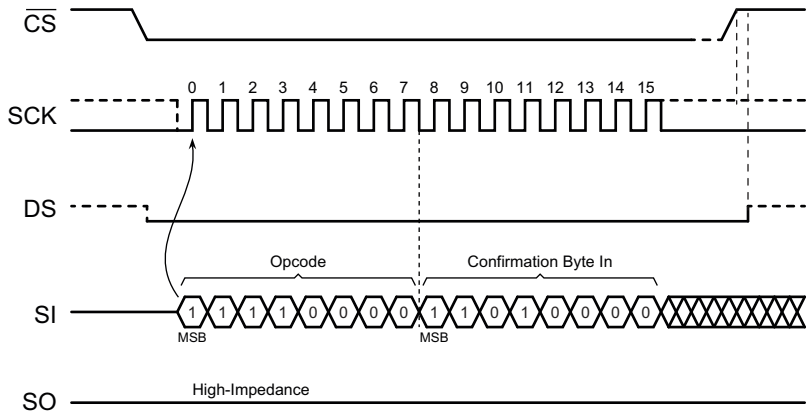


Figure 12-18.Terminate Operation in QPI Mode - STR

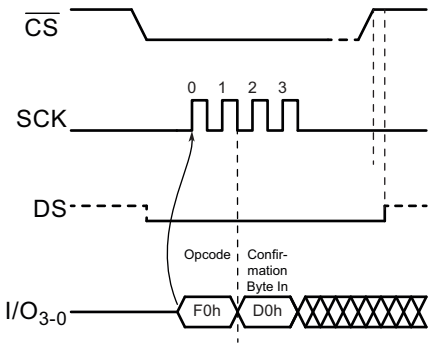


Figure 12-19.Terminate Operation in Octal Mode - STR

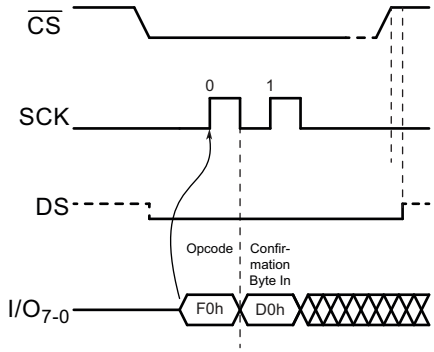


Figure 12-20.Terminate Operation in QPI Mode - DTR

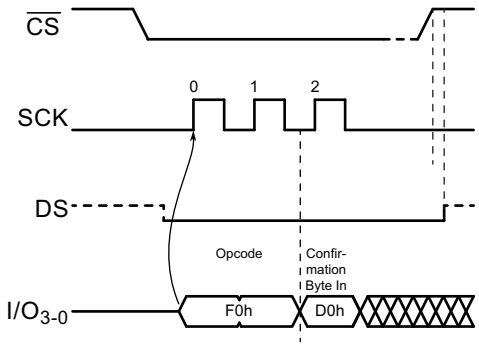
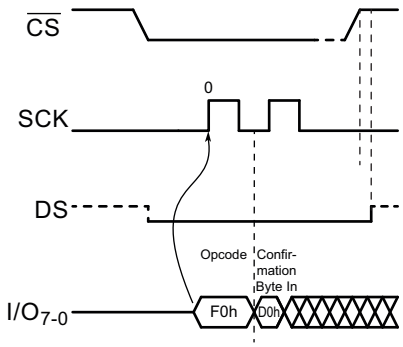


Figure 12-21.Terminate Operation in Octal Mode - DTR



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

The ATXP064 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 will return to its default power-on state and lose all the current volatile settings. All volatile status registers, including the Sector Protection Bits, will be reset to their default values. All non-volatile status registers will 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 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 ATXP064, the device will take approximately t_{SWRES} to reset. During this period, no instruction will be 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 device. It is recommended to check the $\overline{RDY}$/BSY bit in Status Register Byte 1 and the PS and ES bits in Status Register Byte 2 before issuing the Reset instruction sequence.

Figure 12-22. Reset Enable and Reset

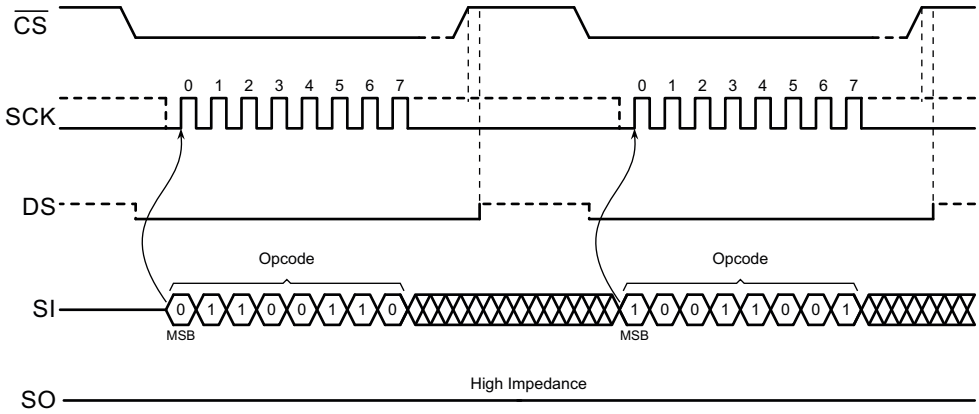


Figure 12-23. Reset Enable and Reset in QPI Mode - STR

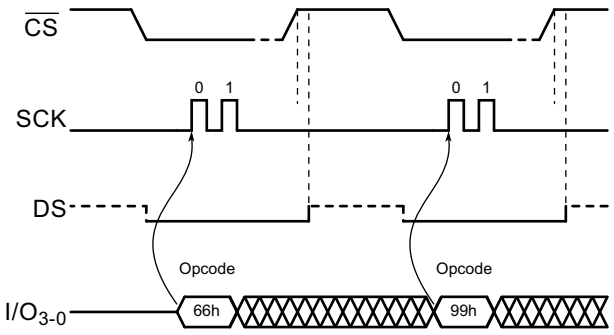


Figure 12-24. Reset Enable and Reset in Octal Mode - STR

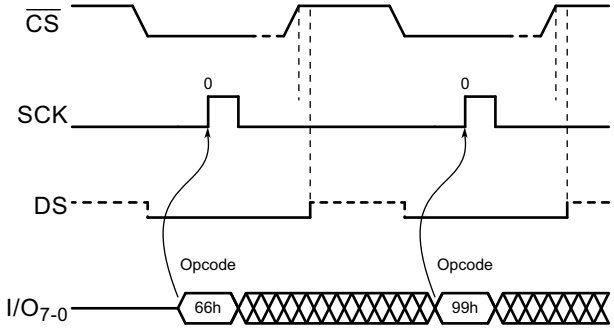


Figure 12-25. Reset Enable and Reset in QPI Mode - DTR

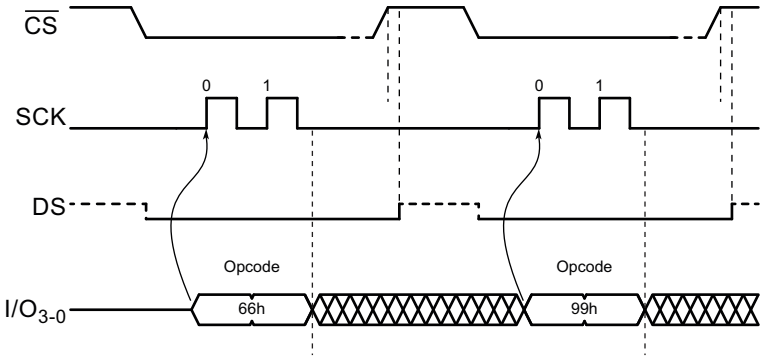
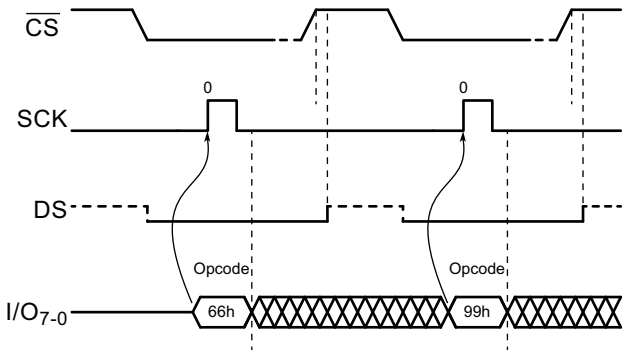


Figure 12-26. Reset Enable and Reset in Octal Mode - DTR



12.10 JEDEC Standard 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 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.

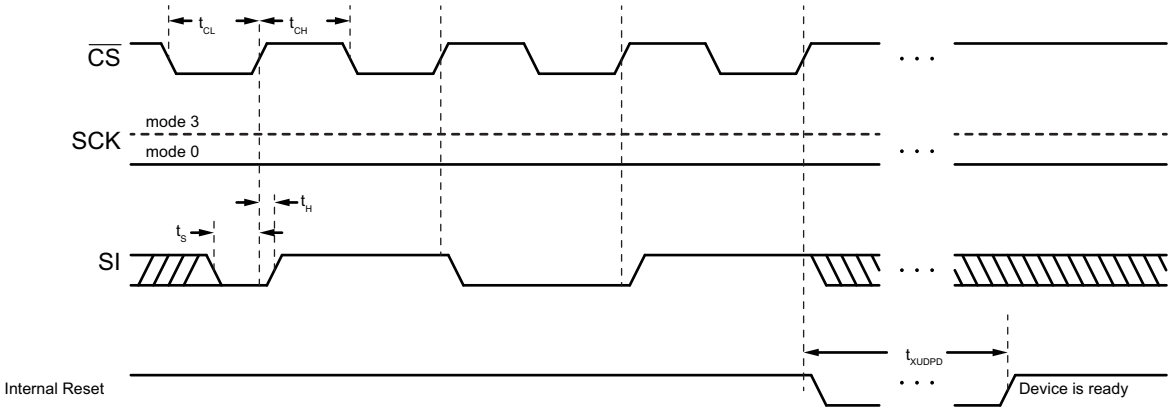
After a JEDEC Hardware Reset while the device is in Ultra-Deep Power-Down mode, the SRAM buffer will be reset to an undefined value. All volatile status registers, including the Sector Protection Bits, will be reset to their default values, except in the following case: If the AUDP bit in Status Register Byte 2 was set prior to the last Program or Erase command, so the device entered the Ultra-Deep Power-Down mode after a Program or Erase command finished, and the JEDEC Hardware Reset is used to wake up the device from Ultra-Deep Power-Down mode, then the EPE bit will not be reset. In this case the EPE will still show the correct status after the latest Program or Erase command. All non-volatile status registers will keep the value they had prior to reset.

After a JEDEC Hardware Reset 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 registers, including the Sector Protection Bits, will be reset to their default values. All non-volatile status registers will 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 register, the value of that register may be corrupted.

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

Figure 12-27 below illustrates the timing for the JEDEC Hardware Reset operation.

Figure 12-27. JEDEC Standard Hardware Reset



12.11 Hardware Reset ($\overline{\text{RESET}}$ Pin)

The Hardware Reset pin can be used to wake up the device from Ultra-Deep Power-Down. 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 ($\overline{\text{RESET}}$) will terminate the operation in progress and reset the internal state machine to an idle state. The device will remain in the reset condition as long as a low level is present on the $\overline{\text{RESET}}$ 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.

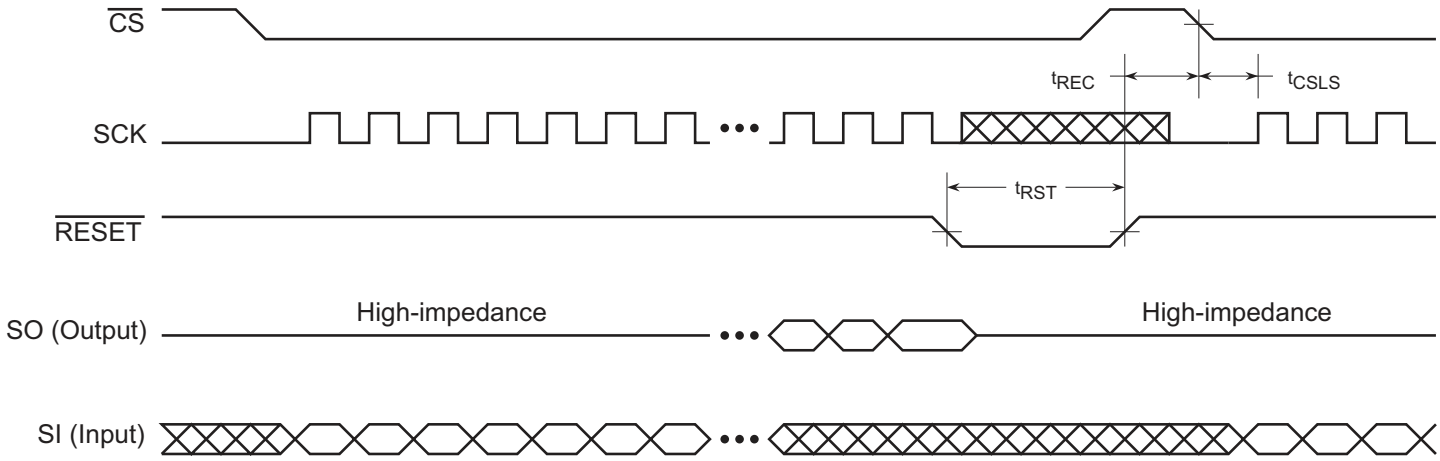
After a Hardware Reset initiated by the $\overline{\text{RESET}}$ pin while the device is in Ultra-Deep Power-Down mode, the SRAM buffer will be reset to an undefined value. All volatile status registers, including the Sector Protection Bits, will be reset to their default values, except in the following case: If the AUDP bit in Status Register Byte 2 was set prior to the last Program or Erase command, so the device entered the Ultra-Deep Power-Down mode after a Program or Erase command finished, and $\overline{\text{RESET}}$ is used to wake up the device from Ultra-Deep Power-Down mode, then the EPE bit will not be reset. In this case the EPE will still show the correct status after the latest Program or Erase command. All non-volatile status registers will 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 registers will be reset to their default values, except the Sector Protection bits. The Sector Protection bits will remain unchanged in this case. All non-volatile status registers will 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 register, the value of that register may be corrupted.

The device will always revert back to Standard SPI Mode after $\overline{\text{RESET}}$.

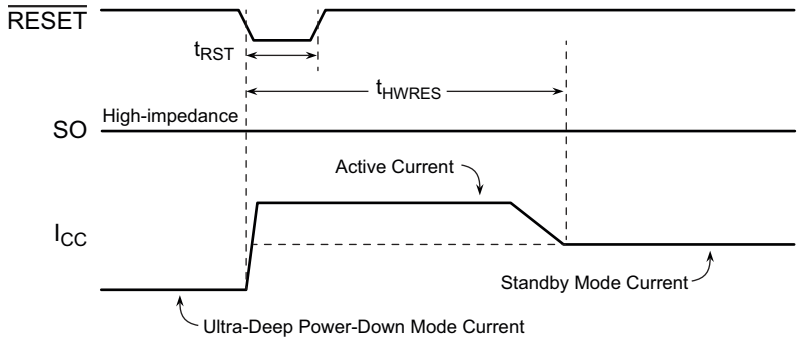
Figures 12-28 and 12-29 below illustrates the timing for the Hardware $\overline{\text{RESET}}$ sequence. See AC Characteristics – All Other Parameters for details on Reset timing.

Figure 12-28.Hardware Reset Sequence



Note: The CS signal should be in the high state before the RESET signal is deasserted.

Figure 12-29.Hardware Reset Sequence from UDPD



12.12 Reset Summary

The following table summarize the effects of various Reset functions:

Table 12-4. Effects of Reset Functions

Action	Volatile Registers	Non-Volatile Registers	Sector Protection Registers	EPE bit
Power Cycle	Default state	Unchanged (1)	Default state (protected)	See Table 11-4
JEDEC Standard Hardware Reset (2)	Default state	Unchanged(1)	Default state (protected)	See Table 11-4
Hardware Reset (RESET Pin) (not in UDPD) (3)	Default state	Unchanged(1)	Unchanged	See Table 11-4
Exit Ultra-Deep Power-Down (Power cycle, JEDEC Standard Hardware Reset, Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh))	Default state	Unchanged(1)	Default state (protected)	See Table 11-4
Terminate Operation, (F0h), Reset Enable (66h) and Reset (99h)	Default state	Unchanged(1)	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 will continue 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 QPI Mode, data byte values like F0h, 0Fh, A5h or 5Ah will create an alternating pulse train. In Octal Mode, the Echo command will return the same value on all lines continuously. To create alternating wave forms in Octal Mode, use the Echo with Inversion command.

Figure 12-30.Echo in QPI Mode - STR

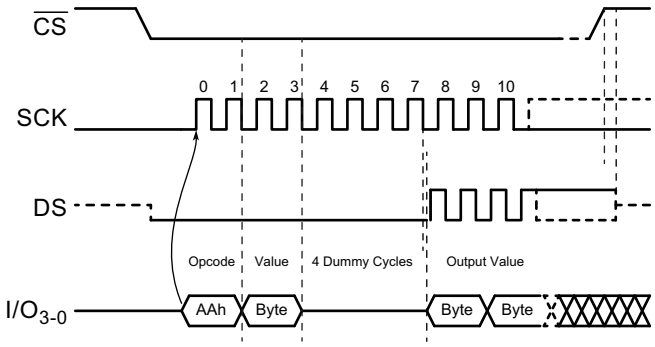


Figure 12-31.Echo in Octal Mode - STR

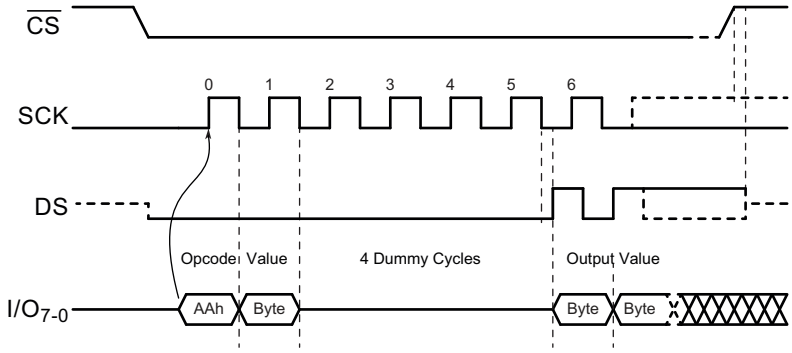


Figure 12-32.Echo in QPI Mode - DTR

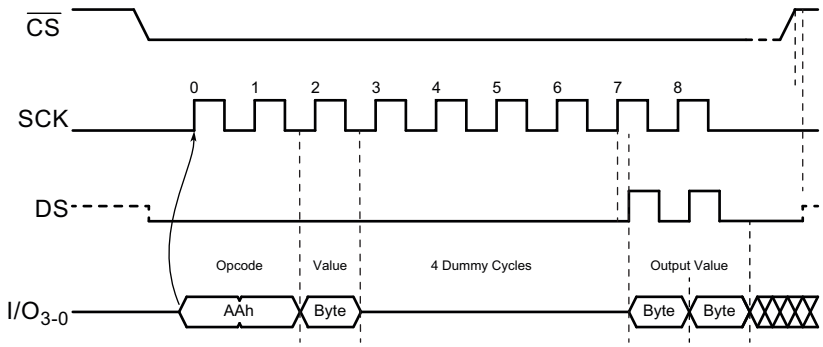
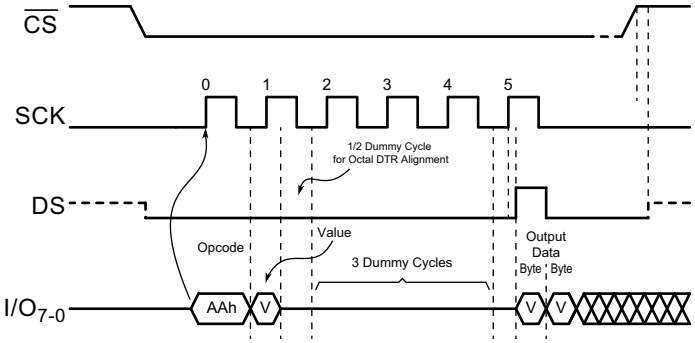


Figure 12-33.Echo in Octal Mode - DTR



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 will continue to output data as long as $\overline{CS}$ is held low and SCK is being clocked. It will alternate 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 will create an alternating pulse train. The value entered with the command will decide the value of the first byte returned, the next will be inverted and so on. So a value of 00h or FFh will cause all the I/O lines to have the same value and alternate between 00h and FFh, while a value of AAh or 55h will alternate between AAh and 55h.

his command is not supported in SPI Mode.

In QPI Mode, data byte values like F0h or 0Fh will create an alternating pulse train with a frequency of $f_{SCK}/2$. A value like 36h or 6Ch will create alternating pulse trains with a frequency of $f_{SCK}/2$ that are delayed one clock from one I/O pin to the next. The output will then be a quadrature encoded signal.

Figure 12-34.Echo with Inversion in QPI Mode - STR

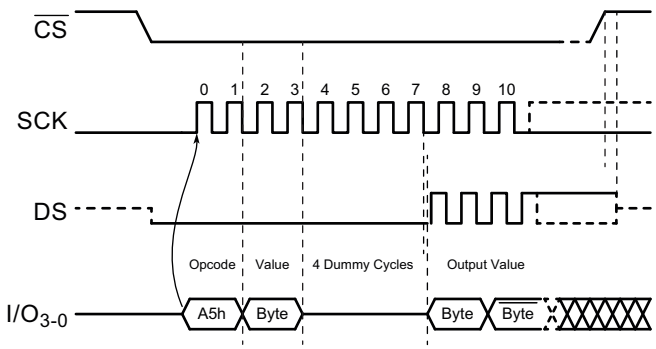


Figure 12-35.Echo with Inversion in Octal Mode - STR

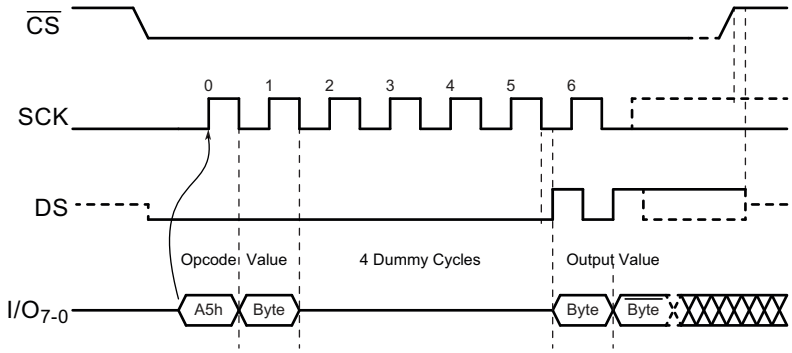


Figure 12-36.Echo with Inversion in QPI Mode - DTR

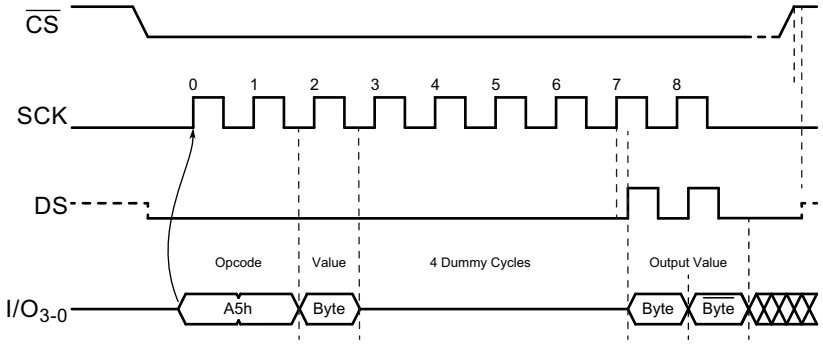
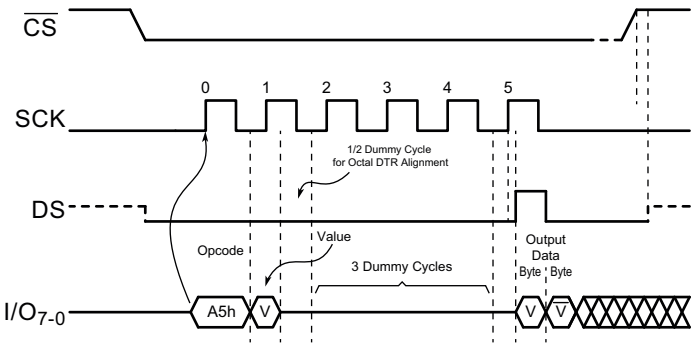


Figure 12-37.Echo with Inversion in Octal Mode - DTR



12.15 Enter QPI Mode (38h)

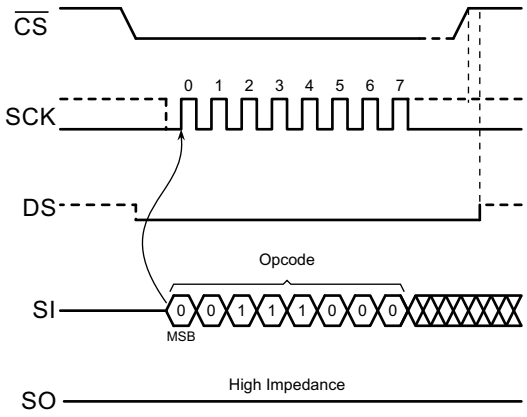
In order to enter the QPI Mode from the Standard SPI Mode, an Enter QPI Mode command must be issued. The Enter QPI Mode command can only be entered from Standard SPI Mode. If the device is in Octal Mode, the Return to Standard SPI Mode (FFh) command has to be issued first.

Before the Enter QPI Mode command can be issued, the Write Enable (06h) command must have been previously issued to set the WEL bit in the Status Register to a logical “1”.

To issue the Enter QPI Mode command, the $\overline{CS}$ pin must first be asserted and the opcode of 38h must be clocked into the device.

When the device is switched from SPI Mode to QPI Mode, the existing Program Suspend status, Erase Suspend status, and the Wrap Length settings will remain unchanged. The WEL bit in the Status Register will be reset back to the logical “0” state.

Figure 12-38.Enter QPI Mode



12.16 Enter Octal Mode (E8h)

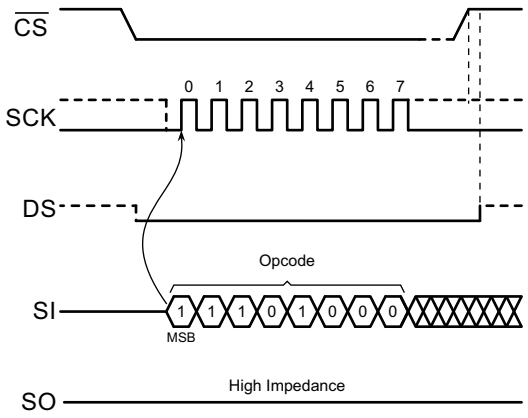
In order to enter the Octal Mode from the Standard SPI Mode, an Enter Octal Mode command must be issued. The Enter Octal Mode command can only be entered from Standard SPI Mode. If the device is in QPI Mode, the Return to Standard SPI Mode (FFh) command has to be issued first.

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 Register to a logical “1”.

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

When the device is switched from SPI Mode to QPI Mode, the existing Program Suspend status, Erase Suspend status, and the Wrap Length settings will remain unchanged. The WEL bit in the Status Register will be reset back to the logical “0” state.

Figure 12-39.Enter Octal Mode



12.17 Return to Standard SPI Mode (FFh)

In order to exit the QPI Mode or 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 Register to a logical “1”.

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

When the device is switched from QPI Mode to SPI Mode, the existing Program Suspend status, Erase Suspend status, and the Wrap Length settings will remain unchanged. The WEL bit in Status Register Byte 1 will be reset back to the logical ‘0’ state. STR’/DTR, OME and QPIE in Status Register Byte 2 will all be set to 0. The device will return to Standard SPI Mode, STR, independent of which mode it was in prior to issuing this command.

Figure 12-40.Return to Standard SPI Mode from QPI Mode - STR

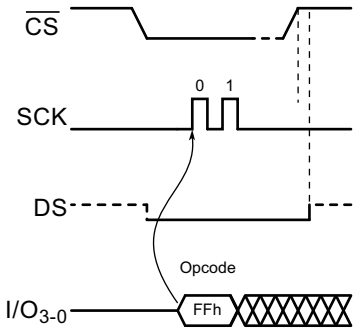


Figure 12-41.Return to Standard SPI Mode from Octal Mode - STR

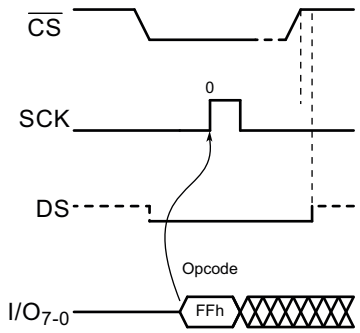


Figure 12-42.Return to Standard SPI Mode from QPI Mode - DTR

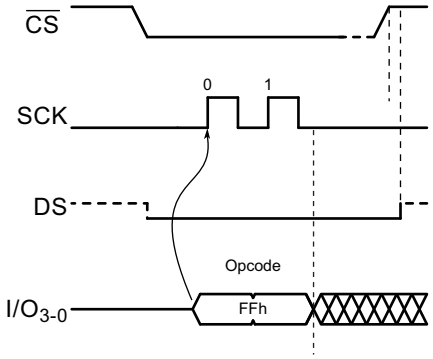
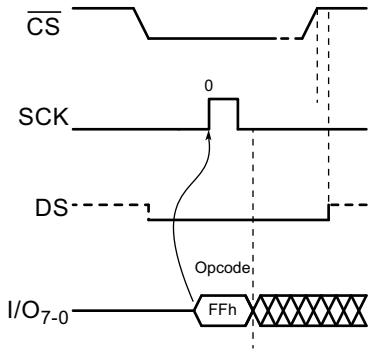


Figure 12-43.Return to Standard SPI Mode from Octal Mode - DTR



12.18 Read SFDP (5Ah)

The ATXP064 / ATXP064R contains a 512-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 opcode of 5Ah must be clocked into the device. After the opcode has been clocked in, the *three* address bytes must be clocked in to specify the starting address location of the first byte to read within the 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 DTR modes. Octal DTR mode has an additional ½ 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 (both Octal and QPI modes), SFDP command can only be used for addressing data on a 32-bit (4-byte) boundary. (Address bits A1 and A0 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 will continue reading back at the beginning of the register (000000h). No delays will be incurred when wrapping around from the end of the register to the beginning of the register.

Deasserting the $\overline{CS}$ pin will terminate the read operation and put the 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. The format of the SFDP register follows the format provided in JEDEC Standard No. 216 Rev B.

Figure 12-44.Read SFDP

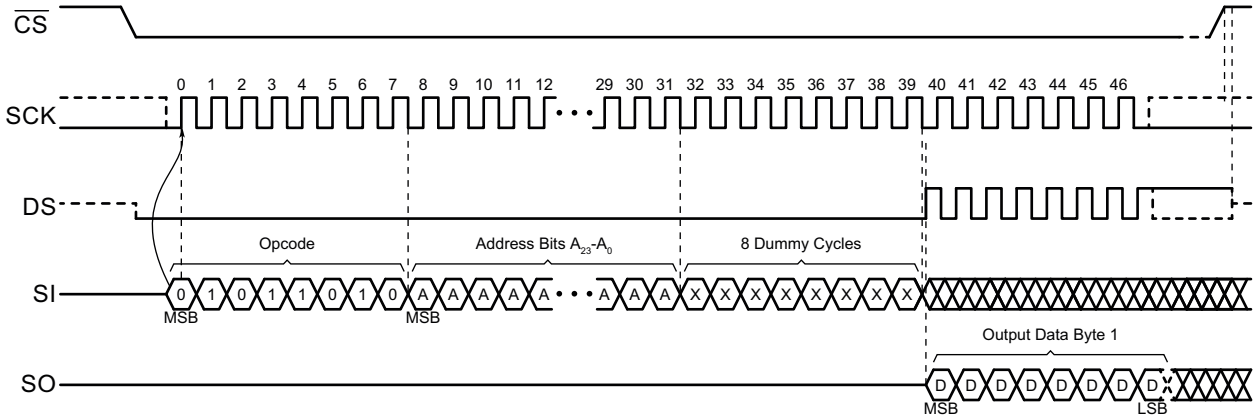


Figure 12-45.Read SFDP in QPI Mode - STR

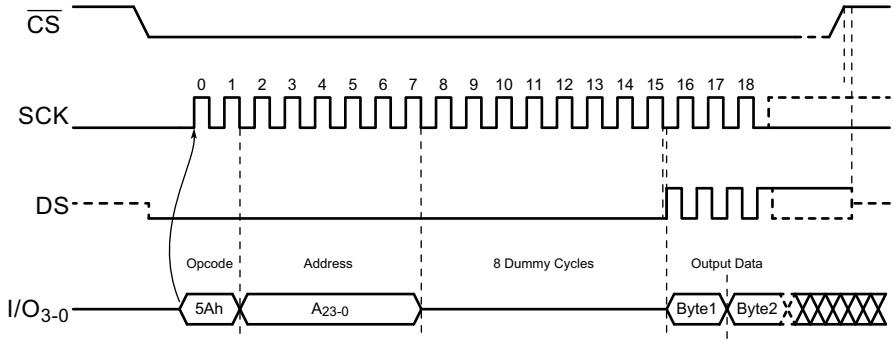


Figure 12-46.Read SFDP in Octal Mode - STR

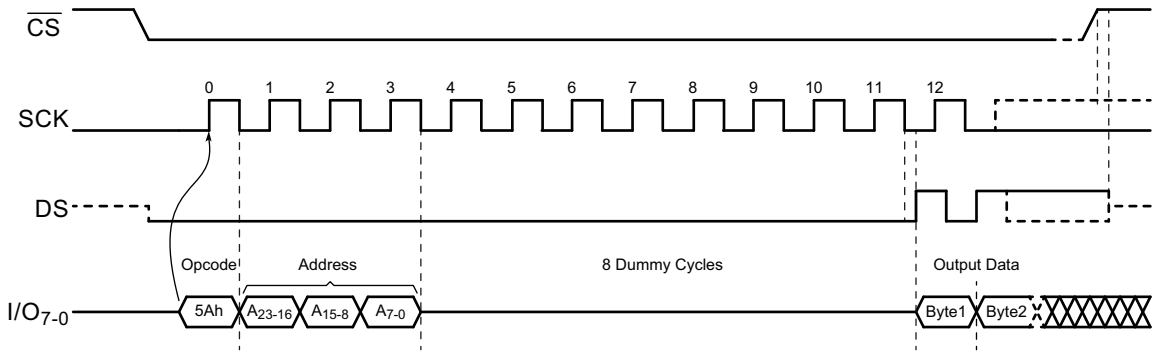


Figure 12-47.Read SFDP in QPI Mode - DTR

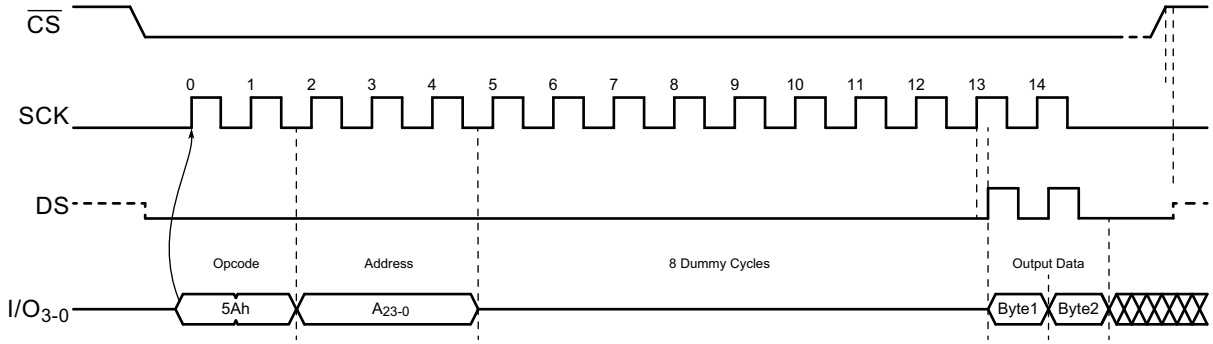
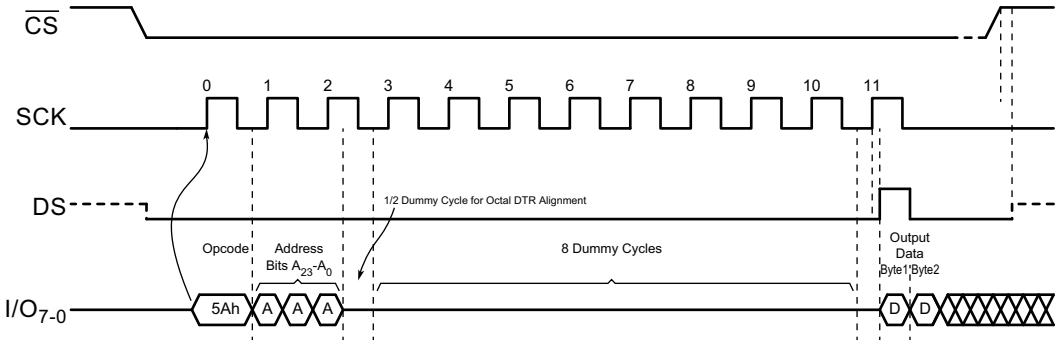


Figure 12-48.Read SFDP in Octal Mode - DTR



The first 76 bytes of the SFDP register contains SFDP parameters. The remaining portion of the SFDP register will read as 0xff.

The SFDP data is organized as 32-bit double words. The first 2 double words contain the SFDP header, the next 2 contain the first (and in this case only) parameter header. The next 9 double words contain the JEDEC Flash Parameter Tables.

Table 12-5. SFDP Register Summary - Values

Description	Bit Position				Address
	[31:24]	[23:16]	[15:8]	[7:0]	
SFDP Signature Header	50h	44h	46h	53h	0000003h - 000000h
	FFH	00h	01h	06h	0000007h - 000004h
1st Parameter Header	10h	01h	06h	00h	000000Bh - 000008h
	FFh	00h	00h	10h	000000Fh - 00000Ch
Flash Parameters 1	FFh	88h	20h	FDh	0000013h - 000010h
Flash Parameters 2	07h	FFh	FFh	FFh	0000017h - 000014h
Flash Parameters 3	00h	00h	00h	00h	000001Bh - 000018h
Flash Parameters 4	00h	00h	00h	00h	000001Fh - 00001Ch
Flash Parameters 5	FFh	FFh	FFh	FEh	0000023h - 000020h
Flash Parameters 6	00h	00h	FFh	FFh	0000027h - 000024h
Flash Parameters 7	0Bh	08h	FFh	FFh	000002Bh - 000028h
Flash Parameters 8	52h	0Fh	20h	0Ch	000002Fh - 00002Ch
Flash Parameters 9	60h	16h	D8h	10h	0000033h - 000030h
Flash Parameters 10	B6h	EDh	7Ah	20h	0000037h - 000034h
Flash Parameters 11	CDh	21h	F3h	80h	000003Bh - 000038h
Flash Parameters 12	3Dh	F5h	61h	20h	000003Fh - 00003Ch
Flash Parameters 13	75h	7Ah	75h	7Ah	0000043h - 000040h
Flash Parameters 14	5Ch	D5h	A7h	F7h	0000047h - 000044h
Flash Parameters 15	FFh	00h	00h	21h	000004Bh - 000048h
Flash Parameters 16	00h	00h	08h	80h	000004Fh - 00004Ch

12.18.1 SFDP Header

The SFDP Header is located at address 000000h of the SFDP data structure. It identifies the SFDP Signature, the number of parameter headers, and the SFDP revision numbers. Both the SFDP header and the individual parameter table headers include Major and Minor revision numbers.

Table 12-6. SFDP Header: 1st DWORD = 50444653h

Bits	Value	Description
31:24	50h	SFDP Signature Allows a user to know that the information is valid. Signature[31:0]: 50444653h = ASCII “P”,”D”, “F”,”S”. Will read as “SFDP” when starting read from address 0.
23:16	44h	
15:8	46h	
7:0	53h	

Table 12-7. SFDP Header: 2nd DWORD = FF000106h

Bits	Value	Description
31:24	FFh	Unused. Contains FFh and can never be changed
23:16	00h	Number of Parameter Headers (NPH). Specifies the number of parameter headers in the SFDP data structure. This number is 0-based. Therefore, 0 indicates 1 parameter header.
15:8	01h	SFDP Major Revision Number. The value in this field is 1 for devices which implement the JESD216B revision.
7:0	06h	SFDP Minor Revision Number. This 8-bit field indicates the minor revision number of this standard. The value in this field is 6 for devices which implement the JESD216B revision.

12.18.2 Basic Flash Parameter Header

The Parameter Tables contain coded information describing the features and capabilities of the ATXP064 / ATXP064R. The first Parameter Table is defined by JEDEC and its starting address is specified by the PTP field of the Flash Parameter Header. This table identifies some of the basic features of the SPI protocol for flash memory devices. The first Parameter Header starts at byte offset 08h. For ATXP064 / ATXP064R, there are no additional parameter headers.

Table 12-8. Flash Parameter Header: 1st DWORD = 10010600h

Bits	Value	Description
31:24	10h	Parameter Table Length. This field specifies how many DWORDs are in the Parameter table. This field is 1's based. Therefore, 10h indicates 16 DWORDs.
23:16	01h	Parameter Table Major Revision Number. The value in this field is 1 for this table as defined by JESD216B.
15:8	06h	Parameter Table Minor Revision Number. The value in this field is 6 for this table as defined by JESD216B.
7:0	00h	Parameter ID LSB. The LSB of the JEDEC Basic Flash Parameter Table is 00h.

Flash Parameter Header: 2nd DWORD = FF000010h

Bits	Value	Description
31:24	FFh	Parameter ID MSB FFh indicates Jedec Basic Parameter Table.
23:0	00 00 10h	Parameter Table Pointer (PTP). This address specifies the start of this header's Parameter Table in the SFDP structure. This is a byte address.

12.18.3 Basic Flash Parameter Tables

Table 12-9. Flash Parameter Table: 1st DWORD = FF8C20FDh

Bits	Value	Description
31:23	1FFh	Unused, all bits set to 1.
22	0	0: (1-1-4) Fast Read NOT supported.
21	0	0: (1-4-4) Fast Read NOT supported.
20	0	0: (1-2-2) Fast Read NOT supported.
19	1	1: DTR Clocking supported
18:17	10	Number of bytes used in addressing flash array read, write and erase: 10b: 4-Byte only addressing
16	0	0: (1-1-2) Fast Read NOT supported.
15:8	20	4 Kilobyte Erase Instruction
7:5	111	Unused, all bits set to 1.
4	1	Write Enable Instruction Select for Writing to Volatile Status Register 1: Flash device requires instruction 06h as the write enable prior to performing a volatile write to the status register.
3	1	Volatile Status Register Block Protect bits 1: Block Protect bits in device's status register are solely volatile.
2	1	Write Granularity 1: Buffer size is larger than 64 bytes. (256 bytes for ATXP064 / ATXP064R.)
1:0	01	Block/Sector Erase Sizes 01b: 4 kilobyte Erase is supported throughout the device.

Table 12-10. Flash Parameter Table: 2nd DWORD = 07FFFFFFh

Bits	Value	Description
31:0	0	Flash Memory Density. For densities 2 gigabits or less, bit-31 is set to 0b.
30:0	07FFFFFFh	Flash Memory Density. The field 30:0 defines the size in bits. 07FFFFFFh = 128 megabits

Table 12-11. Flash Parameter Table: 3rd DWORD = 00000000h

Bits	Value	Description
31:24	00h	(1-1-4) Fast Read Instruction: Not supported.
23:21	000	(1-1-4) Fast Read Number of Mode Clocks. This field is 000b as Mode Bits are not supported.
20:16	00000	(1-1-4) Fast Read Number of Wait states (dummy clocks) needed before valid output. This field is 00000b as (1-1-4) Fast Read is not supported.

Table 12-11. Flash Parameter Table: 3rd DWORD = 00000000h

Bits	Value	Description
15:8	00h	(1-4-4) Fast Read Instruction. This field is 00h as (1-1-4) Fast Read is not supported.
7:5	000	Quad Input Address Quad Output (1-4-4) Fast Read Number of Mode Clocks. This field is 000b as Mode bits are not supported,
4:0	00000b	(1-4-4) Fast Read Number of Wait states (dummy clocks) needed before valid output. This field is 00000b as (1-1-4) Fast Read is not supported.

Table 12-12. Flash Parameter Table: 4th DWORD = 00000000h

Bits	Value	Description
31:24	00h	(1-2-2) Fast Read Instruction: Not supported.
23:21	000	(1-2-2) Fast Read Number of Mode Clocks. This field is 000b as Mode Bits are not supported.
20:16	00000	(1-2-2) Fast Read Number of Wait states (dummy clocks) needed before valid output. This field is 00000b as (1-2-2) Fast Read is not supported.
15:8	00h	(1-2-2) Fast Read Instruction. This field is 00h as (1-2-2) Fast Read is not supported.
7:5	000	(1-2-2) Fast Read Number of Mode Clocks. This field is 000b as Mode bits are not supported,
4:0	00000b	(1-2-2) Fast Read Number of Wait states (dummy clocks) needed before valid output. This field is 00000b as (1-2-2) Fast Read is not supported.

Table 12-13. Flash Parameter Table: 5th DWORD = FFFFFFFEh

Bits	Value	Description
31:5	FFFFFFFh	Reserved. These bits default to all 1's.
4	1	1: (4-4-4) Fast Read supported.
3:1	111	Reserved. These bits default to all 1's.
0	0	0: (2-2-2) Fast Read NOT supported.

Table 12-14. Flash Parameter Table: 6th DWORD = 0000FFFFh

Bits	Value	Description
31:24	00h	(2-2-2) Fast Read Instruction: Not supported.
23:21	000	(2-2-2) Fast Read Number of Mode Clocks. This field is 000b as Mode Bits are not supported.
20:16	00000	(2-2-2) Fast Read Number of Wait states (dummy clocks) needed before valid output. This field is 00000b as (2-2-2) Fast Read is not supported.
15:0	FFFFh	Reserved. These bits default to all 1's

Table 12-15. Flash Parameter Table: 7th DWORD= 0B08FFFFh

Bits	Value	Description
31:24	0Bh	(4-4-4) Fast Read Instruction. Instruction for quad input instruction/address, quad output data Fast Read.
23:21	000	(4-4-4) Fast Read Number of Mode Clocks. This field is 000b as Mode bits are not supported.
20:16	00100	(4-4-4) Fast Read Number of Wait states (dummy clocks) needed before valid output. This field is set to 16, as this value will always work independent of frequency, and is the default value after power up. The minimum number of wait states required is a function of operating frequency and can be changed as described in Section 11.3, Status/Control Register Byte 3.
15:0	FFFFh	Reserved. These bits default to all 1's

Table 12-16. Flash Parameter Table: 8th DWORD = 520F200Ch

Bits	Value	Description
31:24	52h	Erase Type 2 Instruction. Instruction used to erase the number of bytes specified by Erase Type 2 Size (bits 23-16).
23:16	0Fh	Erase Type 2 Size = 32 kilobytes.
15:8	20h	Erase Type 1 Instruction. Instruction used to erase the number of bytes specified by Erase Type 1 Size (bits 7-0).
7:0	0Ch	Erase Type 1 Size = 4 kilobytes.

Table 12-17. Flash Parameter Table: 9th DWORD = 6017D810h

Bits	Value	Description
31:24	60h	Erase Type 4 Instruction. (Chip Erase) Instruction used to erase the number of bytes specified by Erase Type 4 Size (bits 23-16).
23:16	16h	Erase Type 4 Size = 4 megabytes.
15:8	D8h	Erase Type 3 Instruction. Instruction used to erase the number of bytes specified by Erase Type 3 Size (bits 7-0).
7:0	10h	Erase Type 3 Size = 64 kilobytes.

Table 12-18. Flash Parameter Table: 10th DWORD = B6ED7A20h

Bits	Value	Description
31:30	10	Erase Type 4 Erase, Typical time - units: 10b = 128 ms.
29:25	11011	Erase Type 4 Erase, Typical time - count = 27. Formula: typical time = (count + 1) * units = 28 * 128 ms = 3584 ms ≈ 3600 ms.
24:23	01	Erase Type 3 Erase, Typical time - units: 01b = 16 ms.
22:18	11011	Erase Type 3 Erase, Typical time - count = 27. Formula: typical time = (count + 1) * units = 28 * 16 ms = 448 ms ≈ 450 ms.
17:16	01	Erase Type 2 Erase, Typical time - units: 01b = 16 ms.
15:11	01111	Erase Type 2 Erase, Typical time - count = 15. Formula: typical time = (count + 1) * units = 16 * 16 ms = 256 ms ≈ 250 ms.

Table 12-18. Flash Parameter Table: 10th DWORD = B6ED7A20h

Bits	Value	Description
10:9	01	Erase Type 1 Erase, Typical time - units: 01b = 16 ms.
8:4	00010	Erase Type 1 Erase, Typical time - count = 2. Formula: typical time = (count + 1) * units = 2 * 16ms = 32 ms ≈ 35 ms.
3:0	0000	Multiplier from typical erase time to maximum erase time - count. Formula: Erase Type n (or Chip) erase maximum time = 2 * (count + 1) * Erase Type n erase typical time. The value 0 gives Maximum Erase Time = 2x Typical Erase Time. The actual factor for ATXP064 is less than 1.35, but values that low are not supported by the Jedec SFDP Standard.

Table 12-19. Flash Parameter Table: 11th DWORD = CD21F380h

Bits	Value	Description
31	1	Reserved
30:29	10	Chip Erase, Typical time - units: 01b = 256 ms.
28:24	01101	Chip Erase, Typical time - count = 13. Formula: typical time = (count + 1) * units = 14 * 256 ms = 3584 ms ≈ 3600 ms.
23	0	Byte Program Typical time, additional byte - units: 0 = 1 µs.
22:19	0100	Byte Program Typical time, additional byte - count = 4. Time the device typically takes to write each additional byte after the first. User must poll device busy to determine if the operation has completed. Formula: additional byte time = (count + 1) * units/byte = 5 * 1 µs/byte = 5 µs/byte ≈ 4.9µs/byte. NOTE: The programming time for small numbers of bytes does not scale linearly up to a full page programming time. When the number of bytes being programmed exceeds ½ of a page size, users should base estimates on the Page Program Typical Time in this DWORD.
18	0	Byte Program Typical time, first byte - units: 0 = 1 µs.
17:14	0111	Byte Program Typical time, first byte - count = 7. Time the device typically takes to write the first byte in a sequence. User must poll device busy to determine if the operation has completed. Formula: first byte typical time = (count + 1) * units/byte = 8 * 1 µs/byte = 8 µs/byte.
13	1	Page Program Typical time - units: 1 = 64 µs.
12:8	10011	Byte Program Typical time, first byte - count = 19. Time the device typically takes to write a full page. User must poll device busy to determine if the operation has completed. The user may scale this by ½ or ¼ to determine approximate times for ½ and ¼ page program operations. Formula: typical page program time = (count + 1)*units = 20 * 64 µs = 1280 µs ≈ 1.25 ms.
7:4	1000	Page Size: 256 bytes. (Value = 8, 2^8 = 256.)
3:0	0000	Multiplier from typical time to max time for Page or byte program Formula: maximum time = 2 * (count + 1)*typical time = 2x typical time.

Table 12-20. Flash Parameter Table: 12th DWORD = 3DF56120h

Bits	Value	Description
31	0	0: Suspend / Resume is supported
30:29	01	Suspend in-progress erase max latency - units: 01 = 1 µs.
28:24	11101	Suspend in-progress erase max latency - count = 29. Maximum time required by the flash device to suspend an in-progress erase and be ready to accept another command which accesses the flash array. Formula: erase max latency = (count + 1) * units = 30 * 1µs = 30 µs.
23:20	1111	Erase Resume to Suspend Interval. The device requires this typical amount of time to make progress on the erase before allowing another suspend. It is possible to immediately suspend again after a resume - there is no required minimum time between resuming an operation and suspending the operation again. However, the device requires some average amount of active operation time, after a resume, to make progress on the operation, before another suspend. This parameter recommends an average interval of time that should be allowed between a resume and the next suspend in order for the operation to eventually complete. If there are some intervals less than the recommended value there should be a similar number of intervals that are longer than the recommended value. If the interval is consistently less than the recommended value the operation may never finish. Formula: erase resume to suspend interval = (count + 1) * 64 µs = 16 * 64µs = 1024 µs ≈ 1 ms.
19:18	01	Suspend in-progress program max latency- units: 01 = 1 µs.
17:13	01011	Suspend in-progress program max latency - units: 01 = 1 µs.Maximum time required by the flash device to suspend an in-progress program and be ready to accept another command which accesses the flash array. Formula: suspend in-progress program max latency =(count + 1) * units = 12 * 1µs = 12 µs.
12:9	0000	Program Resume to Suspend Interval The device requires this typical amount of time to make progress on the program operation before allowing another suspend. It is possible to immediately suspend again after a resume - there is no required minimum time between resuming an operation and suspending the operation again. However, the device requires some average amount of active operation time, after a resume, to make progress on the operation, before another suspend. This parameter recommends an average interval of time that should be allowed between a resume and the next suspend in order for the operation to eventually complete. If there are some intervals less than the recommended value there should be a similar number of intervals that are longer than the recommended value. If the interval is consistently less than the recommended value the operation may never finish. Formula: program resume to suspend interval =(count + 1) * 64 µs = 1 * 64µs = 64 µs. Actual value for ATXP064 is 2 µs, but SFDP does not support values below 64 µs.
8	1	Reserved.
7	0	Prohibited Operations During Erase Suspend: Additional erase or program restrictions apply. See Table 8-1 for details.
6	0	Prohibited Operations During Erase Suspend. Refer to vendor data sheet for read restrictions See Table 8-1 for details.
5	1	Prohibited Operations During Erase Suspend: May not initiate a page program in the erase suspended erase type size.
4	0	Prohibited Operations During Erase Suspend: May not initiate a new erase anywhere (erase nesting not permitted).
3	0	Prohibited Operations During Program Suspend: Additional erase or program restrictions apply. See Table 8-1 for details.

Table 12-20. Flash Parameter Table: 12th DWORD = 3DF56120h

Bits	Value	Description
2	0	Prohibited Operations During Program Suspend: Refer to vendor data sheet for read restrictions See Table 8-1 for details.
1	0	Prohibited Operations During Program Suspend: May not initiate a new page program anywhere (program nesting not permitted).
0	0	Prohibited Operations During Program Suspend: May not initiate a new erase anywhere (erase nesting not permitted).

Table 12-21. Flash Parameter Table: 13th DWORD = 757A757Ah

Bits	Value	Description
31:24	75h	Suspend Instruction. Instruction used to suspend a write or erase type operation.
23:16	7Ah	Resume Instruction Instruction used to resume a write or erase type operation.
15:8	75h	Program Suspend Instruction. Instruction used to suspend a program operation. (Same value as the “Suspend Instruction” field above.)
7:0	7Ah	Program Resume Instruction. Instruction used to resume a program operation. (Same value as the “Resume Instruction” field above.)

Table 12-22. Flash Parameter Table: 14th DWORD = 5CD5A7F7h

Bits	Value	Description
31	0	0: Deep Power-Down Supported. See Section 12.2, Deep Power-Down (B9h). ATXP064 also supports Ultra-Deep Power-Down. See Section 12.5, Ultra-Deep Power-Down (79h).
30:23	B9h	Instruction used to enter Deep Power-Down. See Section 12.2, Deep Power-Down (B9h).
22:15	ABh	Instruction used to exit Deep Power-Down. See Section 12.4, Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh).
14:13	01	Exit Deep Power-Down to next operation delay - units: 01 = 1 µs.
12:8	00111	Exit Deep Power-Down to next operation delay - count = 7. Maximum time required by the flash device to exit Deep Power-Down and be ready to accept any command. Formula: exit Deep Power-Down to next operation delay = (count+1) * units = 8 * 1 µs = 8 µs.
7:2	111101	Status Register Polling Device Busy This bit field defines various ways the flash device's busy status may be polled. Supported: Reading the Status Register with 05h instruction and checking WIP bit[0] (0=ready; 1=busy). (Not supported: Read instruction 70h for reading Flag Status Register bit[7].)
1:0	11	Reserved

Table 12-23. Flash Parameter Table: 15th DWORD = FF000021h

Bits	Value	Description
31:24	FFh	Reserved
23	0	HOLD or RESET Disable: 0: Not supported
22:20	000	Quad Enable Requirements (QER): This field describes whether the device contains a Quad Enable (QE) bit used to enable (1-1-4) and (1-4-4) quad read or quad program operations. The ATXP064 / ATXP064R does not support (1-1-4) and (1-4-4) operations as indicated by 3rd DWORD, therefore this value is N/A.
19:16	0000	(0-4-4) Mode Entry Method: (0-4-4) Mode is not supported on the ATXP064 / ATXP064R.
15:10	0000	(0-4-4) Mode Exit Method: (0-4-4) Mode is not supported on the ATXP064 / ATXP064R.
9	0	(0-4-4) Mode is not supported.
8:4	00010	(4-4-4) mode enable sequence: x_xx1xb: issue instruction 38h.
3:0	0001	(4-4-4) mode disable sequence: xxx1b: issue FFh instruction.

Table 12-24. Flash Parameter Table: 16th DWORD = 40000882h

Bits	Value	Description
31:24	01000000	Enter 4-Byte Addressing: The ATXP064 / ATXP064R always operates in 4-Byte address mode.
23:14	000h	Exit 4-Byte Addressing: Exit 4-Byte Addressing not supported on this device.
13:8	001000	Soft Reset and Rescue Sequence Support: xx_1xxxb: issue instruction F0h.
7	1	Reserved.
6:0	0000010	Volatile or Non-Volatile Register and Write Enable Instruction for Status Register 1: xxx_xx1xb: Volatile Status Register 1, status register powers-up with bits set to “1”s, use instruction 06h to enable write.

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

		ATXP064
Operating Temperature (Case, packaged parts)	Ind.	-40°C to 85°C
Operating Temperature (KGD)	Extended	-40°C to 105°C
V_{CC} Power Supply		1.65 - 1.95V

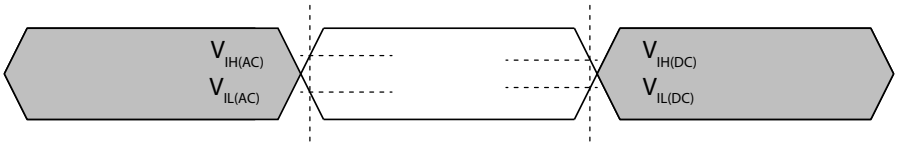
13.3 DC Characteristics

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, QPI and 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)	1-pin STR (Read @ 1pf load)		1mA + 30µA/MHz	10	mA
		8-pin DTR (Read @ 1pf load)		1mA + 142µA/MHz	30	mA
		4-pin DTR (Read @ 1pf load)		1mA + 91µA/MHz	20	mA
		4-pin STR (Read @ 1pf load)		1mA + 65µA/MHz	15	mA
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})			1	µA

Symbol	Parameter	Condition	Min	Typ	Max	Units
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	DTR, f = 133MHz			0.5	ns
C_{Load}	Capacitive load	DTR, f = 133MHz, Driver type 0x00			6	pF
		DTR, f = 133MHz, Driver type 0x02			4.5	
		DTR, f = 133MHz, Driver type 0x03			3	
C_{in}	Input Capacitance	Packaged parts, WLCSP		4		pF
		KGD		3		

1. Device reading random data. On average, I/O pins will remain unchanged for 50% of bits read and will be switching for 50% of bits read.

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 opcodes)	SPI mode			66	MHz
		QPI and Octal Mode			133	
f_{RDLF}	Maximum Clock Frequency for 03h, 13h, 5Ah and D4h opcodes (Read Array – Low Frequency, Read SFDP, and Buffer Read)				50	MHz

13.5 AC Characteristics – All Other Parameters

Symbol[4]	Parameter		Min	Typ	Max	Units
t_{CLKH}	Input SCK High Time	SPI mode (excluding 03h and 13h opcodes)	6.75			ns
		SPI mode (03h and 13h opcodes)	9			
		QPI and Octal modes, SDR mode	3			
t_{CLKL}	Input SCK Low Time	SPI mode (excluding 03h and 13h opcodes)	6.75			ns
		SPI mode (03h and 13h opcodes)	9			
		QPI and Octal modes, SDR mode	3			
t_{CKMPW}	Input SCK: Minimum pulse width	QPI and Octal modes, DDR mode	3.375			ns
t_{CLKR}[1]	Input SCK Rise Time, Peak-to-Peak (Slew Rate)				1	ns
t_{CLKF}[1]	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
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})	Chip Select Low Setup Time (relative to SCK)	Up to 100 MHz	5			ns
		100 - 133 MHz	6			
t_{CKLCSH} (t_{CSLH})	Chip Select Low Hold Time (relative to SCK)	Up to 100 MHz	5			ns
		100 - 133 MHz	6			
t_{CSHCKH} (t_{CSHS})	Chip Select High Setup Time (relative to SCK)	Up to 100 MHz	5			ns
		100 - 133 MHz	6			

13.5 AC Characteristics – All Other Parameters

Symbol[4]	Parameter		Min	Typ	Max	Units
t_{CKLCSL} (t_{CSHH})	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_{DS}[2]	Data In Setup Time		2			ns
t_{DH}[2]	Data In Hold Time		0.4			ns
t_{DIS}[1]	Output Disable Time				7.5	ns
t_V	Output Valid Time (SCK to DS or I/O out); C_L =10pF				6	ns
t_{OH}	Output Hold Time		0			ns
t_{WPS}[1]	Write Protect Setup Time		20			ns
t_{WPH}[1]	Write Protect Hold Time		100			ns
t_{EDPD}[1]	Chip Select High to Deep Power-Down				0.5	µs
t_{EUDPD}[1]	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}	Exit Ultra-Deep Power-Down Time			50	70	µs
t_{RDPD}[1]	Chip Select High to Standby Mode (Resume from Deep Power-Down or Ultra-Deep Power-Down (ABh) Time)			5	8	µs
t_{RQ}	Output Skew (referenced to DS)				0.6	ns
t_{RQH}	Output Hold Skew (referenced to DS)				0.6	ns
SR_{OUT}	Output Slew Rate		1.7			V/ns

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 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		1.65V-1.95V Min	Typ	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)(3)	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 Register Time - Volatile Registers (all commands)				200	ns
t_{WRSRNV}	Write Status Register Time - Non-Volatile Registers (71h command)			20	40	ms

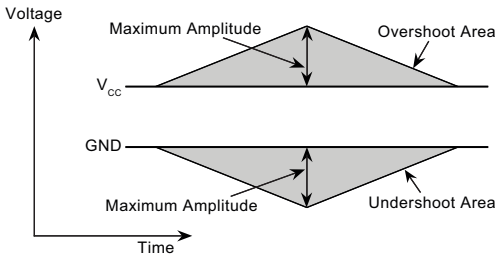
Note:
1. Maximum values indicate worst-case performance after 100,000 Program/Erase cycles.
2. For ATXP064R only: Programming and erase times for concurrent read and write will be longer than for regular programming. The increase in programming and erase times will depend on the frequency of read commands. 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

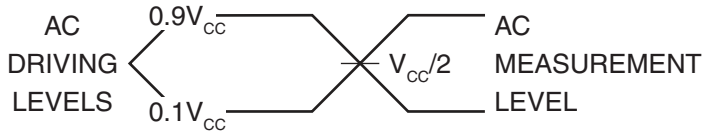
Table 13-1. Overshoot/Undershoot Specification

Parameter	1.65V-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

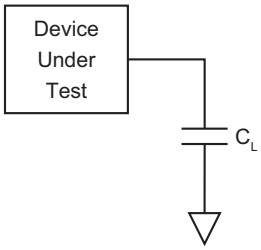


13.8 Input Test Waveforms and Measurement Levels



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) will be in a high impedance state, and a high-to-low transition on the $\overline{CS}$ pin will be required to start a valid instruction. The SPI mode (Mode 3 or Mode 0) will be 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 will be 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. 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 will not respond to any commands.

If the first operation to the device after power-up will be 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 will be a maximum time of t_{PUW}. After the t_{PUW} time, the device will be in the 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 VCC 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 will reset 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

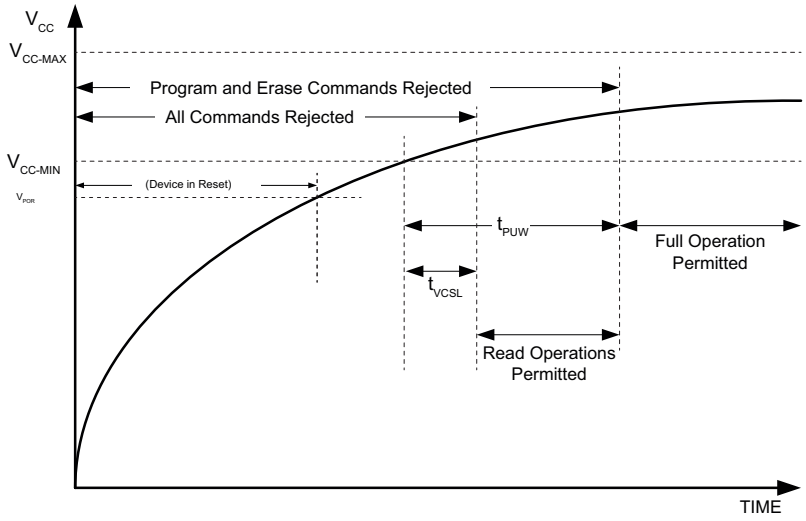
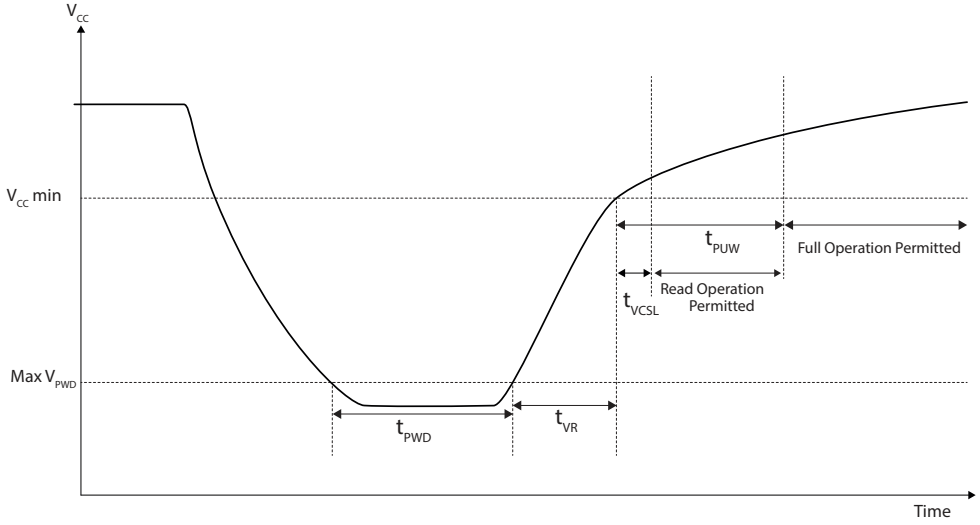


Figure 14-2. Power-Up After Brown-Out Timing



15. AC Waveforms

Figure 15-1. Serial Input Timing

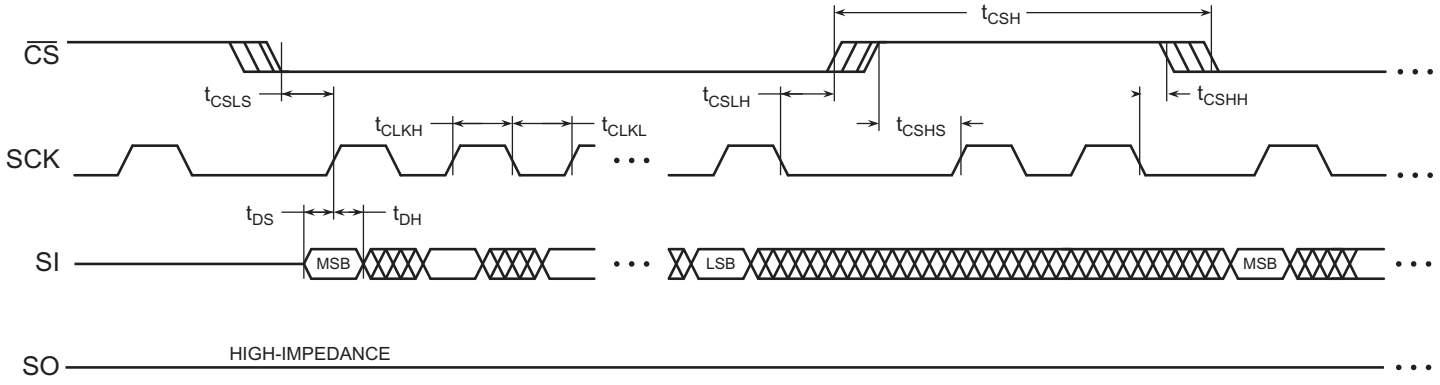


Figure 15-2. Serial Output Timing

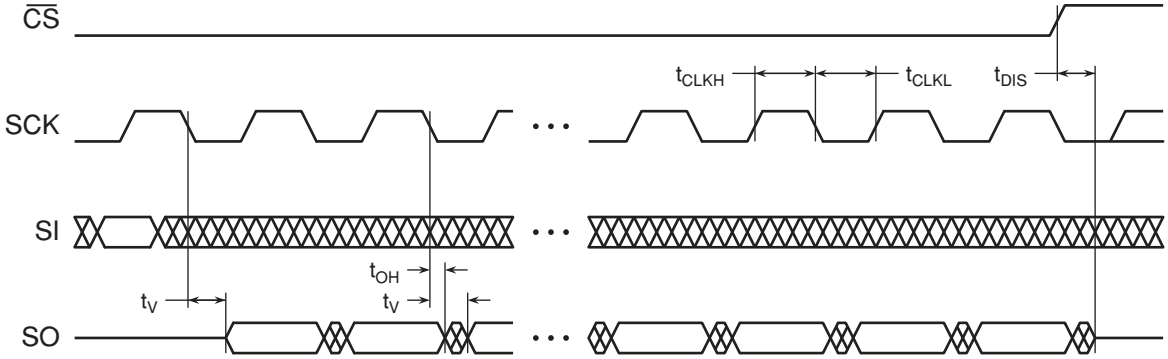


Figure 15-3. DS Pre- and Post-amble Timing (QPI DTR Mode shown)

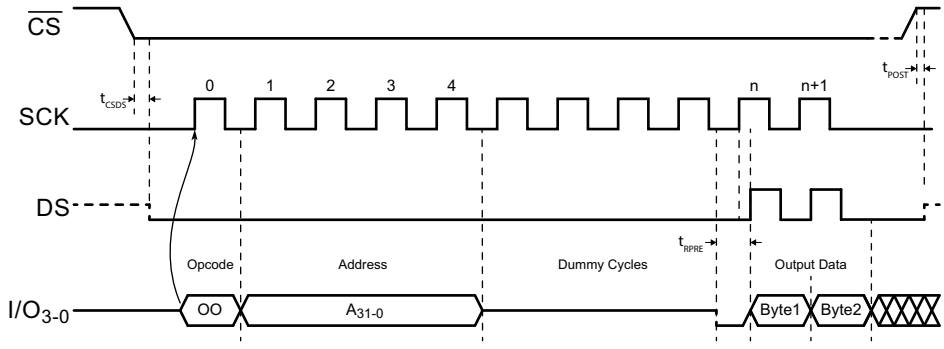


Figure 15-4. $\overline{WP}$ Timing for Write Status Register Command When BPL = 1

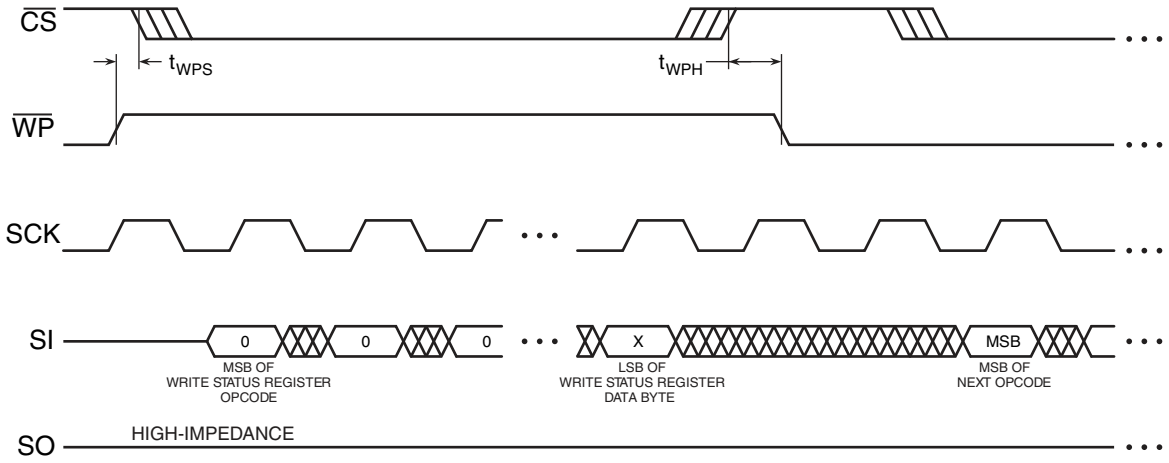


Figure 15-5. Device Input Timing DTR

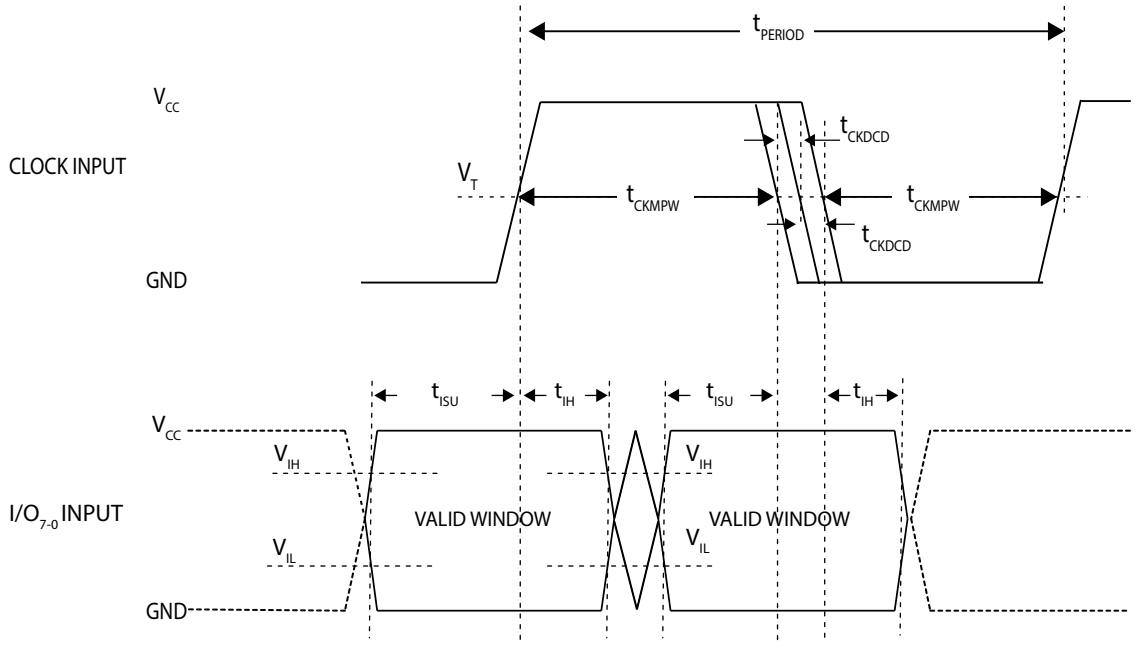
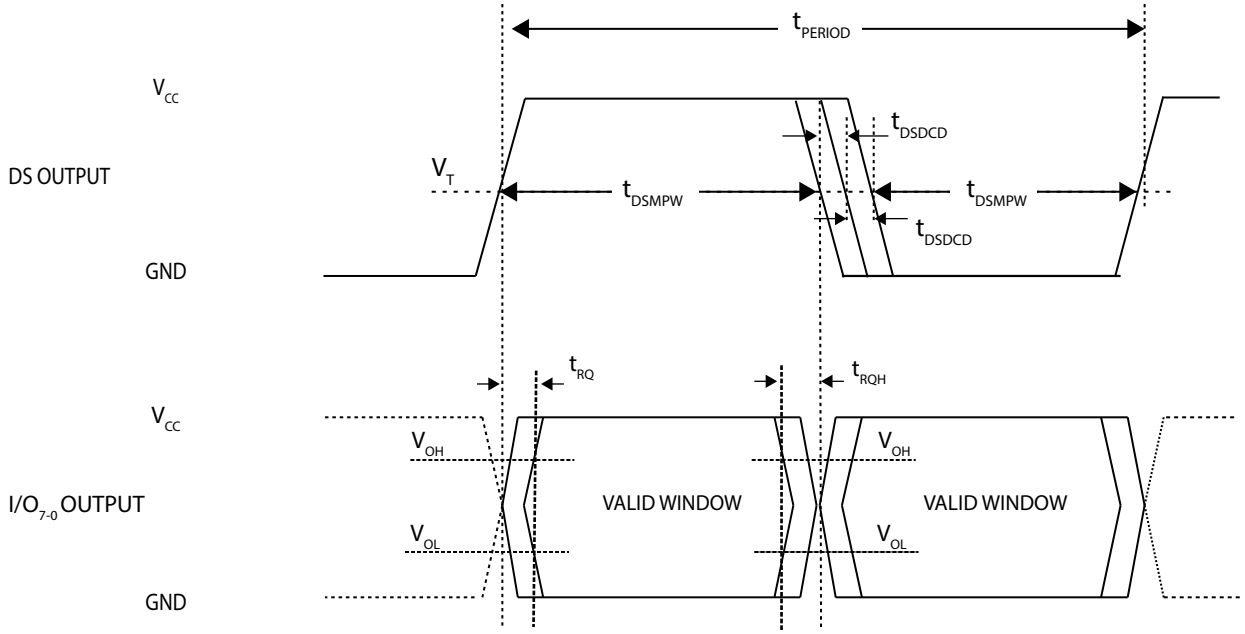


Figure 15-6. Device Output Timing DTR



16. Ordering Information

16.1 Ordering Code Detail

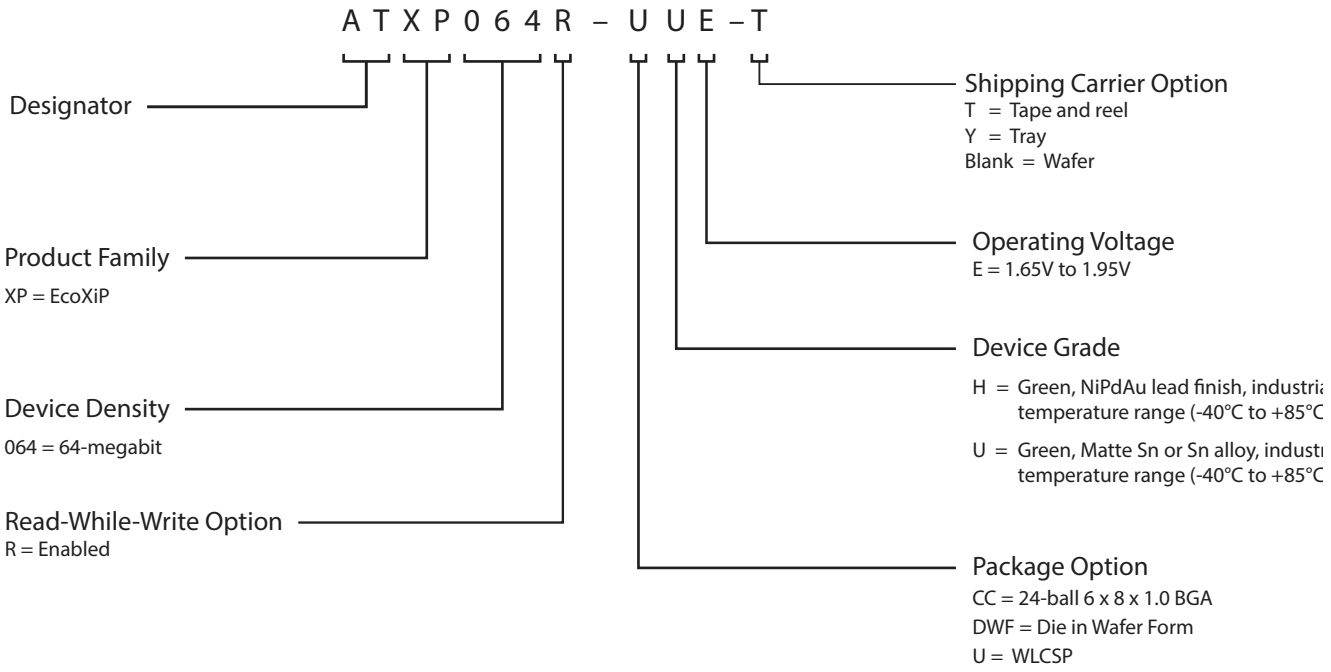


Table 16-1. ATXP064 Ordering Codes (without Read-While-Write feature)

Ordering Code [1]	Package	Lead Finish	Operating Voltage	Max. Freq. (MHz)	Operation Range
ATXP064-CCUE-Y	24CBGA	SnAgCu	1.65V to 1.95V	133	Industrial (-40°C to +85°C)
ATXP064-CCUE-T					
ATXP064-UUE-T [2]	WLCSP				
ATXP064-DWF[2]	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 mechanical drawing or Die Sales information.

Table 16-2. ATXP064R Ordering Codes (with Read-While-Write feature)

Ordering Code [1]	Package	Lead Finish	Operating Voltage	Max. Freq. (MHz)	Operation Range
ATXP064R-CCUE-Y	24CBGA	SnAgCu	1.65V to 1.95V	133	Industrial (-40°C to +85°C)
ATXP064R-CCUE-T					
ATXP064R-UUE-T [2]	WLCSP				
ATXP064R-DWF[2]	DWF	N/A			Extended (-40°C to +105°C)

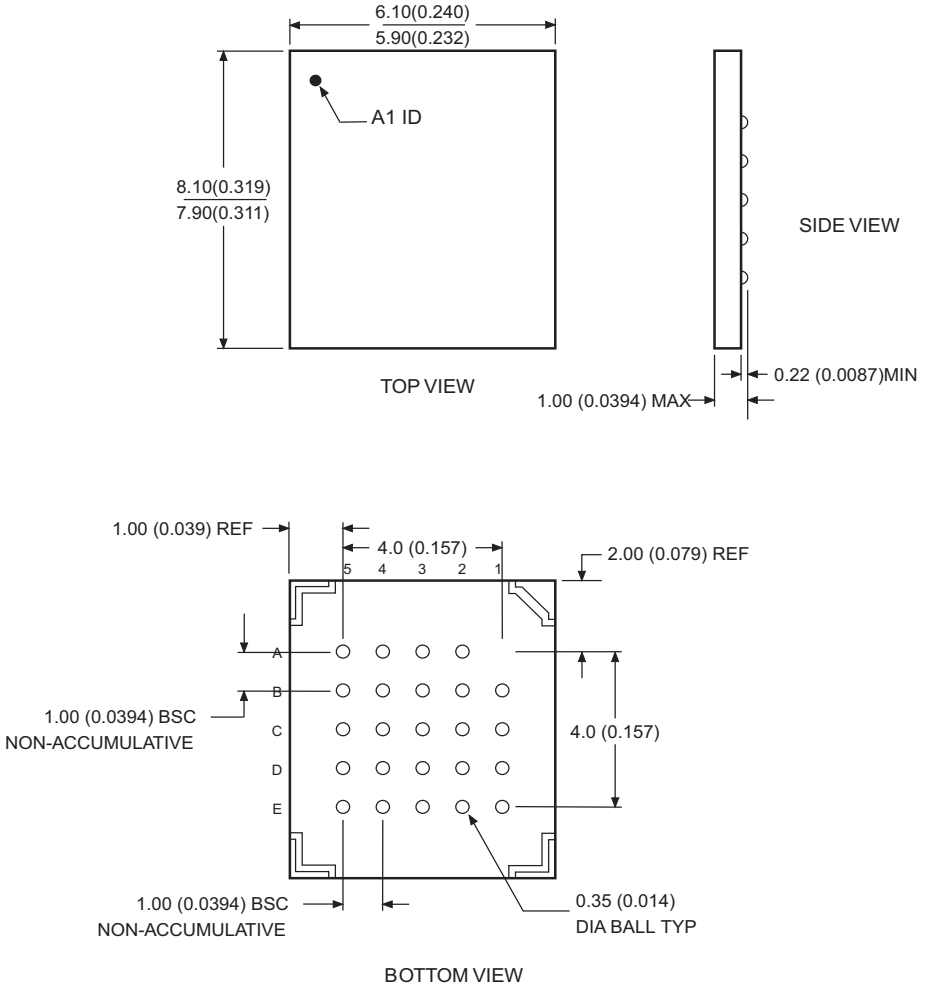
1. The shipping carrier option code is not marked on the device.
2. Contact Adesto for mechanical drawing or Die Sales information.

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)
WLCSP	Wafer Level CSP / die Ball Grid Array (dBGA)
DWF	Die in Wafer Form

17. Packaging Information

Figure 17-1. 24C2 — 24-ball CBGA

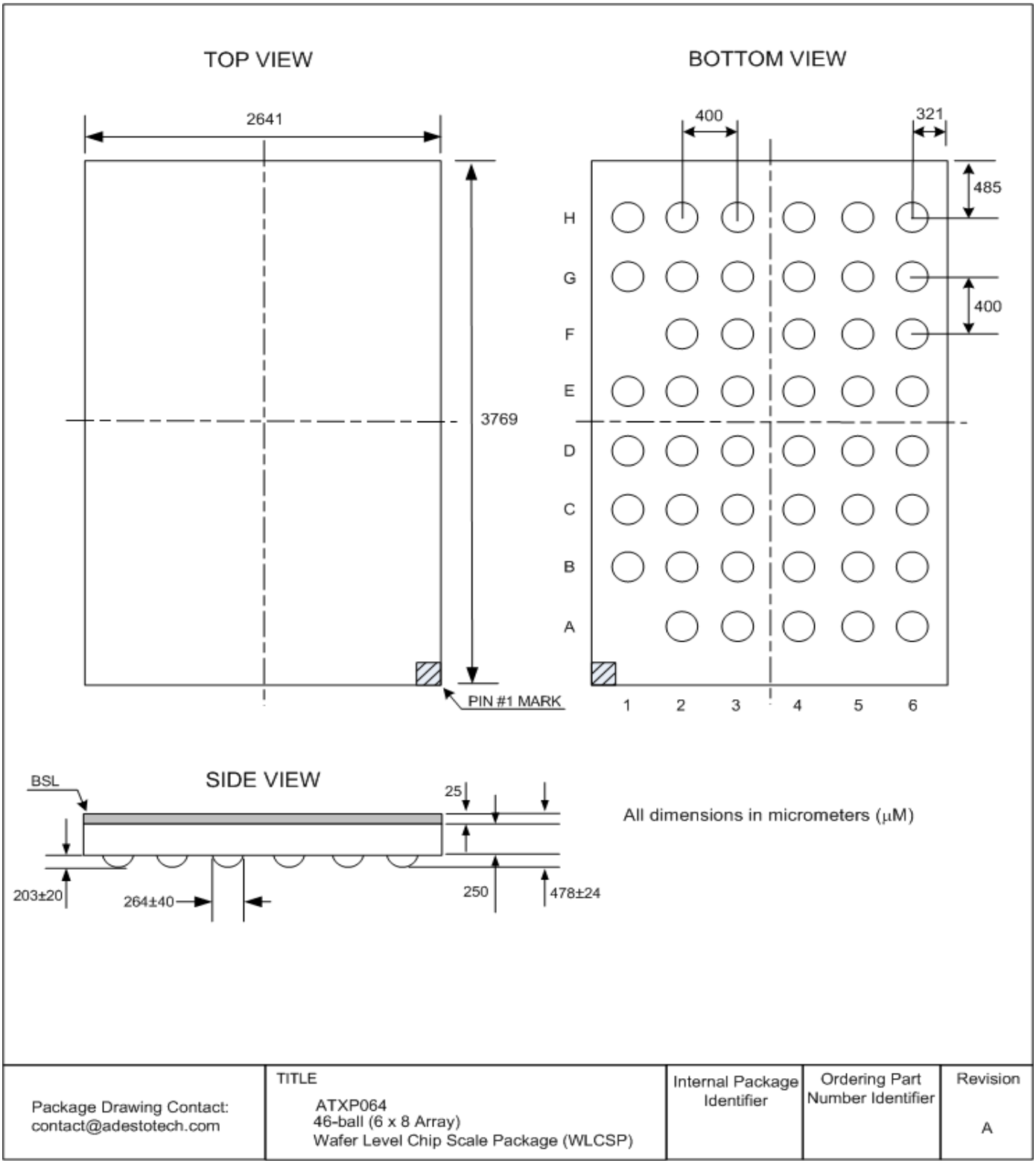
Dimensions in Millimeters and (Inches).
Controlling dimension: Millimeters.



06/2016

adesto™ TECHNOLOGIES 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

Figure 17-2. 46-ball 6 x 8 Array WLCSP



18. Revision History

Revision	Date	Change History
A	04/2019	Initial release.

Corporate Office

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