钍地半导体
Tudi Semiconductor

Product Specification

TUDI-EPCS16SI8N

Serial Configuration (EPCS) Devices Datasheet

网址 www.sztdbdt.com

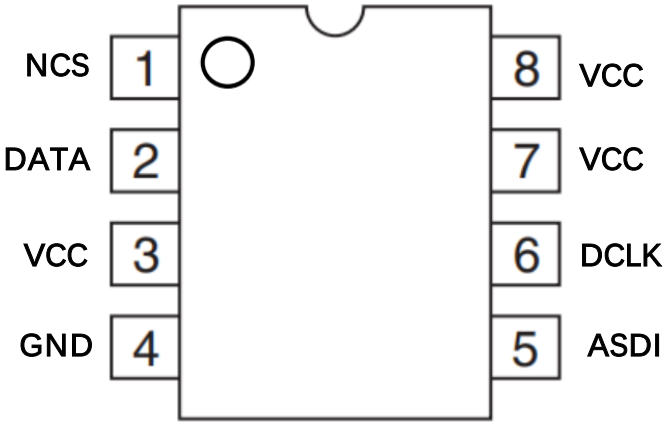
用芯智造·卓越品质

semiconductor device manufacturer

- Design
- research and development
- production
- and sales

Features

- Supports active serial (AS) x1 configuration scheme
- Easy-to-use four-pin interface
- Low cost, low pin count, and non-volatile memory
- Low current during configuration and near-zero standby mode current
- 2.7-V to 3.6-V operation
- EPCS16 devices available in 8-pin small-outline integrated circuit (SOIC) package
- Enables the Niosprocessor to access unused flash memory through AS memory interface
- Reprogrammable memory with more than 100,000 erase or program cycles



Pin Diagram

- Write protection support for memory sectors using status register bits
- In-system programming (ISP) support with SRunner software driver
- ISP support with USB-Blaster, EthernetBlaster, or ByteBlaster II download cables
- Additional programming support with the APU and programming hardware from BP Microsystems, System General, and other vendors
- By default, the memory array is erased and thePin Diagram bits are set to 1

Description

To configure a system using an SRAM-based device, each time you power on the device, you must load the configuration data. The EPCS device is a flash memory device that can store configuration data that you use for FPGA configuration purpose after power on. You can use the EPCS device on all FPGA that support AS x1 configuration scheme.
For an 8-pin SOIC package, you can migrate vertically from the EPCS1 device to the EPCS16 device.
With the new data decompression feature supported, you can determine using which EPCS device to store the configuration data for configuring your FPGA.
Example 1 shows how you can calculate the compression ratio to determine which EPCS device is suitable for the FPGA.

Example 1. Compression Ratio Calculation

EPCS16 = 16 ,777 ,216 bits
Preliminary data indicates that compression typically reduces the configuration bitstream size by 35%to 55%.Assume worst case that is 35% decompression.
16 ,777 ,216 bits x 0.65 = 10,905,190 bits The EPCS16 device is suitable.

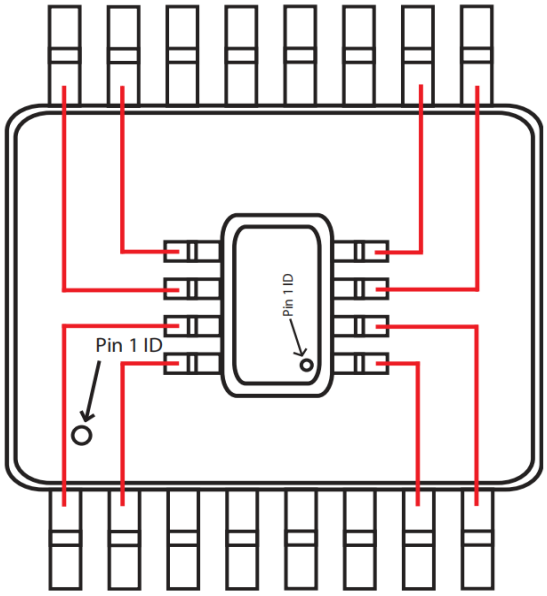
Pin description

Pin Name	Pin Number	Pin Type	Description
DATA	2	Output	The DATA output signal transfers data serially out of the EPCS device to the FPGA during the read operation or configuration. During the read operation or configuration, the EPCS device is enabled by pulling the nCS signal low. The DATA signal transitions on the falling edge of the DCLK signal.
ASDI	5	Input	The ASDI signal is used to transfer data serially into the EPCS device. This pin are also receiving data that are programmed into the EPCS device. Data is latched on the rising edge of the DCLK signal.
nCS	1	Input	The nCS signal toggles at the beginning and the end of a valid instruction. When this signal goes high, the device is deselected and the DATA pin is tri-stated. When this signal goes low, the device is enabled and in an active mode. After power up, the EPCS device requires a falling edge on the nCS signal before the EPCS device begins any operation.
DCLK	6	Input	The FPGA provides the DCLK signal. This signal provides the timing for the serial interface. The data presented on the ASDI pin is latched to the EPCS device on the rising edge of the DCLK signal. The data on the DATA pin changes after the falling edge of the DCLK signal and is latched into the FPGA on the next falling edge of the DCLK signal.
VCC	3.7.8	Power	Connect the power pins to a 3.3-V power supply
GND	4	GND	Ground pin.

Pin Information

The following lists the control pins on the EPCS device:

- Serial data output (DATA)
- AS data input (ASDI)
- Serial clock (DCLK)
- Chip select (nCS)



Layout Recommendation for Vertical Migration from the EPCS1 Device to the EPCS128 Device

Device	Memory Size (bits)	On-Chip Decompressio	ISP Support	Cascading Support	Reprogrammable	Recommended Operating
EPCS16	16,777,216	No	Yes	No	Yes	3.3

Table 1. Altera EPCS Devices

NOTE:
For more information about programming EPCS devices using the Altera Programming Unit (APU) or Master Programming Unit (MPU), refer to the Programming Hardware Datasheet. The EPCS device can be re-programmed in system with ByteBlasterII download cable or an external microprocessor using SRunner. For more information, refer to AN418: SRunner:An Embedded Solution for Serial Configuration Device Programming.

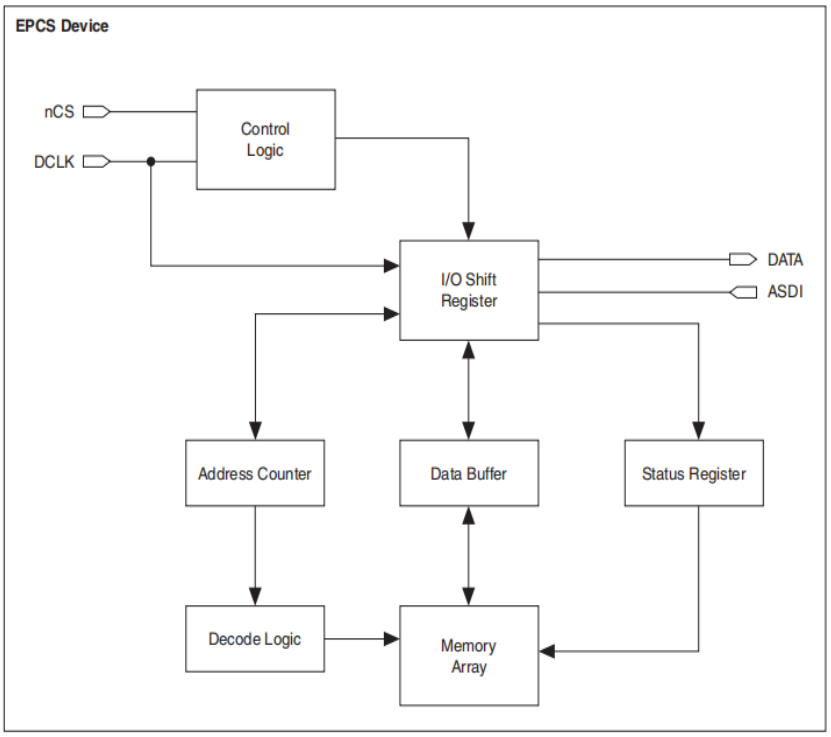


Figure 1. EPCS Device Block Diagram

Accessing Memory in EPCS Devices

You can access the unused memory locations of the EPCS device to store or retrieve data through the Nios processor and SOPC Builder. SOPC Builder is an Altera tool for creating bus-based (especially microprocessor-based) systems in Altera devices.SOPC Builder assembles library components such as processors and memories into custom microprocessor systems. SOPC Builder includes the EPCS device controller core, which is an interface core designed specifically to work with the EPCS device. With this core, you can create a system with a Nios embedded processor that allows software access to any memory location within the EPCS device.

Active Serial FPGA Configuration

The following FPGAs support the AS configuration scheme with EPCS devices:

- Arria®series
- Cyclone®series
- All device families in the Stratix?series except the Stratix device family

There are four signals on the EPCS device that interface directly with the FPGA ’ s control signals. The EPCS device signals are DATA, DCLK,ASDI, and nCS interface with the DATA0, DCLK, ASDO, and nCSO control signals on the FPGA, respectively.

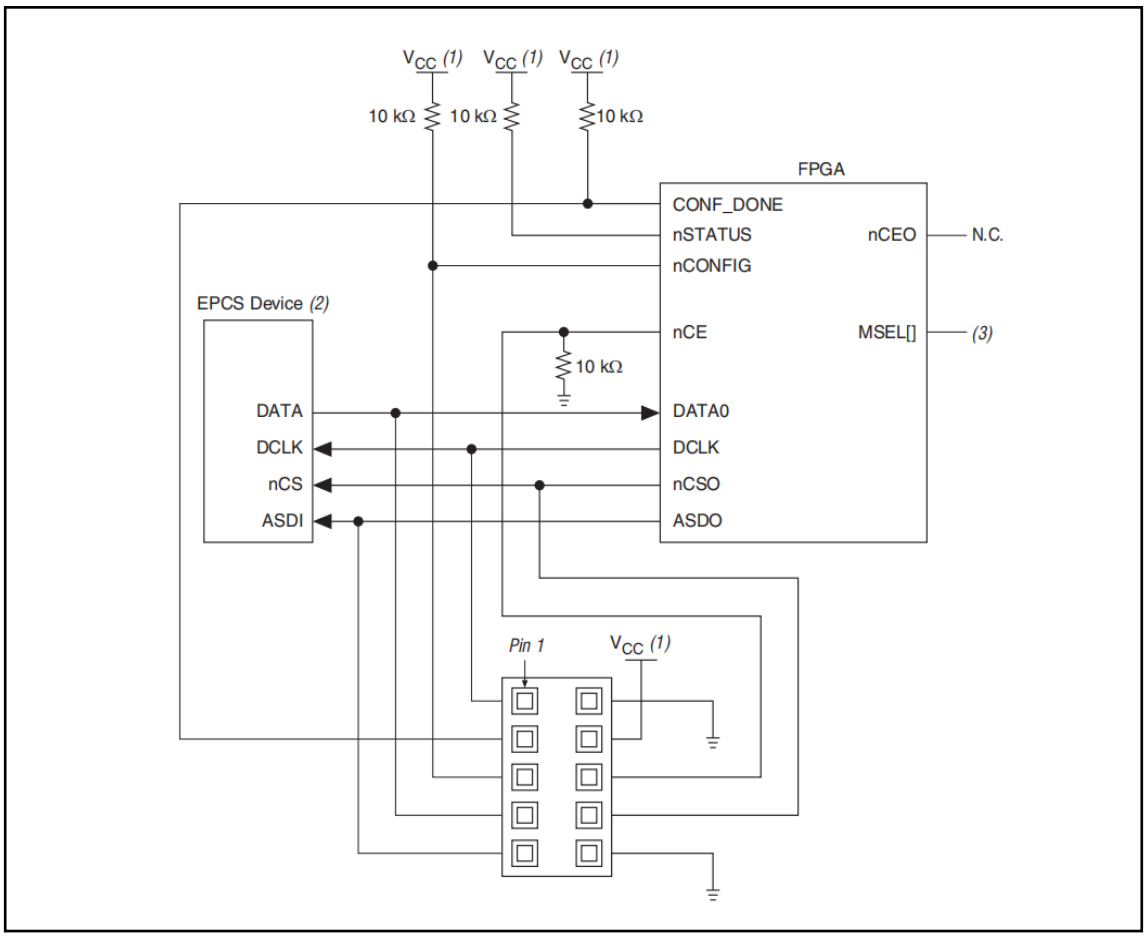


Figure2.FPGA Configuration in AS Mode Using a Download Cable(1),(4)

Notes:

(1) For more information about the Vccvalue, refer to the configuration chapter in the appropriate device handbook.(2) EPCS devices cannot be cascaded.

(3) Connect the MSEL[]input pins to select the AS configuration mode. For more information , refer to the configuration chapter in the appropriate device handbook.

(4) For more information about configuration pin I/O requirements in an AS configuration scheme for an Altera FPGA, refer to the configuration chapter in the appropriate device handbook.

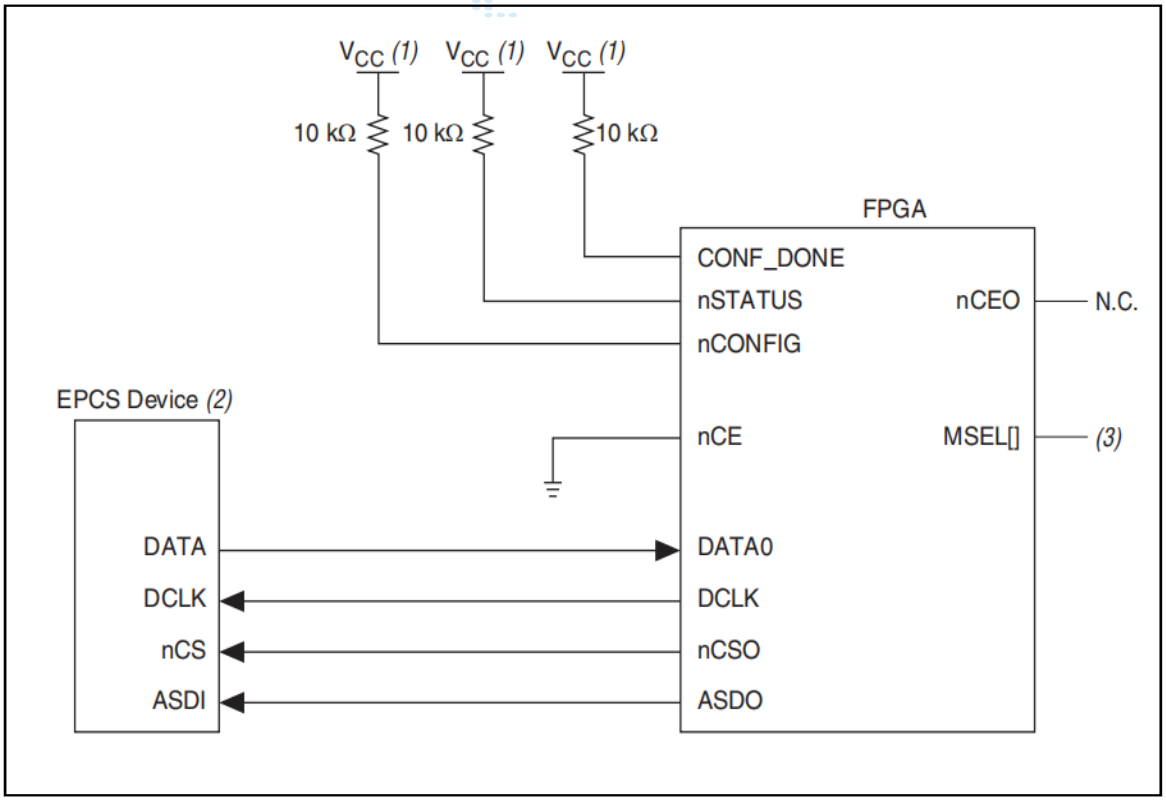


Figure3.Use the APU or a third-party programmer to modify the FPGA configuration (1), (4) in AS mode

Notes:

(1) For more information about the Vccvalue, refer to the configuration chapter in the appropriate device handbook.

(2) EPCS devices cannot be cascaded.

(3) Connect the MSEL[]input pins to select the AS configuration mode. For more information, refer to the configuration chapter in the appropriate device handbook.

(4) For more information about configuration pin I/O requirements in an AS configuration scheme for an Altera FPGA, refer to the configuration chapter in the appropriate device handbook.

In an AS configuration, the FPGA acts as the configuration master in the configuration flow and provides the clock to the EPCS device. The FPGA enables the EPCS device by pulling the nCS signal low using the nCSO signal as shown in Figure 2 and Figure 3. Then,the FPGA sends the instructions and addresses to the EPCS device using the ASDO signal. The EPCS device responds to the instructions by sending the configuration data to the FPGA ' s DATA0 pin on the falling edge of DCLK. The data is latched into the FPGA on the next DCLK signal ' s falling edge.

Before the FPGA enters configuration mode, ensure that $V_{CC}$of the EPCS device is ready. If $V_{CC}$is not ready, you must hold nCONFIG low until all power rails of EPCS device are ready.

The FPGA controls the nSTATUS and CONF_DONE pins during configuration in the AS mode. If the CONF_DONE signal does not go high at the end of configuration, or if the signal goes high too early, the FPGA pulses its nSTATUS pin low to start areconfiguration. If the

configuration is successful, the FPGA releases the CONF_DONE pin, allowing the external 10-kresistor to pull the CONF_DONE signal high. The FPGA initialization begins after the CONF_DONE pin goes high. After the initialization, the FPGA enters user mode.

For more information about configuring the FPGAs in AS configuration mode or other configuration modes, refer to the configuration chapter in the appropriate device handbook.

You can configure multiple devices with a single EPCS device. However, you cannot cascade EPCS devices. To ensure that the programming file size of the cascaded FPGAs does not exceed the capacity of an EPCS device, refer to Table 1 .

Figure 4 shows the AS configuration scheme with multiple FPGAs in the chain. The first FPGA is the configuration master and its MSEL[]pins are set to AS mode. The following FPGAs are configuration slave devices and their MSEL[]pins are set to PS mode.

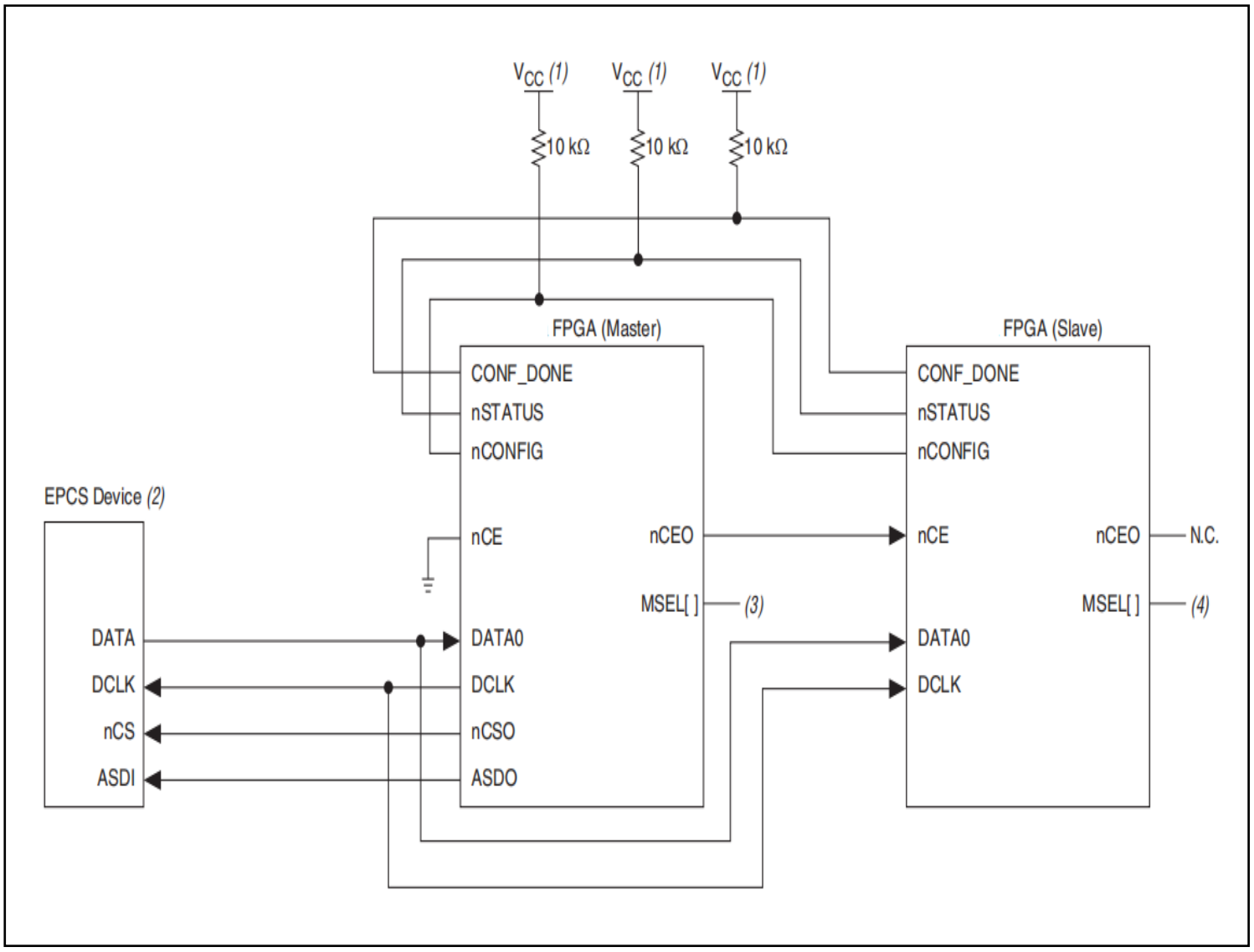


Figure 4. Multiple devices (1), (5) under AS mode

Notes :

(1) For more information about the V_{CC} value, refer to the configuration chapter in the appropriate device handbook.

(2) EPCS devices cannot be cascaded.

(3) Connect the MSEL[]input pins to select the AS configuration mode. For more information, refer to the configuration chapter in the appropriate device handbook.

(4) Connect the MSEL[]input pins to select the PS configuration mode. For more information, refer to the configuration chapter in the appropriate device handbook.

(5) For more information about configuration pin I/O requirements in an AS configuration scheme for an FPGA, refer to the configuration chapter in the appropriate device handbook.

Details	EPCS16
Bytes	2,097,152 bytes (16 Mb)
Number of sectors	32
Bytes per sector	65,536 bytes (512 Kb)
Pages per sector	256
Total number off pages	8,192
Bytes per page	256 bytes

Table 2: Memory Array Organization in EPCS Devices

Operation Codes

This section describes the operations that you can use to access the memory in EPCS devices. Use the DATA, DCLK, ASDI, and nCS signals to access the memory in EPCS devices.When performing the operation, addresses and data are shifted in and out of the device serially,with MSB first.

The device samples the AS data input on the first rising edge of the DCLK after the active low chip select (nCS) input signal is driven low. Shift the operation code, with MSB first, into the EPCS device serially through the AS data input (ASDI) pin. Each operation code bit is latched into the EPCS device on the rising edge of the DCLK.

Different operations require a different sequence of inputs. While executing an operation,you must shift in the desired operation code,followed by the address bytes or data bytes,both address and data bytes, or none of them. The device must drivenCS pin high after the last bit of the operation sequence is shifted in. Table 4 lists the operation sequence for every operation supported by the EPCS devices.For read operations, the data read is shifted out on the DATA pin.You can drive the nCSpin high after any bit of the data-out sequence is shifted out.

For write and erase operations, drive the nCS pin high at a byte boundary that is in a multiple of eight clock pulses. Otherwise, the operation is rejected and not executed.

All attempts to access the memory contents while a write or erase cycle is in progress are rejected, and the write or erase cycle will continue unaffected.

Sector	Address Range (Byte Addresses in HEX)	
	Start	End
31	H'1F0000	H'1FFFFF
30	H'1E0000	H'1EFFFF
29	H'1D0000	H'1DFFFF
28	H'1C0000	H'1CFFFF
27	H'1B0000	H'1BFFFF
26	H'1A0000	H'1AFFFF
25	H'190000	H'19FFFF
24	H'180000	H'18FFFF
23	H'170000	H'17FFFF
22	H'160000	H'16FFFF
21	H'150000	H'15FFFF
20	H'140000	H'14FFFF
19	H'130000	H'13FFFF
18	H'120000	H'12FFFF
17	H'110000	H'11FFFF
16	H'100000	H'10FFFF
15	H'0F0000	H'0FFFFF
14	H'0E0000	H'0EFFFF
13	H'0D0000	H'0DFFFF
12	H'0C0000	H'0CFFFF
11	H'0B0000	H'0BFFFF
10	H'0A0000	H'0AFFFF
9	H'090000	H'09FFFF
8	H'080000	H'08FFFF
7	H'070000	H'07FFFF
6	H'060000	H'06FFFF
5	H'050000	H'05FFFF
4	H'040000	H'04FFFF
3	H'030000	H'03FFFF
2	H'020000	H'02FFFF
1	H'010000	H'01FFFF
0	H'000000	H'00FFFF

Table 3. Address Range for Sectors in EPCS16 Devices

Operation	Operation Code (1)	Address Bytes	Dummy Bytes	Data Bytes	DCLK fAx(MHz)
Write enable	00000110	0	0	0	25
Write disable	00000100	0	0	0	25
Read status	00000101	0	0	1 to infinite(2)	32
Read bytes	00000011	3	0	1 to infinite (2)	20
Read silicon ID (4)	10101011	0	3	1 to infinite (2)	32
Fast read	00001011	3	1	1 to infinite (2)	40
Write status	00000001	0	0	1	25
Write bytes	00000010	3	0	1 to 256(3)	25
Erase bulk	11000111	0	0	0	25
Erase sector	11011000	3	0	0	25
Read device identification	10011111	0	2	1 to infinite (2)	25

Table 4. EPCS Devices Operation Codes

Notes :

(1) List MSB first and LSB last.

(2) The status register, data, or silicon ID is read out at least once on the DATA pin and is continuously read out until the nCS pin is driven high.

(3) A write bytes operation requires at least one data byte on the DATA pin. If more than 256 bytes are sent to the device, only the last 256 bytes are written to the memory.(4) The read silicon ID operation is available only for EPCS16 devices.

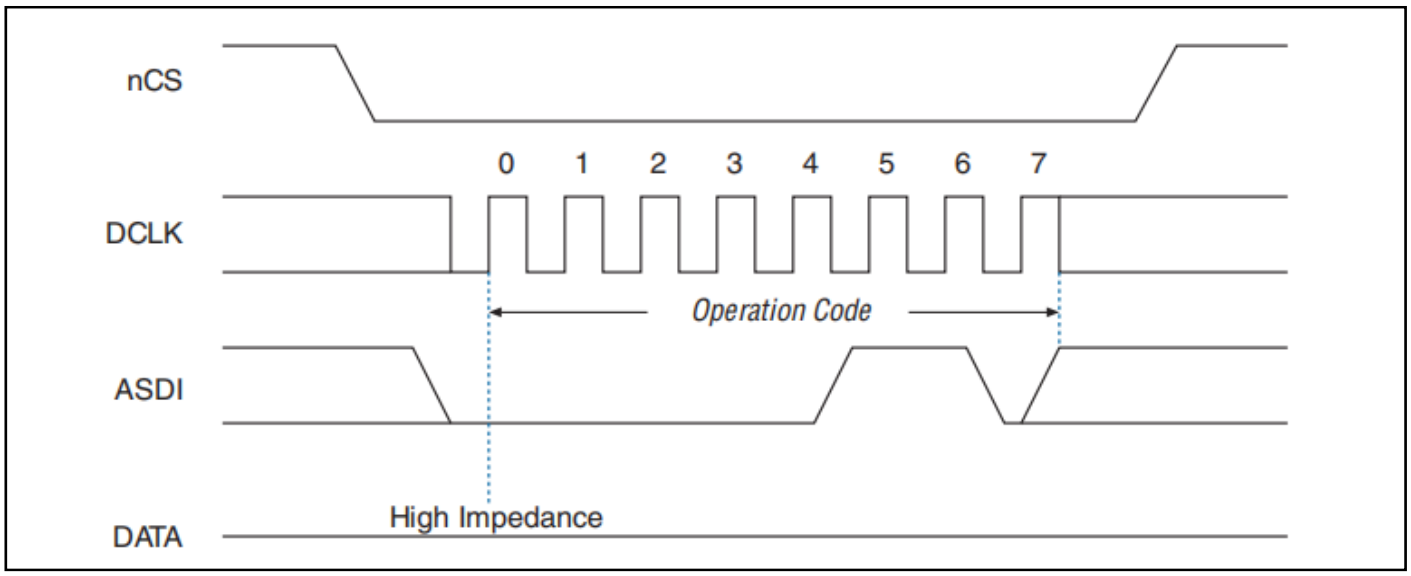


Figure 5. Write Enable Operation Timing Diagram

Write Enable Operation

The write enable operation code is b'0000 0110, and it lists the MSB first. The write enable operation sets the write enable latch bit, which is bit 1 in the status register. Always set the write enable latch bit before write bytes,write status, erase bulk, and erase sector operations. Figure 5 shows the instruction sequence of the write enable operation.

Write Disable Operation

The write disable operation code is b'0000 0100 and it lists the MSB first. The write disable operation resets the write enable latch bit,which is bit 1 in the status register. To prevent the memory from being written unintentionally, the write enable latch bit is automatically reset when implementing the write disable operation, and under the following conditions:

- Power up
- Write bytes operation completion
- Write status operation completion
- Erase bulk operation completion
- Erase sector operation completion

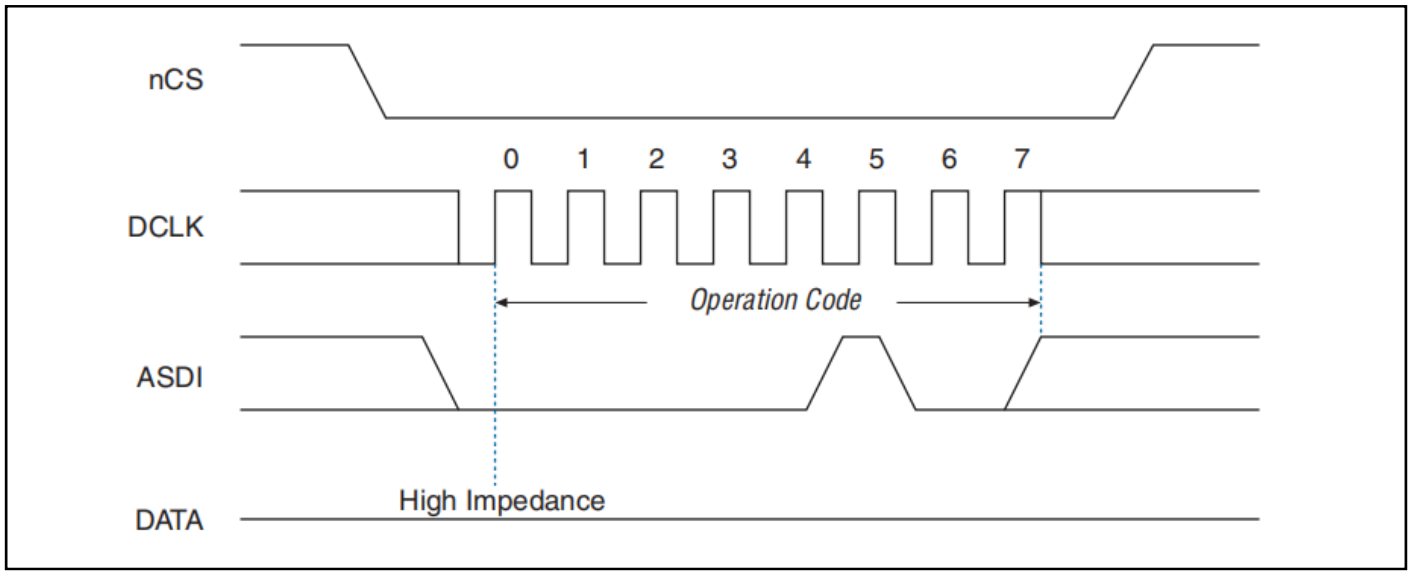


Figure 6. Write Disable Operation Timing Diagram

Read Status Operation

The read status operation code is b'0000 0101 and it lists the MSB first.You can use the read status operation to read the status register. Figure 7 and Figure 8 show the status bits in the status register of the EPCS devices.

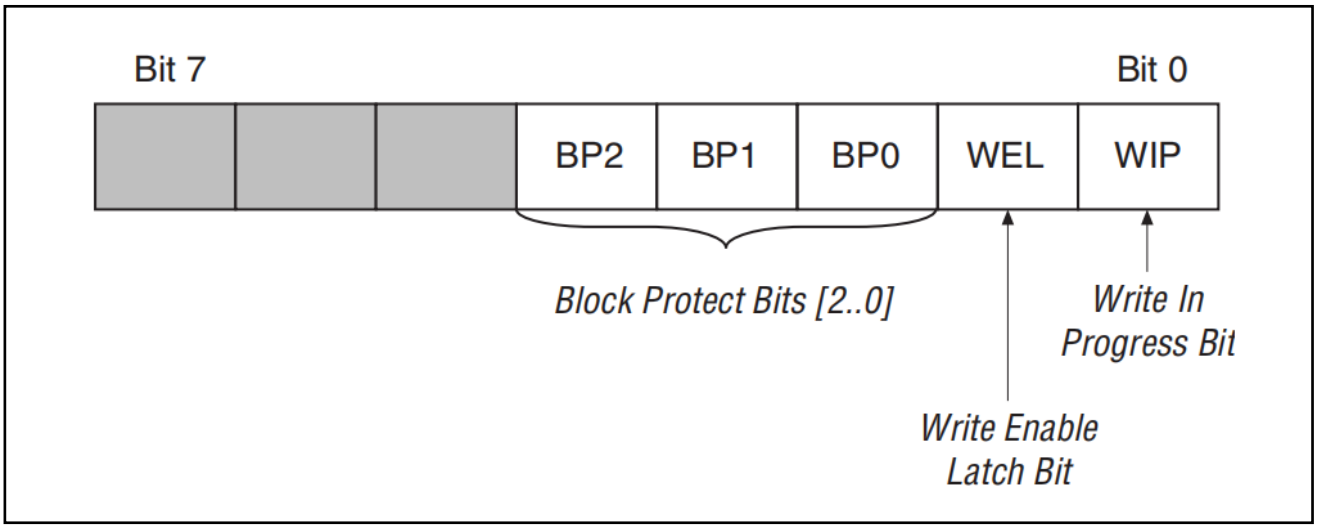


Figure 7. EPCS16 Status Register Status Bits

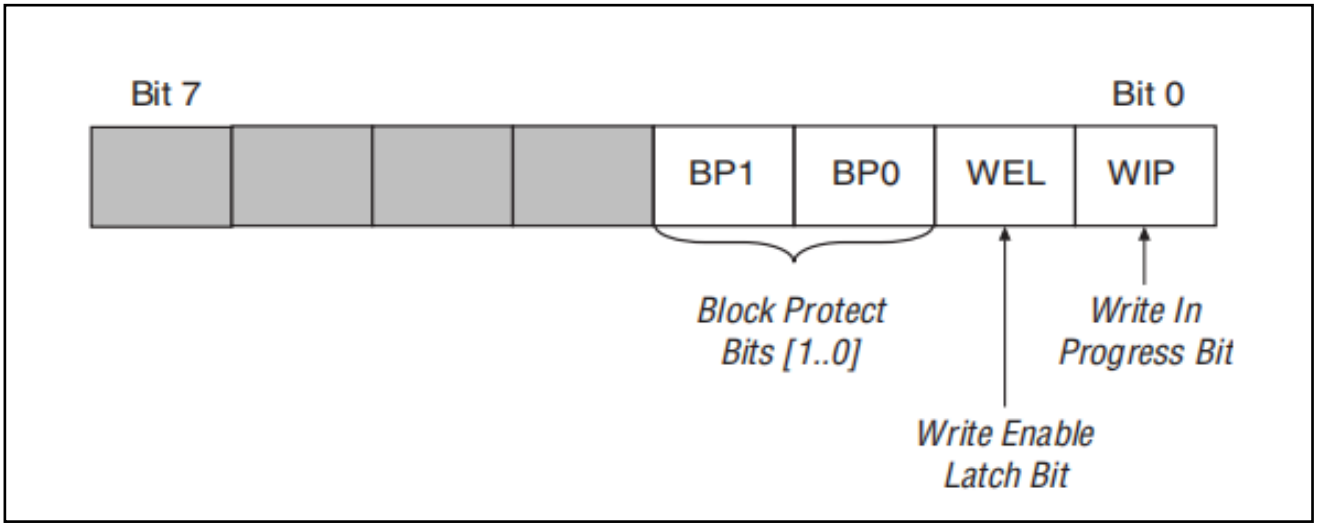


Figure 8.EPCS1 Status Register Status Bits

Setting the write in progress bit to 1 indicates that the EPCS device is busy with a write or erase cycle. Resetting the write in progress bit to 0 indicates no write or erase cycle is in progress.

Resetting the write enable latch bit to 0 indicates that no write or erase cycle is accepted. Set the write enable latch bit to 1 before every write bytes,write status,erase bulk, and erase sector operations.

The non-volatile block protect bits determine the area of the memory protected from being written or erased unintentionally. Table 5 list the protected area in the EPCS devices with reference to the block protect bits. The erase bulk operation is only available when all the block protect bits are set to 0. When any of the block protect bits are set to 1, the relevant area is protected from being written by a write bytes operation or erased by an erase sector operation.

Status Register Content			Memory Content	
BP2 Bit	BP1 Bit	BPO Bit	Protected Area	Unprotected Area
0	0	0	None	All sectors(32 sectors O to 31)
0	0	1	Upper 32nd (Sector 31)	Lower 31/32nds (31 sectors—0 to 30)
0	1	0	Upper sixteenth (two sectors—30 and 31)	Lower 15/16ths (30 sectors—0 to 29)
0	1	1	Upper eighth (four sectors—28 to 31)	Lower seven-eighths (28 sectors—0 to 27)
1	0	0	Upper quarter (eight sectors—24 to 31)	Lower three-quarters (24 sectors—0 to 23)
1	0	1	Upper half (sixteen sectors—16 to 31)	Lower half (16 sectors—0 to 15)
1	1	0	Allsectors(32 sectors—0 to 31)	None
1	1	1	All sectors (32 sectors—0 to 31)	None

Table 5. Block Protection Bits in the EPCS16 Device

You can read the status register at any time, even during a write or erase cycle is in progress. When one of these cycles is in progress, you can check the write in progress bit (bit 0 of the status register)before sending a new operation to the device. The device can also read the status register continuously, as shown in Figure 9.

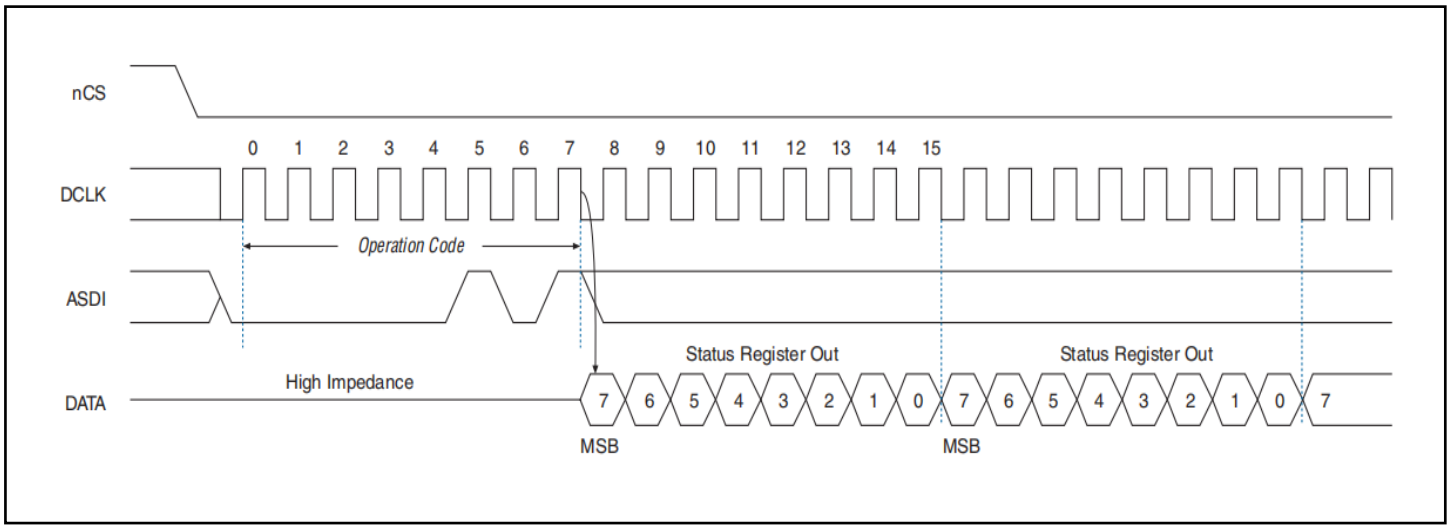


Figure 9. Read Status Operation Timing Diagram

Write Status Operation

The write status operation code is b'0000 0001 and it lists the MSB first. Use the write status operation to set the status register block protection bits. The write status operation does not affect the other bits. Therefore, you can implement this operation to protect certain memory sectors, as listed in Table 5. After setting the block protect bits, the protected memory sectors are treated as read-only memory. You must execute the write enable operation before the write status operation so the device sets the status register ' s write enable latch bit to 1.

The write status operation is implemented by driving the nCS signal low, followed by shifting in the write status operation code and one data byte for the status register on the ASDI pin. Figure 10 shows the instruction sequence of the write status operation. The nCS must be driven high after the eighth bit of the data byte has been latched in, otherwise the write status operation is not executed.

Immediately after the nCS signal drives high, the device initiates the self-timed write status cycle . The self-timed write status cycle usually takes 5 ms for all EPCS devices and is guaranteed to be less than 15 ms.For more information,refer to the tWSvalue in Table 7.You must account for this delay to ensure that the status register is written with desired block protect bits. Alternatively, you can check the write in progress bit in the status register by executing the read status operation while the self-timed write status cycle is in progress. The write in progress bit is 1 during the self-timed write status cycle and 0 when it is complete.

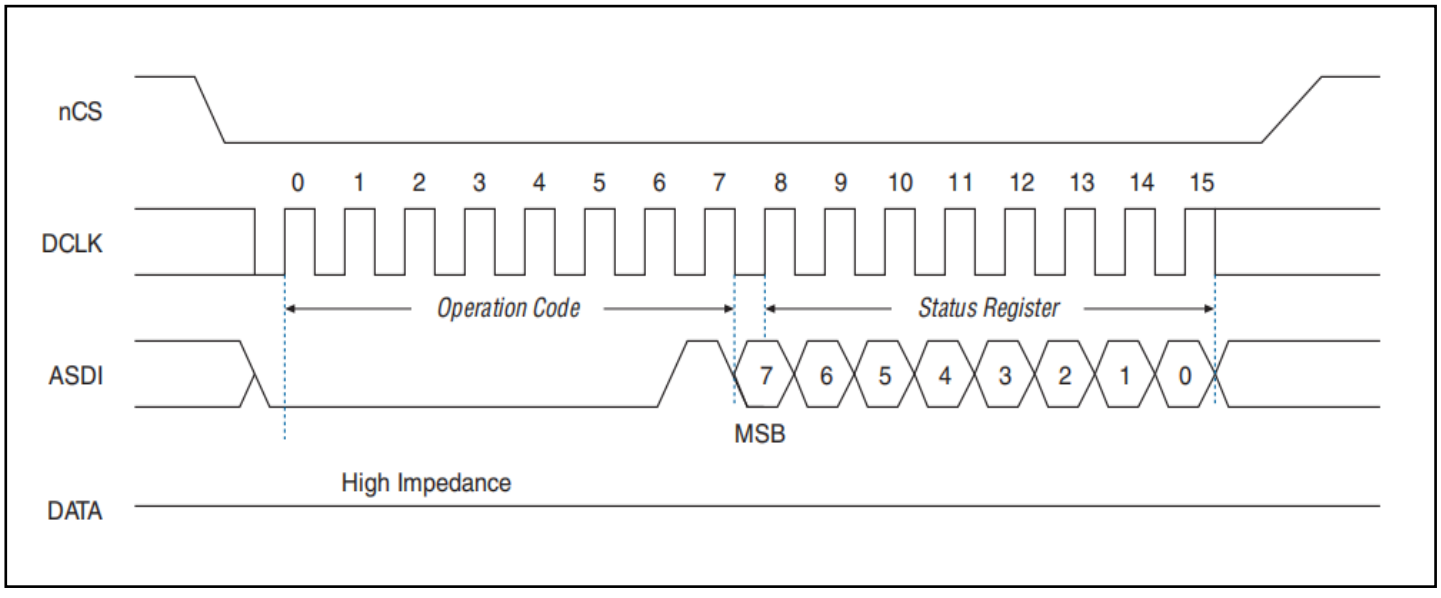


Figure 10. Write Status Operation Timing Diagram

Read Bytes Operation

The read bytes operation code is b'0000 0011 and it lists the MSB first. To read the memory contents of the EPCS device, the device is first selected by driving the nCS signal low. Then, the read bytes operation code is shifted in followed by a 3-byte address (A[23 ..0]). Each address bit must be latched in on the rising edge of the DCLK signal. After the address is latched in, the memory contents of the specified address are shifted out serially on the DATA pin,beginning with the MSB. For reading Raw Programming Data files (**.rpd**), the content is shifted out serially beginning with the LSB. Each data bit is shifted out on the falling edge of the DCLK signal. The maximum DCLK frequency during the read bytes operation is 20 MHz.

The first byte address can be at any location. The device automatically increases the address to the next higher address after shifting out each byte of data. Therefore, the device can read the whole memory with a single read bytes operation. When the device reaches the highest address, the address counter restarts at 0x000000, allowing the memory contents to be read out indefinitely until the read bytes operation is terminated by driving the nCS signal high. The device can drive the nCS signal high at any time after data is shifted out. If the read bytes operation is shifted in while a write or erase cycle is in progress,the operation is not executed and does not affect the write or erase cycle in progress.

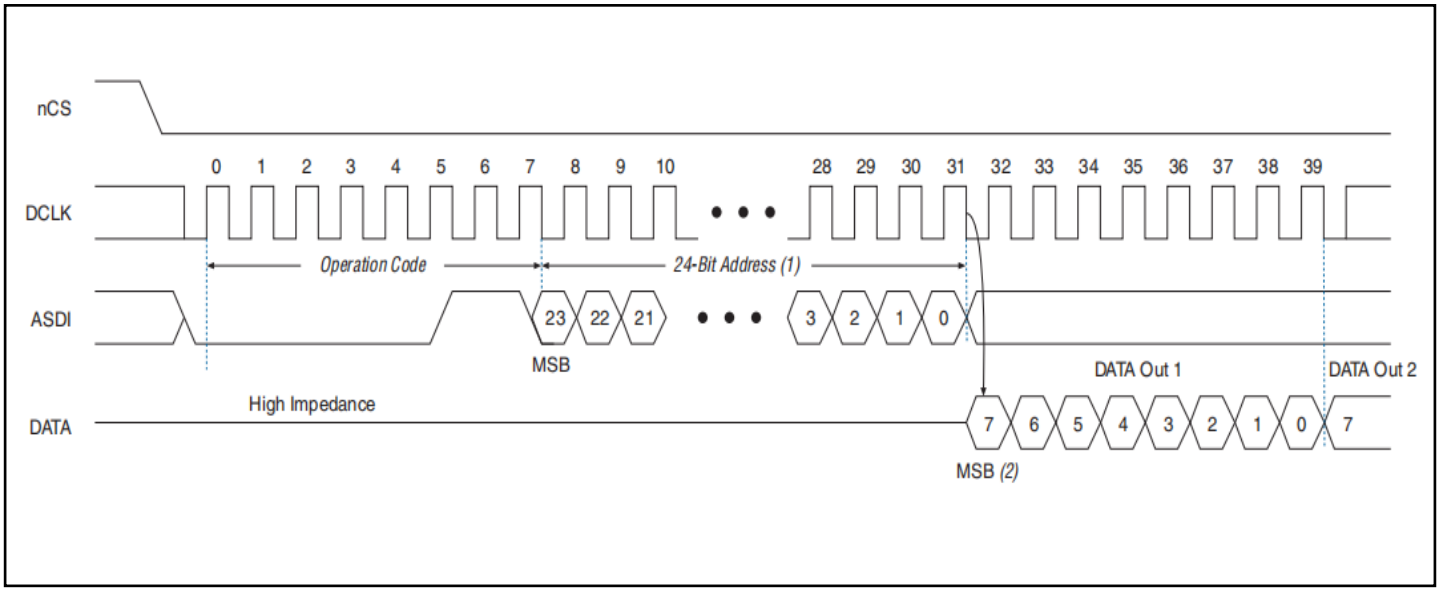


Figure 11. Read Bytes Operation Timing Diagram

Notes :

(1) Address bits A[23 ..21]are don't-care bits in the EPCS16 device.

(2) For .rpd files, the read sequence shifts out the LSB of the data byte first.

Fast Read Operation

The fast read operation code is b ’ 0000 1011 and it lists the MSB first. You can select the device by driving the nCS signal low. The fast read instruction code is followed by a 3-byte address (A23 -A0) and a dummy byte with each bit being latched-in during the rising edge of the DCLK signal. Then,the memory contents at that address is shifted out on DATA with each bit being shifted out at a maximum frequency of 40 MHz during the falling edge of the DCLK signal.

The first addressed byte can be at any location. The address is automatically increased to the next higher address after each byte of data is shifted out. Therefore, the whole memory can be read with a single fast read instruction. When the highest address is reached, the address counter rolls over to 000000h, allowing the read sequence to continue indefinitely.

The fast read instruction is terminated by driving the nCS signal high at any time during data output. Any fast read instruction is rejected during the erase,program, or write operations without affecting the operation that is in progress.

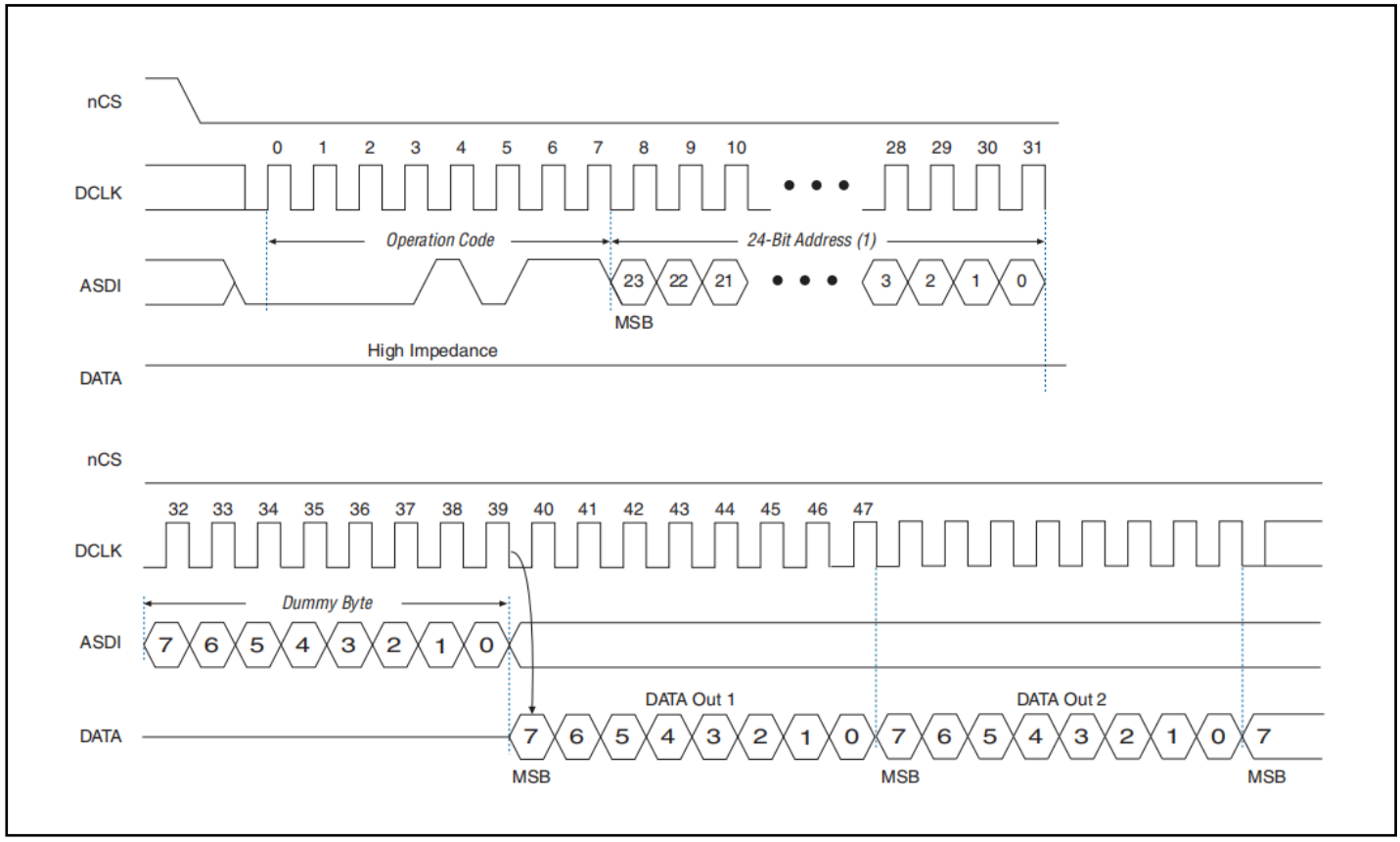


Figure 12. Fast Read Operation Timing Diagram

Note :

(1) Address bits A[23 ..21]are don't-care bits in the EPCS16 device.

Read Silicon ID Operation

The read silicon ID operation code is b'1010 1011 and it lists the MSB first. Only EPCS16 devices support this operation. This operation reads the 8-bit silicon ID of the EPCS device from the DATA output pin.If this operation is shifted in during an erase or write cycle, it is ignored and does not affect the cycle that is in progress.Table 6 lists the EPCS device silicon IDs.

EPCS Device	Silicon ID (Binary Value)
EPCS16	b'00010100

Table 6. EPCS Device Silicon ID

The device implements the read silicon ID operation by driving the nCS signal low and then shifting in the read silicon ID operation code, followed by three dummy bytes on the ASDI pin. The 8-bit silicon ID of the EPCS device is then shifted out on the DATA pin on the falling edge of the DCLK signal. The device can terminate the read silicon ID operation by driving the nCS signal high after reading the silicon ID at least one time. Sending additional clock cycles on DCLK while nCS is driven low can cause the silicon ID to be shifted out repeatedly.

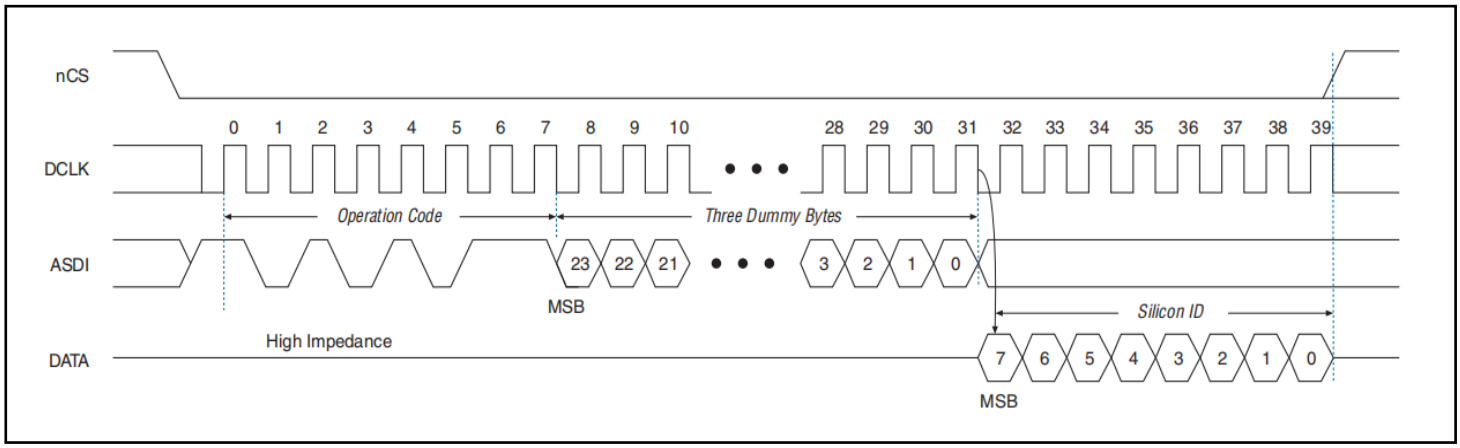


Figure 13. Read Silicon ID Operation Timing Diagram (1)

Note :
(1) Only EPCS1, EPCS4, EPCS16, and EPCS64 devices support the read silicon ID operation.

Write Bytes Operation

The write bytes operation code is b'0000 0010 and it lists the MSB first. This operation allows bytes to be written to the memory.You must execute the write enable operation before the write bytes operation to set the write enable latch bit in the status register to 1.
The write bytes operation is implemented by driving the nCS signal low, followed by the write bytes operation code, three address bytes, and at least one data byte on the ASDI pin. If the eight LSBs (A[7 ..0]) are not all 0, all sent data that goes beyond the end of the current page is not written into the next page. Instead,this data is written at the start address of the same page (from the address whose eight LSBs are all 0). You must ensure the nCS signal is set low during the entire write bytes operation.
If more than 256 data bytes are shifted into the EPCS device with a write bytes operation, the previously latched data is discarded and the last 256 bytes are written to the page.However, if less than 256 data bytes are shifted into the EPCS device, they are guaranteed to be written at the specified addresses and the other bytes of the same page are not affected.
If your design requires writing more than 256 data bytes to the memory, more than one page of memory is required. Send the write enable and write bytes operation codes, followed by three new targeted address bytes and 256 data bytes,before a new page is written.
The nCS signal must be driven high after the eighth bit of the last data byte has been latched in. Otherwise,the device does not execute the write bytes operation. The write enable latch bit in the status register is reset to 0 before the completion of each write bytes operation. Therefore, the write enable operation must be carried out before the next write bytes operation.
The device initiates a self-timed write cycle immediately after the nCS signal is driven high.For more information about the self-timed write cycle time,refer to the tWBvalue in Table 7.You must account for this amount of delay before another page of memory is written. Alternatively, you can check the write in progress bit in the status register by executing the read status operation while the self-timed write cycle is in progress.The write in progress bit is set to 1 during the self-timed write cycle and0 when it is complete.
You must erase all the memory bytes of the EPCS devices to all 1 or 0xFF before you implement the write bytes operation.You can erase all the memory bytes by executing the erase sector operation in a sector or the erase bulk operation throughout the entire memory.

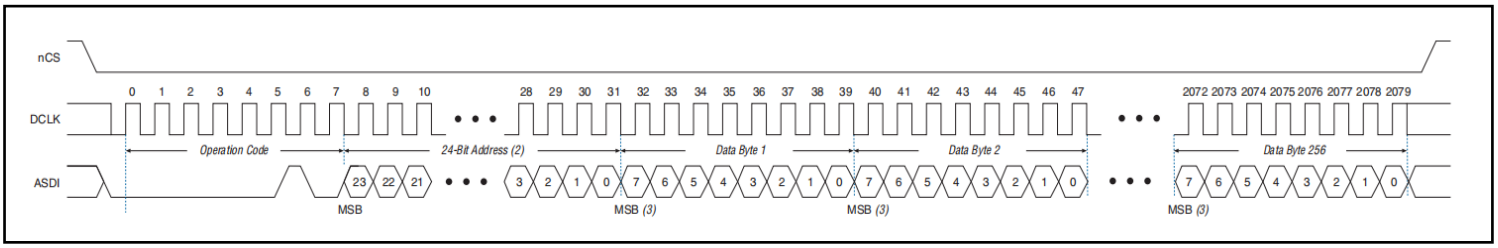


Figure 15. Write Bytes Operation Timing Diagram (1)

Notes :

(1) Use the erase sector operation or the erase bulk operation to initialize the memory bytes of the EPCS devices to all 1 or 0xFF before implementing the write bytes operation.(2) Address bits A[23 .. 21] are don't-care bits in the EPCS16 device. (3) For . rpd files, write the LSB of the data byte first.

Erase Bulk Operation

The erase bulk operation code is b'1100 0111 and it lists the MSB first. This operation sets all the memory bits to 1 or 0xFF. Similar to the write bytes operation, you must execute the write enable operation before the erase bulk operation so that the write enable latch bit in the status register is set to 1.

You can implement the erase bulk operation by driving the nCS signal low and then shifting in the erase bulk operation code on the ASDI pin. The nCS signal must be driven high after the eighth bit of the erase bulk operation code has been latched in.

The device initiates a self-timed erase bulk cycle immediately after the nCS signal is driven high. For more information about the self-timed erase bulk cycle time, refer to the tEBvalue in Table 7.

You must account for this delay before accessing the memory contents. Alternatively, you can check the write in progress bit in the status register by executing the read status operation while the self-timed erase cycle is in progress. The write in progress bit is set to 1 during the self-timed erase cycle and 0 when it is complete. The write enable latch bit in the status register is reset to 0 before the erase cycle is complete.

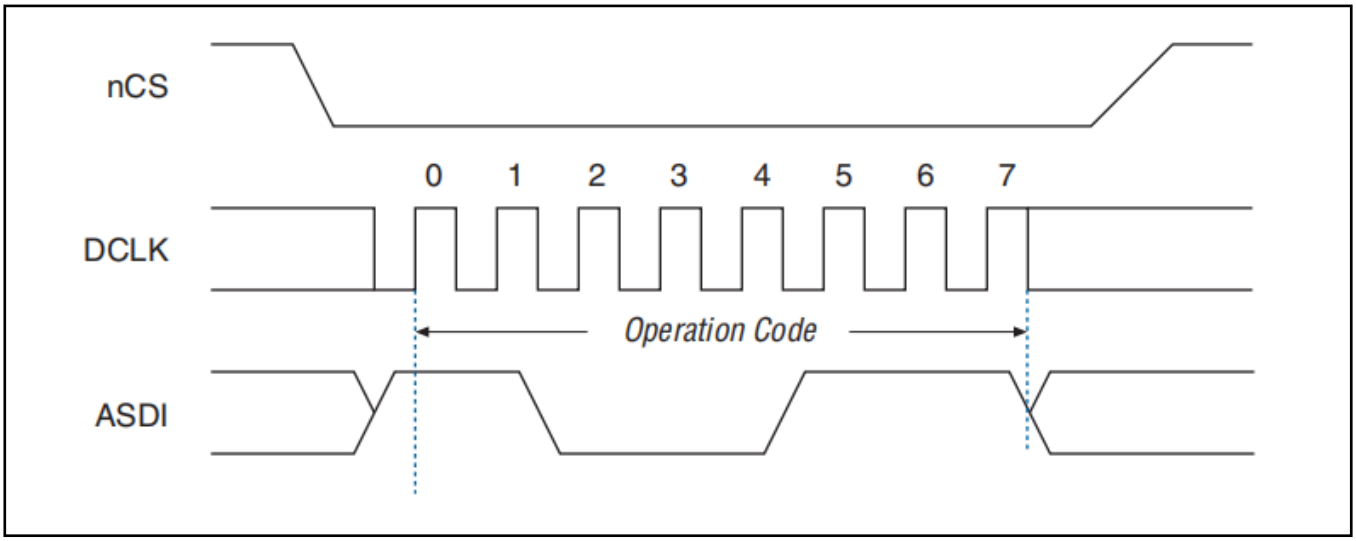


Figure 16. Erase Bulk Operation Timing Diagram

Erase Sector Operation

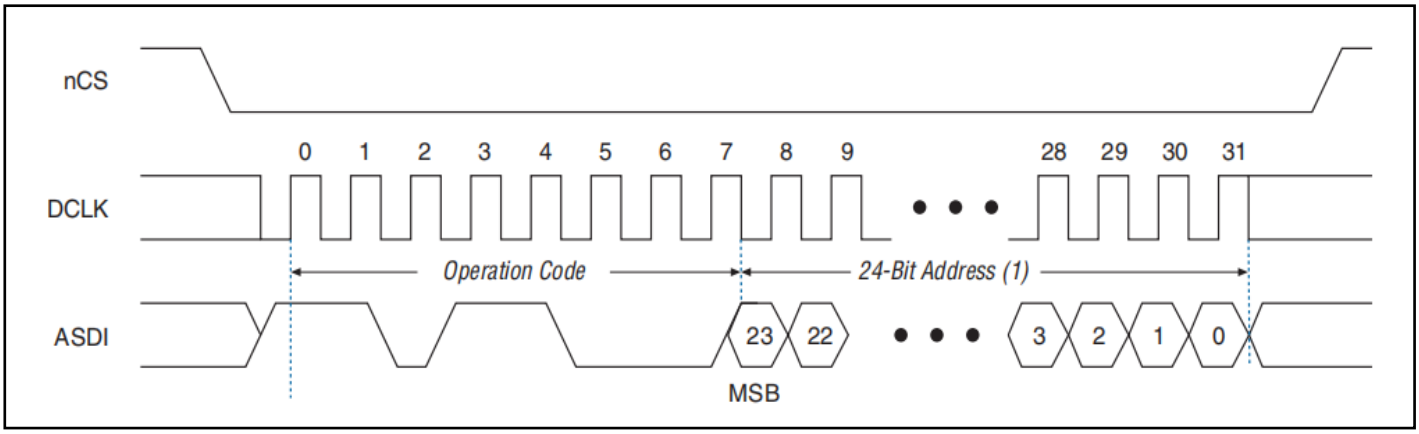
The erase sector operation code is b'1101 1000 and it lists the MSB first. This operation allows you to erase a certain sector in the EPCS device by setting all the bits inside the sector to 1 or 0xFF. This operation is useful if you want to access the unused sectors as general purpose memory in your applications. You must execute the write enable operation before the erase sector operation so that the write enable latch bit in the status register is set to 1.

You can implement the erase sector operation by first driving the nCS signal low, then you shift in the erase sector operation code, followed by the three address bytes of the chosen sector on the ASDI pin. The three address bytes for the erase sector operation can be any

address inside the specified sector. For more information about the sector address range, refer to Table 3 . Drive the nCS signal high after the eighth bit of the erase sector operation code has been latched in.

The device initiates the self-timed erase sector cycle immediately after the nCS signal is driven high.For more information about the self-timed erase sector cycle time,refer to the tESvalue in Table 7.

You must account for this delay before accessing the memory contents. Alternatively, you can check the write in progress bit in the status register by executing the read status operation while the self-timed erase sector cycle is in progress. The write in progress bit is set to 1 during the self-timed erase sector cycle and 0 when it is complete. The write enable latch bit in the status register resets to 0 before the erase cycle is complete.Figure 17 shows the instruction sequence of the erase sector operation.



Note :

(1) Address bits A[23 ..21]are don't-care bits in the EPCS16 device.

Power and Operation

This section describes the power modes, power-on reset (POR) delay, error detection, and initial programming state of the EPCS devices.

Power Mode

EPCS devices support active and standby power modes. When the nCS signal is low, the device is enabled and is in active power mode. The FPGA is configured while the EPCS device is in active power mode. When the nCS signal is high, the device is disabled but remains in active power mode until all internal cycles are completed, such as write or erase operations. The EPCS device then goes into standby power mode. The ICC1and ICC0parameters list the VCCsupply current when the device is inactive and standby power modes. For more information, refer to Table 12.

Power-On Reset

During the initial power-up, a POR delay occurs to ensure the system voltage levels have stabilized. During the AS configuration, the FPGA controls the configuration and has a longer POR delay than the EPCS device.
For more information about the POR delay time, refer to the configuration chapter in the appropriate device handbook.

Error Detection

During the AS configuration with the EPCS device, the FPGA monitors the configuration status through the nSTATUS and CONF_DONE pins. If an error condition occurs, if the nSTATUS pin drives low or if the CONF_DONE pin does not go high, the FPGA begins reconfiguration by pulsing the nSTATUS and nCSO signals,which controls the chip select (nCS) pin on the EPCS device.After an error, the configuration automatically restarts if the Auto-Restart Upon Frame Error option is turned on in the Quartus?II software.If the option is turned off, the system must monitor the nSTATUS signal for errors and then pulse the nCONFIG signal low to restart configuration.

Timing Information

Figure 18 shows the timing waveform for the write operation of the EPCS device.

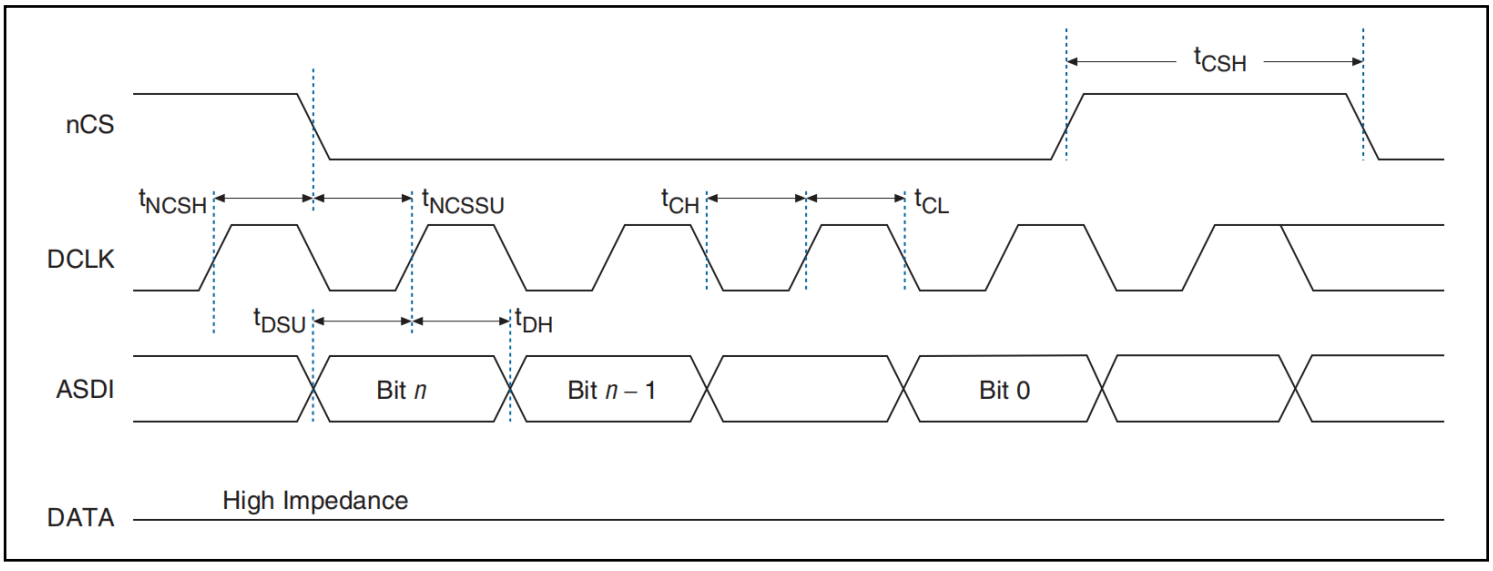


Figure 18. Write Operation Timing Diagram

lists the EPCS device timing parameters for the write operation.

Symbol	Parameter	Min	Typ	Max	Unit
fwCLK	Write clock frequency (from the FPGA,download cable,or embedded processor)for write enable,write disable,read status,read silicon ID,write bytes,erase bulk,and erase sector operations	—	—	25	MHz
tcH	DCLK high time	20	—	—	ns
tcL	DCLK low time	20	—	—	ns
tNCSSU	Chip select(ncs)setup time	10	—	—	ns
tNCSH	Chip select (ncs)hold time	10	—	—	ns
tosu	Data(ASDI)in setup time before the rising edge on DCLK	5	—	—	ns
tDH	Data(ASDI)hold time after rising edge on DCLK	5	—	—	ns
tcsH	Chip select(ncs)high time	100	—	—	ns
twB(1)	Write bytes cycle time	—	1.5	5	ms
tws(1	Write status cycle time	—	5	15	ms
	Erase bulk cycle time	—	17	40	S
tes(1)	Erase sector cycle time	—	2	3	S

Table 7. Write Operation Parameters

shows the timing waveform for the read operation of the EPCS device.

Symbol	Parameter	Min	Max	Unit
fRCLK	Read clock frequency(from the FPGA or embedded processor)for the read bytes operation	—	20	MHz
	Fast read clock frequency(from the FPGA or embedded processor)for the fast read bytes operation	—	40	MHz
tcH	DCLK high time	11	—	ns
tcL	DCLK low time	11	—	ns
toDIs	Output disable time after read	—	8	ns
tnCLK2D	Clock falling edge to DATA	—	Q	ns

Table 8. Read Operation Parameters

lists the EPCS device timing parameters for the read operation.

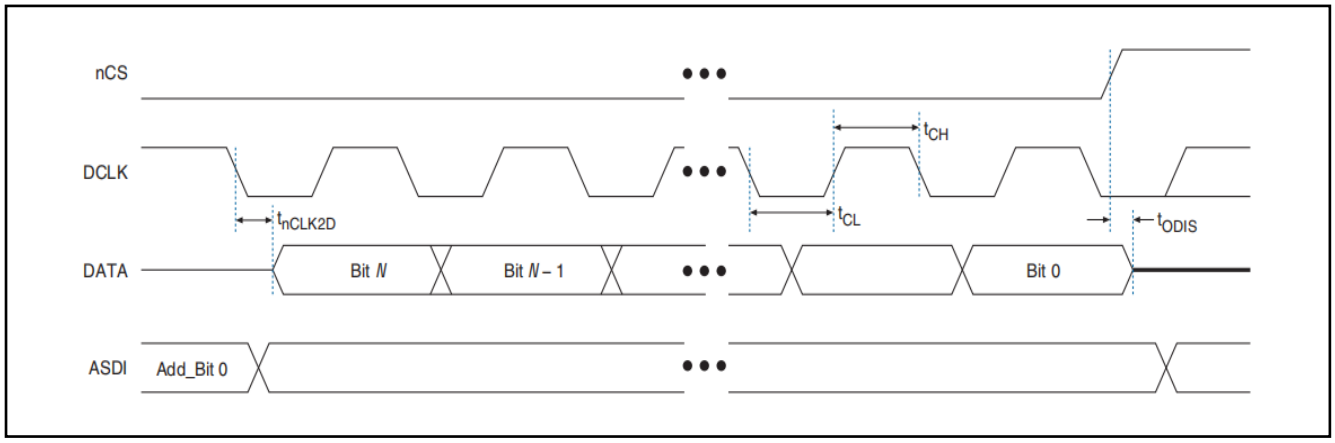


Figure 19. Read Operation Timing Diagram

shows the timing waveform for the AS configuration scheme of the FPGA using an EPCS device.

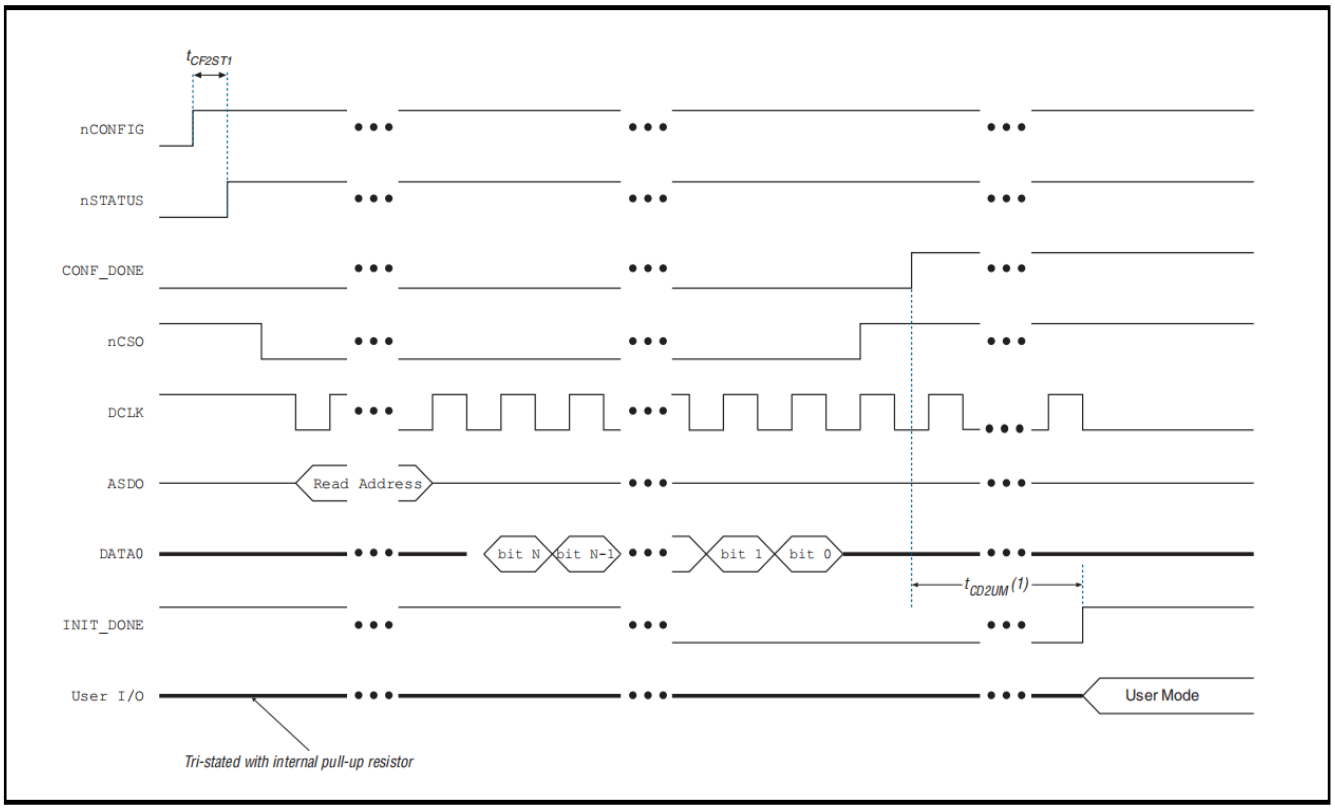


Figure 20. AS Configuration Timing Diagram

Note :

(1) t_{CD2UM} is an FPGA-dependent parameter. For more information, refer to the configuration chapter in the appropriate device handbook.For more information about the timing parameters in Figure 20, refer to the configuration chapter in the appropriate device handbook.

Programming and Configuration File Support

The Quartus II software provides programming support for EPCS devices. When you select an EPCS device, the Quartus II software automatically generates the Programmer Object File (.pof) to program the device. The software allows you to select the appropriate EPCS device density that most efficiently stores the configuration data for the selected FPGA.
You can program the EPCS device in-system by an external microprocessor using the SRunner software driver. The SRunner software driver is developed for embedded EPCS device programming that you can customize to fit in different embedded systems. The SRunner software driver reads .rpd files and writes to the EPCS devices.
The programming time is comparable to the Quartus II software programming time. Because the FPGA reads the LSB of the .rpd data first during the configuration process, the LSB of .rpd bytes must be shifted out first during the read bytes operation and shifted in first during the write bytes operation.
Writing and reading the .rpd file to and from the EPCS device is different from the other data and address bytes.
For more information about the SRunner software driver, refer to AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming.You can program EPCS devices using the APU with the appropriate programming adapter, such as PLMSEPC-8, using the Quartus II software or the USB-Blaster, EthernetBlaster, or ByteBlaster II download cable. In addition, many third-party programmers, such as the BP Microsystems and System General programmers, offer programming hardware that supports EPCS devices.
During the ISP of an EPCS device using the USB-Blaster, EthernetBlaster, or ByteBlaster II download cable, the cable pulls the nCONFIG signal low to reset the FPGA and overrides the 10-kΩ pull-down resistor on the nCE pin of the FPGA, as shown in Figure 2 . The download cable then uses the four interface pins—DATA, nCS, ASDI, and DCLK—to program the EPCS device. When programming is complete, the download cable releases the four interface pins of the EPCS device and the nCE pin of the FPGA and pulses the nCONFIG signal to start the configuration process.
The FPGA can program the EPCS device in-system using the JTAG interface with the SFL. This solution allows you to indirectly program the EPCS device using the same JTAG interface that is used to configure the FPGA.

Operating Conditions

Table 9 through Table 13 list information about the absolute maximum ratings, recommended operating conditions, DC operating conditions, and capacitance for EPCS devices.

Symbol	Parameter	Condition	Min	Max	Unit
Vcc	Supply voltage	With respect to GND	-0.6	4.0	V
V_1	DC input voltage	With respect to GND	-0.6	4.0	V
IMAX	DC Vcc or GND current	—	—	15	mA
IouT	DC output current per pin	—	-25	25	mA
PD	Power dissipation	—	—	54	mW
TsTG	Storage temperature	No bias	-65	150	C
TAMB	Ambient temperature	Under bias	-65	135	C
Tj	Junction temperature	Under bias	—	135	℃

Table 9. Absolute Maximum Ratings (1)

Symbol	Parameter	Conditions	Min	Max	Unit
Vcc	Supply voltage	(2)	2.7	3.6	V
V_1	Input voltage	With respect to GND	-0.3	0.3+Vcc	V
Vo	Output voltage	—	0	Vcc	V
TA	Operating temperature	For industrial use	-40	85	C
tR	Input rise time	—	—	5	ns
t	Input fall time	—	—	5	ns

Table 10. Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
VH	High-level input voltage	—	0.6×Vcc	Vcc+0.4	V
V_1 L	Low-level input voltage	—	-0.5	0.3×Vcc	V
VoH	High-level output voltage	loH=-100μA (3)	Vcc-0.2	—	V
VoL	Low-level output voltage	loL=1.6mA(3)	—	0.4	V
	Input leakage current	V_1=Vcc or GND	-10	10	μA
loz	Tri-state output off-state current	Vo=Vcc or GND	-10	10	μA

Table 11. DC Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
Icco	Vcc supply current (standby mode)	—	—	50	μA
Icc1	Vcc supply current (during active power mode)	—	5	15	mA

Table 12. ICC Supply Current

Symbol	Parameter	Conditions	Min	Max	Unit
CIn	Input pin capacitance	VIN=0V	—	6	pF
CouT	Output pin capacitance	VouT=0V	—	8	pF

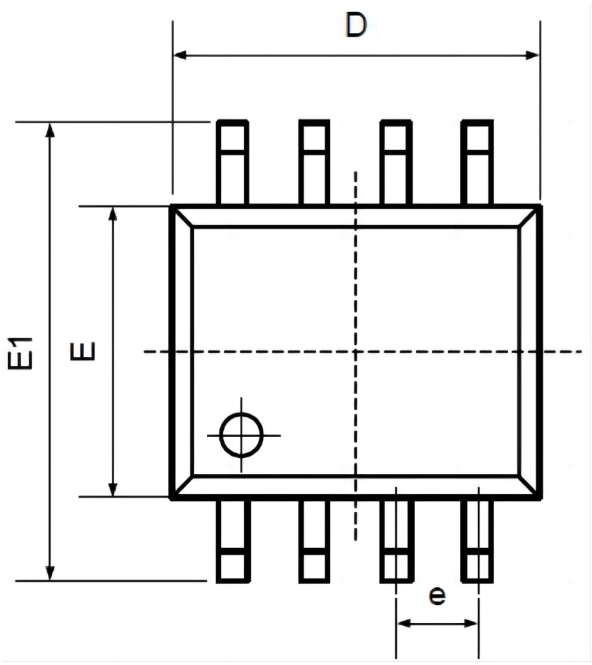
Table 13. Capacitance (4)

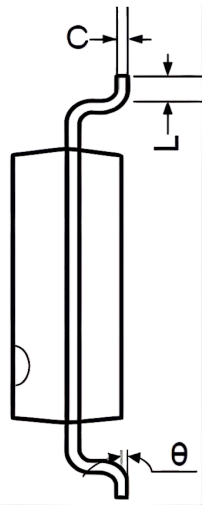
Notes :
(1) For more information, refer to the Operating Requirements for Altera Devices Data Sheet.
(2) Maximum VCC rise time is 100 ms.
(3) The IOH parameter refers to the high-level TTL or CMOS output current and the I OL parameter refers to the low-level TTL or CMOS output current.
(4) Capacitance is sample-tested only at TA = 25 × C and at a 20-MHz frequency

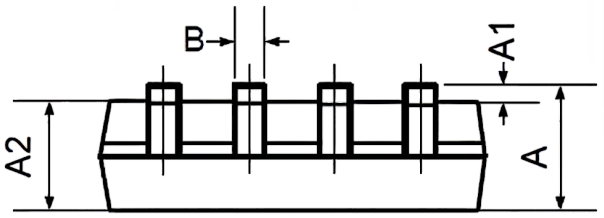
Order information

Order Number	Package	Package Quantity	Marking On The park
EPCS16SI8N-TUDI	SOP8	Tape,Reel,2500	EPCS16N

Package SOP8







Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	1.350	1.750	0.053	0.069
A1	0.100	0.250	0.004	0.010
A2	1.350	1.550	0.053	0.061
B	0.330	0.510	0.013	0.020
C	0.190	0.250	0.007	0.010
D	4.780	5.000	0.188	0.197
E	3.800	4.000	0.150	0.157
E1	5.800	6.300	0.228	0.248
e	1.270TYP		0.050TYP	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°

Important statement:

●TUDI Semiconductor reserves the right to modify the product manual without prior notice! Before placing an order, customers need to confirm whether the obtained information is the latest version and verify the completeness of the relevant information.

●Any semi-guide product is subject to failure or malfunction under specified conditions. It is the buyer's responsibility to comply with safety standards when using TUDI Semiconductor products for system design and whole machine manufacturing.And take the appropriate safety measures to avoid the potential in the risk of loss of personal injury or loss of property situation!

●TUDI Semiconductor products have not been licensed for life support, military, and aerospace applications,and therefore TUDI Semiconductor is not responsible for any consequences arising from the use of this product in these areas.

●If any or all TUDI Semiconductor products (including technical data, services) described or contained in this document are subject to any applicable local export control laws and regulations, they may not be exported without an export license from the relevant authorities in accordance with such laws.

●The specifications of any and all TUDI Semiconductor products described or contained in this document specify the performance, characteristics, and functionality of said products in their standalone state, but do not guarantee the performance, characteristics, and functionality of said products installed in Customer's products or equipment. In order to verify symptoms and conditions that cannot be evaluated in a standalone device, the Customer should ultimately evaluate and test the device installed in the Customer's product device.

●TUDI Semiconductor documentation is only allowed to be copied without any alteration of the content and with the relevant authorization.TUDI Semiconductor assumes no responsibility or liability for altered documents.

●TUDI Semiconductor is committed to becoming the preferred semiconductor brand for customers, and TUDI Semiconductor will strive to provide customers with better performance and better quallity products.