

# EPCQ-L Serial Configuration Devices Datasheet

2017.05.22

CF52013



Subscribe



Send Feedback

## Supported Devices

Table 1: EPCQ-L Devices

Device	Memory Size (bits)	On-Chip Decompression Support <sup>(1)</sup>	ISP Support	Cascading Support <sup>(2)</sup>	Reprogrammable	Recommended Operating Voltage (V)	Number of Die (256MB)
EPCQ-L256	268,435,456	No	Yes	No	Yes	1.8	1
EPCQ-L512	536,870,912	No	Yes	No	Yes	1.8	2
EPCQ-L1024	1,073,741,824	No	Yes	No	Yes	1.8	4

## Features

EPCQ-L devices offer the following features:

- Compatibility with Stratix 10 FPGAs
- Compatibility with Arria 10 FPGAs and SOCs
- Native support for active serial (AS) x4
- Backward compatibility for AS x1 on Arria 10 devices
- Low pin count and non-volatile memory
- 1.8-V operation
- Stacked die device
- Manufactured on NOR technology
- Available in FBGA24 package
- Reprogrammable memory with more than 100,000 erase or program cycles
- Write protection support for memory sectors using status register bits
- Fast read and extended quad input fast read of the entire memory using a single operation code
- Write bytes and extended quad input fast write bytes of the entire memory using a single operation code
- In-system programming (ISP) support with the SRunner software driver

<sup>(1)</sup> EPCQ-L devices are compatible with decompression built into Arria 10 devices.

<sup>(2)</sup> Multiple EPCQ-L devices may be used on a single FPGA device.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

\*Other names and brands may be claimed as the property of others.

ISO  
9001:2008  
Registered

**ALTERA**  
now part of Intel

- ISP support with Intel® FPGA Download Cable II, Intel FPGA Download Cable, Ethernet Download Cable II, or Ethernet Download Cable download cables
- By default, the memory array is erased and the bits are set to 1
- During user mode, you can use the ALTASMI\_PARALLEL IP core to access the EPCQ-L device

**Warning:** EPCQ-L devices are only compatible with Stratix 10 and Arria 10 devices.

## Operating Conditions

This section covers information about the absolute maximum ratings, recommended operating conditions, DC operating conditions, ICC supply current, and capacitance for EPCQ-L devices.

### Absolute Maximum Ratings

**Table 2: Absolute Maximum Ratings**

Symbol	Parameter	Condition	Min	Max	Unit
$V_{CC}$	Supply voltage	With respect to GND	-0.6	2.4	V
$V_{IO}^{(3), (4)}$	DC input/output voltage	With respect to GND	-0.6	$V_{CC} + 0.6$	V
$T_{STG}$	Storage temperature	No bias	-65	150	°C

### Recommended Operating Conditions

**Table 3: Recommended Operating Conditions**

Symbol	Parameter	Condition	Min	Max	Unit
$V_{CC}$	Supply voltage	<sup>(5)</sup>	1.7	2.0	V
$V_I$	Input voltage	With respect to GND	-0.5	$0.4 + V_{CC}$	V
$T_A^{(6)}$	Operating temperature	For industrial use	-40	85	°C
$t_R$	Input rise time	—	—	5	ns
$t_F$	Input fall time	—	—	5	ns

<sup>(3)</sup> During signal transitions, minimum voltage may undershoot to -1V for periods less than 10 ns.

<sup>(4)</sup> During signal transitions, maximum voltage may overshoot to  $V_{CC} + 1V$  for periods less than 10 ns.

<sup>(5)</sup> The maximum  $V_{CC}$  rise time is 100 ms.

<sup>(6)</sup> EPCQ-L devices can be paired with industrial-grade FPGAs operating at junction temperatures up to 100°C as long as the ambient temperature for the EPCQ-L device does not exceed 85°C.

**Related Information****EPCQ-L Package and Thermal Resistance**

Provides more information about EPCQ-L thermal resistance.

**DC Operating Conditions****Table 4: DC Operating Conditions**

Symbol	Parameter	Condition	Min	Max	Unit
$V_{IH}$	High-level input voltage	—	$0.7 \times V_{CC}$	$V_{CC} + 0.4$	V
$V_{IL}$	Low-level input voltage	—	-0.5	$0.3 \times V_{CC}$	V
$V_{OH}$	High-level output voltage	$I_{OH} = -100 \mu\text{A}$ <sup>(7)</sup>	$V_{CC} - 0.2$	—	V
$V_{OL}$	Low-level output voltage	$I_{OL} = 1.6 \text{ mA}$ <sup>(8)</sup>	—	0.4	V
$I_I$	Input leakage current	$V_I = V_{CC}$ or GND	-2	2	$\mu\text{A}$

**ICC Supply Current****Table 5:  $I_{CC}$  Supply Current**

Symbol	Parameter	Condition	Min	Max	Unit
$I_{CC0}$	$V_{CC}$ supply current	Standby	—	100	$\mu\text{A}$
$I_{CC1}$	$V_{CC}$ supply current	During active power mode	—	20	mA

**Capacitance****Table 6: Capacitance**

Symbol	Parameter <sup>(9)</sup>	Condition	Min	Max	Unit
$C_{IN}$	Input pin capacitance	$V_{IN} = 0 \text{ V}$	—	6	pF
$C_{IN/OUT}$	Input/Output pin capacitance	$V_{OUT} = 0 \text{ V}$	—	8	pF

<sup>(7)</sup> The  $I_{OH}$  parameter refers to the high-level TTL or CMOS output current.

<sup>(8)</sup> The  $I_{OL}$  parameter refers to the low-level TTL or CMOS output current.

<sup>(9)</sup> Capacitance is sample-tested only at  $T_A = 25^\circ\text{C}$  and at a 54-MHz frequency.

## Pin Information

Figure 1: Pin-Out Diagram for EPCQ-L Devices in the FBGA24 Package

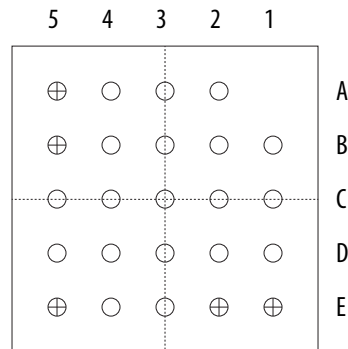


Table 7: Signals for EPCQ-L Devices

Signals	Balls
nCS	C2
DCLK	B2
DATA0	D3
DATA1	D2
DATA2	C4
DATA3	D4
VCC	B4
GND	B3

Signals	Balls
DNU <sup>(10)</sup>	A2
	A3
	A4
	A5
	B1
	B5
	C1
	C3
	C5
	D1
	D5
	E1
	E2
	E3
	E4
	E5

## EPCQ-L Device Pin Description

Table 8: EPCQ-L Device Pin Description

Pin Name	Pin Type	Description
nCS	Input	<p>The active low nCS input signal toggles at the beginning and end of a valid operation. When this signal is high, the device is deselected and the DATA pin is tri-stated.</p> <p>When this signal is low, the device is enabled and is in active mode. After power up, the EPCQ-L device requires a falling edge on the nCS signal before you begin any operation.</p>
DCLK	Input	<p>The FPGA provides the DCLK signal. This signal provides the timing for the serial interface. The data presented on the DATA0 pin is latched to the EPCQ-L 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 in to the FPGA on the next falling edge of the DCLK signal.</p>

<sup>(10)</sup> Do not use

Pin Name	Pin Type	Description
DATA0	I/O	<p>For AS x1 mode, use this pin as an input signal pin to write or program the EPCQ-L device. During write or program operations, the data is latched on the rising edge of the <code>DCLK</code> signal.</p> <p>For AS x4 mode, use this pin as an I/O signal pin. During write or program operations, this pin acts as an input pin that serially transfers data into the EPCQ-L device. The data is latched on the rising edge of the <code>DCLK</code> signal. During read or configuration operations, this pin acts as an output signal pin that serially transfers data out of the EPCQ-L device to the FPGA. The data is shifted out on the falling edge of the <code>DCLK</code> signal.</p> <p>During the extended quad input fast write bytes operation, this pin acts as an input pin that serially transfers data into the EPCQ-L device. The data is latched on the rising edge of the <code>DCLK</code> signal. During extended quad input fast read operation, this pin acts as an output signal pin that serially transfers data out of the EPCQ-L device to the FPGA. The data is shifted out on the falling edge of the <code>DCLK</code> signal.</p>
DATA1	I/O	<p>For AS x1 mode, use this pin as an output signal pin that serially transfers data out of the EPCQ-L device to the FPGA during read or configuration operations. For AS x4 mode, use this pin as an I/O signal pin. The transition of the signal is on the falling edge of the <code>DCLK</code> signal.</p> <p>During the extended quad input fast write bytes operation, this pin acts as an input signal pin that serially transfers data into the EPCQ-L device. The data is latched on the rising edge of the <code>DCLK</code> signal.</p> <p>During extended quad input fast read operation, this pin acts as an output signal pin that serially transfer data out of the EPCQ-L device to the FPGA. The data is shifted out on the falling edge of the <code>DCLK</code> signal. During read, configuration, or program operations, you can enable the EPCQ-L device by pulling the <code>nCS</code> signal low.</p>
DATA2	I/O	<p>For AS x1 mode, this pin must connect to a 1.8-V power supply.</p> <p>For AS x4 mode, use this pin as an output signal that serially transfers data out of the EPCQ-L device to the FPGA during read or configuration operations. The transition of the signal is on the falling edge of the <code>DCLK</code> signal.</p> <p>During the extended quad input fast write bytes operation, this pin acts as an input pin that serially transfers data into the EPCQ-L device. The data is latched on the rising edge of the <code>DCLK</code> signal. During the extended quad input fast read operation, this pin acts as an output signal pin that serially transfers data out of the EPCQ-L device to the FPGA. The data is shifted out on the falling edge of the <code>DCLK</code> signal.</p>

Pin Name	Pin Type	Description
DATA3	I/O	<p>For AS x1 mode, this pin must connect to a 1.8-V power supply.</p> <p>For AS x4 mode, use this pin as an output signal that serially transfers data out of the EPCQ-L device to the FPGA during read or configuration operations. The transition of the signal is on the falling edge of the <code>DCLK</code> signal.</p> <p>During the extended quad input fast write bytes operation, this pin acts as an input pin that serially transfers data into the EPCQ-L device. The data is latched on the rising edge of the <code>DCLK</code> signal. During the extended quad input fast read operation, this pin acts as an output signal pin that serially transfers data out of the EPCQ-L device to the FPGA. The data is shifted out on the falling edge of the <code>DCLK</code> signal.</p>
VCC	Power	Connect the power pins to a 1.8-V power supply.
GND	Ground	Ground pin.

## Device Package and Ordering Code

This section describes the package offered in EPCQ-L devices and the ordering codes for each EPCQ-L device.

### Package

The EPCQ-L256, EPCQ-L512, and EPCQ-L1024 devices are available in FBGA24 packages.

#### Related Information

#### [EPCQ-L Device Package Information](#)

Provides more information about EPCQ-L packaging specifications, thermal resistance and dimensions.

### Ordering Code

Table 9: EPCQ-L Device Ordering Codes

Device	Ordering Code <sup>(11)</sup>
EPCQ-L256	EPCQL256F24IN
EPCQ-L512	EPCQL512F24IN
EPCQ-L1024	EPCQL1024F24IN

<sup>(11)</sup> N indicates that the device is lead free.

## Memory Array Organization

**Table 10: Memory Array Organization in EPCQ-L Devices**

Details	EPCQ-L256	EPCQ-L512	EPCQ-L1024
Bytes	33,554,432 bytes (256 Mb)	67,108,864 bytes (512 Mb)	134,217,728 bytes (1,024 Mb)
Number of sectors	512	1,024	2,048
Bytes per sector	65,536 bytes (512 Kb)		
Total numbers of subsectors <sup>(12)</sup>	8,192	16,384	32,768
Bytes per subsector	4,096 bytes (32 Kb)		
Pages per sector	256		
Total number of pages	131,072	262,144	524,288
Bytes per page	256 bytes		

### Address Range for EPCQ-L256

**Table 11: Address Range for Sectors 511..0 and Subsectors 8191..0 in EPCQ-L256 Devices**

Sector	Subsector	Address Range (Byte Addresses in HEX)	
		Start	End
511	8191	01FFF000h	01FFFFFFh
	...	...	...
	8176	01FF0000h	01FF0FFFh
...	...	...	...
255	4095	00FFF000h	00FFFFFFh
	...	...	...
	4080	00FF0000h	00FF0FFFh
...	...	...	...
127	2047	007FF000h	007FFFFFFh
	...	...	...
	2032	007F0000h	007F0FFFh
...	...	...	...

<sup>(12)</sup> Every sector is further divided into 16 subsectors with 4 KB of memory. Therefore, there are 8,192 (512 x 16) subsectors for the EPCQ-L256 device, 16,384 (1,024 x 16) subsectors for the EPCQ-L512 device, and 32,768 (2,048 x 16) subsectors for the EPCQ-L1024 device.



Sector	Subsector	Address Range (Byte Addresses in HEX)	
		Start	End
63	1023	003FF000h	003FFFFFFh
	...	...	...
	1008	003F0000h	003F0FFFh
...	...	...	...
0	15	0000F000h	0000FFFFh
	...	...	...
	0	00000000h	00000FFFh

## Address Range for EPCQ-L512

Table 12: Address Range for Sectors 1023..0 and Subsectors 16383..0 in EPCQ-L256 Devices

Sector	Subsector	Address Range (Byte Addresses in HEX)	
		Start	End
1023	16383	03FFF000h	03FFFFFFh
	...	...	...
	16368	03FF0000h	03FF0FFFh
...	...	...	...
511	8191	01FFF000h	01FFFFFFh
	...	...	...
	8176	01FF0000h	01FF0FFFh
...	...	...	...
255	4095	00FFF000h	00FFFFFFh
	...	...	...
	4080	00FF0000h	00FF0FFFh
...	...	...	...
127	2047	007FF000h	007FFFFFFh
	...	...	...
	2032	007F0000h	007F0FFFh
...	...	...	...
63	1023	003FF000h	003FFFFFFh
	...	...	...
	1008	003F0000h	003F0FFFh
...	...	...	...

Sector	Subsector	Address Range (Byte Addresses in HEX)	
		Start	End
0	15	0000F000h	0000FFFFh
	...	...	...
	0	00000000h	0000FFFFh

## Address Range for EPCQ-L1024

Table 13: Address Range for Sectors 2047..0 and Subsectors 32767..0 in EPCQ-L1024 Devices

Sector	Subsector	Address Range (Byte Addresses in HEX)	
		Start	End
2047	32767	07FFF000h	07FFFFFFh
	...	...	...
	32750	07FF0000h	07FF0FFFh
...	...	...	...
1023	16383	03FFF000h	03FFFFFFh
	...	...	...
	16368	03FF0000h	03FF0FFFh
...	...	...	...
511	8191	01FFF000h	01FFFFFFh
	...	...	...
	8176	01FF0000h	01FF0FFFh
...	...	...	...
255	4095	00FFF000h	00FFFFFFh
	...	...	...
	4080	00FF0000h	00FF0FFFh
...	...	...	...
127	2047	007FF000h	007FFFFFFh
	...	...	...
	2032	007F0000h	007F0FFFh
...	...	...	...
63	1023	003FF000h	003FFFFFFh
	...	...	...
	1008	003F0000h	003F0FFFh
...	...	...	...

Sector	Subsector	Address Range (Byte Addresses in HEX)	
		Start	End
0	15	0000F000h	0000FFFFh
	...	...	...
	0	00000000h	0000FFFFh

## Memory Operations

This section describes the operations that you can use to access the memory in EPCQ-L devices. When performing the operation, addresses and data are shifted in and out of the device serially, with the MSB first.

### Timing Requirements

When the active low chip select ( $nCS$ ) signal is driven low, shift in the operation code into the EPCQ-L device using the serial data ( $DATA0$ ) pin. Each operation code bit is latched into the EPCQ-L device on the rising edge of the  $DCLK$ .

While executing an operation, shift-in the desired operation code, followed by the address or data bytes as listed in [Table 24](#). The device must drive the  $nCS$  pin high after the last bit of the operation sequence is shifted in.

For read operations, the data read is shifted out on the  $DATA0$  pin. You can drive the  $nCS$  pin high when any bit of the data 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 continues unaffected.

### Addressing Mode

To access the EPCQ-L256, EPCQ-L512, or EPCQ-L1024 memory, you must use the 4-byte addressing mode. In 4-byte addressing mode, the address width is 32-bit. To enable the 4-byte addressing mode, you must execute the 4BYTEADDREN operation. This addressing mode takes effect immediately after you execute the 4BYTEADDREN operation and remains active in the subsequent power-ups. To disable the 4-byte addressing mode, you must execute the 4BYTEADDREX operation.

**Note:** If you are using the Quartus® Prime II software or the SRunner software to program the EPCQ-L256, EPCQ-L512, or EPCQ-L1024 device, you do not need to execute the 4BYTEADDREN operation. These software tools automatically enable the 4-byte addressing mode when programming the device.

# Registers

## Status Register

**Table 14: Status Register Bits**

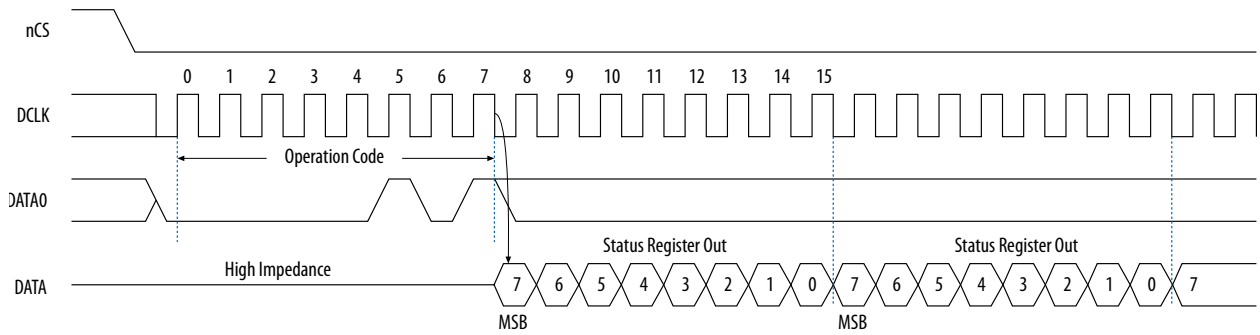
Bit	Name	Value	Description
7	None		
6	BP3 (Block Protect Bit) <sup>(13)</sup>	<b>Table 15</b> through <b>Table 20</b> list the protected area with reference to the block protect bits.	Determine the area of the memory protected from being written or erased unintentionally.
5	TB (Top/Bottom Bit)	<ul style="list-style-type: none"> <li>1=Protected area starts from the bottom of the memory array.</li> <li>0=Protected area starts from the top of the memory array.</li> </ul>	Determine that the protected area starts from the top or bottom of the memory array.
4	BP2 <sup>(13)</sup>	<b>Table 15</b> through <b>Table 20</b> list the protected area with reference to the block protect bits.	Determine the area of the memory protected from being written or erased unintentionally.
3	BP1 <sup>(13)</sup>		
2	BP0 <sup>(13)</sup>		
1	WEL (Write Enable Latch Bit)	<ul style="list-style-type: none"> <li>1=Allows the following operation to run:               <ul style="list-style-type: none"> <li>Write Bytes</li> <li>Write Status Register</li> <li>Erase Bulk</li> <li>Erase Die</li> <li>Erase Sector</li> </ul> </li> <li>0=Rejects the above mentioned operations.</li> </ul>	Allows or rejects certain operation to run.
0	WIP (Write in Progress Bit)	<ul style="list-style-type: none"> <li>1=One of the following operation is in progress:               <ul style="list-style-type: none"> <li>Write Status Register</li> <li>Write NVCR</li> <li>Write Bytes</li> <li>Erase</li> </ul> </li> <li>0=no write or erase cycle in progress</li> </ul>	Indicates if there is a command in progress.

### Read Status Register Operation

The status register can be read continuously and at anytime, including during a write or erase operations.

<sup>(13)</sup> The erase bulk and erase die 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.

Figure 2: Read Status Register Operation Timing Diagram



Block Protection Bits in EPCQ-L256 when TB Bit is Set to 0

Table 15: Block Protection Bits in EPCQ-L256 when TB Bit is Set to 0

Status Register Content					Memory Content	
TB Bit	BP3 Bit	BP2 Bit	BP1 Bit	BP0 Bit	Protected Area	Unprotected Area
0	0	0	0	0	None	All sectors
0	0	0	0	1	Sector 511	Sectors (0 to 510)
0	0	0	1	0	Sectors (510 to 511)	Sectors (0 to 509)
0	0	0	1	1	Sectors (508 to 511)	Sectors (0 to 507)
0	0	1	0	0	Sectors (504 to 511)	Sectors (0 to 503)
0	0	1	0	1	Sectors (496 to 511)	Sectors (0 to 495)
0	0	1	1	0	Sectors (480 to 511)	Sectors (0 to 479)
0	0	1	1	1	Sectors (448 to 511)	Sectors (0 to 447)
0	1	0	0	0	Sectors (384 to 511)	Sectors (0 to 383)
0	1	0	0	1	Sectors (256 to 511)	Sectors (0 to 255)
0	1	0	1	0	All sectors	None
0	1	0	1	1	All sectors	None
0	1	1	0	0	All sectors	None
0	1	1	0	1	All sectors	None
0	1	1	1	0	All sectors	None
0	1	1	1	1	All sectors	None

## Block Protection Bits in EPCQ-L256 when TB Bit is Set to 1

Table 16: Block Protection Bits in EPCQ-L256 when TB Bit is Set to 1

Status Register Content					Memory Content	
TB Bit	BP3 Bit	BP2 Bit	BP1 Bit	BP0 Bit	Protected Area	Unprotected Area
1	0	0	0	0	None	All sectors
1	0	0	0	1	Sector 0	Sectors (1 to 511)
1	0	0	1	0	Sectors (0 to 1)	Sectors (2 to 511)
1	0	0	1	1	Sectors (0 to 3)	Sectors (4 to 511)
1	0	1	0	0	Sectors (0 to 7)	Sectors (8 to 511)
1	0	1	0	1	Sectors (0 to 15)	Sectors (16 to 511)
1	0	1	1	0	Sectors (0 to 31)	Sectors (32 to 511)
1	0	1	1	1	Sectors (0 to 63)	Sectors (64 to 511)
1	1	0	0	0	Sectors (0 to 127)	Sectors (128 to 511)
1	1	0	0	1	Sectors (0 to 255)	Sectors (256 to 511)
1	1	0	1	0	All sectors	None
1	1	0	1	1	All sectors	None
1	1	1	0	0	All sectors	None
1	1	1	0	1	All sectors	None
1	1	1	1	0	All sectors	None
1	1	1	1	1	All sectors	None

## Block Protection Bits in EPCQ-L512 when TB Bit is Set to 0

Table 17: Block Protection Bits in EPCQ-L512 when TB Bit is Set to 0

Status Register Content					Memory Content	
TB Bit	BP3 Bit	BP2 Bit	BP1 Bit	BP0 Bit	Protected Area	Unprotected Area
0	0	0	0	0	None	All sectors
0	0	0	0	1	Sector 1023	Sectors (0 to 1022)
0	0	0	1	0	Sectors (1022 to 1023)	Sectors (0 to 1021)
0	0	0	1	1	Sectors (1020 to 1023)	Sectors (0 to 1019)
0	0	1	0	0	Sectors (1016 to 1023)	Sectors (0 to 1015)
0	0	1	0	1	Sectors (1008 to 1023)	Sectors (0 to 1007)
0	0	1	1	0	Sectors (992 to 1023)	Sectors (0 to 991)
0	0	1	1	1	Sectors (960 to 1023)	Sectors (0 to 959)

Status Register Content					Memory Content	
TB Bit	BP3 Bit	BP2 Bit	BP1 Bit	BP0 Bit	Protected Area	Unprotected Area
0	1	0	0	0	Sectors (896 to 1023)	Sectors (0 to 895)
0	1	0	0	1	Sectors (768 to 1023)	Sectors (0 to 767)
0	1	0	1	0	Sectors (512 to 1023)	Sectors (0 to 511)
0	1	0	1	1	All sectors	None
0	1	1	0	0	All sectors	None
0	1	1	0	1	All sectors	None
0	1	1	1	0	All sectors	None
0	1	1	1	1	All sectors	None

### Block Protection Bits in EPCQ-L512 when TB Bit is Set to 1

Table 18: Block Protection Bits in EPCQ-L512 when TB Bit is Set to 1

Status Register Content					Memory Content	
TB Bit	BP3 Bit	BP2 Bit	BP1 Bit	BP0 Bit	Protected Area	Unprotected Area
1	0	0	0	0	None	All sectors
1	0	0	0	1	Sector 0	Sectors (1 to 1023)
1	0	0	1	0	Sectors (0 to 1)	Sectors (2 to 1023)
1	0	0	1	1	Sectors (0 to 3)	Sectors (4 to 1023)
1	0	1	0	0	Sectors (0 to 7)	Sectors (8 to 1023)
1	0	1	0	1	Sectors (0 to 15)	Sectors (16 to 1023)
1	0	1	1	0	Sectors (0 to 31)	Sectors (32 to 1023)
1	0	1	1	1	Sectors (0 to 63)	Sectors (64 to 1023)
1	1	0	0	0	Sectors (0 to 127)	Sectors (128 to 1023)
1	1	0	0	1	Sectors (0 to 255)	Sectors (256 to 1023)
1	1	0	1	0	Sectors (0 to 511)	Sectors (512 to 1023)
1	1	0	1	1	All sectors	None
1	1	1	0	0	All sectors	None
1	1	1	0	1	All sectors	None
1	1	1	1	0	All sectors	None
1	1	1	1	1	All sectors	None

## Block Protection Bits in EPCQ-L1024 when TB Bit is Set to 0

Table 19: Block Protection Bits in EPCQ-L1024 when TB Bit is Set to 0

Status Register Content					Memory Content	
TB Bit	BP3 Bit	BP2 Bit	BP1 Bit	BP0 Bit	Protected Area	Unprotected Area
0	0	0	0	0	None	All sectors
0	0	0	0	1	Sector 2047	Sectors (0 to 2046)
0	0	0	1	0	Sectors (2046 to 2047)	Sectors (0 to 2045)
0	0	0	1	1	Sectors (2044 to 2047)	Sectors (0 to 2043)
0	0	1	0	0	Sectors (2040 to 2047)	Sectors (0 to 2039)
0	0	1	0	1	Sectors (2032 to 2047)	Sectors (0 to 2031)
0	0	1	1	0	Sectors (2016 to 2047)	Sectors (0 to 2015)
0	0	1	1	1	Sectors (1984 to 2047)	Sectors (0 to 1983)
0	1	0	0	0	Sectors (1920 to 2047)	Sectors (0 to 1919)
0	1	0	0	1	Sectors (1792 to 2047)	Sectors (0 to 1791)
0	1	0	1	0	Sectors (1536 to 2047)	Sectors (0 to 1535)
0	1	0	1	1	Sectors (1024 to 2047)	Sectors (0 to 1023)
0	1	1	0	0	All sectors	None
0	1	1	0	1	All sectors	None
0	1	1	1	0	All sectors	None
0	1	1	1	1	All sectors	None

## Block Protection Bits in EPCQ-L1024 when TB Bit is Set to 1

Table 20: Block Protection Bits in EPCQ-L1024 when TB Bit is Set to 1

Status Register Content					Memory Content	
TB Bit	BP3 Bit	BP2 Bit	BP1 Bit	BP0 Bit	Protected Area	Unprotected Area
1	0	0	0	0	None	All sectors
1	0	0	0	1	Sector 0	Sectors (1 to 2047)
1	0	0	1	0	Sectors (0 to 1)	Sectors (2 to 2047)
1	0	0	1	1	Sectors (0 to 3)	Sectors (4 to 2047)
1	0	1	0	0	Sectors (0 to 7)	Sectors (8 to 2047)
1	0	1	0	1	Sectors (0 to 15)	Sectors (16 to 2047)
1	0	1	1	0	Sectors (0 to 31)	Sectors (32 to 2047)
1	0	1	1	1	Sectors (0 to 63)	Sectors (64 to 2047)



Status Register Content					Memory Content	
TB Bit	BP3 Bit	BP2 Bit	BP1 Bit	BP0 Bit	Protected Area	Unprotected Area
1	1	0	0	0	Sectors (0 to 127)	Sectors (128 to 2047)
1	1	0	0	1	Sectors (0 to 255)	Sectors (256 to 2047)
1	1	0	1	0	Sectors (0 to 511)	Sectors (512 to 2047)
1	1	0	1	1	Sectors (0 to 1023)	Sectors (1024 to 2047)
1	1	1	0	0	All sectors	None
1	1	1	0	1	All sectors	None
1	1	1	1	0	All sectors	None
1	1	1	1	1	All sectors	None

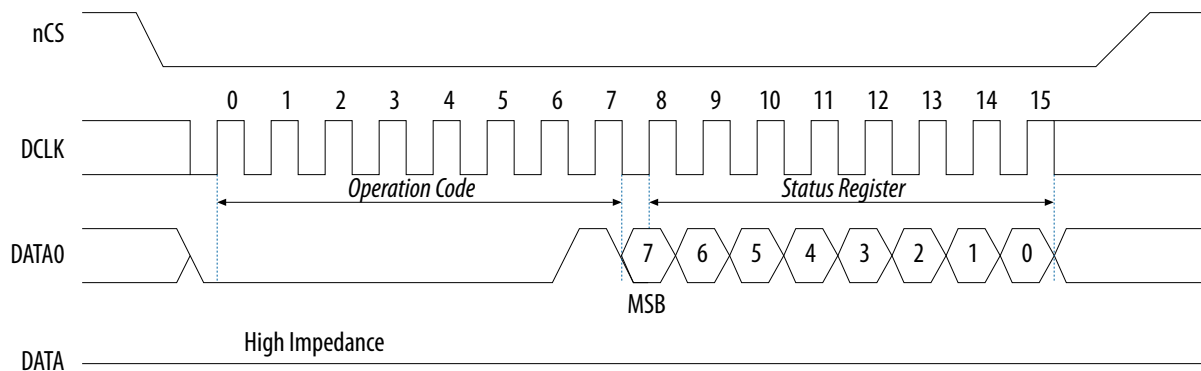
### Write Status Register Operation

The write status register operation does not affect the write enable latch and write in progress bits. You can use the write status register operation to set the status register block protection and top or bottom bits. Therefore, you can implement this operation to protect certain memory sectors. Refer to [Table 15](#) through [Table 20](#). 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.

When the operation is in progress, the write or erase controller bit of the flag status register is set to 0. To obtain the operation status, the flag status register must be polled<sup>(14)</sup>, with  $nCS$  toggled twice in between commands. When the operation completes, the write or erase controller bit is cleared to 1. The end of operation can be detected when the flag status register outputs the write or erase controller bit to 1 each time it is polled.

The following figure shows the timing diagram for the write status register operation.

**Figure 3: Write Status Register Operation Timing Diagram**



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 EPCQ-L devices and is guaranteed to be less than 8 ms.

<sup>(14)</sup> Poll the flag status register once for EPCQL256, twice for EPCQL512 or four times for EPQL1024.

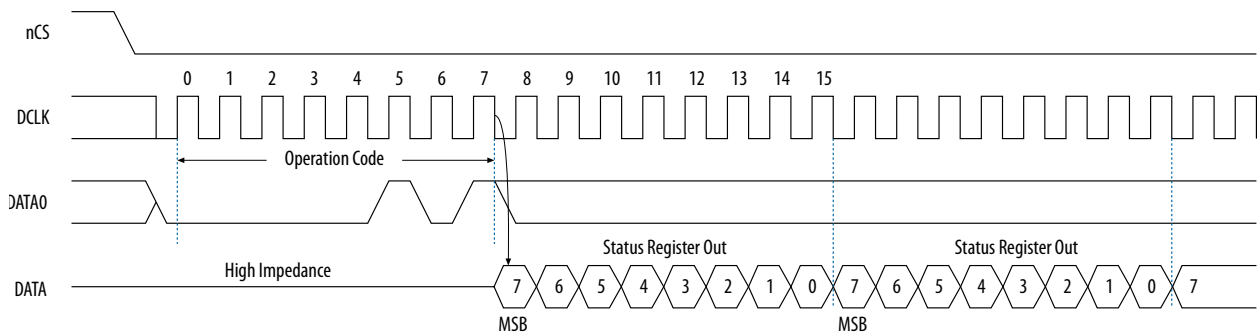
For details about  $t_{WS}$ , refer to [Table 26](#). You must account for this delay to ensure that the status register is written with the desired block protect bits. Alternatively, you can check the write in progress bit in the status register by executing the read status register operation while the self-timed write status cycle is in progress. The flash controller sets the write in progress bit to 1 during the self-timed write status cycle and 0 when it is complete.

## Flag Status Register

### Read Flag Status Register

The Read flag status register can be read continuously and at any time, including during a write or erase operation. You must read the Read flag status register every time a write or erase command is issued.

**Figure 4: Read Flag Status Register Operation Timing Diagram**



**Table 21: Flag Status Register Bits**

Bit	Name	Value	Description
7	Write or Erase Controller <sup>(15)</sup>	<ul style="list-style-type: none"> <li>1=Ready</li> <li>0=Busy</li> </ul>	Indicates whether one of the following operation is in progress: <ul style="list-style-type: none"> <li>Write Status Register</li> <li>Write NVCR</li> <li>Write Bytes</li> <li>Erase</li> </ul>
6	Erase suspend	<ul style="list-style-type: none"> <li>1=In effect</li> <li>0=Not in effect</li> </ul>	Indicates whether an Erase operation has been or is going to be suspended. <b>Note:</b> Status bits are reset automatically
5	Erase	<ul style="list-style-type: none"> <li>1=Failure or protection error</li> <li>0=Clear</li> </ul>	Indicates whether an Erase operation has succeeded or failed.

<sup>(15)</sup> Write or erase controller bit = NOT write in progress bit.

Bit	Name	Value	Description
4	Write	<ul style="list-style-type: none"> <li>1=Failure or protection error</li> <li>0=Clear</li> </ul>	Indicates whether a Write Bytes operation has succeeded or failed; also an attempt to write a 0 to a 1 when VPP = VPPH and the data pattern is a multiple of 64 bits.
3	VPP	<ul style="list-style-type: none"> <li>1=Disabled (Default)</li> <li>0=Enabled</li> </ul>	Indicates an invalid voltage on VPP during a Write Bytes or Erase operation.
2	Write suspend	<ul style="list-style-type: none"> <li>1=In effect</li> <li>0=Not in effect</li> </ul>	Indicates whether a Write Bytes operation has been or will be suspended.
1	Protection	<ul style="list-style-type: none"> <li>1=Failure or protection error</li> <li>0=Clear</li> </ul>	Indicates whether an Erase or Write Bytes operation has attempted to modify the protected array sector.
0	Addressing	<ul style="list-style-type: none"> <li>1=4-bytes addressing</li> <li>0=3-bytes addressing</li> </ul>	Indicates the addressing mode used.

## Non-Volatile Configuration Register

**Table 22: Dummy Clock Cycles and Address Bytes for the Non-Volatile Configuration Register Operation**

FPGA Device	Address Bytes	Dummy Clock Cycles	
		AS x1	AS x4
Arria 10	4-byte addressing		10

**Table 23: Non-Volatile Configuration Register Operation Bit Definition**

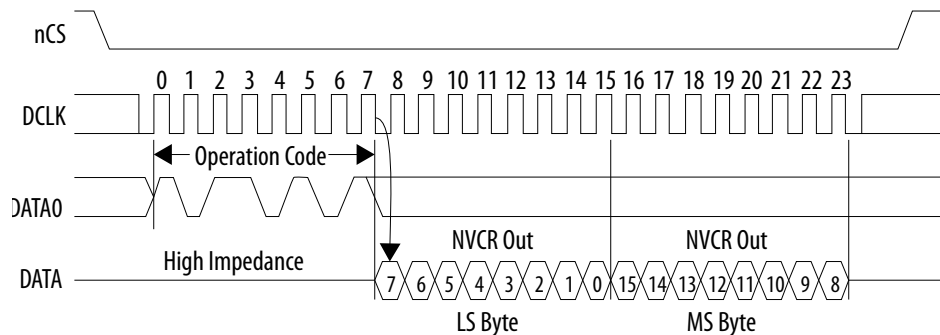
Bit	Description	Default Value
15:12	Number of dummy cycles. When this number is from 0001 to 1110, the dummy cycles is from 1 to 14.	0000 or 1111 <sup>(16)</sup>
11:5	Set these bits to 11111111.	11111111
4	Don't care.	1
3:1	Set these bits to 111.	111
0	Address byte setting. <ul style="list-style-type: none"> <li>0—4-byte addressing</li> <li>1—3-byte addressing</li> </ul>	1

<sup>(16)</sup> The default dummy clock cycles is 10 for extended quad input fast read and 8 for extended dual input fast and standard fast read.

## Read Non-Volatile Configuration Register Operation

To execute a read non-volatile configuration register, drive the  $nCS$  low. For extended SPI protocol, the operation code is input on  $DATA0$ , and output on  $DATA1$ . You can terminate the operation by driving the  $nCS$  low at any time during data output. The nonvolatile configuration register can be read continuously. After all 16 bits of the register have been read, a 0 is output.

**Figure 5: Read Non-Volatile Configuration Register Operation Timing Diagram**



## Write Non-Volatile Configuration Register Operation

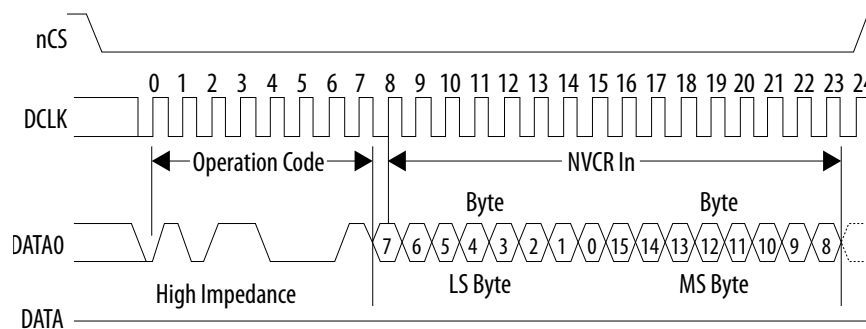
You need to write the non-volatile configuration registers for EPCQ-L devices for different configuration schemes. If you are using the `.jic` file, the Quartus Prime programmer sets the number of dummy clock cycles and address bytes. If you are using an external programmer tools (3rd party programmer tools), you must set the non-volatile configuration registers.

To set the non-volatile configuration register, follow these steps:

1. Execute the write enable operation.
2. Execute the write non-volatile configuration register operation.
3. Set the 16-bit register value.

Set the 16-bit register value as `b'1110 1110 xxxxx 1111` where `xxxxx` is the dummy clock value. When the `xxxxx` value is from 0001 to 1110, the dummy clock value is from 1 to 14. When `xxxxx` is 0000 or 1111, the dummy clock value is at the default value, which is 8 for standard fast read (AS x1) mode and 10 for extended quad input fast read (AS x4) mode.

**Figure 6: Write Non-Volatile Configuration Register Operation Timing Diagram**



## Summary of Operation Codes

**Table 24: Operation Codes for EPCQ-L Devices**

Operation	Operation Code <sup>(17)</sup>	Address Bytes	Dummy Cycles	Data Bytes	DCLK fMAX (MHz)
Read status register	b'0000 0101	0	0	1 to infinite <sup>(18)</sup>	100
Read flag status register	b'0111 0000	0	0	1 to infinite	100
Read bytes	b'0000 0011	4	0	1 to infinite <sup>(18)</sup>	50
Read non-volatile configuration register	b'1011 0101	0	0	2	100
Read device identification	b'1001 111x	0	2	1 to 20 <sup>(18)</sup>	100
Fast read (AS x1)	b'0000 1011	4	8 <sup>(19)</sup>	1 to infinite <sup>(18)</sup>	100
Extended quad input fast read (AS x4)	b'1110 1011	4	10 <sup>(19)</sup>	1 to infinite <sup>(18)</sup>	100
Dual I/O fast read	b'1011 1011	4	10	1 to infinite	100
Write enable	b'0000 0110	0	0	0	100
Write disable	b'0000 0100	0	0	0	100
Write status	b'0000 0001	0	0	1	100
Write bytes	b'0000 0010	4	0	1 to 256 <sup>(20)</sup>	100
Write non-volatile configuration register	b'1011 0001	0	0	2	100

<sup>(17)</sup> List MSB first and LSB last.

<sup>(18)</sup> The status register, data, or read device identification is read out at least once and is continuously read out until the nCS pin is driven high.

<sup>(19)</sup> You can configure the number of dummy cycles.

<sup>(20)</sup> A write bytes operation requires at least one data byte. If more than 256 bytes are sent to the device, only the last 256 bytes are written to the memory.

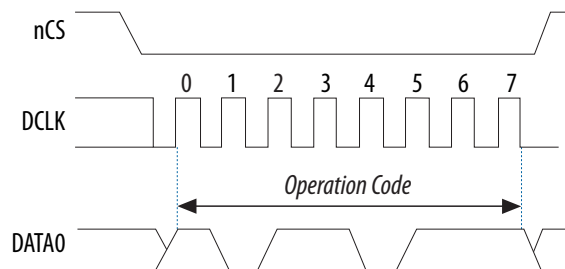
Operation	Operation Code <sup>(17)</sup>	Address Bytes	Dummy Cycles	Data Bytes	DCLK fMAX (MHz)
Extended quad input fast write bytes	b'0001 0010	4	0	1 to 256 <sup>(20)</sup>	100
Extended dual input fast write bytes	b'1101 0010	4	0	1 to 256	100
Erase bulk <sup>(21)</sup>	b'1100 0111	4	0	0	100
Erase die <sup>(22)</sup>	b'1100 0100	4	0	0	100
Erase sector	b'1101 1000	4	0	0	100
Erase subsector	b'0010 0000	4	0	0	100
4BYTEADDR EN	b'1011 0111	0	0	0	100
4BYTEADDR EX	b'1110 1001	0	0	0	100

## 4BYTEADDREN and 4BYTEADDREX Operations

To enable 4BYTEADDREN or 4BYTEADDREX operations, you can select the device by driving the `nCS` signal low, followed by shifting in the operation code through `DATA0`.

The following figure shows the timing diagram for the 4BYTEADDREN operation.

**Figure 7: 4BYTEADDREN Timing Diagram**

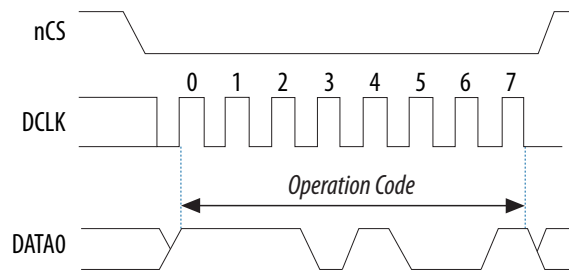


The following figure shows the timing diagram for the 4BYTEADDREX operation.

<sup>(17)</sup> List MSB first and LSB last.

<sup>(21)</sup> Erase bulk is applicable to EPCQ-L256 only.

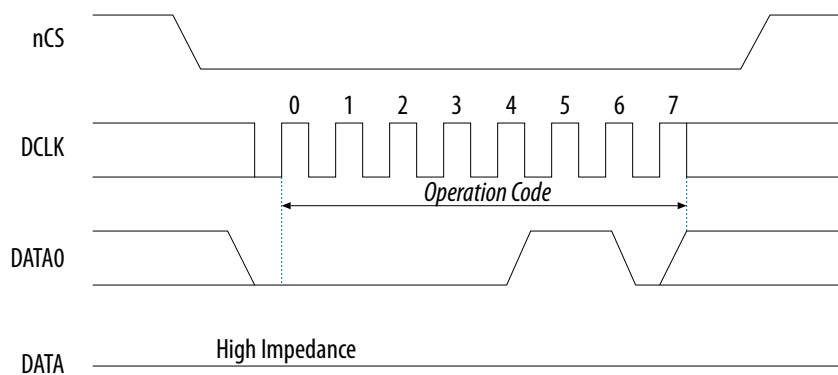
<sup>(22)</sup> Erase die is applicable to EPCQ-L512 and EPCQ-L1024 only.

**Figure 8: 4BYTEADDREX Timing Diagram**

## Write Enable Operation

When you enable the write enable operation, the write enable latch bit is set to 1 in the status register. You must execute this operation before the write bytes, write status, erase bulk, erase sector, erase die, extended quad input fast write bytes, 4BYTEADDREN, and 4BYTEADDREX operations.

The following figure shows the timing diagram for the write enable operation.

**Figure 9: Write Enable Operation Timing Diagram**

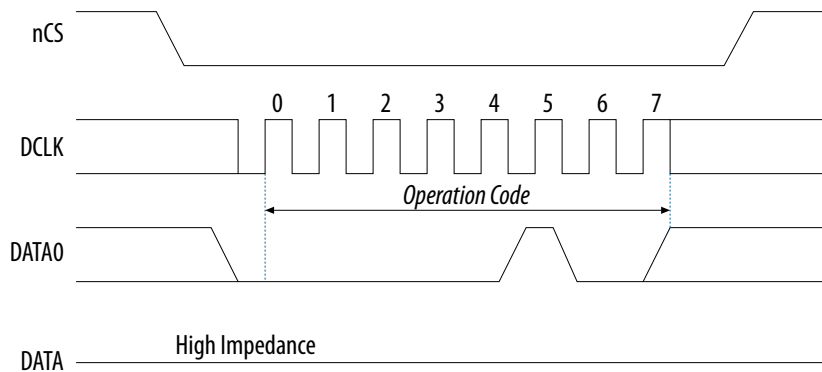
## Write Disable Operation

The write disable operation resets the write enable latch bit 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
- Erase die operation completion
- Extended quad input fast write bytes operation completion

The following figure shows the timing diagram for the write disable operation.

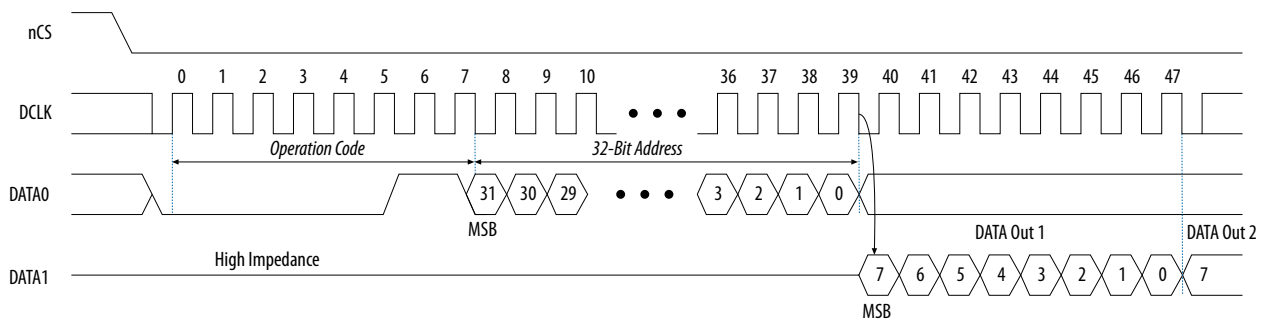
Figure 10: Write Disable Operation Timing Diagram



## Read Bytes Operation

When you execute the read bytes operation, you first shift in the read bytes operation code, followed by a 4-byte addressing mode ( $A[31..0]$ ). Each address bit is 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  $DATA1$  pin, beginning with the MSB. When reading back data programmed from a Raw Programming Data File (**.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 50 MHz.

Figure 11: Read Bytes Operation Timing Diagram



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. When the device reaches the highest address, the address counter restarts at the beginning of the same die, allowing the memory contents to be read out indefinitely until the read bytes operation is terminated by driving the  $nCS$  signal high. A complete device reading is done by executing the read operation:

- two times for EPCQ-L512 devices
- four times for EPCQ-L1024 devices

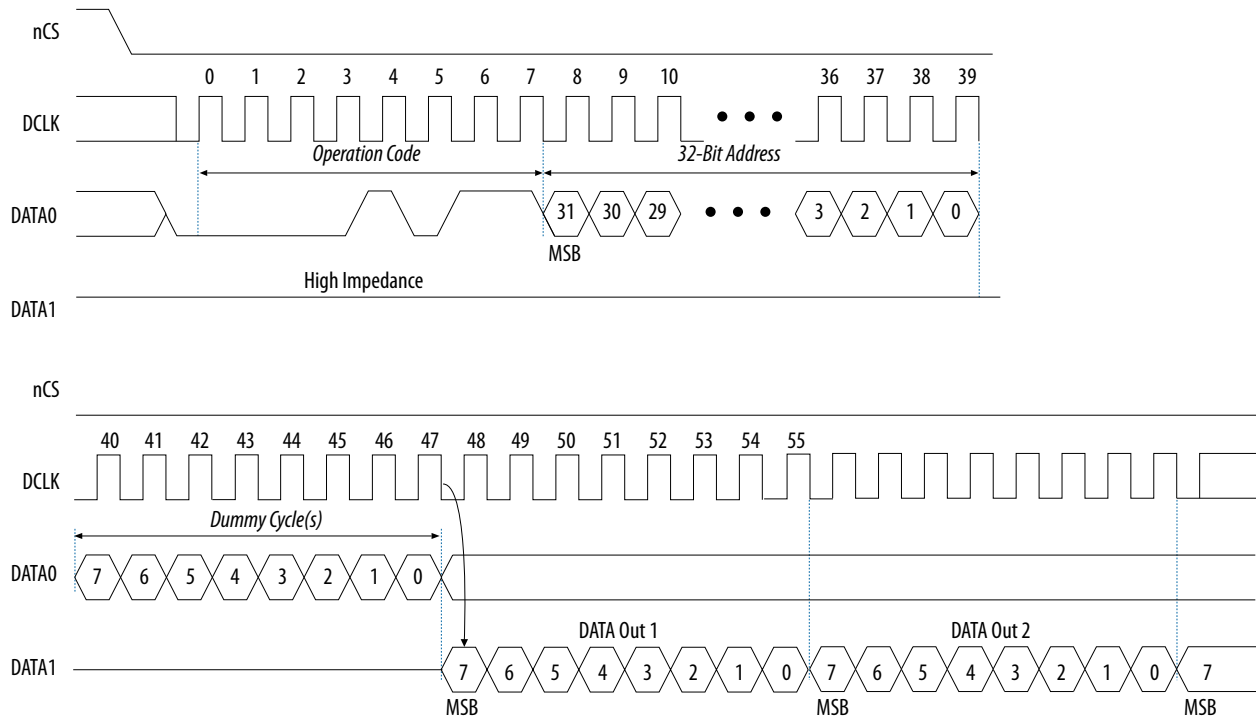
If the read bytes operation is shifted in while a write or erase cycle is in progress, the operation is not executed and does not affect the write or erase cycle in progress.



## Fast Read Operation

When you execute the fast read operation, you first shift in the fast read operation code, followed by a 4-byte addressing mode ( $A[31..0]$ ), and dummy cycle(s) 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  $DATA1$  with each bit being shifted out at a maximum frequency of 100 MHz during the falling edge of the  $DCLK$  signal.

Figure 12: Fast Read Operation Timing Diagram



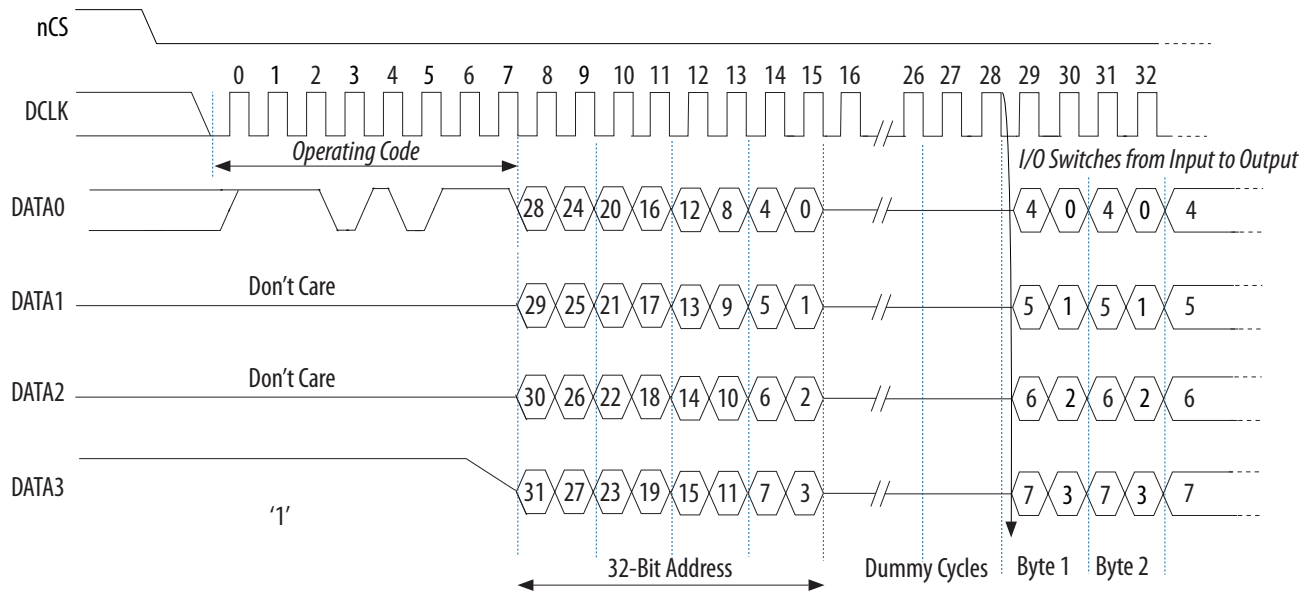
The first byte address can be at any location. The device automatically increases the address to the next higher address after shifting out each byte of data. When the device reaches the highest address, the address counter restarts at the beginning of the same die, allowing the read sequence to continue indefinitely. A complete device reading is done by executing the read operation:

- two times for EPCQ-L512 devices
- four times for EPCQ-L1024 devices

You can terminate the fast read operation by driving the  $nCS$  signal high at any time during data output. If the fast read operation is shifted in while an erase, program, or write cycle is in progress, the operation is not executed and does not affect the erase, program, or write cycle in progress.

## Extended Quad Input Fast Read Operation

This operation is similar to the fast read operation except that the data and addresses are shifted in and out on the  $DATA0$ ,  $DATA1$ ,  $DATA2$ , and  $DATA3$  pins.

**Figure 13: Extended Quad Input Fast Read Operation**

When the device reaches the highest address, the address counter restarts at the beginning of the same die, allowing the read sequence to continue indefinitely. A complete device reading is done by executing the read operation:

- two times for EPCQ-L512 devices
- four times for EPCQ-L1024 devices

## Read Device Identification Operation

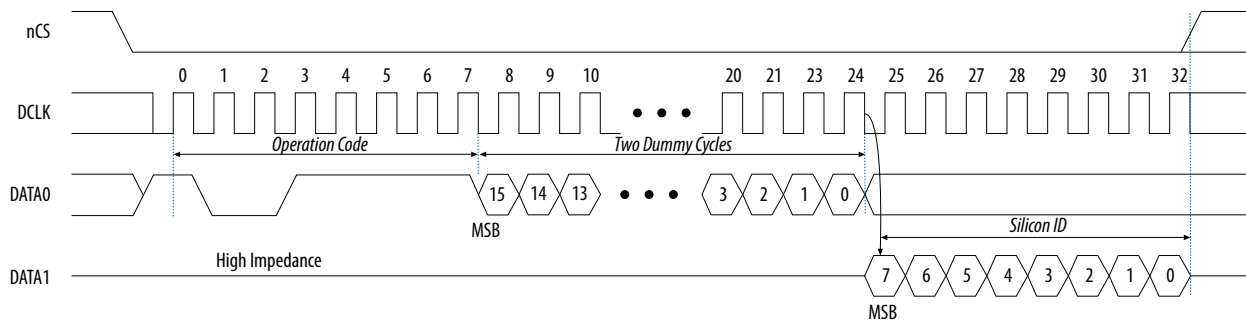
This operation reads the 8-bit device identification of the EPCQ-L device from the `DATA1` output pin. If this operation is shifted in while an erase or write cycle is in progress, the operation is not executed and does not affect the erase or write cycle in progress.

**Table 25: EPCQ-L Device Identification**

EPCQ-L Device	Silicon ID (Binary Value)
EPCQ-L256	b'0001 1001
EPCQ-L512	b'0010 0000
EPCQ-L1024	b'0010 0001

The 8-bit device identification of the EPCQ-L device is shifted out on the `DATA1` pin on the falling edge of the `DCLK` signal. LSB is first shifted into the FPGA device.

Figure 14: Read Device Identification Operation Timing Diagram



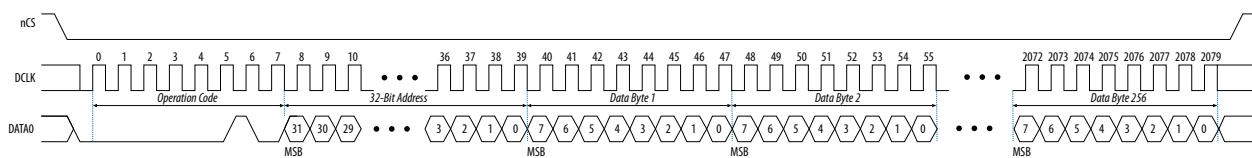
## Write Bytes Operation

This operation allows bytes to be written to the memory. You must execute the write enable operation before the write bytes operation. After the write bytes operation is completed, the write enable latch bit in the status register is set to 0.

When you execute the write bytes operation, you shift in the write bytes operation code, followed by a 4-byte addressing mode ( $A[31..0]$ ), and at least one data byte on the  $DATA0$  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. You must ensure the  $nCS$  signal is set low during the entire write bytes operation.

The following figure shows the operation sequence of the write bytes operation.

Figure 15: Write Bytes Operation Timing Diagram



If more than 256 data bytes are shifted into the EPCQ-L 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 EPCQ-L device, they are guaranteed to be written at the specified addresses and the other bytes of the same page are not affected.

The device initiates a self-timed write cycle immediately after the  $nCS$  signal is driven high. For details about the self-timed write cycle time, refer to  $t_{WB}$  in [Table 26](#). 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 and 0 when it is complete.

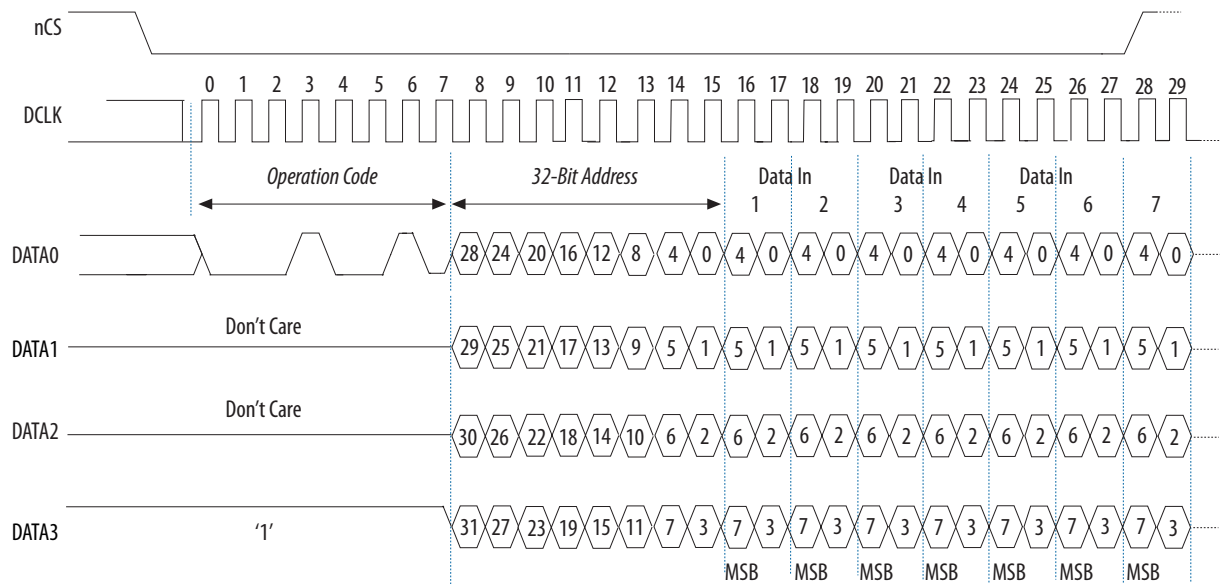
**Note:** You must erase all the memory bytes of EPCQ-L devices before you implement the write bytes operation. You can erase all the memory bytes by executing the erase sector operation in a sector or the erase bulk or erase die operation throughout the entire memory.

## Extended Quad Input Fast Write Bytes Operation

This operation is similar to the write bytes operation except that the data and addresses are shifted in on the DATA0, DATA1, DATA2, and DATA3 pins.

The following figure shows the operation sequence of the extended quad input fast write bytes operation.

**Figure 16: Extended Quad Input Fast Write Bytes Operation Sequence**



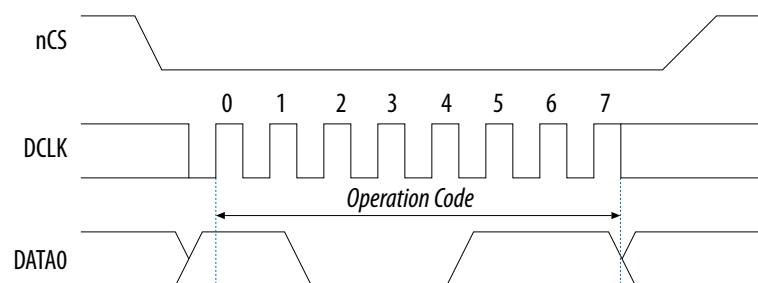
## Erase Bulk Operation

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.

If you are using the EPCQ-L256 device and wish to erase the whole memory of your device, you cannot use the erase die operation and instead must execute the erase bulk operation.

You can implement the erase bulk operation by driving the  $nCS$  signal low and then shifting in the erase bulk operation code on the  $DATA0$  pin. The  $nCS$  signal must be driven high after the eighth bit of the erase bulk operation code has been latched in.

**Figure 17: Erase Bulk Operation Timing Diagram**



The device initiates a self-timed erase bulk cycle immediately after the `nCS` signal is driven high. For details about the self-timed erase bulk cycle time, refer to  $t_{WB}$  in [Table 26](#).

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.

## Erase Die Operation

This operation sets all the memory bits of a particular die in an EPCQ-L512 or EPCQ-L1024 device to 1 or `0xFF`. Similar to the write bytes operation, you must execute the write enable operation before the erase die operation.

If you are using the EPCQ-L512 or EPCQ-L1024 device, you must execute the erase die operation to erase the memory of your device. You need to issue the erase die operation for each die in your device. For example, you need to issue the erase die operation twice for the EPCQ-L512 device and four times for the EPCQ-L1024 device. EPCQ-L512 and EPCQ-L1024 devices have more than one die per device.

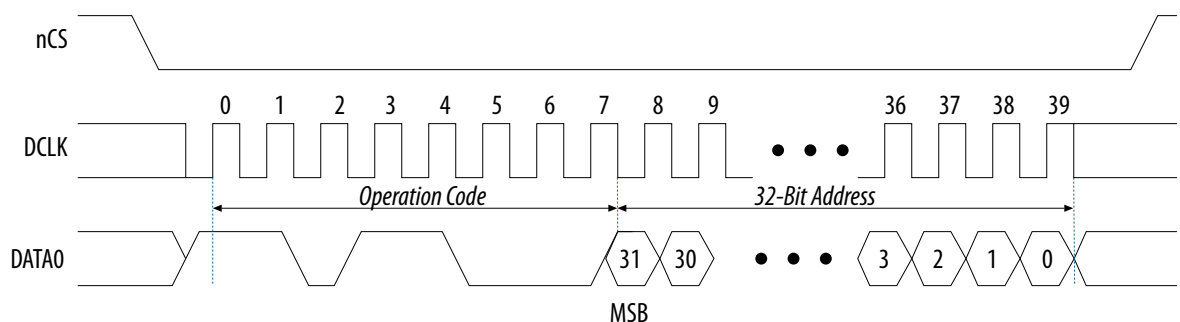
You can implement the erase die operation by driving the `nCS` signal low and then shifting in the erase die operation code on the `DATA0` pin, followed by the address bytes, any address within the single 256 Mb die is valid. The `nCS` signal must be driven high after the eighth bit of the erase die operation code has been latched in.

## Erase Sector Operation

The erase sector operation allows you to erase a certain sector in the EPCQ-L 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 a general purpose memory in your applications. You must execute the write enable operation before the erase sector operation.

When you execute the erase sector operation, you must first shift in the erase sector operation code, followed by the 4-byte addressing mode (`A[31..0]`) of the chosen sector on the `DATA0` pin. The 4-byte addressing mode for the erase sector operation can be any address inside the specified sector. For details about the sector address range, refer to [Table 11](#) through [Table 13](#). Drive the `nCS` signal high after the eighth bit of the erase sector operation code has been latched in.

Figure 18: Erase Sector Operation Timing Diagram



The device initiates a self-timed erase sector cycle immediately after the  $\overline{\text{nCS}}$  signal is driven high. For details about the self-timed erase sector cycle time, refer to  $t_{\text{ES}}$  in [Table 26](#). 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 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 set to 0 before the self-timed erase cycle is complete.

## Power Mode

EPCQ-L devices support active and standby power modes. When the  $\overline{\text{nCS}}$  signal is low, the device is enabled and is in active power mode. The FPGA is configured while the EPCQ-L device is in active power mode. When the  $\overline{\text{nCS}}$  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 EPCQ-L device then goes into standby power mode. The  $I_{\text{CC1}}$  and  $I_{\text{CC0}}$  parameters list the  $V_{\text{CC}}$  supply current when the device is in active and standby power modes. Refer to [Table 5](#).

## Timing Information

### Write Operation Timing

Figure 19: Write Operation Timing Diagram

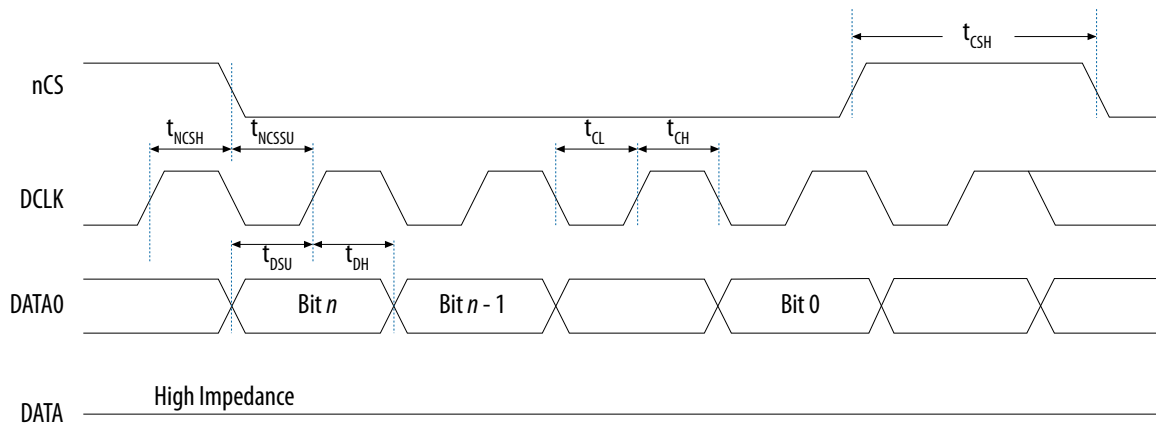


Table 26: Write Operation Timing Parameters

Symbol	Parameter	Min	Typical	Max	Unit
$f_{\text{WCLK}}$	Write clock frequency (from the FPGA, download cable, or embedded processor) for write enable, write disable, read status, read device identification, write bytes, erase bulk, erase die, and erase sector operations	—	—	100	MHz
$t_{\text{CH}}$	DCLK high time	4	—	—	ns

Symbol	Parameter	Min	Typical	Max	Unit
$t_{CL}$	DCLK low time	4	—	—	ns
$t_{NCSSU}$	Chip select ( $nCS$ ) setup time	4	—	—	ns
$t_{NCSH}$	Chip select ( $nCS$ ) hold time	4	—	—	ns
$t_{DSU}$	DATA[ ] in setup time before the rising edge on DCLK	2	—	—	ns
$t_{DH}$	DATA[ ] hold time after the rising edge on DCLK	3	—	—	ns
$t_{CSH}$	Chip select ( $nCS$ ) high time	50	—	—	ns
$t_{WB}$	Write bytes cycle time	—	0.6	5	ms
$t_{WS}$	Write status cycle time	—	1.3	8	ms
$t_{EB}$	Erase bulk cycle time for EPCQ-L256	—	240	480	s
	Erase die cycle time for EPCQ-L512				
	Erase die cycle time for EPCQ-L1024				
$t_{ES}$	Erase sector cycle time for EPCQ-L256	—	0.7	3	s
	Erase sector cycle time for EPCQ-L512				
	Erase sector cycle time for EPCQ-L1024				
$t_{ESS}$	Erase subsector cycle time for EPCQ-L256	—	0.25	0.8	s
	Erase subsector cycle time for EPCQ-L512				
	Erase subsector cycle time for EPCQ-L1024				

## Read Operation Timing

Figure 20: Read Operation Timing Diagram

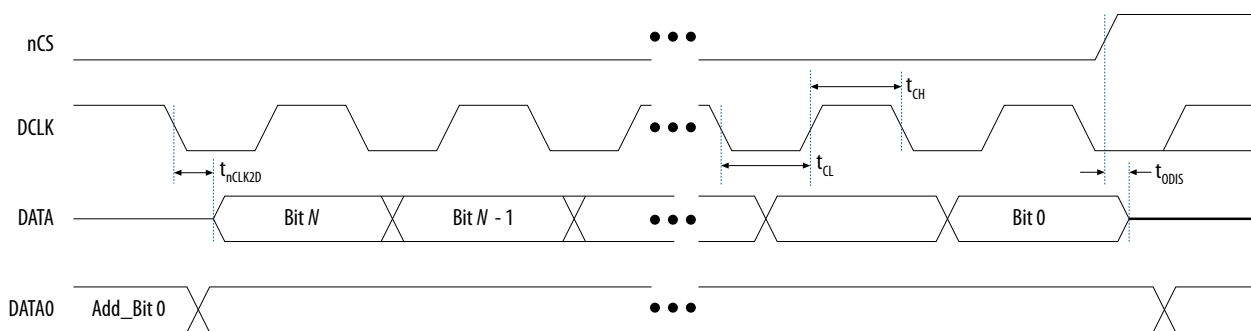


Table 27: Read Operation Timing Parameters

Symbol	Parameter	Min	Max	Unit
$f_{\text{RCLK}}$	Read clock frequency (from the FPGA or embedded processor) for read bytes operations	—	50	MHz
	Fast read clock frequency (from the FPGA or embedded processor) for fast read bytes operation	—	100	MHz
$t_{\text{CH}}$	DCLK high time	4	—	ns
$t_{\text{CL}}$	DCLK low time	4	—	ns
$t_{\text{ODIS}}$	Output disable time after read	—	8	ns
$t_{\text{nCLK2D}}$	Clock falling edge to DATA	—	7	ns

## Programming and Configuration File Support

The Quartus Prime software provides programming support for EPCQ-L devices. When you select an EPCQ-L device, the Quartus Prime software automatically generates the Programmer Object File (.pof) to program the device. The software allows you to select the appropriate EPCQ-L device density that most efficiently stores the configuration data for the selected FPGA.

You can program the EPCQ-L device in-system by an external microprocessor using the SRrunner software driver. The SRrunner software driver is developed for embedded EPCQ-L device programming that you can customize to fit in different embedded systems. The SRrunner software driver reads .rpd files and writes to the EPCQ-L devices. The programming time is comparable to the Quartus Prime 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 EPCQ-L device is different from the other data and address bytes.

During the ISP of an EPCQ-L device, the cable pulls the  $n_{\text{CONFIG}}$  signal low to reset the FPGA and overrides the 10-k $\Omega$  pull-down resistor on the  $n_{\text{CE}}$  pin of the FPGA. The download cable then uses the interface pins depending on the selected AS mode to program the EPCQ-L device. When programming is complete, the download cable releases the interface pins of the EPCQ-L device and the  $n_{\text{CE}}$  pin of the FPGA and pulses the  $n_{\text{CONFIG}}$  signal to start the configuration process.

The FPGA can program the EPCQ-L device in-system using the JTAG interface with the serial flash loader (SFL). This solution allows you to indirectly program the EPCQ-L device using the same JTAG interface that is used to configure the FPGA.

### Related Information

- [Using the Serial FlashLoader with the Quartus II Software](#)
- [Altera ASMI Parallel IP Core User Guide](#)
- [Intel FPGA USB Download Cable II User Guide](#)
- [Intel FPGA USB Download Cable User Guide](#)



- [Intel FPGA Ethernet Download Cable II User Guide](#)
- [Intel FPGA Ethernet Download Cable User Guide](#)
- [Configuration, Design Security, and Remote System Upgrades in Arria 10 Devices](#)

## Document Revision History

Date	Version	Changes
May 2017	2017.05.22	<ul style="list-style-type: none"> <li>• Added Read flag status register, Dual I/O fast read, and Extended dual input fast write bytes operations.</li> <li>• Updated instances of write status operation to write status register operation.</li> <li>• Added <i>Flag Status Register Bit Content</i> table.</li> <li>• Updated <i>Read Status Register Operations</i>.</li> <li>• Updated <i>Read Status Register</i>.</li> </ul>
December 2016	2016.12.16	<ul style="list-style-type: none"> <li>• Updated address bytes for erase bulk and erase die to 4.</li> <li>• Added erase subsector in <i>Operation Codes for EPCQ-L Devices</i> table.</li> <li>• Updated <math>t_{ESS}</math> Max to 0.8.</li> </ul>
October 2016	2016.10.31	<ul style="list-style-type: none"> <li>• Added Stratix 10 support.</li> <li>• Changed instances of Quartus II to Quartus Prime.</li> <li>• Changed instances of USB-Blaster to FPGA USB Download Cable.</li> <li>• Changed instances of EthernetBlaster to FPGA Ethernet Download Cable.</li> </ul>
May 2016	2016.05.30	Updated <i>Signals for EPCQ-L Devices</i> table by replacing NC with DNU.
March 2016	2016.03.31	Removed 'Preliminary' terms in Address Range for EPCQ-L256, EPCQ-L512 and EPCQ-L1024.
December 2015	2015.12.14	Added link to EPCQ-L packaging information website.

Date	Version	Changes
January 2015	2015.01.23	<ul style="list-style-type: none"><li>• Updated the package name to FBGA24.</li><li>• Changed erase bulk operation statement for EPCQ-L256 devices.</li><li>• Added stacked die device in 'Features'.</li><li>• Added Number of die column in 'Supported Devices'.</li><li>• Updated Read Bytes and Fast Read operation description to reflect stacked die properties.</li><li>• Added read non-volatile configuration register.</li><li>• Updated AS x1 dummy clock cycles for non-volatile configuration registers.</li><li>• Updated write non-volatile configuration register 16-bit register value.</li></ul>
June 2014	2014.06.17	Initial release.