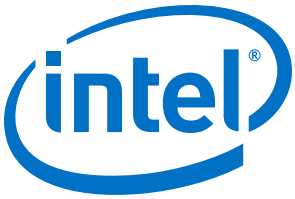


# Quad-Serial Configuration (EPCQ) Devices Datasheet

Subscribe
Send Feedback

CF52012 | 2020.01.23
Latest document on the web: PDF | HTML

# Contents

# 1. Quad-Serial Configuration (EPCQ) Devices Datasheet

This datasheet describes quad-serial configuration (EPCQ) devices. EPCQ is an in-system programmable NOR flash memory.

## 1.1. Supported Devices

### Table 1. Supported Intel EPCQ Devices

*Note:* EPCQ devices are scheduled for product obsolescence and discontinued support as described in PDN1708 and PDN1802. Intel® recommends that you use the EPCQ-A configuration devices.

| Device | Memory Size (bits) | On-Chip Decompression Support | ISP Support | Cascading Support | Reprogrammable | Recommended Operating Voltage (V) |
|---|---|---|---|---|---|---|
| EPCQ16 | 16,777,216 | No | Yes | No | Yes | 3.3 |
| EPCQ32 | 33,554,432 | No | Yes | No | Yes | 3.3 |
| EPCQ64 | 67,108,864 | No | Yes | No | Yes | 3.3 |
| EPCQ128 | 134,217,728 | No | Yes | No | Yes | 3.3 |
| EPCQ256 | 268,435,456 | No | Yes | No | Yes | 3.3 |
| EPCQ512/A (1) | 536,870,912 | No | Yes | No | Yes | 3.3 |

### Related Information

- PDN1708
  Product discontinuance notification.
- PDN1802
  Product discontinuance notification.
- AN822: Intel Configuration Device Migration Guideline
  Provides more information about migrating EPCQ to EPCQ-A devices.
- Quad-Serial Configuration (EPCQ-A) Devices Datasheet

(1) EPCQ512/A is shown in the Intel Quartus® Prime software as EPCQ512.

## 1.2. Features

EPCQ devices offer the following features:

- Serial or quad-serial FPGA configuration in devices that support active serial (AS) x1 or AS x4 configuration schemes(2)
- Low cost, low pin count, and non-volatile memory
- 2.7-V to 3.6-V operation
- Available in 8- or 16- small-outline integrated circuit (SOIC) package
- Reprogrammable memory with up to 100,000 erase or program cycles
- Write protection support for memory sectors using status register bits
- Fast read, extended dual input fast read, and extended quad input fast read of the entire memory using a single operation code
- Write bytes, extended dual input fast write bytes, and extended quad input fast write bytes of the entire memory using a single operation code
- Reprogrammable with an external microprocessor using the SRunner software driver
- In-system programming (ISP) support with the SRunner software driver
- ISP support with Intel FPGA Download CableIntel FPGA Download Cable II, Intel FPGA Ethernet Cable
- By default, the memory array is erased and the bits are set to 1
- More than 20 years data retention

**Related Information**

Errata Sheet for EPCQ Devices

## 1.3. Operating Conditions

Tables in this section list information about the absolute maximum ratings, recommended operating conditions, DC operating conditions, $I_{CC}$ supply current, and capacitance for EPCQ devices.

*Note:* The values of the tables in this section are finalized for EPCQ16, EPCQ32, EPCQ64, EPCQ128, EPCQ256, and EPCQ512/A devices. The operating conditions for all of these devices are the same unless indicated otherwise.

### 1.3.1. Absolute Maximum Ratings

**Table 2. Absolute Maximum Ratings**

| Symbol | Parameter | Condition | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{CC}$ | Supply voltage | With respect to GND | –0.6 | 4 | V |
| $V_I$ (3) | DC input voltage | With respect to GND | –0.6 | 4 | V |
| | | | | | ***continued...*** |

(2) AS x4 is not supported in EPCQ512 devices. Refer to the *Errata Sheet for EPCQ Devices* for more information.

| Symbol | Parameter | Condition | Min | Max | Unit |
|---|---|---|---|---|---|
| $I_{MAX}$ | DC $V_{CC}$ or GND current | — | — | 20 | mA |
| $I_{OUT}$ | DC output current per pin | — | –25 | 25 | mA |
| $P_D$ | Power dissipation | — | — | 72 | mW |
| $T_{STG}$ | Storage temperature | No bias | –65 | 150 | °C |
| $T_J$ | Junction temperature | Under bias | — | 125 | °C |

## 1.3.2. Recommended Operating Conditions

### Table 3. Recommended Operating Conditions

| Symbol | Parameter | Condition | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{CC}$ | Supply voltage | (4) | 2.7 | 3.6 | V |
| $V_I$ | Input voltage | With respect to GND | -0.5 | 0.4 + $V_{CC}$ | V |
| $V_O$ | Output voltage | — | 0 | $V_{CC}$ | V |
| $T_A$ (5) | Operating temperature | For industrial use | -40 | 85 | °C |
| $t_R$ | Input rise time for all devices except EPCQ512/A | — | — | 5 | ns |
| | Input rise time for EPCQ512/A | — | — | 1.5 | ns |
| $t_F$ | Input fall time for all devices except EPCQ512/A | — | — | 5 | ns |
| | Input fall time for EPCQ512/A | — | — | 1.5 | ns |

**Related Information**

EPCQ Package and Thermal Resistance
Provides more information about EPCQ thermal resistance.

## 1.3.3. DC Operating Conditions

### Table 4. DC Operating Conditions

| Symbol | Parameter | Condition | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{IH}$ | High-level input voltage | — | 0.7 x VCC | VCC + 0.4 | V |
| $V_{IL}$ | Low-level input voltage | — | -0.5 | 0.3 x $V_{CC}$ | V |
| $V_{OH}$ | High-level output voltage | $I_{OH}$ = -100 µA (6) | $V_{CC}$ - 0.2 | — | V |

*continued...*

(3) For periods of less than 2 ns, $V_{IL}$ can undershoot to –1.0 V and $V_{IH}$ can overshoot to $V_{CC}$ + 1.0 V.

(4) The maximum $V_{CC}$ rise time is 100 ms.

(5) EPCQ devices can be paired with Intel FPGA industrial-grade FPGAs operating at junction temperatures up to 100°C as long as the ambient temperature does not exceed 85°C.

(6) The $I_{OH}$ parameter refers to the high-level TTL or CMOS output current.

| Symbol | Parameter | Condition | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{OL}$ | Low-level output voltage | $I_{OL}$ = 1.6 mA (7) | — | 0.4 | V |
| $I_I$ | Input leakage current | $V_I$ =$V_{CC}$ or GND | -10 | 10 | µA |
| $I_{OZ}$ | Tri-state output off-state current | $V_O$ = $V_{CC}$ or GND | -10 | 10 | µA |

## 1.3.4. ICC Supply Current

### Table 5. $I_{CC}$ Supply Current

| Symbol | Parameter | Condition | Min | Max | Unit |
|---|---|---|---|---|---|
| $I_{CC0}$ | $V_{CC}$ supply current | Standby | — | 100 | µA |
| $I_{CC1}$ | $V_{CC}$ supply current for all devices except EPCQ512/A | During active power mode | 5 | 20 | mA |
| | $V_{CC}$ supply current for EPCQ512/A | | — | 60 | mA |

## 1.3.5. Capacitance

### Table 6. Capacitance

Capacitance is sample-tested only at $T_A$ = 25 x C and at a 54 MHz frequency.

| Symbol | Parameter | Condition | Min | Max | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input pin capacitance | $V_{IN}$ =0 V | — | 6 | pF |
| $C_{OUT}$ | Output pin capacitance | $V_{OUT}$ =0 V | — | 8 | pF |

# 1.4. Memory Array Organization

### Table 7. Supported Memory Array Organization in EPCQ Devices

| Details | EPCQ16 | EPCQ32 | EPCQ64 | EPCQ128 | EPCQ256 | EPCQ512/A |
|---|---|---|---|---|---|---|
| Bytes | 2,097,152 bytes [16 megabits (Mb)] | 4,194,304 bytes (32 Mb) | 8,388,608 bytes (64 Mb) | 16,777,216 bytes (128 Mb) | 33,554,432 bytes (256 Mb) | 67,108,864 bytes (512 Mb) |
| Number of sectors | 32 | 64 | 128 | 256 | 512 | 1,024 |
| Bytes per sector | 65,536 bytes [512 kilobits (Kb)] | | | | | |
| Total numbers of subsectors (8) | 512 | 1,024 | 2,048 | 4,096 | 8,192 | 16,384 |
| Bytes per subsector | 4,096 bytes (32 Kb) | | | | | |

***continued...***

(7) The $I_{OL}$ parameter refers to the low-level TTL or CMOS output current.

(8) Every sector is further divided into 16 subsectors with 4 KB of memory. Therefore, there are 512 (32 x 16) subsectors for the EPCQ16 device, 1,024 (64 x 16) subsectors for the EPCQ32 device, 2,048 (128 x 16) subsectors for the EPCQ64 device, 4,096 (256 x 16) subsectors for the EPCQ128 device, 8,192 (512 x 16) subsectors for the EPCQ256 device, and 16,384 (1,024 x 16) subsectors for the EPCQ512/A device.

| Details | EPCQ16 | EPCQ32 | EPCQ64 | EPCQ128 | EPCQ256 | EPCQ512/A |
|---|---|---|---|---|---|---|
| Pages per sector | 256 | | | | | |
| Total number of pages | 8,192 | 16,384 | 32,768 | 65,536 | 131,072 | 262,144 |
| Bytes per page | 256 bytes | | | | | |

## 1.4.1. Address Range for EPCQ16

**Table 8. Address Range for Sectors 31..0 and Subsectors 511..0 in EPCQ16 Devices**

| Sector | Subsector | Address Range (Byte Addresses in HEX) | |
|---|---|---|---|
| | | Start | End |
| 31 | 511 | 1FF000 | 1FFFFF |
| | 510 | 1FE000 | 1FEFFF |
| | . | . | . |
| | 498 | 1F2000 | 1F2FFF |
| | 497 | 1F1000 | 1F1FFF |
| | 496 | 1F0000 | 1F0FFF |
| 30 | 495 | 1EF000 | 1EFFFF |
| | 494 | 1EE000 | 1EEFFF |
| | . | . | . |
| | 482 | 1E2000 | 1E2FFF |
| | 481 | 1E1000 | 1E1FFF |
| | 480 | 1E0000 | 1E0FFF |
| 1 | 31 | 1F000 | 1FFFF |
| | 30 | 1E000 | 1EFFF |
| | . | . | . |
| | 18 | 12000 | 12FFF |
| | 17 | 11000 | 11FFF |
| | 16 | 10000 | 10FFF |
| 0 | 15 | F000 | FFFF |
| | 14 | E000 | EFFF |
| | . | . | . |
| | 2 | 2000 | 2FFF |
| | 1 | 1000 | 1FFF |
| | 0 | H'0000000 | H'0000FFF |

## 1.4.2. Address Range for EPCQ32

**Table 9. Address Range for Sectors 63..0 and Subsectors 1023..0 in EPCQ32 Devices**

| Sector | Subsector | Address Range (Byte Addresses in HEX) | |
|---|---|---|---|
| | | Start | End |
| 63 | 1023 | 3FF000 | 3FFFFF |
| | 1022 | 3FE000 | 3FEFFF |
| | . | . | . |
| | 1010 | 3F2000 | 3F2FFF |
| | 1009 | 3F1000 | 3F1FFF |
| | 1008 | 3F0000 | 3F0FFF |
| 62 | 1007 | 3EF000 | 3EFFFF |
| | 1006 | 3EE000 | 3EEFFF |
| | . | . | . |
| | 994 | 3E2000 | 3E2FFF |
| | 993 | 3E1000 | 3E1FFF |
| | 992 | 3E0000 | 3E0FFF |
| 1 | 31 | 1F000 | 1FFFF |
| | 30 | 1E000 | 1EFFF |
| | . | . | . |
| | 18 | 12000 | 12FFF |
| | 17 | 11000 | 11FFF |
| | 16 | 10000 | 10FFF |
| 0 | 15 | F000 | FFFF |
| | 14 | E000 | EFFF |
| | . | . | . |
| | 2 | 2000 | 2FFF |
| | 1 | 1000 | 1FFF |
| | 0 | H'0000000 | H'0000FFF |

## 1.4.3. Address Range for EPCQ64

**Table 10. Address Range for Sectors 127..0 and Subsectors 2047..0 in EPCQ64 Devices**

| Sector | Subsector | Address Range (Byte Addresses in HEX) | |
|---|---|---|---|
| | | Start | End |
| 127 | 2047 | 7FF000 | 7FFFFF |
| | 2046 | 7FE000 | 7FEFFF |
| | . | . | . |

*continued...*

| Sector | Subsector | Address Range (Byte Addresses in HEX) | |
|---|---|---|---|
| | | Start | End |
| | 2034 | 7F2000 | 7F2FFF |
| | 2033 | 7F1000 | 7F1FFF |
| | 2032 | 7F0000 | 7F0FFF |
| 64 | 1039 | 40F000 | 40FFFF |
| | 1038 | 40E000 | 40EFFF |
| | . | . | . |
| | 1026 | 402000 | 402FFF |
| | 1025 | 401000 | 401FFF |
| | 1024 | 400000 | 400FFF |
| 63 | 1023 | 3FF000 | 3FFFFF |
| | 1022 | 3FE000 | 3FEFFF |
| | . | . | . |
| | 1010 | 3F2000 | 3F2FFF |
| | 1009 | 3F1000 | 3F1FFF |
| | 1008 | 3F0000 | 3F0FFF |
| 62 | 1007 | 3EF000 | 3EFFFF |
| | 1006 | 3EE000 | 3EEFFF |
| | . | . | . |
| | 994 | 3E2000 | 3E2FFF |
| | 993 | 3E1000 | 3E1FFF |
| | 992 | 3E0000 | 3E0FFF |
| 1 | 31 | 1F000 | 1FFFF |
| | 30 | 1E000 | 1EFFF |
| | . | . | . |
| | 18 | 12000 | 12FFF |
| | 17 | 11000 | 11FFF |
| | 16 | 10000 | 10FFF |
| 0 | 15 | F000 | FFFF |
| | 14 | E000 | EFFF |
| | . | . | . |
| | 2 | 2000 | 2FFF |
| | 1 | 1000 | 1FFF |
| | 0 | H'0000000 | H'0000FFF |

## 1.4.4. Address Range for EPCQ128

**Table 11. Address Range for Sectors 255..0 and Subsectors 4095..0 in EPCQ128 Devices**

| Sector | Subsector | Address Range (Byte Addresses in HEX) | |
|---|---|---|---|
| | | Start | End |
| 255 | 4095 | FFF000 | FFFFFF |
| | 4094 | FFE000 | FFEFFF |
| | . | . | . |
| | 4082 | FF2000 | FF2FFF |
| | 4081 | FF1000 | FF1FFF |
| | 4080 | FF0000 | FF0FFF |
| 254 | 4079 | FEF000 | FEFFFF |
| | 4078 | FEE000 | FEEFFF |
| | . | . | . |
| | 4066 | FE2000 | FE2FFF |
| | 4065 | FE1000 | FE1FFF |
| | 4064 | FE0000 | FE0FFF |
| 129 | 2079 | 81F000 | 81FFFF |
| | 2078 | 81E000 | 81EFFF |
| | . | . | . |
| | 2066 | 812000 | 812FFF |
| | 2065 | 811000 | 811FFF |
| | 2064 | 810000 | 810FFF |
| 128 | 2063 | 80F000 | 80FFFF |
| | 2062 | 80E000 | 80EFFF |
| | . | . | . |
| | 2050 | 802000 | 802FFF |
| | 2049 | 801000 | 801FFF |
| | 2048 | 800000 | 800FFF |
| 127 | 2047 | 7FF000 | 7FFFFF |
| | 2046 | 7FE000 | 7FEFFF |
| | . | . | . |
| | 2034 | 7F2000 | 7F2FFF |
| | 2033 | 7F1000 | 7F1FFF |
| | 2032 | 7F0000 | 7F0FFF |

*continued...*

| Sector | Subsector | Address Range (Byte Addresses in HEX) | |
|---|---|---|---|
| | | Start | End |
| 64 | 1039 | 40F000 | 40FFFF |
| | 1038 | 40E000 | 40EFFF |
| | . | . | . |
| | 1026 | 402000 | 402FFF |
| | 1025 | 401000 | 401FFF |
| | 1024 | 400000 | 400FFF |
| 63 | 1023 | 3FF000 | 3FFFFF |
| | 1022 | 3FE000 | 3FEFFF |
| | . | . | . |
| | 1010 | 3F2000 | 3F2FFF |
| | 1009 | 3F1000 | 3F1FFF |
| | 1008 | 3F0000 | 3F0FFF |
| 62 | 1007 | 3EF000 | 3EFFFF |
| | 1006 | 3EE000 | 3EEFFF |
| | . | . | . |
| | 994 | 3E2000 | 3E2FFF |
| | 993 | 3E1000 | 3E1FFF |
| | 992 | 3E0000 | 3E0FFF |
| 1 | 31 | 1F000 | 1FFFF |
| | 30 | 1E000 | 1EFFF |
| | . | . | . |
| | 18 | 12000 | 12FFF |
| | 17 | 11000 | 11FFF |
| | 16 | 10000 | 10FFF |
| 0 | 15 | F000 | FFFF |
| | 14 | E000 | EFFF |
| | . | . | . |
| | 2 | 2000 | 2FFF |
| | 1 | 1000 | 1FFF |
| | 0 | H'0000000 | H'0000FFF |

## 1.4.5. Address Range for EPCQ256

**Table 12. Address Range for Sectors 511..0 and Subsectors 8191..0 in EPCQ256 Devices**

| Sector | Subsector | Address Range (Byte Addresses in HEX) | |
|---|---|---|---|
| | | Start | End |
| 511 | 8191 | 1FFF000 | 1FFFFFF |
| | 8190 | 1FFE000 | 1FFEFFF |
| | . | . | . |
| | 8178 | 1FF2000 | 1FF2FFF |
| | 8177 | 1FF1000 | 1FF1FFF |
| | 8176 | 1FF0000 | 1FF0FFF |
| 510 | 8175 | 1FEF000 | 1FEFFFF |
| | 8174 | 1FEE000 | 1FEEFFF |
| | . | . | . |
| | 8162 | 1FE2000 | 1FE2FFF |
| | 8161 | 1FE1000 | 1FE1FFF |
| | 8160 | 1FE0000 | 1FE0FFF |
| 257 | 4127 | 101F000 | 101FFFF |
| | 4126 | 101E000 | 101EFFF |
| | . | . | . |
| | 4114 | 1012000 | 1012FFF |
| | 4113 | 1011000 | 1011FFF |
| | 4112 | 1010000 | 1010FFF |
| 256 | 4111 | 100F000 | 100FFFF |
| | 4110 | 100E000 | 100EFFF |
| | . | . | . |
| | 4098 | 1002000 | 1002FFF |
| | 4097 | 1001000 | 1001FFF |
| | 4096 | 1000000 | 1000FFF |
| 255 | 4095 | FFF000 | FFFFFF |
| | 4094 | FFE000 | FFEFFF |
| | . | . | . |
| | 4082 | FF2000 | FF2FFF |
| | 4081 | FF1000 | FF1FFF |
| | 4080 | FF0000 | FF0FFF |

*continued...*

| Sector | Subsector | Address Range (Byte Addresses in HEX) | |
|---|---|---|---|
| | | Start | End |
| 254 | 4079 | FEF000 | FEFFFF |
| | 4078 | FEE000 | FEEFFF |
| | . | . | . |
| | 4066 | FE2000 | FE2FFF |
| | 4065 | FE1000 | FE1FFF |
| | 4064 | FE0000 | FE0FFF |
| 129 | 2079 | 81F000 | 81FFFF |
| | 2078 | 81E000 | 81EFFF |
| | . | . | . |
| | 2066 | 812000 | 812FFF |
| | 2065 | 811000 | 811FFF |
| | 2064 | 810000 | 810FFF |
| 128 | 2063 | 80F000 | 80FFFF |
| | 2062 | 80E000 | 80EFFF |
| | . | . | . |
| | 2050 | 802000 | 802FFF |
| | 2049 | 801000 | 801FFF |
| | 2048 | 800000 | 800FFF |
| 127 | 2047 | 7FF000 | 7FFFFF |
| | 2046 | 7FE000 | 7FEFFF |
| | . | . | . |
| | 2034 | 7F2000 | 7F2FFF |
| | 2033 | 7F1000 | 7F1FFF |
| | 2032 | 7F0000 | 7F0FFF |
| 64 | 1039 | 40F000 | 40FFFF |
| | 1038 | 40E000 | 40EFFF |
| | . | . | . |
| | 1026 | 402000 | 402FFF |
| | 1025 | 401000 | 401FFF |
| | 1024 | 400000 | 400FFF |
| 63 | 1023 | 3FF000 | 3FFFFF |
| | 1022 | 3FE000 | 3FEFFF |
| | . | . | . |

*continued...*

| Sector | Subsector | Address Range (Byte Addresses in HEX) | |
|---|---|---|---|
| | | Start | End |
| | 1010 | 3F2000 | 3F2FFF |
| | 1009 | 3F1000 | 3F1FFF |
| | 1008 | 3F0000 | 3F0FFF |
| 62 | 1007 | 3EF000 | 3EFFFF |
| | 1006 | 3EE000 | 3EEFFF |
| | . | . | . |
| | 994 | 3E2000 | 3E2FFF |
| | 993 | 3E1000 | 3E1FFF |
| | 992 | 3E0000 | 3E0FFF |
| 1 | 31 | 1F000 | 1FFFF |
| | 30 | 1E000 | 1EFFF |
| | . | . | . |
| | 18 | 12000 | 12FFF |
| | 17 | 11000 | 11FFF |
| | 16 | 10000 | 10FFF |
| 0 | 15 | F000 | FFFF |
| | 14 | E000 | EFFF |
| | . | . | . |
| | 2 | 2000 | 2FFF |
| | 1 | 1000 | 1FFF |
| | 0 | H'0000000 | H'0000FFF |

### 1.4.6. Address Range for EPCQ512/A

**Table 13. Address Range for Sectors 1023..0 and Subsectors 16383..0 in EPCQ512/A Devices**

| Sector | Subsector | Address Range (Byte Addresses in HEX) | |
|---|---|---|---|
| | | Start | End |
| 1023 | 16383 | 3FFF000 | 3FFFFFF |
| | . | . | . |
| | 16368 | 3FF0000 | 3FF0FFF |
| . | . | . | . |
| 511 | 8191 | 1FFF000 | 1FFFFFF |
| | . | . | . |
| | 8176 | FF0000 | 1FF0FFF |
| . | . | . | . |
| 255 | 4095 | FFF000 | FFFFFF |
| | . | . | . |
| | 4080 | FF0000 | FF0FFF |
| . | . | . | . |
| 127 | 2047 | 7FF000 | 7FFFFF |
| | . | . | . |
| | 2032 | 7F0000 | 7F0FFF |
| . | . | . | . |
| 63 | 1023 | 3FF000 | 3FFFFF |
| | . | . | . |
| | 1008 | 3F0000 | 3F0FFF |
| . | . | . | . |
| 0 | 15 | F000 | FFFF |
| | . | . | . |
| | 0 | H'0000000 | H'0000FFF |

## 1.5. Memory Operations

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

### 1.5.1. Timing Requirements

When the active low chip select (nCS) signal is driven low, shift in the operation code into the EPCQ device using the serial data (DATA) pin. Each operation code bit is latched into the EPCQ 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. See related information for more information about the address and data bytes. 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 DATA 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.

### 1.5.2. Addressing Mode

The 3-byte addressing mode is enabled by default. To access the EPCQ256 or EPCQ512/A memory, you must use the 4-byte addressing mode. In 4-byte addressing mode, the address width is 32-bit address. 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 Intel Quartus Prime software or the SRunner software to program the EPCQ256 or EPCQ512/A device, you do not need to execute the 4BYTEADDREN operation. The Intel Quartus Prime software enables the 4-byte addressing mode when programming the device automatically.

## 1.6. Registers

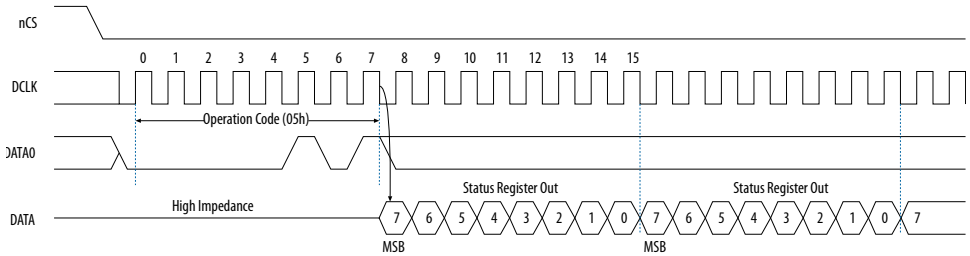
### 1.6.1. Status Register

**Table 14. Status Register Bits**

| Bit | R/W | Default Value | Name | Value | Description |
|---|---|---|---|---|---|
| 7 | R/W | 0 | None | | |
| 6 | R/W | 0 | BP3 (Block Protect Bit)[9] [10] | Table 15 on page 19 through Table 26 on page 24 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 | R/W | 0 | TB (Top/Bottom Bit) | • 1=Protected area starts from the bottom of the memory array.<br>• 0=Protected area starts from the top of the memory array. | Determine that the protected area starts from the top or bottom of the memory array. |
| 4 | R/W | 0 | BP2[9] | Table 15 on page 19 through Table 26 on page 24 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[9] | | |
| 2 | | | BP0[9] | | |
| 1 | R | 0 | WEL (Write Enable Latch Bit) | • 1=Allows the following operation to run:<br>— Write Bytes<br>— Write Status Register<br>— Erase Bulk<br>— Erase Die<br>— Erase Sector<br>• 0=Rejects the above mentioned operations. | Allows or rejects certain operation to run. |
| 0 | R | 0 | WIP (Write in Progress Bit) | • 1=One of the following operation is in progress:<br>— Write Status Register<br>— Write NVCR<br>— Write Bytes<br>— Erase<br>• 0=no write or erase cycle in progress | Indicates if there is a command in progress. |

### 1.6.1.1. Read Status Register Operation (05h)

**Figure 1. Read Status Operation Timing Diagram**




(9) 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.

(10) Applicable for EPCQ64, EPCQ128, EPCQ256, and EPCQ512/A devices only.

#### 1.6.1.1.1. Block Protection Bits in EPCQ16 when TB Bit is Set to 0

**Table 15. Block Protection Bits in EPCQ16 when TB Bit is Set to 0**

| Status Register Content | | | | Memory Content | |
|---|---|---|---|---|---|
| TB Bit | BP2 Bit | BP1 Bit | BP0 Bit | Protected Area | Unprotected Area |
| 0 | 0 | 0 | 0 | None | All sectors |
| 0 | 0 | 0 | 1 | Sector 31 | Sectors (0 to 30) |
| 0 | 0 | 1 | 0 | Sectors (30 to 31) | Sectors (0 to 29) |
| 0 | 0 | 1 | 1 | Sectors (28 to 31) | Sectors (0 to 27) |
| 0 | 1 | 0 | 0 | Sectors (24 to 31) | Sectors (0 to 23) |
| 0 | 1 | 0 | 1 | Sectors (16 to 31) | Sectors (0 to 15) |
| 0 | 1 | 1 | 0 | All sectors | None |
| 0 | 1 | 1 | 1 | All sectors | None |

#### 1.6.1.1.2. Block Protection Bits in EPCQ16 when TB Bit is Set to 1

**Table 16. Block Protection Bits in EPCQ16 when TB Bit is Set to 1**

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

#### 1.6.1.1.3. Block Protection Bits in EPCQ32 when TB Bit is Set to 0

**Table 17. Block Protection Bits in EPCQ32 when TB Bit is Set to 0**

| Status Register Content | | | | Memory Content | |
|---|---|---|---|---|---|
| TB Bit | BP2 Bit | BP1 Bit | BP0 Bit | Protected Area | Unprotected Area |
| 0 | 0 | 0 | 0 | None | All sectors |
| 0 | 0 | 0 | 1 | Sector 63 | Sectors (0 to 62) |
| 0 | 0 | 1 | 0 | Sectors (62 to 63) | Sectors (0 to 61) |
| 0 | 0 | 1 | 1 | Sectors (60 to 63) | Sectors (0 to 59) |
| 0 | 1 | 0 | 0 | Sectors (56 to 63) | Sectors (0 to 55) |

***continued...***

| Status Register Content | | | | Memory Content | |
|---|---|---|---|---|---|
| TB Bit | BP2 Bit | BP1 Bit | BP0 Bit | Protected Area | Unprotected Area |
| 0 | 1 | 0 | 1 | Sectors (48 to 63) | Sectors (0 to 47) |
| 0 | 1 | 1 | 0 | Sectors (32 to 63) | Sectors (0 to 31) |
| 0 | 1 | 1 | 1 | All sectors | None |

#### 1.6.1.1.4. Block Protection Bits in EPCQ32 when TB Bit is Set to 1

**Table 18. Block Protection Bits in EPCQ32 when TB Bit is Set to 1**

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

#### 1.6.1.1.5. Block Protection Bits in EPCQ64 when TB Bit is Set to 0

**Table 19. Block Protection Bits in EPCQ64 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 127 | Sectors (0 to 126) |
| 0 | 0 | 0 | 1 | 0 | Sectors (126 to 127) | Sectors (0 to 125) |
| 0 | 0 | 0 | 1 | 1 | Sectors (124 to 127) | Sectors (0 to 123) |
| 0 | 0 | 1 | 0 | 0 | Sectors (120 to 127) | Sectors (0 to 119) |
| 0 | 0 | 1 | 0 | 1 | Sectors (112 to 127) | Sectors (0 to 111) |
| 0 | 0 | 1 | 1 | 0 | Sectors (96 to 127) | Sectors (0 to 95) |
| 0 | 0 | 1 | 1 | 1 | Sectors (64 to 127) | Sectors (0 to 63) |
| 0 | 1 | 0 | 0 | 0 | All sectors | None |
| 0 | 1 | 0 | 0 | 1 | All sectors | None |
| 0 | 1 | 0 | 1 | 0 | All sectors | None |
| 0 | 1 | 0 | 1 | 1 | All sectors | None |
| 0 | 1 | 1 | 0 | 0 | All sectors | None |

*continued...*

| Status Register Content | | | | | Memory Content | |
|---|---|---|---|---|---|---|
| **TB Bit** | **BP3 Bit** | **BP2 Bit** | **BP1 Bit** | **BP0 Bit** | **Protected Area** | **Unprotected Area** |
| 0 | 1 | 1 | 0 | 1 | All sectors | None |
| 0 | 1 | 1 | 1 | 0 | All sectors | None |
| 0 | 1 | 1 | 1 | 1 | All sectors | None |

#### 1.6.1.1.6. Block Protection Bits in EPCQ64 when TB Bit is Set to 1

**Table 20. Block Protection Bits in EPCQ64 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 127) |
| 1 | 0 | 0 | 1 | 0 | Sectors (0 to 1) | Sectors (2 to 127) |
| 1 | 0 | 0 | 1 | 1 | Sectors (0 to 3) | Sectors (4 to 127) |
| 1 | 0 | 1 | 0 | 0 | Sectors (0 to 7) | Sectors (8 to 127) |
| 1 | 0 | 1 | 0 | 1 | Sectors (0 to 15) | Sectors (16 to 127) |
| 1 | 0 | 1 | 1 | 0 | Sectors (0 to 31) | Sectors (32 to 127) |
| 1 | 0 | 1 | 1 | 1 | Sectors (0 to 63) | Sectors (64 to 127) |
| 1 | 1 | 0 | 0 | 0 | All sectors | None |
| 1 | 1 | 0 | 0 | 1 | All sectors | None |
| 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 |

#### 1.6.1.1.7. Block Protection Bits in EPCQ128 when TB Bit is Set to 0

**Table 21. Block Protection Bits in EPCQ128 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 255 | Sectors (0 to 254) |
| 0 | 0 | 0 | 1 | 0 | Sectors (254 to 255) | Sectors (0 to 253) |
| 0 | 0 | 0 | 1 | 1 | Sectors (252 to 255) | Sectors (0 to 251) |
| 0 | 0 | 1 | 0 | 0 | Sectors (248 to 255) | Sectors (0 to 247) |
| 0 | 0 | 1 | 0 | 1 | Sectors (240 to 255) | Sectors (0 to 239) |

***continued...***

| Status Register Content | | | | | Memory Content | |
|---|---|---|---|---|---|---|
| TB Bit | BP3 Bit | BP2 Bit | BP1 Bit | BP0 Bit | Protected Area | Unprotected Area |
| 0 | 0 | 1 | 1 | 0 | Sectors (224 to 255) | Sectors (0 to 223) |
| 0 | 0 | 1 | 1 | 1 | Sectors (192 to 255) | Sectors (0 to 191) |
| 0 | 1 | 0 | 0 | 0 | Sectors (128 to 255) | Sectors (0 to 127) |
| 0 | 1 | 0 | 0 | 1 | All sectors | None |
| 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 |

#### 1.6.1.1.8. Block Protection Bits in EPCQ128 when TB Bit is Set to 1

**Table 22. Block Protection Bits in EPCQ128 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 255) |
| 1 | 0 | 0 | 1 | 0 | Sectors (0 to 1) | Sectors (2 to 255) |
| 1 | 0 | 0 | 1 | 1 | Sectors (0 to 3) | Sectors (4 to 255) |
| 1 | 0 | 1 | 0 | 0 | Sectors (0 to 7) | Sectors (8 to 255) |
| 1 | 0 | 1 | 0 | 1 | Sectors (0 to 15) | Sectors (16 to 255) |
| 1 | 0 | 1 | 1 | 0 | Sectors (0 to 31) | Sectors (32 to 255) |
| 1 | 0 | 1 | 1 | 1 | Sectors (0 to 63) | Sectors (64 to 255) |
| 1 | 1 | 0 | 0 | 0 | Sectors (0 to 127) | Sectors (128 to 255) |
| 1 | 1 | 0 | 0 | 1 | All sectors | None |
| 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 |

#### 1.6.1.1.9. Block Protection Bits in EPCQ256 when TB Bit is Set to 0

**Table 23. Block Protection Bits in EPCQ256 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 |

#### 1.6.1.1.10. Block Protection Bits in EPCQ256 when TB Bit is Set to 1

**Table 24. Block Protection Bits in EPCQ256 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 |

***continued...***

| Status Register Content | | | | | Memory Content | |
|---|---|---|---|---|---|---|
| TB Bit | BP3 Bit | BP2 Bit | BP1 Bit | BP0 Bit | Protected Area | Unprotected Area |
| 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 |

#### 1.6.1.1.11. Block Protection Bits in EPCQ512/A when TB is Set to 0

**Table 25. Block Protection Bits in EPCQ512/A 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) |
| 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 |

#### 1.6.1.1.12. Block Protection Bits in EPCQ512/A when TB is Set to 1

**Table 26. Block Protection Bits in EPCQ512/A 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) |

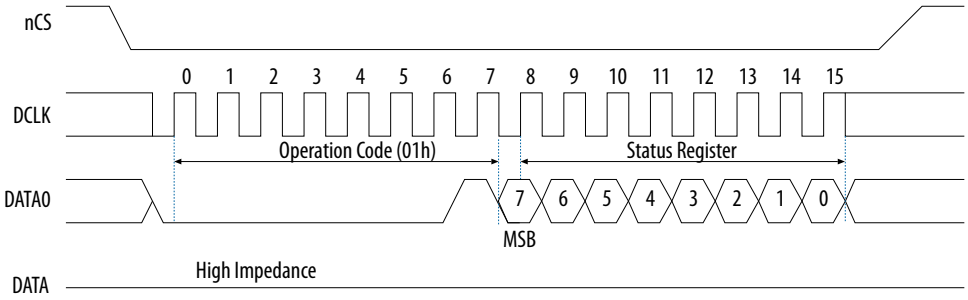
*continued...*

| Status Register Content | | | | | Memory Content | |
|---|---|---|---|---|---|---|
| **TB Bit** | **BP3 Bit** | **BP2 Bit** | **BP1 Bit** | **BP0 Bit** | **Protected Area** | **Unprotected Area** |
| 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 |

### 1.6.1.2. Write Status Register Operation (01h)

The write status operation does not affect the write enable latch and write in progress bits. You can use the write status operation to set the status register block protection and top or bottom bits. Therefore, you can implement this operation to protect certain memory sectors. 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.

**Figure 2. Write Status 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 devices and is guaranteed to be less than 8 ms. For details about $t_{WS}$, refer to the related information below. 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 operation while the self-timed write status cycle is in progress. Set the write in progress bit to 1 during the self-timed write status cycle and 0 when it is complete.

**Related Information**

Write Operation Timing on page 40
The Write Operation Parameters provides more information about $t_{WS}$, $t_{ES}$ and $t_{WB}$.

## 1.6.2. Flag Status Register

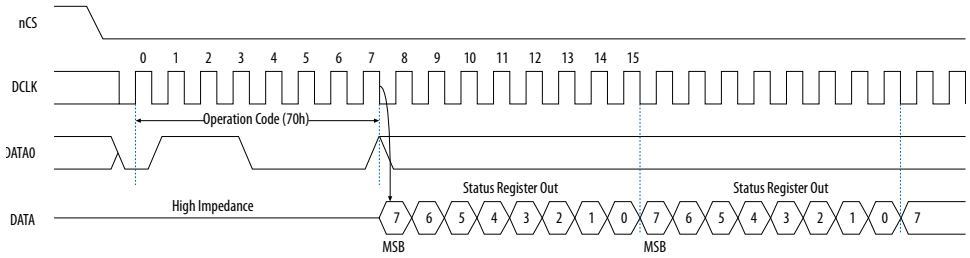
**Table 27. Flag Status Register Bits**

| Bit | Name | Value | Description |
|---|---|---|---|
| 7 | Write or Erase Controller(11) | • 1=Ready<br>• 0=Busy | Indicates whether one of the following operation is in progress:<br>• Write Status Register<br>• Write NVCR<br>• Write Bytes<br>• Erase |
| 6 | Erase suspend | • 1=In effect<br>• 0=Not in effect | Indicates whether an Erase operation has been or is going to be suspended.<br>*Note:* Status bits are reset automatically |
| 5 | Erase | • 1=Failure or protection error<br>• 0=Clear | Indicates whether an Erase operation has succeeded or failed. |
| 4 | Write | • 1=Failure or protection error<br>• 0=Clear | 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 | Reserved | | |
| 2 | Write suspend | • 1=In effect<br>• 0=Not in effect | Indicates whether a Write Bytes operation has been or will be suspended. |
| 1 | Protection | • 1=Failure or protection error<br>• 0=Clear | Indicates whether an Erase or Write Bytes operation has attempted to modify the protected array sector. |
| 0 | Addressing | • 1=4-bytes addressing<br>• 0=3-bytes addressing | Indicates the addressing mode used. |

### 1.6.2.1. Read Flag Status Register Operation(70h)

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 3. Read Flag Status Register Operation Timing Diagram**




(11) Write or erase controller bit = NOT write in progress bit.

## 1.6.3. Non-Volatile Configuration Register

**Table 28. Dummy Clock Cycles and Address Bytes for the Non-Volatile Configuration Register Operation**

| FPGA Device | Dummy Clock Cycles | |
|---|---|---|
| | AS x1 | AS x4 |
| • Arria® GX<br>• Arria II<br>• Cyclone®<br>• Cyclone II<br>• Cyclone III<br>• Cyclone IV<br>• Stratix®<br>• Stratix GX<br>• Stratix II<br>• Stratix II GX<br>• Stratix III<br>• Stratix IV<br>• Intel Cyclone 10 LP | 8 | — |

**Table 29. Dummy Clock Cycles and Address Bytes for the Non-Volatile Configuration Register Operation for Arria V, Cyclone V and Stratix V Devices**

| FPGA Device | Address Bytes(12) | Dummy Clock Cycles | |
|---|---|---|---|
| | | AS x1 | AS x4 |
| • Arria V<br>• Cyclone V<br>• Stratix V | 3-byte addressing | 12 | 12 |
| | 4-byte addressing | 4 | 10 |

**Table 30. Non-Volatile Configuration Register Operation Bit Definition**

| Bit | Description | Default Value |
|---|---|---|
| 15:12 | Number of dummy clock cycles. When this number is from 0001 to 1110, the dummy clock cycles is from 1 to 14. | 0000 or 1111 (13) |
| 11:5 | Set these bits to 1111111. | 1111111 |
| 4 | Don't care. | 1 |
| 3:1 | Set these bits to 111. | 111 |
| 0 | Address byte setting.<br>• 0—4-byte addressing<br>• 1—3-byte addressing | 1 |

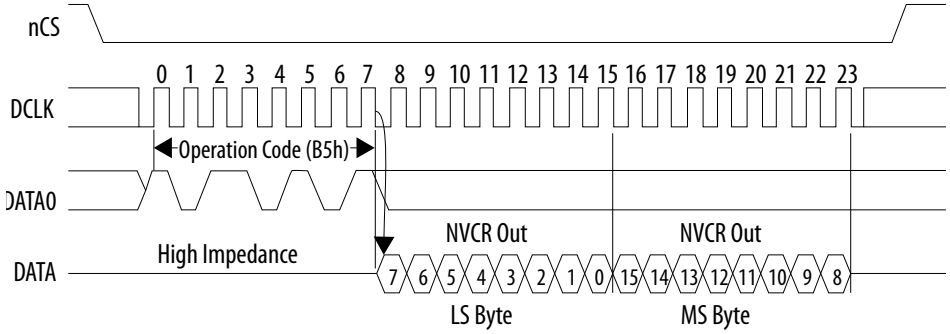
(12) The 4-byte addressing mode is used for EPCQ256 and EPCQ512/A devices.

(13) The default dummy clock cycles is 10 for extended quad input fast read and 8 for extended dual input fast and standard fast read.

### 1.6.3.1. Read Non-Volatile Configuration Register Operation (B5h)

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 4. Read Non-Volatile Configuration Register Operation Timing Diagram**




### 1.6.3.2. Write Non-Volatile Configuration Register Operation (B1h)

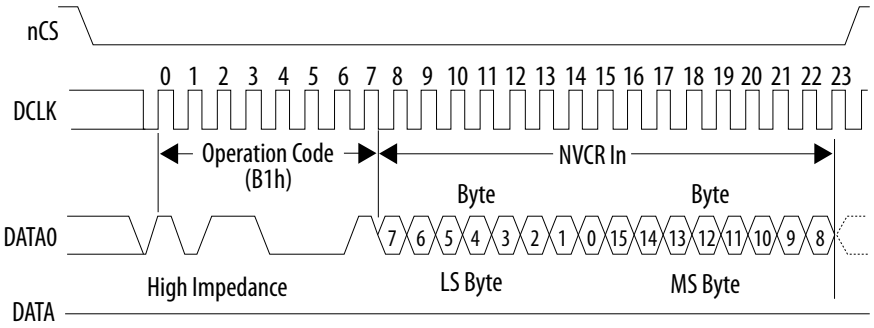
You need to write the non-volatile configuration registers for EPCQ devices for different configuration schemes. If you are using the **.jic** file, the Intel Quartus Prime programmer sets the number of dummy clock cycles and address bytes accordingly. If you are using an external programmer tools, you 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 111y xxxx 1111` where `y` is the address byte (`0` for 4-byte addressing and `1` for 3-byte addressing) and `xxxx` is the dummy clock value. When the `xxxx` value is from `0001` to `1110`, the dummy clock value is from 1 to 14. When `xxxx` 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 5. Write Non-Volatile Configuration Register Operation Timing Diagram**




## 1.7. Summary of Operation Codes

**Table 31. Summary of Operation Codes**

| Operation | Operation Code[14] | Address Bytes | Dummy Clock Cycles | Data Bytes | DCLK $f_{MAX}$ (MHz) |
|---|---|---|---|---|---|
| Read status register | 05h | 0 | 0 | 1 to infinite[15] | 100 |
| Read flag status register | 70h | 0 | 0 | 1 to infinite | 100 |
| Read bytes | 03h | 3 or 4 | 0 | 1 to infinite[15] | 50 |
| Read non-volatile configuration register | B5h | 0 | 0 | 2 | 100 |
| Read device identification | 9Fh | 0 | 2 | 1 | 100 |
| Fast read | 0Bh | 3 or 4 | 8[16] | 1 to infinite[15] | 100 |
| Extended dual input fast read | BBh | 3 or 4 | 8[16] | 1 to infinite[15] | 100 |
| Extended quad input fast read | EBh | 3 or 4 | 10[16] | 1 to infinite[15] | 100 |
| Write enable | 06h | 0 | 0 | 0 | 100 |
| Write disable | 04h | 0 | 0 | 0 | 100 |
| Write status | 01h | 0 | 0 | 1 | 100 |
| Write bytes | 02h | 3 or 4 | 0 | 1 to 256[17] | 100 |
| Write non-volatile configuration register | B1h | 0 | 0 | 2 | 100 |

*continued...*

(14) List MSB first and LSB last.

(15) The status register or data, is read out at least once and is continuously read out until the nCS pin is driven high.

(16) You can configure the number of dummy clock cycles. Refer to Non-Volatile Configuration Register on page 27 for more information.

(17) 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.

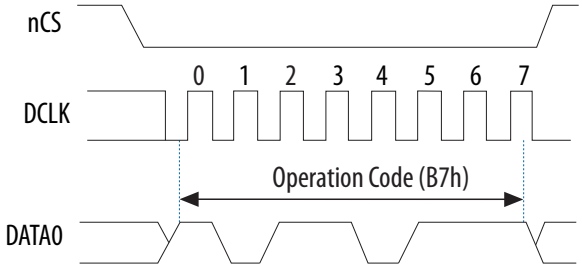
| Operation | Operation Code[14] | Address Bytes | Dummy Clock Cycles | Data Bytes | DCLK $f_{MAX}$ (MHz) |
|---|---|---|---|---|---|
| Extended dual input fast write bytes | D2h | 3 or 4 | 0 | 1 to 256[17] | 100 |
| Extended quad input fast write bytes for EPCQ16, EPCQ32, EPCQ64, EPCQ128 and EPCQ256 devices | 12h | 3 or 4 | 0 | 1 to 256[17] | 100 |
| Extended quad input fast write bytes for EPCQ512/A devices | 38h | 3 or 4 | 0 | 1 to 256[17] | 100 |
| Erase bulk | C7h | 0 | 0 | 0 | 100 |
| Erase sector | D8h | 3 or 4 | 0 | 0 | 100 |
| Erase subsector | 20h | 3 | 0 | 0 | 100 |
| 4BYTEADDREN[18] | B7h | 0 | 0 | 0 | 100 |
| 4BYTEADDREX[18] | E9h | 0 | 0 | 0 | 100 |

### 1.7.1. 4BYTEADDREN and 4BYTEADDREX Operations (B7h and E9h)

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.

*Note:* You must execute a write enable operation before you can enable the 4BYTEADDREN or 4BYTEADDREX operation.

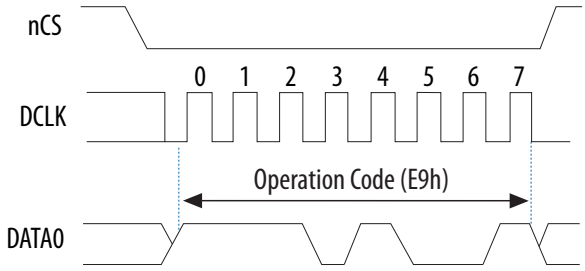
**Figure 6. 4BYTEADDREN Timing Diagram**




(14) List MSB first and LSB last.

(18) This operation is applicable for EPCQ256 and EPCQ512/A devices only.

**Figure 7. 4BYTEADDREX Timing Diagram**



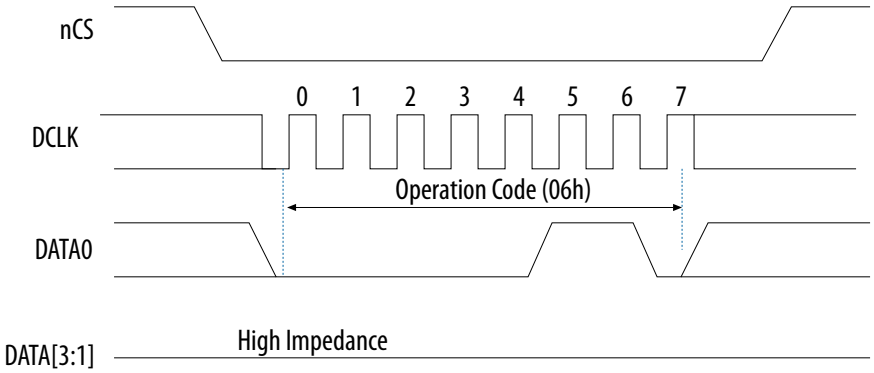

**Related Information**

Write Enable Operation (06h) on page 31

## 1.7.2. Write Enable Operation (06h)

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, extended dual input fast write bytes, extended quad input fast write bytes, 4BYTEADDREN, and 4BYTEADDREX operations.

**Figure 8. Write Enable Operation Timing Diagram**

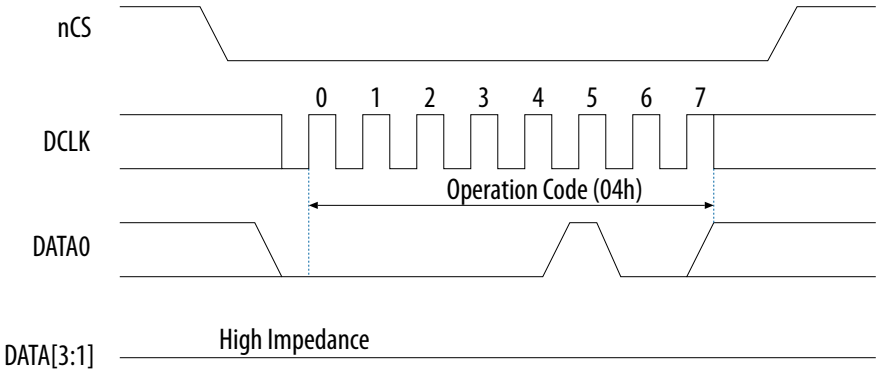



## 1.7.3. Write Disable Operation (04h)

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
- Extended dual input fast write bytes operation completion
- Extended quad input fast write bytes operation completion

**Figure 9. Write Disable Operation Timing Diagram**

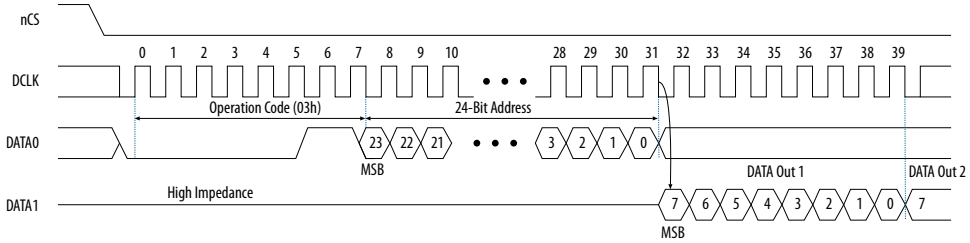



## 1.7.4. Read Bytes Operation (03h)

When you execute the read bytes operation, you first shift in the read bytes operation code, followed by a 3-byte addressing mode (`A[23..0]`) or a 4-byte addressing mode (`A[31..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 DATA1 pin, beginning with the MSB. For reading 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 10. Read Bytes Operation Timing Diagram**

To access the entire EPCQ256 or EPCQ512/A memory, use 4-byte addressing mode. In the 4-byte addressing mode, the address width is 32-bit address.




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. 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.

### Related Information

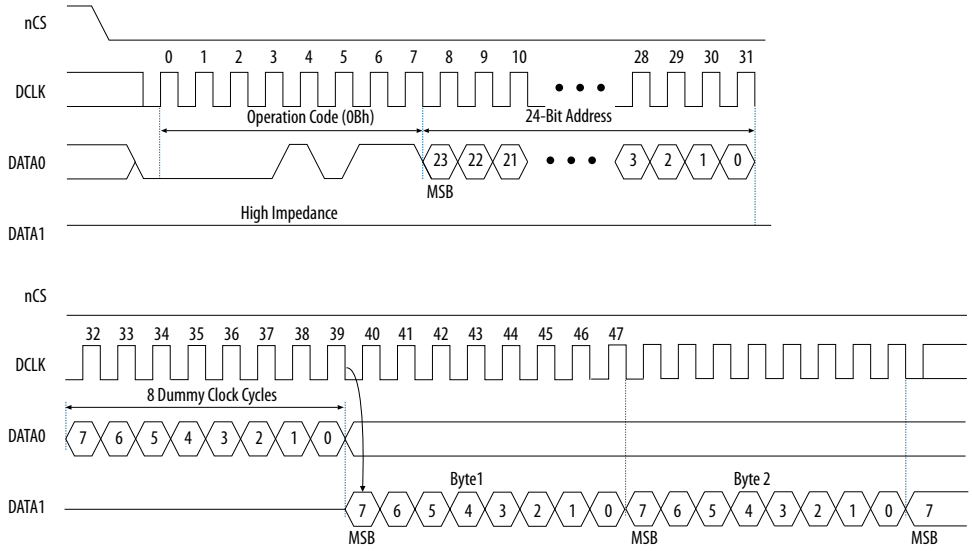
Write Operation Timing on page 40
The Write Operation Parameters provides more information about $t_{WS}$, $t_{ES}$ and $t_{WB}$.

## 1.7.5. Fast Read Operation (0Bh)

When you execute the fast read operation, you first shift in the fast read operation code, followed by a 3-byte addressing mode (`A[23..0]`) or a 4-byte addressing mode (`A[31..0]`), and dummy clock 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 11. Fast Read Operation Timing Diagram**

To access the entire EPCQ256 or EPCQ512/A memory, use 4-byte addressing mode. In the 4-byte addressing mode, the address width is 32-bit address.




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 fast read operation. When the device reaches the highest address, the address counter restarts at `0x000000`, allowing the read sequence to continue indefinitely.

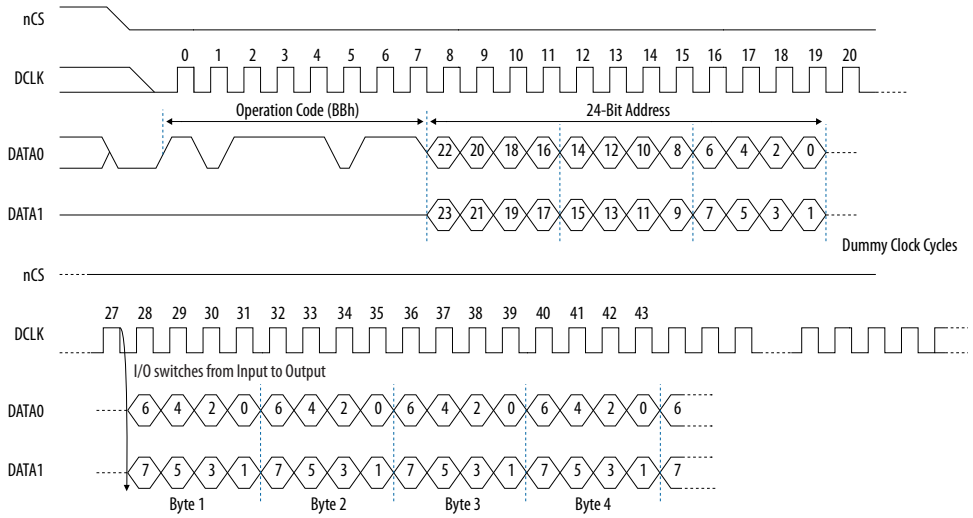
You can terminate the fast read operation by driving the nCS signal high at any time during data output. If the fast read operation is shifted in while an erase, program, or write cycle is in progress, the operation is not executed and does not affect the erase, program, or write cycle in progress.

## 1.7.6. Extended Dual Input Fast Read Operation (BBh)

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

**Figure 12. Extended Dual Input Fast Read Operation Timing Diagram**

To access the entire EPCQ256 or EPCQ512/A memory, use 4-byte addressing mode. In the 4-byte addressing mode, the address width is 32-bit address.

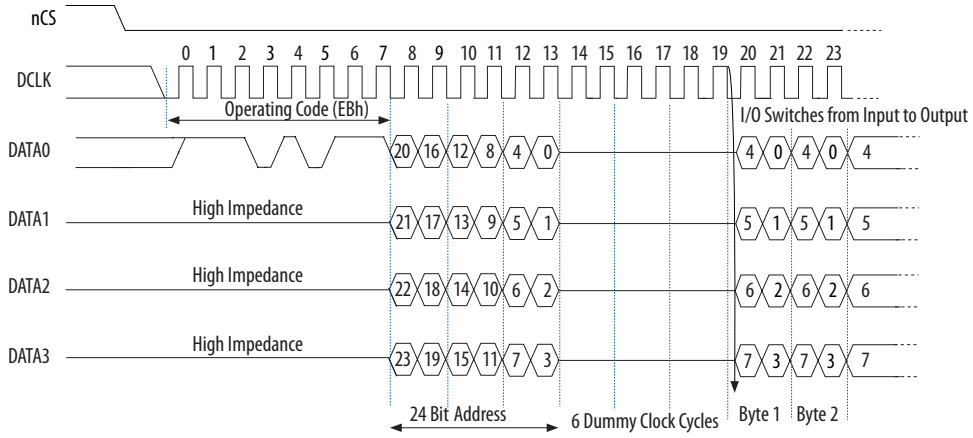



## 1.7.7. Extended Quad Input Fast Read Operation (EBh)

This operation is similar to the extended dual input 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**

To access the entire EPCQ256 or EPCQ512/A memory, use 4-byte addressing mode. In the 4-byte addressing mode, the address width is 32-bit address.




## 1.7.8. Read Device Identification Operation (9Fh)

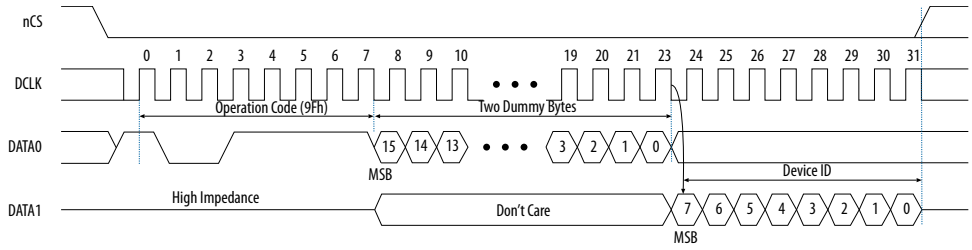
This operation reads the 8-bit device identification of the EPCQ device from the DATA1 output pin. If this operation is shifted in while an erase or write cycle is in progress, the operation is not executed and does not affect the erase or write cycle in progress.

**Table 32. EPCQ Device Identification**

| EPCQ Device | Silicon ID (Binary Value) |
|---|---|
| EPCQ16 | b'0001 0101 |
| EPCQ32 | b'0001 0110 |
| EPCQ64 | b'0001 0111 |
| EPCQ128 | b'0001 1000 |
| EPCQ256 | b'0001 1001 |
| EPCQ512/A | b'0010 0000 |

The 8-bit device identification of the EPCQ device is shifted out on the DATA1 pin on the falling edge of the DCLK signal.

**Figure 14. Read Device Identification Operation Timing Diagram**



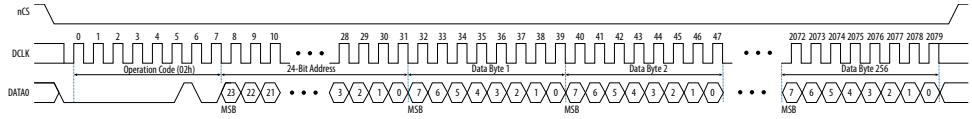

## 1.7.9. Write Bytes Operation (02h)

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 3-byte addressing mode (A[23..0]) or 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.

**Figure 15. Write Bytes Operation Timing Diagram**

To access the entire EPCQ256 or EPCQ512/A memory, use 4-byte addressing mode. In the 4-byte addressing mode, the address width is 32-bit address.




If more than 256 data bytes are shifted into the EPCQ 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 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 the related information below. 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.

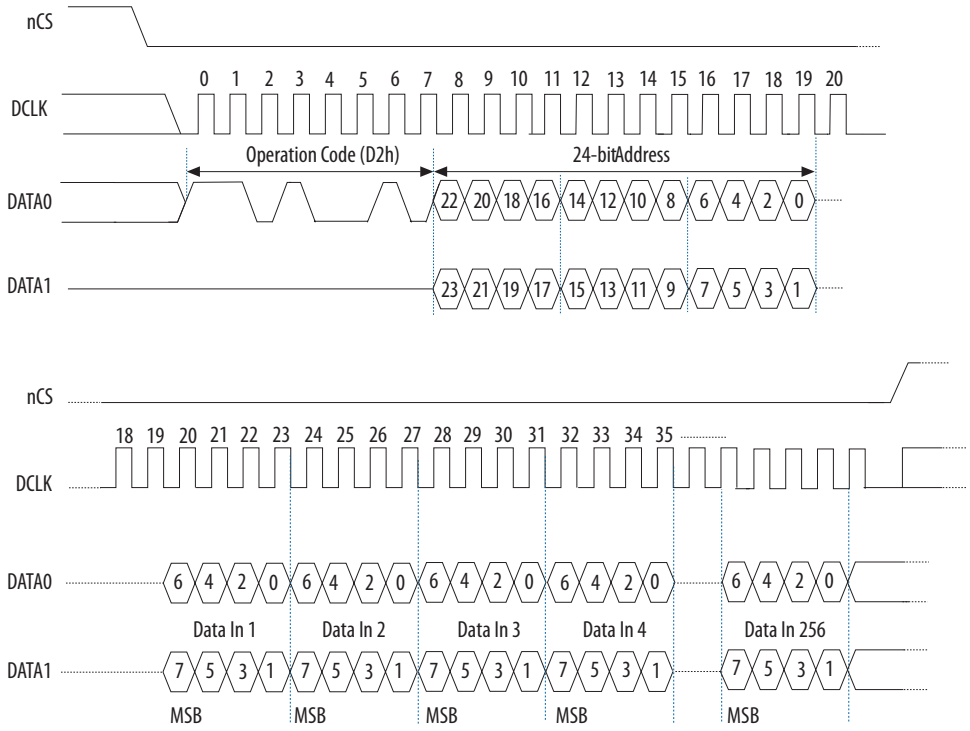
*Note:* You must erase all the memory bytes of EPCQ devices before you implement the write bytes operation. You can erase all the memory bytes by executing the erase sector operation in a sector or the erase bulk operation throughout the entire memory

## 1.7.10. Extended Dual Input Fast Write Bytes Operation (D2h)

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

**Figure 16. Extended Dual Input Fast Write Bytes Timing Diagram**

To access the entire EPCQ256 or EPCQ512/A memory, use 4-byte addressing mode. In the 4-byte addressing mode, the address width is 32-bit address.




## 1.7.11. Extended Quad Input Fast Write Bytes Operation (12h or 38h)

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

**Figure 17. Extended Quad Input Fast Write Bytes Operation Timing Diagram for EPCQ16, EPCQ32, EPCQ64, EPCQ128 and EPCQ256**

To access the entire EPCQ256 memory, use 4-byte addressing mode. In the 4-byte addressing mode, the address width is 32-bit address.

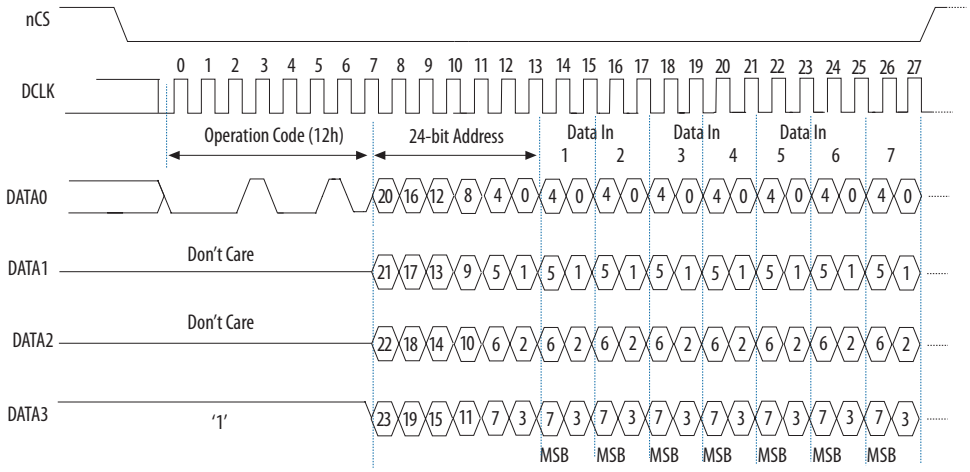
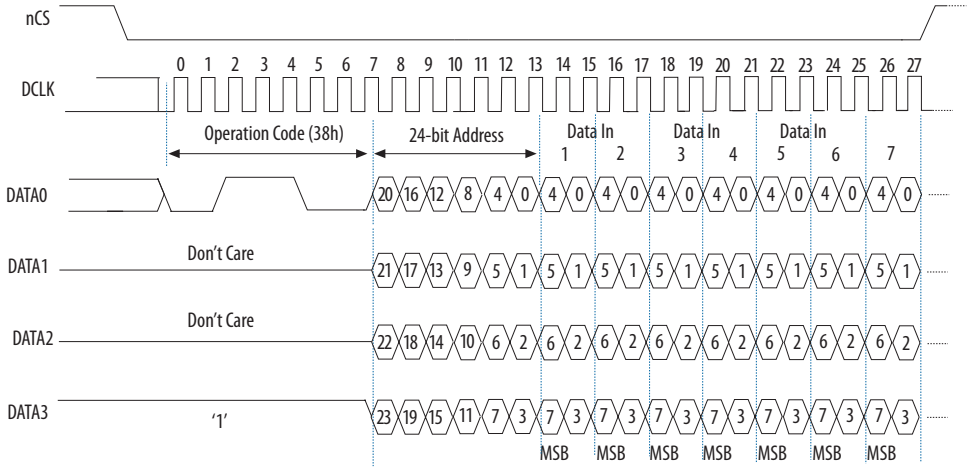



**Figure 18. Extended Quad Input Fast Write Bytes Operation Timing Diagram for EPCQ512/A Devices**

To access the entire EPCQ512/A memory, use 4-byte addressing mode. In the 4-byte addressing mode, the address width is 32-bit address.

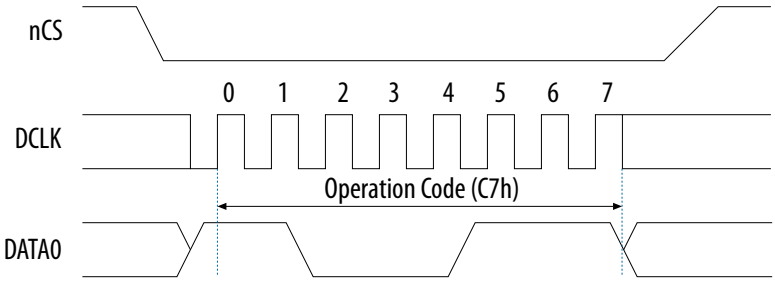



## 1.7.12. Erase Bulk Operation (C7h)

This operation sets all the memory bits to 1or 0xFF. Similar to the write bytes operation, you must execute the write enable operation before 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 19. 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 $t_{EB}$ in the related information below.

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.

**Related Information**

Write Operation Timing on page 40
The Write Operation Parameters provides more information about $t_{WS}$, $t_{ES}$ and $t_{WB}$.

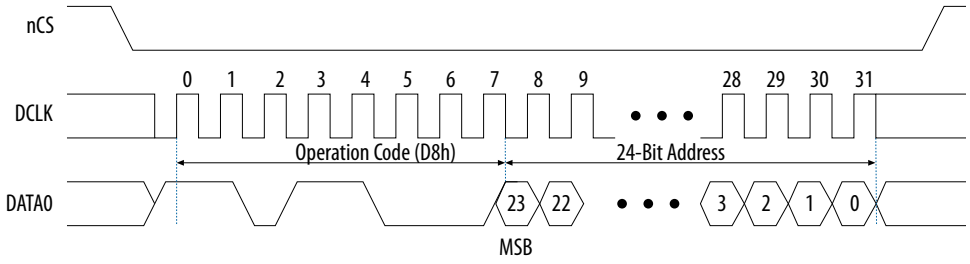
## 1.7.13. Erase Sector Operation (D8h)

The erase sector operation allows you to erase a certain sector in the EPCQ 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 3-byte addressing mode (A[23..0]) or the 4-byte addressing mode (A[31..0]) of the chosen sector on the DATA0 pin. The 3-byte addressing mode or the 4-byte addressing mode for the erase sector operation can be any address inside the specified sector. Drive the nCS signal high after the eighth bit of the erase sector operation code has been latched in.

**Figure 20. Erase Sector Operation Timing Diagram**

To access the entire EPCQ256 or EPCQ512/A memory, use 4-byte addressing mode. In the 4-byte addressing mode, the address width is 32-bit address.



The device initiates a self-timed erase sector cycle immediately after the nCS signal is driven high. For details about the self-timed erase sector cycle time, refer to $t_{ES}$ in the related information below. 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.

**Related Information**

Write Operation Timing on page 40
The Write Operation Parameters provides more information about $t_{WS}$, $t_{ES}$ and $t_{WB}$.

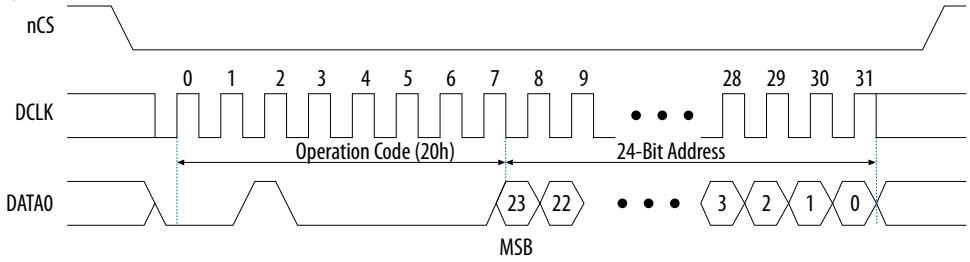
## 1.7.14. Erase Subsector Operation

The erase subsector operation allows you to erase a certain subsector in the EPCQ device by setting all the bits inside the subsector to 1 or 0xFF. This operation is useful if you want to access the unused subsectors as a general purpose memory in your applications. You must execute the write enable operation before the erase subsector operation.

When you execute the erase subsector operation, you must first shift in the erase subsector operation code, followed by the 3-byte addressing mode (A[23..0]) or the 4-byte addressing mode (A[31..0]) of the chosen subsector on the DATA0 pin. The 3-byte addressing mode or the 4-byte addressing mode for the erase subsector operation can be any address inside the specified subsector. For details about the subsector address range, refer to the related information below. Drive the nCS signal high after the eighth bit of the erase subsector operation code has been latched in.

**Figure 21. Erase Subsector Operation Timing Diagram**

To access the entire EPCQ256 or EPCQ512/A memory, use 4-byte addressing mode. In the 4-byte addressing mode, the address width is 32-bit address.




The device initiates a self-timed erase subsector cycle immediately after the nCS signal is driven high. For details about the self-timed erase subsector cycle time, refer to related the information below. 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.

**Related Information**

- Write Operation Timing on page 40

- Memory Array Organization on page 7

## 1.8. Power Mode

EPCQ 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 EPCQ 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 EPCQ device then goes into standby power mode. The $I_{CC1}$ and $I_{CC0}$ parameters list the $V_{CC}$ supply current when the device is in active and standby power modes.

## 1.9. Timing Information

*Note:* The values in the following tables are finalized for EPCQ16, EPCQ32, EPCQ64, EPCQ128, EPCQ256, and EPCQ512/A devices.

**Related Information**

Summary of Operation Codes on page 29

### 1.9.1. Write Operation Timing

**Figure 22. Write Operation Timing Diagram**

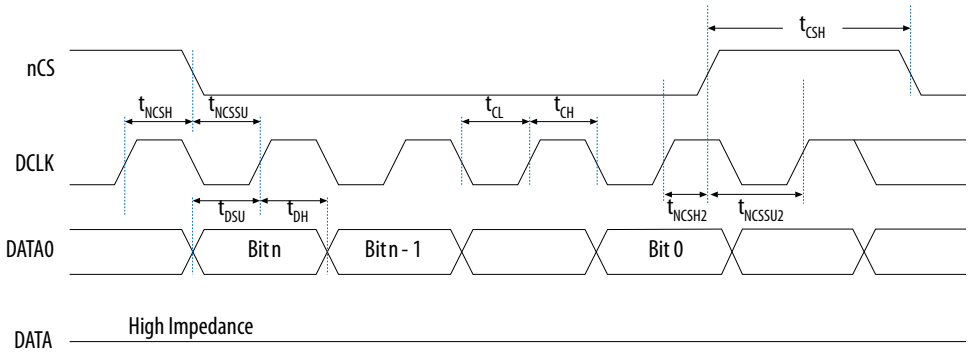



**Table 33. Write Operation Parameters**

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| $f_{WCLK}$ | 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, and erase sector operations for all devices except EPCQ512/A | — | — | 108 | MHz |
| | 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, and erase sector operations for EPCQ512/A | — | — | 133 | MHz |
| $t_{CH}$ (19) | DCLK high time for all devices except EPCQ512/A | 4 | — | — | ns |

continued...

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| | DCLK high time for EPCQ512/A | 3.375 | — | — | ns |
| $t_{CL}$ (19) | DCLK low time for all devices except EPCQ512/A | 4 | — | — | ns |
| | DCLK low time for EPCQ512/A | 3.375 | — | — | ns |
| $t_{NCSSU}$ | Chip select (nCS) setup time for all devices except EPCQ512/A | 4 | — | — | ns |
| | DCLK low time for EPCQ512/A | 3.375 | — | — | ns |
| $t_{NCSH}$ | Chip select (nCS) hold time for all devices except EPCQ512/A | 4 | — | — | ns |
| | Chip select (nCS) hold time for EPCQ512/A | 3.375 | — | — | ns |
| $t_{DSU}$ | DATA[] in setup time before the rising edge on DCLK for all devices except EPCQ512/A | 2 | — | — | ns |
| | DATA[] in setup time before the rising edge on DCLK for EPCQ512/A | 1.75 | — | — | ns |
| $t_{DH}$ | DATA[] hold time after the rising edge on DCLK for all devices except EPCQ512/A | 3 | — | — | ns |
| | DATA[] hold time after the rising edge on DCLK for EPCQ512/A | 2.5 | — | — | ns |
| $t_{CSH}$ | Chip select (nCS) high time | 50 | — | — | ns |
| $t_{WB}$ (20) | Write bytes cycle time | — | 0.6 | 5 | ms |
| $t_{WS}$ (20) | Write status cycle time | — | 1.3 | 8 | ms |
| $t_{EB}$ (20) | Erase bulk cycle time for EPCQ16 | — | 30 | 60 | s |
| | Erase bulk cycle time for EPCQ32 | | 30 | 60 | |
| | Erase bulk cycle time for EPCQ64 | | 60 | 250 | |
| | Erase bulk cycle time for EPCQ128 | | 170 | 250 | |
| | Erase bulk cycle time for EPCQ256 | | 240 | 480 | |
| | Erase bulk cycle time for EPCQ512/A | | 153 | 460 | |
| $t_{ES}$ (20) | Erase sector cycle time for all devices except EPCQ512/A | — | 0.7 | 3 | s |
| | Erase sector cycle time for EPCQ512/A | | 0.15 | 1 | |
| $t_{ESS}$ (20) | Erase subsector cycle time for all devices except EPCQ512/A | — | 0.3 | 1.5 | s |
| | Erase subsector cycle time for EPCQ512/A | | 0.05 | 0.4 | |

(19) The value must be larger /than or equal to 1/f $W_{CLK}$.

(20) The Write Operation Timing Diagram does not show these parameters.

### 1.9.2. Read Operation Timing

**Figure 23. Read Operation Timing Diagram**

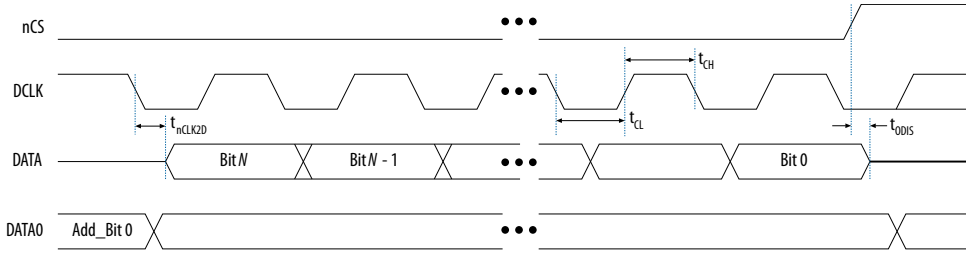



**Table 34. Read Operation Parameters**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $f_{RCLK}$ | 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 |
| $t_{CH}$ | DCLK high time | 4 | — | ns |
| $t_{CL}$ | DCLK low time | 4 | — | ns |
| $t_{ODIS}$ | Output disable time after read | — | 8 | ns |
| $t_{nCLK2D}$ | Clock falling edge to DATA | — | 7 | ns |

## 1.10. Programming and Configuration File Support

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

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

During the ISP of an EPCQ device using the Intel FPGA Download CableIntel FPGA Download Cable II, Intel FPGA Ethernet 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. The download cable then uses the interface pins depending on the selected AS mode to program the EPCQ device. When programming is complete, the download cable releases the interface pins of the EPCQ device and the nCE pin of the FPGA and pulses the nCONFIG signal to start the configuration process.

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

**Related Information**

- Using the Intel FPGA Serial Flash Loader IP Core with the Quartus Prime Software
- Intel FPGA ASMI Parallel IP Core User Guide
- Intel FPGA Download Cable II User Guide
- Intel FPGA Ethernet Cable
- Configuration, Design Security, and Remote System Upgrades in Arria V Devices
- Configuration, Design Security, and Remote System Upgrades in Cyclone V Devices
- Configuration, Design Security, and Remote System Upgrades in Stratix V Devices

## 1.11. Pin Information

The following lists the control pins on the EPCQ device:

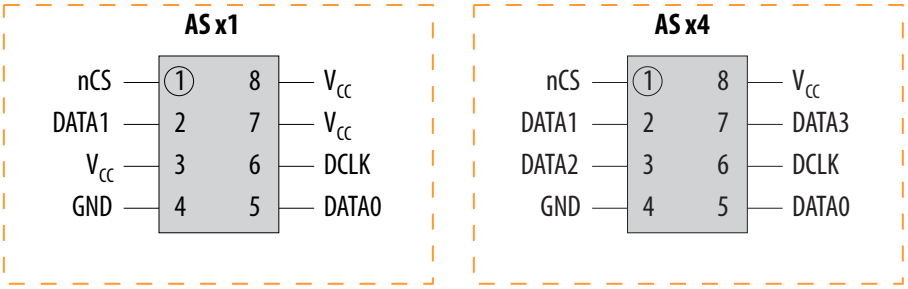
- Serial data 3 (DATA3)
- Serial data 2 (DATA2)
- Serial data 1 (DATA1)
- Serial data 0 (DATA0)
- Serial clock (DCLK)
- Chip select (nCS)

**Related Information**

- Configuration, Design Security, and Remote System Upgrades in Arria V Devices
- Configuration, Design Security, and Remote System Upgrades in Cyclone V Devices
- Configuration, Design Security, and Remote System Upgrades in Stratix V Devices

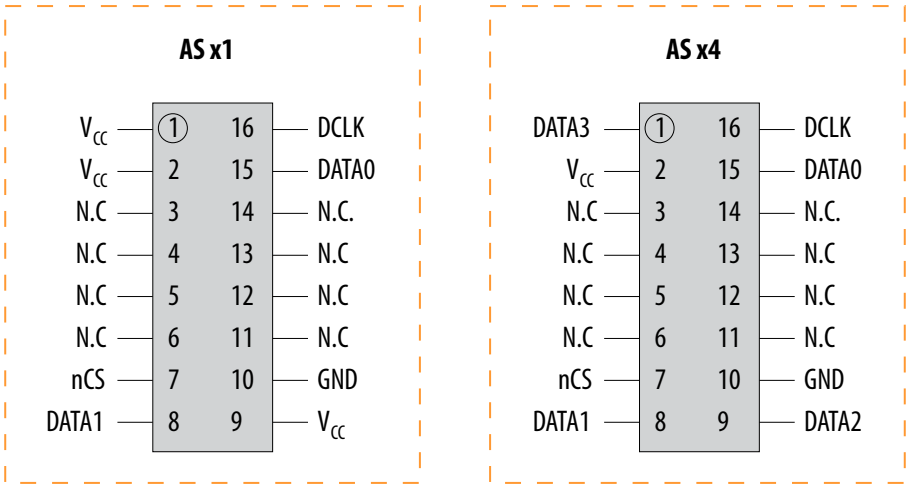
### 1.11.1. Pin-Out Diagram for EPCQ16 and EPCQ32 Devices

**Figure 24. AS x1 and AS x4 Pin-Out Diagrams for EPCQ16 and EPCQ32 Devices**



## 1.11.2. Pin-Out Diagram for EPCQ64, EPCQ128, EPCQ256, and EPCQ512/A Devices

**Figure 25. AS x1 and AS x4 Pin-Out Diagrams for EPCQ64, EPCQ128, EPCQ256, and EPCQ512/A Devices**




*Notes:*

*N.C pins must be left unconnected.*

## 1.11.3. EPCQ Device Pin Description

**Table 35. EPCQ Device Pin Description**

| Pin Name | AS x1 Pin-Out Diagram | | AS x4 Pin-Out Diagram | | Pin Type | Description |
|---|---|---|---|---|---|---|
| | Pin Number in 8-Pin SOIC Package | Pin Number in 16-Pin SOIC Package | Pin Number in 8-Pin SOIC Package | Pin Number in 16-Pin SOIC Package | | |
| DATA0 | 5 | 15 | 5 | 15 | I/O | For AS x1 mode, use this pin as an input signal pin to write or program the EPCQ device. During write or program operations, the data is latched on the rising edge of the DCLK signal.<br>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 device. The data is latched on the rising edge of the DCLK signal. During read or configuration operations, this pin acts as an output signal pin that serially transfers data out of the EPCQ device to the FPGA. The data is shifted out on the falling edge of the DCLK signal.<br>During the extended quad input fast write bytes or extended dual input fast write bytes operations, this pin acts as an input pin that serially transfers data into the EPCQ device. The data is latched on the rising edge of the DCLK signal. During extended dual input fast read or extended quad input fast read operations, this pin acts as an output signal pin |

*continued...*

| Pin Name | AS x1 Pin-Out Diagram | | AS x4 Pin-Out Diagram | | Pin Type | Description |
|---|---|---|---|---|---|---|
| | Pin Number in 8-Pin SOIC Package | Pin Number in 16-Pin SOIC Package | Pin Number in 8-Pin SOIC Package | Pin Number in 16-Pin SOIC Package | | |
| | | | | | | that serially transfers data out of the EPCQ device to the FPGA. The data is shifted out on the falling edge of the DCLK signal. |
| DATA1 | 2 | 8 | 2 | 8 | I/O | For AS x1 and x4 modes, use this pin as an output signal pin that serially transfers data out of the EPCQ device to the FPGA during read or configuration operations. The transition of the signal is on the falling edge of the DCLK signal.<br>During the extended dual input fast write bytes or extended quad input fast write bytes operation, this pin acts as an input signal pin that serially transfers data into the EPCQ device. The data is latched on the rising edge of the DCLK signal.<br>During extended dual input fast read or extended quad input fast read operations, this pin acts as an output signal pin that serially transfer data out of the EPCQ device to the FPGA. The data is shifted out on the falling edge of the DCLK signal. During read, configuration, or program operations, you can enable the EPCQ device by pulling the nCS signal low. |
| DATA2 | — | — | 3 | 9 | I/O | For AS x1 mode, extended dual input fast write bytes operation and extended dual input fast read operation, this pin must connect to a 3.3-V power supply.<br>For AS x4 mode, use this pin as an output signal that serially transfers data out of the EPCQ device to the FPGA during read or configuration operations. The transition of the signal is on the falling edge of the DCLK signal.<br>During the extended quad input fast write bytes operation, this pin acts as an input pin that serially transfers data into the EPCQ device. The data is latched on the rising edge of the DCLK 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 device to the FPGA. The data is shifted out on the falling edge of the DCLK signal. |
| DATA3 | — | — | 7 | 1 | I/O | For AS x1 mode, extended dual input fast write bytes operation and extended dual input fast read operation, this pin must connect to a 3.3-V power supply.<br>For AS x4 mode, use this pin as an output signal that serially transfers data out of the EPCQ device to the FPGA during read or configuration operations. The transition of the signal is on the falling edge of the DCLK signal.<br>During the extended quad input fast write bytes operation, this pin acts as an input pin that serially transfers data into the EPCQ device. The data is latched on the rising edge of the DCLK 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 device to the FPGA. The data is shifted out on the falling edge of the DCLK signal. |
| nCS | 1 | 7 | 1 | 7 | Input | 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. When this signal is low, the device is enabled and is in active mode. After power up, the EPCQ device requires a falling edge on the nCS signal before you begin any operation. |

***continued...***

| Pin Name | AS x1 Pin-Out Diagram | | AS x4 Pin-Out Diagram | | Pin Type | Description |
|---|---|---|---|---|---|---|
| | Pin Number in 8-Pin SOIC Package | Pin Number in 16-Pin SOIC Package | Pin Number in 8-Pin SOIC Package | Pin Number in 16-Pin SOIC Package | | |
| DCLK | 6 | 16 | 6 | 16 | Input | 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 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. |
| $V_{CC}$ | 8 | 2 | 8 | 2 | Power | Connect the power pins to a 3.3-V power supply. |
| GND | 4 | 10 | 4 | 10 | Ground | Ground pin. |

## 1.12. Device Package and Ordering Code

**Related Information**

Packaging Specifications and Dimensions

### 1.12.1. Package

The EPCQ16 and EPCQ32 devices are available in 8-pin SOIC packages. The EPCQ64, EPCQ128, EPCQ256, and EPCQ512/A devices are available in 16-pin SOIC packages.

For a 16-pin SOIC package, you can migrate vertically from EPCQ64 device to EPCQ128, EPCQ256, or EPCQ512/A device. You can also migrate EPCQ128 device to EPCQ256 or EPCQ512/A device, and EPCQ256 device to EPCQ512/A device.

### 1.12.2. Ordering Code

**Table 36. EPCQ Device Ordering Codes**

| Device | Ordering Code[21] |
|---|---|
| EPCQ16 | EPCQ16SI8N |
| EPCQ32 | EPCQ32SI8N |
| EPCQ64 | EPCQ64SI16N |
| EPCQ128 | EPCQ128SI16N |
| EPCQ256 | EPCQ256SI16N |
| EPCQ512/A | EPCQ512ASI16N |

(21) N indicates that the device is lead free.

# 1.13. Document Revision History for Quad-Serial Configuration (EPCQ) Devices Datasheet

| Document Version | Changes |
|---|---|
| 2020.01.23 | Updated the *Read Device Identification Operation Timing Diagram*. |
| 2018.06.01 | • Updated the description for $t_{DSU}$ parameter.<br>• Added data retention feature information.<br>• Added information about the read flag status register operation.<br>• Added a note to *Absolute Maximum Ratings* to state the maximum undershoot and overshoot for $V_I$.<br>• Updated the product obsolescence note in *Supported Intel EPCQ Devices* table.<br>• Added a link to PDN1802. |

| Date | Version | Changes |
|---|---|---|
| November 2017 | 2017.11.06 | • Updated operation code from binary to hexadecimal value in *Summary of Operation Codes* table.<br>• Added operation code in hex value in each operation timing diagrams.<br>• Updated operation timing diagrams to improve operation code and DLCK alignment accuracy.<br>• Added *Registers* section and included read status, write status, read non-volatile configuration, and read non-volatile configuration registers.<br>• Added note stating EPCQ16, EPCQ32, EPCQ64, and EPCQ128 are scheduled for product obsolescence.<br>• Added *Extended Quad Input Fast Write Bytes Operation Timing Diagram for EPCQ512/A* figure.<br>• Added extended quad input fast write bytes operation code for EPCQ512/A in *Summary of Operation Codes* table.<br>• Removed ambient temperature in *Absolute Maximum Ratings*.<br>• Added note to EPCQ512/A device is shown as EPCQ512 inIntel Quartus Prime software.<br>• Updated *Write Operation Timing* for EPCQ512/A devices.<br>• Updated *Recommended Operating Conditions* and *ICC Supply Current* for EPCQ512/A devices.<br>• Updated capacitance is sample-tested at a 54 MHz frequency. |
| May 2016 | 2016.05.30 | • Removed instances of 'Preliminary' and notes about pending characterization.<br>• Replaced EPCQ512 instances with EPCQ512/A.<br>• Updated ordering code for EPCQ512/A.<br>• Updated *AS x1 and AS x4 Pin-Out Diagrams for EPCQ64, EPCQ128, EPCQ256, and EPCQ512/A Devices* figure.<br>• Updated Power dissipation and Joint temperature in the *Absolute Maximum Ratings* table. |
| January 2015 | 2015.01.23 | • Updated non-volatile configuration register operation code.<br>• Added erase subsector operation.<br>• Added read non-volatile configuration register operation.<br>• Updated AS x1 dummy clock cycles for non-volatile configuration registers.<br>• Updated the erase and program cycle to up to 100,000 cycles. |

***continued...***

| Date | Version | Changes |
|---|---|---|
| | | • Updated write non-volatile configuration register 16-bit register value.<br>• Added Non-volatile Configuration Register Operation Bit Definition table.<br>• Added read status operation timing diagram.<br>• Updated EPCQ Device Pin Description table.<br>• Added a link to the Packaging Specifications and Dimensions page. |
| January 2014 | 2014.01.10 | • Added EPCQ512 device support.<br>• Added the write non-volatile configuration register operation.<br>• Added a link to the ALTASMI_PARALLEL IP Core User Guide.<br>• Removed preliminary for EPCQ16, EPCQ32, EPCQ64, EPCQ128, and EPCQ256 devices.<br>• Updated block protection bits for EPCQ16, EPCQ32, EPCQ64, EPCQ128, and EPCQ256 devices.<br>• Updated the dummy byte term to dummy cycle.<br>• Updated the dummy cycles for the read device identification operation in the Operation Codes for EPCQ Devices.<br>• Updated the $t_{CL}$ and $t_{CH}$ parameters in the write operation timing diagram.<br>• Updated the Package section.<br>• Updated the erase bulk cycle time for EPCQ16 and EPCQ32 devices.<br>• Updated the operating temperature in the Recommended Operating Conditions. |
| July 2012 | 3.0 | • Added Table 3, Table 4, and Table 5 to include the address range for EPCQ16, EPCQ32, and EPCQ64 devices.<br>• Added Table 9, Table 10, Table 11, Table 12, Table 13, and Table 14 to include the block protection bits for EPCQ16, EPCQ32, and EPCQ64 devices.<br>• Added Figure 5, Figure 20 and Figure 21 to include EPCQ16 and EPCQ32 devices.<br>• Updated the "Device Package and Ordering Code" section.<br>• Updated Table 1, Table 2, Table 19, Table 20, Table 27, and Table 28 to include EPCQ16, EPCQ32, and EPCQ64 devices.<br>• Updated the address bytes for the extended quad input fast write bytes operation in Table 8.<br>• Updated Figure 22 and Figure 23 to include EPCQ64 devices. |
| January 2012 | 2.0 | • Added Figure 2.<br>• Updated "Read Bytes Operation" and "Fast Read Operation" sections.<br>• Updated Figure 1, Figure 3, Figure 4, Figure 7, and Figure 13.<br>• Updated Table 5, Table 11, Table 12, and Table 14.<br>• Minor text edits. |
| June 2011 | 1.0 | Initial release. |

Общество с ограниченной ответственностью «МосЧип» ИНН 7719860671 / КПП 771901001
Адрес: 105318, г.Москва, ул.Щербаковская д.3, офис 1107

**Данный компонент на территории Российской Федерации**

**Вы можете приобрести в компании MosChip.**

Для оперативного оформления запроса Вам необходимо перейти по данной ссылке:

http://moschip.ru/get-element

Вы можете разместить у нас заказ для любого Вашего проекта, будь то серийное производство или разработка единичного прибора.

В нашем ассортименте представлены ведущие мировые производители активных и пассивных электронных компонентов.

Нашей специализацией является поставка электронной компонентной базы двойного назначения, продукции таких производителей как XILINX, Intel (ex.ALTERA), Vicor, Microchip, Texas Instruments, Analog Devices, Mini-Circuits, Amphenol, Glenair.

Сотрудничество с глобальными дистрибьюторами электронных компонентов, предоставляет возможность заказывать и получать с международных складов практически любой перечень компонентов в оптимальные для Вас сроки.

На всех этапах разработки и производства наши партнеры могут получить квалифицированную поддержку опытных инженеров.

Система менеджмента качества компании отвечает требованиям в соответствии с ГОСТ Р ИСО 9001, ГОСТ РВ 0015-002 и ЭС РД 009

**Офис по работе с юридическими лицами:**

105318, г.Москва, ул.Щербаковская д.3, офис 1107, 1118, ДЦ «Щербаковский»

Телефон: +7 495 668-12-70 (многоканальный)

Факс: +7 495 668-12-70 (доб.304)

E-mail: info@moschip.ru

Skype отдела продаж:
moschip.ru
moschip.ru_4
moschip.ru_6
moschip.ru_9