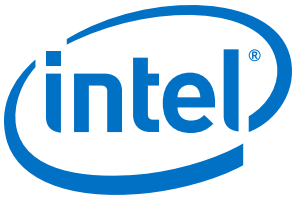


AN 822: Intel® FPGA Configuration Device Migration Guideline

Subscribe
Send Feedback

AN-822 | 2020.04.29
Latest document on the web: PDF | HTML

Contents

1. Intel® FPGA Configuration Device Migration Guideline

This document describes the guidelines for migrating from the Serial Configuration (EPCS) and Quad-Serial Configuration (EPCQ) devices to the Quad-Serial Configuration (EPCQ-A) devices.

Related Information

- Serial Configuration (EPCS) Devices Datasheet
- Quad-Serial Configuration (EPCQ) Devices Datasheet
- Quad-Serial Configuration (EPCQ-A) Devices Datasheet

1.1. Migration Considerations

The EPCQ-A devices are conditionally compatible for a direct migration from EPCQ and EPCS devices.

You must consider the following items to determine the compatibility and the next step of action for a successful device migration.

IP Cores

If you are using Intel® IP cores, you may need to regenerate and recompile your design. In certain conditions, the programming files can be reused without recompilation. Refer to IP Core Compatibility on page 5 for more information about IP core compatibility. Refer to Table 3 on page 6 if you are not using IP cores that interface with the configuration device.

Pins, Package and Capacity

Migration can only be done to an EPCQ-A device that has sufficient capacity for the programming file and have the same pin count package.

Pin 3 (nRESET) on the EPCQ64A and EPCQ128A devices act as a reset pin. This pin has an internal pull-up, and if you do not use the reset function, connect the nRESET pin to either V_{CC} or leave it unconnected. Refer to Pin Information on page 16 for more information about the pin-outs and descriptions.

Figure 1. EPCS to EPCQ Migration Pin Package and Capacity Summary

		To EPCQ-A				
	Variant	EPCQ4A[1]	EPCQ16A	EPCQ32A	EPCQ64A	EPCQ128A
From EPCS	EPCS1	Yes	Yes	Yes	No	No
	EPCS4	Yes	Yes	Yes	No	No
	EPCS16	Yes[2]	Yes	Yes	No	No
	EPCS64	No	No	No	Yes	Yes
	EPCS128	No	No	No	Yes[2]	Yes
From EPCQ	EPCQ16	No	Yes	Yes	No	No
	EPCQ32	No	Yes[2]	Yes	No	No
	EPCQ64	No	No	No	Yes	Yes
	EPCQ128	No	No	No	Yes[2]	Yes

Note:
1. EPCQ4A devices support Active Serial x1 configuration only.
2. Migration is compatible only if the destination EPCQ-A device has the sufficient capacity for the programming file.

Operation Commands

The dummy clock requirement of the fast read (0Bh) and extended quad input fast read (EBh) commands:

- EPCQ—the dummy clock is configurable with the non-volatile configuration register (NVCR). When the EPCQ is used with a Cyclone® V, Arria® V or Stratix® V device, the dummy clock is configured to be 4, 10 or 12, depending on the byte-addressing mode and ASx1 or ASx4 configuration. However, in EPCQ-A devices, the dummy clock is fixed at 8 and 6 for fast read and extended quad input fast read respectively. Therefore you must regenerate the programming files, such as `.pof`, `.jic`, and `.rpd`.
- EPCS—the dummy clock is fixed at 8 for fast read, therefore you do not have to regenerate the programming files if all other conditions are met. Table 3 on page 6 defines the need to regenerate the programming files. Refer to IP Core and Programming File Migration Guideline on page 7 for more information about the conditions.

Status Register

Status Register contains the Top/Bottom (TB) bit (bit 5), Block Protect (BP) bits (bit 4, bit 3, bit 2) for sector protection bits. EPCS devices do not have TP bit and some EPCQ device densities have BP3 (bit 6), while bit 6 is reserved in EPCQ-A devices. Due to this differences, you may need to recompile the programming file if your design uses the sector protect feature. Refer to Status Register on page 21 for more information about status registers and sector protect bits.

Sector Size

All of the EPCS, EPCQ and EPCQ-A devices have the sector size of 512kb except for EPCS128 which has 2Mb. This impacts the sector erase operation. If the design is erasing the flash during user mode, you must update your design to comply the sector size when migrating from EPCS128 to EPCQ128A. After updating your design, regenerate a new programming file for the EPCQ-A device.

1.2. Software Migration Guidelines

When you use a legacy device that is not supported in the Intel Quartus® Prime software version 17.1 and later and you need to modify your design to migrate the configuration device to EPCQ-A, you need to use 13.1.4 patch 4.70r. For more information, refer to the readme patch instructions and the patches for Windows and Linux in the *Software Download* page of the *FPGA Configuration Devices Support* website.

Related Information

FPGA Configuration Devices Support Software Download

1.2.1. IP Core Compatibility

Table 1. EPCS to EPCQ-A Device Migration IP Core Compatibility

IP Core	Compatibility	Condition
ASMI Parallel	Yes/No	• If sector protect is used, refer to Sector Protect on page 21 to determine compatibility. • EPCS128 has different sector size than EPCQ128A, not compatible if sector erase is used.
Serial Flash Controller	Yes/No	
Serial Flash Loader (SFL)	Yes/No	• Compatible for Cyclone V, Arria V and Stratix V devices. • For devices earlier than Cyclone V, Arria V and Stratix V, it is compatible if the **Enhanced SFL**(1) is enabled.
Remote Update	Yes	

Table 2. EPCQ to EPCQ-A Device Migration IP Core Compatibility

IP Core	Compatibility	Condition
ASMI Parallel	Yes/No	• If sector protect is used, refer to sector protect table comparison to determine compatibility. • Not compatible if read dummy clock is enabled.
ASMI Parallel II	No	—
Serial Flash Controller	No	—
Serial Flash Controller II	No	—
Generic QSPI Controller	No	—

continued...

(1) Enhanced SFL is an option available in the Serial Flash Loader IP core when using with devices earlier than Cyclone V, Arria V and Stratix V.

IP Core	Compatibility	Condition
Generic QSPI Controller II	No	—
Serial Flash Loader	Yes/No	• Compatible for Cyclone V, Arria V and Stratix V devices. • For devices earlier than Cyclone V, Arria V and Stratix V, it is compatible if the **Enhanced SFL**(1) is enabled.
Remote Update	Yes	—

Related Information

- Altera Remote Update IP Core User Guide
- Altera ASMI Parallel IP Core User Guide
- Converting .sof to .jic Files in the Quartus Prime Software
- Programming Serial Configuration Devices Using the Quartus Prime Programmer and .jic Files

1.2.2. Programming File Compatibility

Note: This section describes programming file compatibility for designs without Intel FPGA IP cores.

List of supported programming files:

- Programmer Object File (`.pof`)
- JTAG Indirect Configuration File (`.jic`)
- Raw Programming Data (`.rpd`)
- STAPL File (`.jam`/`.jbc`)
- Serial Vector Format (`.svf`)

Note: Compression and encryption would not affect the programming file compatibility.

Table 3. Programming File Compatibility Guide

Note: For designs that do not contain IP cores which interface with the configuration device, depending upon the FPGA family and configuration scheme implemented, the existing programming files may be compatible with EPCQ-A devices without the need to regenerate the programming files.

Device Family	Original Configuration Device	Configuration Device Density	Programming Files supported	Disable EPCS/ EPCQ ID check Setting	Compatible with EPCQ-A(2)(3)
Legacy FPGA devices	EPCS	64Mb & below	`.pof`/`.jic`/`.rpd`/`.jam`/`.jbc`	Any	Yes
			`.svf`	Any	Yes(4)

continued...

(2) Table assumes other compatibility considerations are satisfied.

(3) Table assumes the programming files do not contain any ASMI Parallel IP or Serial Flash Loader IP.

Device Family	Original Configuration Device	Configuration Device Density	Programming Files supported	Disable EPCS/ EPCQ ID check Setting	Compatible with EPCQ-A[2][3]
		128Mb	`.pof/.jic/.rpd`	Any	Yes[5]
			`.svf`	Any	Yes[4]
			`.jam/.jbc`	Any	No[6]
	EPCQ	Any	`.pof/.jic/.rpd/.jam/.jbc`	On[7]	Yes
			`.svf`	On[7]	Yes[4]
Cyclone V, Arria V, and Stratix V devices	EPCS	64Mb & below	`.pof/.jic/.rpd/.jam/.jbc`	On[8]	Yes
			`.svf`	On[8]	Yes[4]
		128Mb	`.pof/.jic/.rpd`	On[8]	Yes[5]
			`.svf`	On[8]	Yes[4]
			`.jam/.jbc`	Any	No[6]
	EPCQ[9]	Any	Any	Any	No

Refer to IP Core and Programming File Migration Guideline on page 7 for more information about guidelines on incompatible programming files.

1.2.3. IP Core and Programming File Migration Guideline

Note: This section describes programming file compatibility for designs with Intel FPGA IP cores that interface with the configuration device.

(2) Table assumes other compatibility considerations are satisfied.

(3) Table assumes the programming files do not contain any ASMI Parallel IP or Serial Flash Loader IP.

(4) Only supported for `.svf` files generated for EPCS devices to be used to program an EPCQ-A, and not the other way round.

(5) In `.rpd` file, the binary data is the same between EPCS128 and EPCQ128A. However due to different sector size, a proper erasing procedure is required when programming each device.

(6) Due to different sector size, the `.jam/.jbc` file is different between EPCS and EPCQ.

(7) In Intel Quartus Prime version 15.1 or later, automatic mode turns on this option automatically.

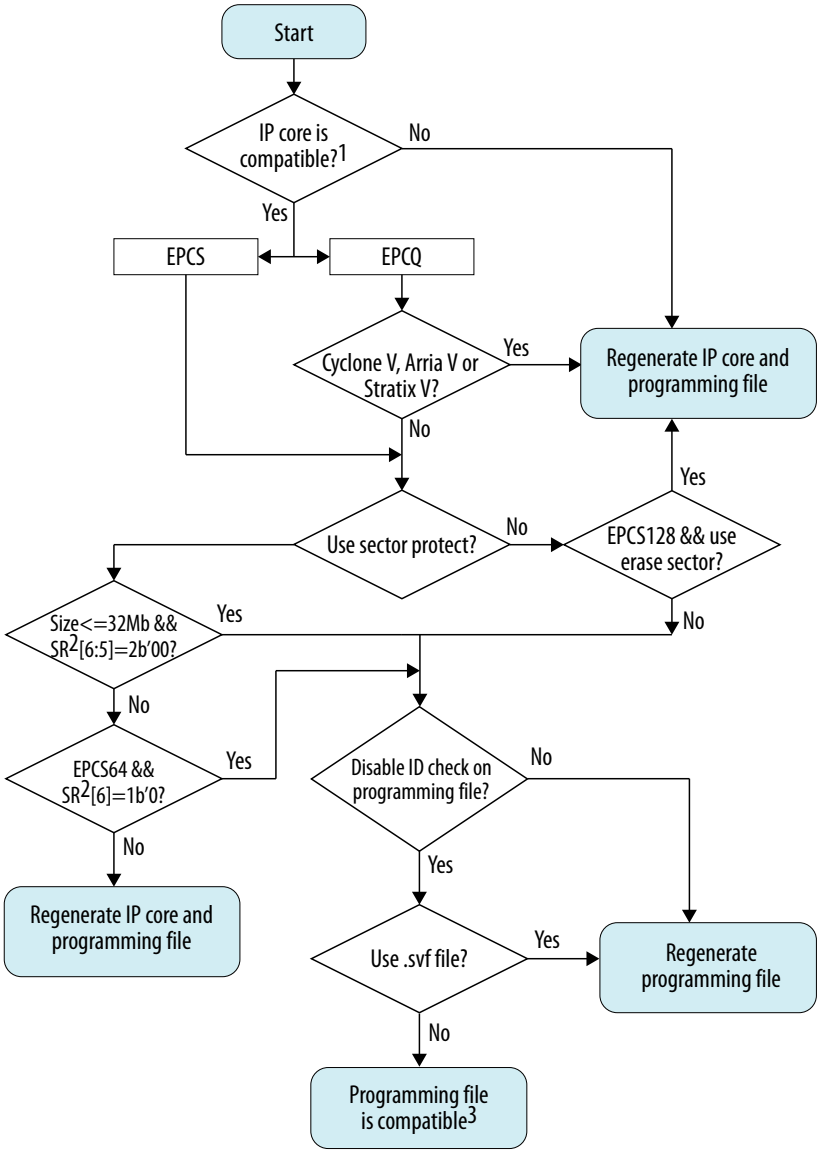
(8) Other than Intel Quartus Prime version 13.0 to 15.0, automatic mode turns on this option automatically.

(9) EPCQ programming files are not compatible with EPCQ-A in AS x1 or AS x4 modes.

Refer to the following diagram to determine the subsequent tasks and guidelines for migration:

- IP core and programming file are incompatible—regenerate IP core and programming file shown in IP Core Regeneration Guideline on page 9.
- Programming file is incompatible—regenerate programming file shown in Programming File Regeneration Guideline on page 10.
- IP core and programming file are compatible—no additional task required and you can reuse the existing programming file.

Figure 2. IP Core and Programming File Compatibility Flow Chart



Notes:
1. Check IP core compatibility in *IP Core Compatibility* section.
2. SR is status register.
3. IP core and programming file can be migrated without regeneration.

1.2.3.1. IP Core Regeneration Guideline

To regenerate the IP core and programming file with the correct settings, perform the following steps:

1. Regenerate the desired IP cores.
   a. If you use the sector protect feature:

- For EPCQ4A, EPCQ16A, and EPCQ32A—ensure the status register bit [6:5] is set to 0.
- For EPCQ64A—ensure the status register bit [6] is set to 0.

b. For EPCS128 to EPCQ128A migration—sector erase must comply to the sector size of EPCQ128A.

Note: Sector size for EPCS128 and EPCQ128A are 2Mb and 512kb respectively.

2. Recompile the configuration bitstream to obtain the `.sof` file.

1.2.3.2. Programming File Regeneration Guideline

To regenerate the programming file with the correct settings, perform the following steps:

1. Convert the `.sof` file to the desired programming file in the Intel Quartus Prime **Convert Programing File** tool and ensure that you:
   - Select the correct EPCQ-A device you are migrating to.
   - Enable the **Disable EPCS/EPCQ ID check** option. This option is available in the **Advanced Option** settings of the Convert Programming File tool in Intel Quartus Prime software.
   - For migration from EPCS devices—Select AS x1 configuration mode.
2. Program the programming file into EPCQ-A device using Intel Quartus Prime Programmer.

1.2.4. Software Support for EPCQ-A Devices

Table 4. Intel Quartus Prime Software Support for EPCQ-A Devices

For more information about using the EPCQ-A device with the ASMI Parallel II Intel FPGA IP Core in the Intel Quartus Prime Standard Edition software before version 17.1, refer to the *How can I use EPCQ-A Serial Configuration Devices in ASMI Parallel II Intel FPGA IP when using Intel Quartus Prime Standard Edition software version 17.0.2 and earlier?*

Intel Quartus Prime	IP Cores	Programmer and Programming File Generation
Intel Quartus Prime Pro Edition 17.1	No	Yes
Intel Quartus Prime Standard Edition 17.1	Yes	Yes

Related Information

How can I use EPCQ-A Serial Configuration Devices in ASMI Parallel II Intel® FPGA IP when using Intel® Quartus® Prime Standard Edition software version 17.0.2 and earlier?

1.3. Specification Comparison

The following tables show the side-by-side comparison of the EPCS, EPCQ and EPCQ-A operating conditions. For more detailed and up-to-date information, refer to the respective device datasheet.

1.3.1. Operating Conditions

Table 5. EPCS, EPCQ and EPCQ-A Devices Operating Conditions

Parameter	Condition	Symbol	Min			Max			Unit
			EPCS	EPCQ	EPCQ-A	EPCS	EPCQ	EPCQ-A	
Supply voltage	The maximum V_{CC} rise time is 100 ms.	V_{CC}	2.7			3.6			V
Operating temperature	For industrial use	T_A	-40			85			°C
High-level input voltage	—	V_{IH}	0.7 x V_{CC}(10) (11)			V_{CC} + 0.4			V
Low-level input voltage	—	V_{IL}	-0.5			0.3 x V_{CC}			V
High-level output voltage	I_{OH} =-100µA	V_{OH}	V_{CC} - 0.2			—			V
Low-level output voltage	I_{OL} =100µA	V_{OL}	—			0.2 or 0.4(12)			V

1.3.2. Timing Specifications

The following tables show the general comparison of the EPCS, EPCQ, and EPCQ-A operation timing. For more detailed and up-to-date information, refer to the respective device datasheet.

Caution: You need to take note of these values to avoid from the migration to fail:

- t_{DH}
- t_{DSU}
- t_{nCLK2D}/t_{CLQV}
- t_{CLQX}(13)

(10) The FPGA 2.5V I/O V_{OH} level is insufficient to achieve EPCQ or EPCQ-A V_{IH} threshold across entire voltage range.

(11) The minimum V_{OH} for 3V or 3.3V LVTTL is 2.4V in the Intel FPGA device. This specification is based on the worst condition. Since the input current of EPCQ and EPCQ-A is small enough, the V_{OH} of Intel FPGA device does not violate the minimum V_{IH} of EPCQ and EPCQ-A in the usual usage. Intel recommends that you perform simulation using the IBIS model to ensure the required specifications are achieved.

(12) 0.2 V for EPCQ16A, EPCQ32A, EPCQ64A, and EPCQ128A. 0.4 V is for EPCQ4A

(13) Refer to Evaluating Data Setup and Hold Timing Slack on page 24 evaluate the data setup and hold timing slack.

1.3.2.1. Read Operation Timing

Table 6. EPCS and EPCQ-A Devices Read Operation Timing Parameters

Symbol	Parameter	Capacity	Min		Max		Unit
			EPCS	EPCQ-A	EPCS	EPCQ-A	
f_{RCLK}	Read clock frequency	All	—	—	20	50	MHz
	Fast read clock frequency	All	—	—	40	100	MHz
t_{CH}	DCLK high time	4 Mb	11	4 or 6(14)	—	—	ns
		All others	11	3.4 or 9(15)	—	—	
t_{CL}	DCLK low time	4Mb	11	4 or 6(14)	—	—	ns
		All others	11	3.4 or 9(15)	—	—	
t_{ODIS}	Output disable time after read	All	—	—	8	7	ns
t_{nCLK2D} / t_{CLQV}(16)	Clock falling edge to DATA	4Mb	—	—	8	8	ns
		All others	—	—	8	6	

Table 7. EPCQ and EPCQ-A Devices Read Operation Timing Parameters

Symbol	Parameter	Capacity	Min		Max		Unit
			EPCQ	EPCQ-A	EPCQ	EPCQ-A	
f_{RCLK}	Read clock frequency	All	—	—	50	50	MHz
	Fast read clock frequency	All	—	—	100	100	MHz
t_{CH}	DCLK high time	All	4	3.4 or 9(15)	—	—	ns
t_{CL}	DCLK low time	All	4	3.4 or 9(15)	—	—	ns
t_{ODIS}	Output disable time after read	All	—	—	8	7	ns
t_{nCLK2D} / t_{CLQV}(16)	Clock falling edge to DATA	All	—	—	7	6	ns

1.3.2.2. Write Operation Timing

Table 8. EPCS and EPCQ-A Devices Write Operation Timing Parameters

Symbol	Operation	Capacity	Min		Typical		Max		Unit
			EPCS	EPCQ-A	EPCS	EPCQ-A	EPCS	EPCQ-A	
f_{WCLK}	Write clock frequency	All	—		—		25	100	MHz
t_{CH}	DCLK high	4	20	4	—		—		ns

continued...

(14) 4 ns is for normal read and 6 ns is for fast read.

(15) 3.4 ns is for normal read and 9 ns is for fast read.

(16) t_{nCLK2D} is used in EPCS and EPCQ devices while t_{CLQV} is used in EPCQ-A devices.

Symbol	Operation	Capacity	Min		Typical		Max		Unit
			EPCS	EPCQ-A	EPCS	EPCQ-A	EPCS	EPCQ-A	
		All others	20	3.4	—		—		
t_{CL}	DCLK low	4	20	4	—		—		ns
		All others	20	4	—		—		
t_{NCSSU}	Chip select (nCS) setup	All	10	5	—		—		ns
t_{NCSH}	Chip select (nCS) hold	All	10	5	—		—		ns
t_{DSU}	DATA[ ] in setup before DCLK rising edge	All	5	2	—		—		ns
t_{DH}	DATA[ ] hold time after DCLK rising edge	4	5	5	—		—		ns
		All others	5	3	—		—		
t_{CSH}	Chip select (nCS) high	4	100	100	—		—		ns
		All others	100	10 / 50(17)	—		—		
t_{WB}	Write bytes cycle	1	—		1.5	—	5	—	ms
		4	—		1.5	0.4	5	0.8	ms
		16	—		1.5	0.4	5	3	ms
		32	—		—	0.7	—	3	ms
		64	—		1.5	0.8	5	3	ms
		128	—		2.5	0.7	7	3	ms
t_{WS}	Write status cycle	All	—		5	10	15	15	ms
t_{EB}	Erase bulk cycle	1	—		3	—	6	—	s
		4	—		5	1	10	4	s
		16	—		17	5	40	25	s
		32	—		—	10	—	50	s
		64	—		68	20	160	100	s
		128	—		105	40	250	200	s
t_{ES}	Erase sector cycle	4	—		2	0.15	3	1	s
		All others	—		2	0.15	3	2	s

Table 9. EPCQ and EPCQ-A Devices Write Operation Timing Parameters

Symbol	Operation	Capacity	Min		Typical		Max		Unit
			EPCQ	EPCQ-A	EPCQ	EPCQ-A	EPCQ	EPCQ-A	
f_{WCLK}	Write clock frequency	All	—		—		100	100	MHz
t_{CH}	DCLK high	All	4	3.4	—		—		ns
t_{CL}	DCLK low	All	4	4	—		—		ns

continued...

(17) 10ns for read and 50 ns for erase program and write.

Symbol	Operation	Capacity	Min		Typical		Max		Unit
			EPCQ	EPCQ-A	EPCQ	EPCQ-A	EPCQ	EPCQ-A	
t_{NCSSU}	Chip select (nCS) setup	All	4	5	—		—		ns
t_{NCSH}	Chip select (nCS) hold	All	4	5	—		—		ns
t_{DSU}	DATA[] in setup before DCLK rising edge	All	2	2	—		—		ns
t_{DH}	DATA[] hold time after DCLK rising edge	All	3	3	—		—		ns
t_{CSH}	Chip select (nCS) high	All	50	10 / 50(18)	—		—		ns
t_{WB}	Write bytes cycle	16	—		0.6	0.4	5	3	ms
		32	—		0.6	0.7	5	3	ms
		64	—		0.6	0.8	5	3	ms
		128	—		0.6	0.7	5	3	ms
t_{WS}	Write status cycle	All	—		1.3	10	8	15	ms
t_{EB}	Erase bulk cycle	16	—		30	5	60	25	s
		32	—		30	10	60	50	s
		64	—		60	20	250	100	s
		128	—		170	40	250	200	s
t_{ES}	Erase sector cycle	All others	—		0.7	0.15	3	2	s
		128	—		0.7	0.15	6	2	s

1.3.3. Operation Codes

The following tables summarize EPCS, EPCQ, and EPCQ-A operation codes. For more detailed and up-to-date information, refer to the respective device datasheet.

Table 10. EPCS, EPCQ and EPCQ-A Devices Operation Codes Summary

Operation	Operation Code		
	EPCS	EPCQ	EPCQ-A
Write status	01h		
Write bytes	02h		
Read bytes	03h		
Write disable	04h		
Read status	05h		
Write enable	06h		
Fast read	0Bh		
Read silicon ID	ABh(19)	—	ABh

continued...

(18) 10ns for read and 50 ns for erase program and write.

Operation	Operation Code		
	EPCS	EPCQ	EPCQ-A
Read device ID	9Fh[20]	9Fh	9Fh
Erase bulk	C7h		
Erase sector	D8h		
Erase subsector	—	20h	20h
Extended dual input fast read	—	BBh	BBh
Extended quad input fast read	—	EBh	EBh
Extended dual input fast write bytes	—	D2h	—
Extended quad input fast write bytes	—	12h	—
Quad input fast write bytes	—	—	32h[21]
Read NVCR	—	B5h	—
Write NVCR	—	B1h	—
4BYTEADDREN	—	B7h	—
4BYTEADDEX	—	E9h	—

(19) The read silicon ID is available in EPCS1, EPCS4, EPCS16, and EPCS64 devices only.

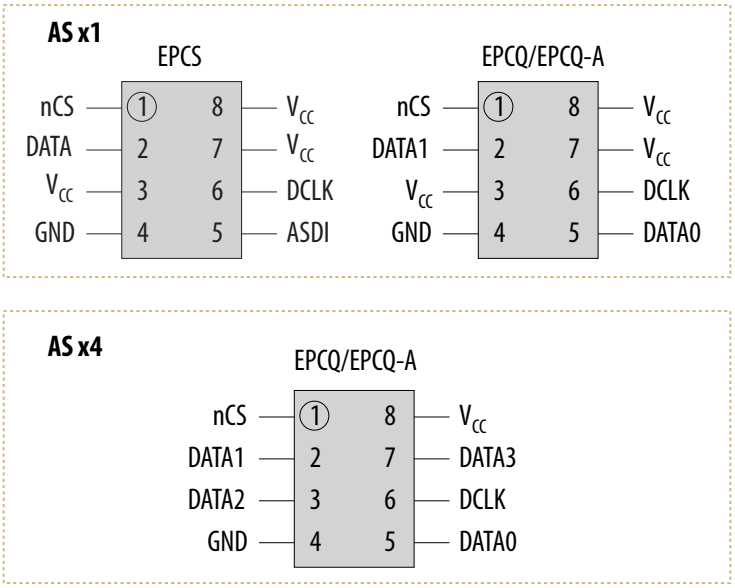
(20) The read device ID is available in EPCS128 devices only.

(21) Quad input fast write bytes operation is not supported in Intel IP cores.

1.3.4. Pin Information

1.3.4.1. 8-pin SOIC Device Pin Information

Figure 3. Pin-Out Diagram for 8-pin SOIC EPCS, EPCQ and EPCQ-A Devices



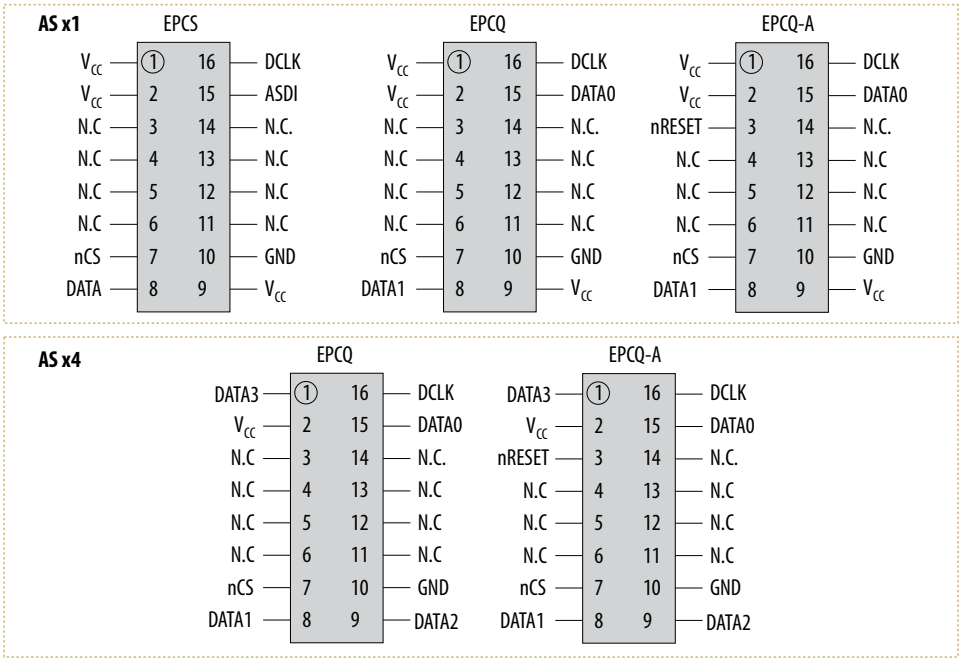
Leave all N.C pins unconnected.

Table 11. Pin Comparison for 8-pin SOIC EPCS, EPCQ and EPCQ-A Devices

Pin Number	AS x1			AS x4	
	EPCS	EPCQ	EPCQ-A	EPCQ	EPCQ-A
1	nCS			nCS	
2	DATA	DATA1		DATA1	
3	V_{CC}			DATA2	
4	GND			GND	
5	ASDI	DATA0		DATA0	
6	DCLK			DCLK	
7	V_{CC}			DATA3	
8	V_{CC}			V_{CC}	

1.3.4.2. 16-pin SOIC Device Pin Information

Figure 4. Pin-Out Diagram for 16-pin SOIC EPCS, EPCQ and EPCQ-A Devices



Notes: Leave all N.C pins unconnected.
There is an internal pull-up resistor for the dedicated nRESET pin. If the reset function is not needed, connect this pin to Vcc or leave it unconnected.

Table 12. Pin Comparison for 16-pin SOIC EPCS, EPCQ and EPCQ-A Devices

Pin Number	AS x1			AS x4	
	EPCS	EPCQ	EPCQ-A	EPCQ	EPCQ-A
1	V_{CC}			DATA3	
2	V_{CC}			V_{CC}	
3	Not connected		nRESET(22)	Not connected	nRESET(22)
4, 5, 6, 11, 12, 13 and 14	Not connected			Not connected	
7	nCS			nCS	
8	DATA	DATA1		DATA1	
9	V_{CC}			DATA2	
10	GND			GND	
15	ASDI	DATA0		DATA0	
16	DCLK			DCLK	

(22) There is an internal pull-up resistor for the dedicated nRESET pin. If the reset function is not needed, connect this pin to V_{CC} or leave it unconnected.

1.3.5. Package Dimensions

1.3.5.1. 8-Pin SOIC Device Package Dimensions

Figure 5. Package Dimension Diagram for 8-Pin SOIC Package Devices

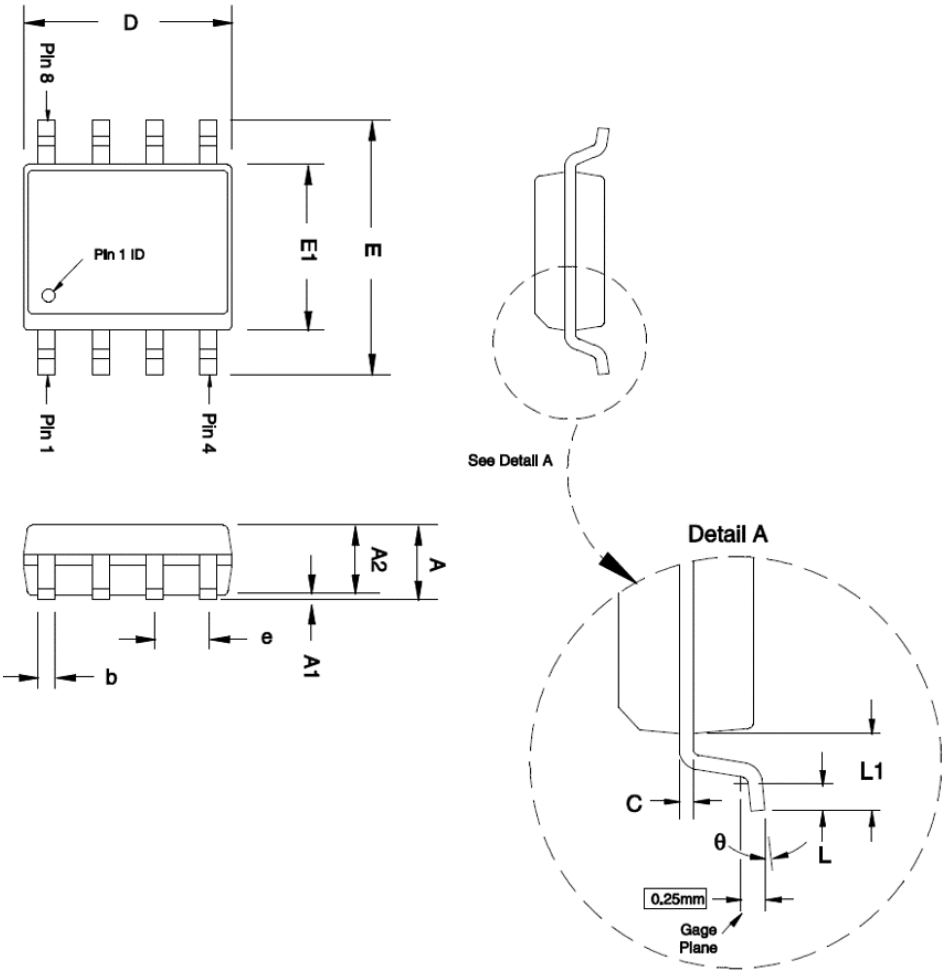


Table 13. Package Dimension Comparison for 8-Pin SOIC Package Devices

Symbol	Min (mm)		Typical (mm)		Max (mm)	
	EPCS/EPCQ	EPCQ-A	EPCS/EPCQ	EPCQ-A	EPCS/EPCQ	EPCQ-A
A	1.35	1.35	—	—	1.75	1.75
A1	0.1	0.1	—	—	0.25	0.25
A2	1.25	—	—	—	1.65	—
D	—	—	4.90 BSC	4.90 BSC	—	—
E	—	—	6.0 BSC	6.0 BSC	—	—
E1	—	—	3.90 BSC	3.90 BSC	—	—

continued...

Symbol	Min (mm)		Typical (mm)		Max (mm)	
	EPCS/EPCQ	EPCQ-A	EPCS/EPCQ	EPCQ-A	EPCS/EPCQ	EPCQ-A
L	0.4	0.4	—	—	1.27	1.27
L1	—	—	1.04 REF	—	—	—
b	0.31	0.33	—	—	0.51	0.51
c	0.17	0.19	—	—	0.25	0.25
e	—	—	1.27 BSC	1.27 BSC	—	—
Theta	0°	0°	—	—	8°	10°

1.3.5.2. 16-Pin SOIC Device Package Dimensions

Figure 6. Package Dimension Diagram for 16-Pin SOIC Package Devices

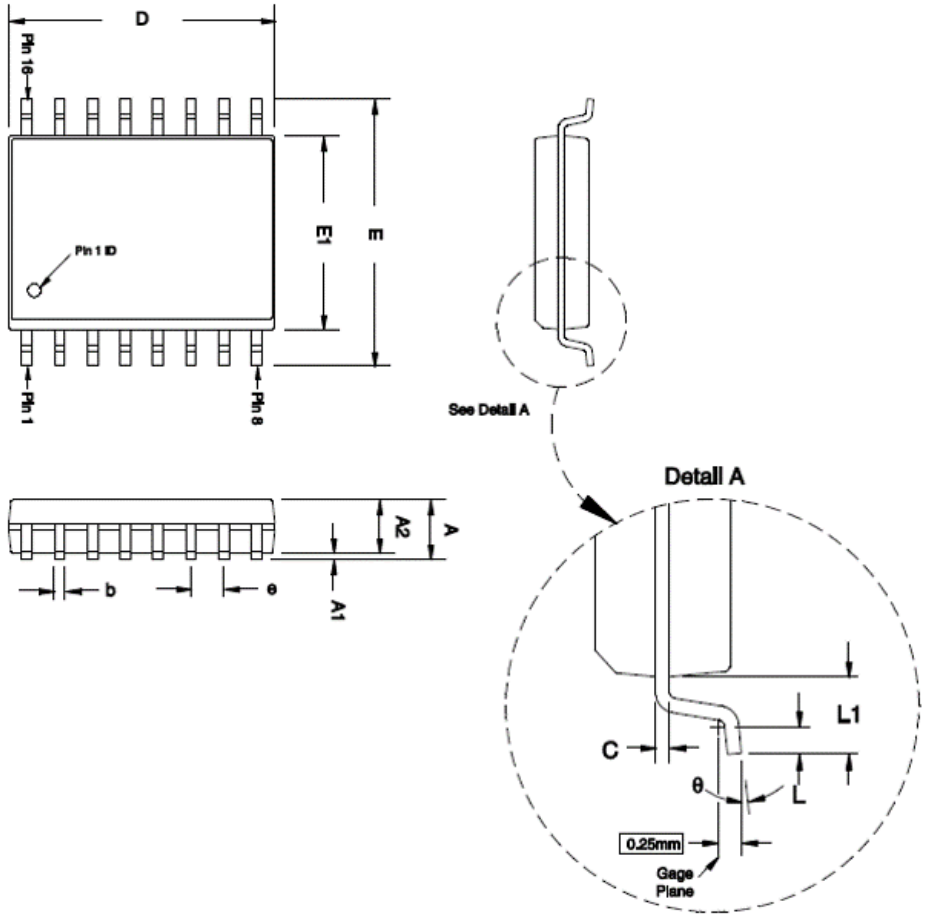


Table 14. Package Dimension Comparison for 16-Pin SOIC Package Devices

Symbol	Min (mm)		Typical (mm)		Max (mm)	
	EPCS/EPCQ	EPCQ-A	EPCS/EPCQ	EPCQ-A	EPCS/EPCQ	EPCQ-A
A	2.35	2.35	—	—	2.65	2.65
A1	0.1	0.1	—	—	0.3	0.3
A2	2.05	2.05	—	—	2.55	2.55
D	—	10.08	10.3 BSC	—	—	10.49
E	—	10.01	10.3 BSC	—	—	10.64
E1	—	7.39	7.50 BSC	—	—	7.59
L	0.4	0.38	—	—	1.27	1.27
L1	—	—	1.40 Ref	1.40 Ref	—	—
b	0.31	0.31	—	—	0.51	0.51

continued...

Symbol	Min (mm)		Typical (mm)		Max (mm)	
	EPCS/EPCQ	EPCQ-A	EPCS/EPCQ	EPCQ-A	EPCS/EPCQ	EPCQ-A
c	0.2	0.20	—	—	0.33	0.33
e	—	—	1.27 BSC	1.27 BSC	—	—
Theta	0°	0°	—	—	8°	8°

1.3.6. Status Register

Table 15. Status Register Bits for EPCS, EPCQ and EPCQ-A Devices

Bit	Name	Description	R/W	EPCS		EPCQ		EPCQ-A
				1	4/16/64/128	16/32	64/128	All
7	RSV	Reserved						
6	BP3[23]	Block Protect Bit 3	R/W	No	No	No	Yes	No
5	TB	Top/Bottom Bit	R/W	No	No	Yes	Yes	Yes
4	BP2	Block Protect Bit 2	R/W	No	Yes	Yes	Yes	Yes
3	BP1	Block Protect Bit 1	R/W	Yes	Yes	Yes	Yes	Yes
2	BP0	Block Protect Bit 0	R/W	Yes	Yes	Yes	Yes	Yes
1	WEL	Write Enable Latch	R	Yes	Yes	Yes	Yes	Yes
0	WIP	Write In Progress	R	Yes	Yes	Yes	Yes	Yes

1.3.6.1. Sector Protect

Table 16. Sector Protect Comparison for EPCS4 and EPCQ4A Devices

Note: Set TB bit to 0 for backward compatibility.

Status Register				EPCS4	EPCQ4A
TB	BP2	BP1	BP0	Protected Sectors (8 sectors)	
x	0	0	0	None	None
0	0	0	1	7	7
0	0	1	0	6-7	6-7
0	0	1	1	4-7	4-7
1	0	0	1	N/A	0
1	0	1	0	N/A	0-1
1	0	1	1	N/A	0-3
x	1	x	x	All	All

(23) This is a reserved bit in EPCQ-A device and must be set to 0.

Table 17. Sector Protect Comparison for EPCS16, EPCQ16 and EPCQ16A Devices

Note: Set TB bit to 0 for backward compatibility for migration from EPCS16.

Status Register				EPCS16	EPCQ16	EPCQ16A
TB	BP2	BP1	BP0	Protected Sectors (32 sectors)		
0	0	0	0	None	None	None
0	0	0	1	31	31	31
0	0	1	0	30-31	30-31	30-31
0	0	1	1	28-31	28-31	28-31
0	1	0	0	24-31	24-31	24-31
0	1	0	1	16-31	16-31	16-31
0	1	1	0	All	All	All
0	1	1	1	All	All	All
1	0	0	0	N/A	None	None
1	0	0	1	N/A	0	0
1	0	1	0	N/A	0-1	0-1
1	0	1	1	N/A	0-3	0-3
1	1	0	0	N/A	0-7	0-7
1	1	0	1	N/A	0-15	0-15
1	1	1	0	N/A	All	All
1	1	1	1	N/A	All	All

Table 18. Sector Protect Comparison for EPCQ32 and EPCQ32A Devices

Status Register				EPCQ32	EPCQ32A
TB	BP2	BP1	BP0	Protected Sectors (64 sectors)	
0	0	0	0	None	None
0	0	0	1	63	63
0	0	1	0	62-63	62-63
0	0	1	1	60-63	60-63
0	1	0	0	56-63	56-63
0	1	0	1	48-63	48-63
0	1	1	0	32-63	32-63
0	1	1	1	All	All
1	0	0	0	None	None
1	0	0	1	0	0
1	0	1	0	0-1	0-1
1	0	1	1	0-3	0-3
1	1	0	0	0-7	0-7

continued...

Status Register				EPCQ32	EPCQ32A
TB	BP2	BP1	BP0	Protected Sectors (64 sectors)	
1	1	0	1	0-15	0-15
1	1	1	0	0-31	0-31
1	1	1	1	All	All

Table 19. Sector Protect Comparison for EPCS64, EPCQ64 and EPCQ64A Devices

Note: Set TB bit to 0 for backward compatibility for migration from EPCS64.

Status Register				EPCS64	EPCQ64	EPCQ64A
TB	BP2	BP1	BP0	Protected Sectors (128 sectors)		
0	0	0	0	None	None	None
0	0	0	1	126-127	127	126-127
0	0	1	0	124-127	126-127	124-127
0	0	1	1	120-127	124-127	120-127
0	1	0	0	112-127	120-127	112-127
0	1	0	1	96-127	112-127	96-127
0	1	1	0	64-127	96-127	64-127
0	1	1	1	All	64-127	All
1	0	0	0	N/A	None	None
1	0	0	1	N/A	0	0-1
1	0	1	0	N/A	0-1	0-3
1	0	1	1	N/A	0-3	0-7
1	1	0	0	N/A	0-7	0-15
1	1	0	1	N/A	0-15	0-31
1	1	1	0	N/A	0-31	0-63
1	1	1	1	N/A	0-63	All

Table 20. Sector Protect Comparison for EPCS128, EPCQ128 and EPCQ128A Devices

Status Register				EPCS128[24]	EPCQ128[25]	EPCQ128A[25]
TB	BP2	BP1	BP0	Protected Sectors (64 sectors)		
0	0	0	0	None	None	None
0	0	0	1	63	255	252-255
0	0	1	0	62-63	254-255	248-255
0	0	1	1	60-63	252-255	240-255

continued...

(24) 262144 byte (2MB) per sector

(25) 65536 byte (512KB) per sector

Status Register				EPCS128[24]	EPCQ128[25]	EPCQ128A[25]
TB	BP2	BP1	BP0	Protected Sectors (64 sectors)		
0	1	0	0	56-63	248-255	224-255
0	1	0	1	48-63	240-255	192-255
0	1	1	0	32-63	224-255	128-255
0	1	1	1	All	192-255	All
1	0	0	0	N/A	None	None
1	0	0	1	N/A	0	0-3
1	0	1	0	N/A	0-1	0-7
1	0	1	1	N/A	0-3	0-15
1	1	0	0	N/A	0-7	0-31
1	1	0	1	N/A	0-15	0-63
1	1	1	0	N/A	0-31	0-127
1	1	1	1	N/A	0-63	All

1.4. Evaluating Data Setup and Hold Timing Slack

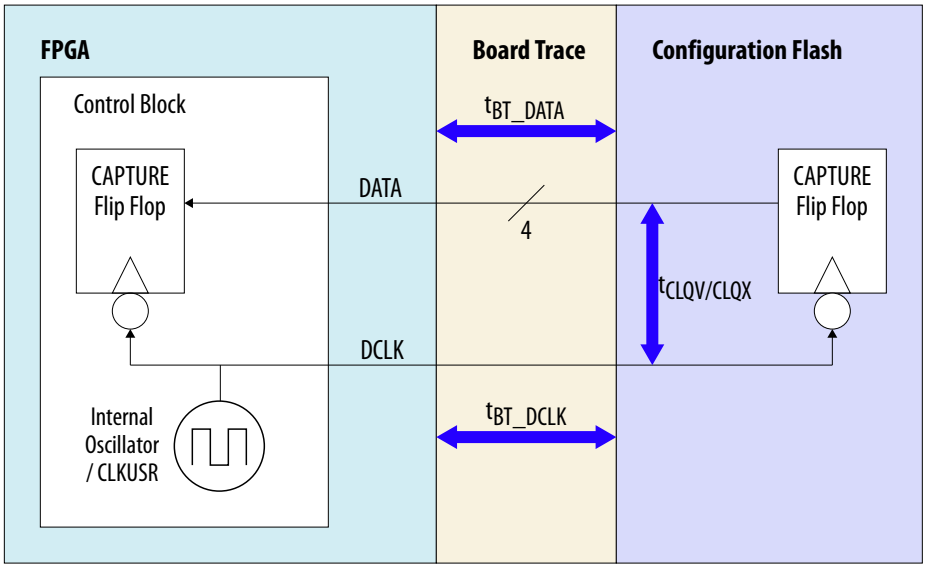
In AS configuration scheme, the FPGA will initiate the configuration process after POR. During the configuration process, the FPGA issues flash operation commands such as read device ID, normal read and erase bulk. You must ensure that the FPGA is able to read the data correctly from the configuration devices. This is done by ensuring the setup time, t_{DSU} and hold time, t_{DH} meets the requirements explained in the respective FPGA device datasheets. To evaluate the t_{DSU} and t_{DH} in your system, follow the guideline below.

(24) 262144 byte (2MB) per sector

(25) 65536 byte (512KB) per sector

Figure 7. FPGA to EPCQ-A Board Trace Block Diagram

Other type of configuration device or flash is supported by the FPGA.



The data setup timing slack must be equal or larger than the minimum data setup time, t_{DSU}

$t_{DCLK} - (t_{BT\_DCLK} + t_{CLQV} + t_{BT\_DATA}) \geq t_{DSU}$

The hold timing slack must be equal or larger than the minimum data hold time, t_{DH}:

$t_{BT\_DCLK} + t_{CLQX} + t_{BT\_DATA} \geq t_{DH}$

- t_{DCLK} = Period for a DCLK cycle
- t_{BT_DCLK} = Board trace propagation delay for DCLK from FPGA to EPCQ-A
- t_{CLQV} = Clock low to output valid
- t_{CLQX} = Output hold time
- t_{BT_DATA} = Board trace propagation delay for Data from EPCQ-A to FPGA
- t_{DSU} = Minimum data setup time required by FPGA
- t_{DH} = Minimum data hold time required by FPGA

Related Information

Quad-Serial Configuration (EPCQ-A) Devices Datasheet

1.5. Migration Method from EPCQ to EPCQ-A for Arria V, Cyclone V, and Stratix V Devices

Note: If you are using a Cyclone V device and you intend to migrate to a faster configuration device or flash device, Intel recommends you to use the Cyclone V QS device. The Cyclone V QS device has an improved specification with lower t_{DH} requirement compared to a non-Cyclone V QS device. For more details about the AS timing specifications for the Cyclone V QS device, refer to the *Cyclone V Device Datasheet*. To improve the AS configuration timing to meet t_{DSU} and t_{DH} of the Cyclone V QS device or non-Cyclone V QS device, refer to the board design guidelines.

1.5.1. Board Design Guidelines for the Active Serial (AS) Configuration Scheme

The board design guidelines provide recommendation on how to add extra delay on DCLK or DATA signals via trace length, resistor-capacitor (RC) network, or buffer to meet the FPGA hold time (t_{DH}) and setup time (t_{DSU}) specification for DCLK running at 50 MHz or lower and 100 MHz. The additional delay helps to improve the hold timing slack of the AS configuration scheme.

The following are the board design guidelines for DCLK running at 50 MHz or lower use case and 100 MHz use case. The recommendations are applicable for migrating from an existing EPCQ device or a third-party serial flash device to the EPCQ-A device or a faster third-party serial flash device.

Table 21. Configuration Device Migration Recommendations

Supporting DCLK at 50-MHz or Lower Operation	Supporting DCLK at 100-MHz Operation
Recommendation 1—Add an extra RC network on the DCLK line near flash end.	You must design the board trace for the DCLK and DATA with the following recommendation to meet the timing requirement for the FPGA and the EPCQ-A device. You can use the data setup or hold timing equations to identify the range for the minimum and maximum delays. The range is essential to be used as a reference to ensure the total propagation delay for the DCLK and DATA signals must be designed within the range.
Recommendation 2—Add an extra buffer on the DCLK line between the FPGA and the serial flash device.	

The equations to analyze the setup or hold time can be used to calculate the extra delay by using the RC network or buffer on the DCLK signal is needed to ensure it meets the FPGA t_{DH} and t_{DSU} specifications. The following example shows you how to calculate the extra delay.

Example 1. Calculating the Extra Delay

- Assumption for the system setup:
  - DCLK frequency: 50MHz (Period 20ns)
  - Cyclone V AS timing specifications:
    - Minimum t_{DSU}: 1.5ns
    - Minimum t_{DH}: 2.9ns
  - Intel EPCQ64ASI16N flash specifications:
    - t_{CLQV}: 6ns
    - t_{CLQX}: 1.5ns
- To identify the allowed maximum extra delay:

```
tDCLK - (tBT_DCLK + tCLQV + tBT_DATAmax) ≥ tDSU
= (tBT_DCLK + tBT_DATAmax) ≤ tDCLK - tCLQV - tDSU
= (tBT_DCLK + tBT_DATAmax) ≤ 20ns - 6ns - 1.5ns
= (tBT_DCLK + tBT_DATAmax) ≤ 12.5ns
```

  The allowed maximum extra delay is 12.5ns.
- To identify the required minimum extra delay:

```
tBT_DCLK + tCLQX + tBT_DATAmin ≥ tDH
= tBT_DCLK + tBT_DATAmin ≥ tDH - tCLQX
= tBT_DCLK + tBT_DATAmin ≥ 2.9ns - 1.5ns
= tBT_DCLK + tBT_DATAmin ≥ 1.4ns
```

  The required minimum delay is 1.4ns.

As a conclusion, the extra delay added must fall within the calculated range to meet the required FPGA data hold time (t_{DH}) while not violating the data setup time (t_{DSU}) specification.

1.5.1.1. IBIS Simulation Example

In this board design guidelines, the following IBIS simulation was performed to provide recommendation on the topology to estimate the minimum and maximum delays that can be achieved on the DCLK and DATA signals.

The FPGA[26] and EPCQ-A IBIS models are used in the following IBIS simulation example by adding extra RC network or buffer on the DCLK signal. The simulation result is used to ensure the delay added on the DCLK is sufficient to meet the actual FPGA t_{DH} and t_{DSU} specifications and ensure the signal quality is good.

(26) The Cyclone V FPGA is used in the following IBIS simulation example.

Example 2. IBIS Simulation Setup for Arria V, Cyclone V, and Stratix V Devices

- Cyclone V LVTTL-3.0V I/O model is used in the IBIS simulation example shown below
  - DCLK: lvttl30_ctnio_d12s1
  - DATA: lvttl30_ctnio_d8s1
- Arria V LVTTL-3.0V I/O model
  - DCLK: lvttl30_ctio_d12s1
  - DATA: lvttl30_ctio_d8s1
- Stratix V LVTTL-3.3V I/O model
  - DCLK: lvttl_ctnio_d12s1
  - DATA: lvttl_ctnio_d8s1

Table 22. RLC Value Used for the DCLK and DATA Pins

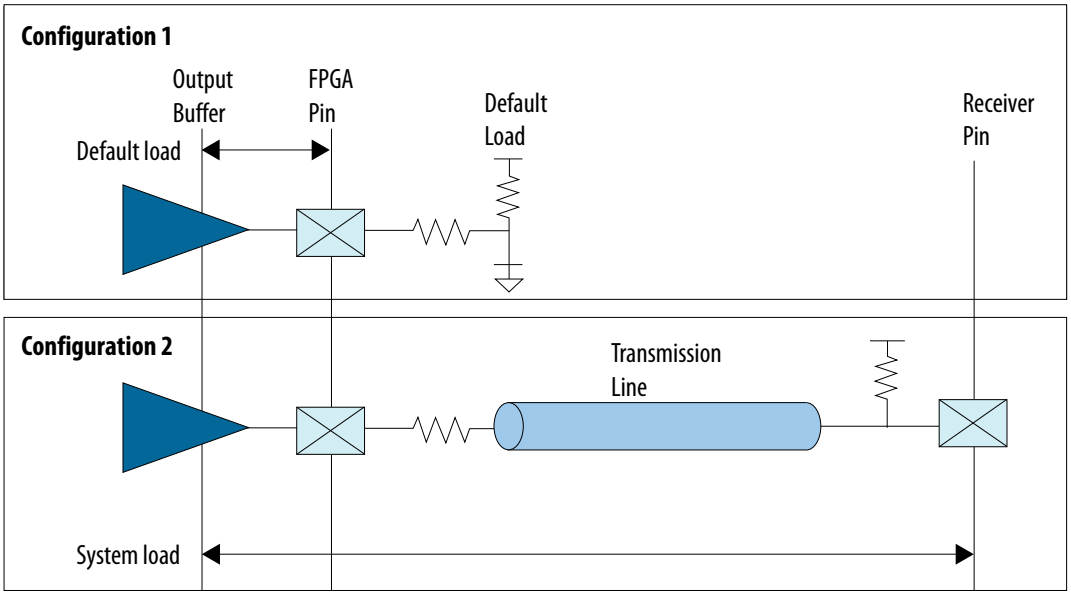
The RLC value could be varied depending on the package size.

Pin Name	R (mΩ)	L (nH)	C (pF)
DCLK	252.71	2.26	1.07
DATA	633.45	5.61	1.74

- Intel EPCQ64ASI16N model is used
- V_{CC} at 3.3V ±5% is used for both the FPGA and flash devices
- One inch trace with 50Ω trace impedance on the typical FR4 board is used
- Simulation is done on both the Slow and Fast IC corners

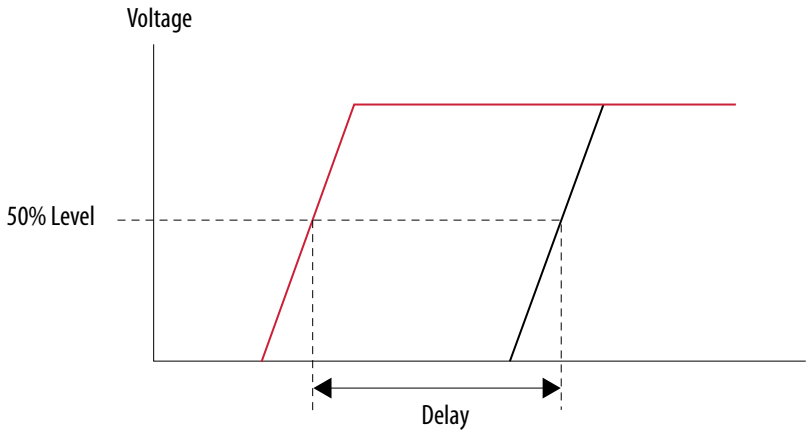
In addition, Hyperlynx IBIS simulation is used to predict the effects of a given link. The delay of a given link can be estimated accurately using the technique of comparing the simulated delay between the default setup (Configuration 1) and the actual system setup (Configuration 2) as shown in the figure below. The actual use of this methodology to measure the AS configuration timing are performed in the following section.

Figure 8. Simulated Delay Comparison



Note: When measuring the delay of a signal at 50% of the voltage level as shown in the figure below, the IBIS simulation was performed by measuring the voltage at 50% of the voltage level. You can measure at the voltage input high (V_{IH}) or voltage input low (V_{IL}) level.

Figure 9. Voltage Level Delay Measurement



1.5.1.2. Example for Adding the RC Network on DCLK

1.5.1.2.1. Example for Simulating the DCLK Link

The following two simulation setups are used to measure the minimum or maximum delay on the DCLK signal.

Figure 10. Simulation Setup with the Default Load on the DCLK Signal

You must follow the recommended default load setup.

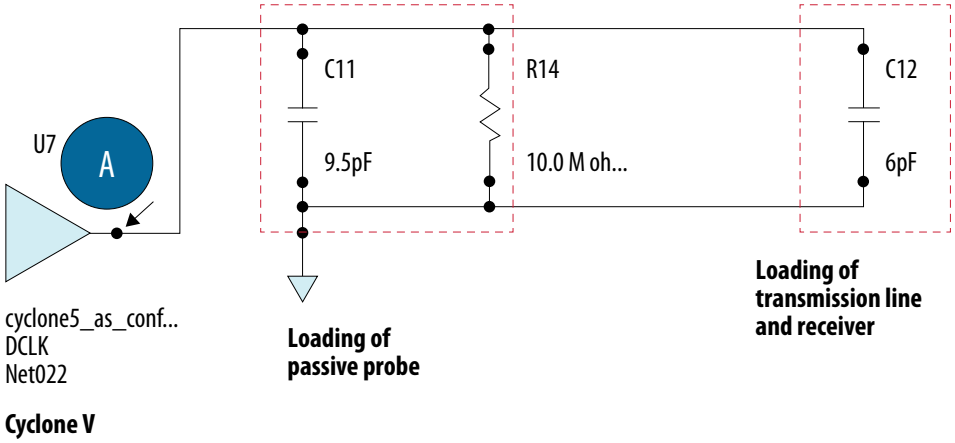
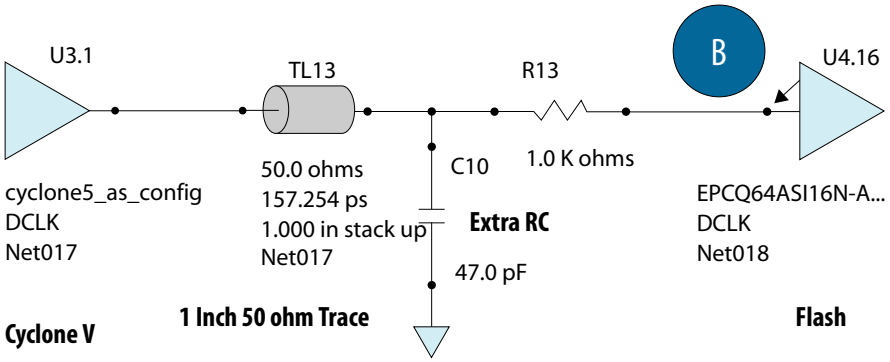


Figure 11. Simulation Setup with the Actual System Load including the RC Network Solution on the DCLK Signal

The RC network must be included.



By running simulation on both slow and fast corners, the minimum and maximum delays on the DCLK can be measured by comparing the delay measured between A and B.

Figure 12. Minimum DCLK Delay

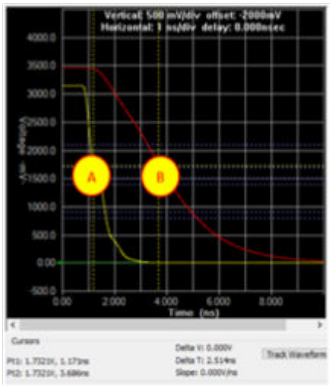
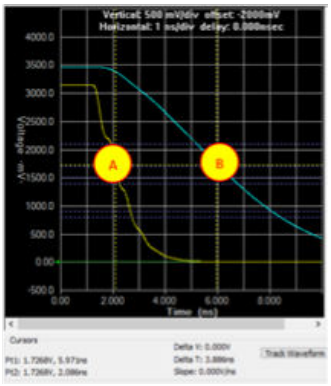


Figure 13. Maximum DCLK Delay



1.5.1.2.2. Example for Simulating the DATA Link

The following three simulation setups are essential to measure the minimum or maximum delay on the DATA signal.

Figure 14. Simulation Setup with the Default Maximum Load on the DATA Signal

- The setup is used to measure the maximum delay on the DATA signal.
- The maximum flash clock to output delay (t_{CLQV}) value in the *EPCQ-A Serial Configuration Device Datasheet* is based on 30pF loading.

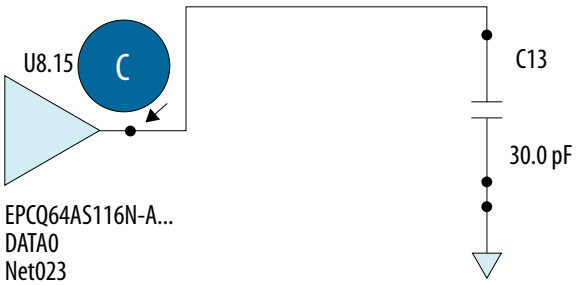


Figure 15. Simulation Setup with the Default Minimum Load on the DATA Signal

- The setup is used to measure the minimum delay on the DATA signal.
- The minimum flash clock to output delay (t_{CLQX}) value in the *EPCQ-A Serial Configuration Device Datasheet* is based on 0pF loading.

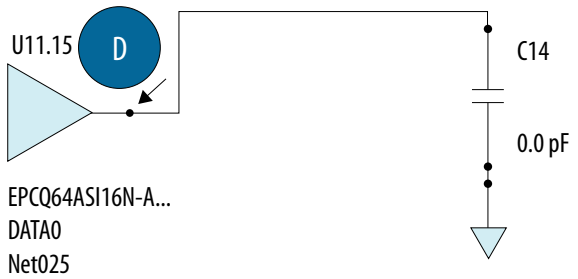


Figure 16. Simulation Setup with the Actual System Load on the DATA Signal

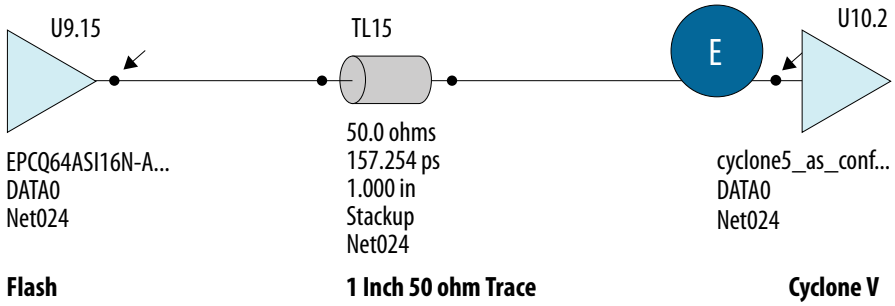
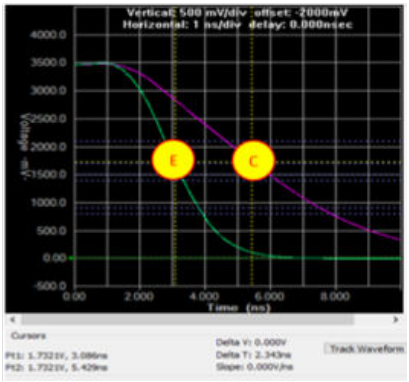


Figure 17. Maximum DATA Delay



The maximum DATA delay is calculated by substituting the delay measured between E and C in the simulation into the following equation.

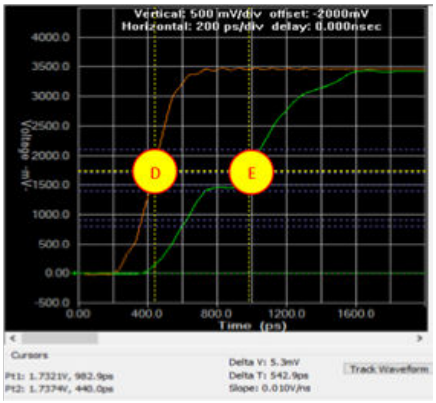
Maximum DATA delay = t_{CLQV} + (delay measurement E – delay measurement C)

Maximum DATA delay = 6ns – 2.343ns

Maximum DATA delay = 3.657ns

The results clearly show that the maximum DATA delay could be smaller depending on the actual system loading which helps to improve the FPGA setup time slack. Using the t_{CLQV} value defined in the *EPCQ-A Serial Configuration Device Datasheet* is too pessimistic in the setup time analysis.

Figure 18. Minimum DATA Delay



The minimum DATA delay is calculated by substituting the delay measured between E and D in the simulation into the following equation.

Minimum DATA delay = t_{CLQX} + (delay measurement E – delay measurement D)

Minimum DATA delay = 1.5ns + 0.542ns

Minimum DATA delay = 2.042ns

The results clearly show that the minimum DATA delay could be larger depending on the actual system loading. Using the t_{CLQX} value defined in the *EPCQ-A Serial Configuration Device Datasheet* is too pessimistic for the hold time analysis.

1.5.1.2.3. Simulation Results

The following table shows the minimum and maximum delays on the DCLK and DATA links can be obtained accurately via the IBIS simulation.

Table 23. Minimum and Maximum Delays on the DCLK and DATA Links

Signal	IC Corner	Total Delay Measured via IBIS Simulation at $V_{CC}/2$ (ns)[27]	Notes
DCLK	Fast/Strong	2.514	Minimum DCLK delay
DCLK	Slow/Weak	3.886	Maximum DCLK delay
DATA	Fast/Strong	2.042	Minimum DATA delay
DATA	Slow/Weak	3.657	Maximum DATA delay

Ultimately, the data setup time slack and the data hold time slack can be obtained by substituting the delay obtained from the IBIS simulation in the following equations.

(27) The data listed is based on the manual measurement from the simulation waveform.

Assumption for the system setup:

- DCLK frequency: 50MHz (period 20ns)
- Cyclone V AS timing specifications:
  - Minimum t_{DSU}: 1.5ns
  - Minimum t_{DH}: 2.9ns

This equation shows the data setup time slack calculation.

Data setup time slack = t_{DCLK} – t_{DSU} – (maximum DCLK delay + maximum DATA delay)

Data setup time slack = 20ns – 1.5ns – (3.886ns + 3.657ns)

Data setup time slack = 10.957ns

This equation shows the data hold time slack calculation.

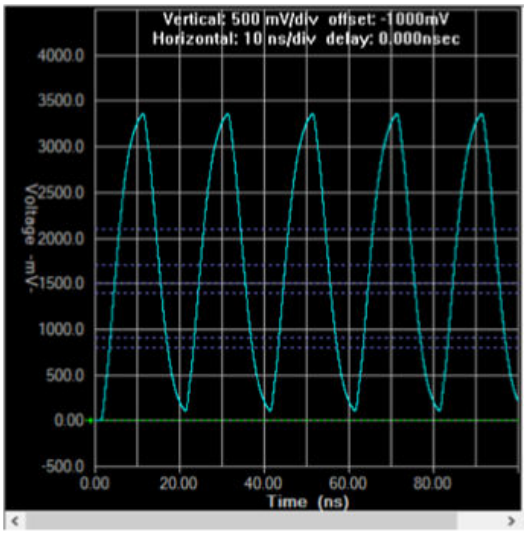
Data hold time slack = (Minimum DCLK delay + minimum DATA delay) – t_{DH}

Data hold time slack = (2.514ns + 2.042ns) – 2.9ns

Data hold time slack = 1.656ns

The DCLK signal quality is good with the extra RC network as shown in the simulation waveform below. The DCLK signal is simulated using 50 Mbps clock pattern.

Figure 19. Simulating the DCLK Signal at 50 Mbps Operation



By adding the recommended RC network, the DCLK and DATA trace lengths can be designed to as short as possible and are able to meet the minimum data setup or hold time required by the Cyclone V FPGA.

A smaller maximum DATA delay can be obtained via the simulation technique. The FPGA AS configuration data setup time slack can be improved with the simulation value compared to the flash datasheet specifications.

A larger minimum DATA delay can be obtained via the simulation technique. The FPGA AS configuration data hold time slack can be improved with simulation value compared to the flash datasheet specifications.

1.5.1.3. Example for Adding Buffer on the DCLK

The following two simulation setups are used to measure the minimum or maximum delay on the DCLK signal.

Figure 20. Simulation Setup with the Default Load on the DCLK Signal

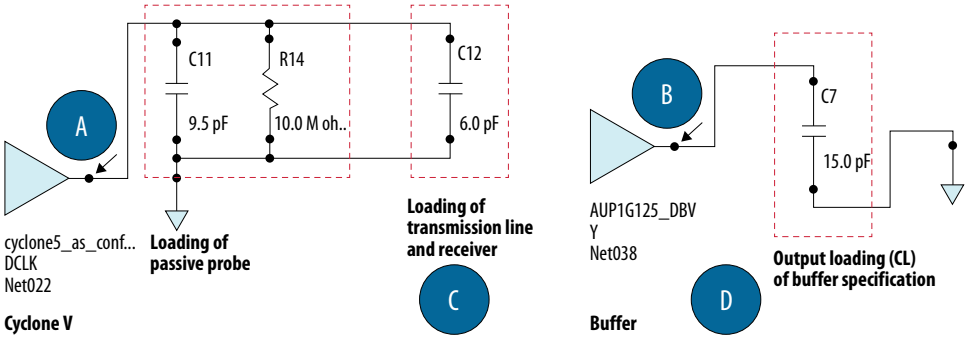


Figure 21. Simulation Setup with the Actual System Load on the DCLK Signal

Buffer model must be included.

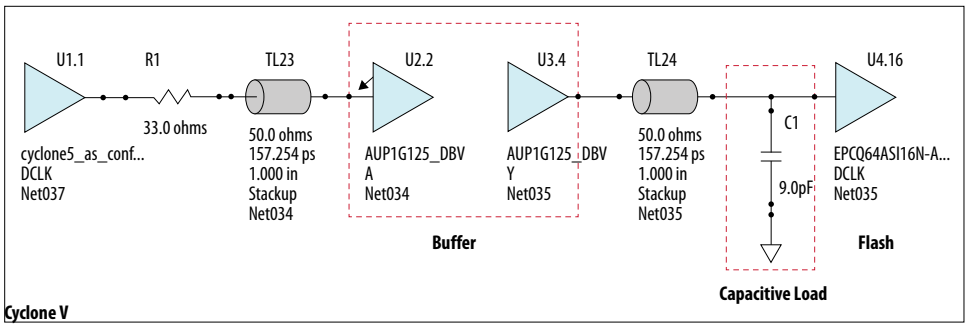
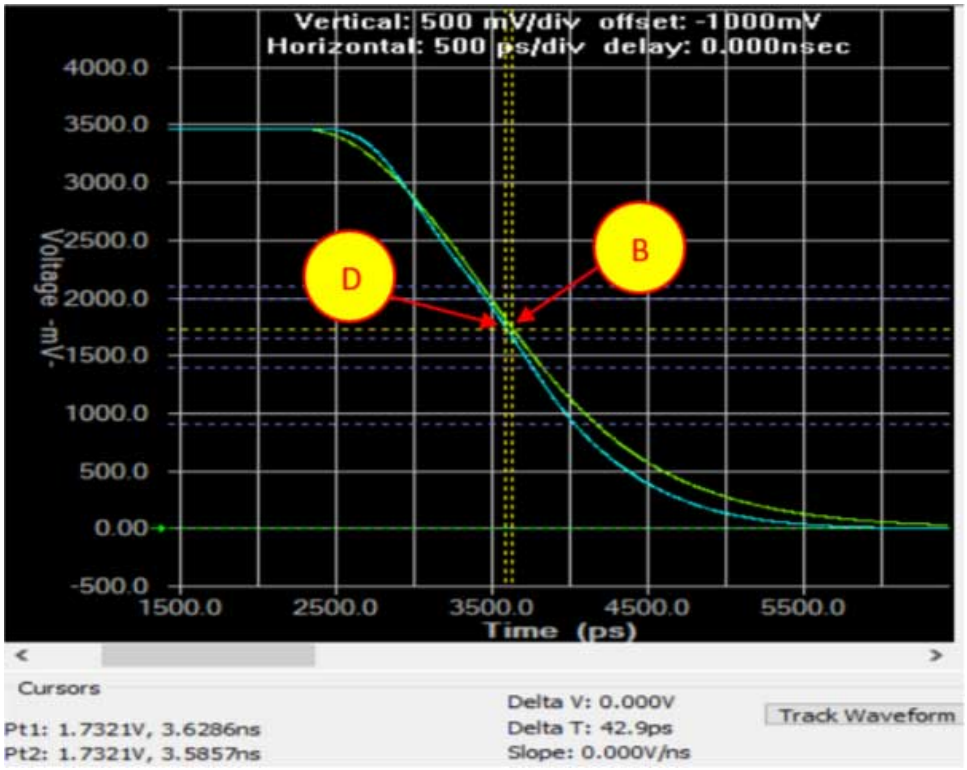


Figure 22. Minimum DCLK Delay for Delay Measurements A and C

Figure 23. Minimum DCLK Delay for Delay Measurements B and D



The minimum DCLK delay is shown in the following equation.

Minimum DCLK delay = (delay measurement C – delay measurement A) + [minimum buffer delay + (delay measurement D – delay measurement B)]

Minimum DCLK delay = –0.080ns + 1.8ns[28] – 0.0425ns

Minimum DCLK delay = 1.677ns

(28) The minimum buffer delay specification is 1.8ns at 15pF loading.

Figure 24. Maximum DCLK Delay for Delay Measurements A and C

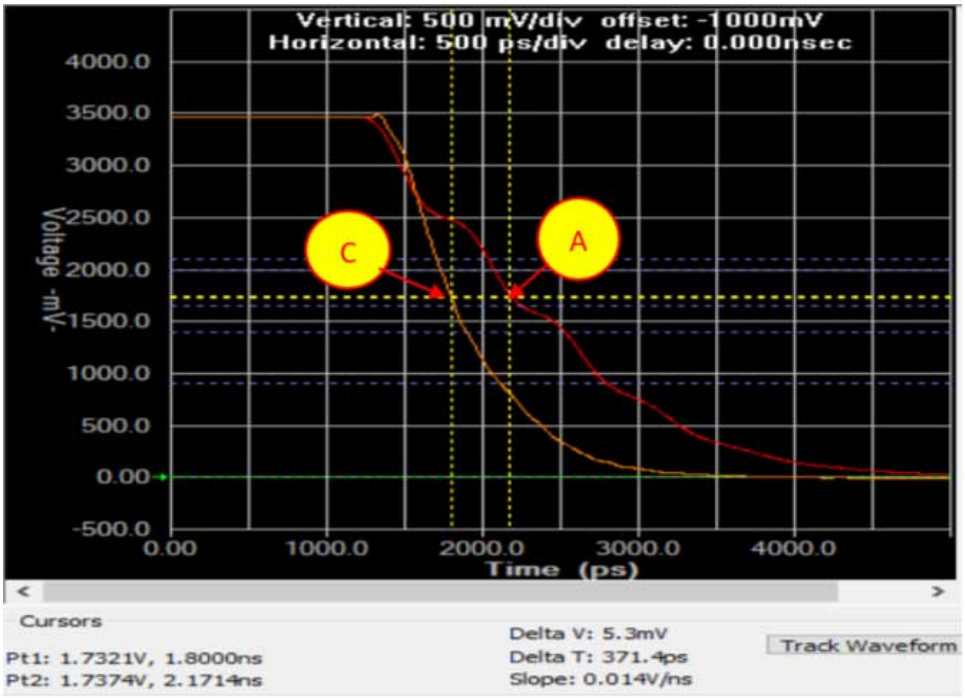
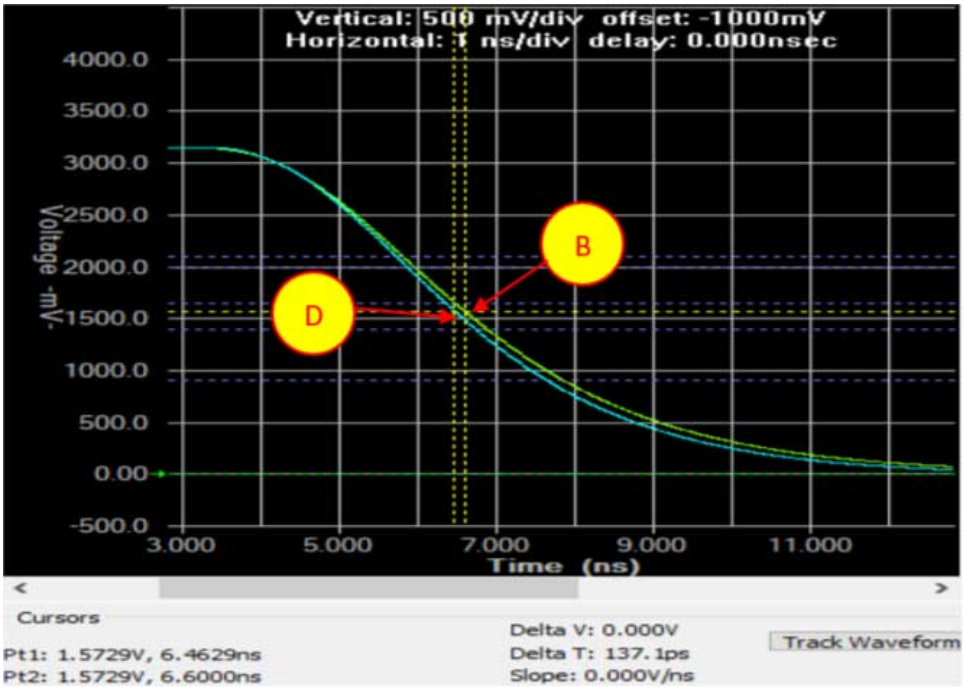


Figure 25. Maximum DCLK Delay for Delay Measurements B and D



The maximum DCLK delay is shown in the following equation.

Maximum DCLK delay = (delay measurement C – delay measurement A) + [maximum buffer delay + (delay measurement D – delay measurement B)]

Maximum DCLK delay = –0.371ns + 5.7ns[29] – 0.137ns

Maximum DCLK delay = 5.191ns

1.5.1.3.1. Simulation Results

The following table shows the minimum and maximum delays that can be obtained from the type of buffer used in this example.

Table 24. Minimum and Maximum DCLK Delays

- The data listed is based on the manual measurement from the simulation waveform.
- The EPCQ-A device DCLK loading is 6pF, a 9pF capacitive load is added in the simulation setup to match the expected output loading of the buffer specification.

IC Corner	Output Loading, CL (pF)	Cyclone V DCLK to Buffer Delay (ns)	Buffer to EPCQ-A DCLK Delay (ns)	Total DCLK Delay at $V_{CC}/2$ (ns)	Notes
Fast/Strong	15	–0.080	1.757	1.677	Minimum DCLK delay
Slow/Weak	15	–0.371	5.562	5.191	Maximum DCLK delay

Ultimately, the data setup time slack and the data hold time slack can be obtained by substituting the delay obtained from the IBIS simulation in the following equations.

Assumption for the system setup:

- DCLK frequency: 50MHz (period 20ns)
- Cyclone V AS timing specifications:
  - Minimum t_{DSU}: 1.5ns
  - Minimum t_{DH}: 2.9ns

This equation shows the data setup time slack calculation.

Data setup time slack = t_{DCLK} – t_{DSU} – (maximum DCLK delay + maximum DATA delay)

Data setup time slack = 20ns – 1.5ns – (5.191ns + 3.657ns)

Data setup time slack = 9.652ns

This equation shows the data hold time slack calculation.

Data hold time slack = (Minimum DCLK delay + minimum DATA delay) – t_{DH}

Data hold time slack = (1.677ns + 2.042ns) – 2.9ns

Data hold time slack = 0.819ns

(29) The maximum buffer delay specification is 5.7ns at 15pF loading.

The DCLK signal quality is good with the extra buffer as shown in the simulation waveform below. The DCLK signal is simulated using 50 Mbps clock pattern.

Figure 26. Simulation Results for Cyclone V FPGA-to-Buffer

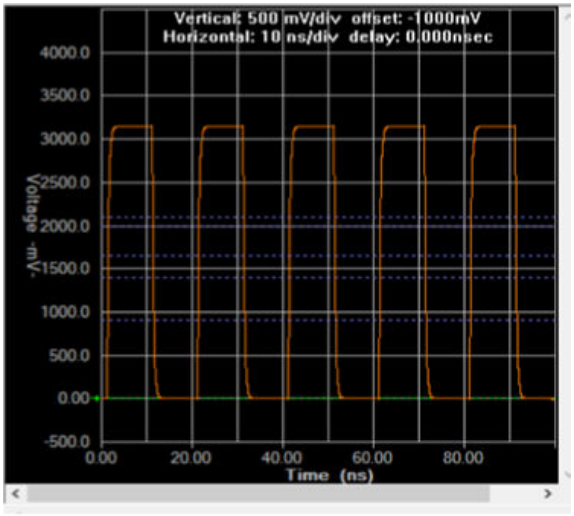
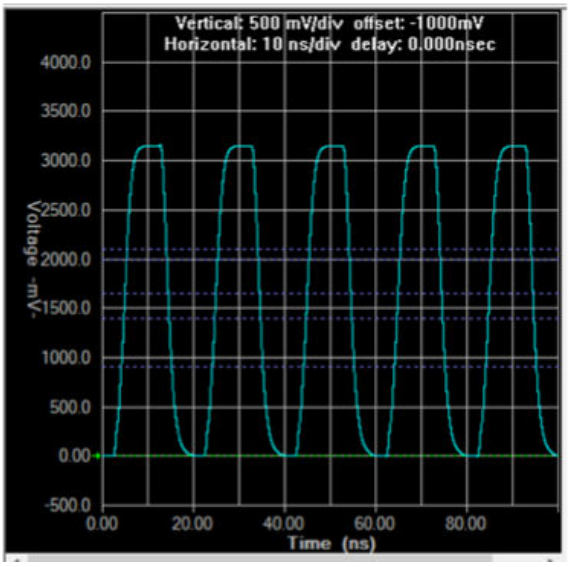


Figure 27. Simulation Results for Buffer-to-EPCQ-A Device



You can select any buffer in the market as long as the total delay for buffer, DCLK, and DATA is meeting the FPGA t_{DSU} and t_{DH} and the EPCQ-A setup and hold timing requirements.

By adding buffer, the DCLK and DATA trace lengths can be designed to as short as possible. The additional delay obtained from buffer, including other timing components help to meet the minimum data setup and hold time required by the Cyclone V FPGA.

Note: You need to add the appropriate capacitive load on the DCLK line. For more details, refer to the buffer's data sheet. Adding the capacitive load is essential to obtain the desired delay from the buffer. You must take the EPCQ-A or the third-party flash input pin capacitance into consideration to determine the additional capacitive load value.

1.5.1.4. Simulating the DCLK Signal at 100 MHz Operation

Example to add the RC network or buffer solution is not applicable to the 100 MHz operation because the additional delay uncertainty imposed by the RC network or buffer can cause setup time violation on the FPGA.

1.5.1.5. Trace Propagation Delay Recommendation For DCLK Running at a 100-MHz Operation

The following trace propagation delay recommendations are applicable when the EPCQ-A device is used.

- For –6 speed grade, the Cyclone V FPGA t_{DH} specification is 2.5ns.
  - The total propagation delay for the DCLK and DATA must be within the following range: 1.0ns < delay for DCLK + DATA < 2.5ns
  - Assuming the DCLK and DATA traces have the same length, the DCLK and DATA delay must be within the following range: 0.5ns < delay for DCLK + DATA < 1.25ns
- For –7 and –8 speed grades, the Cyclone V FPGA t_{DH} specification is 2.9ns.
  - The total propagation delay for the DCLK and DATA must be within the following range: 1.4ns < delay for DCLK + DATA < 2.5ns
  - Assuming the DCLK and DATA traces have the same length, the DCLK and DATA delay must be within the following range: 0.7ns < delay for DCLK + DATA < 1.25ns
- You must perform IBIS or link simulation to ensure the signal quality is good.

1.6. Cyclone V to Cyclone V QS Device Migration Reference Manual

As flash device manufacturers in the market are moving to new process technology, improved flash memories become faster with shorter delay. This has indirectly cause some challenges in the AS configuration system of the Cyclone V device to meet the data hold time when migrating to a newer flash device. Therefore, the Cyclone V QS device which has significantly improved the minimum data hold time requirement in comparison with the non-Cyclone V QS device has been produced and manufactured. Intel recommends using the Cyclone V device with the QS suffix to replace the Cyclone V device when migrating to a newer or faster flash device.

Note that the Cyclone V device with the QS suffix part are only available for the following packages:

- U15 (324 pins)
- U19 (484 pins)
- F23 (484 pins)
- U23 (672 pins)

- F27 (672 pins)
- F31 (896 pins)
- F35 (1152 pins)

The Cyclone V device with the QS suffix part are not available for the following packages:

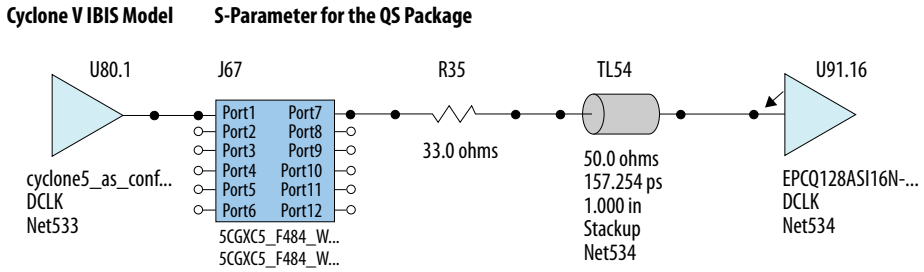
- M11 (301 pins)
- M13 (383 pins)
- M15 (484 pins)
- F17 (256 pins)

To reduce the required minimum data hold time (t_{DH}) for the AS configuration mode, a longer routing trace is implemented on the DCLK net in the Cyclone V QS package. The implementation will result in a higher overshoot or undershoot on the DCLK output signal at the receiver, depending on the actual board design. The overshoot or undershoot behavior is an expected behavior in certain combination of driver characteristics, package routing impedance, board design, board trace impedance, and receiver load configuration. The magnitude, slope, and duration of the overshoot or undershoot will depend on the total system transmission line length, impedance discontinuities, board design (serial resistor placement), and load configuration. The anticipated overshoot or undershoot behavior is within the overshoot or undershoot level and the duration permitted by the Intel FPGA configuration device (EPCQ-A) and third-party configuration devices supported by Intel. You can perform IBIS simulation based on the DCLK operating frequency and your system setup to ensure the overshoot or undershoot level and the duration are within the specifications permitted by the configuration device when migrating to a Cyclone V QS package.

The following steps are the manual intervention required to setup the IBIS simulation deck and to simulate the active serial interfaces for the Cyclone V QS package.

1. Use the Cyclone V IBIS model and the S-parameter file for the QS package. You can download the S-parameter file for the Cyclone V QS package using the *Cyclone V IBIS Models* (cyclone5.zip) from the *IBIS Models for Intel Devices* page provided on the ***Related Information*** section below.

Figure 28. Cyclone V QS IBIS Model with the S-parameter File



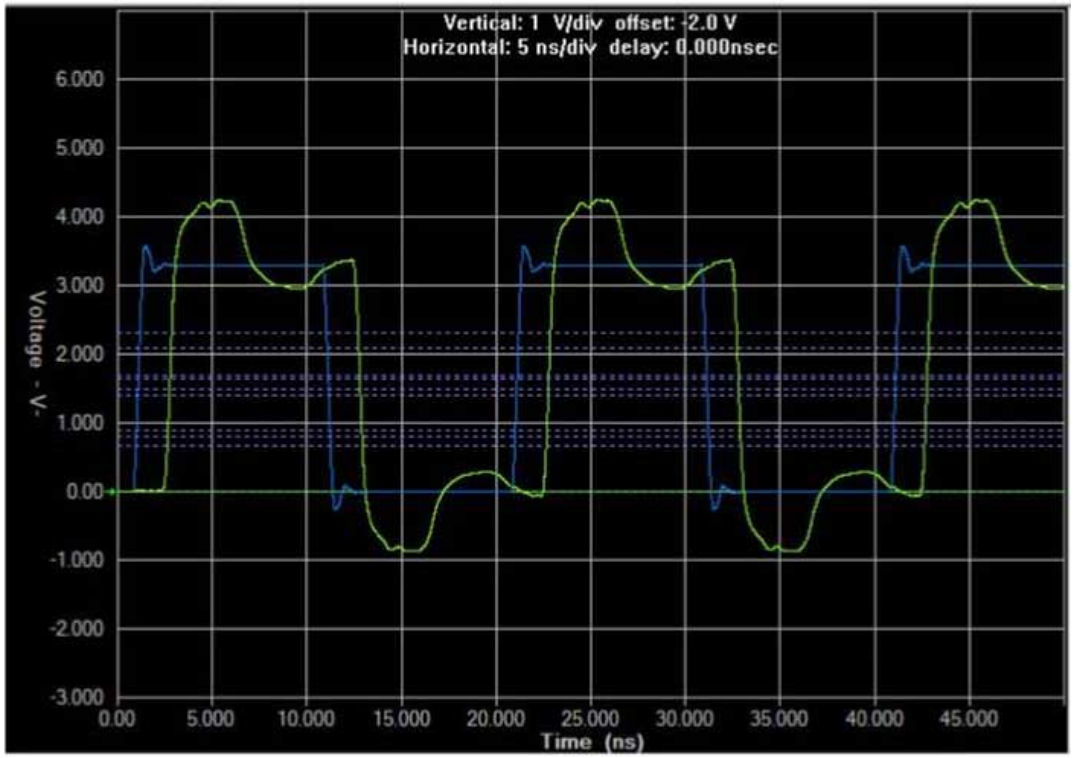
2. Manually set the package-lumped RLC and per pin RLC for the active serial interfaces to zero in the Cyclone V IBIS model.

```
|*****************************************************************************
|
|               IBIS Models for Cyclone 5 Device by ALTERA
|                  modified by Bruce for AS configuration simulation
|                   DO NOT USE
|
|*****************************************************************************

[IBIS Ver]    4.2
[File Name]   cyclone5_as_config_no_pkg.ibs
[File Rev]    3.0
[Date]        February 05, 2018
[Source]      Altera Corporation
[Notes]       - The models are designated as "Correlated" and have been
[Disclaimer] Data is for modeling purposes only and is not guaranteed.
[Copyright]  Copyright (C) 1991-2015 Altera Corporation. All rights reserved.
[Component]  cyclone5_as_config_no_pkg
[Manufacturer] Altera Corporation
[Package]
| RLC values are for 1152 FBGA package
| modified by Bruce for AS configuration simulation
| variable  typ        min        max
R_pkg       0          0          0
L_pkg       0          0          0
C_pkg       0          0          0

[Pin]     signal_name     model_name      R_pin     L_pin     C_pin
|
1     DCLK        lvtt130_rtnio_d12s1     0         0         0
2     DATA0       lvtt130_rtnio_d8s1      0         0         0
3     DATA1       lvtt130_rtnio_d8s1      0         0         0
4     DATA2       lvtt130_rtnio_d8s1      0         0         0
5     DATA3       lvtt130_rtnio_d8s1      0         0         0
6     nCS0        lvtt130_rtnio_d8s1      0         0         0
```

Example 3. IBIS Simulation for the Cyclone V QS Package when Interfacing with Intel EPCQ128ASI16N Model

Figure 29. Overshoot/Undershoot Observed on the DCLK Signal with DCLK Operating at 50MHz

- The overshoot or undershoot level is within the transient voltage specification stipulated in the *Absolute Maximum Rating Specification for EPCQ-A Devices* in the *EPCQ-A Serial Configuration Device Datasheet*.
- Blue signal: Simulation with non-Cyclone V QS package.
- Green signal: Simulation with Cyclone V QS package.



In the event of violating the overshoot or undershoot level permitted by the configuration device, Intel recommends you to add series termination resistor at the source side of the DCLK pin to reduce the overshoot or undershoot of the DCLK signal. Adding the series termination is essential to ensure the source side is terminated with the same impedance that matches the transmission line impedance. Although adding the series termination helps to reduce the overshoot or undershoot on the DCLK, this in turn has some drawbacks that increases the propagation delay and the rise or fall time of the signal. Therefore, you must re-analyze the AS configuration system timing to ensure the setup and hold time meets the requirements stipulated in the *Cyclone V FPGA Device Datasheet*.

Related Information

IBIS Models for Intel Devices
Provides link to download the S-parameter file for the Cyclone V QS package using the *Cyclone V IBIS Models* (cyclone5.zip).

1.7. Document Revision History for AN 822: Intel FPGA Configuration Device Migration Guideline

Document Version	Changes
2020.04.29	Updated the unit for resistance to mΩ in the *RLC Value Used for the DCLK and DATA Pins* table.
2020.04.16	Updated the reference to the *IBIS Models for Intel Devices* in the *Cyclone V to Cyclone V QS Device Migration Reference Manual* section.
2020.04.10	• Added the *Migration Method from EPCQ to EPCQ-A for Arria V, Cyclone V, and Stratix V* section. • Added the *Board Design Guidelines for Cyclone V to Cyclone V QS Device Migration* section. • Added reference to *How can I use EPCQ-A Serial Configuration Devices in ASMI Parallel II Intel FPGA IP when using Intel Quartus Prime Standard Edition software version 17.0.2 and earlier?* in the *Software Support for EPCQ-A Devices* section.
2019.10.17	• Added a note to the minimum value of the EPCQ and EPCQ-A devices of the V_{IH} parameter in the *EPCS, EPCQ and EPCQ-A Devices Operating Conditions* table. • Added a note about compression and encryption in the *Programming File Compatibility* section. • Added a note about using the 13.1.4 patch 4.70r in the *Software Migration Guidelines* section. • Updated the package dimensions in the following tables: — *Package Dimension Comparison for 8-Pin SOIC Package Devices* table — *Package Dimension Comparison for 16-Pin SOIC Package Devices* table • Updated the minimum value of the V_{IH} parameter for EPCS in the *EPCS, EPCQ and EPCQ-A Devices Operating Conditions* table.
2018.03.30	• Updated EPCQA instances to EPCQ-A. • Added a note and updated the compatibility conditions in *Programming File Compatibility Guide* table. • Updated document title from *AN 822: Intel Configuration Device Migration Guideline* to *AN 822: Intel FPGA Configuration Device Migration Guideline*. • Removed Preliminary tag for *EPCS to EPCQ-A Device Migration IP Core Compatibility* and *EPCQ to EPCQ-A Device Migration IP Core Compatibility* tables.

Date	Version	Changes
January 2018	2018.01.11	Edited the 2.5V I/O note in *EPCS, EPCQ and EPCQA Devices Operating Conditions* table.
December 2017	2017.12.15	Updated t_{nCLK2D} / t_{CLQV} for EPCQA device in *EPCS and EPCQA Devices Read Operation Timing Parameters* table.
August 2017	2017.08.14	• Updated Pin 1 in *16-pin SOIC Device Pin Information*. • Updated condition for Low-level output voltage in *EPCS, EPCQ and EPCQA Devices Operating Conditions*.
August 2017	2017.08.04	Updated minimum value for Low-level output voltage in *EPCS, EPCQ and EPCQA Device Operating Conditions*.
August 2017	2017.08.02	Initial release.