

MAX 10 High-Speed LVDS I/O User Guide



Subscribe



Send Feedback

UG-M10LVDS
2015.05.04

101 Innovation Drive
San Jose, CA 95134
www.altera.com



Contents

MAX 10 High-Speed LVDS I/O Design Overview.....	1-1
Altera Soft LVDS Implementation Overview.....	1-2
MAX 10 High-Speed LVDS Architecture and Features.....	2-1
MAX 10 LVDS Channels Support	2-1
MAX 10 LVDS SERDES I/O Standards Support.....	2-7
MAX 10 High-Speed LVDS Circuitry.....	2-10
MAX 10 High-Speed LVDS I/O Location.....	2-11
Differential I/O Pins in Low Speed Region.....	2-14
MAX 10 LVDS Transmitter Design.....	3-1
High-Speed I/O Transmitter Circuitry.....	3-1
LVDS Transmitter Programmable I/O Features.....	3-1
Programmable Pre-Emphasis.....	3-1
Programmable Differential Output Voltage.....	3-2
LVDS Transmitter I/O Termination Schemes.....	3-3
Emulated LVDS External Termination.....	3-3
Sub-LVDS Transmitter External Termination.....	3-3
SLVS Transmitter External Termination.....	3-4
Emulated RSDS, Emulated Mini-LVDS, and Emulated PPDS Transmitter External Termination.....	3-4
LVDS Transmitter FPGA Design Implementation.....	3-5
Altera Soft LVDS IP Core in Transmitter Mode.....	3-6
High-Speed I/O Timing Budget.....	3-10
Guidelines: LVDS Transmitter Channels Placement.....	3-11
Guidelines: LVDS Channels PLL Placement.....	3-11
Guidelines: LVDS Transmitter Logic Placement.....	3-12
LVDS Transmitter Debug and Troubleshooting.....	3-12
Perform RTL Simulation Before Hardware Debug.....	3-12
Geometry-Based and Physics-Based I/O Rules.....	3-13
MAX 10 LVDS Receiver Design.....	4-1
High-Speed I/O Receiver Circuitry.....	4-1
Soft Deserializer.....	4-1
Data Realignment Block (Bit Slip).....	4-2
LVDS Receiver I/O Termination Schemes.....	4-2
LVDS, Mini-LVDS, and RSDS Receiver External Termination.....	4-2
SLVS Receiver External Termination.....	4-3
Sub-LVDS Receiver External Termination.....	4-3
TMDS Receiver External Termination.....	4-4

HiSpi Receiver External Termination.....	4-4
LVPECL External Termination.....	4-5
LVDS Receiver FPGA Design Implementation.....	4-6
Altera Soft LVDS IP Core in Receiver Mode.....	4-6
High-Speed I/O Timing Budget.....	4-11
Guidelines: Floating LVDS Input Pins.....	4-14
Guidelines: LVDS Receiver Channels Placement.....	4-14
Guidelines: LVDS Channels PLL Placement.....	4-14
Guidelines: LVDS Receiver Logic Placement.....	4-15
Guidelines: LVDS Receiver Timing Constraints.....	4-15
LVDS Receiver Debug and Troubleshooting.....	4-15
Perform RTL Simulation Before Hardware Debug.....	4-15
Geometry-Based and Physics-Based I/O Rules.....	4-16
MAX 10 LVDS Transmitter and Receiver Design.....	5-1
Transmitter-Receiver Interfacing.....	5-1
LVDS Transmitter and Receiver FPGA Design Implementation.....	5-2
LVDS Transmitter and Receiver PLL Sharing Implementation.....	5-2
Initializing the Altera Soft LVDS IP Core.....	5-3
LVDS Transmitter and Receiver Debug and Troubleshooting.....	5-3
Perform RTL Simulation Before Hardware Debug.....	5-3
Geometry-Based and Physics-Based I/O Rules.....	5-3
MAX 10 High-Speed LVDS Board Design Considerations.....	6-1
Guideline: Improve Signal Quality.....	6-1
Guidelines: Control Channel-to-Channel Skew.....	6-1
Receiver Input Skew Margin.....	6-2
RSKM Report for LVDS Receiver.....	6-3
Guidelines: Determine Board Design Constraints.....	6-4
Guidelines: Perform Board Level Simulations.....	6-4
Altera Soft LVDS IP Core References.....	7-1
Altera Soft LVDS Parameter Settings	7-1
Altera Soft LVDS Interface Signals.....	7-6
Additional Information for MAX 10 High-Speed LVDS I/O User Guide.....	A-1
Document Revision History for MAX 10 High-Speed LVDS I/O User Guide.....	A-1

MAX 10 High-Speed LVDS I/O Design Overview

1

2014.12.15

UG-M10LVDS



Subscribe



Send Feedback

The MAX[®] 10 device family supports high-speed LVDS protocols through the LVDS I/O banks and the Altera Soft LVDS IP core.

Table 1-1: Summary of LVDS I/O Buffers Support in MAX 10 I/O Banks

I/O Buffer Type	I/O Bank Support
True LVDS input buffer	All I/O banks
True LVDS output buffer	Only bottom I/O banks
Emulated LVDS output buffer	All I/O banks

The LVDS I/O standards support differs between MAX 10 D and S variants. Refer to related information for more details.

Related Information

- [MAX 10 High-Speed LVDS Architecture and Features](#) on page 2-1
Provides information about the high-speed LVDS architecture and the features supported by the device.
- [MAX 10 LVDS Transmitter Design](#) on page 3-1
Provides information and guidelines for implementing LVDS transmitter in MAX 10 devices using the Altera Soft LVDS IP core.
- [MAX 10 LVDS Receiver Design](#) on page 4-1
Provides information and guidelines for implementing LVDS receiver in MAX 10 devices using the Altera Soft LVDS IP core.
- [MAX 10 LVDS Transmitter and Receiver Design](#) on page 5-1
Provides design guidelines for implementing both LVDS transmitters and receivers in the same MAX 10 device.
- [Altera Soft LVDS IP Core References](#) on page 7-1
Lists the parameters and signals of Altera Soft LVDS IP core for MAX 10 devices.
- [MAX 10 LVDS SERDES I/O Standards Support](#) on page 2-7
Lists the supported LVDS I/O standards and the support in different MAX 10 device variants.

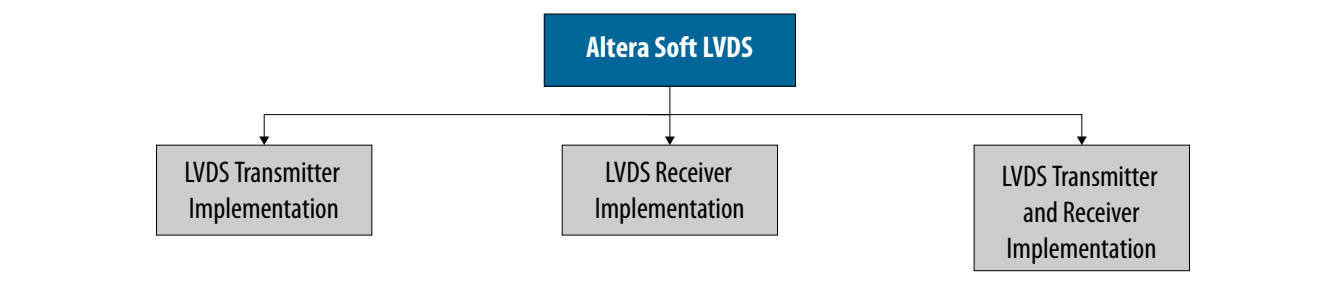
© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered

Altera Soft LVDS Implementation Overview

You can implement LVDS applications in MAX 10 devices as transmitter-only, receiver-only, or a combination of transmitters and receivers.

Figure 1-1: MAX 10 LVDS Implementation Overview



MAX 10 High-Speed LVDS Architecture and Features

2

2014.12.15

UG-M10LVDS



Subscribe



Send Feedback

The MAX 10 devices use registers and logic in the core fabric to implement LVDS input and output interfaces.

- For LVDS transmitters and receivers, MAX 10 devices use the the double data rate I/O (DDIO) registers that reside in the I/O elements (IOE). This architecture improves performance with regards to the receiver input skew margin (RSKM) or transmitter channel-to-channel skew (TCCS).
- For the LVDS serializer/deserializer (SERDES), MAX 10 devices use logic elements (LE) registers.

Related Information

- [MAX 10 High-Speed LVDS I/O Design Overview](#) on page 1-1
- [MAX 10 LVDS SERDES I/O Standards Support](#) on page 2-7
Lists the supported LVDS I/O standards and the support in different MAX 10 device variants.

MAX 10 LVDS Channels Support

The LVDS channels available vary among MAX 10 devices. All I/O banks in MAX 10 devices support true LVDS input buffers and emulated LVDS output buffers. However, only the bottom I/O banks support true LVDS output buffers.

© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered

Table 2-1: LVDS Buffers in MAX 10 Devices

This table lists the LVDS buffer support for I/O banks on each side of the devices.

Product Line	Package	Side	True LVDS Pairs		Emulated LVDS Pairs
			TX	RX	
10M02	V36	Top	0	1	1
		Right	0	3	3
		Left	0	3	3
		Bottom	3	3	3
	M153	Top	0	12	12
		Right	0	12	12
		Left	0	12	12
		Bottom	9	49	13
	U169	Top	0	12	12
		Right	0	17	17
		Left	0	15	15
		Bottom	9	14	14
	U324	Top	0	13	13
		Right	0	24	24
		Left	0	20	20
		Bottom	9	16	16
	E144	Top	0	10	10
		Right	0	12	12
		Left	0	11	11
		Bottom	7	12	12

Product Line	Package	Side	True LVDS Pairs		Emulated LVDS Pairs
			TX	RX	
10M04	M153	Top	0	12	12
		Right	0	12	12
		Left	0	12	12
		Bottom	9	13	13
	U169	Top	0	12	12
		Right	0	17	17
		Left	0	15	15
		Bottom	9	14	14
	U324	Top	0	27	27
		Right	0	31	31
		Left	0	28	28
		Bottom	15	28	28
	F256	Top	0	19	19
		Right	0	22	22
		Left	0	19	19
		Bottom	13	20	20
	E144	Top	0	8	8
		Right	0	12	12
		Left	0	11	11
		Bottom	10	10	10

Product Line	Package	Side	True LVDS Pairs		Emulated LVDS Pairs
			TX	RX	
10M08	V81	Top	0	5	5
		Right	0	7	7
		Left	0	6	6
		Bottom	7	7	7
	M153	Top	0	12	12
		Right	0	12	12
		Left	0	12	12
		Bottom	9	13	13
	U169	Top	0	12	12
		Right	0	17	17
		Left	0	15	15
		Bottom	9	14	14
	U324	Top	0	27	27
		Right	0	31	31
		Left	0	28	28
		Bottom	15	28	28
	F256	Top	0	19	19
		Right	0	22	22
		Left	0	19	19
		Bottom	13	20	20
	E144	Top	0	8	8
		Right	0	12	12
		Left	0	11	11
		Bottom	10	10	10
	F484	Top	0	27	27
		Right	0	33	33
		Left	0	28	28
		Bottom	15	28	28

Product Line	Package	Side	True LVDS Pairs		Emulated LVDS Pairs
			TX	RX	
10M16	U169	Top	0	12	12
		Right	0	17	17
		Left	0	15	15
		Bottom	9	14	14
	U324	Top	0	27	27
		Right	0	31	31
		Left	0	28	28
		Bottom	15	28	28
	F256	Top	0	19	19
		Right	0	22	22
		Left	0	19	19
		Bottom	13	20	20
	E144	Top	0	8	8
		Right	0	12	12
		Left	0	11	11
		Bottom	10	10	10
	F484	Top	0	39	39
		Right	0	38	38
		Left	0	32	32
		Bottom	22	42	42

Product Line	Package	Side	True LVDS Pairs		Emulated LVDS Pairs
			TX	RX	
10M25	F256	Top	0	19	19
		Right	0	22	22
		Left	0	19	19
		Bottom	13	20	20
	E144	Top	0	8	8
		Right	0	12	12
		Left	0	11	11
		Bottom	10	10	10
	F484	Top	0	41	41
		Right	0	48	48
		Left	0	36	36
		Bottom	24	46	46
10M40	F256	Top	0	19	19
		Right	0	22	22
		Left	0	19	19
		Bottom	13	20	20
	E144	Top	0	9	9
		Right	0	12	12
		Left	0	11	11
		Bottom	10	10	10
	F484	Top	0	41	41
		Right	0	48	48
		Left	0	36	36
		Bottom	24	46	46
	F672	Top	0	53	53
		Right	0	70	70
		Left	0	60	60
		Bottom	30	58	58

Product Line	Package	Side	True LVDS Pairs		Emulated LVDS Pairs
			TX	RX	
10M50	F256	Top	0	19	19
		Right	0	22	22
		Left	0	19	19
		Bottom	13	20	20
	E144	Top	0	9	9
		Right	0	12	12
		Left	0	11	11
		Bottom	10	10	10
	F484	Top	0	41	41
		Right	0	48	48
		Left	0	36	36
		Bottom	24	46	46
	F672	Top	0	53	53
		Right	0	70	70
		Left	0	60	60
		Bottom	30	58	58

MAX 10 LVDS SERDES I/O Standards Support

The MAX 10 D and S device variants support different LVDS I/O standards. All I/O banks in MAX 10 devices support true LVDS input buffers and emulated LVDS output buffers. However, only the bottom I/O banks support true LVDS output buffers.

Table 2-2: MAX 10 LVDS I/O Standards Support

Single and dual supply MAX 10 devices support different I/O standards. For more information about single and dual supply devices, refer to the device overview.

I/O Standard	I/O Bank	TX	RX	MAX 10 Device Support		Notes
				Dual Supply Device	Single Supply Device	
True LVDS	All	Bottom banks only	Yes	Yes	Yes	- All I/O banks support true LVDS input buffers. - Only the bottom I/O banks support true LVDS output buffers.
Emulated LVDS (three resistors)	All	Yes	—	Yes	Yes	All I/O banks support emulated LVDS output buffers.
True RSDS	Bottom	Yes	—	Yes	—	—
Emulated RSDS (single resistor)	All	Yes	—	Yes	—	All I/O banks support emulated RSDS output buffers.
Emulated RSDS (three resistors)	All	Yes	—	Yes	—	All I/O banks support emulated RSDS output buffers.
True Mini-LVDS	Bottom	Yes	—	Yes	—	—
Emulated Mini-LVDS (three resistors)	All	Yes	—	Yes	—	All I/O banks support emulated Mini-LVDS output buffers.
PPDS	Bottom	Yes	—	Yes	—	—
Emulated PPDS (three resistors)	All	Yes	—	Yes	—	—

I/O Standard	I/O Bank	TX	RX	MAX 10 Device Support		Notes
				Dual Supply Device	Single Supply Device	
Bus LVDS	All	Yes	Yes	Yes	Yes	- Bus LVDS (BLVDS) output uses two single-ended outputs with the second output programmed as inverted. - BLVDS input uses LVDS input buffer. - You can tristate BLVDS output.
LVPECL	All	—	Yes	Yes	Yes	Supported only on dual function clock input pins.
TMDS	All	—	Yes	Yes	—	- Requires external termination but does not require V_{REF}. - Requires external level shifter to support 3.3 V TMDS input. This level shifter must convert the TMDS signal from AC-coupled to DC-coupled before you connect it to the MAX 10 input buffer. - TMDS receiver support uses dedicated 2.5 V LVDS input buffer.
Sub-LVDS	All	Yes	Yes	Yes	—	- Transmitter supports only emulated Sub-LVDS using emulated 1.8 V differential signal as output. - Requires external output termination. - Does not require V_{REF}. - Sub-LVDS receiver support uses dedicated 2.5 V LVDS input buffer.

I/O Standard	I/O Bank	TX	RX	MAX 10 Device Support		Notes
				Dual Supply Device	Single Supply Device	
SLVS	All	Yes	Yes	Yes	—	• SLVS transmitter support uses emulated LVDS output. • Requires external termination but does not require V_{REF}. • SLVS receiver support uses dedicated 2.5 V LVDS input buffer.
HiSpi	All	—	Yes	Yes	—	• Only input is supported because HiSpi is a unidirectional I/O standard. • Requires external termination but does not require V_{REF}. • HiSpi receiver support uses dedicated 2.5 V LVDS input buffer.

Related Information

- [MAX 10 FPGA Device Overview](#)
- [Emulated LVDS External Termination](#) on page 3-3
- [Emulated RSDS, Emulated Mini-LVDS, and Emulated PPDS Transmitter External Termination](#) on page 3-4
- [TMDS Receiver External Termination](#) on page 4-4
- [Sub-LVDS Transmitter External Termination](#) on page 3-3
- [Sub-LVDS Receiver External Termination](#) on page 4-3
- [SLVS Transmitter External Termination](#) on page 3-4
- [SLVS Receiver External Termination](#) on page 4-3
- [HiSpi Receiver External Termination](#) on page 4-4

MAX 10 High-Speed LVDS Circuitry

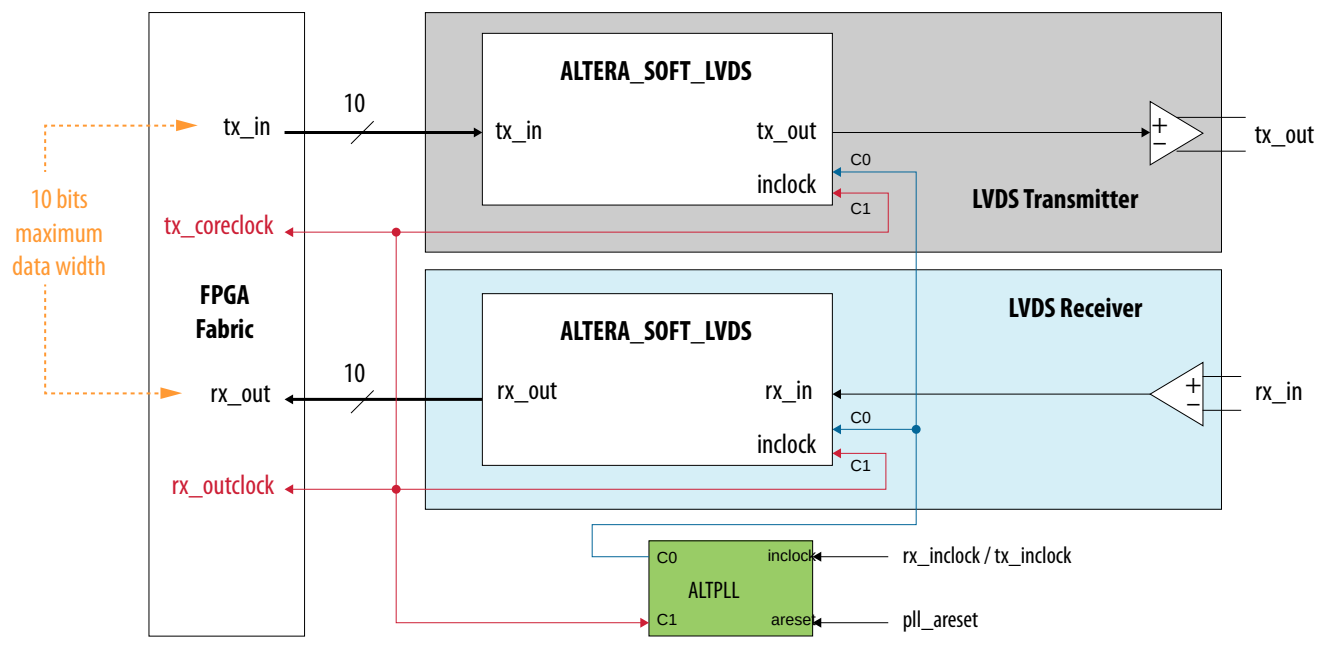
The LVDS solution uses the I/O elements and registers in the MAX 10 devices. The Altera Soft LVDS IP core implements the serializer and deserializer as soft SERDES blocks in the core logic.

The MAX 10 devices do not contain dedicated serialization or deserialization circuitry:

- You can use I/O pins and core fabric to implement a high-speed differential interface in the device.
- The MAX 10 solution uses shift registers, internal PLLs, and I/O elements to perform the serial-to-parallel and parallel-to-serial conversions of incoming and outgoing data.
- The Quartus® II software uses the parameter settings of the Altera Soft LVDS IP core to automatically construct the differential SERDES in the core fabric.

Figure 2-1: Soft LVDS SERDES

This figure shows a transmitter and receiver block diagram for the soft LVDS SERDES circuitry with the interface signals of the transmitter and receiver data paths.



Related Information

[MAX 10 Clocking and PLL User Guide](#)

Provides more information about the PLL and the PLL output counters.

MAX 10 High-Speed LVDS I/O Location

The I/O banks in MAX 10 devices support true LVDS input and emulated LVDS output on all I/O banks. Only the bottom I/O banks support true LVDS output.

Figure 2-2: LVDS Support in I/O Banks of 10M02 Devices

This figure shows a top view of the silicon die. Each bank is labeled with the actual bank number. LVPECL support only in banks 2 and 6.

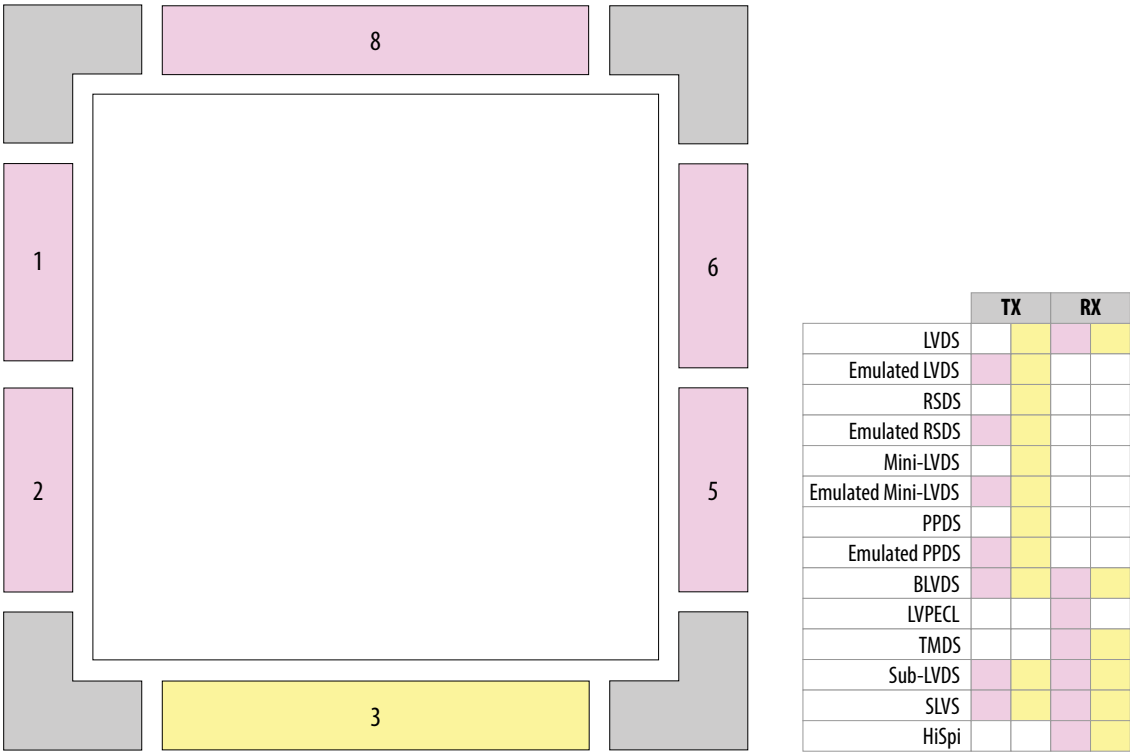


Figure 2-3: LVDS Support in I/O Banks of 10M04 and 10M08 Devices

This figure shows a top view of the silicon die. Each bank is labeled with the actual bank number. LVPECL support only in banks 2 and 6.

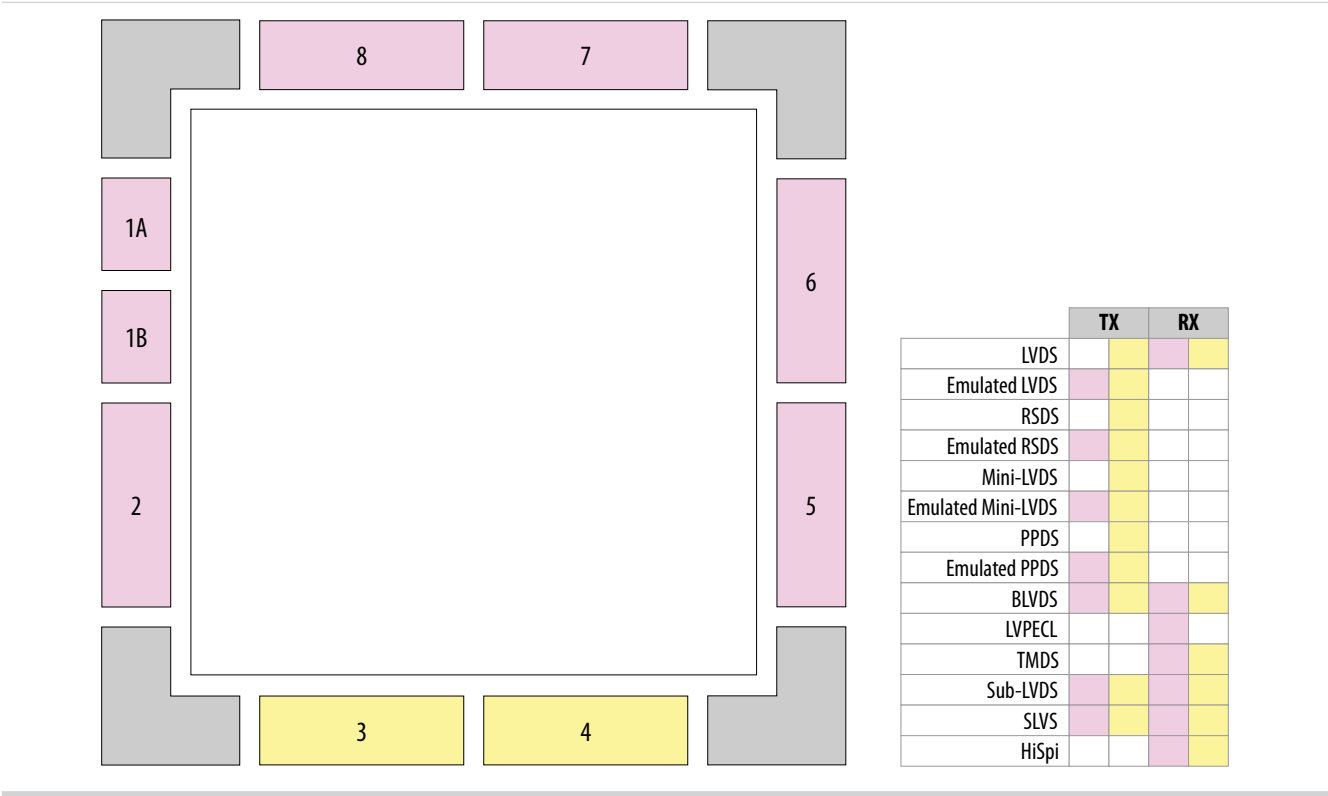
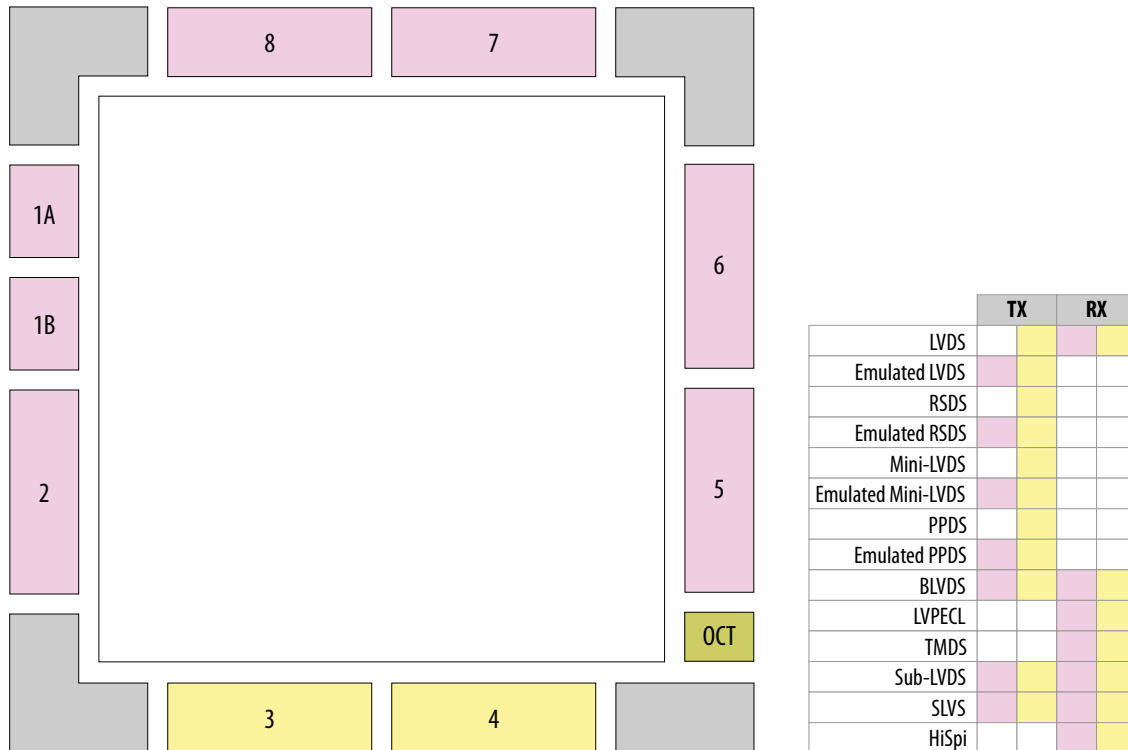


Figure 2-4: LVDS Support in I/O Banks of 10M16, 10M25, 10M40, and 10M50 Devices

This figure shows a top view of the silicon die. Each bank is labeled with the actual bank number. LVPECL support only in banks 2, 3, 6, and 8.



Differential I/O Pins in Low Speed Region

Some of the differential I/O pins are located in the low speed region of the MAX 10 device.

- For each user I/O pin (excluding configuration pin) that you place in the low speed region, the Quartus II software generates an informational warning message.
- Refer to the device pinout to identify the low speed I/O pins.
- Refer to the device datasheet for the performance information of these I/O pins.

Related Information

- [MAX 10 Device Pin-Out Files](#)
Provides pin-out files for each MAX 10 device.
- [MAX 10 Device Datasheet](#)

2014.12.15

UG-M10LVDS



Subscribe



Send Feedback

You can implement transmitter-only applications using the MAX 10 LVDS solution. You can use the Altera Soft LVDS IP core to instantiate soft SERDES circuitry. The soft SERDES circuitry works with the clocks and differential I/O pins to create a high-speed differential transmitter circuit.

Related Information

- [MAX 10 High-Speed LVDS I/O Design Overview](#) on page 1-1
- [MAX 10 LVDS SERDES I/O Standards Support](#) on page 2-7
Lists the supported LVDS I/O standards and the support in different MAX 10 device variants.

High-Speed I/O Transmitter Circuitry

The LVDS transmitter circuitry uses the I/O elements and registers in the MAX 10 devices. The Altera Soft LVDS IP core implements the serializer as a soft SERDES block in the core logic.

Related Information

[MAX 10 High-Speed LVDS Circuitry](#) on page 2-10

LVDS Transmitter Programmable I/O Features

You can program some features of the I/O buffers and pins in MAX 10 devices according to your design requirements. For high-speed LVDS transmitter applications, you can program the pre-emphasis setting.

Programmable Pre-Emphasis

The differential output voltage (V_{OD}) setting and the output impedance of the driver set the output current limit of a high-speed transmission signal. At a high frequency, the slew rate may not be fast enough to reach the full V_{OD} level before the next edge, producing pattern-dependent jitter. Pre-emphasis momentarily boosts the output current during switching to increase the output slew rate.

Pre-emphasis increases the amplitude of the high-frequency component of the output signal. This increase compensates for the frequency-dependent attenuation along the transmission line.

The overshoot introduced by the extra current occurs only during change of state switching. This overshoot increases the output slew rate but does not ring, unlike the overshoot caused by signal reflection. The amount of pre-emphasis required depends on the attenuation of the high-frequency component along the transmission line.

© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered

Figure 3-1: LVDS Output with Programmable Pre-Emphasis

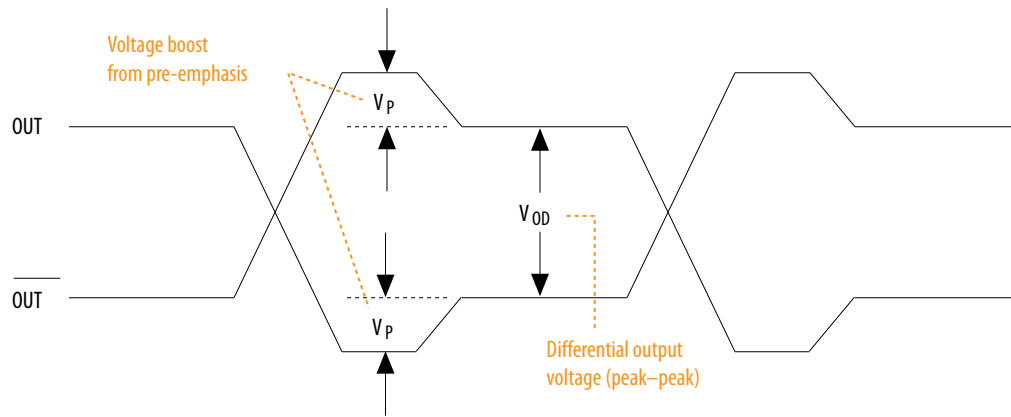


Table 3-1: Quartus II Software Assignment for Programmable Pre-Emphasis

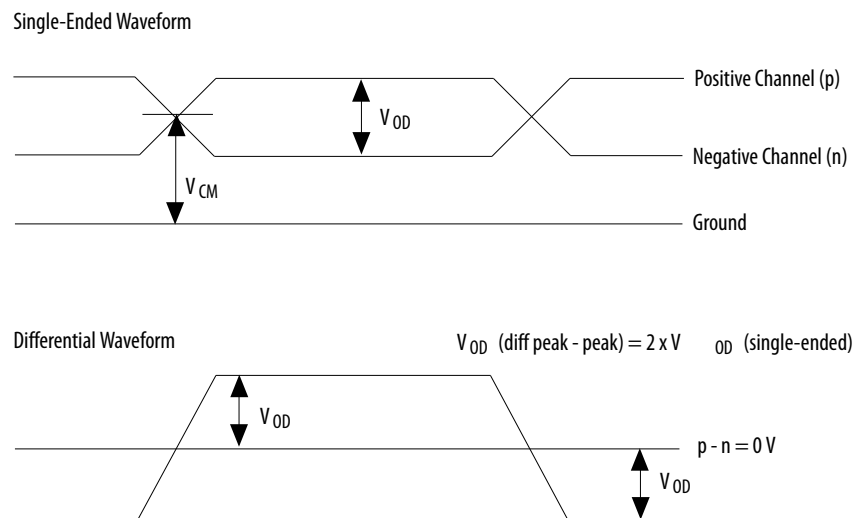
Field	Assignment
To	tx_out
Assignment name	Programmable Pre-emphasis
Allowed values	0 (disabled), 1 (enabled). Default is 1.

Programmable Differential Output Voltage

The programmable V_{OD} settings allow you to adjust the output eye opening to optimize the trace length and power consumption. A higher V_{OD} swing improves voltage margins at the receiver end, and a smaller V_{OD} swing reduces power consumption.

Figure 3-2: Differential V_{OD}

This figure shows the V_{OD} of the differential LVDS output.



You can statically adjust the V_{OD} of the differential signal by changing the V_{OD} settings in the Quartus II software Assignment Editor.

Table 3-2: Quartus II Software Assignment Editor—Programmable V_{OD}

Field	Assignment
To	tx_out
Assignment name	Programmable Differential Output Voltage (V_{OD})
Allowed values	0 (low), 1 (medium), 2 (high). Default is 2.

LVDS Transmitter I/O Termination Schemes

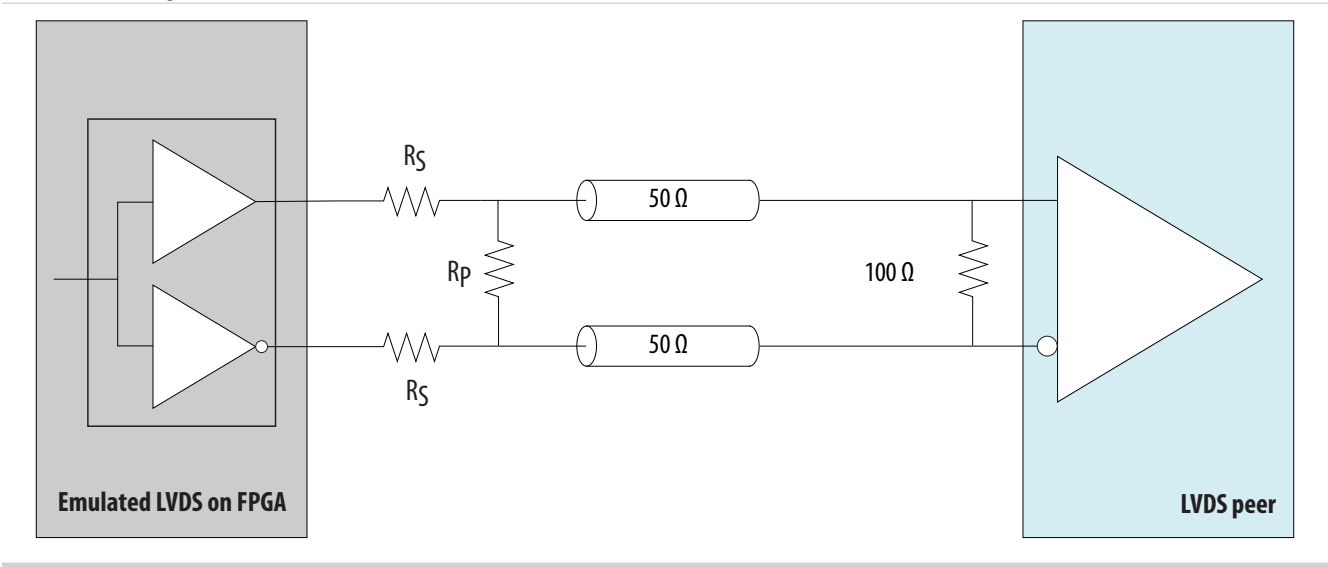
For transmitter applications in MAX 10 devices, you must implement external termination for some I/O standards.

Emulated LVDS External Termination

The emulated LVDS transmitter requires a three-resistor external termination scheme.

Figure 3-3: External Termination for Emulated LVDS Transmitter

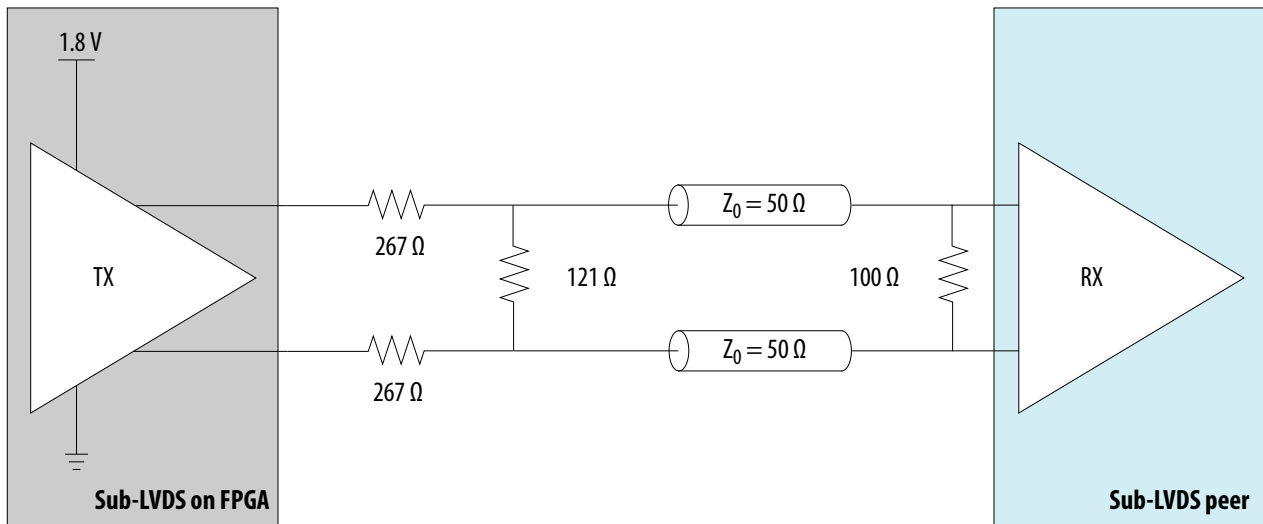
In this figure, $R_S = 120\ \Omega$ and $R_P = 170\ \Omega$.



Sub-LVDS Transmitter External Termination

The Sub-LVDS transmitter requires a three-resistor external termination scheme.

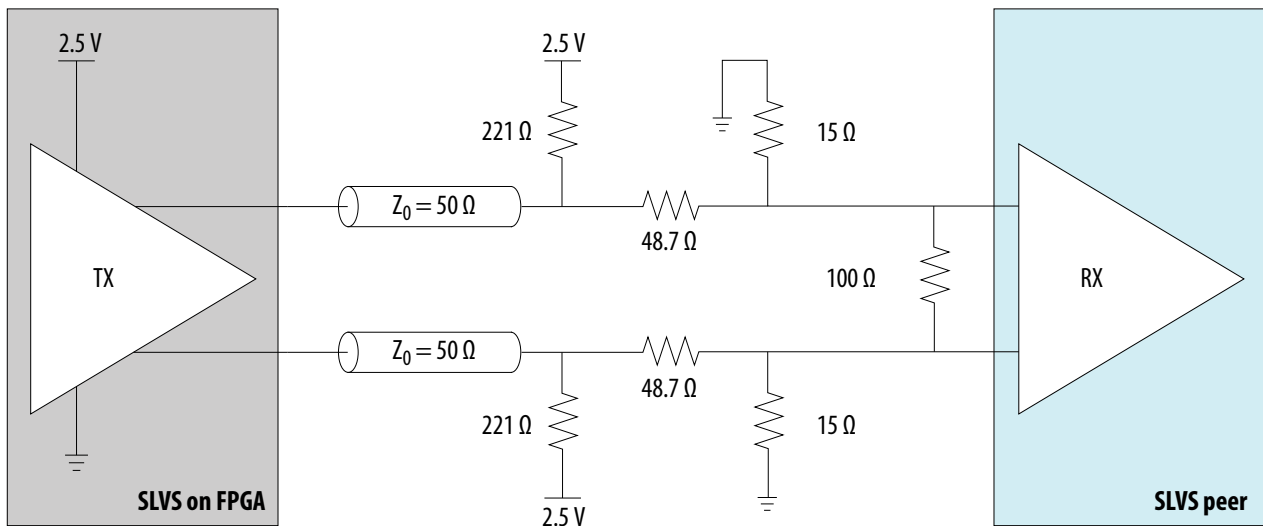
Figure 3-4: External Termination for Sub-LVDS Transmitter



SLVS Transmitter External Termination

The SLVS transmitter requires a three-resistor external termination scheme.

Figure 3-5: External Termination for SLVS Transmitter

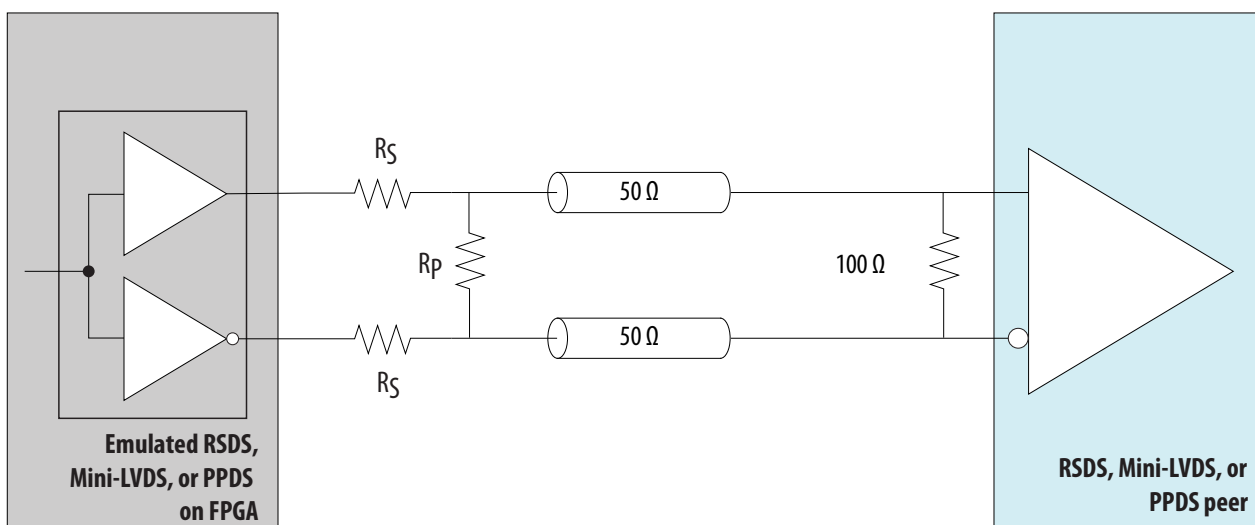
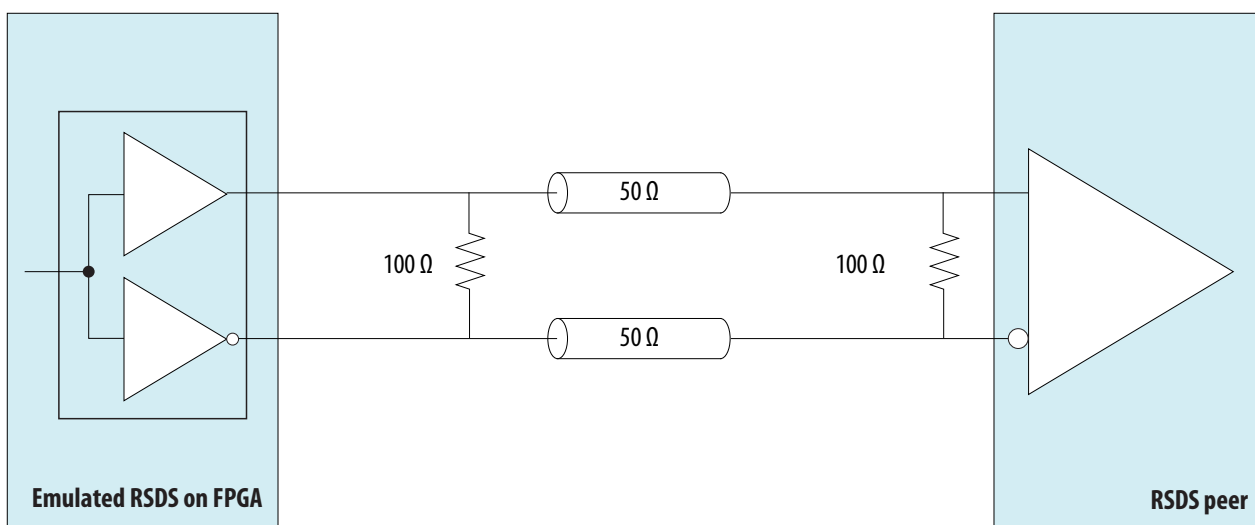


Emulated RSDS, Emulated Mini-LVDS, and Emulated PPDS Transmitter External Termination

The emulated RSDS, emulated mini-LVDS, or emulated PPDS transmitter requires a three-resistor external termination scheme. You can also use a single-resistor external termination for the emulated RSDS transmitter.

Figure 3-6: External Termination for Emulated RSDS, Mini-LVDS, or PPDS Transmitter

In this figure, R_S is $120\ \Omega$ and R_P is $170\ \Omega$.

**Figure 3-7: Single-Resistor External Termination for Emulated RSDS Transmitter**

LVDS Transmitter FPGA Design Implementation

MAX 10 devices use a soft SERDES architecture to support high-speed I/O interfaces. The Quartus II software creates the SERDES circuits in the core fabric by using the Altera Soft LVDS IP core. To improve the timing performance and support the SERDES, MAX 10 devices use the I/O registers and LE registers in the core fabric.

Altera Soft LVDS IP Core in Transmitter Mode

In the Quartus II software, you can design your high-speed transmitter interfaces using the Altera Soft LVDS IP core. This IP core uses the resources optimally in the MAX 10 devices to create the high-speed I/O interfaces.

- You can use the Altera Soft LVDS parameter editor to customize your serializer based on your design requirements.
- The high-speed I/O interface created using the Altera Soft LVDS IP core always sends the most significant bit (MSB) of your parallel data first.

Related Information

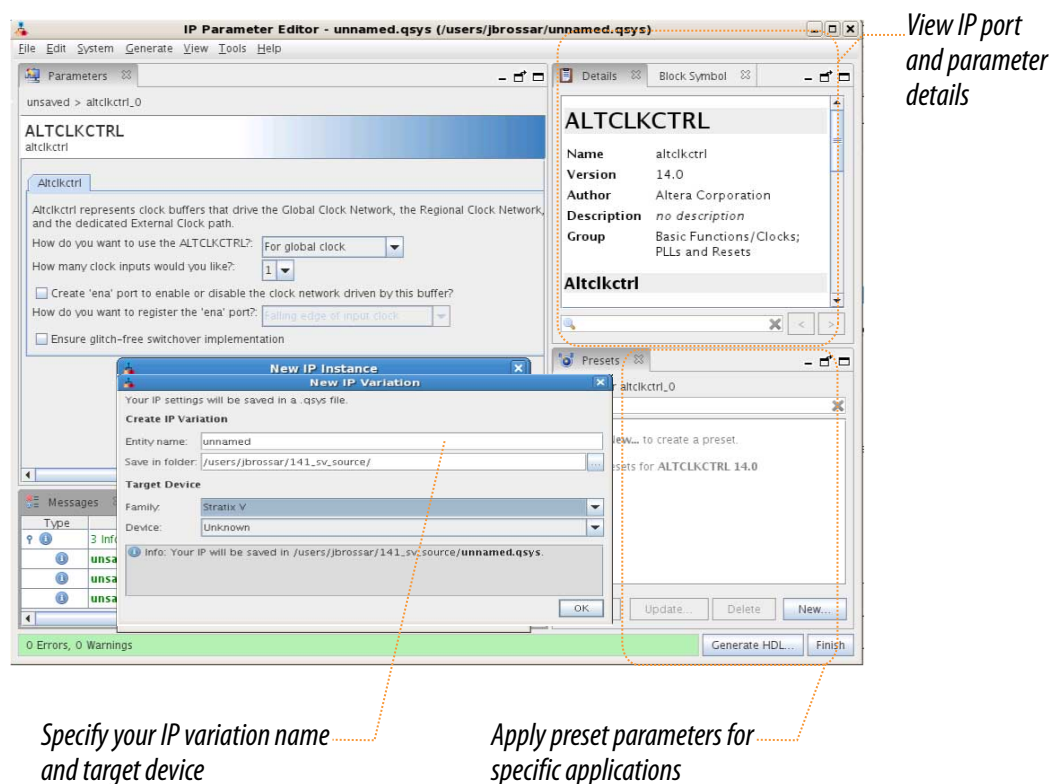
[Altera Soft LVDS Parameter Settings](#) on page 7-1

Specifying IP Core Parameters and Options

You can quickly configure a custom IP variation in the parameter editor. Use the following steps to specify IP core options and parameters in the parameter editor. Refer to *Specifying IP Core Parameters and Options (Legacy Parameter Editors)* for configuration of IP cores using the legacy parameter editor.

1. In the IP Catalog (**Tools > IP Catalog**), locate and double-click the name of the IP core to customize. The parameter editor appears.
2. Specify a top-level name for your custom IP variation. The parameter editor saves the IP variation settings in a file named `<your_ip>.qsys`. Click **OK**.
3. Specify the parameters and options for your IP variation in the parameter editor, including one or more of the following. Refer to your IP core user guide for information about specific IP core parameters.
 - Optionally select preset parameter values if provided for your IP core. Presets specify initial parameter values for specific applications.
 - Specify parameters defining the IP core functionality, port configurations, and device-specific features.
 - Specify options for processing the IP core files in other EDA tools.
4. Click **Generate HDL**, the **Generation** dialog box appears.
5. Specify output file generation options, and then click **Generate**. The IP variation files generate according to your specifications.
6. To generate a simulation testbench, click **Generate > Generate Testbench System**.
7. To generate an HDL instantiation template that you can copy and paste into your text editor, click **Generate > HDL Example**.
8. Click **Finish**. The parameter editor adds the top-level `.qsys` file to the current project automatically. If you are prompted to manually add the `.qsys` file to the project, click **Project > Add/Remove Files in Project** to add the file.
9. After generating and instantiating your IP variation, make appropriate pin assignments to connect ports.

Figure 3-8: IP Parameter Editor



Related Information

[Altera Soft LVDS Parameter Settings](#) on page 7-1

PLL Source Selection for Altera Soft LVDS IP Core

You can create the LVDS interface components by instantiating the Altera Soft LVDS IP core with an internal or external PLL.

Instantiate Altera Soft LVDS IP Core with Internal PLL

You can set the Altera Soft LVDS IP core to build the SERDES components and instantiate the PLL internally.

- To use this method, turn off the **Use external PLL** option in the **PLL Settings** tab and set the necessary settings in the **PLL Settings** and **Transmitter Settings** tab.
- The Altera Soft LVDS IP core integrates the PLL into the LVDS block.
- The drawback of this method is that you can use the PLL only for the particular LVDS instance.

Instantiate Altera Soft LVDS IP Core with External PLL

You can set the Altera Soft LVDS IP core to build only the SERDES components but use an external PLL source.

- To use this method, turn on the **Use external PLL** option in the **PLL Settings** tab.
- Follow the required clock setting to the input ports as listed in the notification panel.
- You can create your own clocking source using the ALTPLL IP core.
- Use this method to optimize PLL usage with other functions in the core.

Related Information

- [MAX 10 Clocking and PLL User Guide](#)
Provides more information about the PLL and the PLL output counters.
- [MAX 10 Clocking and PLL User Guide](#)
Provides more information about the PLL and the PLL output counters.

Guidelines: LVDS TX Interface Using External PLL

You can instantiate the Altera Soft LVDS IP core with the **Use External PLL** option. Using external PLL, you can control the PLL settings. For example, you can dynamically reconfigure the PLL to support different data rates and dynamic phase shifts. To use this option, you must instantiate the ALTPLL IP core to generate the various clock signals.

If you turn on the **Use External PLL** option for the Altera Soft LVDS transmitter, you require the following signals from the ALTPLL IP core:

- Serial clock input to the tx_inclock port of the Altera Soft LVDS transmitter.
- Parallel clock used to clock the transmitter FPGA fabric logic and connected to the tx_syncclock port.

Related Information

[MAX 10 Clocking and PLL User Guide](#)

Provides more information about the PLL and the PLL output counters.

ALTPLL Signal Interface with Altera Soft LVDS Transmitter

You can choose any of the PLL output clock ports to generate the LVDS interface clocks.

If you use the ALTPLL IP core as the external PLL source of the Altera Soft LVDS transmitter, use the source-synchronous compensation mode.

Table 3-3: Example: Signal Interface en ALTPLL and Altera Soft LVDS Transmitter

From the ALTPLL IP Core	To the Altera Soft LVDS Transmitter
Fast clock output (c0) The fast clock output (c0) can only drive tx_inclock on the Altera Soft LVDS transmitter.	tx_inclock
Slow clock output (c1)	tx_syncclock

Determining External PLL Clock Parameters for Altera Soft LVDS Transmitter

To determine the ALTPLL IP core clock parameter for the Altera Soft LVDS IP core transmitter, follow these steps in your design:

1. Instantiate the Altera Soft LVDS IP core transmitter using internal PLL.
2. Compile the design up to TimeQuest timing analysis.
3. In the **Table of Contents** section of the **Compilation Report** window, navigate to **TimeQuest Timing Analyzer > Clocks**.
4. Note the clock parameters used by the internal PLL for the Altera Soft LVDS IP core transmitter.

In the list of clocks, `clk0` is the fast clock.

Figure 3-9: Clock Parameters Example for Altera Soft LVDS Transmitter

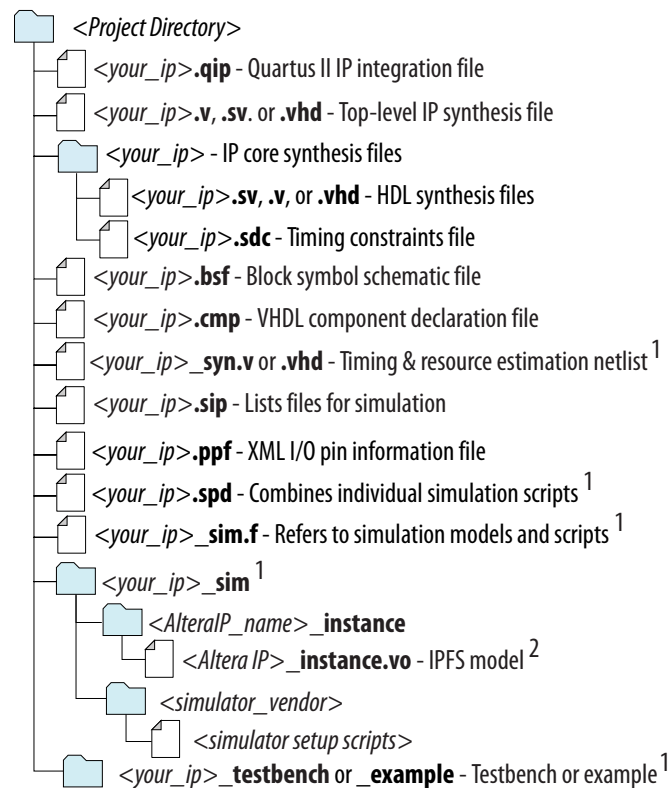
Clocks						
	Clock Name	Type	Period	Frequency	Duty Cycle	Phase
1	<code>inst lvdstx_inst lvds_tx_pll clk[0]</code>	Generated	2.500	400.0 MHz	50.00	-90.0
2	<code>inst lvdstx_inst lvds_tx_pll clk[1]</code>	Generated	12.500	80.0 MHz	50.00	-18.0
3	<code>tx_inclock</code>	Base	10.000	100.0 MHz		

Configure the ALTPLL output clocks with the parameters you noted in this procedure and connect the clock outputs to the correct Altera Soft LVDS clock input ports.

Files Generated for Altera IP Cores (Legacy Parameter Editor)

The Quartus II software version 14.0 and previous generates the following output for IP cores that use the legacy MegaWizard parameter editor.

Figure 3-10: IP Core Generated Files

**Notes:**

1. If supported and enabled for your IP variation
2. If functional simulation models are generated

Initializing the Altera Soft LVDS IP Core

The PLL locks to the reference clock before the Altera Soft LVDS IP core implements the SERDES blocks for data transfer.

During device initialization the PLL starts to lock to the reference clock and becomes operational when it achieves lock during user mode. If the clock reference is not stable, it corrupts the phase shifts of the PLL output clock. This phase shifts corruption can cause failure and corrupt data transfer between the high-speed LVDS domain and the low-speed parallel domain.

To avoid data corruption, follow these steps when initializing the Altera Soft LVDS IP core:

1. Assert the `pll_areset` signal for at least 10 ns.
2. After at least 10 ns, deassert the `pll_areset` signal.
3. Wait until the PLL lock becomes stable.

After the PLL lock port asserts and is stable, the SERDES blocks are ready for operation.

High-Speed I/O Timing Budget

The LVDS I/O standard enables high-speed transmission of data that results in better overall system performance. To take advantage of fast system performance, you must analyze the timing for high-speed

signals. Timing analysis for the differential block is different than traditional synchronous timing analysis techniques.

The basis of the source synchronous timing analysis is the skew between the data and the clock signals instead of the clock-to-output setup times. High-speed differential data transmission requires the use of timing parameters provided by IC vendors and is strongly influenced by board skew, cable skew, and clock jitter.

Transmitter Channel-to-Channel Skew

The receiver input skew margin (RSKM) calculation uses the transmitter channel-to-channel skew (TCCS)—an important parameter based on the MAX 10 transmitter in a source-synchronous differential interface.

- For LVDS transmitters, the TimeQuest Timing Analyzer provides the TCCS value in the TCCS report (report_TCCS) in the Quartus II compilation report. The report shows TCCS values for serial output ports.
- You can also get the TCCS value from the device datasheet.

Related Information

[MAX 10 Device Datasheet](#)

Guidelines: LVDS Transmitter Channels Placement

To maintain an acceptable noise level on the V_{CCIO} supply, observe the placement restrictions for single-ended I/O pins in relation to differential pads.

Altera recommends that you create a Quartus II design, specify your device I/O assignments, and compile your design to validate your pin placement. The Quartus II software verifies your pin connections against the I/O assignment and placement rules to ensure that the device will operate properly.

You can use the Quartus II Pin Planner Package view to ease differential I/O assignment planning:

- On the **View** menu, click **Show Differential Pin Pair Connections** to highlight the differential pin pairing. The differential pin pairs are connected with red lines.
- For differential pins, you only need to assign the signal to a positive pin. The Quartus II software automatically assigns the negative pin if the positive pin is assigned with a differential I/O standard.

In MAX 10 devices, the routing of each differential pin pair is matched. Consequently, the skew between the positive and the negative pins is minimal. The internal routings of both pins in a differential pair are matched even if the pins are non-adjacent.

MAX 10 support x18 bundling mode. To control skew, place all LVDS channels in the same bundle side by side and the channels must not cross I/O banks. In your PCB design, Altera recommends package skew compensation to minimize skew and maximize performance.

Guidelines: LVDS Channels PLL Placement

Each PLL in the MAX 10 device can drive only the LVDS channels in I/O banks on the same edge as the PLL.

Table 3-4: Examples of Usable PLL to Drive I/O Banks in MAX 10 Devices

I/O Bank Edge	Input refclk	GCLK mux	Usable PLL
Left	Left	Left	Top left or bottom left

I/O Bank Edge	Input refclk	GCLK mux	Usable PLL
Bottom	Bottom	Bottom	Bottom left or bottom right
Right	Right	Right	Top right or bottom right
Top	Top	Top	Top left or top right

Guidelines: LVDS Transmitter Logic Placement

The Quartus II software automatically optimizes the SERDES logic placement to meet the timing requirements. Therefore, you do not have to perform placement constraints on the Altera Soft LVDS IP core logic.

To improve the performance of the Quartus II Fitter, you can create LogicLock™ regions in the device floorplan to confine the transmitter SERDES logic placement.

- The TCCS parameter is guaranteed per datasheet specification to the entire bank of differential I/Os that are located in the same side. This guarantee applies if the transmitter SERDES logic is placed within the LAB adjacent to the output pins.
- Constrain the transmitter SERDES logic to the LAB adjacent to the data output pins and clock output pins to improve the TCCS performance.

Related Information

[Quartus II Incremental Compilation for Hierarchical and Team-Based Design chapter, Volume 1: Design and Synthesis, Quartus II Handbook](#)

Provides step by step instructions about creating a design floorplan with LogicLock location assignments.

LVDS Transmitter Debug and Troubleshooting

You can obtain useful information about the LVDS interface performance with board-level verification using the FPGA prototype.

Although the focus of the board-level verification is to verify the FPGA functionality in your end system, you can take additional steps to examine the margins. Using oscilloscopes, you can examine the margins to verify the predicted size of the data-valid window, and the setup and hold margins at the I/O interface.

You can also use the Altera SignalTap® II Logic Analyzer to perform system level verification to correlate the system against your design targets.

Related Information

[In-System Debugging Using External Logic Analyzers chapter, Volume 3: Verification, Quartus II Handbook](#)

Perform RTL Simulation Before Hardware Debug

Before you debug on hardware, Altera recommends that you perform an RTL simulation. Using the RTL simulation, you can check the code functionality before testing in real hardware.

For example, you can use the RTL simulation to verify that when you send a training pattern from a remote transmitter, the bitslipping mechanism in your LVDS receiver works.

Geometry-Based and Physics-Based I/O Rules

You need to consider the I/O placement rules related to LVDS. The Quartus II software generates critical warning or error messages if the I/O placements rules are violated.

For more information, refer to the related information.

Related Information

[MAX 10 General Purpose I/O User Guide](#)

2014.12.15

UG-M10LVDS



Subscribe



Send Feedback

You can implement receiver-only applications using the MAX 10 LVDS solution. You can use the Altera Soft LVDS IP core to instantiate soft SERDES circuitry. The soft SERDES circuitry works with the clocks and differential I/O pins to create a high-speed differential receiver circuit.

Related Information

- [MAX 10 High-Speed LVDS I/O Design Overview](#) on page 1-1
- [MAX 10 LVDS SERDES I/O Standards Support](#) on page 2-7
Lists the supported LVDS I/O standards and the support in different MAX 10 device variants.

High-Speed I/O Receiver Circuitry

The LVDS receiver circuitry uses the I/O elements and registers in the MAX 10 devices. The deserializer is implemented in the core logic as a soft SERDES blocks.

In the receiver mode, the following blocks are available in the differential receiver datapath:

- Deserializer
- Data realignment block (bit slip)

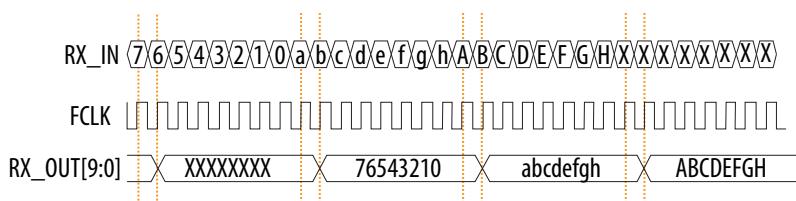
Related Information

[MAX 10 High-Speed LVDS Circuitry](#) on page 2-10

Soft Deserializer

The soft deserializer converts a 1-bit serial data stream into a parallel data stream based on the deserialization factor.

Figure 4-1: LVDS x8 Deserializer Waveform



© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered



Signal	Description
rx_in	LVDS data stream, input to the Altera Soft LVDS channel.
fclk	Clock used for receiver.
loaden	Enable signal for deserialization generated by the Altera Soft LVDS IP core.
rx_out[9:0]	Deserialized data.

Data Realignment Block (Bit Slip)

Skew in the transmitted data and skew added by the transmission link cause channel-to-channel skew on the received serial data streams. To compensate for channel-to-channel skew and establish the correct received word boundary at each channel, each receiver channel contains a data realignment circuit. The data realignment circuit realigns the data by inserting bit latencies into the serial stream.

To align the data manually, use the data realignment circuit to insert a latency of one $RxFCLK$ cycle. The data realignment circuit slips the data one bit for every RX_DATA_ALIGN pulse. You must wait at least two core clock cycles before checking to see if the data is aligned. This wait is necessary because it takes at least two core clock cycles to purge the corrupted data.

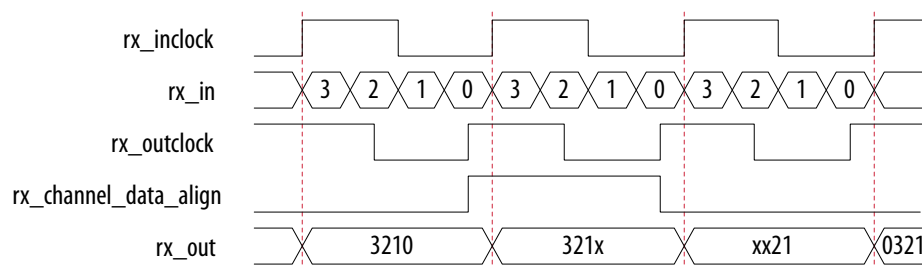
An optional $RX_CHANNEL_DATA_ALIGN$ port controls the bit insertion of each receiver independently of the internal logic. The data slips one bit on the rising edge of $RX_CHANNEL_DATA_ALIGN$.

The $RX_CHANNEL_DATA_ALIGN$ signal has these requirements:

- The minimum pulse width is one period of the parallel clock in the logic array.
- The minimum low time between pulses is one period of the parallel clock.
- The signal is edge-triggered.
- The valid data is available two parallel clock cycles after the rising edge of $RX_CHANNEL_DATA_ALIGN$.

Figure 4-2: Data Realignment Timing

This figure shows receiver output (RX_OUT) after one bit slip pulse with the deserialization factor set to 4.



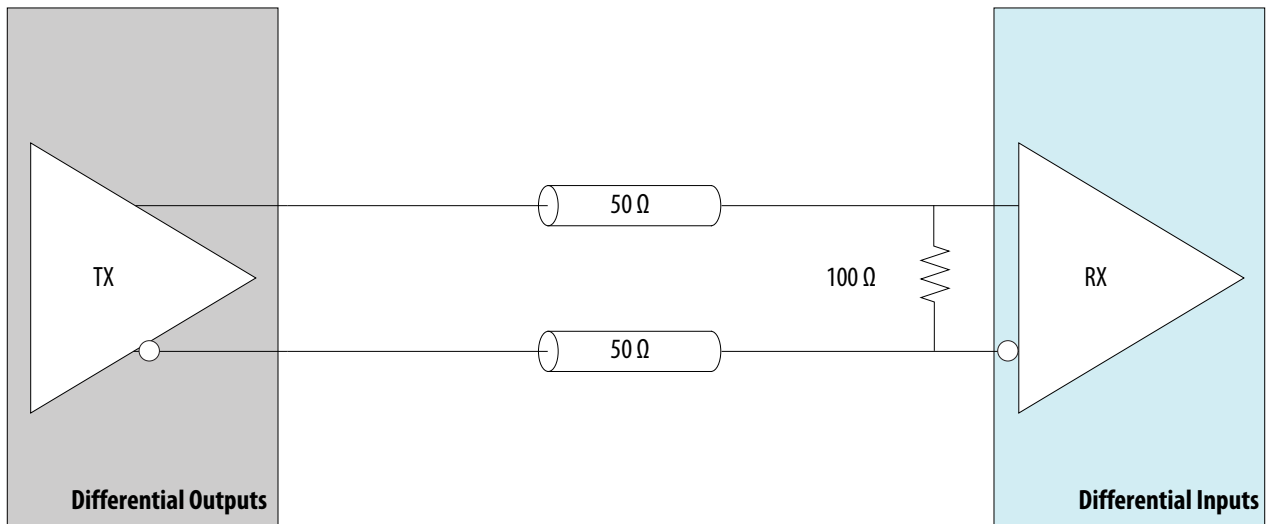
LVDS Receiver I/O Termination Schemes

All LVDS receiver channels require termination to achieve better signal quality and ensure impedance matching with the transmission line and driver.

LVDS, Mini-LVDS, and RSDS Receiver External Termination

The LVDS, mini-LVDS, or RSDS receiver requires a single resistor external termination scheme.

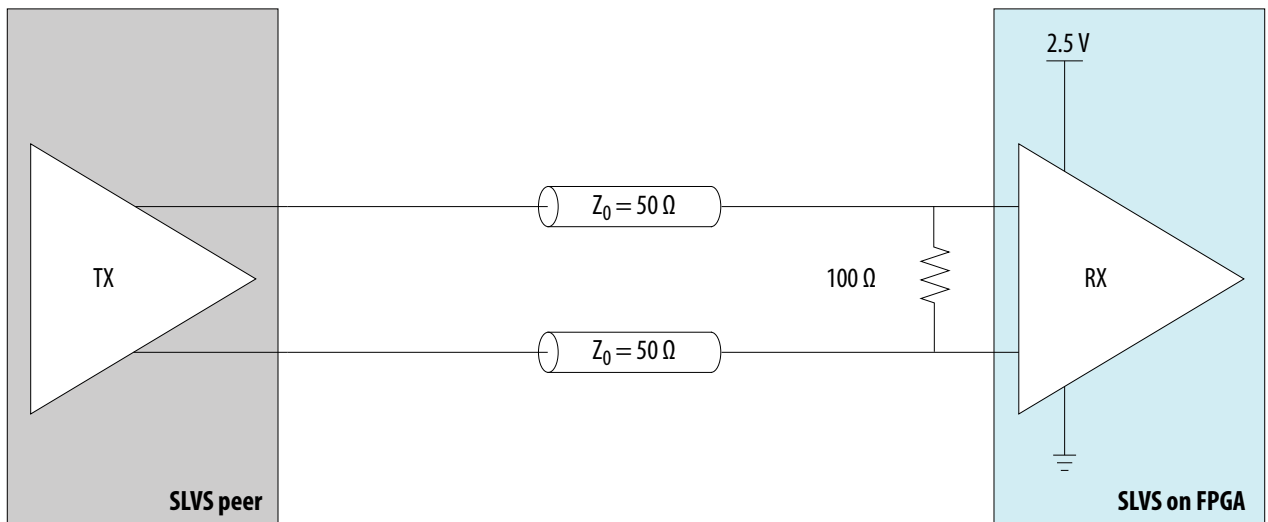
Figure 4-3: External Termination for LVDS I/O Standard



SLVS Receiver External Termination

The SLVS receiver requires a single-resistor external termination scheme.

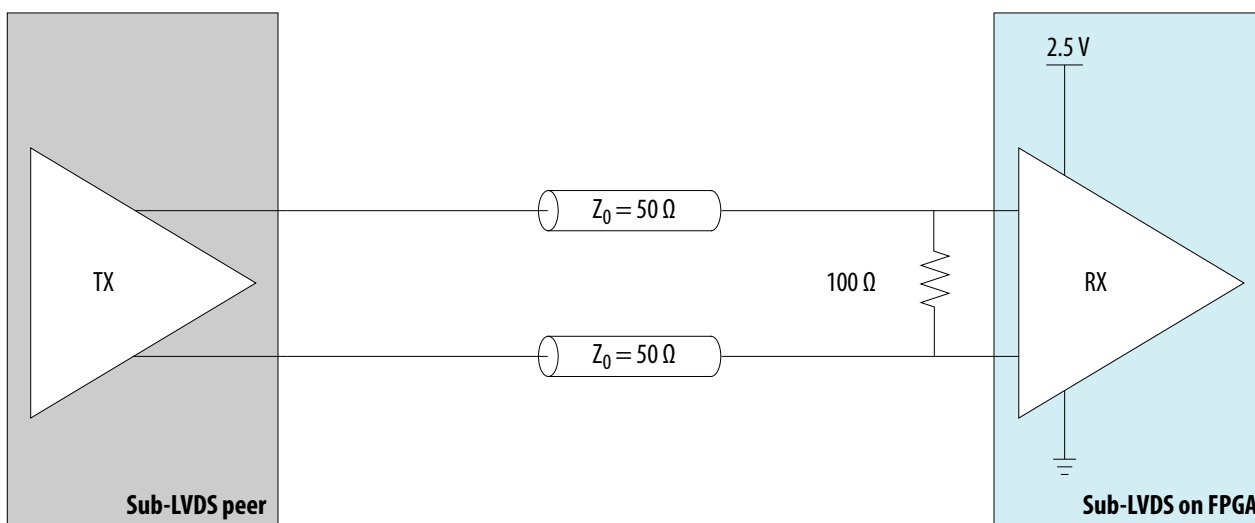
Figure 4-4: External Termination for SLVS Receiver



Sub-LVDS Receiver External Termination

The Sub-LVDS receiver requires a single-resistor external termination scheme.

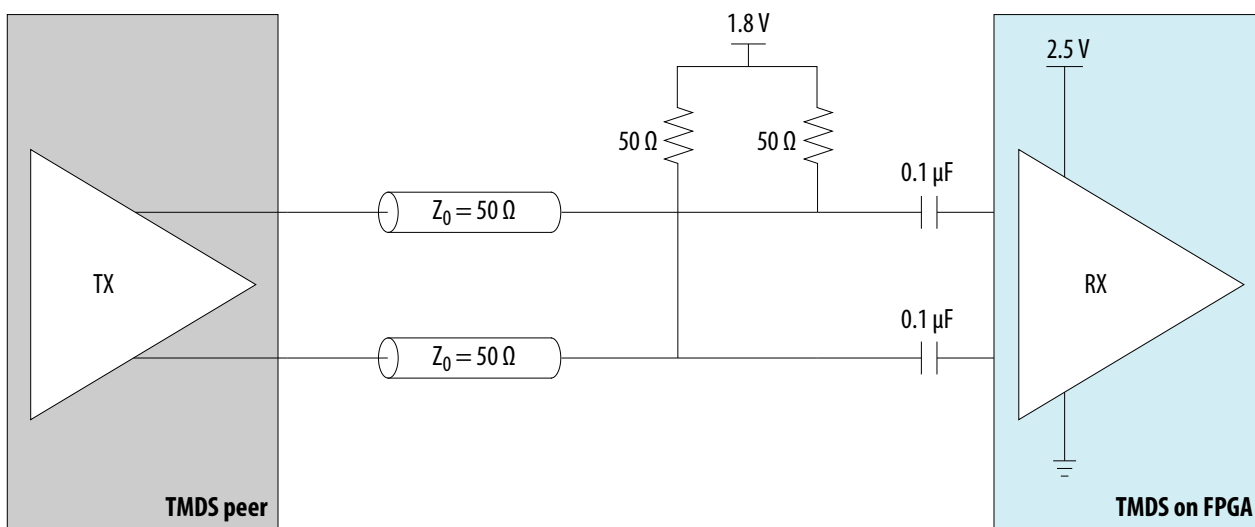
Figure 4-5: External Termination for Sub-LVDS Receiver



TMDS Receiver External Termination

Figure 4-6: External Termination for TMDS Receiver

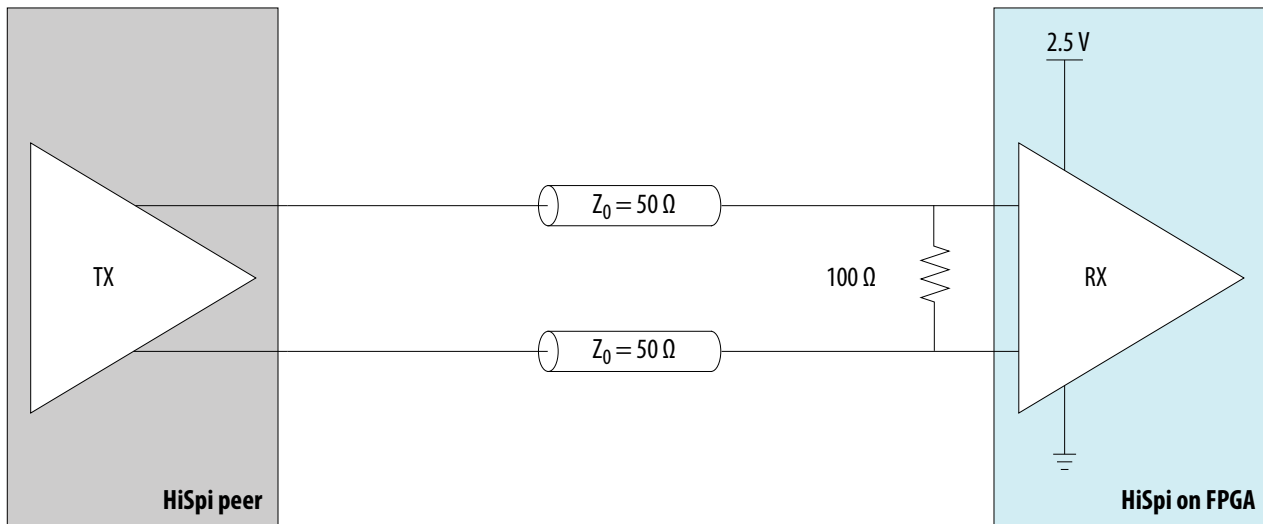
This diagram shows the external level shifter that is required for the TMDS input standards support in MAX 10 devices.



HiSpi Receiver External Termination

The HiSpi receiver requires a single-resistor external termination scheme.

Figure 4-7: External Termination for HiSpi Receiver



LVPECL External Termination

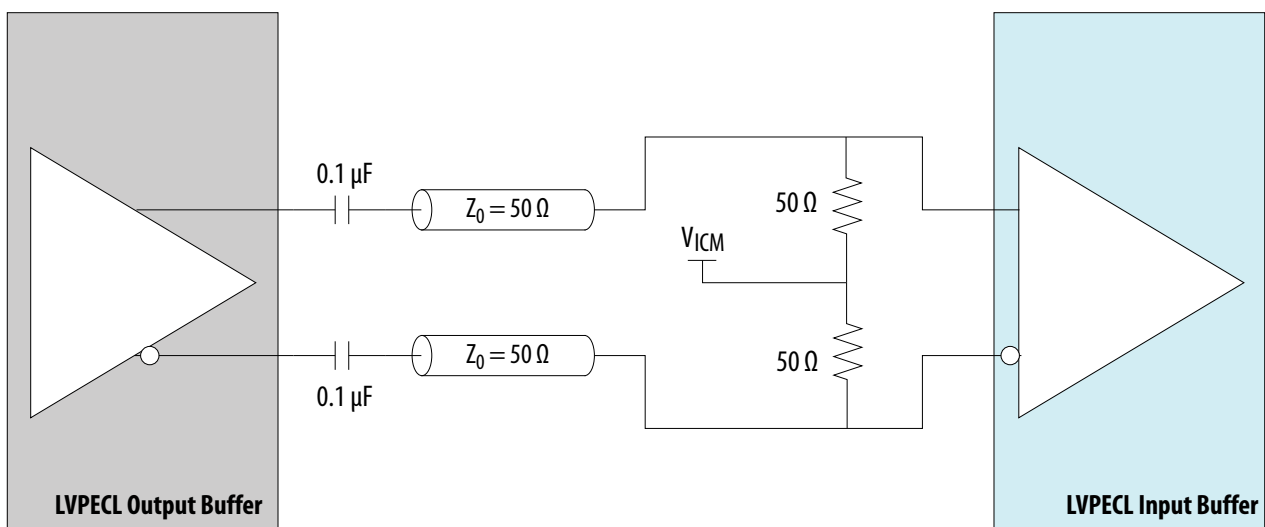
The MAX 10 devices support the LVPECL I/O standard on input clock pins only.

- LVDS input buffers support LVPECL input operation.
- LVPECL output operation is not supported.

Use AC coupling if the LVPECL common-mode voltage of the output buffer does not match the LVPECL input common-mode voltage.

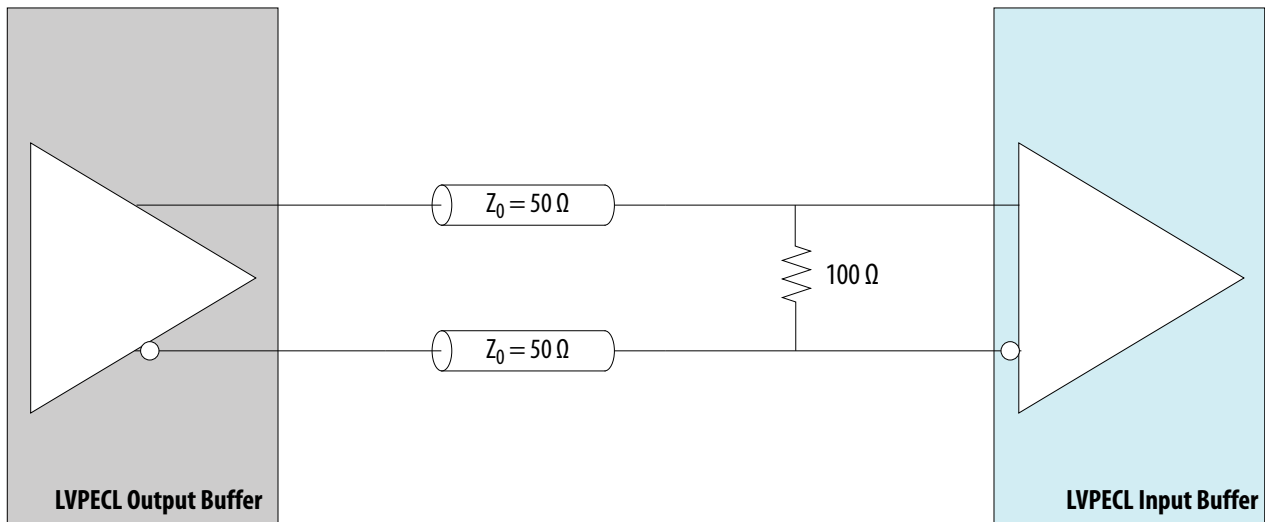
Note: Altera recommends that you use IBIS models to verify your LVPECL AC/DC-coupled termination.

Figure 4-8: LVPECL AC-Coupled Termination



Support for DC-coupled LVPECL is available if the LVPECL output common mode voltage is within the MAX 10 LVPECL input buffer specification.

Figure 4-9: LVPECL DC-Coupled Termination



For information about the V_{ICM} specification, refer to the device datasheet.

Related Information

[MAX 10 Device Datasheet](#)

LVDS Receiver FPGA Design Implementation

MAX 10 devices use a soft SERDES architecture to support high-speed I/O interfaces. The Quartus II software creates the SERDES circuits in the core fabric by using the Altera Soft LVDS IP core. To improve the timing performance and support the SERDES, MAX 10 devices use the I/O registers and LE registers in the core fabric.

Altera Soft LVDS IP Core in Receiver Mode

In the Quartus II software, you can design your high-speed receiver interfaces using the Altera Soft LVDS IP core. This IP core uses the resources in the MAX 10 devices optimally to create the high-speed I/O interfaces.

- You can use the Altera Soft LVDS parameter editor to customize your deserializer based on your design requirements.
- The Altera Soft LVDS IP core implements the high-speed deserializer in the core fabric.

Related Information

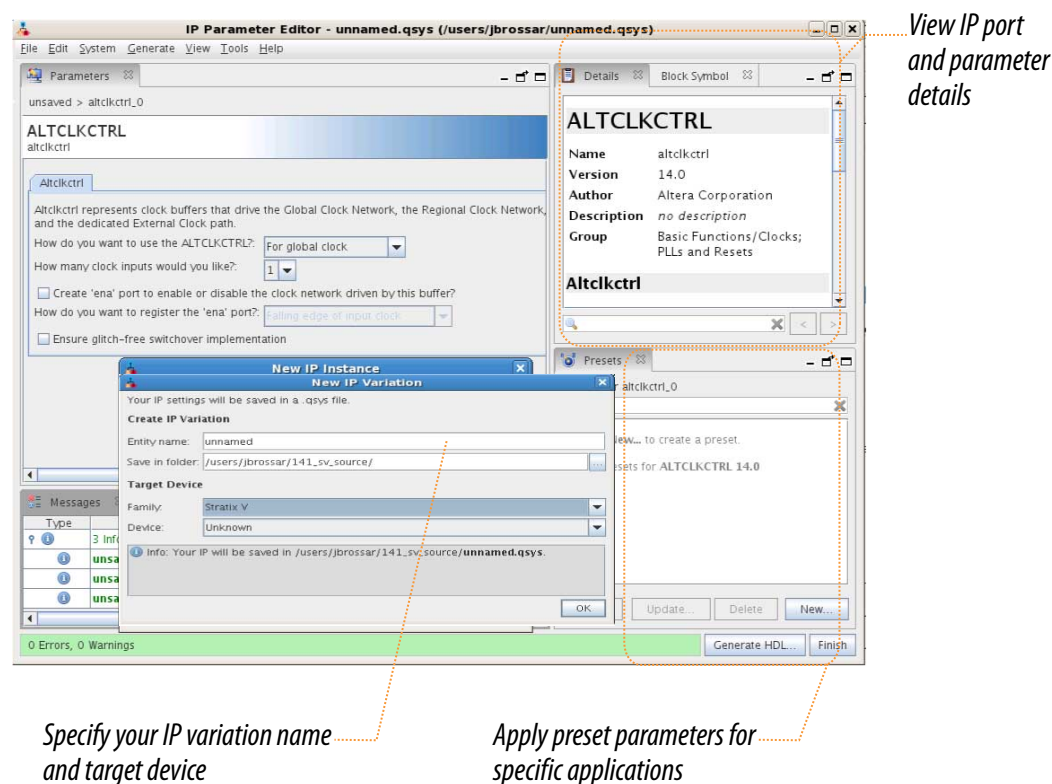
[Altera Soft LVDS Parameter Settings](#) on page 7-1

Specifying IP Core Parameters and Options

You can quickly configure a custom IP variation in the parameter editor. Use the following steps to specify IP core options and parameters in the parameter editor. Refer to *Specifying IP Core Parameters and Options (Legacy Parameter Editors)* for configuration of IP cores using the legacy parameter editor.

1. In the IP Catalog (**Tools > IP Catalog**), locate and double-click the name of the IP core to customize. The parameter editor appears.
2. Specify a top-level name for your custom IP variation. The parameter editor saves the IP variation settings in a file named *<your_ip>.qsys*. Click **OK**.
3. Specify the parameters and options for your IP variation in the parameter editor, including one or more of the following. Refer to your IP core user guide for information about specific IP core parameters.
 - Optionally select preset parameter values if provided for your IP core. Presets specify initial parameter values for specific applications.
 - Specify parameters defining the IP core functionality, port configurations, and device-specific features.
 - Specify options for processing the IP core files in other EDA tools.
4. Click **Generate HDL**, the **Generation** dialog box appears.
5. Specify output file generation options, and then click **Generate**. The IP variation files generate according to your specifications.
6. To generate a simulation testbench, click **Generate > Generate Testbench System**.
7. To generate an HDL instantiation template that you can copy and paste into your text editor, click **Generate > HDL Example**.
8. Click **Finish**. The parameter editor adds the top-level *.qsys* file to the current project automatically. If you are prompted to manually add the *.qsys* file to the project, click **Project > Add/Remove Files in Project** to add the file.
9. After generating and instantiating your IP variation, make appropriate pin assignments to connect ports.

Figure 4-10: IP Parameter Editor



Related Information

[Altera Soft LVDS Parameter Settings](#) on page 7-1

PLL Source Selection for Altera Soft LVDS IP Core

You can create the LVDS interface components by instantiating the Altera Soft LVDS IP core with an internal or external PLL.

Instantiate Altera Soft LVDS IP Core with Internal PLL

You can set the Altera Soft LVDS IP core to build the SERDES components and instantiate the PLL internally.

- To use this method, turn off the **Use external PLL** option in the **PLL Settings** tab.
- The Altera Soft LVDS IP core integrates the PLL into the LVDS block.
- The drawback of this method is that you can use the PLL only for the particular LVDS instance.

Instantiate Altera Soft LVDS IP Core with External PLL

You can set the Altera Soft LVDS IP core to build only the SERDES components but use an external PLL source.

- To use this method, turn on the **Use external PLL** option in the **PLL Settings** tab.
- Follow the required clock setting to the input ports as listed in the notification panel.
- You can create your own clocking source using the ALTPLL IP core.
- Use this method to optimize PLL usage with other functions in the core.

Related Information

- [MAX 10 Clocking and PLL User Guide](#)
Provides more information about the PLL and the PLL output counters.
- [MAX 10 Clocking and PLL User Guide](#)
Provides more information about the PLL and the PLL output counters.

Guidelines: LVDS RX Interface Using External PLL

You can instantiate the Altera Soft LVDS IP core with the **Use External PLL** option. Using external PLL, you can control the PLL settings. For example, you can dynamically reconfigure the PLL to support different data rates and dynamic phase shifts. To use this option, you must instantiate the ALTPLL IP core to generate the various clock signals.

If you turn on the **Use External PLL** option for the Altera Soft LVDS receiver, you require the following signals from the ALTPLL IP core:

- Serial clock input to the rx_inclock port of the Altera Soft LVDS transmitter.
- Parallel clock used to clock the transmitter FPGA fabric logic.
- The locked signal for Altera Soft LVDS PLL reset port.

Related Information[MAX 10 Clocking and PLL User Guide](#)

Provides more information about the PLL and the PLL output counters.

ALTPLL Signal Interface with Altera Soft LVDS Receiver

You can choose any of the PLL output clock ports to generate the LVDS interface clocks.

If you use the ALTPLL IP core as the external PLL source of the Altera Soft LVDS transmitter, use the source-synchronous compensation mode.

Table 4-1: Example: Signal Interface Between ALTPLL and Altera Soft LVDS Receiver with Even Deserialization Factor

From the ALTPLL IP Core	To the Altera Soft LVDS Receiver
Fast clock output (c0) The serial clock output (c0) can only drive rx_inclock on the Altera Soft LVDS receiver.	rx_inclock
Slow clock output (c1)	rx_syncclock

Table 4-2: Example: Signal Interface Between ALTPLL and Altera Soft LVDS Receiver with Odd Deserialization Factor

From the ALTPLL IP Core	To the Altera Soft LVDS Receiver
Fast clock output (c0) The serial clock output (c0) can only drive rx_inclock on the Altera Soft LVDS receiver.	rx_inclock

From the ALTPLL IP Core	To the Altera Soft LVDS Receiver
Slow clock output (c1)	rx_syncclock
Read clock (c2) output from the PLL	rx_readclock (clock input port for reading operation from RAM buffer and read counter)

Determining External PLL Clock Parameters for Altera Soft LVDS Receiver

To determine the ALTPLL IP core clock parameter for the Altera Soft LVDS IP core receiver, follow these steps in your design:

1. Instantiate the Altera Soft LVDS IP core receiver using internal PLL.
2. Compile the design up to TimeQuest timing analysis.
3. In the **Table of Contents** section of the **Compilation Report** window, navigate to **TimeQuest Timing Analyzer > Clocks**.
4. Note the clock parameters used by the internal PLL for the Altera Soft LVDS IP core receiver.
In the list of clocks, clk0 is the fast clock.

Figure 4-11: Clock Parameters Example for Altera Soft LVDS Receiver

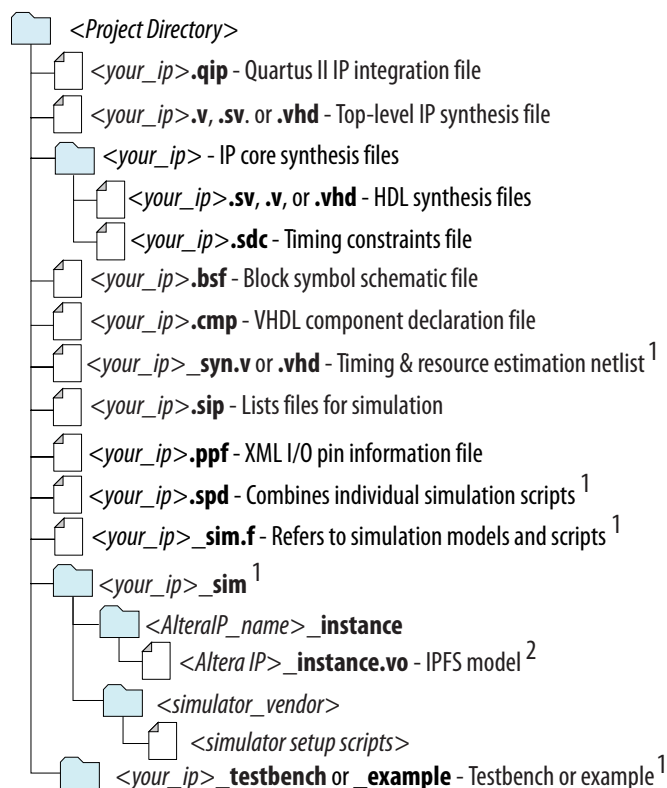
Clocks						
	Clock Name	Type	Period	Frequency	Duty Cycle	Phase
1	inst lvdsrx_inst lvds_rx_pll clk[0]	Generated	2.500	400.0 MHz	50.00	-90.0
2	inst lvdsrx_inst lvds_rx_pll clk[1]	Generated	12.500	80.0 MHz	50.00	-18.0
3	inst lvdsrx_inst lvds_rx_pll clk[2]	Generated	6.250	160.0 MHz	50.00	-36.0
4	rx_inclock	Base	10.000	100.0 MHz		

Configure the ALTPLL output clocks with the parameters you noted in this procedure and connect the clock outputs to the correct Altera Soft LVDS clock input ports.

Files Generated for Altera IP Cores (Legacy Parameter Editor)

The Quartus II software version 14.0 and previous generates the following output for IP cores that use the legacy MegaWizard parameter editor.

Figure 4-12: IP Core Generated Files

**Notes:**

1. If supported and enabled for your IP variation
2. If functional simulation models are generated

Initializing the Altera Soft LVDS IP Core

The PLL locks to the reference clock before the Altera Soft LVDS IP core implements the SERDES blocks for data transfer.

During device initialization the PLL starts to lock to the reference clock and becomes operational when it achieves lock during user mode. If the clock reference is not stable, it corrupts the phase shifts of the PLL output clock. This phase shifts corruption can cause failure and corrupt data transfer between the high-speed LVDS domain and the low-speed parallel domain.

To avoid data corruption, follow these steps when initializing the Altera Soft LVDS IP core:

1. Assert the `pll_areset` signal for at least 10 ns.
2. After at least 10 ns, deassert the `pll_areset` signal.
3. Wait until the PLL lock becomes stable.

After the PLL lock port asserts and is stable, the SERDES blocks are ready for operation.

High-Speed I/O Timing Budget

The LVDS I/O standard enables high-speed transmission of data that results in better overall system performance. To take advantage of fast system performance, you must analyze the timing for high-speed

signals. Timing analysis for the differential block is different than traditional synchronous timing analysis techniques.

The basis of the source synchronous timing analysis is the skew between the data and the clock signals instead of the clock-to-output setup times. High-speed differential data transmission requires the use of timing parameters provided by IC vendors and is strongly influenced by board skew, cable skew, and clock jitter.

Receiver Input Skew Margin

Use RSKM, TCCS, and sampling window (SW) specifications for high-speed source-synchronous differential signals in the receiver data path.

Figure 4-13: RSKM Equation

$$RSKM = \frac{TUI - SW - TCCS}{2}$$

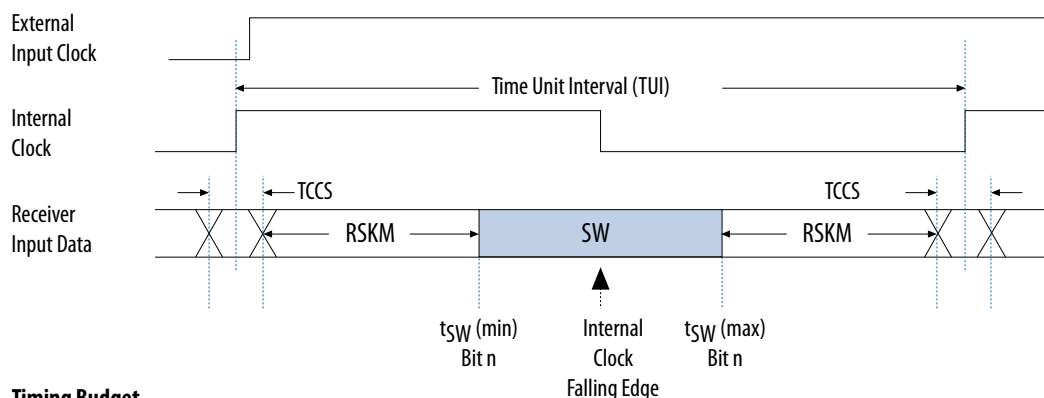
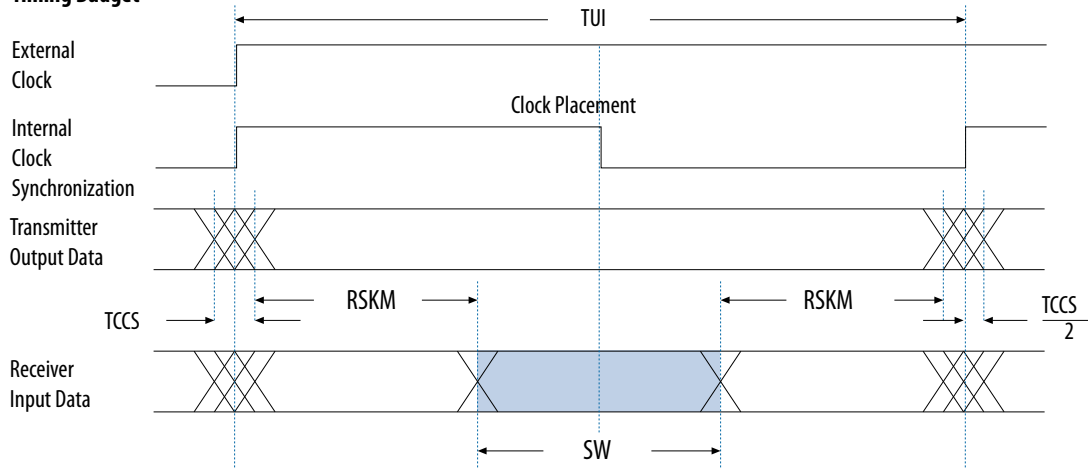
Conventions used for the equation:

- RSKM—the timing margin between the clock input of the receiver and the data input sampling window, and the jitter induced from core noise and I/O switching noise.
- Time unit interval (TUI)—time period of the serial data.
- SW—the period of time that the input data must be stable to ensure that the LVDS receiver samples the data successfully. The SW is a device property and varies according to device speed grade.
- TCCS—the timing difference between the fastest and the slowest output edges across channels driven by the same PLL. This measurement includes the t_{CO} variation, clock, and clock skew.

You must calculate the RSKM value, based on the data rate and device, to determine if the LVDS receiver can sample the data:

- A positive RSKM value, after deducting transmitter jitter, indicates that the LVDS receiver can sample the data properly.
- A negative RSKM, after deducting transmitter jitter, indicates that it cannot sample the data properly.

Figure 4-14: Differential High-Speed Timing Diagram and Timing Budget

Timing Diagram**Timing Budget****RSKM Report for LVDS Receiver**

For LVDS receivers, the Quartus II software provides an RSKM report showing the SW, TUI, and RSKM values.

- To generate the RSKM report, run the `report_RSKM` command in the TimeQuest Timing Analyzer. The RSKM report is available in the TimeQuest Timing Analyzer section of the Quartus II compilation report.
- To obtain a more realistic RSKM value, assign the input delay to the LVDS receiver through the constraints menu of the TimeQuest Timing Analyzer. The input delay is determined according to the data arrival time at the LVDS receiver port, with respect to the reference clock.
- If you set the input delay in the settings parameters for the **Set Input Delay** option, set the clock name to the clock that references the source-synchronous clock that feeds the LVDS receiver.
- If you do not set any input delay in the TimeQuest Timing Analyzer, the receiver channel-to-channel skew defaults to zero.
- You can also directly set the input delay in a Synopsys Design Constraint file (.sdc) by using the `set_input_delay` command.

Example: RSKM Calculation

This example shows the RSKM calculation for FPGA devices at 1 Gbps data rate with a 200 ps board channel-to-channel skew.

- $TCCS = 100 \text{ ps}$
- $SW = 300 \text{ ps}$
- $TUI = 1000 \text{ ps}$
- $\text{Total } RCCS = TCCS + \text{Board channel-to-channel skew} = 100 \text{ ps} + 200 \text{ ps} = 300 \text{ ps}$
- $RSKM = (TUI - SW - RCCS) / 2 = (1000 \text{ ps} - 300 \text{ ps} - 300 \text{ ps}) / 2 = 200 \text{ ps}$

If the RSKM is greater than 0 ps after deducting transmitter jitter, the receiver will work correctly.

Guidelines: Floating LVDS Input Pins

You can implement floating LVDS input pins in MAX 10 devices.

For floating LVDS input pins, apply a 100 Ω differential resistance across the P and N legs of the LVDS receiver. You can use external termination.

If you use floating LVDS input pins, Altera recommends that you use external biasing schemes to reduce noise injection and current consumption.

Guidelines: LVDS Receiver Channels Placement

To maintain an acceptable noise level on the V_{CCIO} supply, observe the placement restrictions for single-ended I/O pins in relation to differential pads.

Altera recommends that you create a Quartus II design, specify your device I/O assignments, and compile your design to validate your pin placement. The Quartus II software verifies your pin connections against the I/O assignment and placement rules to ensure that the device will operate properly.

You can use the Quartus II Pin Planner Package view to ease differential I/O assignment planning:

- On the **View** menu, click **Show Differential Pin Pair Connections** to highlight the differential pin pairing. The differential pin pairs are connected with red lines.
- For differential pins, you only need to assign the signal to a positive pin. The Quartus II software automatically assigns the negative pin if the positive pin is assigned with a differential I/O standard.

In MAX 10 devices, the routing of each differential pin pair is matched. Consequently, the skew between the positive and the negative pins is minimal. The internal routings of both pins in a differential pair are matched even if the pins are non-adjacent.

MAX 10 support x18 bundling mode. To control skew, place all LVDS channels in the same bundle side by side and the channels must not cross I/O banks. In your PCB design, Altera recommends package skew compensation to minimize skew and maximize performance.

Guidelines: LVDS Channels PLL Placement

Each PLL in the MAX 10 device can drive only the LVDS channels in I/O banks on the same edge as the PLL.

Table 4-3: Examples of Usable PLL to Drive I/O Banks in MAX 10 Devices

I/O Bank Edge	Input refclk	GCLK mux	Usable PLL
Left	Left	Left	Top left or bottom left

I/O Bank Edge	Input refclk	GCLK mux	Usable PLL
Bottom	Bottom	Bottom	Bottom left or bottom right
Right	Right	Right	Top right or bottom right
Top	Top	Top	Top left or top right

Guidelines: LVDS Receiver Logic Placement

The Quartus II software automatically optimizes the SERDES logic placement to meet the timing requirements. Therefore, you do not have to perform placement constraints on the Altera Soft LVDS IP core logic.

To improve the performance of the Quartus II Fitter, you can create LogicLock regions in the device floorplan to confine the transmitter SERDES logic placement.

- The TCCS parameter is guaranteed per datasheet specification to the entire bank of differential I/Os that are located in the same side. This guarantee applies if the transmitter SERDES logic is placed within the LAB adjacent to the output pins.
- Constrain the transmitter SERDES logic to the LAB adjacent to the data output pins and clock output pins to improve the TCCS performance.

Guidelines: LVDS Receiver Timing Constraints

For receiver designs that uses the core logic to implement the SERDES circuits, you must set proper timing constraints.

For LVDS receiver data paths where the PLL operation is in source-synchronous compensation mode, the Quartus II compiler automatically ensures that the associated delay chain settings are set correctly.

However, if the input clock and data at the receiver are not edge- or center-aligned, it may be necessary for you to set the timing constraints in the Quartus II TimeQuest Timing Analyzer. The timing constraints specify the timing requirements necessary to ensure reliable data capture.

LVDS Receiver Debug and Troubleshooting

You can obtain useful information about the LVDS interface performance with board-level verification using the FPGA prototype.

Although the focus of the board-level verification is to verify the FPGA functionality in your end system, you can take additional steps to examine the margins. Using oscilloscopes, you can examine the margins to verify the predicted size of the data-valid window, and the setup and hold margins at the I/O interface.

You can also use the Altera SignalTap II Logic Analyzer to perform system level verification to correlate the system against your design targets.

Perform RTL Simulation Before Hardware Debug

Before you debug on hardware, Altera recommends that you perform an RTL simulation. Using the RTL simulation, you can check the code functionality before testing in real hardware.

For example, you can use the RTL simulation to verify that when you send a training pattern from a remote transmitter, the bitslipping mechanism in your LVDS receiver works.

Geometry-Based and Physics-Based I/O Rules

You need to consider the I/O placement rules related to LVDS. The Quartus II software generates critical warning or error messages if the I/O placements rules are violated.

For more information, refer to the related information.

Related Information

[MAX 10 General Purpose I/O User Guide](#)

2014.12.15

UG-M10LVDS



Subscribe



Send Feedback

You can implement mixed transmitter and receiver applications using the MAX 10 LVDS solution. You can use the Altera Soft LVDS IP core to instantiate soft SERDES circuitry. The soft SERDES circuitry works with the clocks and differential I/O pins to create high-speed differential transmitter and receiver circuits.

In a mixed transmitter and receiver implementation, the transmitter and receiver can share some FPGA resources.

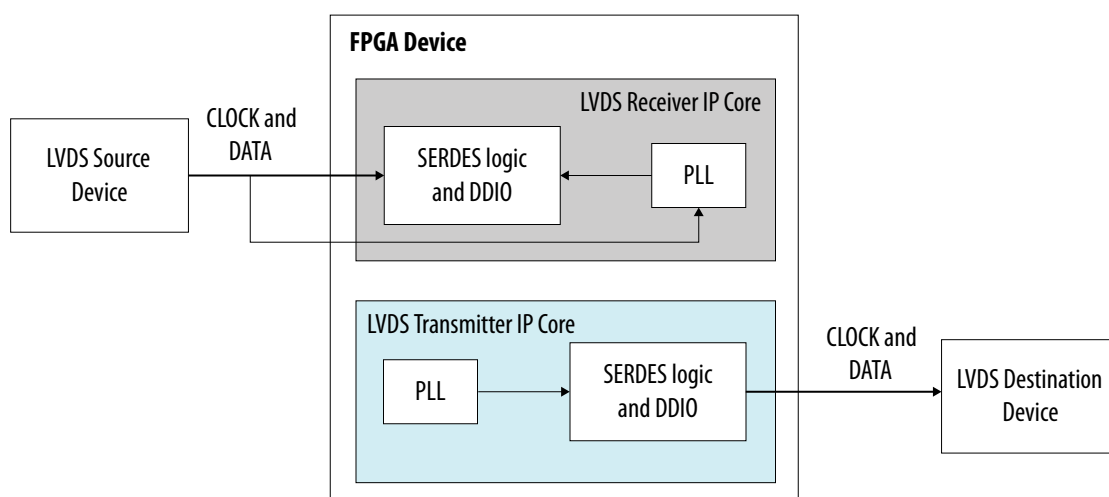
Related Information

- [MAX 10 High-Speed LVDS I/O Design Overview](#) on page 1-1
 - [MAX 10 LVDS SERDES I/O Standards Support](#) on page 2-7
- Lists the supported LVDS I/O standards and the support in different MAX 10 device variants.

Transmitter–Receiver Interfacing

You can instantiate the components for the Altera Soft LVDS interfaces by using internal or external PLLs.

Figure 5-1: Typical Altera Soft LVDS Interfaces with Internal PLL

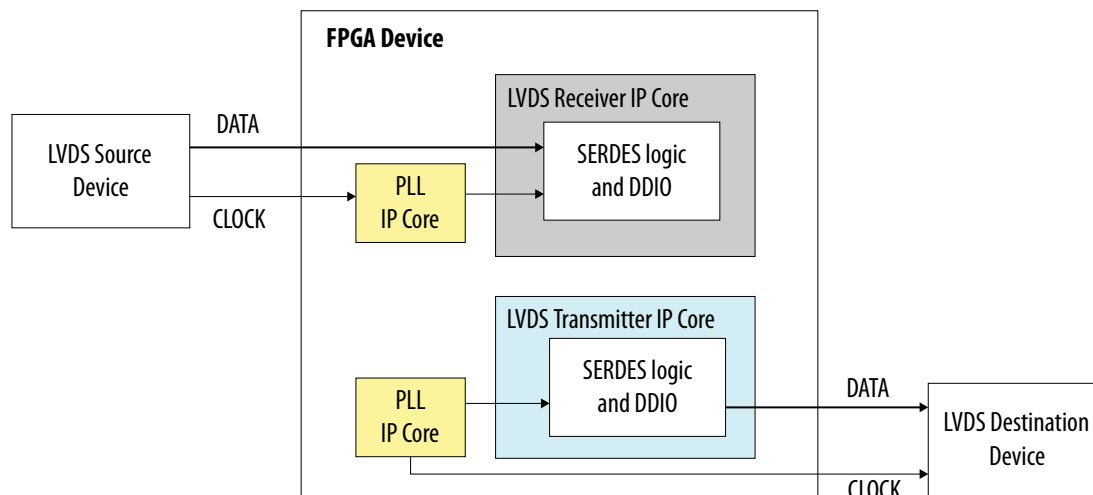


© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered



Figure 5-2: Typical Altera Soft LVDS Interfaces with External PLL

**Related Information**

- [MAX 10 LVDS Transmitter Design](#) on page 3-1
Provides more information about specific features and support of the LVDS transmitters.
- [MAX 10 LVDS Receiver Design](#) on page 4-1
Provides more information about specific features and support of the LVDS receivers.

LVDS Transmitter and Receiver FPGA Design Implementation

MAX 10 devices use a soft SERDES architecture to support high-speed I/O interfaces. The Quartus II software creates the SERDES circuits in the core fabric by using the Altera Soft LVDS IP core. To improve the timing performance and support the SERDES, MAX 10 devices use the I/O registers and LE registers in the core fabric.

LVDS Transmitter and Receiver PLL Sharing Implementation

In applications where an LVDS transmitter and receiver are required, you typically need two PLLs—one for each interface. Using the Altera Soft LVDS IP core, you can reduce PLL usage by sharing one PLL between the transmitter and receiver.

- Turn on the **Use common PLL(s) for receivers and transmitters** option to allow the Quartus II compiler to share the same PLL.
- To share a PLL, several PLLs must have the same PLL settings, such as PLL feedback mode, clock frequency, and phase settings. The LVDS transmitters and receivers must use the same input clock frequency and reset input.
- If you are sharing a PLL, you can use more counters to enable different deserialization factor and data rates for the transmitters and receivers. However, because you are using more PLL counters, the PLL input clock frequency and the PLL counter resolution cause limitations in clocking the transmitters and receivers.

Note: The number of PLLs available differs among MAX 10 packages. Altera recommends that you select a MAX 10 device package that provides sufficient number of PLL clockouts for your design.

Initializing the Altera Soft LVDS IP Core

The PLL locks to the reference clock before the Altera Soft LVDS IP core implements the SERDES blocks for data transfer.

During device initialization the PLL starts to lock to the reference clock and becomes operational when it achieves lock during user mode. If the clock reference is not stable, it corrupts the phase shifts of the PLL output clock. This phase shifts corruption can cause failure and corrupt data transfer between the high-speed LVDS domain and the low-speed parallel domain.

To avoid data corruption, follow these steps when initializing the Altera Soft LVDS IP core:

1. Assert the `pll_areset` signal for at least 10 ns.
2. After at least 10 ns, deassert the `pll_areset` signal.
3. Wait until the PLL lock becomes stable.

After the PLL lock port asserts and is stable, the SERDES blocks are ready for operation.

LVDS Transmitter and Receiver Debug and Troubleshooting

You can obtain useful information about the LVDS interface performance with board-level verification using the FPGA prototype.

Although the focus of the board-level verification is to verify the FPGA functionality in your end system, you can take additional steps to examine the margins. Using oscilloscopes, you can examine the margins to verify the predicted size of the data-valid window, and the setup and hold margins at the I/O interface.

You can also use the Altera SignalTap II Logic Analyzer to perform system level verification to correlate the system against your design targets.

Perform RTL Simulation Before Hardware Debug

Before you debug on hardware, Altera recommends that you perform an RTL simulation. Using the RTL simulation, you can check the code functionality before testing in real hardware.

For example, you can use the RTL simulation to verify that when you send a training pattern from a remote transmitter, the bitslipping mechanism in your LVDS receiver works.

Geometry-Based and Physics-Based I/O Rules

You need to consider the I/O placement rules related to LVDS. The Quartus II software generates critical warning or error messages if the I/O placements rules are violated.

For more information, refer to the related information.

Related Information

[MAX 10 General Purpose I/O User Guide](#)

2014.12.15

UG-M10LVDS



Subscribe



Send Feedback

To achieve optimal performance from the MAX 10 device, you must consider critical issues such as impedance of traces and connectors, differential routing, and termination techniques.

Guideline: Improve Signal Quality

To improve signal quality, follow these board design guidelines:

- Base your board designs on controlled differential impedance. Calculate and compare all parameters such as trace width, trace thickness, and the distance between two differential traces.
- Maintain equal distance between traces in differential I/O standard pairs as much as possible. Routing the pair of traces close to each other maximizes the common-mode rejection ratio (CMRR).
- Keep the traces as short as possible to limit signal integrity issues. Longer traces have more inductance and capacitance.
- Place termination resistors as close to receiver input pins as possible.
- Use surface mount components.
- Avoid 90° corners on board traces.
- Use high-performance connectors.
- Design backplane and card traces so that trace impedance matches the impedance of the connector and termination.
- Keep an equal number of vias for both signal traces.
- Create equal trace lengths to avoid skew between signals. Unequal trace lengths result in misplaced crossing points and decrease system margins as the transmitter-channel-to-channel skew (TCCS) value increases.
- Limit vias because they cause discontinuities.
- Keep toggling single-ended I/O signals away from differential signals to avoid possible noise coupling.
- Do not route single-ended I/O clock signals to layers adjacent to differential signals.
- Analyze system-level signals.

Guidelines: Control Channel-to-Channel Skew

For the MAX 10 devices, perform PCB trace compensation to adjust the trace length of each LVDS channel. Adjusting the trace length improves the channel-to-channel skew when interfacing with

© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered



receivers. The Quartus II software Fitter Report panel reports the amount of delay you must add to each trace for the MAX 10 device.

You can use the recommended trace delay numbers published under the LVDS Transmitter/Receiver Package Skew Compensation panel and manually compensate the skew on the PCB board trace. This skew compensation reduces the channel-to-channel skew so you can comply with the timing budget between LVDS channels.

At the package level, you must control the LVDS I/O skew for each I/O bank and each side of the device. If you plan to vertically migrate from one device to another using the same board design, you must control the package migration skew for each migratable LVDS I/O pin.

Receiver Input Skew Margin

Use RSKM, TCCS, and sampling window (SW) specifications for high-speed source-synchronous differential signals in the receiver data path.

Figure 6-1: RSKM Equation

$$RSKM = \frac{TUI - SW - TCCS}{2}$$

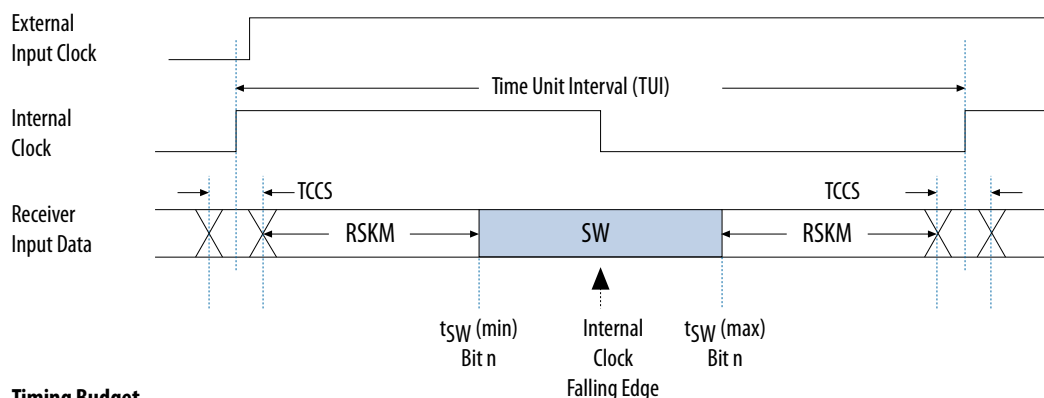
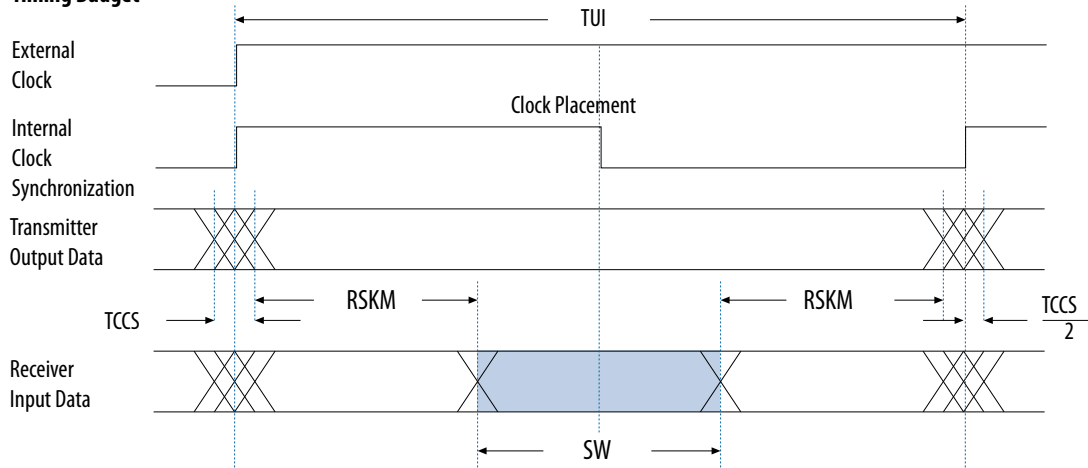
Conventions used for the equation:

- RSKM—the timing margin between the clock input of the receiver and the data input sampling window, and the jitter induced from core noise and I/O switching noise.
- Time unit interval (TUI)—time period of the serial data.
- SW—the period of time that the input data must be stable to ensure that the LVDS receiver samples the data successfully. The SW is a device property and varies according to device speed grade.
- TCCS—the timing difference between the fastest and the slowest output edges across channels driven by the same PLL. This measurement includes the t_{CO} variation, clock, and clock skew.

You must calculate the RSKM value, based on the data rate and device, to determine if the LVDS receiver can sample the data:

- A positive RSKM value, after deducting transmitter jitter, indicates that the LVDS receiver can sample the data properly.
- A negative RSKM, after deducting transmitter jitter, indicates that it cannot sample the data properly.

Figure 6-2: Differential High-Speed Timing Diagram and Timing Budget

Timing Diagram**Timing Budget****RSKM Report for LVDS Receiver**

For LVDS receivers, the Quartus II software provides an RSKM report showing the SW, TUI, and RSKM values.

- To generate the RSKM report, run the `report_RSKM` command in the TimeQuest Timing Analyzer. The RSKM report is available in the TimeQuest Timing Analyzer section of the Quartus II compilation report.
- To obtain a more realistic RSKM value, assign the input delay to the LVDS receiver through the constraints menu of the TimeQuest Timing Analyzer. The input delay is determined according to the data arrival time at the LVDS receiver port, with respect to the reference clock.
- If you set the input delay in the settings parameters for the **Set Input Delay** option, set the clock name to the clock that references the source-synchronous clock that feeds the LVDS receiver.
- If you do not set any input delay in the TimeQuest Timing Analyzer, the receiver channel-to-channel skew defaults to zero.
- You can also directly set the input delay in a Synopsys Design Constraint file (.sdc) by using the `set_input_delay` command.

Example: RSKM Calculation

This example shows the RSKM calculation for FPGA devices at 1 Gbps data rate with a 200 ps board channel-to-channel skew.

- $TCCS = 100 \text{ ps}$
- $SW = 300 \text{ ps}$
- $TUI = 1000 \text{ ps}$
- $\text{Total } RCCS = TCCS + \text{Board channel-to-channel skew} = 100 \text{ ps} + 200 \text{ ps} = 300 \text{ ps}$
- $RSKM = (TUI - SW - RCCS) / 2 = (1000 \text{ ps} - 300 \text{ ps} - 300 \text{ ps}) / 2 = 200 \text{ ps}$

If the RSKM is greater than 0 ps after deducting transmitter jitter, the receiver will work correctly.

Guidelines: Determine Board Design Constraints

After you have closed timing for your FPGA design, examine your board design to determine the different factors that can impact signal integrity. These factors affect overall timing at the receiving device in the LVDS interface.

The time margin for the LVDS receiver (indicated by the RSKM value) is the timing budget allocation for board level effects such as:

- Skew—these factors cause board-level skew:
 - Board trace lengths
 - Connectors usage
 - Parasitic circuits variations
- Jitter—jitter effects are derived from factors such as crosstalk.
- Noise—on board resources with imperfect power supplies and reference planes may also cause noise.

To ensure successful operation of the Altera Soft LVDS IP core receiver, do not exceed the timing budget.

Related Information

[Board Design Guidelines Solution Center](#)

Provides resources related to board design for Altera devices.

Guidelines: Perform Board Level Simulations

After you determined the system requirements and finalized the board design constraints, use an electronic design automation (EDA) simulation tool to perform board-level simulations. Use the IBIS or HSPICE models of the FPGA and the target LVDS device for the simulation.

The board-level simulation ensures optimum board setup where you can determine if the data window conforms to the input specification (electrical and timing) of the LVDS receiver.

You can use the programmable pre-emphasis feature on the true LVDS output buffers to compensate the frequency-dependent attenuation of the transmission line. With this feature, you can maximize the data eye opening at the far end receiver especially on long transmission lines.

Related Information

- [Altera IBIS Models](#)
Provides IBIS models of Altera devices for download.
- [Altera HSPICE Models](#)
Provides SPICE models of Altera devices for download.

2014.12.15

UG-M10LVDS



Subscribe



Send Feedback

You can set various parameter settings for the Altera Soft LVDS IP core to customize its behaviors, ports, and signals.

The Quartus II software generates your customized Altera Soft LVDS IP core according to the parameter options that you set in the parameter editor.

Related Information

- [MAX 10 High-Speed LVDS I/O Design Overview](#) on page 1-1
- [MAX 10 LVDS SERDES I/O Standards Support](#) on page 2-7
Lists the supported LVDS I/O standards and the support in different MAX 10 device variants.

Altera Soft LVDS Parameter Settings

There are four groups of options: **General** , **PLL Settings** , **Receiver Settings** , and **Transmitter Settings**

Table 7-1: Altera Soft LVDS Parameters - General

Parameter	Condition	Allowed Values	Description
Functional mode	—	• RX • TX	Specifies the functional mode for the Altera Soft LVDS IP core: • RX—specifies the IP is an LVDS receiver. • TX—specifies the IP is an LVDS transmitter.
Number of channels	—	1–18	Specifies the number of LVDS channels.
SERDES factor	—	1, 2, 4, 7, 8, 10	Specifies the number of bits per channel.

© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered

Table 7-2: Altera Soft LVDS Parameters - PLL Settings

Parameter	Condition	Allowed Values	Description
Use external PLL	—	- On - Off	Specifies whether the Altera Soft LVDS IP core generates a PLL or connects to a user-specified PLL.
Data rate	—	No limit	Specifies the data rate going out of the PLL. The multiplication value for the PLL is OUTPUT_DATA_RATE divided by INCLOCK_PERIOD.
Inclock frequency	—	Depends on Data rate .	Specifies the input clock frequency to the PLL in MHz.
Enable rx_locked port	- General, Functional mode = RX - Use external PLL = Off	- On - Off	If turned on, enables the rx_locked port.
Enable tx_locked port	- General, Functional mode = TX - Use external PLL = Off	- On - Off	If turned on, enables the tx_locked port.
Enable pll_areset port	Always on	- On - Off	If turned on, enables the pll_areset port in internal PLL mode. In external PLL mode, the pll_areset port is always on.
Enable tx_data_reset port	- General, Functional mode = TX - Use external PLL = Off	- On - Off	If turned on, enables the tx_data_reset port.
Enable rx_data_reset port	- General, Functional mode = RX - Use external PLL = Off	- On - Off	If turned on, enables the rx_data_reset port.

Parameter	Condition	Allowed Values	Description
Use common PLL(s) for receivers and transmitters	Use external PLL = Off	- On - Off	On—specifies that the compiler uses the same PLL for the LVDS receiver and transmitter. Off—specifies that the compiler uses different PLLs for LVDS receivers and transmitters. You can use common PLLs if you use the same input clock source, deserialization factor, pll_ areset source, and data rates.
Enable self-reset on loss lock in PLL	Use external PLL = Off	- On - Off	If turned on, the PLL is reset when it loses lock.
Desired transmitter inclock phase shift	- General, Functional mode = TX - Use external PLL = Off	Depends on Data rate .	Specifies the phase shift parameter used by the PLL for the transmitter.
Desired receiver inclock phase shift	- General, Functional mode = RX - Use external PLL = Off	Depends on Data rate .	Specifies the phase shift parameter used by the PLL for the receiver.

Table 7-3: Altera Soft LVDS Parameters - Receiver Settings

Parameter	Condition	Allowed Values	Description
Enable bit-slip mode	General, Functional mode = RX	- On - Off	If turned on, enables the rx_data_align port.
Enable independent bit-slip controls for each channel	General, Functional mode = RX	- On - Off	If turned on, enables the rx_channel_data_align port. The rx_channel_data_align is an edge-sensitive bit-slip control signal: - Each rising edge on this signal causes the data realignment circuitry to shift the word boundary by one bit. - The minimum pulse width requirement is one parallel clock cycle.

Parameter	Condition	Allowed Values	Description
Enable rx_data_align_reset port	- General, Functional mode = RX - Enable bitflip mode = On - Enable independent bitflip controls for each channel = Off	- On - Off	If turned on, enables the rx_data_align_reset port.
Add extra register for rx_data_align port	- General, Functional mode = RX - Enable bitflip mode = On	- On - Off	If turned on, registers the rx_data_align port. If you turn this option off, you must pre-register the rx_data_align[] port in the logic that feeds the receiver.
Bitflip rollover value	- General, Functional mode = RX - Enable bitflip mode = On	1–11	Specifies the number of pulses before the circuitry restores the serial data latency to 0.
Use RAM buffer	—	- On - Off	If turned on, the Altera Soft LVDS IP core implements the output synchronization buffer in the embedded memory blocks. This implementation option uses more logic than Use a multiplexer and synchronization register option but results in the correct word alignment.
Use a multiplexer and synchronization register	—	- On - Off	If turned on, the Altera Soft LVDS IP core implements a multiplexer instead of a buffer for output synchronization.
Use logic element based RAM	—	- On - Off	If turned on, the Altera Soft LVDS IP core implements the output synchronization buffer in the logic elements. This implementation option uses more logic than Use a multiplexer and synchronization register option but results in the correct word alignment.

Parameter	Condition	Allowed Values	Description
Register outputs	General, Functional mode = RX	- On - Off	If turned on, registers the rx_out[] port. If you turn this option off, you must pre-register the rx_out[] port in the logic that feeds the receiver.

Table 7-4: Altera Soft LVDS Parameters - Transmitter Settings

Parameter	Condition	Allowed Values	Description
Enable 'tx_outclock' output port	- General, Functional mode = TX - PLL Settings, Use external PLL = Off	- On - Off	If turned on, enables the tx_outclock port. Every tx_outclock signal goes through the shift register logic, except in the following parameter configurations: - When the outclock_divide_by signal = 1 - When the outclock_divide_by signal is equal to the deserialization_factor signal (for odd factors only), and the outclock_duty_cycle signal is 50
Tx_outclock division factor	- General, Functional mode = TX - PLL Settings, Use external PLL = Off - Enable 'tx_outclock' output port = On	1 2 - General, SERDES factor	Specifies that the frequency of the tx_outclock signal is equal to the transmitter output data rate divided by the selected division factor.
Outclock duty cycle	—	—	Specifies the external clock timing constraints.
Desired transmitter outclock phase shift	- General, Functional mode = TX - PLL Settings, Use external PLL = Off - Enable 'tx_outclock' output port = On	Depends on Data rate.	Specifies the phase shift of the output clock relative to the input clock.

Parameter	Condition	Allowed Values	Description
Register 'tx_in' input port	General, Functional mode = TX	- On - Off	If turned on, registers the tx_in[] port. If you turn this option off, you must pre-register the tx_in[] port in the logic that feeds the transmitter.
Clock resource	- General, Functional mode = TX - Register 'tx_in' input port = On	- tx_inclock - tx_coreclock	Specifies which clock resource registers the tx_in input port.
Enable 'tx_coreclock' output port	General, Functional mode = TX	- On - Off	If turned on, enables the tx_coreclock output port.
Clock source for 'tx_coreclock'	- General, Functional mode = TX - Enable 'tx_coreclock' output port = On	- Auto selection - Global clock - Regional clock - Dual-Regional clock	Specifies which clock resource drives the tx_coreclock output port.

Related Information

- [Altera Soft LVDS IP Core in Transmitter Mode](#) on page 3-6
- [Specifying IP Core Parameters and Options](#) on page 3-6
- [Altera Soft LVDS IP Core in Receiver Mode](#) on page 4-6

Altera Soft LVDS Interface Signals

Depending on parameter settings you specify, different signals are available for the Altera Soft LVDS IP core.

Table 7-5: Transmitter Interface Signals

Signal Name	Direction	Width (Bit)	Description
pll_areset	Input	1	Asynchronously resets all counters to the initial values.

Signal Name	Direction	Width (Bit)	Description
tx_data_reset	Input	$\langle n \rangle$	Asynchronous reset for the shift registers, capture registers, and synchronization registers for all channels. - This signal is used if Use external PLL parameter setting is turned on. - This signal does not affect the data realignment block or the PLL.
tx_in[]	Input	$\langle m \rangle$	This signal is parallel data that Altera Soft LVDS IP core transmits serially. Input data is synchronous to the tx_coreclock signal. The data bus width per channel is the same as the serialization factor (SF).
tx_inclock	Input	1	Reference clock input for the transmitter PLL. The parameter editor automatically selects the appropriate PLL multiplication factor based on the data and reference clock frequency.
tx_coreclock	Output	1	Output clock that feeds non-peripheral logic. FPGA fabric-transmitter interface clock—the parallel transmitter data generated in the FPGA fabric is clocked with this clock.
tx_locked	Output	1	Provides the LVDS PLL status: - Remains high when the PLL is locked to the input reference clock. - Remains low when the PLL fails to lock.
tx_out[]	Output	$\langle n \rangle$	Serialized LVDS data output signal of $\langle n \rangle$ channels. tx_out[$(\langle n \rangle - 1) : 0$] drives parallel data from tx_in[$(\langle J \rangle \times \langle n \rangle) - 1 : 0$] where $\langle J \rangle$ is the serialization factor and $\langle n \rangle$ is the number of channels. tx_out[0] drives data from tx_in[$(\langle J \rangle - 1) : 0$]. tx_out[1] drives data from the next $\langle J \rangle$ number of bits on tx_in.

Signal Name	Direction	Width (Bit)	Description
tx_outclock	Output	1	External reference clock. The frequency of this clock is programmable to be the same as the data rate.

Table 7-6: Receiver Interface Signals

signal Name	Direction	Width (Bit)	Description
rx_data_reset	Input	<n>	Asynchronous reset for all channels, excluding the PLL. This signal is available if Use external PLL parameter setting is turned on.
rx_in[]	Input	<n>	LVDS serial data input signal of <n> channels. rx_in[(<n>-1)..0] is deserialized and driven on rx_out[(<J> × <n>)-1..0] where <J> is the deserialization factor and <n> is the number of channels. rx_in[0] drives data to rx_out[(<J>-1)..0]. rx_in[1] drives data to the next <J> number of bits on rx_out.
rx_inclock	Input	1	LVDS reference input clock. The parameter editor automatically selects the appropriate PLL multiplication factor based on the data rate and reference clock frequency selection.
rx_coreclk	Input	<n>	LVDS reference input clock. - Replaces the non-peripheral clock from the PLL. - One clock for each channel.
rx_locked	Output	1	Provides the LVDS PLL status: - Stays high when the PLL is locked to the input reference clock. - Stays low when the PLL fails to lock.
rx_out	Output	<m>	Receiver parallel data output. The data bus width per channel is the same as the deserialization factor (DF).

signal Name	Direction	Width (Bit)	Description
rx_outclock	Output	1	Parallel output clock from the receiver PLL. • This signal is not available if you turn on the Use external PLL parameter setting. • The FPGA fabric–receiver interface clock must be driven by the PLL instantiated through the ALTPLL parameter editor.
rx_data_align	Input	1	Controls the byte alignment circuitry. You can register this signal using the rx_outclock signal.
rx_data_align_reset	Input	1	Resets the byte alignment circuitry. Use the rx_data_align_reset input signal if: • You need to reset the PLL during device operation. • You need to re-establish the word alignment.
rx_channel_data_align	Input	<n>	Controls byte alignment circuitry.
rx_cda_reset	Input	<n>	Asynchronous reset to the data realignment circuitry. This signal resets the data realignment block. The minimum pulse width requirement for this reset is one parallel clock cycle.

Additional Information for MAX 10 High-Speed LVDS I/O User Guide



2014.12.15

UG-M10LVDS



Subscribe



Send Feedback

Document Revision History for MAX 10 High-Speed LVDS I/O User Guide

Date	Version	Changes
May 2015	2015.05.04	- Removed the F672 package from the 10M25 device. - Updated the number of bottom true receiver channels for package M153 of the 10M02 device from 49 to 13. - Added BLVDS output support in single-supply MAX 10 devices. Previously, BLVDS support for single-supply devices was input only. - Updated the RSKM definition in the topic about receiver input skew margin to include jitter induced from core noise and I/O switching noise. - Updated topics related to using the Altera Soft LVDS IP core (transmitter or receiver) in external PLL mode: - Added rx_readclock, rx_syncclock, and tx_syncclock ports. - Removed pll_areset port. - Added examples for odd and even serialization factors. - Added procedures to obtain the external PLL clock parameters. - Removed similar guidelines in the chapter for the transmitter and receiver design. The updated guidelines for the receiver only and transmitter only designs can apply for designs that use both transmitters and receivers. - Updated parameter settings of the Altera Soft LVDS IP core: - Removed allowed values "6" and "9" from the SERDES factor parameter. - Added allowed value "Off" to the Enable pll_areset port parameter. - Updated the parameter label Register_rx_bitslip_ctrl port to Add extra register for rx_data_align port and specified that you must pre-register the port if you turn it on.

© 2015 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered

Date	Version	Changes
December 2014	2014.12.15	- Updated table listing LVDS channels to include LVDS channel counts for each device package. - Added information in the topics about channels placement that MAX 10 devices support x18 bundling mode. - Updated the examples in topics about channels PLL placement to provide more details. - Added link to the MAX 10 Clocking and PLL User Guide that provides more information about the PLL and the PLL output counters used to clock the soft SERDES.
September 2014	2014.09.22	Initial release.