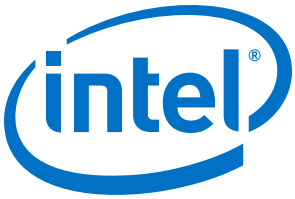


Intel® MAX® 10 High-Speed LVDS I/O User Guide

Updated for Intel® Quartus® Prime Design Suite: **17.1**

Subscribe

Send Feedback

UG-M10LVDS | 2017.12.15

Latest document on the web: **PDF** | **HTML**

Contents

1 Intel® MAX® 10 High-Speed LVDS I/O Overview

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

The LVDS I/O banks in Intel MAX 10 devices feature true and emulated LVDS buffers:

- True LVDS buffers support LVDS using true differential buffers.
- Emulated LVDS buffers use a pair of single-ended pins to emulate differential buffers.

Table 1. Summary of LVDS I/O Buffers Support in Intel 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 Intel MAX 10 D (dual supply) and S (single supply) device variants support different LVDS I/O standards. For a list of LVDS I/O standards supported by the Intel MAX 10 D and S variants, refer to the related information.

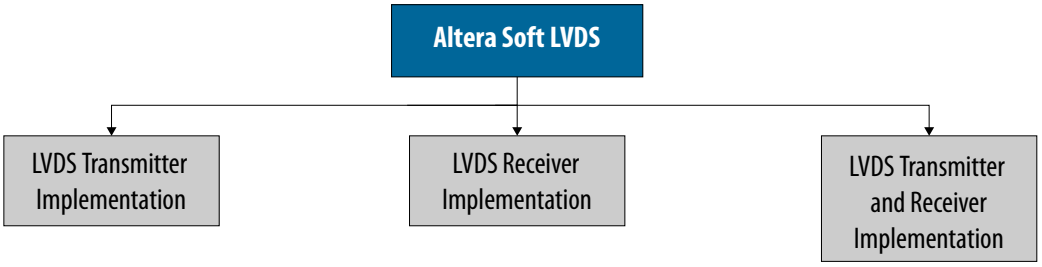
Related Links

- Intel MAX 10 High-Speed LVDS Architecture and Features on page 6
  Provides information about the high-speed LVDS architecture and the features supported by the device.
- Altera Soft LVDS IP Core References on page 46
  Lists the parameters and signals of Altera Soft LVDS IP core for Intel MAX 10 devices.
- Intel MAX 10 LVDS SERDES I/O Standards Support on page 11
  Lists the supported LVDS I/O standards and the support in different Intel MAX 10 device variants.
- Intel MAX 10 High-Speed LVDS I/O User Guide Archives on page 52
  Provides a list of user guides for previous versions of the Altera Soft LVDS IP core.

1.1 Altera Soft LVDS Implementation Overview

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

Figure 1. Intel MAX 10 LVDS Implementation Overview



Related Links

- Intel MAX 10 LVDS Transmitter Design on page 16
  Provides information and guidelines for implementing LVDS transmitter in Intel MAX 10 devices using the Altera Soft LVDS IP core.
- Intel MAX 10 LVDS Receiver Design on page 26
  Provides information and guidelines for implementing LVDS receiver in Intel MAX 10 devices using the Altera Soft LVDS IP core.
- Intel MAX 10 LVDS Transmitter and Receiver Design on page 40
  Provides design guidelines for implementing both LVDS transmitters and receivers in the same Intel MAX 10 device.

2 Intel MAX 10 High-Speed LVDS Architecture and Features

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

- For LVDS transmitters and receivers, Intel 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), Intel MAX 10 devices use logic elements (LE) registers.

2.1 Intel MAX 10 LVDS Channels Support

The LVDS channels available vary among Intel MAX 10 devices. All I/O banks in Intel 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. LVDS Buffers in Intel MAX 10 Devices

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

Product Line	Package	Device Power Supply	Side	True LVDS Pairs		Emulated LVDS Pairs
				TX	RX	
10M02	V36	Dual	Top	0	1	1
			Right	0	3	3
			Left	0	3	3
			Bottom	3	3	3
	M153	Single	Top	0	12	12
			Right	0	12	12
			Left	0	12	12
			Bottom	9	13	13
	U169	Single	Top	0	12	12
			Right	0	17	17
			Left	0	15	15
			Bottom	9	14	14
	U324	Single	Top	0	27	27
			Right	0	31	31
			Left	0	28	28

continued...

Product Line	Package	Device Power Supply	Side	True LVDS Pairs		Emulated LVDS Pairs
				TX	RX	
			Bottom	15	28	28
		Dual	Top	0	13	13
			Right	0	24	24
			Left	0	20	20
			Bottom	9	16	16
	E144	Single	Top	0	10	10
			Right	0	12	12
			Left	0	11	11
			Bottom	7	12	12
10M04	M153	Single	Top	0	12	12
			Right	0	12	12
			Left	0	12	12
			Bottom	9	13	13
	U169	Single	Top	0	12	12
			Right	0	17	17
			Left	0	15	15
			Bottom	9	14	14
	U324	Single	Top	0	27	27
			Right	0	31	31
			Left	0	28	28
			Bottom	15	28	28
		Dual	Top	0	27	27
			Right	0	31	31
			Left	0	28	28
			Bottom	15	28	28
	F256	Dual	Top	0	19	19
			Right	0	22	22
			Left	0	19	19
			Bottom	13	20	20
	E144	Single	Top	0	8	8
			Right	0	12	12
			Left	0	11	11
			Bottom	10	10	10
10M08	V81	Dual	Top	0	5	5
			Right	0	7	7

continued...

Product Line	Package	Device Power Supply	Side	True LVDS Pairs		Emulated LVDS Pairs
				TX	RX	
			Left	0	6	6
			Bottom	7	7	7
	M153	Single	Top	0	12	12
			Right	0	12	12
			Left	0	12	12
			Bottom	9	13	13
	U169	Single	Top	0	12	12
			Right	0	17	17
			Left	0	15	15
			Bottom	9	14	14
	U324	Single	Top	0	27	27
			Right	0	31	31
			Left	0	28	28
			Bottom	15	28	28
		Dual	Top	0	27	27
			Right	0	31	31
			Left	0	28	28
			Bottom	15	28	28
	F256	Dual	Top	0	19	19
			Right	0	22	22
			Left	0	19	19
			Bottom	13	20	20
	E144	Single	Top	0	8	8
			Right	0	12	12
			Left	0	11	11
			Bottom	10	10	10
	F484	Dual	Top	0	27	27
			Right	0	33	33
			Left	0	28	28
			Bottom	15	28	28
10M16	U169	Single	Top	0	12	12
			Right	0	17	17
			Left	0	15	15
			Bottom	9	14	14
	U324	Single	Top	0	27	27

continued...

Product Line	Package	Device Power Supply	Side	True LVDS Pairs		Emulated LVDS Pairs
				TX	RX	
			Right	0	31	31
			Left	0	28	28
			Bottom	15	28	28
		Dual	Top	0	27	27
			Right	0	31	31
			Left	0	28	28
			Bottom	15	28	28
	F256	Dual	Top	0	19	19
			Right	0	22	22
			Left	0	19	19
			Bottom	13	20	20
	E144	Single	Top	0	8	8
			Right	0	12	12
			Left	0	11	11
			Bottom	10	10	10
	F484	Dual	Top	0	39	39
			Right	0	38	38
			Left	0	32	32
			Bottom	22	42	42
10M25	F256	Dual	Top	0	19	19
			Right	0	22	22
			Left	0	19	19
			Bottom	13	20	20
	E144	Single	Top	0	8	8
			Right	0	12	12
			Left	0	11	11
			Bottom	10	10	10
	F484	Dual	Top	0	41	41
			Right	0	48	48
			Left	0	36	36
			Bottom	24	46	46
10M40	F256	Dual	Top	0	19	19
			Right	0	22	22
			Left	0	19	19
			Bottom	13	20	20

continued...

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

Related Links

- Intel MAX 10 Device Pin-Out Files
  Provides pin-out files for each Intel MAX 10 device.
- Intel MAX 10 High-Speed LVDS I/O Location on page 13

2.2 Intel MAX 10 LVDS SERDES I/O Standards Support

The Intel MAX 10 D and S device variants support different LVDS I/O standards. All I/O banks in Intel 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 3. Intel MAX 10 LVDS I/O Standards Support

Single and dual supply Intel 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	Intel 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	Yes	—
Emulated RSDS (single resistor)	All	Yes	—	Yes	—	All I/O banks support emulated RSDS output buffers.
Emulated RSDS (three resistors)	All	Yes	—	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	—	—
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 Intel MAX 10 input buffer. • TMDS receiver support uses dedicated 2.5 V LVDS input buffer.

continued...

I/O Standard	I/O Bank	TX	RX	Intel MAX 10 Device Support		Notes
				Dual Supply Device	Single Supply Device	
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.
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 Links

- Intel MAX 10 FPGA Device Overview
- Emulated LVDS External Termination on page 18
- Emulated RSDS, Emulated Mini-LVDS, and Emulated PPDS Transmitter External Termination on page 19
- TMDS Receiver External Termination on page 29
- Sub-LVDS Transmitter External Termination on page 18
- Sub-LVDS Receiver External Termination on page 28
- SLVS Transmitter External Termination on page 19
- SLVS Receiver External Termination on page 28
- HiSpi Receiver External Termination on page 29

2.3 Intel MAX 10 High-Speed LVDS Circuitry

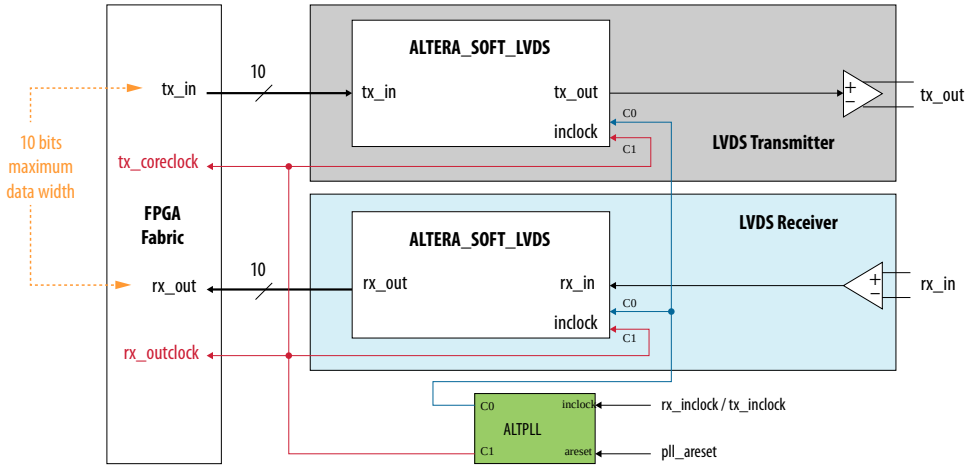
The LVDS solution uses the I/O elements and registers in the Intel MAX 10 devices. The Altera Soft LVDS IP core implements the serializer and deserializer as soft SERDES blocks in the core logic.

The Intel 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 Intel 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 Intel Quartus® Prime software uses the parameter settings of the Altera Soft LVDS IP core to automatically construct the differential SERDES in the core fabric.

Figure 2. 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 Links

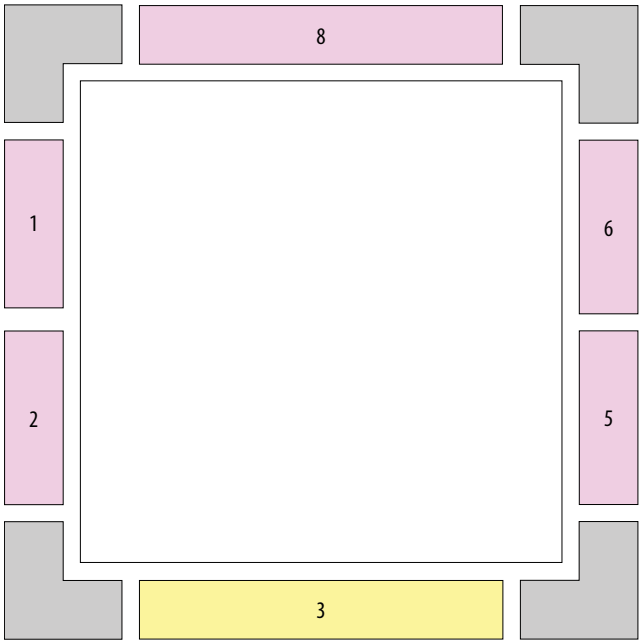
Intel MAX 10 Clocking and PLL User Guide
Provides more information about the PLL and the PLL output counters.

2.4 Intel MAX 10 High-Speed LVDS I/O Location

The I/O banks in Intel 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 3. LVDS Support in I/O Banks of 10M02 Devices (Except Single Power Supply U324 Package)

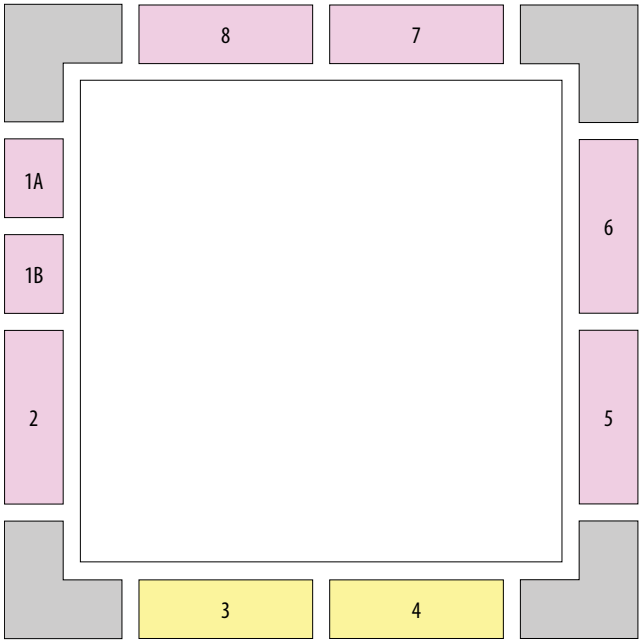
This figure shows a top view of the silicon die. Each bank is labeled with the actual bank number. LVPECL support only in banks 2 and 6.



	TX		RX	
LVDS				
Emulated LVDS				
RSDS				
Emulated RSDS				
Mini-LVDS				
Emulated Mini-LVDS				
PPDS				
Emulated PPDS				
BLVDS				
LVPECL				
TMDS				
Sub-LVDS				
SLVS				
HiSpi				

Figure 4. LVDS Support in I/O Banks of 10M02 (Single Power Supply U324 Package), 10M04, and 10M08 Devices

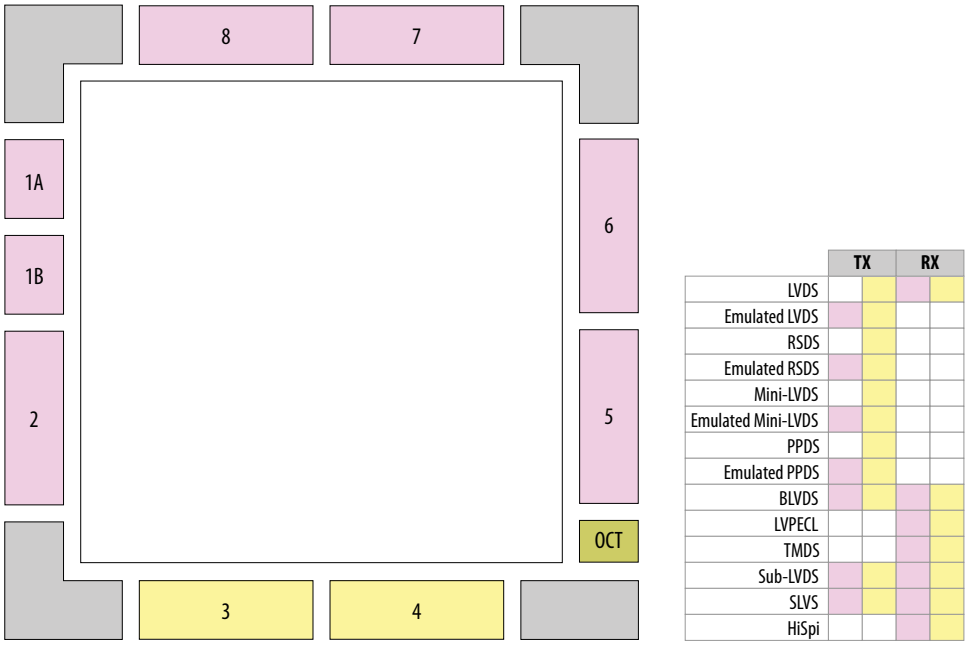
This figure shows a top view of the silicon die. Each bank is labeled with the actual bank number. LVPECL support only in banks 2 and 6.



	TX		RX	
LVDS				
Emulated LVDS				
RSDS				
Emulated RSDS				
Mini-LVDS				
Emulated Mini-LVDS				
PPDS				
Emulated PPDS				
BLVDS				
LVPECL				
TMDS				
Sub-LVDS				
SLVS				
HiSpi				

Figure 5. 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.



Related Links

- PLL Specifications
  Provides PLL performance information for Intel MAX 10 devices.
- High-Speed I/O Specifications
  Provides minimum and maximum data rates for different data widths in Intel MAX 10 devices.

2.5 Differential I/O Pins in Low Speed Region

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

- For each user I/O pin (excluding configuration pin) that you place in the low speed region, the Intel Quartus Prime 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 Links

- Intel MAX 10 Device Pin-Out Files
  Provides pin-out files for each Intel MAX 10 device.
- Intel MAX 10 Device Datasheet
- MAX 10 I/O Banks Locations, MAX 10 General Purpose I/O User Guide
  Shows the locations of the high speed and low speed I/O banks.

3 Intel MAX 10 LVDS Transmitter Design

You can implement transmitter-only applications using the Intel 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.

3.1 High-Speed I/O Transmitter Circuitry

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

Related Links

Intel MAX 10 High-Speed LVDS Circuitry on page 12

3.2 LVDS Transmitter Programmable I/O Features

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

3.2.1 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.

Figure 6. LVDS Output with Programmable Pre-Emphasis

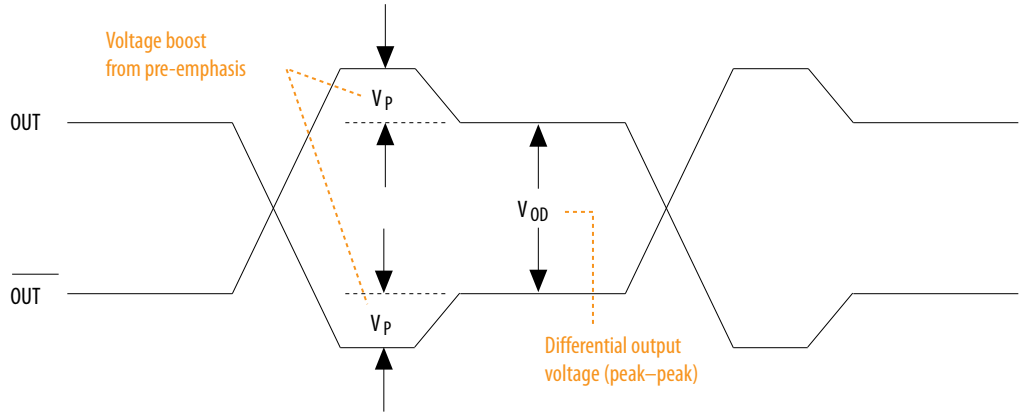


Table 4. Intel Quartus Prime Software Assignment for Programmable Pre-Emphasis

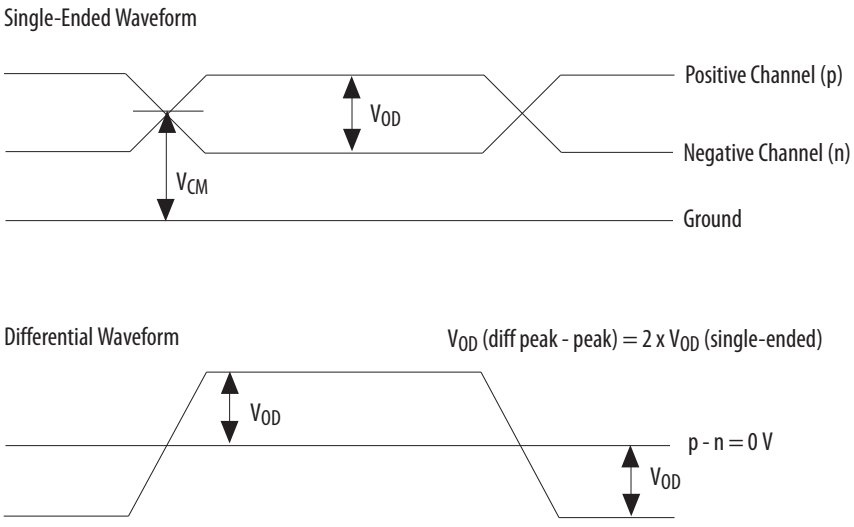
Field	Assignment
To	`tx_out`
Assignment name	Programmable Pre-emphasis
Allowed values	0 (disabled), 1 (enabled). Default is 1.

3.2.2 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 7. 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 Intel Quartus Prime software Assignment Editor.

Table 5. Intel Quartus Prime 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.

3.3 LVDS Transmitter I/O Termination Schemes

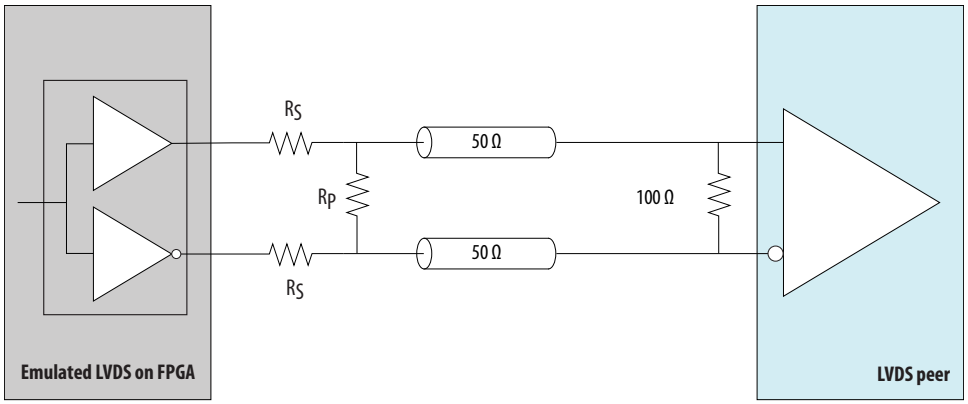
For transmitter applications in Intel MAX 10 devices, you must implement external termination for some I/O standards.

3.3.1 Emulated LVDS External Termination

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

Figure 8. External Termination for Emulated LVDS Transmitter

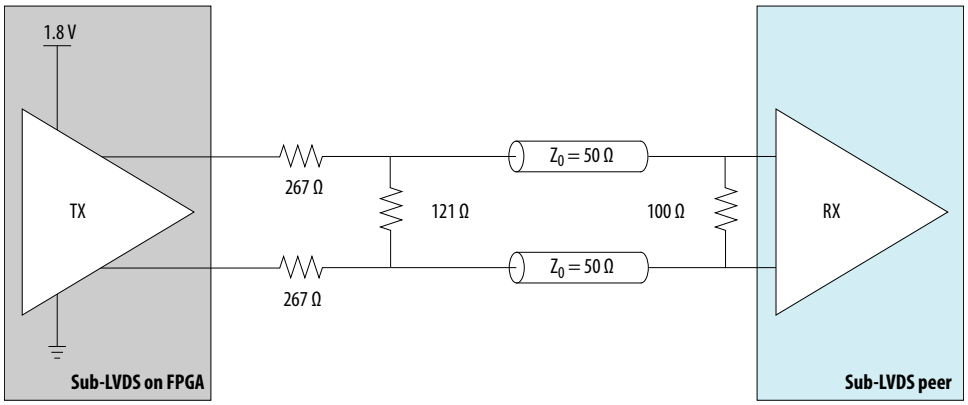
In this figure, R_S = 120 Ω and R_P = 170 Ω.



3.3.2 Sub-LVDS Transmitter External Termination

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

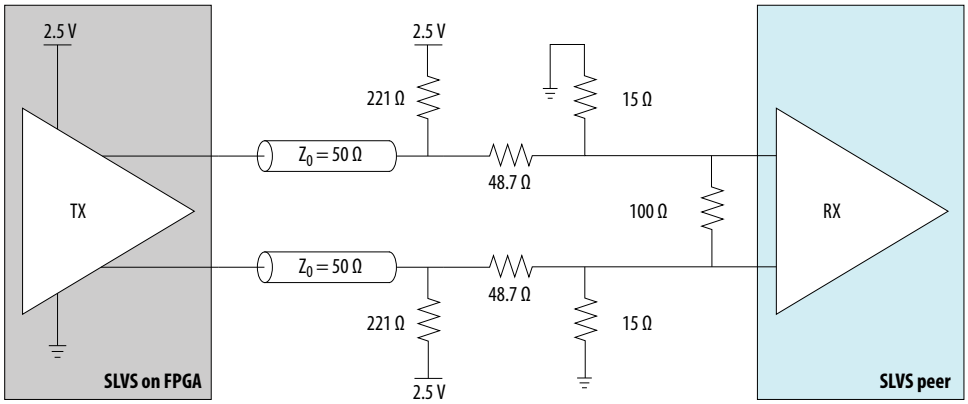
Figure 9. External Termination for Sub-LVDS Transmitter



3.3.3 SLVS Transmitter External Termination

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

Figure 10. External Termination for SLVS Transmitter



3.3.4 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 11. External Termination for Emulated RSDS, Mini-LVDS, or PPDS Transmitter

In this figure, R_S is 120 Ω and R_P is 170 Ω.

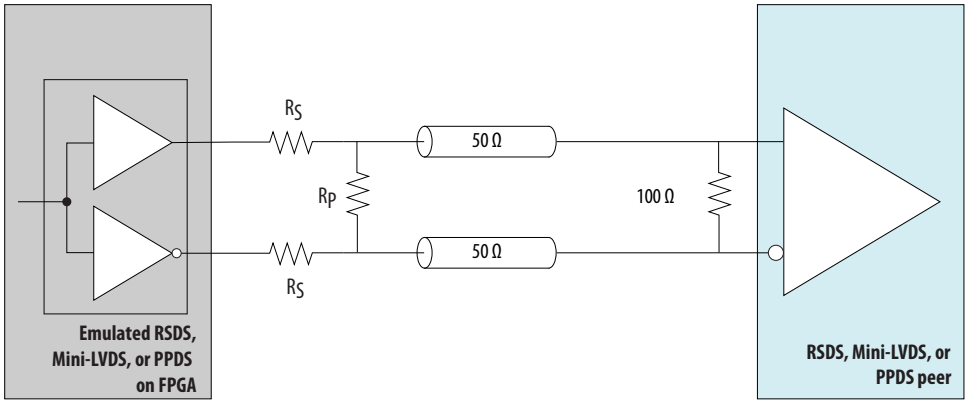
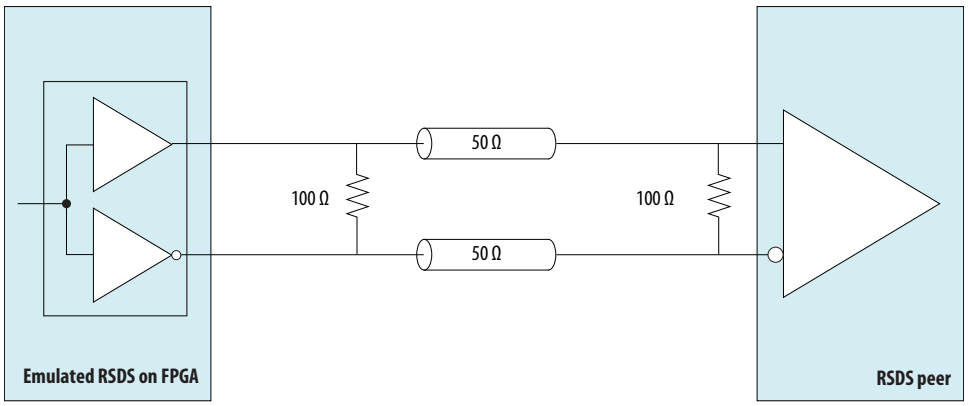


Figure 12. Single-Resistor External Termination for Emulated RSDS Transmitter



3.4 LVDS Transmitter FPGA Design Implementation

Intel MAX 10 devices use a soft SERDES architecture to support high-speed I/O interfaces. The Intel Quartus Prime 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, Intel MAX 10 devices use the I/O registers and LE registers in the core fabric.

3.4.1 Altera Soft LVDS IP Core in Transmitter Mode

In the Intel Quartus Prime 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 Intel 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 Links

- Altera Soft LVDS Parameter Settings on page 46
- Introduction to Intel FPGA IP Cores
  Provides general information about all Intel FPGA IP cores, including parameterizing, generating, upgrading, and simulating IP cores.
- Creating Version-Independent IP and Qsys Simulation Scripts
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- Project Management Best Practices
  Guidelines for efficient management and portability of your project and IP files.

3.4.1.1 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.

3.4.1.1.1 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.

3.4.1.1.2 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 Links

- Intel MAX 10 Clocking and PLL User Guide
  Provides more information about the PLL and the PLL output counters.
- Intel MAX 10 Clocking and PLL User Guide
  Provides more information about the PLL and the PLL output counters.

3.4.1.2 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 Links

Intel MAX 10 Clocking and PLL User Guide
Provides more information about the PLL and the PLL output counters.

3.4.1.2.1 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 6. Example: Signal Interface between 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

3.4.1.2.2 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 13. Clock Parameters Example for Altera Soft LVDS Transmitter

Clocks

	Clock Name	Type	Period	Frequency	Duty Cycle	Phase
1	inst|lvdstx_inst|lvds_tx_pll|clk[0]	Generated	2.500	400.0 MHz	50.00	-90.0
2	inst|lvdstx_inst|lvds_tx_pll|clk[1]	Generated	12.500	80.0 MHz	50.00	-18.0
3	tx_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.

3.4.1.3 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.

3.4.2 High-Speed I/O Timing Budget

The LVDS I/O standard enables high-speed transmission of data. To take advantage of the fast performance, analyze the timing of high-speed signals. 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. Use the timing parameters provided by IC vendors. High-speed differential data transmission is strongly influenced by board skew, cable skew, and clock jitter.

Intel MAX 10 devices implement the SERDES in LEs. You must set proper timing constraints to indicate whether the SERDES captures the data as expected or otherwise. You can set the timing contraints using the Timing Analyzer tool in the Intel Quartus Prime software or manually in the Synopsys* Design Constraints (`.sdc`) file.

3.4.2.1 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 Intel MAX 10 transmitter in a source-synchronous differential interface. You can get the TCCS value from the device datasheet.

Related Links

Intel MAX 10 Device Datasheet

3.4.3 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.

Intel recommends that you create a Intel Quartus Prime design, specify your device I/O assignments, and compile your design to validate your pin placement. The Intel Quartus Prime 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 Intel Quartus Prime 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 Intel Quartus Prime software automatically assigns the negative pin if the positive pin is assigned with a differential I/O standard.

In Intel 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 routes of both pins in a differential pair are matched even if the pins are non-adjacent.

The Altera Soft LVDS IP core for Intel MAX 10 devices supports a maximum of 18 channels per IP instantiation. Each channel can support deserialization factor (parallel data width) from one to ten bits. When you are grouping channels for an application, you must consider the channel to channel skew during Fitter placement. To minimize

skew, place all LVDS channels in the group side by side. For your PCB design, Intel recommends that you perform package skew compensation to minimize skew and maximize performance.

Note: For Intel MAX 10 devices, the Intel Quartus Prime software does not provide a package skew compensation report.

3.4.4 Guidelines: LVDS Channels PLL Placement

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

Table 7. Examples of Usable PLL to Drive I/O Banks in Intel MAX 10 Devices

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

3.4.5 Guidelines: LVDS Transmitter Logic Placement

The Intel Quartus Prime 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 Intel Quartus Prime 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 Links

Intel Quartus Prime Incremental Compilation for Hierarchical and Team-Based Design chapter, Volume 1: Design and Synthesis, Intel Quartus Prime Handbook
Provides step by step instructions about creating a design floorplan with LogicLock location assignments.

3.4.6 Guidelines: Enable LVDS Pre-Emphasis for E144 Package

For Intel MAX 10 devices in the E144 package, Intel recommends that you enable LVDS pre-emphasis to achieve optimum signal integrity (SI) performance. If you do not enable pre-emphasis, undesirable SI condition may be induced in the device resulting in LVDS eye height sensitivity.

3.5 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 Intel SignalTap® II Logic Analyzer to perform system level verification to correlate the system against your design targets.

Related Links

In-System Debugging Using External Logic Analyzers chapter, Volume 3: Verification, Intel Quartus Prime Handbook

3.5.1 Perform RTL Simulation Before Hardware Debug

Before you debug on hardware, Intel 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.

3.5.2 Geometry-Based and Physics-Based I/O Rules

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

For more information, refer to the related information.

Related Links

Intel MAX 10 General Purpose I/O User Guide

4 Intel MAX 10 LVDS Receiver Design

You can implement receiver-only applications using the Intel 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.

4.1 High-Speed I/O Receiver Circuitry

The LVDS receiver circuitry uses the I/O elements and registers in the Intel 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)

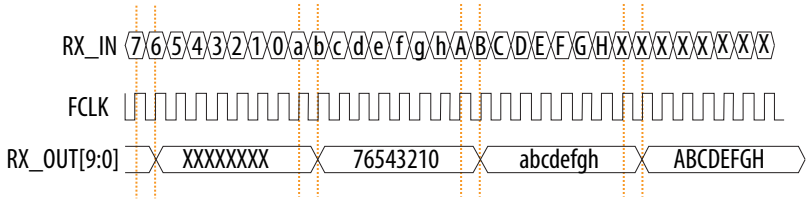
Related Links

Intel MAX 10 High-Speed LVDS Circuitry on page 12

4.1.1 Soft Deserializer

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

Figure 14. LVDS x8 Deserializer Waveform



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.

4.1.2 Data Realignment Block (Bit Slip)

Skew in the transmitted data and skew added by the transmision 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.

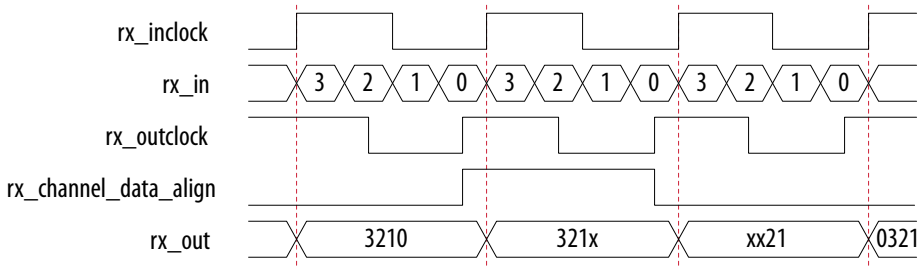
An optional RX_CHANNEL_DATA_ALIGN port controls the bit insertion of each receiver independently of the internal logic. The data slips one bit on the rising edge of RX_CHANNEL_DATA_ALIGN.

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 15. Data Realignment Timing

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



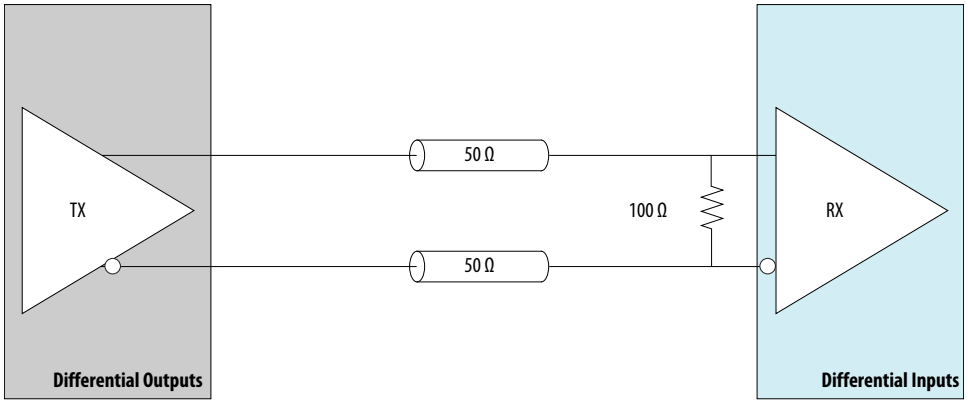
4.2 LVDS Receiver I/O Termination Schemes

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

4.2.1 LVDS, Mini-LVDS, and RSDS Receiver External Termination

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

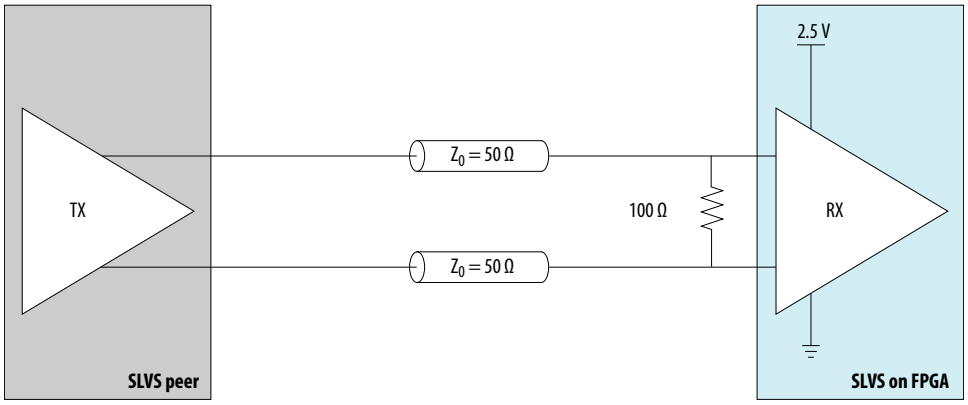
Figure 16. External Termination for LVDS I/O Standard



4.2.2 SLVS Receiver External Termination

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

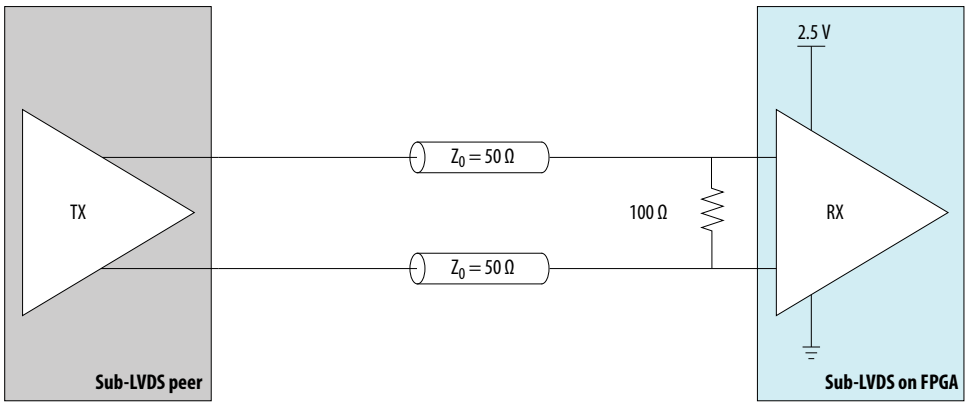
Figure 17. External Termination for SLVS Receiver



4.2.3 Sub-LVDS Receiver External Termination

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

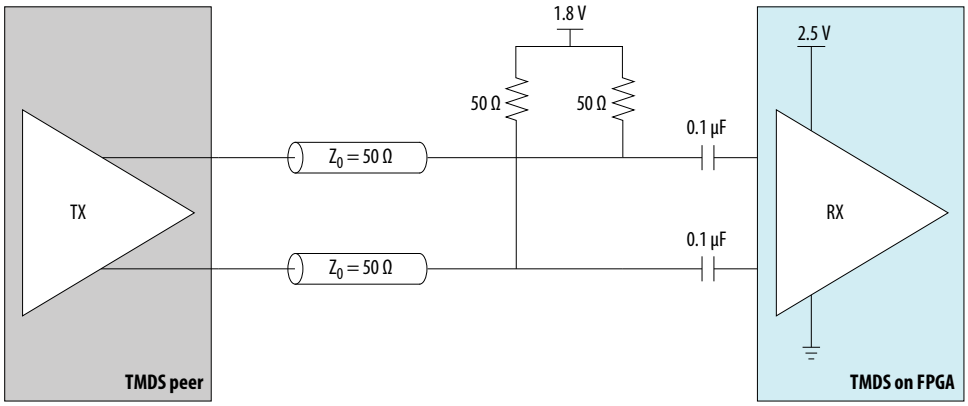
Figure 18. External Termination for Sub-LVDS Receiver



4.2.4 TMDS Receiver External Termination

Figure 19. External Termination for TMDS Receiver

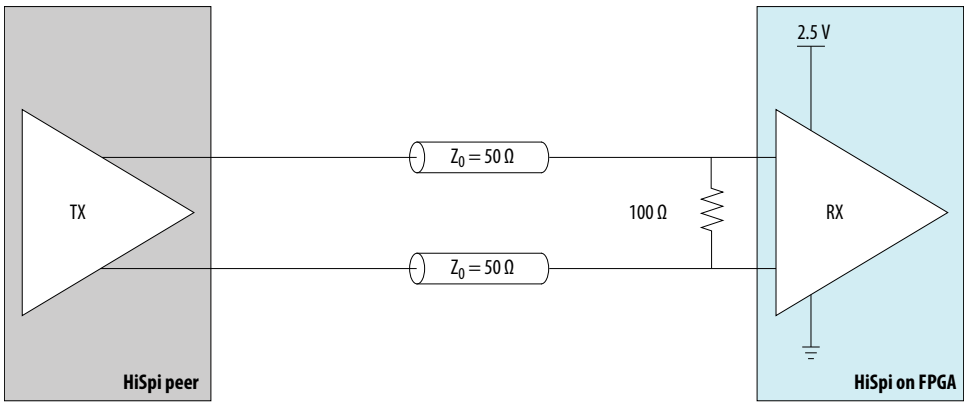
This diagram shows the external level shifter that is required for the TMDS input standards support in Intel MAX 10 devices.



4.2.5 HiSpi Receiver External Termination

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

Figure 20. External Termination for HiSpi Receiver



4.2.6 LVPECL External Termination

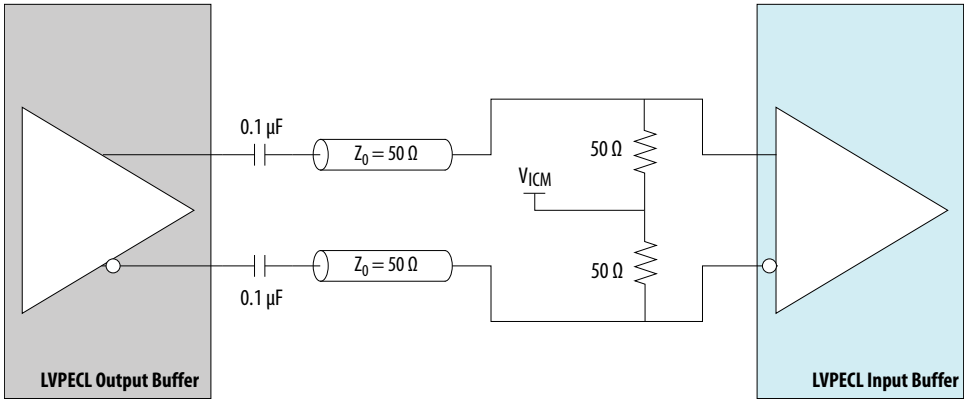
The Intel MAX 10 devices support the LVPECL I/O standard on input clock pins only.

- 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.

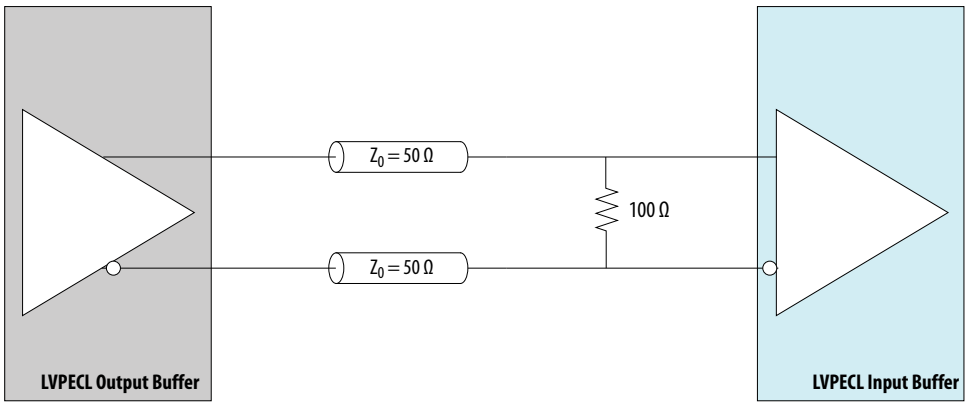
Note: Intel recommends that you use IBIS models to verify your LVPECL AC/DC-coupled termination.

Figure 21. LVPECL AC-Coupled Termination



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

Figure 22. LVPECL DC-Coupled Termination



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

Related Links

Intel MAX 10 Device Datasheet

4.3 LVDS Receiver FPGA Design Implementation

Intel MAX 10 devices use a soft SERDES architecture to support high-speed I/O interfaces. The Intel Quartus Prime 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, Intel MAX 10 devices use the I/O registers and LE registers in the core fabric.

4.3.1 Altera Soft LVDS IP Core in Receiver Mode

In the Intel Quartus Prime software, you can design your high-speed receiver interfaces using the Altera Soft LVDS IP core. This IP core uses the resources in the Intel 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 Links

- Altera Soft LVDS Parameter Settings on page 46
- Introduction to Intel FPGA IP Cores
  Provides general information about all Intel FPGA IP cores, including parameterizing, generating, upgrading, and simulating IP cores.
- Creating Version-Independent IP and Qsys Simulation Scripts
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- Project Management Best Practices
  Guidelines for efficient management and portability of your project and IP files.

4.3.1.1 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.

4.3.1.1.1 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.

4.3.1.1.2 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 Links

- Intel MAX 10 Clocking and PLL User Guide
  Provides more information about the PLL and the PLL output counters.
- Intel MAX 10 Clocking and PLL User Guide
  Provides more information about the PLL and the PLL output counters.

4.3.1.2 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 receiver.
- Parallel clock used to clock the receiver FPGA fabric logic.
- The `locked` signal for Altera Soft LVDS PLL reset port.

Related Links

Intel MAX 10 Clocking and PLL User Guide
Provides more information about the PLL and the PLL output counters.

4.3.1.2.1 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 receiver, use the source-synchronous compensation mode.

Table 8. 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`

Table 9. 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`
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)

4.3.1.2.2 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, `clk[0]` is the fast clock, `clk[1]` is the slow clock, and `clk[2]` is the read clock.

Figure 23. 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.

4.3.1.3 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.

4.3.2 High-Speed I/O Timing Budget

The LVDS I/O standard enables high-speed transmission of data. To take advantage of the fast performance, analyze the timing of high-speed signals. 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. Use the timing parameters provided by IC vendors. High-speed differential data transmission is strongly influenced by board skew, cable skew, and clock jitter.

Intel MAX 10 devices implement the SERDES in LEs. You must set proper timing constraints to indicate whether the SERDES captures the data as expected or otherwise. You can set the timing contraints using the Timing Analyzer tool in the Intel Quartus Prime software or manually in the Synopsys Design Constraints (`.sdc`) file.

4.3.2.1 Receiver Input Skew Margin

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

Related Links

Guidelines: Control Channel-to-Channel Skew on page 43

4.3.2.1.1 RSKM Equation

The RSKM equation expresses the relationship between RSKM, TCCS, and SW.

Figure 24. 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. The TCCS measurement includes the t_{CO} variation, clock, and clock skew.

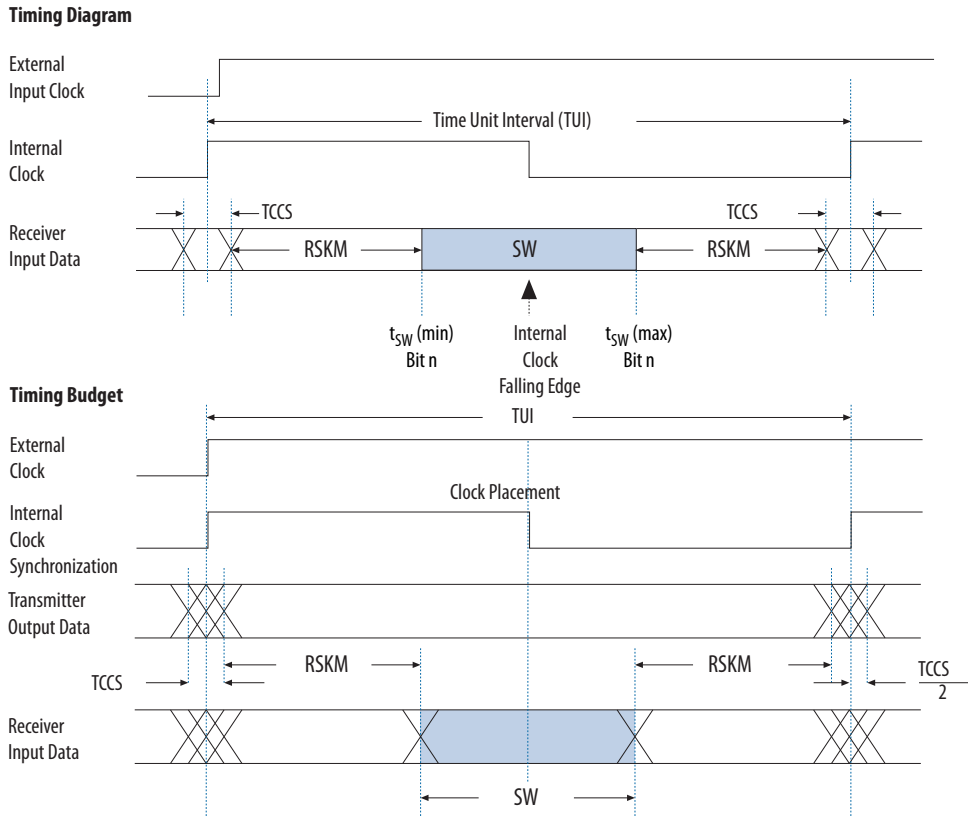
Note: If there is additional board channel-to-channel skew, consider the total receiver channel-to-channel skew (RCCS) instead of TCCS. Total RCCS = TCCS + board channel-to-channel skew.

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 value, after deducting transmitter jitter, indicates that the LVDS receiver cannot sample the data properly.

Figure 25. Differential High-Speed Timing Diagram and Timing Budget

This figure shows the relationship between the RSKM, TCCS, and the SW of the receiver.



Related Links

Guidelines: Control Channel-to-Channel Skew on page 43

4.3.2.1.2 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 ps
- SW = 300 ps
- TUI = 1000 ps
- Total RCCS = TCCS + Board channel-to-channel skew = 100 ps + 200 ps = 300 ps
- RSKM = (TUI – SW – RCCS) / 2 = (1000 ps – 300 ps – 300 ps) / 2 = 200 ps

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

Related Links

Guidelines: Control Channel-to-Channel Skew on page 43

4.3.2.2 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 Intel Quartus Prime 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 Intel Quartus Prime Timing Analyzer. The timing constraints specify the timing requirements necessary to ensure reliable data capture.

Related Links

Timing Constraints, Intel Quartus Prime Standard EditionHandbook Volume 3: Verification

4.3.3 Guidelines: Floating LVDS Input Pins

You can implement floating LVDS input pins in Intel 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, Intel recommends that you use external biasing schemes to reduce noise injection and current consumption.

4.3.4 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.

Intel recommends that you create a Intel Quartus Prime design, specify your device I/O assignments, and compile your design to validate your pin placement. The Intel Quartus Prime 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 Intel Quartus Prime 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 Intel Quartus Prime software automatically assigns the negative pin if the positive pin is assigned with a differential I/O standard.

In Intel 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 routes of both pins in a differential pair are matched even if the pins are non-adjacent.

The Altera Soft LVDS IP core for Intel MAX 10 devices supports a maximum of 18 channels per IP instantiation. Each channel can support deserialization factor (parallel data width) from one to ten bits. When you are grouping channels for an application, you must consider the channel to channel skew during Fitter placement. To minimize

skew, place all LVDS channels in the group side by side. For your PCB design, Intel recommends that you perform package skew compensation to minimize skew and maximize performance.

Note: For Intel MAX 10 devices, the Intel Quartus Prime software does not provide a package skew compensation report.

4.3.5 Guidelines: LVDS Channels PLL Placement

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

Table 10. Examples of Usable PLL to Drive I/O Banks in Intel MAX 10 Devices

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

4.3.6 Guidelines: LVDS Receiver Logic Placement

The Intel Quartus Prime 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 Intel Quartus Prime 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.

4.4 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 Intel SignalTap II Logic Analyzer to perform system level verification to correlate the system against your design targets.

4.4.1 Perform RTL Simulation Before Hardware Debug

Before you debug on hardware, Intel 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.

4.4.2 Geometry-Based and Physics-Based I/O Rules

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

For more information, refer to the related information.

Related Links

Intel MAX 10 General Purpose I/O User Guide

5 Intel MAX 10 LVDS Transmitter and Receiver Design

You can implement mixed transmitter and receiver applications using the Intel 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.

5.1 Transmitter–Receiver Interfacing

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

Figure 26. Typical Altera Soft LVDS Interfaces with Internal PLL

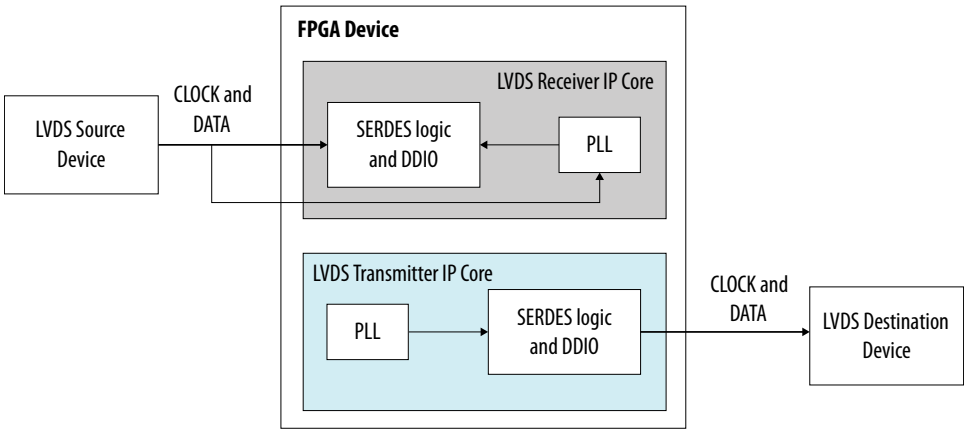
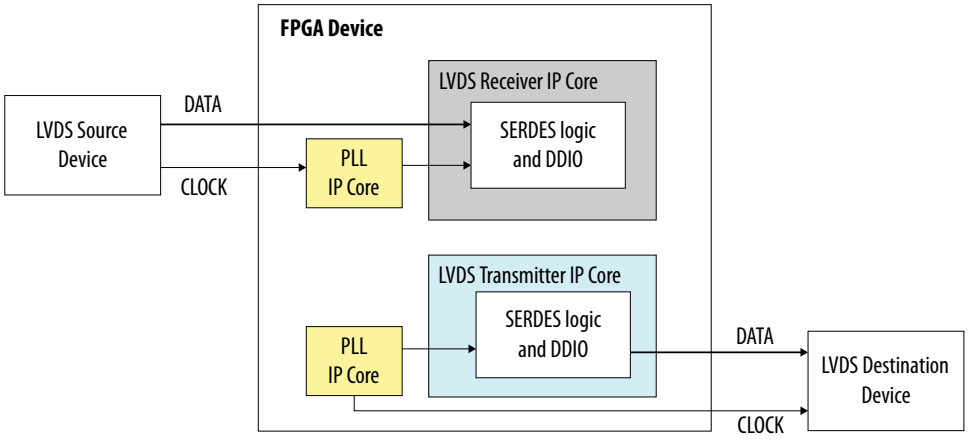


Figure 27. Typical Altera Soft LVDS Interfaces with External PLL



Related Links

- Intel MAX 10 LVDS Transmitter Design on page 16
  Provides more information about specific features and support of the LVDS transmitters.
- Intel MAX 10 LVDS Receiver Design on page 26
  Provides more information about specific features and support of the LVDS receivers.

5.2 LVDS Transmitter and Receiver FPGA Design Implementation

Intel MAX 10 devices use a soft SERDES architecture to support high-speed I/O interfaces. The Intel Quartus Prime 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, Intel MAX 10 devices use the I/O registers and LE registers in the core fabric.

5.2.1 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 Intel Quartus Prime 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 Intel MAX 10 packages. Intel recommends that you select a Intel MAX 10 device package that provides sufficient number of PLL clockouts for your design.

5.2.2 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.

5.3 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 Intel SignalTap II Logic Analyzer to perform system level verification to correlate the system against your design targets.

5.3.1 Perform RTL Simulation Before Hardware Debug

Before you debug on hardware, Intel 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.

5.3.2 Geometry-Based and Physics-Based I/O Rules

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

For more information, refer to the related information.

Related Links

Intel MAX 10 General Purpose I/O User Guide

6 Intel MAX 10 High-Speed LVDS Board Design Considerations

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

6.1 Guidelines: 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.

6.2 Guidelines: Control Channel-to-Channel Skew

For the Intel 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 receivers.

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.

For information about controlling the LVDS I/O and package skew, refer to the related information.

Related Links

- Receiver Input Skew Margin on page 34
- RSKM Equation on page 34
  Explains the relationship between the RSKM, TCCS, and SW.
- Example: RSKM Calculation on page 36

6.3 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.

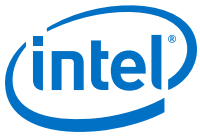
Related Links

Board Design Guidelines Solution Center
Provides resources related to board design for Intel devices.

6.4 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, for example, to compensate for 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 Links

- Intel IBIS Models
  Provides IBIS models of Intel devices for download.
- Intel HSPICE Models
  Provides SPICE models of Intel devices for download.
- IBIS Model Generation
  Provides video that demonstrates how to generate IBIS file using the Intel Quartus Prime software.

7 Altera Soft LVDS IP Core References

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

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

7.1 Altera Soft LVDS Parameter Settings

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

Table 11. Altera Soft LVDS Parameters - General

Parameter	Condition	Allowed Values	Description
Power Supply Mode	—	• Dual Supply • Single Supply	Specifies whether the target device is a single or dual supply device.
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, 5, 6, 7, 8, 9, 10	Specifies the number of bits per channel.

Table 12. Altera Soft LVDS Parameters - PLL Settings

Parameter	Condition	Allowed Values	Description
Use external PLL	Not applicable for x1 and x2 modes.	• On • Off	Specifies whether the Altera Soft LVDS IP core generates a PLL or connects to a user-specified PLL.
Data rate	—	Refer to the device datasheet.	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.

continued...

Parameter	Condition	Allowed Values	Description
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	Use external PLL = Off	• On • Off	If turned on, enables the `pll_areset` port in internal PLL mode. In external PLL mode, the `pll_areset` port is not available.
Enable tx_data_reset port	• General, Functional mode = TX • Use external PLL = On	• On • Off	If turned on, enables the `tx_data_reset` port.
Enable rx_data_reset port	• General, Functional mode = RX • Use external PLL = On	• On • Off	If turned on, enables the `rx_data_reset` port.
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 13. Altera Soft LVDS Parameters - Receiver Settings

Parameter	Condition	Allowed Values	Description
Enable bitslip mode	General, Functional mode = RX	• On • Off	If turned on, enables the `rx_data_align` port.
Enable independent bitslip 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.

continued...

Parameter	Condition	Allowed Values	Description
Enable rx_data_align_reset port	• General, Functional mode = RX • Enable bitslip mode = On • Enable independent bitslip 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 bitslip 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.
Bitslip rollover value	• General, Functional mode = RX • Enable bitslip 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.
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 14. 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.

continued...

Parameter	Condition	Allowed Values	Description
			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	Depends on **SERDES factor**.	Specifies that the frequency of the `tx_outclock` signal is equal to the the transmitter output data rate divided by the selected division factor.
Outclock duty cycle	• General, Functional mode = TX • PLL Settings, Use external PLL = Off • Enable 'tx_outclock' output port = On	Depends on **SERDES factor** and **Tx_outclock division factor**.	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.
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_corecloc k	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 Links

- Altera Soft LVDS IP Core in Transmitter Mode on page 20
- Altera Soft LVDS IP Core in Receiver Mode on page 31

7.2 Altera Soft LVDS Interface Signals

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

Table 15. Transmitter Interface Signals

Signal Name	Direction	Width (Bit)	Description
pll_areset	Input	1	Asynchronously resets all counters to the initial values.
tx_data_reset	Input	<n>	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	<m>	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	<n>	Serialized LVDS data output signal of <n> channels. tx_out[(<n>-1)..0] drives parallel data from tx_in[(<J> × <n>)-1 ..0] where <J> is the serialization factor and <n> is the number of channels. tx_out[0] drives data from tx_in[(<J>-1)..0]. tx_out[1] drives data from the next <J> number of bits on tx_in.
tx_outclock	Output	1	External reference clock. The frequency of this clock is programmable to be the same as the data rate.

Table 16. 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. • You must externally synchronize this signal with the fast clock.
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

continued...

signal Name	Direction	Width (Bit)	Description
			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).
`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.

8 Intel MAX 10 High-Speed LVDS I/O User Guide Archives

If an IP core version is not listed, the user guide for the previous IP core version applies.

IP Core Version	User Guide
16.1	MAX 10 High-Speed LVDS I/O User Guide
16.0	MAX 10 High-Speed LVDS I/O User Guide
15.1	MAX 10 High-Speed LVDS I/O User Guide
15.0	MAX 10 High-Speed LVDS I/O User Guide
14.1	MAX 10 High-Speed LVDS I/O User Guide

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

Date	Version	Changes
December 2017	2017.12.15	• Added short descriptions about the true and emulated LVDS buffers. • Added single power supply U324 package. • Updated the topic about using the Altera Soft LVDS IP core (receiver) in external PLL mode to remove the ALTPLL IP core slow clock output (c1) connection to the Altera Soft LVDS `rx_synclock` input for even deserialization factor. The `rx_synclock` is not used by the Altera Soft LVDS IP core when the deserialization factor is even. • Updated the section about high-speed I/O timing budget: — Updated the high-speed I/O timing budget topic to clarify that Intel MAX 10 devices implements SERDES in LEs. — Removed information about obtaining RSKM report in the Intel Quartus Prime software. The software does not support generating RSKM report for Intel MAX 10 devices. — Removed the topic about assigning input delay to the LVDS receiver. — Added link to timing constraints section of the *Intel Quartus Prime Standard Edition Handbook Volume 3: Verification*. • Updated the condition and description for the **Enable pll_areset port** parameter to specify that it is not available in external PLL mode. • Throughout the document, added links to related information to improve usability. • Further edits for Intel rebranding.
February 2017	2017.02.21	Rebranded as Intel.
October 2016	2016.10.31	• Added related information links in the topic about LVDS channels support. • Restructured and updated the topic about the RSKM. • Added a topic that describes how to assign input delay to the LVDS receiver using the TimeQuest Timing Analyzer.
May 2016	2016.05.02	• Added true RSDS and emulated RSDS (three resistors) transmitter support for single supply Intel MAX 10 devices. • Updated the transmitter and receiver channels placement topics to describe about minimizing skew when you group LVDS channels for an application. • Updated the description of the `rx_data_reset` interface signal to specify that you must externally synchronize it with the fast clock. • Updated the **General** tab of the Altera Soft LVDS parameter settings: — Added the **Power Supply Mode** option. — Updated the allowed values of the SERDES factor parameter.

continued...

Date	Version	Changes
November 2015	2015.11.02	• Updated the high-speed LVDS circuitry figure to correct the flow from C1 in ALTPLL to inclock of ALTERA_SOFT_LVDS. Previously, the figure shows a bidirectional flow. • Updated the steps for determining the external PLL clock parameters for the receiver to clarify the clock names listed by the Intel Quartus Prime compilation report. • Updated the topic about Altera Soft LVDS parameter settings: — Added links to topics about PLL and high-speed I/O performance in the device datasheet. — Corrected the conditions required to use the **Enable tx_data_reset port** and **Enable rx_data_reset port** parameters. You must turn on the **Use external PLL** option first. — Updated the allowed values for the **Tx_outclock division factor** and **Outclock duty cycle** parameters. — Updated the condition for the **Desired transmitter outclock phase shift** parameter. • Removed the topics about generating IP cores and the files generated by the IP core, and added a link to *Introduction to Altera IP Cores*. • Removed the statement about getting TCCS value from the Intel Quartus Prime compilation report. You can get TCCS value from the device datasheet. • Added guidelines topic about enabling LVDS pre-emphasis for Intel MAX 10 devices in the E144 package. • Updated the guidelines to control channel-to-channel skew to remove statements about getting the trace delay amount from the Fitter Report panel. • Added link to video that demonstrates how to generate IBIS file using the Intel Quartus Prime software. • Changed instances of *Quartus II* to *Quartus Prime*.

continued...

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 Intel 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_synclock` 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.
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 Intel 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.