Multi-channel 622 MHz LVDS Data Transfer with Virtex-E Devices

XAPP233 (v1.0) December 21, 1999

Application Note: Brian Von Herzen, Ph.D. & Jon Brunetti

Summary

The Virtex-E FPGA Series provides dedicated on-chip differential receivers between adjacent user I/O pins, which are ideal for receiving LVDS signals at speeds of up to 622 Mbits/s in the -7 speed grade. This application note describes how to create a high-speed LVDS receiver and transmitter on a single Virtex-E FPGA suitable for point-to-point data transmission at a data rate of 622 MHz. The design utilizes a guide file for optimal routing.

Introduction

Low-voltage differential signaling (LVDS) has emerged as a leading standard for differential signaling between boards, chassis and other peripherals. For the first time, FPGAs are able to receive and drive data between boards at speeds of 622 Mb/s with no external buffering. The reference design described here implements a complete point-to-point link using LVDS at 622 Mb/s per data channel.

Two application notes serve as an introduction to LVDS techniques. **XAPP230: The LVDS I/O Standard** describes the basic signaling levels and requirements for LVDS signaling. **XAPP 232: Virtex-E LVDS Drivers and Receivers: Interface Guidelines** provides more detailed implementation guidelines for Virtex-E devices. These application notes serve as a starting point for designers new to LVDS.

The reference design for the LVDS 622 Mb/s receiver utilizes two incoming data channels and one incoming clock channel. The system relies on double data rate (DDR) clocking: new data is present on every transition of the clock signal. Clock and data lines have identical bandwidth requirements under this approach, making it attractive for high-speed systems. This LVDS receiver design extends the philosophy of source synchronous signaling past the board level and onto the FPGA itself.

The basic technique used in this reference design is called source synchronous signaling, which drives clock and data from a single device and forwards the clock along with the data to the destination. Clock and data propagate along adjacent paths with matched time delays.

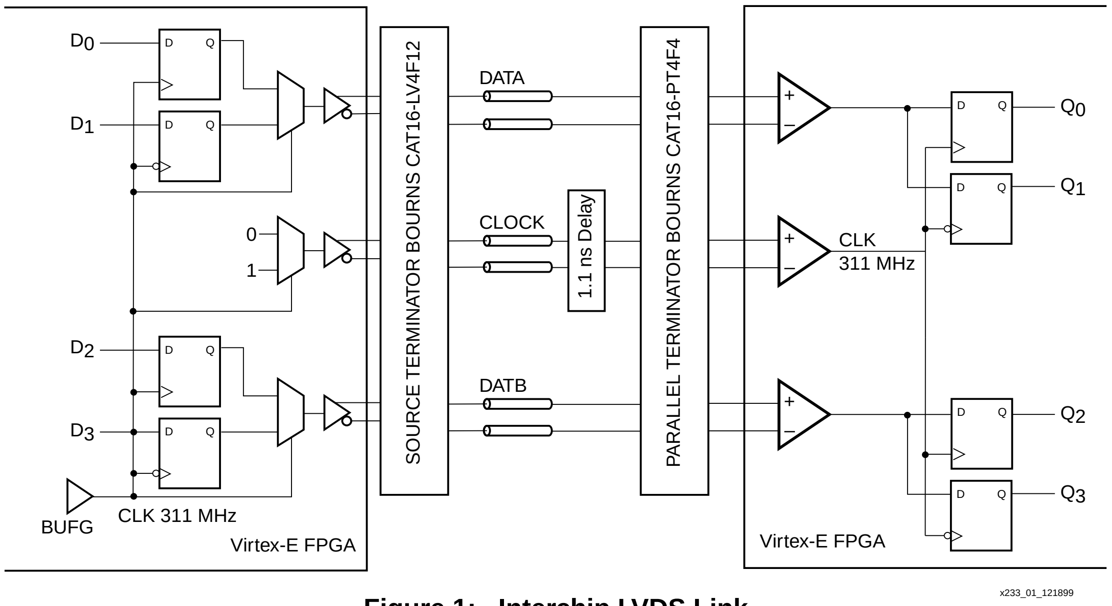


Figure 1: Interchip LVDS Link

Figure 1 shows a complete LVDS link, with two data channels running at 622 Mb/s and one clock channel running at 311 MHz. The transmitters on the left use both edges of the clock to transmit data driven by a clock multiplexer onto the LVDS channel. Clock and data delays are well matched since identical multiplexers generate both clock and data. Clock and data pass through the Virtex-E LVDS driver circuitry and off chip.

The source termination network adjusts the levels to be fully LVDS compliant and also provides source termination of the transmission lines to 50 Ω, attenuating any reflected signals going back to the driver. The transmission lines can be microstrip or stripline at 50 Ω to ground, or twisted pair with 100 Ω differential impedance. The parallel terminator generally consists of a bank of 100 Ω resistors across each differential pair. For maximum timing margin at 622 Mb/s, the clock should be delayed by 1.1 ns relative to the data, using either additional trace delay or a driver with well-characterized propagation delay.

The signals are received by the differential LVDS receivers and pass to flip-flops that sample the data on the rising and falling edge of the forwarded 311 MHz clock. Data can be further demultiplexed down to 78 MHz using a block RAM structure as shown later in this application note.

Figure 2 shows the physical structure of a single data line passing from one Virtex-E device to another. The internal structure of the termination packs are also shown.

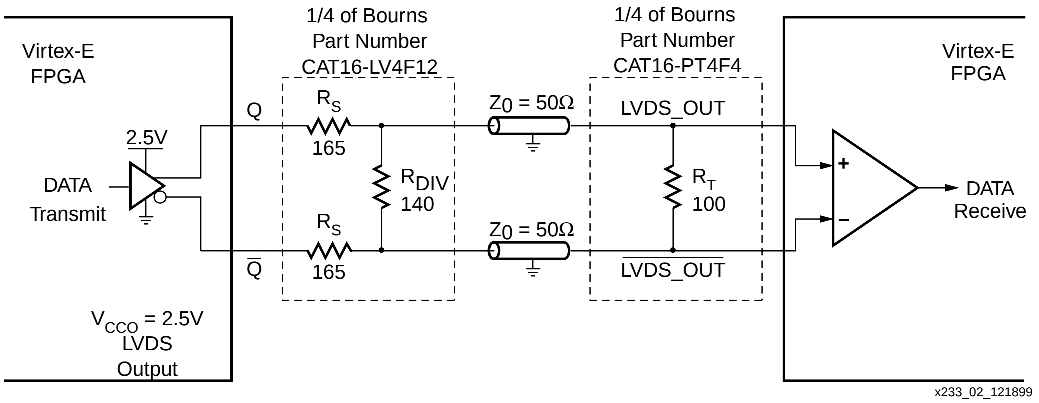


Figure 2: Virtex-E LVDS Line Driver and Receiver Schematic

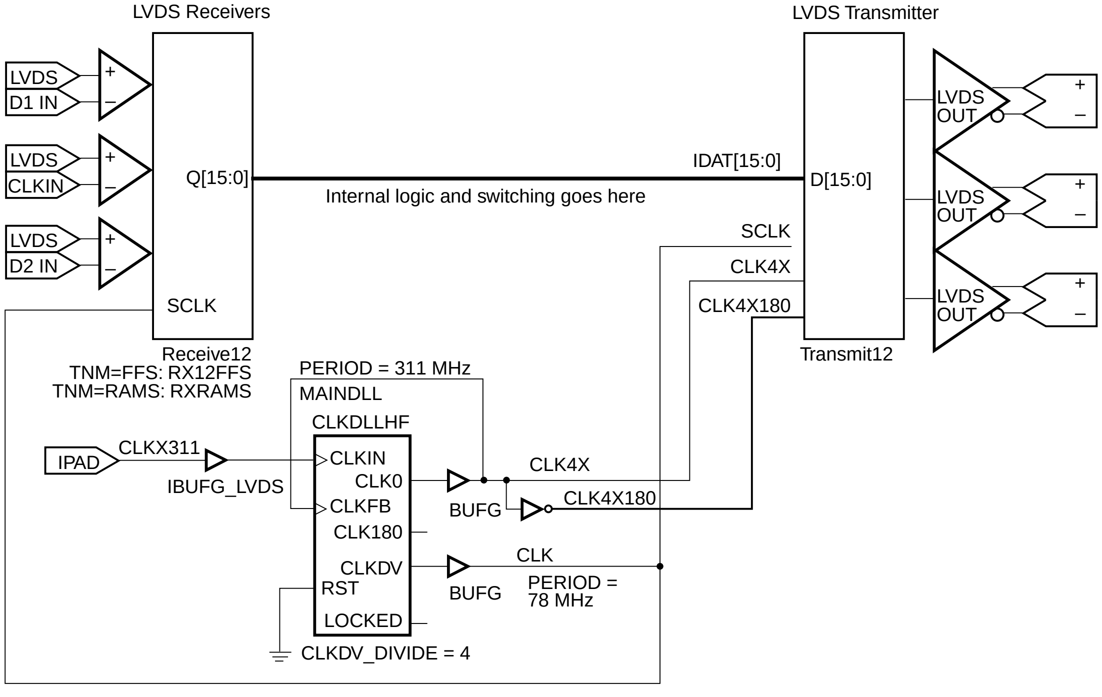


Figure 3: LVDS_ARRAY

Figure 3 (LVDS_ARRAY) shows an LVDS receiver connected to an LVDS transmitter on a single Virtex-E FPGA. The receiver accepts two data channels at 622 Mb/s each and one clock channel at 311 MHz. The clock channel is located between the two data channels to equalize the sampling time on the two data channels. Placing the clock in the middle provides equal distance to both sampling channels in the FPGA. Figure 4 shows the internal design of the receiver block, and Figure 5 shows the internal design of the transmitter block.

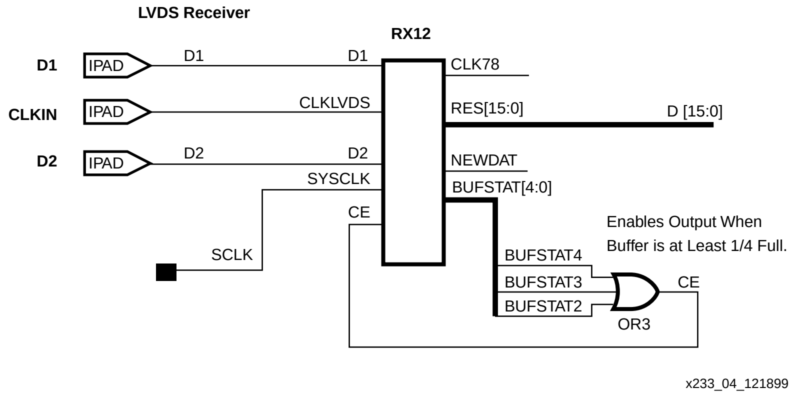


Figure 4: Receiver Block

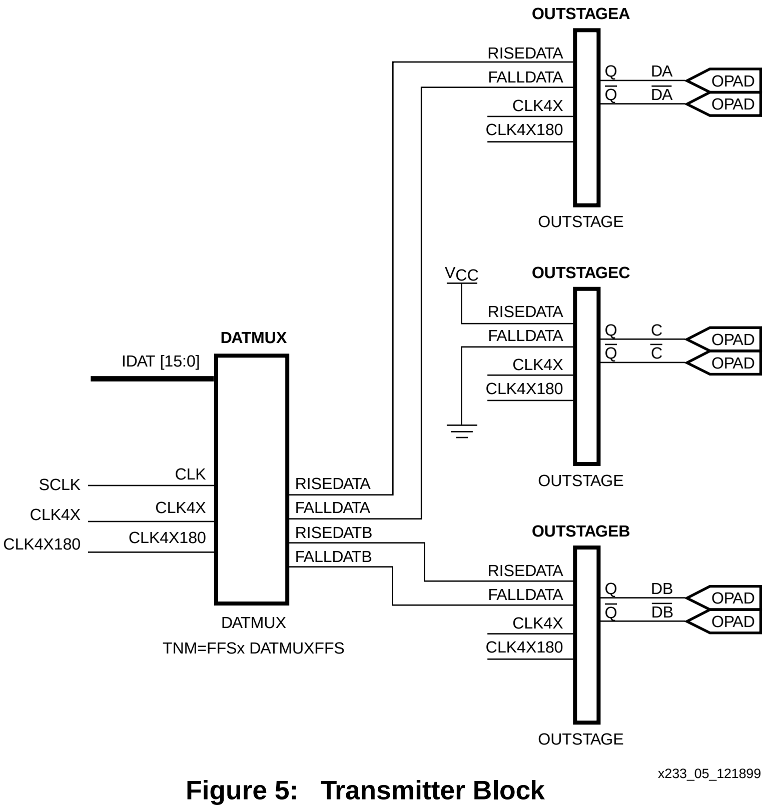


Figure 5: Transmitter Block

Utilizing the Reference Design

A reference design is available at ftp://ftp.xilinx.com/pub/applications/xapp/xapp233.zip. The reference design includes source schematics, along with extracted EDIF net lists for the entire design, the receiver block and the transmitter block. The package also includes floorplan and guide files for the LVDS design, essential to achieving timespecs for 622 MHz operation.

The reference design implements a single receiver and a single transmitter block on a V300E-BG432 device. To implement the reference design, the following procedure can be used:

1. Unpack the zip file and located the LVDS_ARRAY.EDN file.
2. Implement the design using lvds_array.edn as the source file with the following parameters:
   A. The package to choose for implementation is V300E-7BG432C.
   B. No constraints file is needed.
   C. Set the floorplan file to lvds_array.fpn
   D. Set the floorplan guide file to lvds_array.mfp
   E. Set the guide file to guide4.ncd
3. Run the implementation (the default place and route level of 2 is ok), and verify that all the timespecs are met.

The bitgen warnings on LVDS refer to TTL vs. LVDS banking. These warnings may not be an issue if serial proms are not being used. A zipped copy of the implementation files are included for comparison purposes.

An alternative way to utilize these blocks is to instantiate them in the design source and include the receive12.edn file for the receiver and transmit12.edn for the receiver. These blocks should be located at the top level of the design and should have instance names "R" and "T", respectively. This will enable the mapper and placer to recognize the instance and net names and properly locate them in the implementation.

Implementation Notes

Figure 6 (RX12), shows the internal elements of the LVDS receiver, with a two-channel sampler on the left and the block RAM buffer on the right. Data is received at 622 Mb/s from the pins into the RX12 block. It is demultiplexed to 155 Mb/s and gets written 8 bits wide into a block RAM, where it crosses the time domain. It is read out 16 bits wide at 78 MHz (or more). In Figure 7 (SAMP2CH) the samplers for data channels D1 and D2 are on the left, followed by demultiplexing to reduce the data rate to 155 Mb/s. Figure 8 (SAMPLE) shows the primitive sampling unit that operates in a single CLB. The data is sampled on the falling edge of the clock signal by the latch LDC and on the rising edge by the flip-flop "Q2." Data propagates through the transparent latch in 600 ps, followed by a 376 ps route time to Q1 and a 600 ps setup time through the direct data input. Note that the MAXSKEW and MAXDELAY values in the design files serve to illustrate what is achievable in the reference design, and do not represent requirements for the design to function. The outputs Q1 and Q2 both change on the rising edge of the input clock, once every 3.2 ns. Note that the DUMMY_LOAD signal equalizes delays to the FDC and the LDC. When the DUMMY_LOAD is included in proper location in the guide file, the delays can be made to match to within a few picoseconds according to the TRCE program. See the guide file *guide.ncd* in the reference design XAPP233.zip for exact timings.

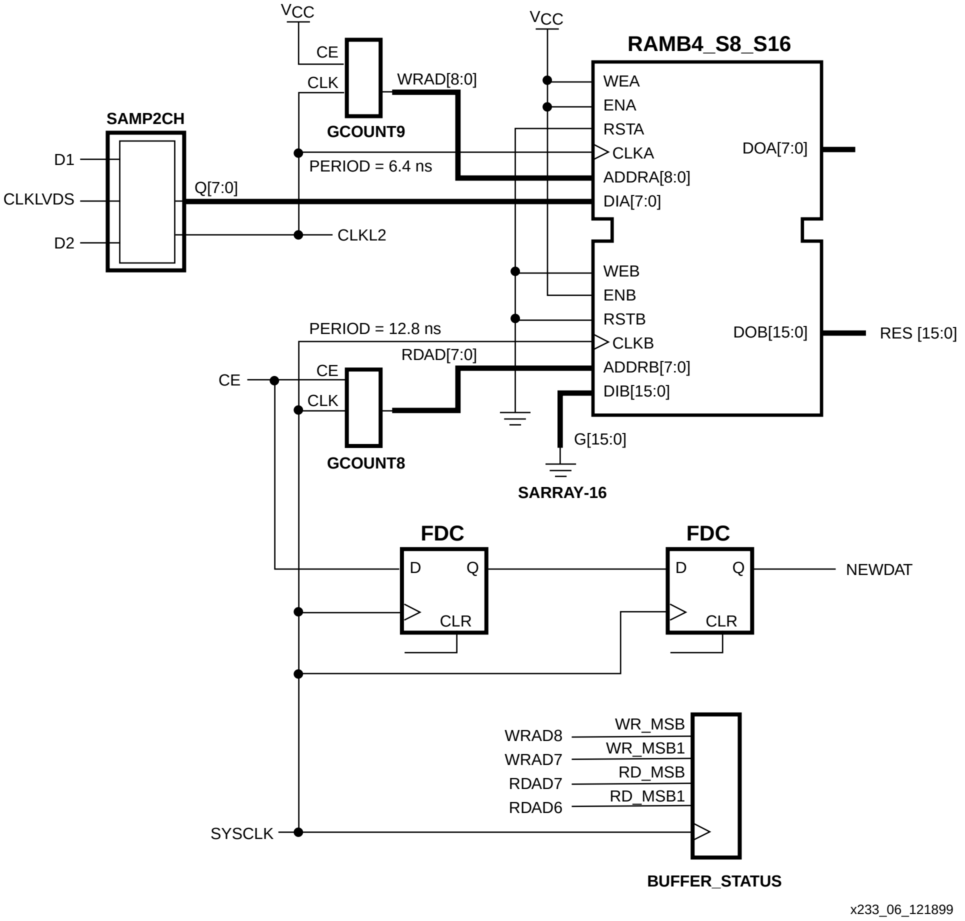


Figure 6: RX12

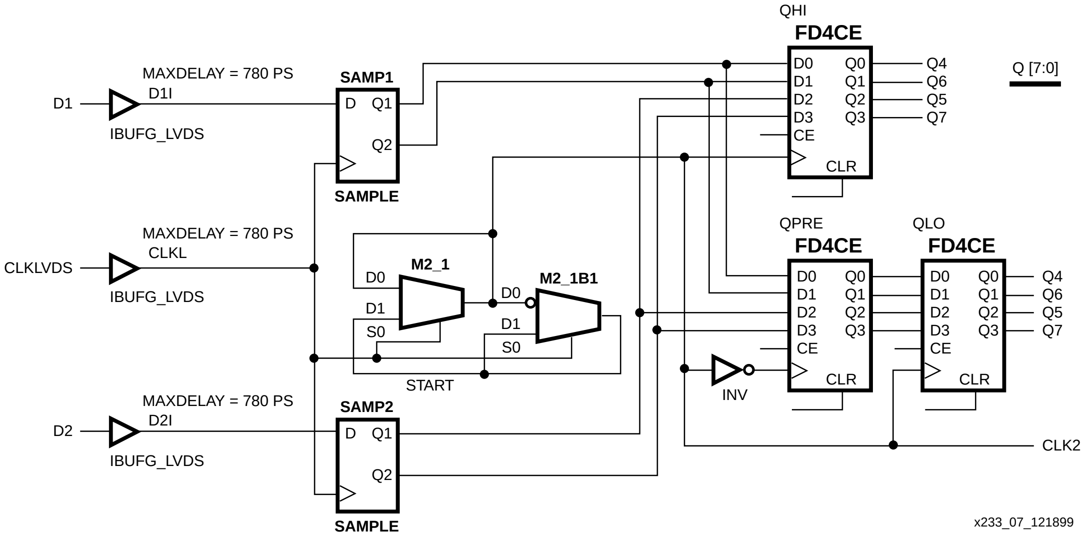


Figure 7: SAMP2CH

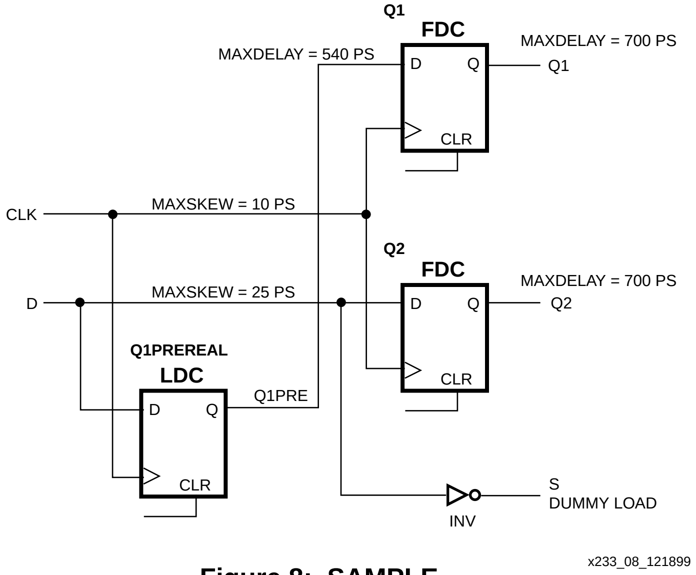


Figure 8: SAMPLE

SAMPLE Timing Analysis

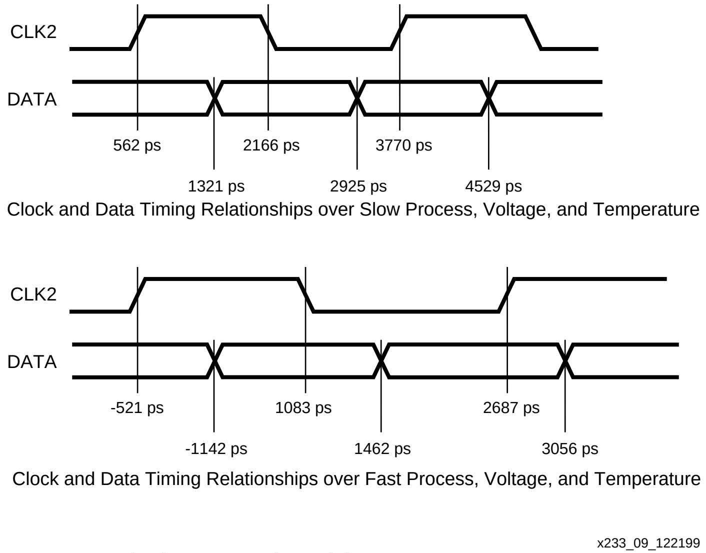
What offset of incoming clock and data will maximize the timing margin? The timings from TRCE indicate a worst-case clock propagation delay of 732 ps for D1 and 753 ps for D2, assuming a design utilizing the Virtex-E -7 speed grade. Data delays vary from 765 to 777 ps depending on the input. The data input to the CLB has a setup time of 600 ps and a hold time of – 245 ps. Maximal sensitivity to data occurs half way from the setup to the hold time, or – 420 ps. The least sensitivity occurs a half cycle later (802 ps at 622 MHz) for a data transition time of 382 ps after the clock. Under slow process, voltage and temperature (PVT) the clock occurs 59 ps after the data. Under fast PVT the clock will occur roughly 30 ps after the data. See Figure 9 for a timing diagram comparing clock to data relationships. Using an intermediate value of 45 ps, we would see maximal timing margin with 382 + 45 = 427 ps offset from clock to data. At a typical propagation delay of 170 ps per inch, a clock line 2.5" shorter than the data lines will produce the desired timing offsets for optimal margin. Given that the setup and hold window is less than 400 ps long, we have more than ± 600 ps of timing margin under this phase condition, out of a data period of 1604 ps. As an alternative, the clock line can be made 1100 ps longer than the data lines at a data rate of 622 Mb/s, with the same resulting timing margins. The optimal time differences for maximum timing margin are a function of the operating frequency, as illustrated in Figure 9.

SAMP2CH Timing Analysis

Popping up one level to Figure 7 (SAMP2CH), the two data channels are sampled and split and are sent to four-bit wide registers QHI and QPRE. These registers are clocked by CLKL2. The CLKL2 frequency has been divided in half using a special LUT prescaler. The purpose of the LUT prescaler is to provide a short propagation delay from the IBUF clock output to the CLKL2 output. The direct route is available from the IBUF to the LUT of only 487 ps propagation delay, and the LUT itself is only 479 ps for a total of 966 ps from CLKL output to CLKL2 output. If this prescaler had been implemented in a conventional flip flop, the propagation delay would have been 1000 ps of routing and 900 ps of clock to output delay for a total of 1900 ps. This additional delay would have worsened the race condition between CLKL2 and data to the 4-bit registers, so a LUT prescaler was implemented instead.

QHI samples on the rising edge of CLKL2 while QPRE samples on the falling edge of CLKL2. There is a potential race condition between clock and data to the four-bit registers. In practice the delays are matched using similar local routing structures. Assuming worst-case data propagation, we have the following timings for clock and data to the QHI and QPRE registers. Note that data is sourced from SAMP1 and SAMP2, which are triggered by the clock signal.

Referring to Figure 7 (SAMP2CH), the parallel paths start at the CLKL source. Data requires 750 ps of CLKL propagation, 886 ps clock to output time, 689 ps route delay to the QHI and QPRE, and 600 ps setup time. The total max delay for data is 750 + 886 + 689 + 600 = 2925 ps. In comparison, the clock path has the following delays: 487 ps CLKL propagation, 479 ps T_{ILO} through the lookup table and 1200 ps through net CLKL2. The total worst-case delay for clock is 2166 ps. Under the worst case, the clock transitions at t = 562 ps, t = 2166 ps and t = 3770 ps. Data transitions at t = 1321, t = 2925, and t = 4529. Figure 9 (CLKL2 Timing over Process Variations) shows the temporal relationships of these transitions, and how they are preserved over the full range of process, voltage and temperature (PVT). These phase relationships provide relatively large timing margins between clock and data, and are improved when the negative hold time of the D flip-flop of the CLB is taken into account. Given that similar local routes are used within a 3x3 CLB block in one region of a single device, data and clock delays track closely over PVT variation. At maximum speed, minimum delays would be roughly twice as fast as maximum delays, implying a clock time of t = 1083 ps and data time of 1462 ps. Note that the order of clock and data transitions is preserved and we still get correct operation of the device over the full range of PVT. This feature of source-synchronous signaling makes it robust over PVT as long as clock and data are well-matched in delay.



Clock and Data Timing Relationships over Slow Process, Voltage, and Temperature

Clock and Data Timing Relationships over Fast Process, Voltage, and Temperature

Figure 9: CLKL2 Timing Relationships over Process, Voltage and Temperature Variations, illustrating how clock and data delays track.

BLOCK RAM UTILIZATION

The SAMP2CH design brings the data rate data down to 155 Mb/s. For designs requiring lower speeds and global synchronization, a single block RAM works well. The block RAM takes eight bits of data at 155 Mb/s and produces 16 bits of data at 78 Mb/s using the fully asynchronous second port feature. A nine-bit counter provides write addressing on the first port, while an eight-bit counter provides read addressing on the second port. The top two bits of these counters are gray-coded so that simultaneous transitions do not occur on these two bits. Gray coding permits a simple calculation to determine how full the buffer is. The buffer level is calculated in Figure 10 (BUFSTAT), which produces a four-bit one-hot vector. This encoding is identical to the FIFO status vector described in detail in XAPP131. Definitions are listed in Table 1.

Table 1: One-Hot Encoding of Buffer Status

Status Line	One-Hot Encoding Definition
BUFSTAT0	The RAMB is between empty and one-quarter full
BUFSTAT1	The RAMB is between one word and one-half full
BUFSTAT2	The RAMB is between one-quarter and three-quarters full
BUFSTAT3	The RAMB is between one-half and completely full
BUFSTAT4	The RAMB is between three-quarters and completely full

Only one BUFSTAT bit is active at any given time. The top-level circuitry in Figure 3 activates the read clock enable whenever BUFSTAT is in states 2, 3 or 4. Since the LVDS input continually fills the buffer, two Kb of latency accrues but throughput is assured since new data is always entering the buffer.

Note that the data can be read at any asynchronous frequency above 77.78 MHz, which is just fast enough to keep the RAMB from overflowing. For example, the internal system clock might run at 100 MHz. When the RAMB level gets too low, CE is lowered and a pause occurs until more data is available. The NEWDAT signal of Figure 6 is a synchronous strobe that occurs during the same cycle that new data becomes available on bus RES[15:0]. NEWDAT and RESULTS[15:0] transition during the same cycle. NEWDAT can serve as a clock enable for subsequent stages of pipeline processing.

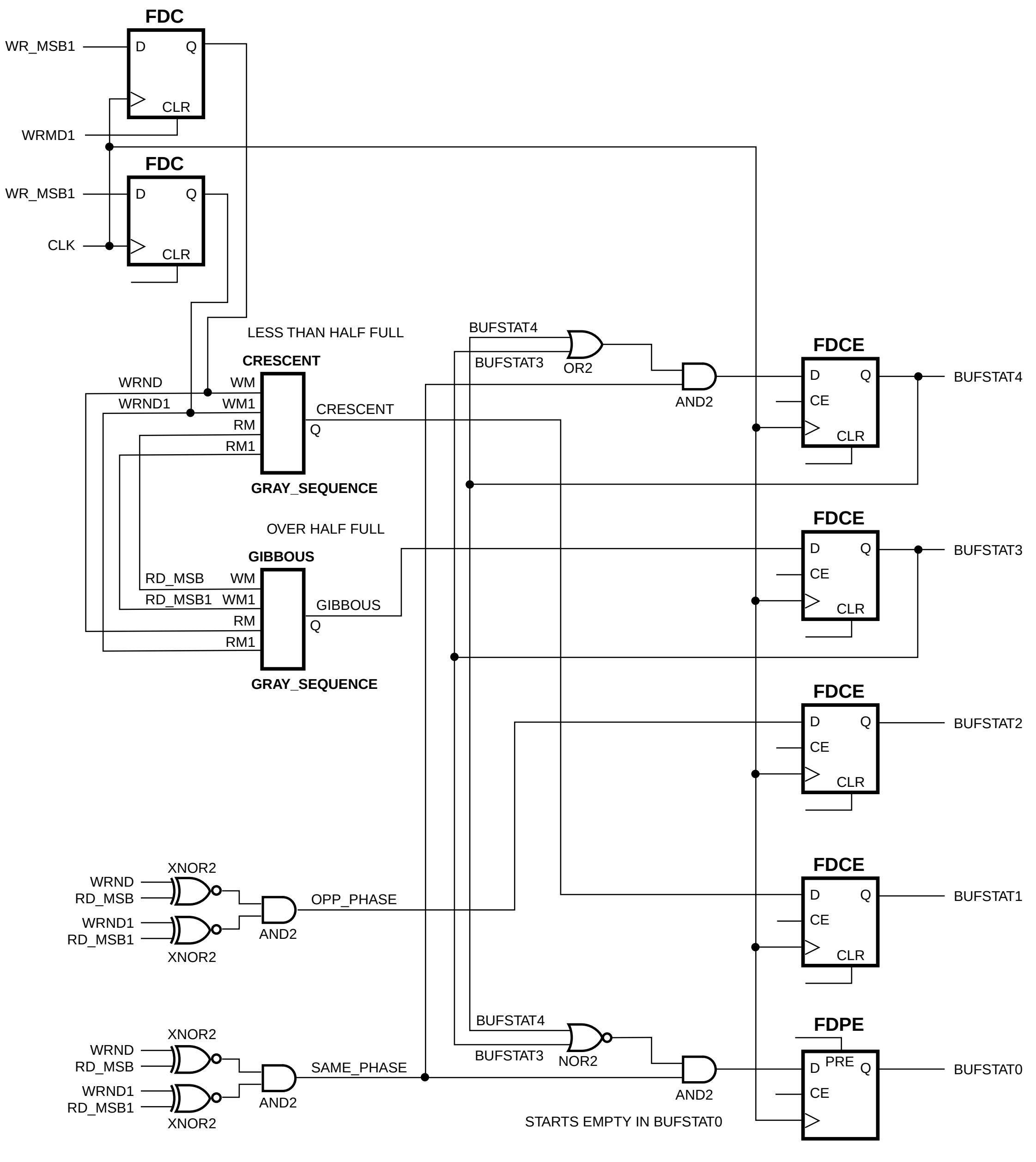


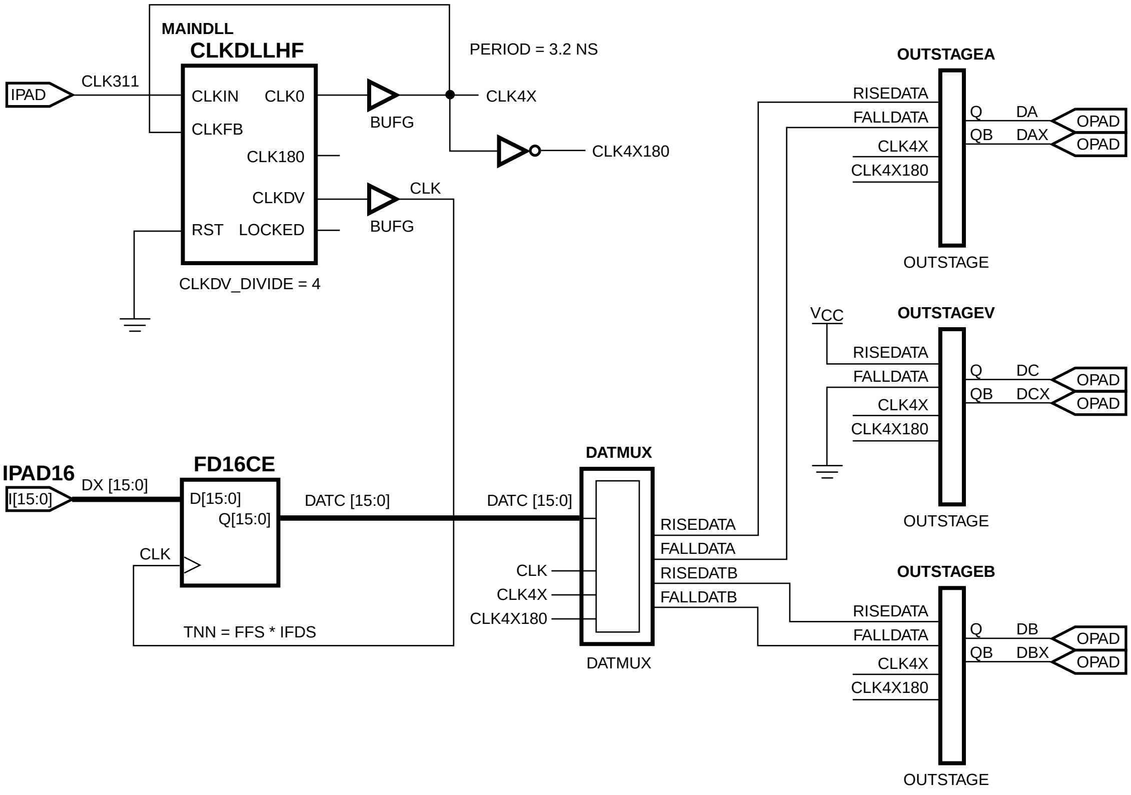
Figure 10: BUFSTAT

LVDS 622 MHz Transmitter Design

The FPGA design for transmitter takes 16 bits at 78 MHz and transmits two bits at 622 MHz plus a clock signal that toggles every 1.6 ns. This design takes the same area as the receiver block, 4 x 4 CLB's, and the transmitter and receiver blocks connect directly together, as shown in Figure 3. The transmitter does not require a block RAM for buffering, but if desired a block RAM FIFO can be inserted before the LVDS transmitter. See XAPP131 for details on the block RAM FIFO.

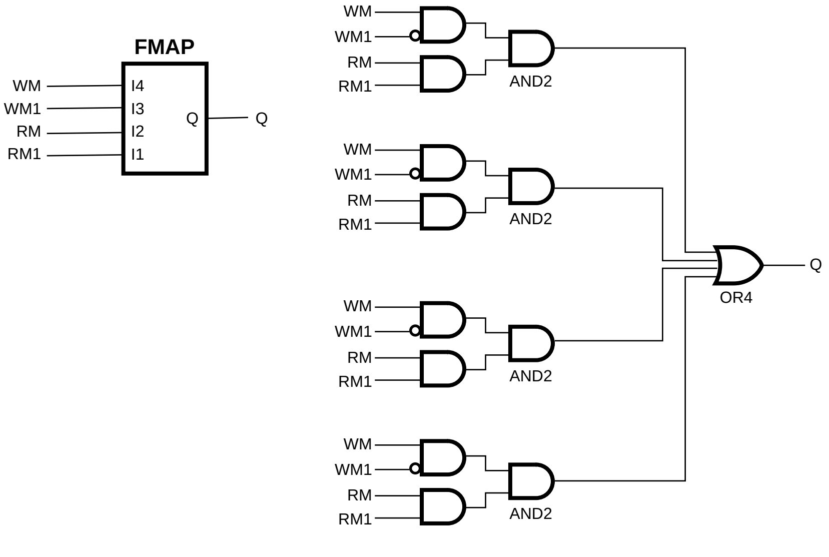
The transmitter design has been functionally simulated and all paths have been covered with the static timing analyzer. The design requires floorplanning and one or two guide routes. Direct routing is used for the fastest lines.

The transmitter design in Figure 11 is divided into three block diagrams. The block diagram in Figure 3 shows the incoming data bus of 16 bits, and an input clock of 311 MHz. A DLL divides the clock by four down to 78 MHz for the input data, and also generates the 311 MHz global clock CLK4X and its complement CLK4X180. The block DATMUX multiplexes the data up to 311 Mb/s. The OUTSTAGE blocks do the final multiplexing to 622 Mb/sec.



x233_13_121899

Figure 11: Transmitter Design Block Diagram



x233_11_121899

Figure 12: Gray_Sequence

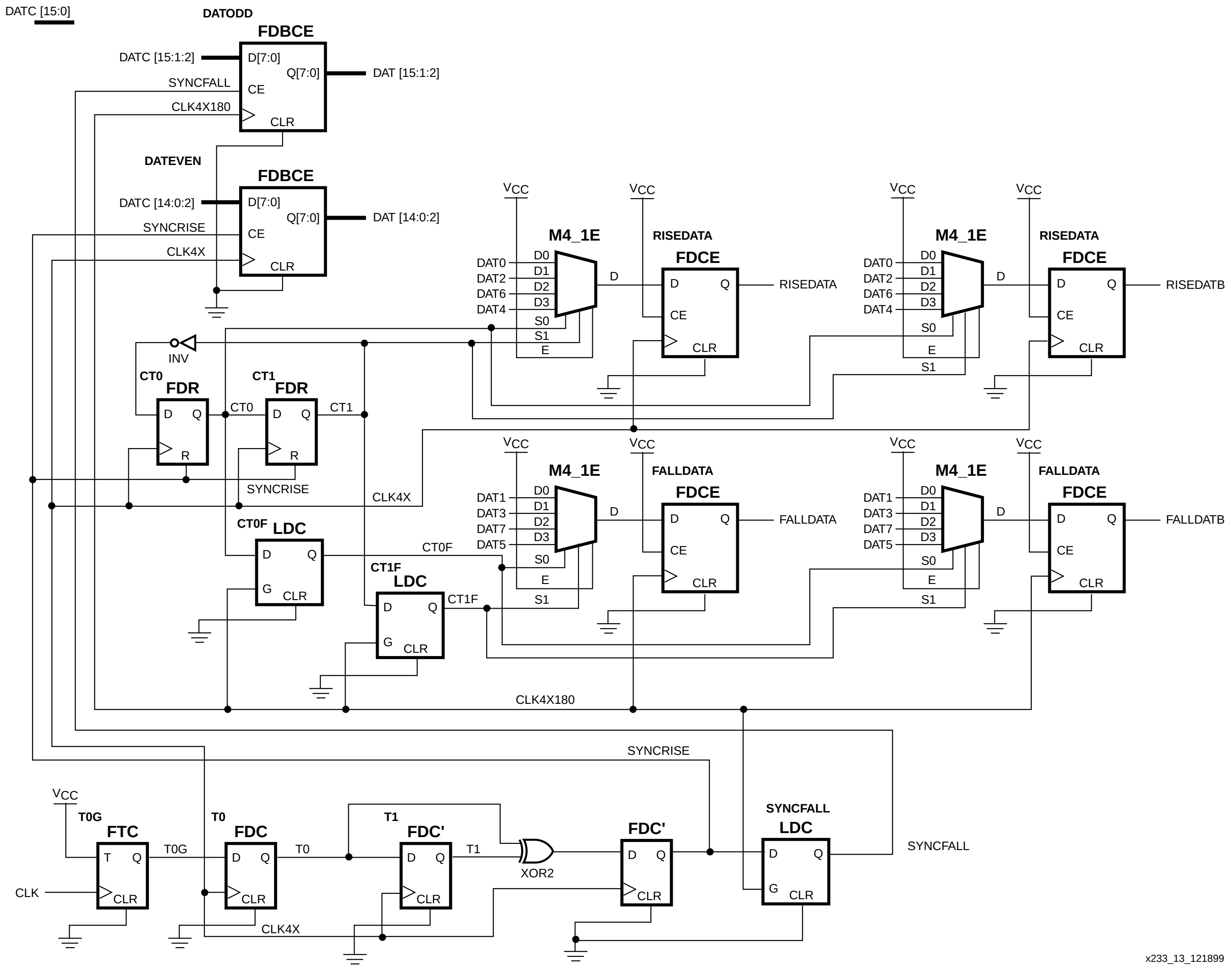


Figure 13: DATMUX

Figure 13 (DATMUX) shows the time-division multiplexing up to 311 Mb/s. Registers DATODD and DATEVEN latch the data 3.0 and 3.5 cycles after CLK rises. These latch signals are generated by the bottom row of toggle and data flip flops. SYNCRISE and SYNCFALL are the enable signals with a 1 out of 4 duty cycle. They also serve to reset the gray-code counter CT0 and CT1. The gray-code addresses the multiplexer feeding the RISEDATA and RISEDATB registers. A simple gray-code sequence is shown in Figure 12. A latched version CT0F and CT1F addresses muxed data on the complementary 311 MHz clock, generating FALLDATA and FALLDATB. These data signals feed OUTSTAGEA and OUTSTAGEB on TRANSMIT12.

Identical output stages drive data bits A, B and clock C. These stages are placed immediately adjacent to each other on the FPGA for optimal matching and delay tracking. This placement assures synchronized clock and data generation, with the placement constrained to identical structures on the FPGA edge. Note that the clock structure is identical to the data except that RISEDAT is always one and FALLDAT is always zero. Using an identical structure and placement assures similar delays between clock and data.

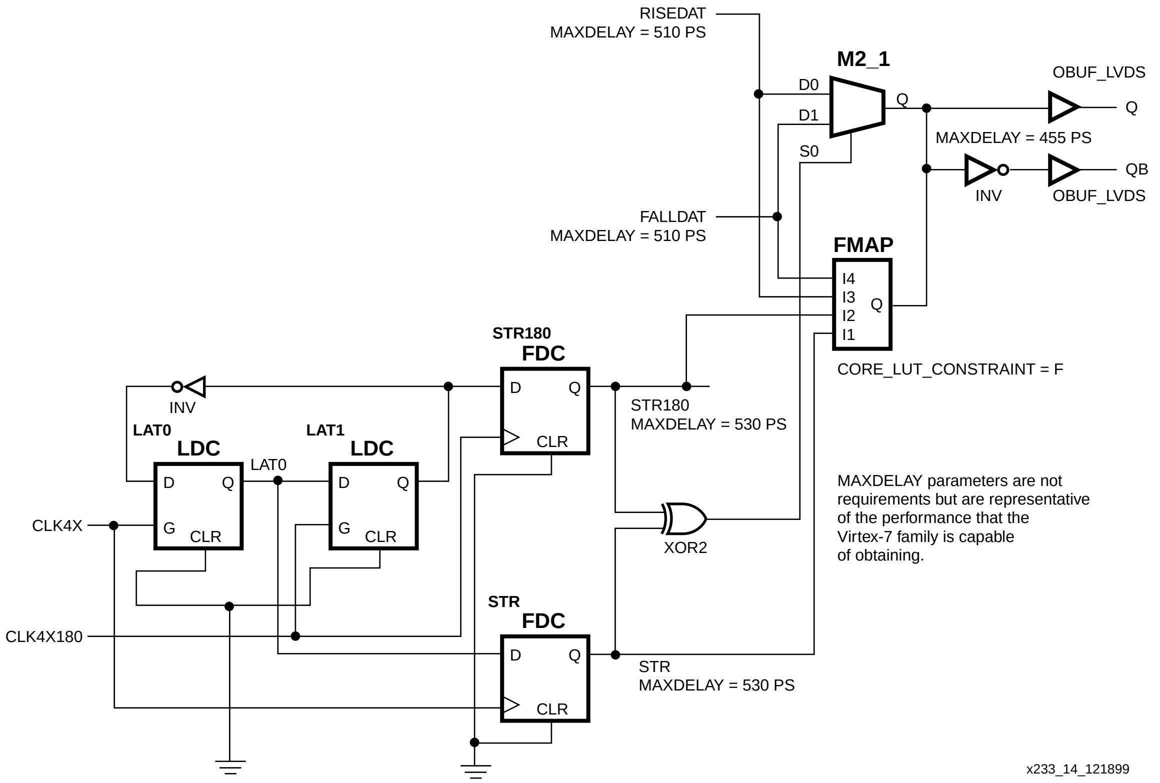


Figure 14: OUTSTAGE

Figure 14 (OUTSTAGE) shows the block diagram for the last stage of multiplexing from 311 to 622 Mb/s. Incoming data RISEDAT arrives on the rising edge of CLK4x, while FALLDAT transitions on the rising edge of CLK4X180. The two latches generate data strobes on the rising edges of CLK4x and CLK4X180. The XOR gate generates a control signal that transitions every 1.6 ns. This signal times the transitions between data, and is closely delay matched for rising and falling transitions to within 10 ps under the TRCE static timing analyzer program. RISEDAT and FALLDAT arrive slightly before the strobe signals and provide the incoming data. The output is wave pipelined, spending roughly 500 ps in routing, 500 ps in the lookup table, and 500 ps going to the IOBs. A direct-connect routing provides minimum delay from CLB to IOB if west or east chip edges are used. The output LVDS drivers provide true and complement output signals to the pins.

The transmit design achieves all its timing objectives simply using a floorplan that constrains the components to particular slices. The router meets timespecs in a -7 speed grade Virtex-E device using level 2 routing effort. A guide file is provided for the transmitter and receiver design.

Design Implementation Methods

The design was implemented using the guide file guide.ncd included in the reference design. The receiver design is contained in a block four CLBs high and four CLBs wide, pitch-matched to the block RAM structure. The transmitter uses the same area and is also pitch-matched to the block rams. In principle it is possible to tile multiple LVDS dual-channel receivers or transmitters, one for every block RAM near the left or right edges of the Virtex-E device. A Virtex 300E could implement up to 24 LVDS channels at 622 Mb/s/channel using this method. The sampler is very sensitive to routing changes, and identical routing should be used for each copy of the LVDS receiver. Only the critical routes are included in the guide file. The non-critical routes can be done automatically by the Xilinx auto-router.

The design was implemented using version 2.1i software, service pack 3, using the floorplan and guide files, and met all timing specifications. The table of timing results appears in the following table:

Table 2: Reference Design Timing Results

Net Timing Constraint	(ns)	Worst Case Timing (ns)
CLK4X	3.215	2.985
R/RX12/CLK12	1.5	1.392
R/RX12/CLK12	6.452	5.305
CLK	12.903	5.617
R/RX12/SAMP2CH/$1N3	0.700	0.535
R/RX12/SAMP2CH/$1N28	0.700	0.535
R/RX12/SAMP2CH/$1N1	0.700	0.440
R/RX12/SAMP2CH/$1N2	0.700	0.440
R/RX12/SAMP2CH/CLKL	0.780	0.753
R/RX12/SAMP2CH/SAMP1/Q1PRE	0.540	0.376
R/RX12/SAMP2CH/D1I	0.025	0.013
R/RX12/SAMP2CH/D1I	0.780	0.778
R/RX12/SAMP2CH/SAMP2/Q1PRE	0.540	0.376
R/RX12/SAMP2CH/D2I	0.025	0.013
R/RX12/SAMP2CH/D2I	0.780	0.778
T/FALLDATA	0.420	0.420
T/RISEDATA	0.420	0.192
T/OUTSTAGEA/STR	0.420	0.407
T/OUTSTAGEA/STR180	0.420	0.420
T/OUTSTAGEA/Q622	0.610	0.607
T/OUTSTAGEC/STR	0.420	0.407
T/OUTSTAGEC/STR180	0.420	0.420
T/OUTSTAGEC/Q622	0.610	0.607
T/FALLDATB	0.420	0.420
T/RISEDATB	0.420	0.192
T/OUTSTAGEB/STR	0.420	0.407
T/OUTSTAGEB/STR180	0.420	0.420
T/OUTSTAGEB/Q622	0.610	.607
TS02 = MAX DELAY from TIMEGRP Latches to TIMEGRP FFS	3.2	2.76
TS03 = MAX DELAY from TIMEGRP LATCHES to TIMEGRP LATCHES	2.10	2.10
TS05 = MAX DELAY from TIMEGRP RXRAMS to TIMEGRP DATMUXFFS	9.6	7.019

Board-Level Considerations

Figure 15 shows a possible implementation of a 32-channel LVDS driver in a BG432 package routed through the source terminator packs to a pair of twisted-pair ribbon cables to a receiving board with parallel termination packs and a BG432 receiver with 32 LVDS channels. Standard termination packs are available from Bourns and other vendors that provide source and destination termination networks with either 8 or 16 pins per pack.

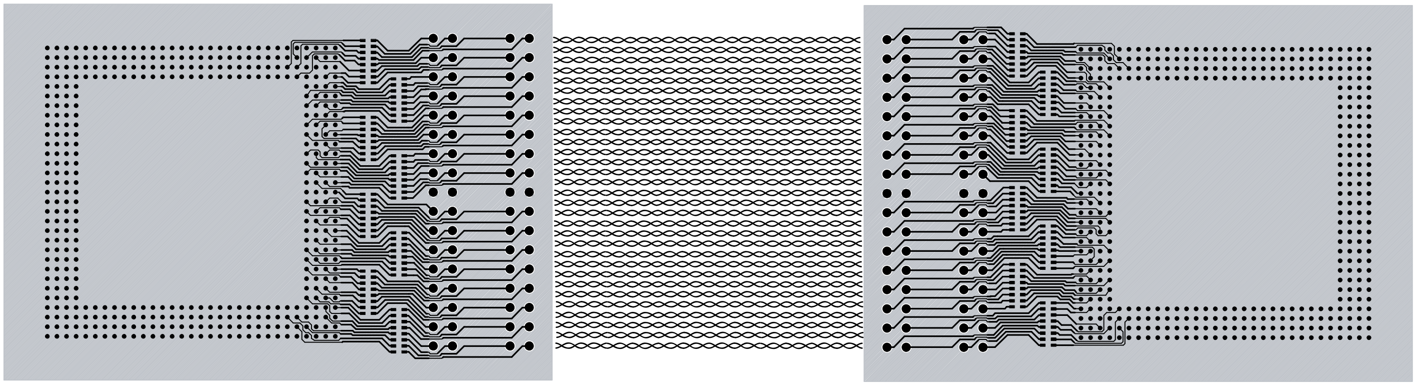


Figure 15: Board Layout

Note that all of the channels are river-routed on the surface layer with no routing vias required. This routing provides the best signal integrity for the LVDS transmission lines and should be used when possible. Other printed-circuit board layout guidelines for the LVDS circuits in Figure 2 and Figure 15 are as follows:

1. A multi-layer printed-circuit board with controlled transmission line impedances is required.
2. All transmission lines between LVDS drivers and receivers should be referenced to a common ground plane except when routed through a balanced differential transmission line such as twisted-pair. For twisted-pair and other balanced lines, a grounded shield allows for common-mode return current. The shield should connect to the ground planes at the beginning and ending of the twisted-pair cable. If no shield connection is available, it is important to take extra care to use symmetric and equal-length routing. Balancing the capacitive load on the differential pair will reduce the conversion of common-mode noise to differential signal. The ground plane should have no breaks under the signal path to avoid large discontinuities from increased inductance
3. Resistors R_S and R_{DIV} should lie close to the Virtex-E outputs for the Virtex-E LVDS line driver. The parallel termination resistor R_T needs to be close to the LVDS inputs at the destination.
4. The LVDS signal lines should have equal length with symmetric routing between source and destination to maximize common-mode rejection. The two LVDS signals should run close together on the PC board. If the trace spacing is less than the dielectric thickness to the ground plane, differential impedance effects must be included to determine the effective transmission line impedance since the trace impedance will be significantly affected by the differential impedance between the two traces. Wider spacings have a smaller effect on the impedance.
5. The differential LVDS output should use a pair of adjacent Virtex-E pins, preferably in the same output block of the Virtex-E FPGA (see FPGA Editor plots for block clustering). The LVDS data must have a single clock driving both of the output IOBs to minimize output skew between the two pins.

Note in Figure 2 the 100 Ω differential parallel termination resistor R_T across the LVDS_OUT and LVDS_OUT outputs at the end of the transmission line. This is the standard LVDS termination. Resistors R_S and R_{DIV} attenuate the signals coming out of the Virtex-E LVDS drivers with V_{CCO} = 2.5V and provide a matched source impedance (series termination) to the transmission lines. The output common-mode voltage is approximately equal to $V_{CCO}/2$. Component value derivations for R_S and R_{DIV} are found in **XAPP 232: Virtex-E LVDS Drivers and Receivers: Interface Guidelines, Appendix A.** The Virtex-E LVDS driver meets or exceeds all of the LVDS specifications listed in **XAPP230: The LVDS I/O Standard**.

The 50 Ω source impedance of the Virtex-E LVDS driver absorbs nearly all differential reflections from the capacitive load at the LVDS destination, which reduces standing waves, undershoot, and signal swing reduction on data bursts or clocks.

The 50 Ω transmission lines of Figure 2 can be implemented using microstrip or stripline transmission techniques. Figure 16 shows a microstrip geometry for a 50 Ω line on FR-4 PCB material. Note that the circuit of Figure 2 includes source and destination terminations. These terminations absorb reflections going either direction and serve to minimize reflected noise and attenuate spurious signals.

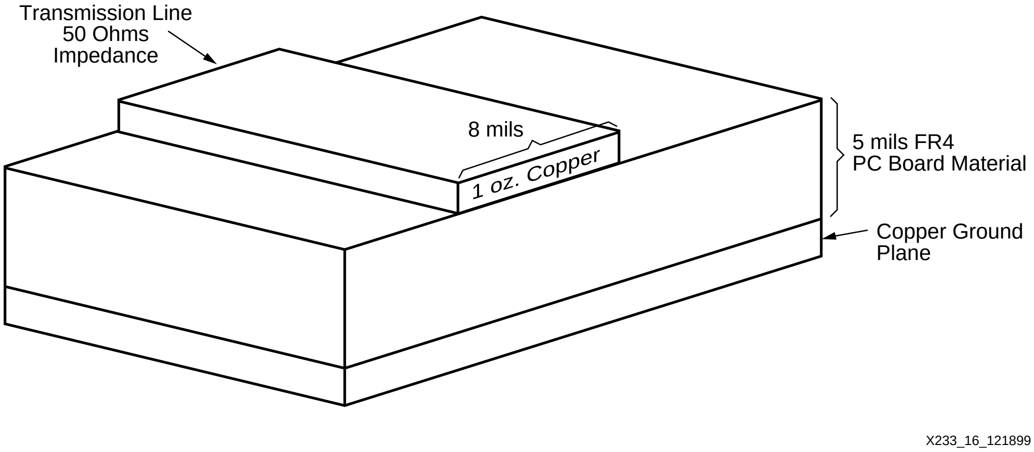


Figure 16: A 50 Ω transmission line construction in microstrip. This simple design is all that is required for microstrip.

CABLING ISSUES

If the LVDS signals are used over cabling, crosstalk minimization is imperative, especially for longer cables. The first observation is that the skew in the cabling from a clock channel to a grouped data channel must be less than the 500 ps timing margin. Remember that the clock path must be 500 ps shorter than the data paths. The source or destination PC board trace could include the shorter clock path, or the clock cable could be shorter than the data.

The cabling can be co-axial, twisted pair or other controlled impedance cable. The cabling can be 50 Ω cables referenced to ground or 100 Ω differential impedance cable.

Cross-talk is minimized with co-axial cable. The next best cable is CAT5, which uses twisted pairs, usually four pairs per cable. The cable is design to have a varying number of twists for each pair, further reducing cross-talk. Twisted-pair ribbon cabling can also be used, but suffers from increased cross-talk because the number of twists per inch is the same for all the pairs. Flat ribbon cable will suffer from more cross-talk, but may be usable if grounds are inserted between every LVDS pair and the cable lengths are short. Remember, the selection of a cable requires the uncontrolled cable skew to be well under 500 ps for the reference design to function reliably at 622 Mb/s.

Simulation Results

Figure 2 shows the complete schematic of the Virtex-E LVDS line driver and receiver. When driving a Virtex-E LVDS line receiver, connect the LVDS_OUT node in Figure 2 to a Virtex-E input (LVDS_IN) and the LVDS_OUT node to the complementary Virtex-E input (LVDS_IN) of the true-differential input.

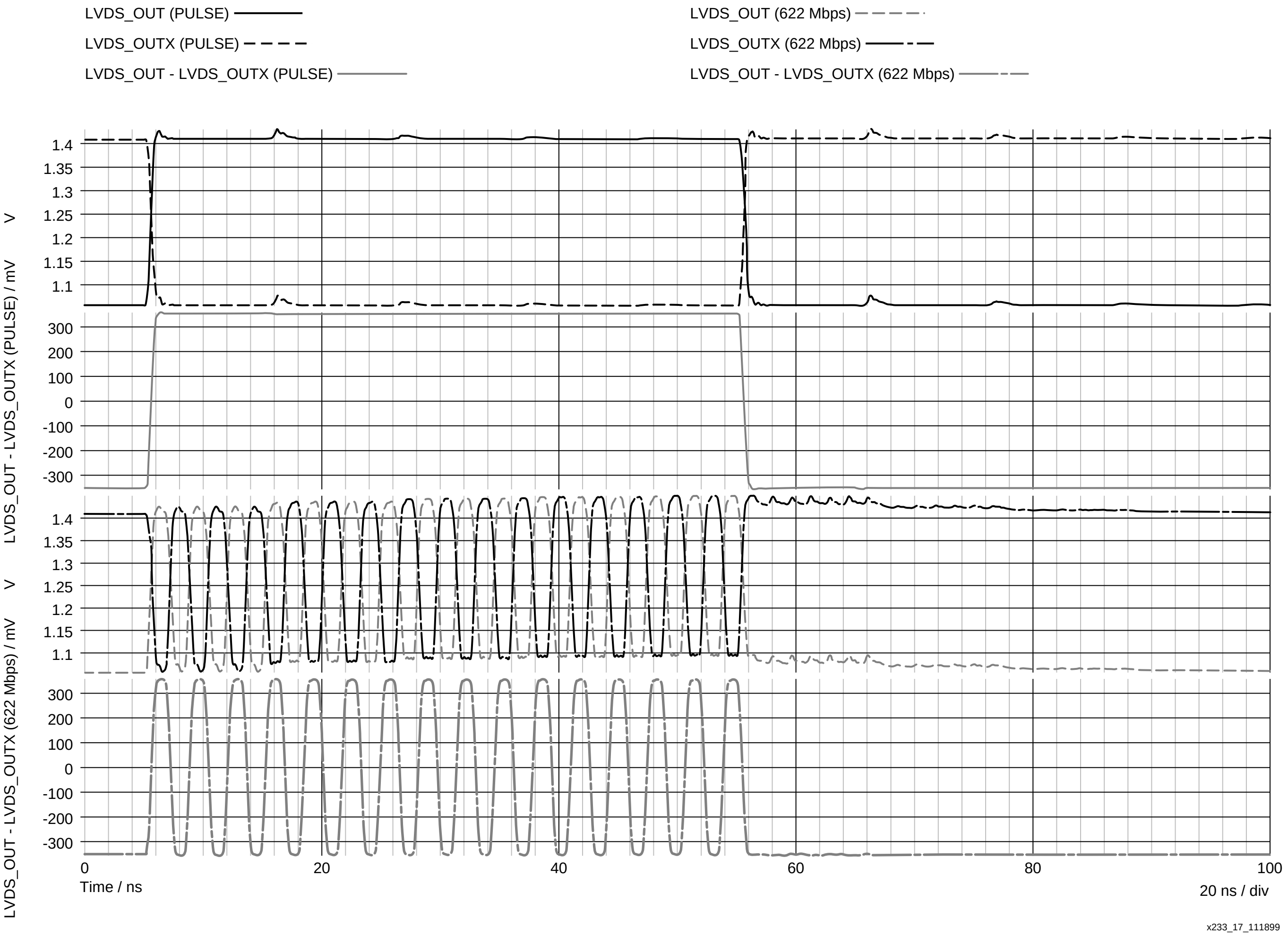


Figure 17: Pulse and 622Mb/s Burst Data Response of the Virtex-E LVDS Driver in Figure 2 with 5 ns Transmission Lines.

The board-level design of Figure 2 was simulated assuming the PCB guidelines of the previous section were used. The SPICE simulation included parasitic package effects for the BG432 package, running bursts of alternating data to measure response time and multi-symbol interference.

Figure 17 shows the pulse and 622 Mb/s burst data response of the Virtex-E LVDS line driver circuit in Figure 2 driving a Virtex-E LVDS receiver in the 432-pin BGA pack-age with long (5 ns) transmission lines. Voltages are measured at the on-die differential input. Notice the differential reflections (LVDS_OUT - $\overline{\text{LVDS\_OUT}}$ on second waveform down) are negligible, confirming that the matched source impedance of the Virtex-E LVDS driver absorbs nearly all differential reflections. The well-matched source impedance of the Virtex-E LVDS driver results in no undershoot or signal swing reduction when driving pulsed data at 622 Mb/s or clocks at 311 MHz, as can be seen on the LVDS_OUT - $\overline{\text{LVDS\_OUT}}$ graph at the bottom of Figure 17.

For the best LVDS signal quality, the Virtex-E LVDS driver will actually improve signal integrity over standard off-the-shelf LVDS drivers due to its matched source and destination terminations.

Conclusion

The Virtex-E series of devices can transmit and receive LVDS at 622 Mb/s or a 311 MHz clock for the – 7 speed grade Virtex-E device. Reliable data transmission is possible over electrical lengths exceeding 5 ns (30 inches), limited only by cable attenuation due to skin effect. Virtex-E devices utilizing LVDS can reliably transfer high-speed data and clocks over long distances between boards, chassis, and peripherals.

Revision History

Date	Version #	Revision
12.21.99	1.0	Initial release.