Efinix T20 EVB

May 6, 2023
Updated: May 13, 2023

Ordering & Setup

Buy it here: https://www.digikey.com/en/products/detail/efinix-inc/T20F256C-DK/10654491

Download Efinity SW here: https://www.efinixinc.com/products-efinity.html

Register your board here: https://www.efinixinc.com/support/profile-devkit.php

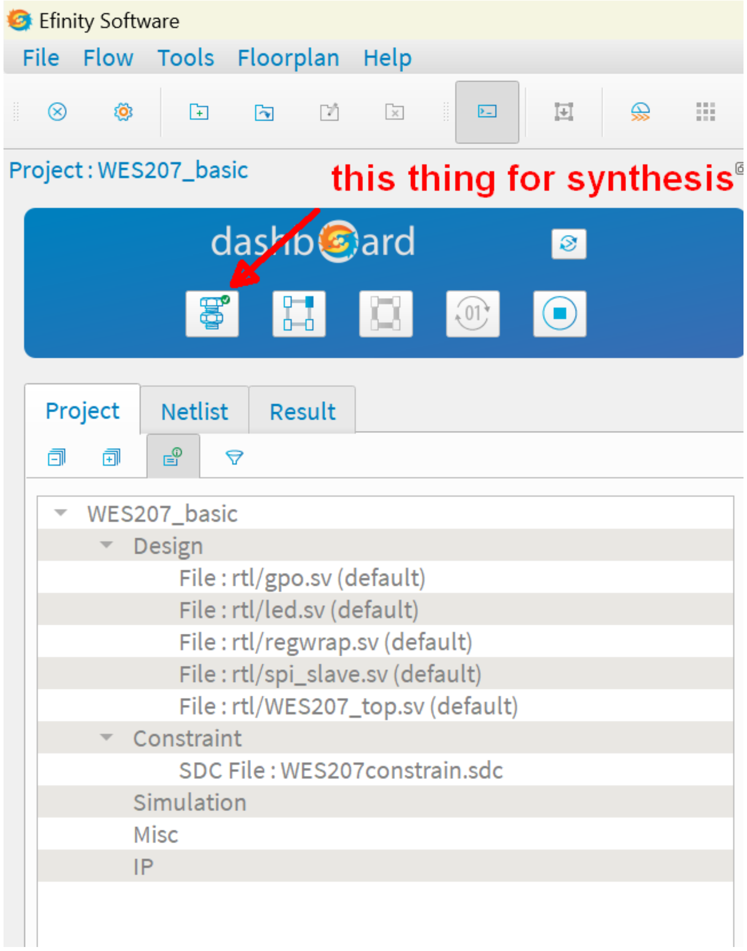
Board

Basic stuff

D3 and D4 are blinky LEDs. If they are blinking, you are good!
Blink rate can be changed by writing SPI register 0x5

Synthesis

The flow is you run synthesis first.

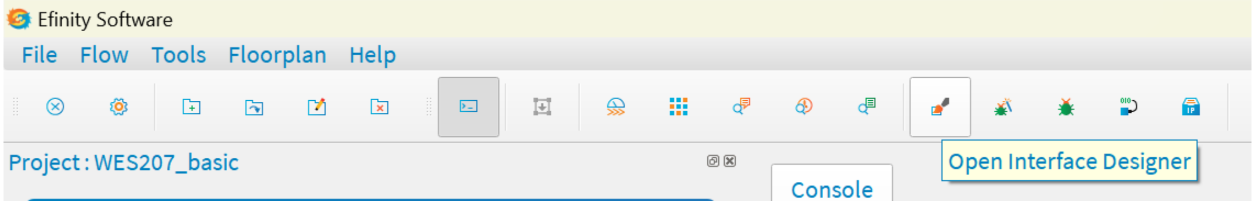


Project : WES207_basic

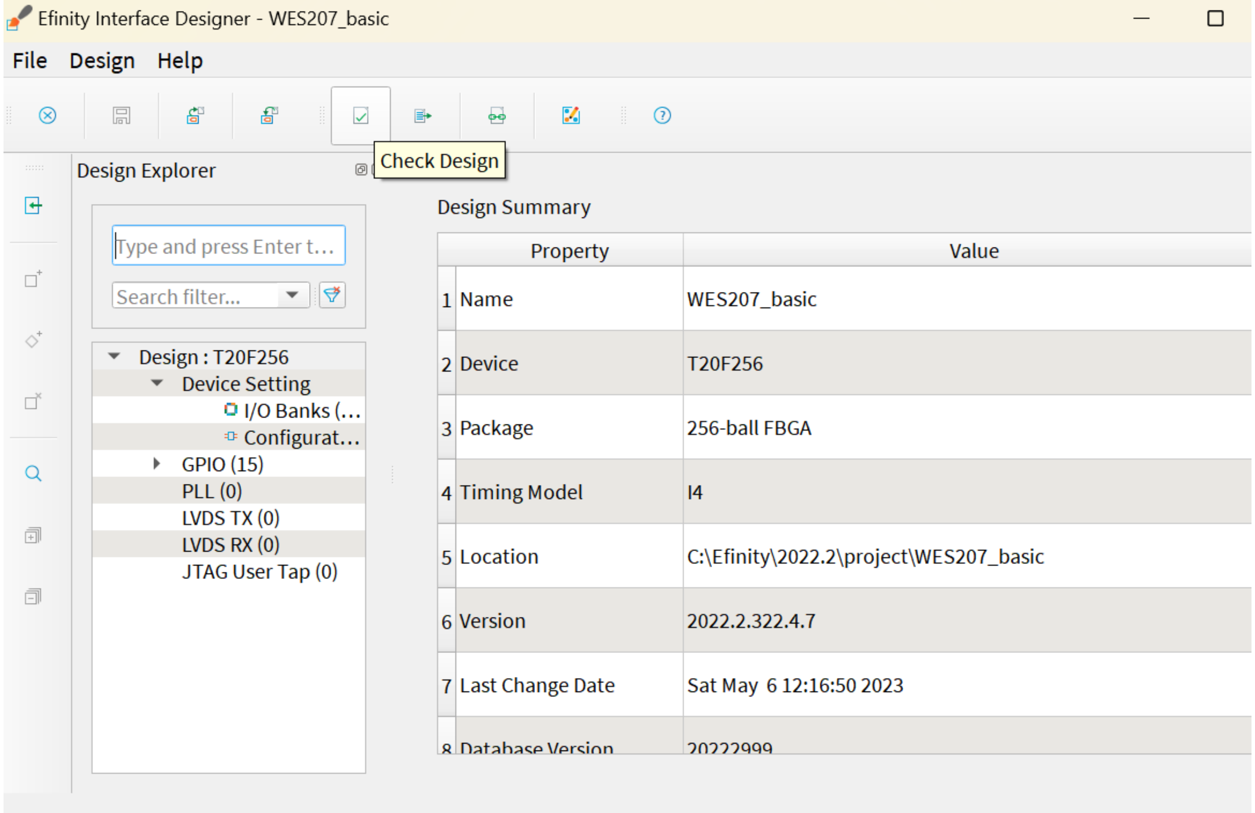
this thing for synthesis

- WES207_basic
  - Design
    - File : rtl/gpo.sv (default)
    - File : rtl/led.sv (default)
    - File : rtl/regwrap.sv (default)
    - File : rtl/spi_slave.sv (default)
    - File : rtl/WES207_top.sv (default)
  - Constraint
    - SDC File : WES207constrain.sdc
  - Simulation
  - Misc
  - IP

if synthesis is clean, you can assign pins. The pins should already be assigned if you pull from the repo but if you need to add pins you would select 'open interface designer'



Project : WES207_basic



Design Summary

	Property	Value
1	Name	WES207_basic
2	Device	T20F256
3	Package	256-ball FBGA
4	Timing Model	I4
5	Location	C:\Efinity\2022.2\project\WES207_basic
6	Version	2022.2.322.4.7
7	Last Change Date	Sat May 6 12:16:50 2023
8	Database Version	20222999

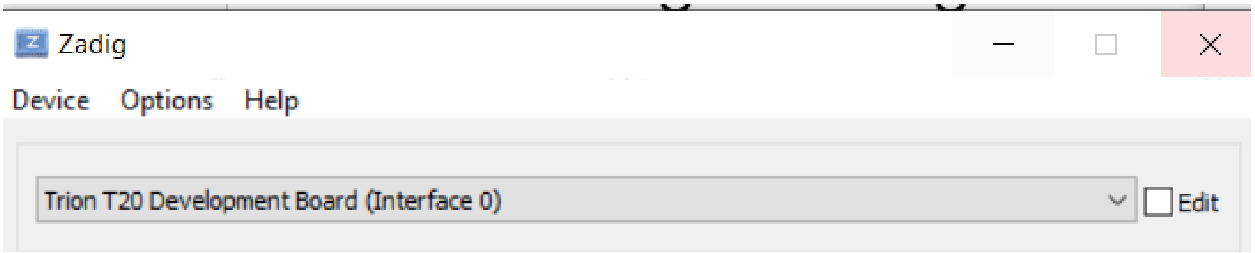
Run ‘check design’ and if it is clean, save it

Finish rest of steps, map, place and route by clicking on the icons under d@ashboard until bitstream is generated

Programming

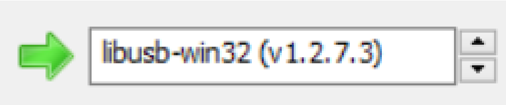
For first time use, the Trion Development Board USB drivers may need to be manually installed. If the board does not show up when you open the programmer, do the following.
Download the zadig exe from here:
https://zadig.akeo.ie/

1. Connect the board to your computer with the appropriate cable and power it up.
2. Run the Zadig software.
3. Choose Options > List All Devices.
4. Select Trion T20 Development Board (Interface 0)



Trion T20 Development Board (Interface 0)

5. Select libusb-win32 as the driver pointed to by the green arrow using the up/down buttons:

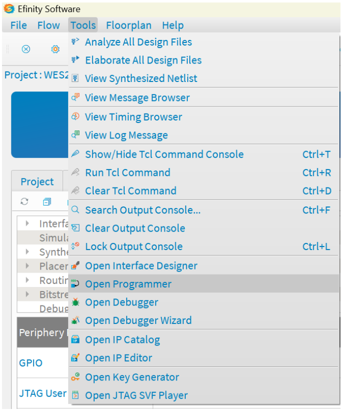


libusb-win32 (v1.2.7.3)

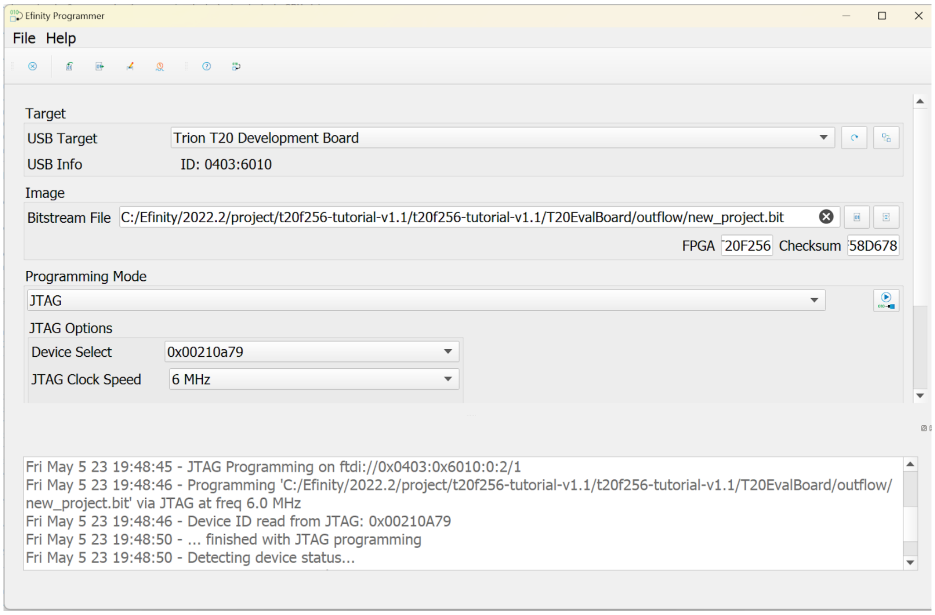
6. Click Replace Driver.
7. Repeat steps 4-6 for “Trion T20 Development Board (Interface 1)”

Important! Plug board in before opening programmer, otherwise it doesn’t see it

Tools>Open Programmer



You should see a window like this:



USB Target: Trion T20 Development Board
USB Info: ID: 0403:6010
Bitstream File: C:/Efinity/2022.2/project/t20f256-tutorial-v1.1/t20f256-tutorial-v1.1/T20EvalBoard/outflow/new_project.bit
FPGA: 20F256 Checksum: 58D678
Programming Mode: JTAG
Device Select: 0x00210a79
JTAG Clock Speed: 6 MHz

```
Fri May 5 23 19:48:45 - JTAG Programming on ftdi://0x0403:0x6010:0:2/1
Fri May 5 23 19:48:46 - Programming 'C:/Efinity/2022.2/project/t20f256-tutorial-v1.1/t20f256-tutorial-v1.1/T20EvalBoard/outflow/
new_project.bit' via JTAG at freq 6.0 MHz
Fri May 5 23 19:48:46 - Device ID read from JTAG: 0x00210A79
Fri May 5 23 19:48:50 - ... finished with JTAG programming
Fri May 5 23 19:48:50 - Detecting device status...
```

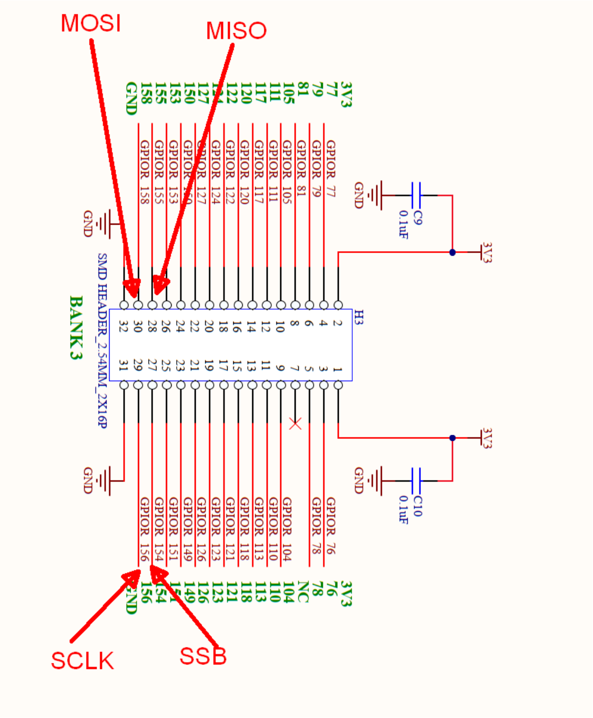
Programming mode can be **JTAG** or **SPI Active**. SPI Active writes the bitstream to the FLASH on board the FPGA so you don’t have to load it from the programmer every time you power on the board.

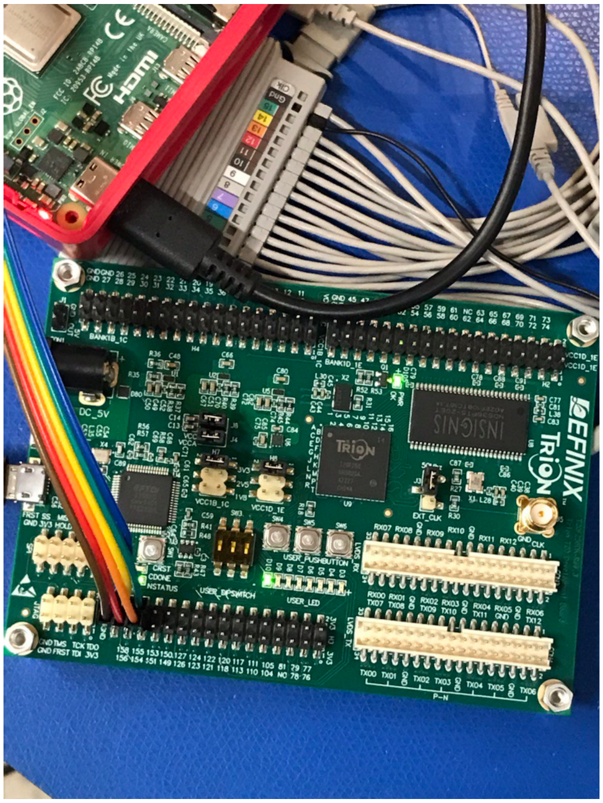
If programming JTAG under **Image** select .bit file. If programming FLASH using SPI Active, use .hex file

NOTE: DO NOT UNPLUG THE BOARD DURING ERASE OR PROGRAMMING. It may not break anything or it may brick the board. Just wait until the flash program completes.

SPI Slave

Spi Slave interface is on Bank 3





Register map

(subject to change)
The register map is accessible using SPI Slave interface

SPI ADDR	Name	Bits	
0x0	GPO_CTRL	7 6:0	1=if bit is set, overwrite GPO setting; 0 = no overwrite GPO Pin output value
0x1	STATUS		
0x2	tx_packet_len		
0x3	tx_packet		
0x4	rx_packet_len		
0x5	rx_packet		
0x6	LED_CTRL	7:0	control 8 MSBs of ctr_cfg_reg in led.sv 8'b0000_0000 - blink really fast 8'b1111_1111 - blink slower
0x7	TX_TEST_CTRL	7:2 ~~1~~ 0	Reserved ~~1=SWAP I/Q~~ ~~0=normal operation~~ 1=if bit is set, sets tx_en=1 and transmit preamble continuously on LVDS TX pins

LVDS

LVDS TX is on CON3 and LVDS RX is on CON4
LVDS can be configured in 2:1 mode, which is basically 2 bits parallel to 1 bit serial (single ended to dual data rate, input and output clocks are the same) up to 8:1 mode, where the parallel input data is 1/8 of the output data rate.
The input clock to the lvds block is **tx_slowclk**, and the output clock is **tx_fastclk.** Tx_slowclk is enumerated in the top_level. Tx_fastclk is only enumerated in the interface designer.
So in other words we could clock the input to the LVDS block (tx_slowclk) at

- 64MHz - 2 bits xferred in on every tx_slowclk rising edge. LVDS tx@ rising and falling edge
- 32MHz - 4 bits xferred in on every tx_slowclk x 4 serialization [2 LVDS tx_fastclk]
- 16MHz - 8 bits xferred in on every tx_slowclk x 8 serialization [4 LVDS tx_fastclk]

We will use

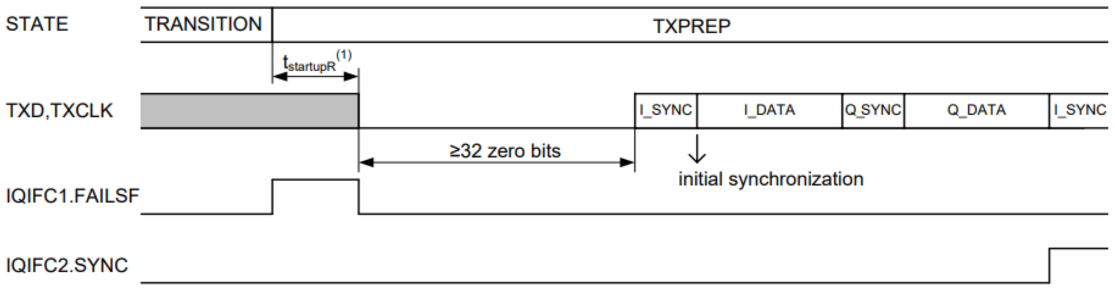
- GPIOB_TX00 p/n pair for LVDS TX
- GPIOB_TX12 p/n pair for LVDS TX CLK (fastclk)

ATMEL TX IQ Interface

The Atmel is expecting 32-bit data, so we will need to chunk the output of the Efinix FPGA to match.

The atmel expects the TX (data to ATMEL) to be edge aligned

Note the startup requirement of the ATMEL device:



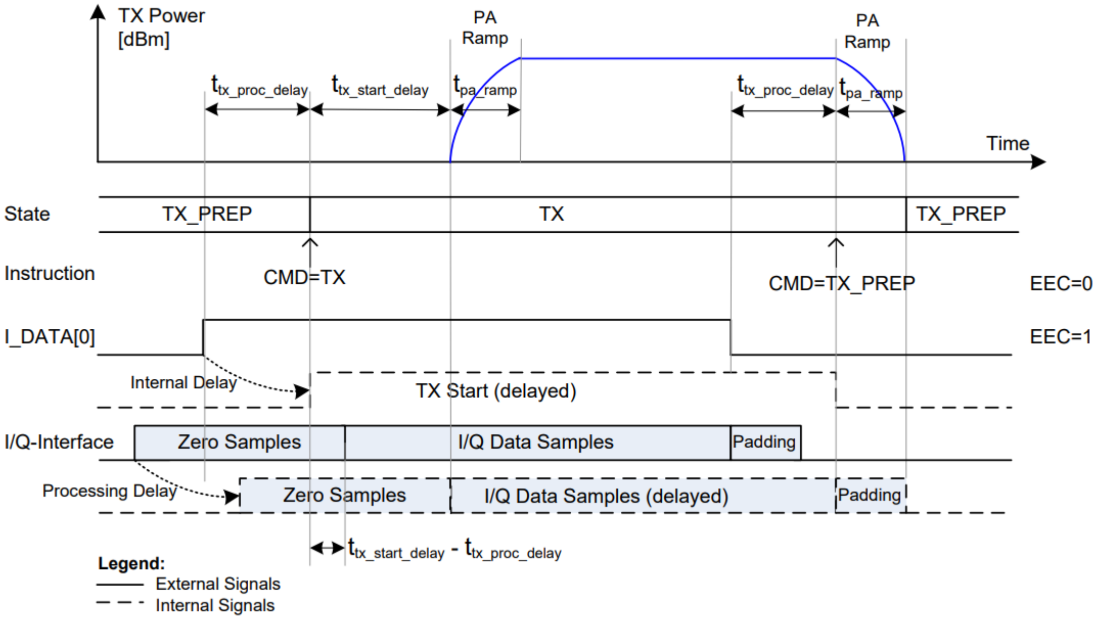
Note: I/Q data interface receiver start-up time (see section "I/Q Data Interface Driver DC and Startup Specification" on page 205)

We need to send >= 32 zero bits before starting transition

If we have the FPGA controltransmit (i.e. RF_IQIFC0.EEC=1), we will need to set the LSB of the I Data (bit 0) to 1'b1. It is not clear to me what the procedure is:

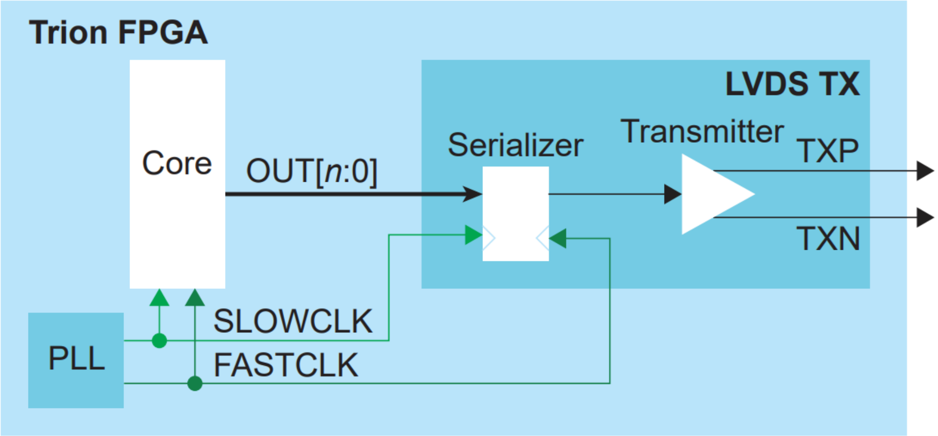
32 zeros, then set I data bit 0=1'b1, then, send I sync/I data/Q sync/q data ??

Figure 6-4. Transmitter Control Timing Diagram for I/Q Radio Mode

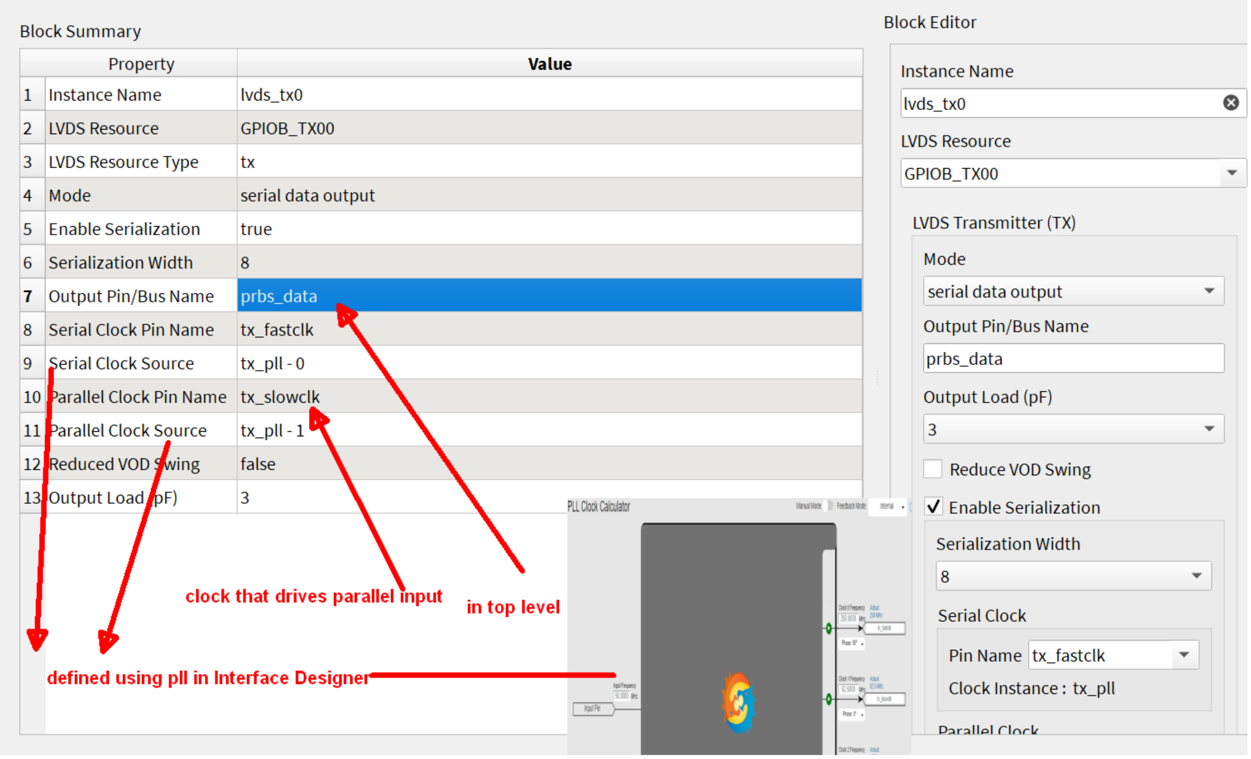


TRION FPGA LVDS Block

Figure 20: LVDS TX Interface Block Diagram



The toplevel will have an output which is OUT[n:0]. The actual LVDS TX pins are not in the toplevel, they are defined using the interface designer.



Block Summary

	Property	Value
1	Instance Name	lvds_tx0
2	LVDS Resource	GPIOB_TX00
3	LVDS Resource Type	tx
4	Mode	serial data output
5	Enable Serialization	true
6	Serialization Width	8
7	Output Pin/Bus Name	prbs_data
8	Serial Clock Pin Name	tx_fastclk
9	Serial Clock Source	tx_pll - 0
10	Parallel Clock Pin Name	tx_slowclk
11	Parallel Clock Source	tx_pll - 1
12	Reduced VOD Swing	false
13	Output Load (pF)	3

Block Editor

Instance Name: lvds_tx0

LVDS Resource: GPIOB_TX00

LVDS Transmitter (TX)

Mode: serial data output

Output Pin/Bus Name: prbs_data

Output Load (pF): 3

☐ Reduce VOD Swing

☑ Enable Serialization

Serialization Width: 8

Serial Clock

Pin Name: tx_fastclk

Clock Instance : tx_pll

Parallel Clock

clock that drives parallel input

in top level

defined using pll in Interface Designer