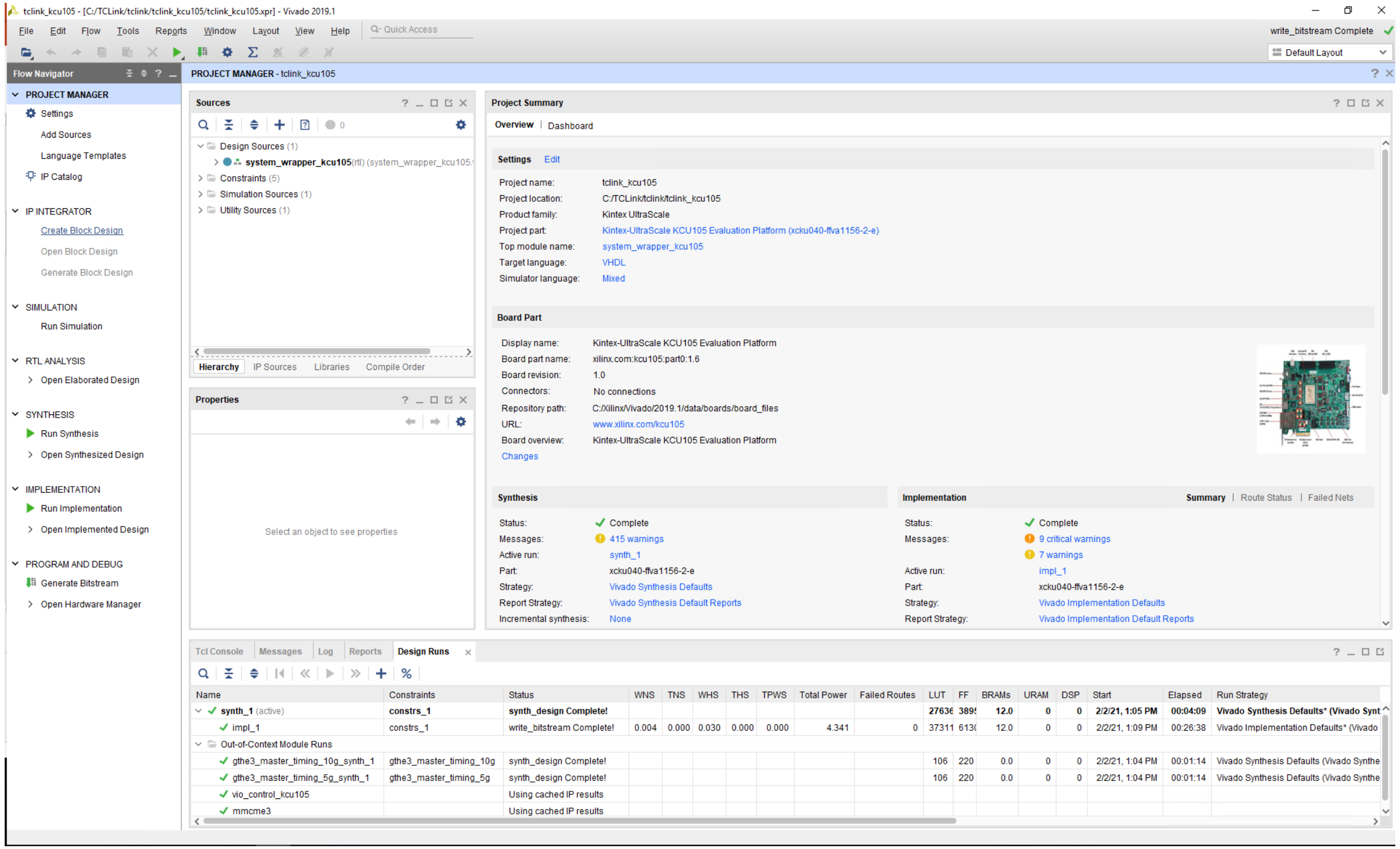
TCLink

KCU105 example design
quick start guide

Eduardo Mendes

Basic configuration

1. clone GIT repository
2. run **tclink_kcu105.tcl** script in **Vivado 2019.1** in batch mode (an example in **tclink_kcu105.bat** for Windows)
3. once the project is created and the physical implementation is finished, you can open the Vivado project



Settings Edit

Project name: tclink_kcu105
Project location: C:/TCLink/tclink/tclink_kcu105
Product family: Kintex UltraScale
Project part: Kintex-UltraScale KCU105 Evaluation Platform (xcku040-ffva1156-2-e)
Top module name: system_wrapper_kcu105
Target language: VHDL
Simulator language: Mixed

Board Part

Display name: Kintex-UltraScale KCU105 Evaluation Platform
Board part name: xilinx.com:kcu105:part0:1.6
Board revision: 1.0
Connectors: No connections
Repository path: C:/Xilinx/Vivado/2019.1/data/boards/board_files
URL: www.xilinx.com/kcu105
Board overview: Kintex-UltraScale KCU105 Evaluation Platform
Changes

Synthesis

Status: ✓ Complete
Messages: 415 warnings
Active run: synth_1
Part: xcku040-ffva1156-2-e
Strategy: Vivado Synthesis Defaults
Report Strategy: Vivado Synthesis Default Reports
Incremental synthesis: None

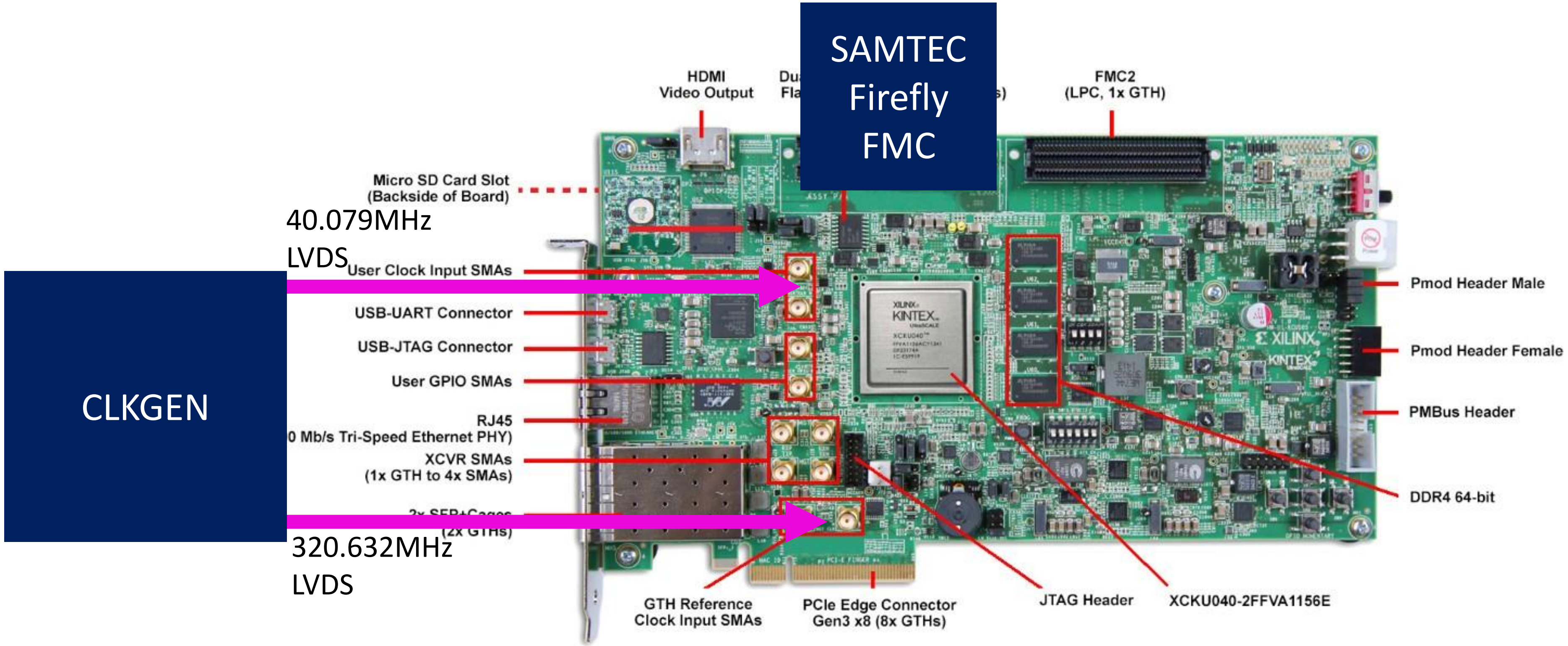
Implementation Summary | Route Status | Failed Nets

Status: ✓ Complete
Messages: 9 critical warnings
7 warnings
Active run: impl_1
Part: xcku040-ffva1156-2-e
Strategy: Vivado Implementation Defaults
Report Strategy: Vivado Implementation Default Reports

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Run Strategy
✓ **synth_1** (active)	**constrs_1**	**synth_design Complete!**								**27636**	**389!**	**12.0**	**0**	**0**	**2/2/21, 1:05 PM**	**00:04:09**	**Vivado Synthesis Defaults* (Vivado Synt**
✓ impl_1	constrs_1	write_bitstream Complete!	0.004	0.000	0.030	0.000	0.000	4.341	0	37311	613(	12.0	0	0	2/2/21, 1:09 PM	00:26:38	Vivado Implementation Defaults* (Vivado
Out-of-Context Module Runs																	
✓ gthe3_master_timing_10g_synth_1	gthe3_master_timing_10g	synth_design Complete!								106	220	0.0	0	0	2/2/21, 1:04 PM	00:01:14	Vivado Synthesis Defaults (Vivado Synthe
✓ gthe3_master_timing_5g_synth_1	gthe3_master_timing_5g	synth_design Complete!								106	220	0.0	0	0	2/2/21, 1:04 PM	00:01:14	Vivado Synthesis Defaults (Vivado Synthe
✓ vio_control_kcu105		Using cached IP results															
✓ mmcme3		Using cached IP results															

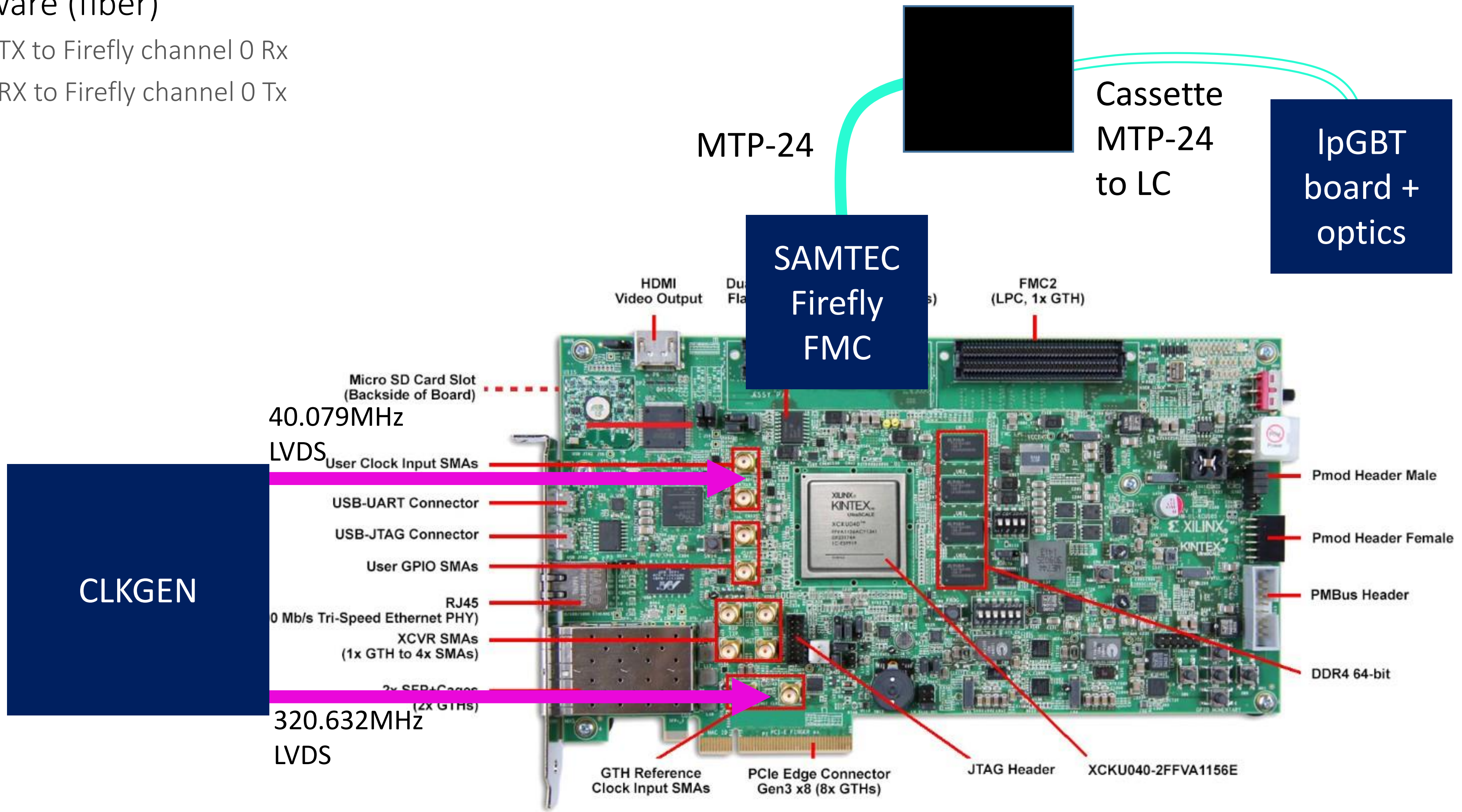
Basic configuration

4. Hardware (clocks, modules)*
   - CLKGEN: Free-running reference clock providing 320MHz and 40MHz for master
   - SAMTEC FIREFLY FMC: SAMTEC FMC containing 1xTX12 and 1xRX12 firefly modules
   - OBS: lpGBT characterization board or VLDB+ with optical modules is also necessary



Basic configuration

5. Hardware (fiber)
   - SFP TX to Firefly channel 0 Rx
   - SFP RX to Firefly channel 0 Tx

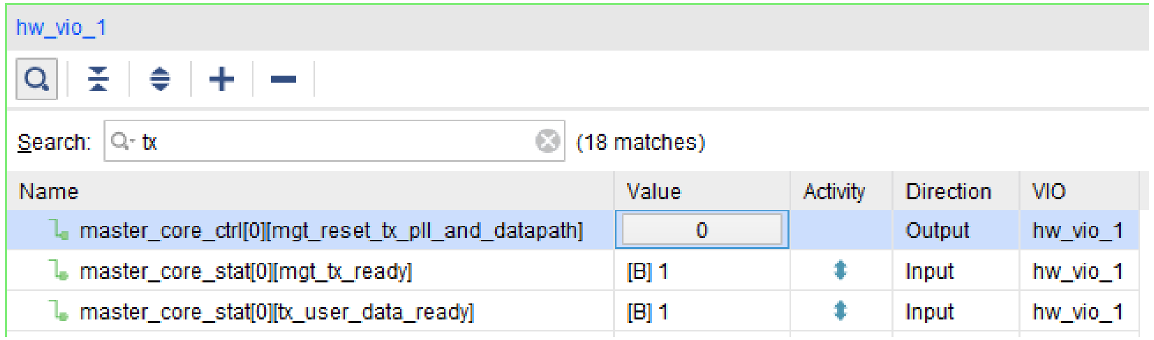


Basic configuration

6. Program FPGA ☺

7. The design can be controlled using the VIO graphical interface in Vivado

8. Reset Master Tx (Reset Tx PLL and datapath)
   - Check tx_ready



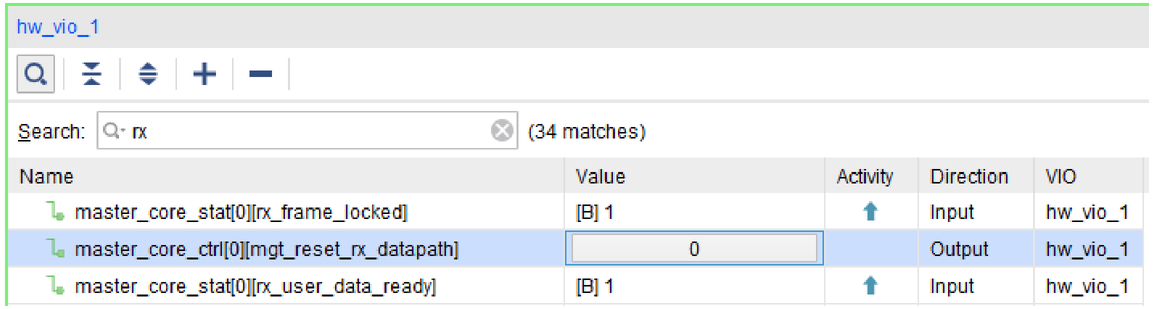
hw_vio_1

Search: tx (18 matches)

Name	Value	Activity	Direction	VIO
master_core_ctrl[0][mgt_reset_tx_pll_and_datapath]	0		Output	hw_vio_1
master_core_stat[0][mgt_tx_ready]	[B] 1	⬍	Input	hw_vio_1
master_core_stat[0][tx_user_data_ready]	[B] 1	⬍	Input	hw_vio_1

9. **Reset and configure lpGBT**
   - Check READY state
   - Check that downlink is error free
   - In case of need, contact VLDB+ support*

10. Reset Master Rx (Reset Rx datapath)
    - Check rx frame is locked
    - OBS: since master has Tx and Rx PLL shared, do not use Reset Rx PLL and datapath



hw_vio_1

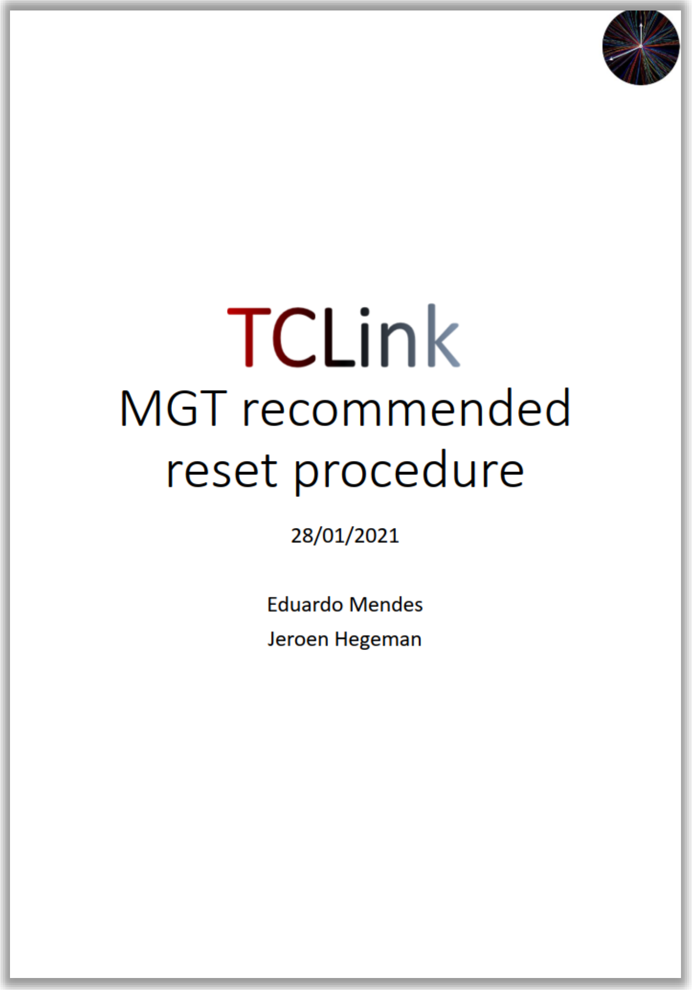
Search: rx (34 matches)

Name	Value	Activity	Direction	VIO
master_core_stat[0][rx_frame_locked]	[B] 1	↑	Input	hw_vio_1
master_core_ctrl[0][mgt_reset_rx_datapath]	0		Output	hw_vio_1
master_core_stat[0][rx_user_data_ready]	[B] 1	↑	Input	hw_vio_1

Basic configuration

-> An example of FSM implementation for the reset scheme is integrated in the core

- Read the design choice documents for the MGT recommended reset procedure to know more



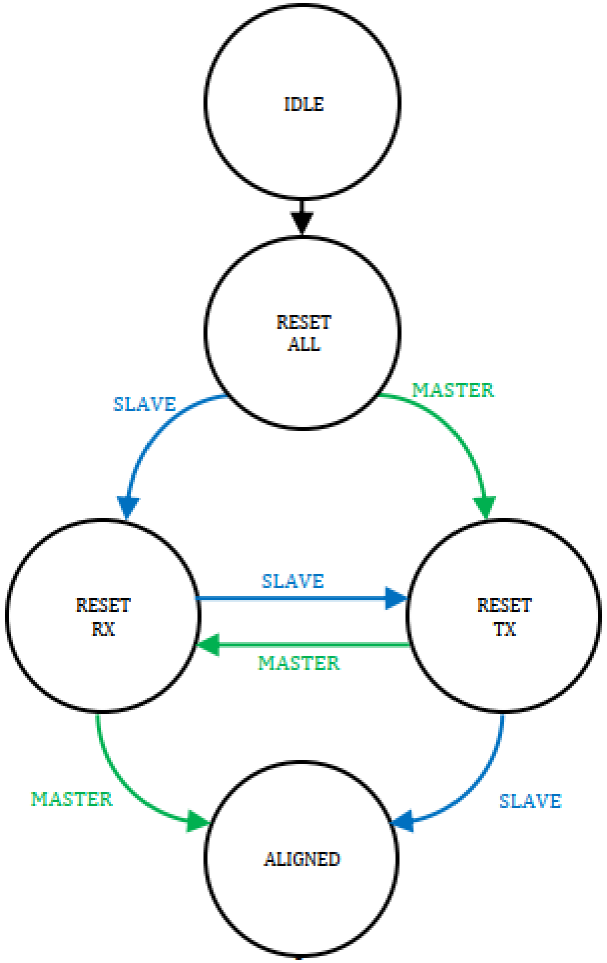
TCLink

MGT recommended reset procedure

28/01/2021

Eduardo Mendes

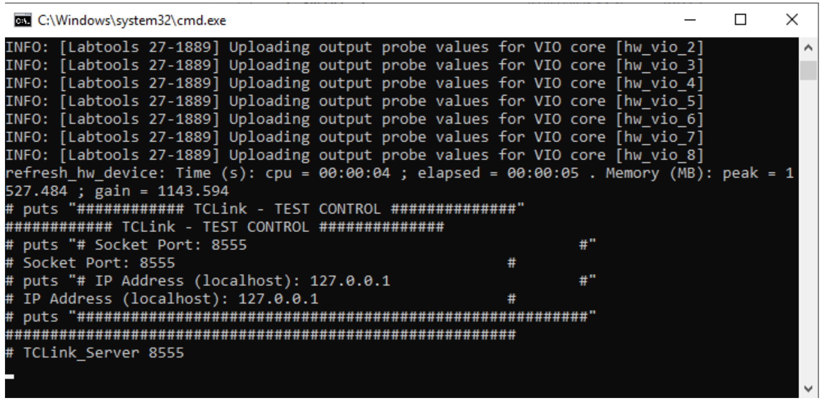
Jeroen Hegeman



Basic configuration

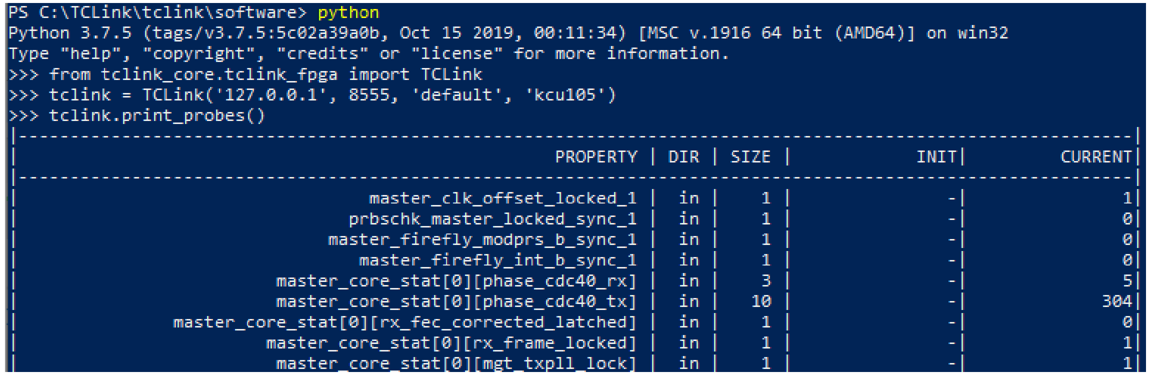
11. The design can be also controlled in **Python3** (for repetitive tests)
    a. Execute **software/jtag_server/jtag_server_lpgbtfpga_kcu105.tcl** in Vivado batch mode
    (an example for Windows in software/jtag_server/execute_jtag_server_kcu105.bat)

Jtag server example



```
INFO: [Labtools 27-1889] Uploading output probe values for VIO core [hw_vio_2]
INFO: [Labtools 27-1889] Uploading output probe values for VIO core [hw_vio_3]
INFO: [Labtools 27-1889] Uploading output probe values for VIO core [hw_vio_4]
INFO: [Labtools 27-1889] Uploading output probe values for VIO core [hw_vio_5]
INFO: [Labtools 27-1889] Uploading output probe values for VIO core [hw_vio_6]
INFO: [Labtools 27-1889] Uploading output probe values for VIO core [hw_vio_7]
INFO: [Labtools 27-1889] Uploading output probe values for VIO core [hw_vio_8]
refresh_hw_device: Time (s): cpu = 00:00:04 ; elapsed = 00:00:05 . Memory (MB): peak = 1
527.484 ; gain = 1143.594
# puts "############ TCLink - TEST CONTROL ##############"
############ TCLink - TEST CONTROL ##############
# puts "# Socket Port: 8555                                 #"
# Socket Port: 8555                                 #
# puts "# IP Address (localhost): 127.0.0.1                 #"
# IP Address (localhost): 127.0.0.1                 #
# puts "########################################################"
########################################################
# TCLink_Server 8555
```

Python example

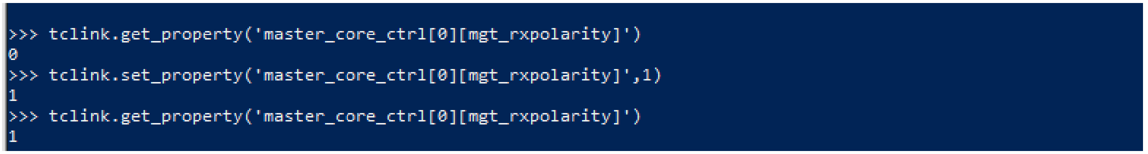


```
PS C:\TCLink\tclink\software> python
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from tclink_core.tclink_fpga import TCLink
>>> tclink = TCLink('127.0.0.1', 8555, 'default', 'kcu105')
>>> tclink.print_probes()
|------------------------------------------------------------------------------------------------------------|
|                                               PROPERTY | DIR | SIZE |                INIT|           CURRENT|
|------------------------------------------------------------------------------------------------------------|
|                              master_clk_offset_locked_1 |  in |    1 |                   -|                 1|
|                            prbschk_master_locked_sync_1 |  in |    1 |                   -|                 0|
|                          master_firefly_modprs_b_sync_1 |  in |    1 |                   -|                 0|
|                             master_firefly_int_b_sync_1 |  in |    1 |                   -|                 0|
|                      master_core_stat[0][phase_cdc40_rx] |  in |    3 |                   -|                 5|
|                      master_core_stat[0][phase_cdc40_tx] |  in |   10 |                   -|               304|
|               master_core_stat[0][rx_fec_corrected_latched] |  in |    1 |                   -|                 0|
|                       master_core_stat[0][rx_frame_locked] |  in |    1 |                   -|                 1|
|                        master_core_stat[0][mgt_txpll_lock] |  in |    1 |                   -|                 1|
```

b. Open socket in python to the local JTAG server

…

c. Check VIO probes available



```
>>> tclink.get_property('master_core_ctrl[0][mgt_rxpolarity]')
0
>>> tclink.set_property('master_core_ctrl[0][mgt_rxpolarity]',1)
1
>>> tclink.get_property('master_core_ctrl[0][mgt_rxpolarity]')
1
```

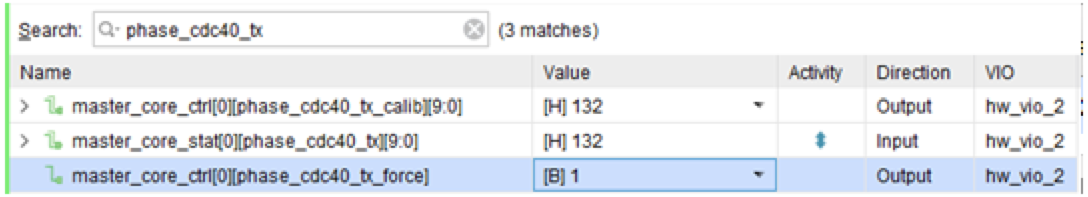
d. Set/get a probe value

e. Additional features:
   a. Preset design (function **preset()**)
   b. Reprogram FPGA (function **fpga_program()**)
   c. Sysmon monitoring (function **save_sysmon_state(name_file)**)

Fixed latency CDC 40-320

- If required, the CDC for Tx (40MHz to 320MHz) and Rx (320MHz to 40MHz) allow a fixed-latency operation after resets
  - In order to achieve fixed-latency, the phase has to be forced after the first reset
  - Example for master0 Tx

```
tx40_phase = dut.get_property(‘master_core_stat[0][phase_cdc40_tx]’)
dut.set_property(‘master_core_ctrl[0][phase_cdc40_tx_calib]’, tx40_phase)
dut.set_property(‘master_core_ctrl[0][phase_cdc40_tx_force]’, 1)
```

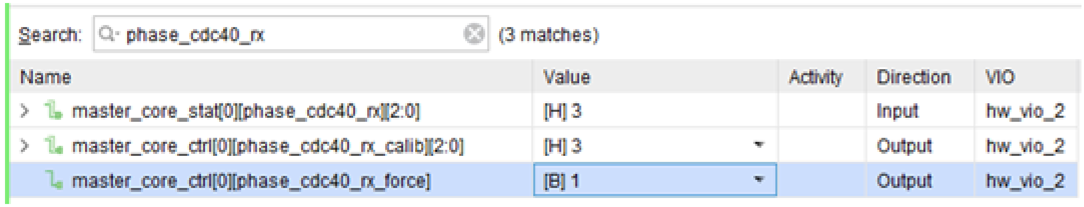


Search: phase_cdc40_tx (3 matches)

Name	Value	Activity	Direction	VIO
master_core_ctrl[0][phase_cdc40_tx_calib][9:0]	[H] 132		Output	hw_vio_2
master_core_stat[0][phase_cdc40_tx][9:0]	[H] 132	⇕	Input	hw_vio_2
master_core_ctrl[0][phase_cdc40_tx_force]	[B] 1		Output	hw_vio_2

  - Example for master0 Rx

```
rx40_phase = dut.get_property(‘master_core_stat[0][phase_cdc40_rx]’)
dut.set_property(‘master_core_ctrl[0][phase_cdc40_rx_calib]’, rx40_phase)
dut.set_property(‘master_core_ctrl[0][phase_cdc40_rx_force]’, 1)
```



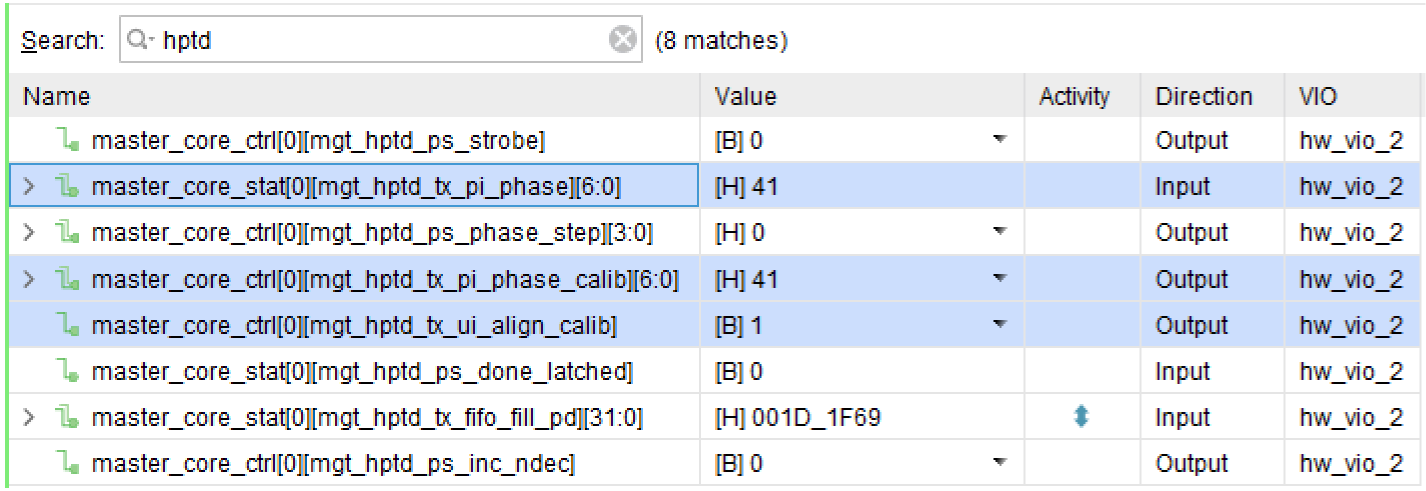
Search: phase_cdc40_rx (3 matches)

Name	Value	Activity	Direction	VIO
master_core_stat[0][phase_cdc40_rx][2:0]	[H] 3		Input	hw_vio_2
master_core_ctrl[0][phase_cdc40_rx_calib][2:0]	[H] 3		Output	hw_vio_2
master_core_ctrl[0][phase_cdc40_rx_force]	[B] 1		Output	hw_vio_2

MGT Tx fixed-phase

- HPTD IP core (more information on HPTD IP documentation)
  - In order to freeze a given Tx PI phase value (after first reset) – example for master0

```
tx_phase = dut.get_property('master_core_stat[0][mgt_hptd_tx_pi_phase]')
dut.set_property('master_core_ctrl[0][mgt_hptd_tx_pi_phase_calib]', tx_phase)
dut.set_property('master_core_ctrl[0][ mgt_hptd_tx_ui_align_calib]', 1)
```



Search: hptd (8 matches)

Name	Value	Activity	Direction	VIO
master_core_ctrl[0][mgt_hptd_ps_strobe]	[B] 0		Output	hw_vio_2
master_core_stat[0][mgt_hptd_tx_pi_phase][6:0]	[H] 41		Input	hw_vio_2
master_core_ctrl[0][mgt_hptd_ps_phase_step][3:0]	[H] 0		Output	hw_vio_2
master_core_ctrl[0][mgt_hptd_tx_pi_phase_calib][6:0]	[H] 41		Output	hw_vio_2
master_core_ctrl[0][mgt_hptd_tx_ui_align_calib]	[B] 1		Output	hw_vio_2
master_core_stat[0][mgt_hptd_ps_done_latched]	[B] 0		Input	hw_vio_2
master_core_stat[0][mgt_hptd_tx_fifo_fill_pd][31:0]	[H] 001D_1F69	⬍	Input	hw_vio_2
master_core_ctrl[0][mgt_hptd_ps_inc_ndec]	[B] 0		Output	hw_vio_2

  - The user can also shift the phase with o(ps) resolution using the mgt_hptd_ps_inc_ndec (increment or decrement), mgt_hptd_ps_phase_step and mgt_hptd_ps_strobe signals

TCLink basic configuration

- Before closing the loop, the offset phase has to be measured (procedure to be done once after first reset):

```
dut.set_property(‘tclink_offset_error’, 0)
Wait for at least one second…
offset = dut.get_property(‘tclink_error_controller’)
dut.set_property(‘tclink_offset_error’, offset)
```

- Close-loop:

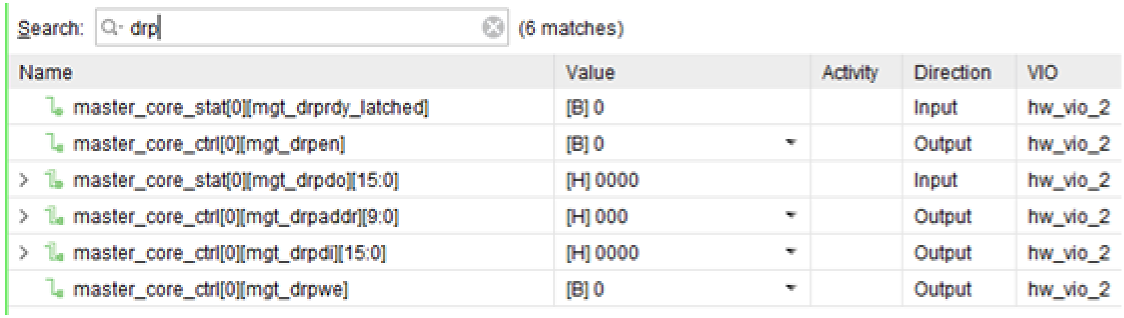
```
dut.set_property(‘tclink_close_loop’, 1)
```

- An example of how to configure TCLink in the example design and run a transfer function in the FPGA (using the TCLink tester) is given in the script software/fpga_transfer_function_kcu105.py

- Read official tclink reference note on how to convert phase measurement values to ps

Additional MGT features

- Dynamic reconfiguration port

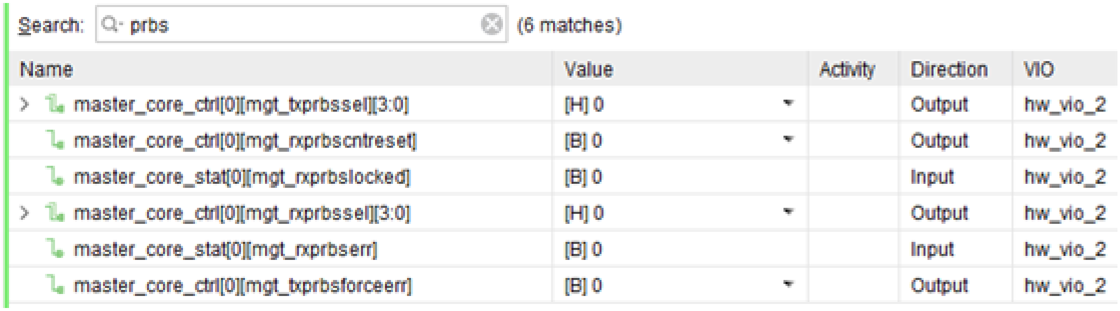


Search: drp (6 matches)

Name	Value	Activity	Direction	VIO
master_core_stat[0][mgt_drprdy_latched]	[B] 0		Input	hw_vio_2
master_core_ctrl[0][mgt_drpen]	[B] 0		Output	hw_vio_2
master_core_stat[0][mgt_drpdo][15:0]	[H] 0000		Input	hw_vio_2
master_core_ctrl[0][mgt_drpaddr][9:0]	[H] 000		Output	hw_vio_2
master_core_ctrl[0][mgt_drpdi][15:0]	[H] 0000		Output	hw_vio_2
master_core_ctrl[0][mgt_drpwe]	[B] 0		Output	hw_vio_2

- Internal MGT PRBS
  - lpGBT-FPGA Rx is kept reset when internal PRBS is selected
  - OBS: txprbs shall not work with lpGBT because design was created for 10.24Gb/s transceiver

0000	= Normal operation
0001	= PRBS-7
0010	= PRBS-9
0011	= PRBS-15
0100	= PRBS-23
0101	= PRBS-31
1001	= Square wave with 2 UI (alternating 0s/1s)
1010	= Square wave with 32 UI

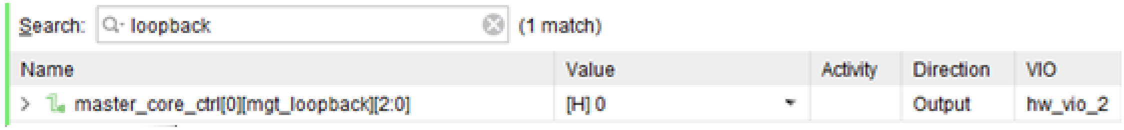


Search: prbs (6 matches)

Name	Value	Activity	Direction	VIO
master_core_ctrl[0][mgt_txprbssel][3:0]	[H] 0		Output	hw_vio_2
master_core_ctrl[0][mgt_rxprbscntreset]	[B] 0		Output	hw_vio_2
master_core_stat[0][mgt_rxprbslocked]	[B] 0		Input	hw_vio_2
master_core_ctrl[0][mgt_rxprbssel][3:0]	[H] 0		Output	hw_vio_2
master_core_stat[0][mgt_rxprbserr]	[B] 0		Input	hw_vio_2
master_core_ctrl[0][mgt_txprbsforceerr]	[B] 0		Output	hw_vio_2

- Transceiver Loopback

000	= Normal operation
001	= Near-End PCS Loopback
010	= Near-End PMA Loopback
100	= Far-End PMA Loopback (do not use - problem when using Tx PI)
110	= Far-End PCS Loopback



Search: loopback (1 match)

Name	Value	Activity	Direction	VIO
master_core_ctrl[0][mgt_loopback][2:0]	[H] 0		Output	hw_vio_2