Optical Loop Back Test for ODMB7 Preproduction

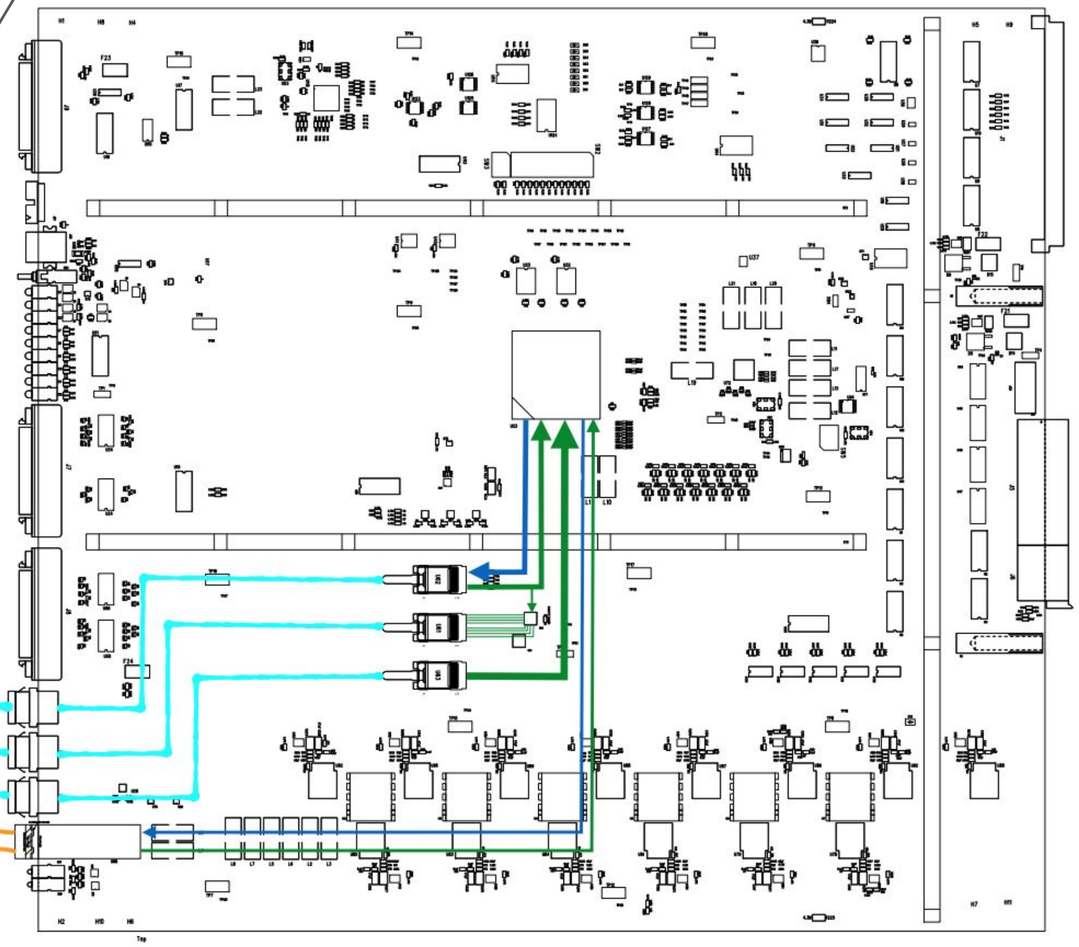
Sicheng Wang

Hardware Connections

Basic idea

Simple counting gives 17 tx and 17 rx, but not all of them go into the FPGA

- All TX ports sending PRBS signals
- All RX ports verify for errors
  - Does not have to be 1–1 with TX ports
- Data flow:
  - FPGA ⟶ B04 ⟶ B04 ⟶ FPGA (4 lines)
    - 1 line
  - Multiplier (12×) ⟶ T12 ⟶ R12 ⟶ FPGA
  - FPGA ⟶ Finisar ⟶ Finisar ⟶ FPAG (1 line)
- All related pins are added to constraint file



B04: 4 in, 4 out

T12: 12 out

R12: 12 in

Finisar: 1 in, 1 out

This test relies on PRBS generator and checker inside the FPGA (GTH transceiver)

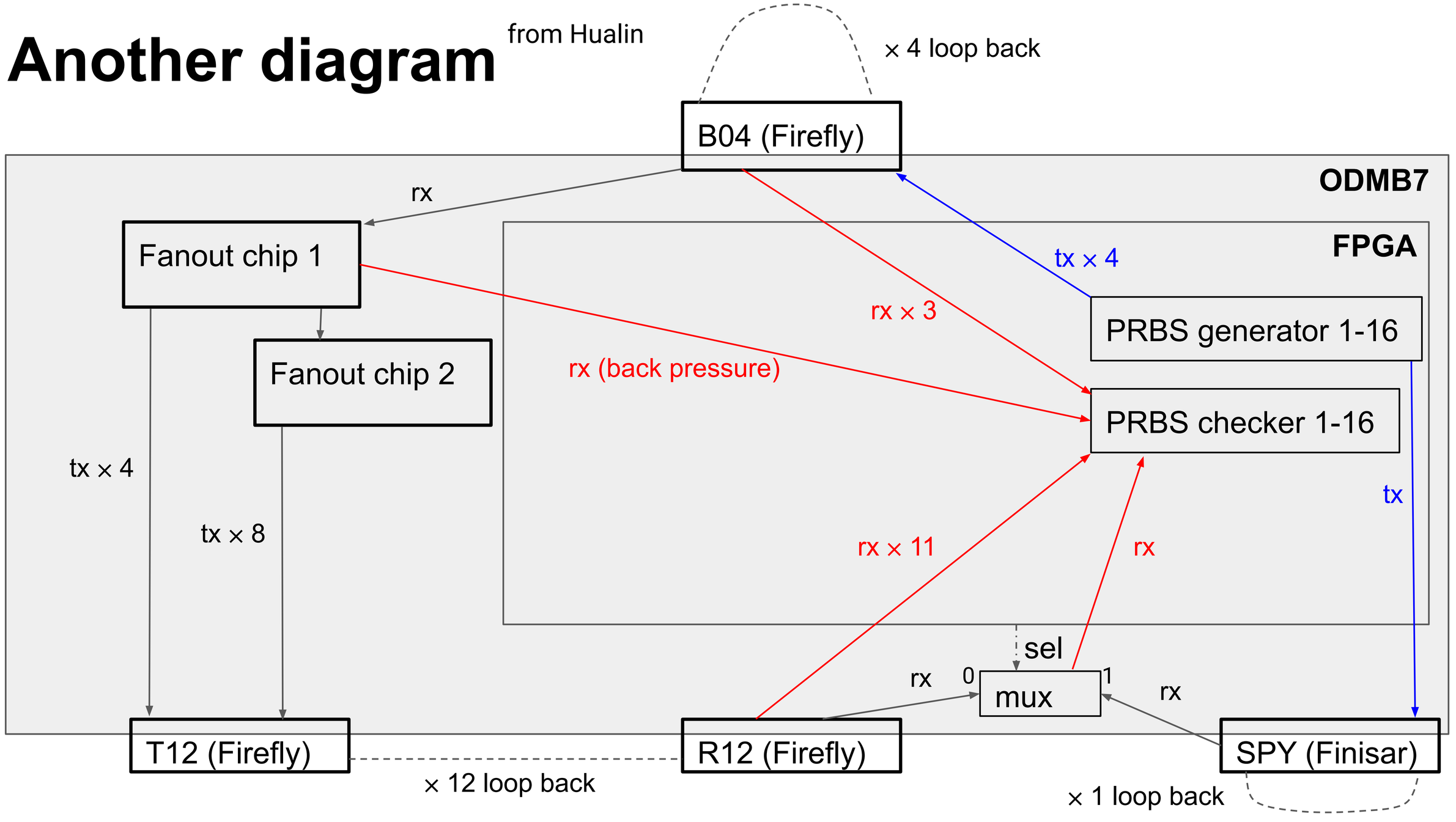
16 PRBS generators/checkers in this version

Firmware compatible with ODMB7 is here:
https://github.com/wsicheng/ODMBDevelopment/tree/master/SingleTestFWs/optical_ibert_gth

Conceptual Diagram

Another diagram

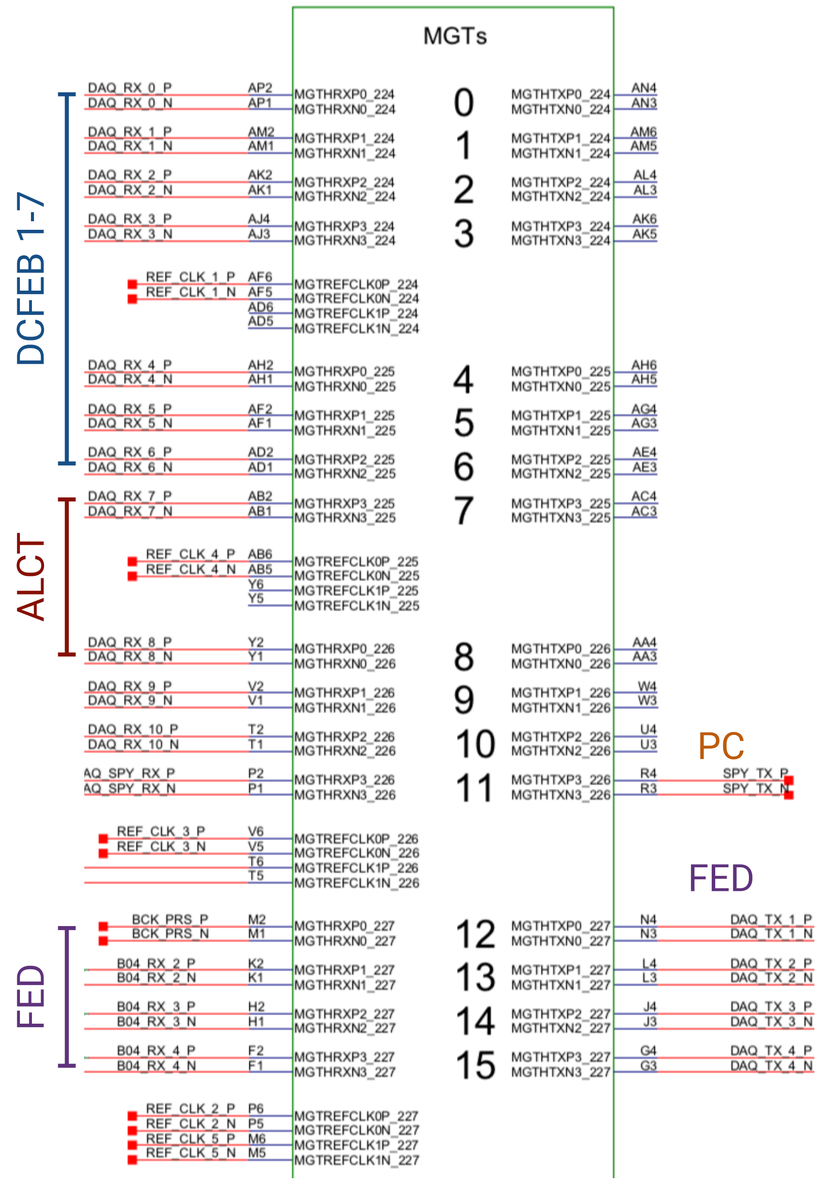
from Hualin



IBERT Test on ODMB7

- Overview for IBERT test
  - ❖ An automated Bit Error Rate test, simple to implement
  - ❖ Same config need to be applied to each quad

- Firmware goals
  - ❖ Test that all optical transmitter lines are working properly
  - ❖ All the ports need to be config to the same speed
  - ❖ Compiled IBERT firmware in 12.48 Gb/s and 4.0 Gb/s
    - ▸ 4.0 Gb/s motivated as Finisar for the SPY channel only support up to 4.0 Gb/s nominally
    - ▸ Both firmware versions have all TXs and RXs activated
    - ▸ Can set SPY_TX_P/N to low power mode (effectively switch off) as a protection in the 12.48 Gb/s version, this need to be done by Tcl script/manual after firmware loaded to the FPGA
  - ❖ Automated Eye Scan + Bit Error Rate test by Tcl script
    - ▸ runScan.tcl



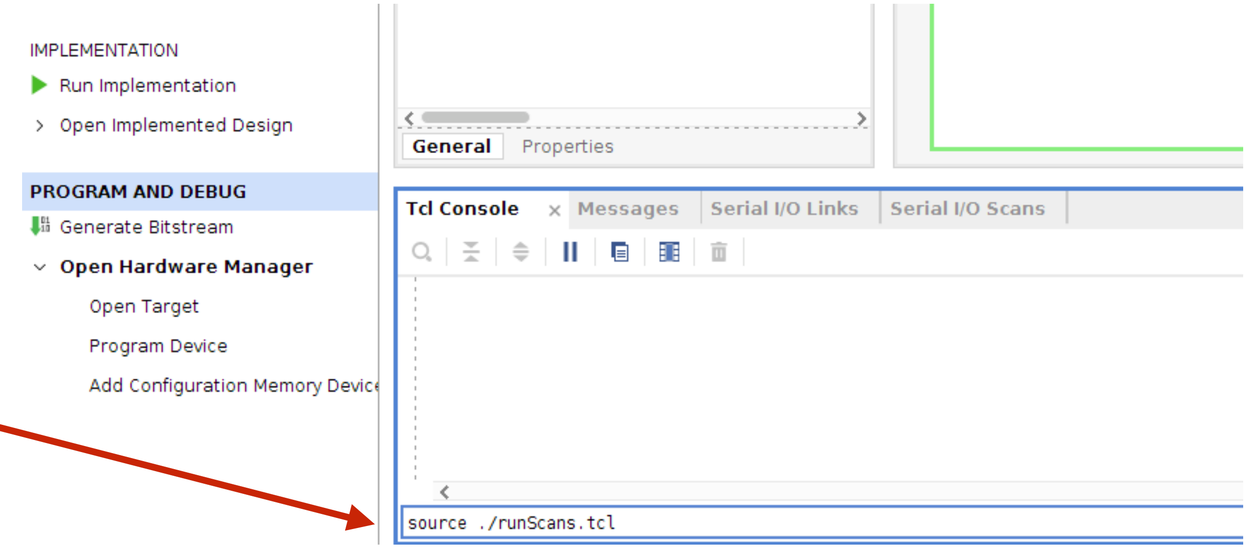
Auto IBERT Procedure

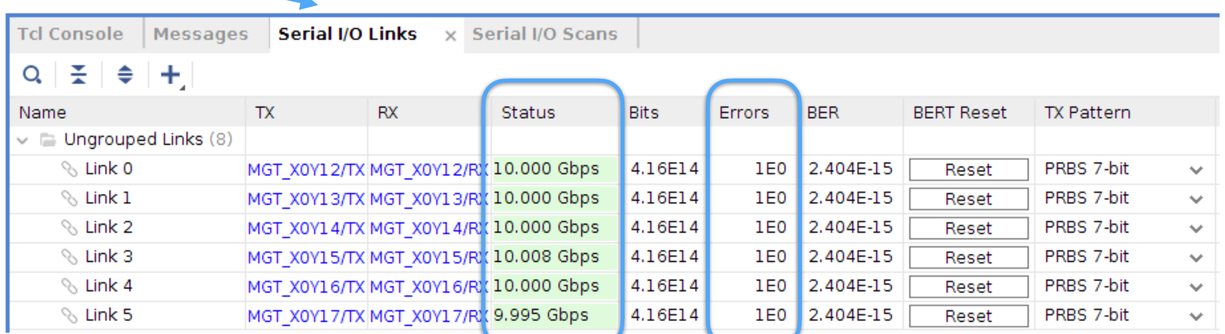
- Step 1: Preparation
  - Connect to board and load firmware *
  - Edit the constants at the top of `runScans.tcl`

- Step 2: Run script from the Tcl console *
  - `source runScans.tcl`

- Step 3: Evaluate results from “Serial I/O Links”
  - Verify that the Link status are good
  - Verify that the Error count is stable at one

  - Look at the IBERT scans plots (next slide)



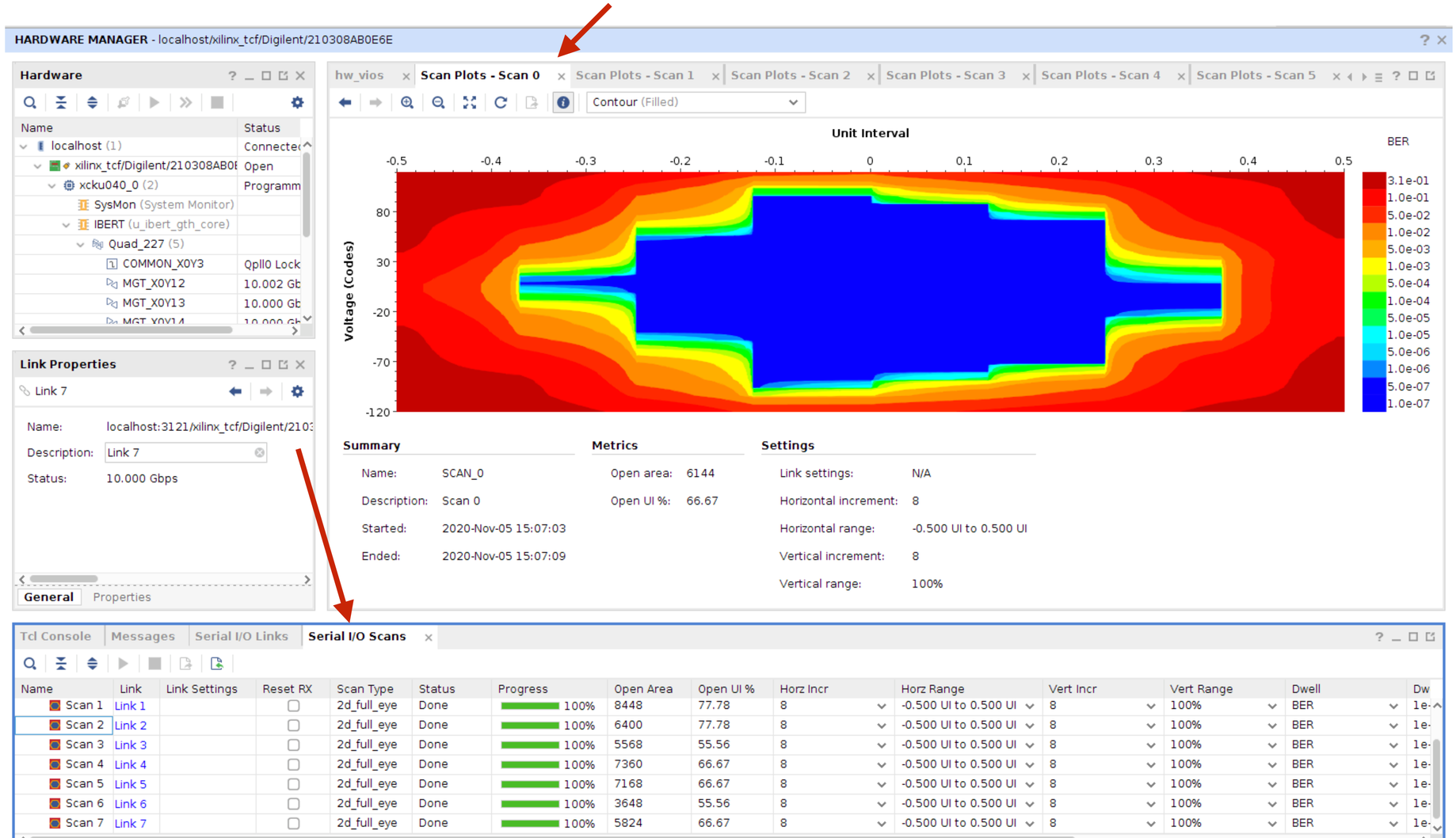


* Firmware and the automated script can be found at:
http://hep.ucsb.edu/cms/odmb_noCVS/firmware/odmb7/preproduction/optical_loopback_ibert/

IBERT Eye Scan

- The scans will pop up one by one during the script running
- The Open area should be the main indicator on link quality

Running eye scan will create errors to the link, so it's normal to see error count increase during running of the eye scan, and the link shall be reset afterward



Name	Link	Link Settings	Reset RX	Scan Type	Status	Progress	Open Area	Open UI %	Horz Incr	Horz Range	Vert Incr	Vert Range	Dwell	Dw
Scan 1	Link 1			2d_full_eye	Done	100%	8448	77.78	8	-0.500 UI to 0.500 UI	8	100%	BER	1e-
Scan 2	Link 2			2d_full_eye	Done	100%	6400	77.78	8	-0.500 UI to 0.500 UI	8	100%	BER	1e-
Scan 3	Link 3			2d_full_eye	Done	100%	5568	55.56	8	-0.500 UI to 0.500 UI	8	100%	BER	1e-
Scan 4	Link 4			2d_full_eye	Done	100%	7360	66.67	8	-0.500 UI to 0.500 UI	8	100%	BER	1e-
Scan 5	Link 5			2d_full_eye	Done	100%	7168	66.67	8	-0.500 UI to 0.500 UI	8	100%	BER	1e-
Scan 6	Link 6			2d_full_eye	Done	100%	3648	55.56	8	-0.500 UI to 0.500 UI	8	100%	BER	1e-
Scan 7	Link 7			2d_full_eye	Done	100%	5824	66.67	8	-0.500 UI to 0.500 UI	8	100%	BER	1e-

More Details on the Script

- Configurable constants
  - FPGA name
  - The PRBS pattern sent and checked by TX/RX
  - Auto load firmware with the script (need to specify filename)
  - Disable the SPY_TX_P/N channels
    - Set to 1 for the 12.48 Gb/s version, 0 for the 4.0 Gb/s version
  - Tag: suffix to the log file

```
8   set DEVICE_NAME {xcku040_0}
9   set PRBS_PATTERN {PRBS 31-bit}

17  set programfpga 0
18  set bitfilename {}
19  set disable_spy_tx 0
20  set tag "test2"
```

- Steps performed by the script
  - Make a link between each available txs and rxs
  - Reset all good links and inject 1 error
  - Config and run Eye Scans on each good link
    - The **open area, open percentage, horizontal percentage** and **vertical percentage** will be written to a log file
  - Reset again and inject 1 error
  - Loop to fetch the BER values in certain time period
    - Write **total bits received, error count, bit error rate (BER), RX pattern** each turn into file for later plotting/analysis

* https://github.com/wsicheng/ODMBDevelopment/blob/master/SingleTestFWs/optical_ibert_gth/scripts/runScans.tcl