UT4 GTY toward 25 Gbps

Yun-Tsung Lai

KEK IPNS

ytlai@post.kek.jp

1

TRG weekly meeting

2nd Mar., 2023

UT4 GTY data link

- Now, some TRG firmwares are using 12.573 Gbps.
  - 16.764 Gbps: Source codes are made, but tested to be unstable with real firmware.
  - 25.146 Gbps: Frequent bit error.

- I resumed the work to make the 25.146 Gbps data link recently.
  - In this slide, the progress will be reported.

Line rate (Gbps)	data path width (bit)	userclk (MHz)	FIFO (dataclk)	FIFO (sysclk)	data/lane with dataclk (bit)	data/lane with sysclk (bit)	
5.588	64 (66)	84.67	3-8	3-2	170	42	
11.176	64 (66)	169.33	3-16	3-4	340	85	
8.382	64 (66)	127	1-4	1-1	256	64	
12.573	64 (66)	190.5	1-6	2-3	384	96	**We are using it**
16.764	64 (66)	254	1-8	1-2	512	128	**BER exists**
25.146	64 (66)	381	1-12	1-3	768	192	**Frequent BER**
~~18.8595~~	~~64 (66)~~		~~1-9~~		~~576~~		**Not supported**
~~20.955~~	~~64 (66)~~		~~1-10~~		~~640~~		**Not supported**
~~23.0505~~	~~64 (66)~~		~~1-11~~		~~704~~		**Not supported**

iBERT

- First, I use iBERT for GTY to study the reason of instability.

- Sweep: Check the eye scan under different conditions (sets of parameters). Mainly four parameters can be tuned:
  - RX termination voltage.
  - TXDIFFCTRL: Driver Swing Control.
  - TXPRECURSOR: Transmitter pre-cursor TX pre-emphasis control.
  - TXPOSTCURSOR: Transmitter post-cursor TX pre-emphasis control.
- Try to see if tuning on the parameters can help on the quality of transmission.

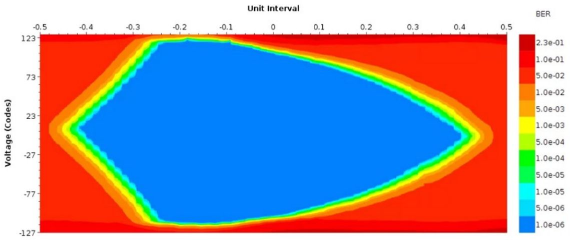
- There are also different options for the test data patterns in the eye scan:
  - PRBS-7
  - PRBS-9
  - PRBS-15
  - PRBS-23
  - PRBS-31
  - Fast clock
  - Slow clock

- Clock source: Input from lemo. Same as the real setup.

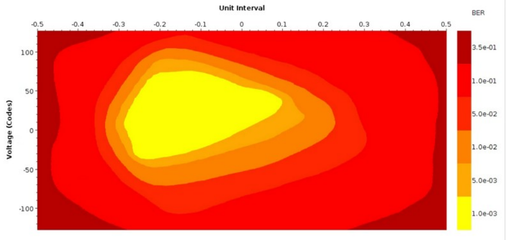
iBERT eyes of GTY0 lane0

- Just a demonstration. We can see the scan results differ with different test patterns.

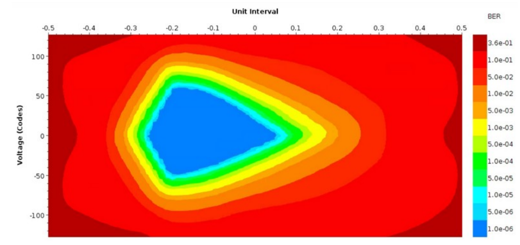
PRBS-7



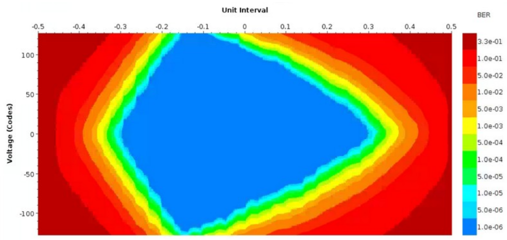
PRBS-9



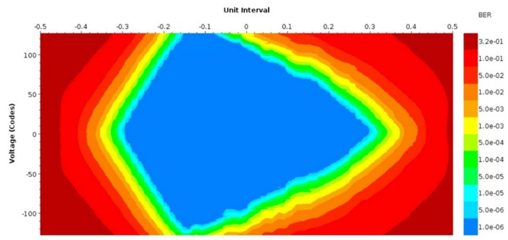
PRBS-15



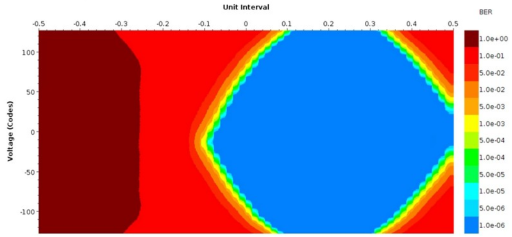
PRBS-23



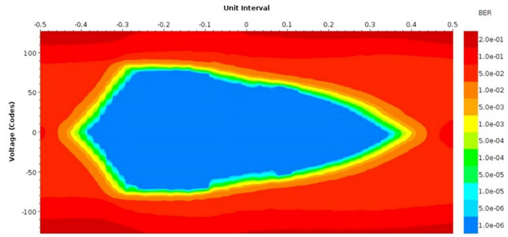
PRBS-31



Fast clock



Slow clock



iBERT sweep with TXDIFFCTRL

- Example from GTY1 lane2.
  - The second columns are open area of the eyes. → Quality of transmission.
- Conclusion: TXDIFFCTRL doesn't affect the quality.

TXDIFFSWING {191 mV (00000)}	3951
TXDIFFSWING {223 mV (00001)}	3897
TXDIFFSWING {254 mV (00010)}	3870
TXDIFFSWING {286 mV (00011)}	3870
TXDIFFSWING {315 mV (00100)}	3987
TXDIFFSWING {347 mV (00101)}	3960
TXDIFFSWING {378 mV (00110)}	3888
TXDIFFSWING {408 mV (00111)}	3870
TXDIFFSWING {439 mV (01000)}	3906
TXDIFFSWING {470 mV (01001)}	3888
TXDIFFSWING {499 mV (01010)}	3861
TXDIFFSWING {529 mV (01011)}	3852
TXDIFFSWING {556 mV (01100)}	3879
TXDIFFSWING {585 mV (01101)}	3906

TXDIFFSWING {191 mV (00000)}	3951
TXDIFFSWING {223 mV (00001)}	3897
TXDIFFSWING {254 mV (00010)}	3870
TXDIFFSWING {286 mV (00011)}	3870
TXDIFFSWING {315 mV (00100)}	3987
TXDIFFSWING {347 mV (00101)}	3960
TXDIFFSWING {378 mV (00110)}	3888
TXDIFFSWING {408 mV (00111)}	3870
TXDIFFSWING {439 mV (01000)}	3906
TXDIFFSWING {470 mV (01001)}	3888
TXDIFFSWING {499 mV (01010)}	3861
TXDIFFSWING {529 mV (01011)}	3852
TXDIFFSWING {556 mV (01100)}	3879
TXDIFFSWING {585 mV (01101)}	3906

iBERT sweep with TXPRECURSOR, TXPOSTCURSOR

- Example from GTY0 lane0.
  - Scan with large ranges.

- Conclusion: Small values (close to 0) for both are obviously better.
  - Default values from IPcore: TXPOST {6.02 dB (10100)} TXPRE {0.00 dB (00000)}

TXPOST {0.00 dB (00000)} TXPRE {0.00 dB (00000)}	6360
TXPOST {0.00 dB (00000)} TXPRE {4.08 dB (01111)}	6246
TXPOST {0.00 dB (00000)} TXPRE {6.02 dB (11111)}	6238
TXPOST {4.08 dB (01111)} TXPRE {0.00 dB (00000)}	5954
TXPOST {4.08 dB (01111)} TXPRE {4.08 dB (01111)}	6184
TXPOST {4.08 dB (01111)} TXPRE {6.02 dB (11111)}	2432
TXPOST {12.96 dB (11111)} TXPRE {0.00 dB (00000)}	50
TXPOST {12.96 dB (11111)} TXPRE {4.08 dB (01111)}	0
TXPOST {12.96 dB (11111)} TXPRE {6.02 dB (11111)}	1

iBERT sweep with TXPRECURSOR, TXPOSTCURSOR

- Example from GTY1 lane2.
  - Scan with small ranges.

- Conclusion: Hard to tell if there is obvious difference.
  - **Will try:**
    TXPOST {6.02 dB (10100)}
    TXPRE {0.00 dB (00000)}
    →
    TXPOST {0.00 dB (00000)}
    TXPRE {0.00 dB (00000)}

TXPOST {0.00 dB (00000)} TXPRE {0.00 dB (00000)}	3960
TXPOST {0.00 dB (00000)} TXPRE {0.22 dB (00001)}	3924
TXPOST {0.00 dB (00000)} TXPRE {0.45 dB (00010)}	4032
TXPOST {0.00 dB (00000)} TXPRE {0.68 dB (00011)}	3942
TXPOST {0.00 dB (00000)} TXPRE {0.92 dB (00100)}	3888
TXPOST {0.22 dB (00001)} TXPRE {0.00 dB (00000)}	3942
TXPOST {0.22 dB (00001)} TXPRE {0.22 dB (00001)}	3933
TXPOST {0.22 dB (00001)} TXPRE {0.45 dB (00010)}	3897
TXPOST {0.22 dB (00001)} TXPRE {0.68 dB (00011)}	3879
TXPOST {0.22 dB (00001)} TXPRE {0.92 dB (00100)}	3996
TXPOST {0.45 dB (00010)} TXPRE {0.00 dB (00000)}	3906
TXPOST {0.45 dB (00010)} TXPRE {0.22 dB (00001)}	3915
TXPOST {0.45 dB (00010)} TXPRE {0.45 dB (00010)}	3951
TXPOST {0.45 dB (00010)} TXPRE {0.68 dB (00011)}	3996
TXPOST {0.45 dB (00010)} TXPRE {0.92 dB (00100)}	3933
TXPOST {0.68 dB (00011)} TXPRE {0.00 dB (00000)}	3924
TXPOST {0.68 dB (00011)} TXPRE {0.22 dB (00001)}	3951
TXPOST {0.68 dB (00011)} TXPRE {0.45 dB (00010)}	4068
TXPOST {0.68 dB (00011)} TXPRE {0.68 dB (00011)}	3924
TXPOST {0.68 dB (00011)} TXPRE {0.92 dB (00100)}	3933
TXPOST {0.92 dB (00100)} TXPRE {0.00 dB (00000)}	3978
TXPOST {0.92 dB (00100)} TXPRE {0.22 dB (00001)}	3960
TXPOST {0.92 dB (00100)} TXPRE {0.45 dB (00010)}	3969
TXPOST {0.92 dB (00100)} TXPRE {0.68 dB (00011)}	3960
TXPOST {0.92 dB (00100)} TXPRE {0.92 dB (00100)}	3933

iBERT sweep with RX termination voltage

- For this sweep, the result of each lane differ, and it also depends on the test pattern.
- Here are the optimal result (the voltage giving the largest open area) of different lanes using different patterns.
  - The default value is 800 mV.
  - it is a kind of internal parameter hard-coded in IP. If we want to use different value, we need to re-generate the IP.
  - Also, we are not sure if this channel dependence is the same for all UT4 boards.
  - Let's keep using 800 mV for the time being.

pattern	00	01	02	03	10	11	12	23	32
7		550	500	700	550	700	700	700	600
9		500	500	500	600	550	350	500	700
15	350	600	700	700	700	700	700	600	700
23		600	700	500	600	850	850	550	700
31		600	700	700	600	600	700	700	700
fast		700	800	600	550	1100	550	800	700
slow		550	500	500	550	500	550	500	600

iBERT: conclusion

- TXDIFFCTRL: No need to change it.

- **TXPOSTCURSOR {6.02 dB (10100)} TXPRECURSOR {0.00 dB (00000)}**
  →
  TXPOSTCURSOR {0.00 dB (00000)} TXPRECURSOR {0.00 dB (00000)}

- RX termination voltage: Keep using 800 mV, but we can try smaller values.

Firmware test

- Then, we start to directly test the firmware with 25.146 Gbps GTY

- If I use:
  - **TXPOSTCURSOR {0.00 dB (00000)} TXPRECURSOR {0.00 dB (00000)}**
    - The transmission seems stable.
  - **TXPOSTCURSOR {6.02 dB (10100)} TXPRECURSOR {0.00 dB (00000)}**
    - Obviously unstable with lots of bit errors.

- Seems to be a critical change.
  - Next, let's see the results of long-term BERT.

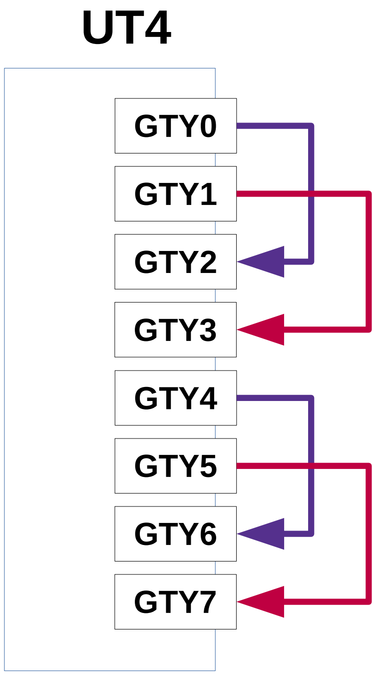
Setup for long-term BERT

- Use single UT4 and my protocol with
  - CDCTRG (31.75 MHz, 768 bits) and GDL (127 MHz, 192 bits) setup.
  - The reverse direction is also used (381 MHz, 64 bits)
  - PRBS-16.

- RX termination voltage:
  - 800 mV: No error in 3 days. BER < 3.5×10^{-18}.
  - 700 mV: No error in 3 days. BER < 3.5×10^{-18}.
  - 550 mV: After 1.5 days, frequency bit error started to happen. But no error in the rest of the time.

- I think we can try to use it.

- Other concerns:
  - Situation seems to depend on firmware compilation.
    - Link down and require reprogramming.
    - Bit error out of a sudden.
    - Late response from FIFO: Data shift.
  - We anyway need to monitor the check-sum while using it.



Latency

- New results for 25.146 Gbps

CDCTRG: 32 MHz dataclk

Latency (sysclk)	Total
UT3 5.58 Gbps	54 (420 ns)
UT4 5.58 Gbps	66 (515 ns)
UT4 8.38 Gbps	46 (359 ns)
UT4 8.38 Gbps no TXFIFO	39 (304 ns)
UT4 8.38 Gbps no TX, RXFIFO	28 (218 ns)
UT4 12.57 Gbps	40 (312 ns)
UT4 16.76 Gbps	32 (250 ns)
UT4 25.15 Gbps	29 (226 ns)

TOP-GRL-GDL: 127 MHz sysclk

Latency (sysclk)	Total
UT3 5.58 Gbps	~450 ns
UT4 5.58 Gbps	64 (500 ns)
UT4 8.38 Gbps	44 (343 ns)
UT4 8.38 Gbps no TXFIFO	39 (304 ns)
UT4 8.38 Gbps no TX, RXFIFO	28 (218 ns)
UT4 12.57 Gbps	34 (265 ns)
UT4 16.76 Gbps	31 (242 ns)
UT4 25.15 Gbps	25 (192 ns)

Summary & To do

- UT4 GTY with 25.146 Gbps is studied and developed.
  - Using iBERT to find the best parameters.
  - Good result in long-term BERT.
  - All the protocol source codes have been made.

- To do:
  - Any module wants to try it first?
    - Check-sum monitor is needed to check it carefully.
  - Will do the same test with
    - 16.764 Gbps: According to Koga-san, it was not so stable.
    - 10.16 Gbps: In order to connect to the new CDC FEE (Mk. II) using 8B/10B.
    - I will do them at a slow pace.

  - Merger firmware update in EHut with Koga-san. Next week?
    - Also discussion about new firmware and the check-sum monitor in SLC.

New CDC FEE (Mk. II)

- Now I am doing some tests with the GTX transmission.
  - Some iBERT are done.
  - Both 2.54 Gbps with 8B/10B (for Belle2Link) and 10.16 Gbps (for TRG) are tested. No major problem.
  - Will then start to make protocol for TRG link to UT4. Also, at a slow pace.

- After some more tests, we will then discuss with E-sys group and CDC group.
  - Then, I will start to modify the circuit schematic.