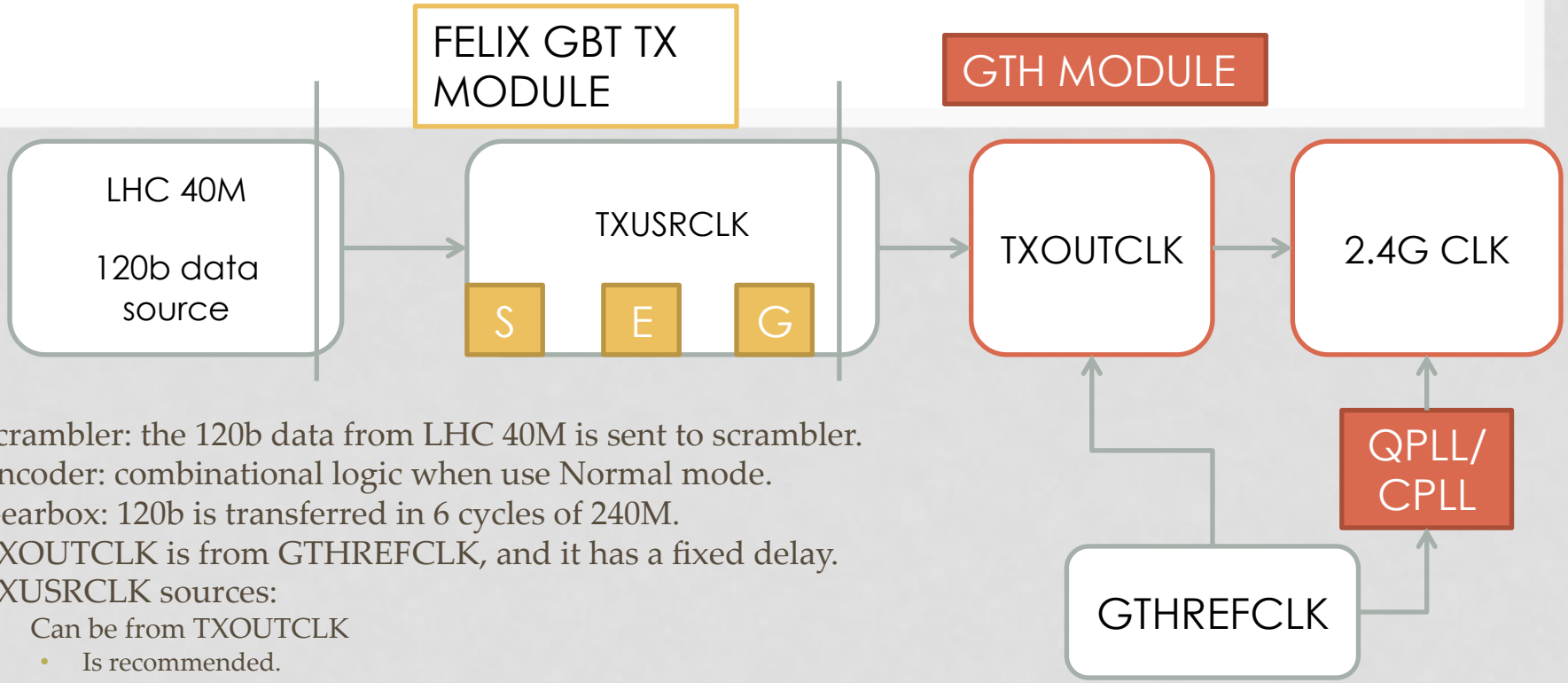
INTRODUCTION TO THE FELIX GBT CORE

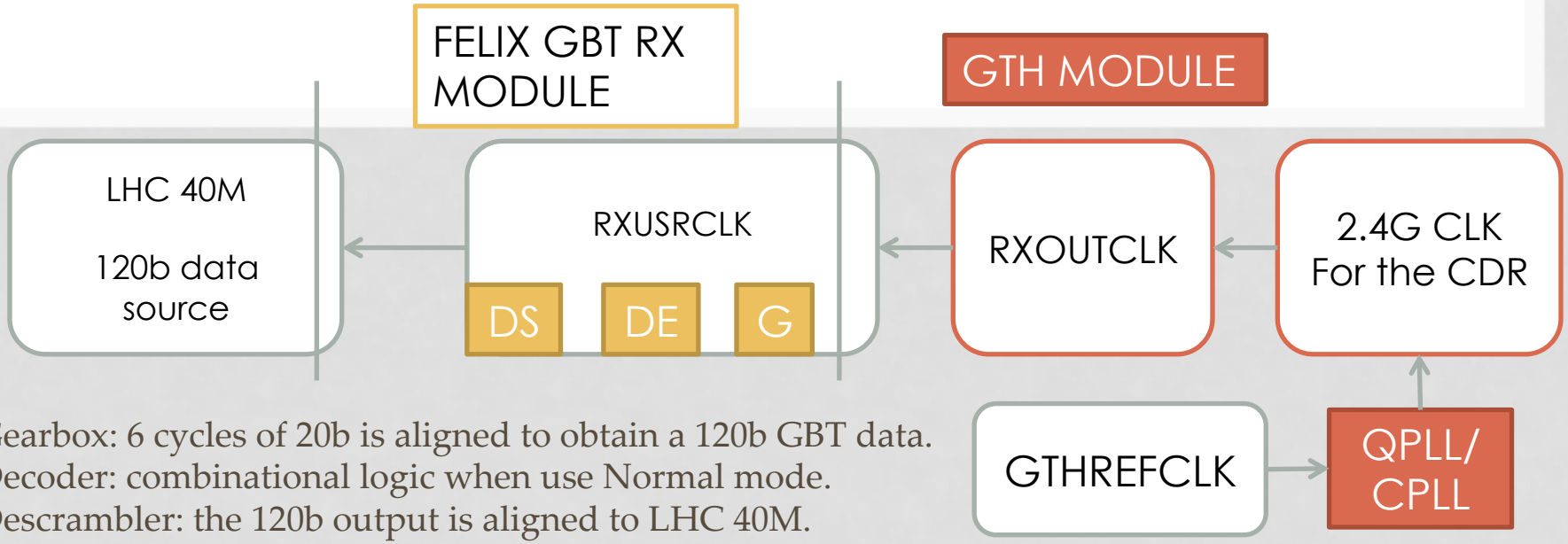
KAI CHEN
BNL
(WILL KEEP UPDATING)
2014/04/01

TX CLOCK DOMAINS



- Scrambler: the 120b data from LHC 40M is sent to scrambler.
- Encoder: combinational logic when use Normal mode.
- Gearbox: 120b is transferred in 6 cycles of 240M.
- TXOUTCLK is from GTHREFCLK, and it has a fixed delay.
- TXUSRCLK sources:
  - Can be from TXOUTCLK
    - Is recommended.
    - Need to handle the time domain crossing from 40M->TXOUTCLK.
      - Can be guaranteed without error.
      - If rising edges of the 40M & TXOUTCLK is close, latency may be unstable when power recylcling.
  - Can be from a MMCM (LHC 40M => 240M).
    - Can adjust phase of 240M
      - To make the latency from 40M => 240M stable.
      - Also need to make no error happens from TXUSRCLK=>TXOUTCLK, and with a fixed latency.
- If phase between GTHEREFCLK and LHC 40M is unstable, then a deterministic latency is hard to be guaranteed.
- Phase of 2.4G:
  - CPLL: 240 => 2.4G
  - QPLL: 240 => 9.6G => 4.8G => 2.4 G, phase may change.

RX CLOCK DOMAINS



- Gearbox: 6 cycles of 20b is aligned to obtain a 120b GBT data.
- Decoder: combinational logic when use Normal mode.
- Descrambler: the 120b output is aligned to LHC 40M.
- RXOUTCLK is generated from the 2.4G data.
  - The data is 2.4G DDR, so the recovered 2.4G may has a 1/4.8G phase difference when we reset GTH. This is named Odd/Even problem in our design.
  - The RXOUTCLK can be shifted with a step of 1/2.4G, totally 10 phases.
- RXUSRCLK can be from RXOUTCLK.
  - When we have a big channel number, we must save clock resources. We can choose some channels as Master channels, their RXOUTCLK are used as RXUSRCLK for some other channels.
  - RXOUTCLK has a bad quality.
- RXUSRCLK can be from a MMCM, LHC 40M is used to generate the 240M.
- We need to adjust the RXOUTCLK to make sure data from RXOUTCLK => RXUSRCLK has no error, will show the details later.
- The Descrambler's 120b output will be aligned to the LHC 40M.

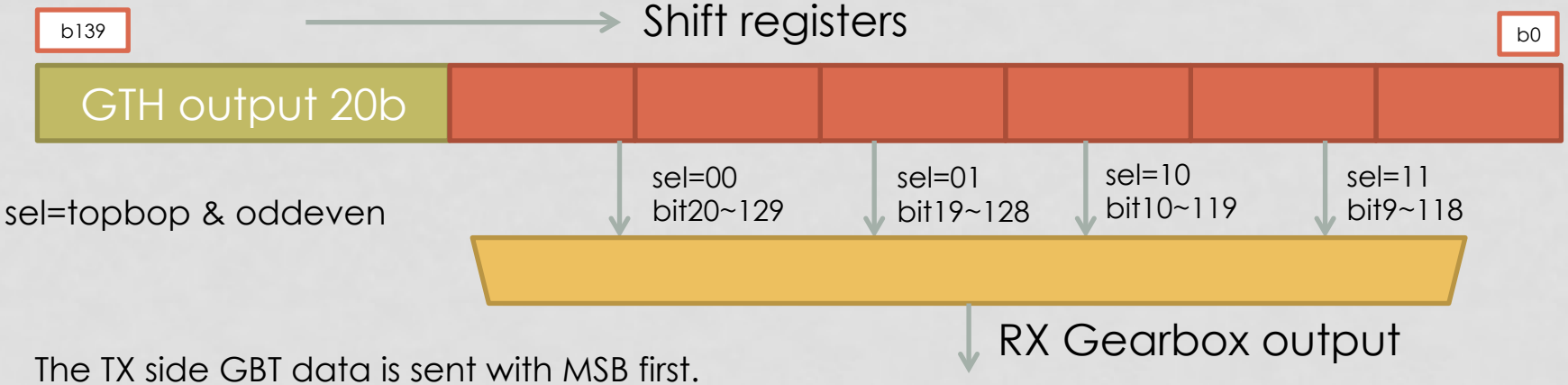
PHASES ABOUT THE RX ALIGNMENT

GTH output 20b is LSB first

16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	19	18	17

After the reset

We can use RXSLIDE to do the bitslip, to shift the 20b with step of 2b.
When we do the shift, the RXOUTCLK will also be shifted by 1/2.4G.



- The TX side GBT data is sent with MSB first.
- For RX Gearbox's output, headers can be on the following bits of the shift register.
  - b20~23 (topbot=0, oddeven=0)
  - b19~22 (topbot=0, oddeven=1)
  - b10~13 (topbot=1, oddeven=0)
  - b9~12 (topbot=1, oddeven=1)
- For each sel value, we have 10 phases. So we totally have 40 phases.

HOW TO DO THE RX ALIGNMENT

- The Oddeven value is fixed after the each GTH reset (but we still need to scan the phases to know it's 1 or 0), so actually we have *two possible sel values at the most*.
  - One in the 10 phases with Topbot=0, the other in the 10 phases with Topbot=1.
  - For these two choices, when the header is found, the RXOUTCLK will be shifted by half cycle of 240M.
  - Inside the GTH, data will be transferred from RXOUTCLK domain to RXUSRCLK domain. When phase between the RXUSRCLK and RXOUTCLK is unknown, the data may be wrong.
  - When we have *two choices*, at least one will make sure no error happens. So finally we have *1 or 2 choices.*
- When we use MMCM output as the RXUSRCLK.
  - The first step: set Topbot=0, scan the 20 phases with Oddeven is 0 and 1.
  - If we can not obtain a stable GBT header(RxDataValid is always 1, or RxAlignmentOK keep 1 for some time after the reset of the Rx_Alignment_Checker) for these 20 phases, then it means RXUSRCLK and RXOUTCLK has a bad phase for *this choice (Topbot=0)*.
    - Then we can set Topbot=1, to scan the other 20 phases to find the Oddeven value, and to find the right phase.
  - If we can obtain a stable GBT header with Sel=01 or 00, Then we change the Topbot to 1, and shift phase by 5 (half 240M cycle), with trigger RXSLIDE 10 times.
    - If for Topbot=1, we can not obtain stable GBT header, then we can change back to Topbot=0, and shift phase by 5 (half 240M cycle), with trigger RXSLIDE 10 times.
    - Else:
      - we also change back to Topbot=0, and shift phase by 5. This phase is phase 0.
      - We can shift the RXOUTCLK with these 10 phases, and use RXUSRCLK to sample them, get a 10 bits value (register RxOutClk_Sampled) for the phase0-phase9. From this value, we obtain the phase between RXOUTCLK & RXUSRCLK for this sel value*(Topbot=0)*.
      - The software need to choose the better topbot value, according to the 10 bits value.
  - This RXUSRCLK has a good quality, while the RXOUTCLK has a jitter of about 1/2.4G.

HOW TO DO THE RX ALIGNMENT

- When we use master RXOUTCLK output as the RXUSRCLK, for example, a quad has 1 master channel and 3 slave channels.
  - For slave channels, both of the RXOUTCLK and RXUSRCLK have bad quality, when data is transferred from one domain to the other, 1~3 phases of the 10 will have error.
  - The first step, process the master channel.
    - Set Topbot=0, scan the 20 phases, to obtain the right phase to obtain stable GBT header.
    - We don't need to test Topbot=1, since RXOUTCLK=RXUSRCLK.
  - The second step, for slave channels.
    1. Set Topbot=0, scan the 20 phases, to find phase which has stable GBT header.
       - If no good phase, then we set Topbot=1, and find the good phase, then end the operation for this channel.
    2. If we find a good phase, then we know Oddeven now, keep it unchanged for following operations. Set Topbot=1, shift phase by 5 (10 RXSLIDE trigger).
       - If no stable GBT is obtained, then go back: Set Topbot=0, shift phase by 5 (10 RXSLIDE trigger), then end the operation for this channel.
       - If stable GBT is obtained, also go back: Set Topbot=0, shift phase by 5 (10 RXSLIDE trigger). Both Topbot=1 and Topbot=0 have a good phase, go to next step, to choose a better(stable) one.
    3. Keep the slave channel configuration unchanged, it has stable GBT header now. We shift the phase of RXUSRCLK (the master RXOUTCLK), for 1~3 of the 10 phases, this slave channel will have no stable GBT header. If it has stable GBT header, we obtain a 1, else 0, totally 10b value 10b-A is obtained. Meanwhile, we record the 10b-B results using the 10 shifted RXUSRCLK to sample same RXOUTCLK.
       - According to the 10b-A results, we can choose the Topbot value. For step 3, the test results shows that for some channels, only 1 of the 10 phases has no stable GBT header, and it need a long time to has error happen.
       - *For these 10b-A = "1111111111" channel, we can use 10b-B to choose Topbot alue.*
       - *We can also use 10b-B for all channels.(better)*

HOW TO DO THE RX ALIGNMENT

- *For both RXUSRCLK cases, Oddeven value is determined by the GTH reset, but Topbot value is set via software.*
- When MMCM output is used as RXUSRCLK, the selection based on the 10b value:
  - When the first 1 is at 1-5 then Topbot=1
  - When the first 1 is at 6-0 then Topbot=0
- When master channel RXOUTCLK is used as RXUSRCLK, for slave channels.
  - If use 10b-A, center is the mean value of the 0 bit's positions.
    - When 0 number is even, center=floor(center)
    - When center of the 0 is 3-7, Topbot=0
    - When center of the 0 is 0-2, or 8-9, Topbot=1
  - If use 10b-B (recommended)
    - When the first 0 is at 7-1 then Topbot=1
    - When the first 0 is at 2-6 then Topbot=0

- The 240M from local MMCM will be used as RXUSRCLK, for future FELIX development.
- If GBT is needed to recover the LHC clock, then we can use the RxOutClk of the master channel, to generate the 40MHz clock. Descrambler_enable can be used to make the 240M->40M always at the same phase of the 6. The MMCM can also shift the output by 1 UI (1/4.8G) when OddEven='1', to solve the odd/even problem, and make the latency of 40M clock fixed, compared with the Tx side.