TTCrx Issues

M.Matveev
Rice University
December 17, 2009
8-bit fine delay value can be programmed into the TTCrx ASIC via the I2C bus (write and read) or via the optical link (write only)

- Used to delay clock, L1A and broadcast commands
- 104 ps step within 25 ns clock period
- Highly non-linear dependence between the 8-bit value and actual delay
- By default set to “0” on power up
- Set once on every CCB at the initialization step via I2C
Fine Delays Issues (1)

- Richard Kellogg (HCAL) has recently reported “mis-lock” problems with their TTCrx (~350 TTCrx chips, not a TTCrq mezzanine, but similar design)

- See his report
  http://indico.cern.ch/getFile.py/access?contribId=1&resId=3&materialId=slides&confId=75770
Fine Delays Issues (2)

- it turns out that the TTCrx powers up into one of two states:
  - the “good” state (with ~95% prob)
  - the “50% mislock” state (with ~5% prob)

Hardware reset of TTCrx (50 us pulse to reset_b pin) doesn’t (always) help. Proposed solution: interrupt the TTC optical signal. This works 100% in the HCAL lab.
Test 1

- Apply hardware reset to TTCrx
- Wait 2 seconds
- Monitor phase shift between Clock40 non-deskewed clock and QPLL (deskewed) clock outputs by eye
- Repeat 200 times
- Repeat on 3 CCB boards

Result: phase shift is always constant, no “unlock” states
Test 2

- Apply hardware reset reset_b to TTCrx
- Wait 2 seconds
- Write 0x5e to the fine delay register
- Wait 2 seconds
- Monitor phase shift between Clock40 non-deskewed clock and QPLL (deskewed) clock outputs by eye
- Repeat 200 times
- Repeat on 3 CCB boards

Result: phase shift was always constant on 2 boards. On one board there were 2 errors (wrong delays). No “unlock” states.
Test 3

- Write 0xe (delay = 0) to the fine delay register
- Wait 2 seconds
- Write 0x5e (delay = 8 ns) to the fine delay register
- Wait 2 seconds
- Monitor phase shift between Clock40 non-deskewed clock and QPLL (deskewed) clock outputs by eye
- Repeat 200 times
- Repeat on 3 CCB boards

Result: 4..5 times (for every board and every test iteration) there was a difference of ~1.5 ns. In ~10% of all iterations there was no change in the phase with the new delay loaded.
Test 4

- Power cycle the TTCrx
- Wait 2 seconds
- Write 0x5e to the fine delay register
- Wait 2 seconds
- Monitor phase shift between the Clock40 non-deskewed clock and QPLL (deskewed) clock outputs by eye
- Repeat 100 times

Result: phase shift was different in 9 cases. In one case the QPLL did not lock. Another write to fine delay register without additional power cycling or reset cured the problem.
Test 5

- Power cycle the CCB
- Wait 2 seconds
- Apply hardware reset (reset_b pin) to TTCrx
- Wait 2 seconds
- Write 0x5e to the fine delay register
- Wait 2 seconds
- Monitor phase shift between Clock40 non-deskewed clock and QPLL (deskewed) clock outputs by eye
- Repeat 100 times

Result: phase shift was always constant. In 6 cases the new delay value made effect only after the second attempt. QPLL was always locked.
Conclusion

- There is low probability (few %) that the actual fine delay value will differ from what was written into the register. The problem is solely related to TTCrx. Hardware reset makes things better.

- Our procedures are adequate. They include the initial hardware reset, loading of fine delay with verification and periodical monitoring of “TTC_Ready”, “QPLL_Locked” statuses and their changes (error counters).

- Based on existing data, do we have any evidence of wrong fine delays due to malfunctioning TTCrx?