Differential Line Test for the FSSR2 ASICs of the Silicon Vertex Tracker

Brian Eng, Mary Ann Antonioli, Sahin Arslan, Peter Bonneau, David Butler, George Jacobs, Mindy Leffel, Tina Mann, Marc McMullen, Anatoly Sitnikov, Robert Teachey, and Amrit Yegneswaran
Physics Division, Thomas Jefferson National Accelerator Facility, Newport News, VA 23606
June 23, 2015

This note is about the differential line test (DLT) developed to identify any problems of no data output from FSSR2 ASICs on the low voltage differential signaling (LVDS) lines for the Silicon Vertex Tracker (SVT).

During the fabrication of the SVT modules, wire bonds were checked after each assembly step. Excessive current draws indicated problems with wire bonds for power and ground. Problems with no data output associated with wire bonds that connected to LVDS lines required special tests.

Most of the common control LVDS lines were checked by the register test (writing/reading register values); the remaining LVDS lines were checked using DLT.

DLT is faster than manually probing each line individually and is easier for non-experts. DLT exercises the FSSR2 ASICs thoroughly and reliably, increasing the chances of finding problems and reducing the chances of false-positive failures.

DLT's testing procedure, Appendix A, is to inject data/state and compare this injected data/state with the output in the data word. For example, enable a single channel and look for the output data in the data word. DLT checks both data and status words. DLT puts the ASIC in any one of the four output line modes and tests each LVDS output line of that mode, Table I.

Output line #	Output line mode			
	1	2	4	6
1	x	x	x	x
2		x	x	x
3			x	x
4			x	x
5				x
6				x

TABLE I. Output lines tested in various modes.

In the DLT program, the output of the ASIC was programmed to provide detailed information about the status of that ASIC. Additionally, the 2.11 firmware of the VXS Silicon Control Module (VSCM) provided the ability, via a register, to check the last received data or status word.

Testing showed that it was possible for the VSCM to get stuck synced to a data word instead of a status word (it is possible for a data word to match the sync bits of a status word) when switching output line modes, which causes the VSCM to never see any status words. A new firmware (2.12) for the VSCM was developed to address this issue.

Testing the LVDS lines with DLT showed that ASICs which provided no output data were in fact failing only a specific output line mode. The line itself was good and the problem lay elsewhere, perhaps in the ASIC.

DLT has been used both at Jefferson Lab and at Fermilab during module production. All modules produced were tested. Only one ASIC out of 364 was identified to have output problems, and was replaced.

APPENDIX A: DIFFERENTIAL LINE TEST PROCEDURE

Output Clock

- Puts chip in 1 output line mode and counts number of status words
  - Note: This could fail due to out1 being bad

MasterReset

- Sets register 3 to 255, performs the MasterReset, register 3 should now return 139
  - Note: Register test is performed prior to differential line test

GotHit & CoreTalking

- Both lines are tested in the same fashion, but are tested separately
- First all discriminators are set to 255 so that the number of hits returned should be 0
  - If any hits are returned, line fails test
- Then the hit/no-hit discriminator is reduced to 200 and 1000 pulses @ 300 mV are injected
  - If the number of hits is <995 or >1005 (0.5% to allow for noise hits), line fails test

Out6

- In 6 output line mode (which has 4 bits): checks bits 22, 23 of status word
  - If result is not 4 (corresponds to only RejectHits being set), line fails test
- Clear RejectHits and set SendData
  - If result is not 8 (SendData set, RejectHits cleared), line fails test

Out5

- In 6 output line mode (which has 4 bits): check bits 17, 18 of status word
  - Perform a MasterReset
    - If either bit is set, line fails test
  - Set modulo register, enable internal pulser
    - If both bits aren't set, line fails test

Out4

- In 4 output line mode (which has 6 bits): check bits 18, 22, 23 of status word
  - Perform a MasterReset
    - If anything besides bit 22 is set, line fails test
  - Clear RejectHits, set SendData, set modulo register
    - If not the expected pattern, line fails test
- In 6 output line mode (4 bits): check bits 15-12 (column number) of data word. In all cases below, set hit/no-hit threshold to 200, and inject pulses @ 300 mV
  - Set kill and inject mask to enable channel 4 (0b1010 column)
  - Set kill and inject mask to enable channel 77 (0b0101 column)
  - If either pattern doesn't match, line fails test

Out3

- In 4 output line mode (which has 6 bits): check bit 17 of status word
  - Perform a MasterReset, check that bit is clear
    - If bit is set, line fails test
  - Perform a MasterReset, set SendData
    - If bit isn't set, line fails test
- In 6 output line mode (which has 4 bits): check 4 MSB of BCO in data word
  - Set up chip to send hits
  - Set hit/no-hit to 200
  - Inject pulses @ 300 mV
  - Sync pulses to BCO 0
    - If BCO isn't 0, line fails test
  - Sync pulses to BCO 80 (0b01010000)
    - If BCO isn't 80, line fails test
  - Sync pulses to BCO 160 (0b1010000)
    - If BCO isn't 160, line fails test

Out2

- In 2 output line mode (which has 12 bits): check bits 18, 22, 23 of status word
  - Perform a MasterReset
    - If anything beside bit 22 is set, line fails test
  - Clear RejectHits, set SendData, set modulo register
    - If not the expected pattern, line fails test
- In 4 output line mode (which has 6 bits): check 6 MSB of BCO in data word
  - Set up chip to send hits
  - Set hit/no-hit to 200
  - Inject pulses @ 300 mV
  - Sync pulses to BCO 0
    - If BCO isn't 0, line fails test
  - Sync pulses to BCO 84 (0b01010100)
    - If BCO isn't 84, line fails test
  - Sync pulses to BCO 168 (0b1010100)
    - If BCO isn't 168, line fails test
- In 6 output line mode (which has 4 bits): check 4 LSB of BCO in data word
  - Set up chip to send hits
  - Set hit/no-hit to 200
  - Inject pulses @ 300 mV
  - Sync pulses to BCO 0
    - If BCO isn't 0, line fails test
  - Sync pulses to BCO 5 (0b00000101)
    - If BCO isn't 5, line fails test
  - Sync pulses to BCO 10 (0b00001010)
    - If BCO isn't 10, line fails test

Out1

- In 1 output line mode (which has 24 bits): check bits 18, 22, 23 of status word
  - Perform a MasterReset
    - If anything beside bit 22 is set, line fails test
  - Clear RejectHits, set SendData, set modulo register
    - If not the expected pattern, line fails test
- In 2 output line mode (which has 12 bits): check BCO in data word
  - Set up chip to send hits
  - Set hit/no-hit to 200
  - Inject pulses @ 300 mV
  - Sync pulses to BCO 0
    - If BCO isn't 0, line fails test
  - Sync pulses to BCO 85 (0b01010101)
    - If BCO isn't 85, line fails test
  - Sync pulses to BCO 170 (0b10101010)
    - If BCO isn't 170, line fails test
- In 4 output line mode (which has 6 bits): check 2 LSB in data word
  - Set up chip to send hits
  - Set hit/no-hit to 200
  - Inject pulses @ 300 mV
  - Sync pulses to BCO 0
    - If BCO isn't 0, line fails test
  - Sync pulses to BCO 1 (0b00000001)
    - If BCO isn't 1, line fails test
  - Sync pulses to BCO 2 (0b00000010)
    - If BCO isn't 2, line fails test
- In 6 output line mode (which has 4 bits): check fADC in data word
  - Set up chip to send hits
  - Inject pulses @ 300mV
  - Set all discriminators to 255
  - fADC should be 0
    - If fADC isn't 0, line fails test
  - Set all discriminators to 0
  - fADC should be 7
    - If fADC isn't 7, line fails test