

MCC-I2.1

MCC Design Group

Production Test

Issue:
Revision: 1.0
Reference: ATLAS ID-xx
Created: 8 December 2003
Last modified: 31 December 2003 10:00
Edited By: R. Beccherle and G. Darbo
ATLAS / Pixel Collaboration

Abstract

This document describes the production test for the MCC-I2.-1 chip.

Table Of Contents

	Abstract	2
	List of Figures	5
	Production Test: Specifications & Results	7
1	Production Test Specification.	9
1.1	Introduction.	9
1.2	Test Specification.	9
1.2.1	Continuity test (test# 100÷110).	9
1.2.2	Supply current (test# 200÷225).	9
1.2.3	Input leakage (test# 300÷324).	10
1.2.4	VDD sweep (test# 350÷360).	10
1.2.5	Vin threshold for LVDS and CMOS inputs (test# 400÷487).	10
1.2.6	Vout for LVDS and CMOS (test# 500÷551).	11
1.2.7	Digital Functional and Scan Chain (test# 600÷608).	11
1.2.8	Delay line (test#700÷799).	12
1.3	Additional Documentation.	13
2	Production Test Results.	15
2.1	Wafer Yields.	15
2.2	Input leakage current.	18
2.3	VDD test.	18
2.4	Vout test.	20
2.5	Delay measurement for the Delay line..	20
3	Appendix: Test Vectors - SimPix Source	23
3.1	Delta-01.spx	23
3.2	Delta-02.spx	31
3.3	Delta-03.spx	33
3.4	Delta-04.spx	39
3.5	Delta-05.spx	40
3.6	Delta-06.spx	42
3.7	Delta-07.spx	45
3.8	Delta-08.spx	52
3.9	Delta-09.spx	53
4	Test Vectors: Scan Chain.	55
4.1	Scan Chain: Fault Coverage.	55
5	Test Logfile	57
5.1	Logfile for a Passed die	57

List of Figures

- Fig. 2-1** p. 15 Rejected chips per wafer.
- Fig. 2-2** p. 16 Histogram of digital test fails for the different vector files.
- Fig. 2-3** p. 17 Wafer maps of the 6 production wafers.
- Fig. 2-4** p. 18 VDD scan for the 6 wafers. Curves represent the number of chip failing the test. Chip are accepted if passing the test at VDD supply voltage lower than 1.87 V.
- Fig. 2-5** p. 19 VDD threshold for the 6 production wafers. Blue is the cumulative result, while purple is the number of chips non passing the test from that value of VDD and below.
- Fig. 2-6** p. 20 Correlation plot of VOH-VOL as function of VOH or VOL for two LVDS drivers (DTOp on the right and DTOn on the left).
- Fig. 2-7** p. 21 Delay line coefficients (Slope/Intercept) for 5 different values of range and 20 delay points per fit.
- Fig. 2-8** p. 22 Delay Line delays for the first 200 chips of wafer VX50V7X. Delays for range 0 (above) and for range 15 (below).

Production Test: Specifications & Results List of Tables

Table 1-2	<i>p. 10</i>	IIH and IIL test limits.
Table 1-1	<i>p. 10</i>	IDD test measurement limits.
Table 1-3	<i>p. 11</i>	Vout for LVDS and CMOS.
Table 1-4	<i>p. 12</i>	Digital test vectors for fault test
Table 2-1	<i>p. 15</i>	Summary yield table for the 6 production wafers.
Table 2-2	<i>p. 16</i>	Faulty chips found by the test vector files.

1 Production Test Specification

1.1 Introduction

This document reports on the production test of MCC-I2.1 chips. Chapter 1 describes the set of test executed on each component, while chapter 2 summarizes the results of the test and yield obtained. The production test has been commissioned to an external firm. The MCC design team has specified the test conditions and the test limits. The test executed on wafer on each MCC includes:

- Continuity test of I/O pads;
- IDD current versus clock frequency;
- Input leakage current;
- VDD voltage sweep;
- Vin threshold;
- Vout;
- Digital functional test;
- Time measurements of Delay Line.

The test have been made with a Theradyne Catalyst™ semiconductor test machine by the electronics testing department of Delta¹ in Denmark. Tests are made at 2.0 VDD supply voltage (unless differently required by specific test) and output LVDS signal are terminated to 100 Ω between positive and negative pads. Test temperature is ambient temperature (~25 °C).

1.2 Test Specification.

1.2.1 Continuity test (test# 100÷110)

This test measures the presence of the protection diodes at the input pads. Continuity is seen with a negative signal to GND and with a positive signal to VDD. Failure of this test is binned as *Bin#10*.

1.2.2 Supply current (test# 200÷225)

This test measures the static and dynamic supply current. Static current is measured with LVDS output drivers terminated to 100 Ω between positive / negative output pairs. The test vector file for this test is Delta-01.spx (see Appendix 3.1). In this vector file for 90% of the time there are WrFIFO and RdFIFO command. The IDD measured is an upper limit of what expected in normal operations. The test limits are shown in Table 1-1. In case of test failure the result is binned as *Bin#10*.

1. Delta: Danish Electronics, Light & Acoustics - www.delta.dk

Table 1-1 IDD test measurement limits.

Test#	Test Type	Lower Limit	Upper Limit
210	Static IDD	35.0 mA	60.0 mA
220	IDD @ 1.25 MHz	35.0 mA	70.0 mA
221	IDD @ 2.5 MHz	35.0 mA	75.0 mA
222	IDD @ 5 MHz	40.0 mA	80.0 mA
223	IDD @ 10 MHz	45.0 mA	95.0 mA
224	IDD @ 20 MHz	50.0mA	120.0 mA
225	IDD @ 40 MHz	80.0 mA	160.0 mA

1.2.3 Input leakage (test# 300÷324).

This test measures the leakage current of CMOS inputs. The test limits take in account the pull-up/pull-down resistance of 100 kΩ. of the CMOS.

The test vector file for this test is Delta-09.spx (see Appendix 3.9). The test limits are shown in Table 1-1. In case of test failure the result is binned as *Bin#6 (never happens)*.

Table 1-2 IIH and IIL test limits.

Test Type	Inputs	Lower Limit	Upper Limit
IIH - Pull-up	AMSDSM, RS1b	-1.0 μA	1.0 μA
IIH - Pull-down	TSTEN, TSTIN	15.0 μA	25.0 μA
IIL - Pull-up	AMSDSM, RS1b	-25.0 μA	-15.0 μA
IIL - Pull-down	TSTEN, TSTIN	-1.0 μA	1.0 μA

1.2.4 VDD sweep (test# 350÷360)

This test simulates a “digital injection scan” with response from 16 FE chips. The test vectors repeat 12 times a loop with 16 contiguous LV1 generated by the MCC and 90 hits per FE generated in the 6th of the 16 coming back events. The test is repeated 30 times, lowering VDD from 2.00 V to 1.70 V in steps of 10 mV.

The MCC are accepted if they work at VDD<1.87 V. Failure of the test is binned as *Bin#8*.

1.2.5 Vin threshold for LVDS and CMOS inputs (test# 400÷487).

For this test a short test vector file (Delta-09, Appendix 3.9) is used.

1.2.6 Vout for LVDS and CMOS (test# 500÷551).

For this test a short test vector file (Delta-09, Appendix 3.9) is used. The output voltage is measured stopping the test at given vectors. Test limits for this test are in the Table 1-3. Failure of the test is binned as *Bin#9*.

Table 1-3 Vout for LVDS and CMOS.

Test Type	Outputs	Lower Limit	Upper Limit
VOL (LVDS)	DTO, DTO2, LV1, SYNC	700 mV	810 mV
VOH (LVDS)		970 mV	1200 mV
VOL (CMOS)	CCK, CCKA, DAO, LD, LDA	0 mV	200 mV
VOH (CMOS)		1800 mV	2000 mV

1.2.7 Digital Functional and Scan Chain (test# 600÷608).

This test is aimed to the verify of the digital functionality of the MCC. To obtain high fault coverage it has been used ATPG (automatic test pattern generation). ATPG is used for the synthesized part of the chip (i.e. the whole chip, but the 16 Receiver FIFO's and the Delay Line blocks). ATPG test vectors are generated by Synopsys using the embedded scan chain. ATPG has a fault coverage of the 87% of the standard cell logics (see Appendix 4.1).

To test the ReceiverFIFO's, test vectors which do read/write operations of '0' and '1' to all the SRAM cells have been generated. (Delta-01, see Appendix 3.1).

Additional test vector files have been added covering the normal foreseen operation in the experiment, like R/W of registers, event building at different output bit rate or FE configuration.

Any error in the digital test is binned as *Bin#3*.

The digital functional test used for fault coverage are:

- Delta-01. (test#600, source code Appendix 3.1)
 Test of SoftReset, R/W and test of the PendingLv1FIFO, R/W of the ReceiverFifo's.
- Delta-02. (test#601, source code Appendix 3.2)
 Test WrReceiver, SYNC generations for all 16 pulse widths, R/W to FE (WrFE and RdFE).
- Delta03. (test#602, source code Appendix 3.3)
 Test of L1ID and BCID counters using generated events, test of contiguous LV1, test skipped LV1 (overflow of the PendingLv1FIFO), test event counter reset (ECR) command.
- Delta04. (test#350, source code Appendix 3.4)
 Simulation of digital injection scan. Repeat 12 times a loop with 16 contiguous LV1 generated by the MCC and 90 hits per FE generated in the 6th of the 16 events reply. This test is sensitive to VDD and it is used to define the VDD threshold above which the MCC works error free. Those test vectors are also used for VDD scan.
- Delta-05. (test#603, source code Appendix 3.5)
 Test of the disable of event building to read the hits from the ReceiverFIFO and Scoreboard.

- Delta-06. (test#608, source code Appendix 3.6)
 Test of all 4 output data rates: 40 x1, 40x2, 80x1, 80x2 Mb/s.
- Delta-07. (test#700÷799 source code Appendix 3.7).
 Delay measurement for Delay Line block.
- Delta-08. (test#604, source code Appendix 3.8)
 Test of the AMS compatibility pinout.
- Delta-09. (test#500 source code Appendix 3.9)
 These vectors are used for Vout tests (see Chapter 1.2.6).
- ScanChain-0, ScanChain-1, ScanChain3. (tests# 605, 606, 607, logfile Appendix 4.1).
 Scan chain vectors generated by ATPG.

Table 1-4 summarize the test vectors used for digital fault coverage and for analog measurements.

Table 1-4 Digital test vectors for fault test

Test Vector File	Test#	Simulation Time (ns)	Clock cycles
Delta_01.vec.gz	600	11 919 560	476 782
Delta_02.vec.gz	601	1 098 135	43 925
Delta_03_New.vec.gz	602	4 082 572	163 303
Delta_04.vec.gz	350÷360	8 399 997	336 000
Delta_05.vec.gz	603	1 352 510	54 100
Delta_06.vec.gz	608	3 718 085	148 723
Delta_07.vec.gz	700÷799	371 360	14 854
Delta_08.vec.gz	604	53 975	2 159
Delta_09.vec.gz	500÷551	32 860	1 314
ScanChain_0.vec.gz	605	56 740 600	567 406
ScanChain_1.vec.gz	606	56 740 600	567 406
ScanChain_2.vec.gz	607	56 392 055	563 921
Total		200 902 310	2 939 895

1.2.8 Delay line (test#700÷799).

Delay line measurements have been carried on for certain values of range (5) and delays (20) for a total of 100 measure points. Delay are measured from beginning of the CAL command and 15 clock cycles are included in the delay actually measured.

The values for the range and the delay are:

- range: 0, 1, 3, 7, 15.
- delay: 0, 1, 2, 3, 5, 7, 11, 15, 19, 23, 27, 31, 35, 39, 43, 47, 51, 55, 59, 63.

Because the absolute delay precision of the DelayLine block is relatively low important, the cut put in the Delay Line are rather broad: Lower limit = 375 ns, upper limit = 700 ns (apart from test#0 where upper limit is 400 ns). The decision of broad cuts was taken to avoid a possible large yield loss due to process variation from wafer to wafer. Post test analysis will be used to eventually reject chips with delay to far from specs.

1.3 Additional Documentation.

MCC-I2 WEB page:

http://www.ge.infn.it/ATLAS/Electronics/home_frames/MCC-I2_frame.html

2 Production Test Results

2.1 Wafer Yields.

The production test have been made on 562 chips per each of the 6 wafer for a total of 3372 components. The chips passing all the cuts are 2666 or 79.1%. The number needed for the 3 layers of the ATLAS detector modules is 1774. There are 50% of spare chips, that should be enough to count for production losses in the Pixel Detector assembly. The yield for the 6 production wafers are reported in the Table 2-1 and shown in graphical format in Figure 2-1.

Table 2-1 Summary yield table for the 6 production wafers.

Bin	Test	Wafer 01 VX50V7X	Wafer 02 VW50V8X	Wafer 03 VU50VAX	Wafer 04 VS50REX	Wafer 05 VR50VDX	Wafer 06 V650VYX
1	Pass	88.4%	86.1%	86.7%	90.4%	51.6%	74.4%
3	Functional	4.6%	2.8%	3.6%	2.7%	7.3%	9.4%
4	Delay line	0.5%	0.0%	0.9%	0.7%	0.2%	0.2%
8	VDD	2.5%	7.8%	4.8%	2.0%	32.4%	5.5%
9	VOUT	0.7%	0.0%	0.0%	0.4%	5.2%	0.9%
10	Continuity	2.1%	2.0%	2.1%	2.3%	2.7%	4.3%
11	Power	1.1%	1.2%	2.0%	1.6%	0.7%	5.3%

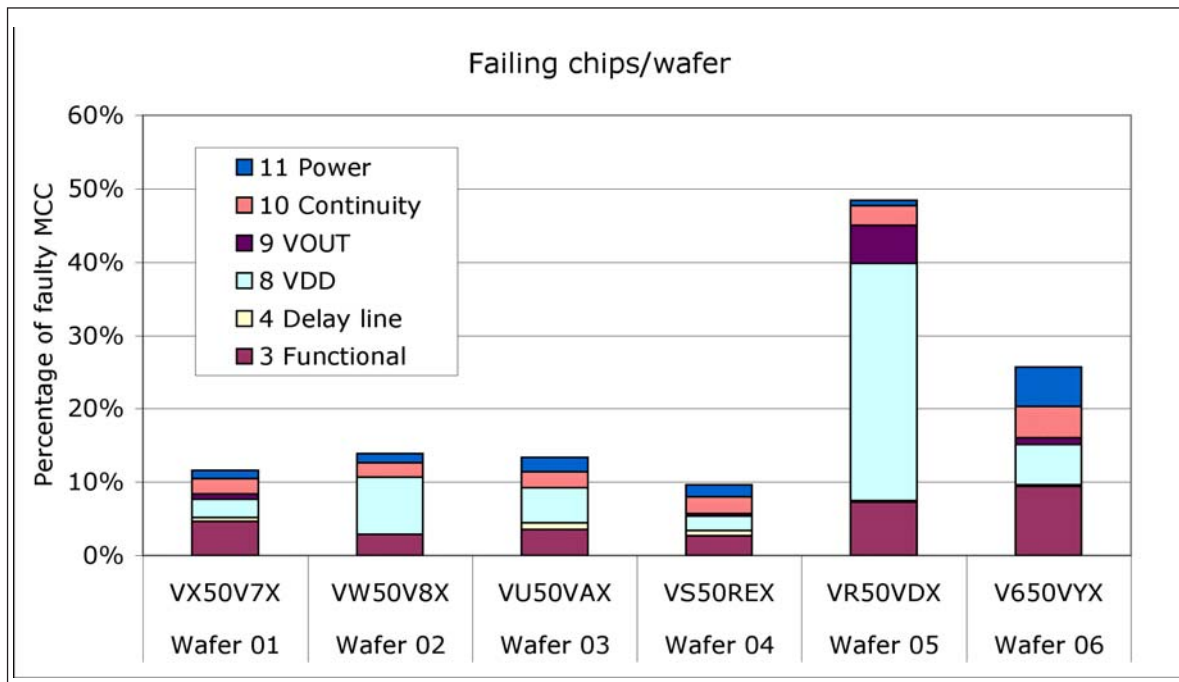


Figure 2-1 Rejected chips per wafer.

Four wafer show a very good yield (between 85÷90%), while other two have low yield. From analysis of the yield, wafer#5 (VR50VDX) has a lot of chips (32%) that works at VDD larger then the cut value (1.86 V), while wafer 6 (V650VYX) has just lower yield for most of the test types.

Nine files of test vectors have been used to improve digital fault coverage: test# 600÷608 (seeChapter 1.2.7). There is a large overlap between the fault found by different test vector files. Scan chain test (test# 605÷607) are particularly good for detecting faults in the standard cell design, while tests like #600 and #603 find faults in the ReciverFIFO's. Table 2-2 and Figure 2-2 represent the faults found by each test file for each of the 6 wafers. Note that the scan chain (that has the highest peak in the figure) does not find all the fault.

Table 2-2 Faulty chips found by the test vector files.

Wafer ID	#600 DLT-01	601 DLT-02	602 DLT-03	603 DLT-05	604 DLT-08	605 SCAN0	606 SCAN1	607 SCAN2	608 DLT-06
VX50V7X	34	31	36	32	32	41	41	41	36
VW50V8X	29	25	30	24	24	32	32	33	28
VU50VAX	31	29	36	28	29	34	34	34	35
VS50REX	27	26	30	25	24	32	33	34	28
VR50VDX	33	41	34	29	29	49	47	46	35
V650VYX	81	62	79	60	60	88	87	88	81

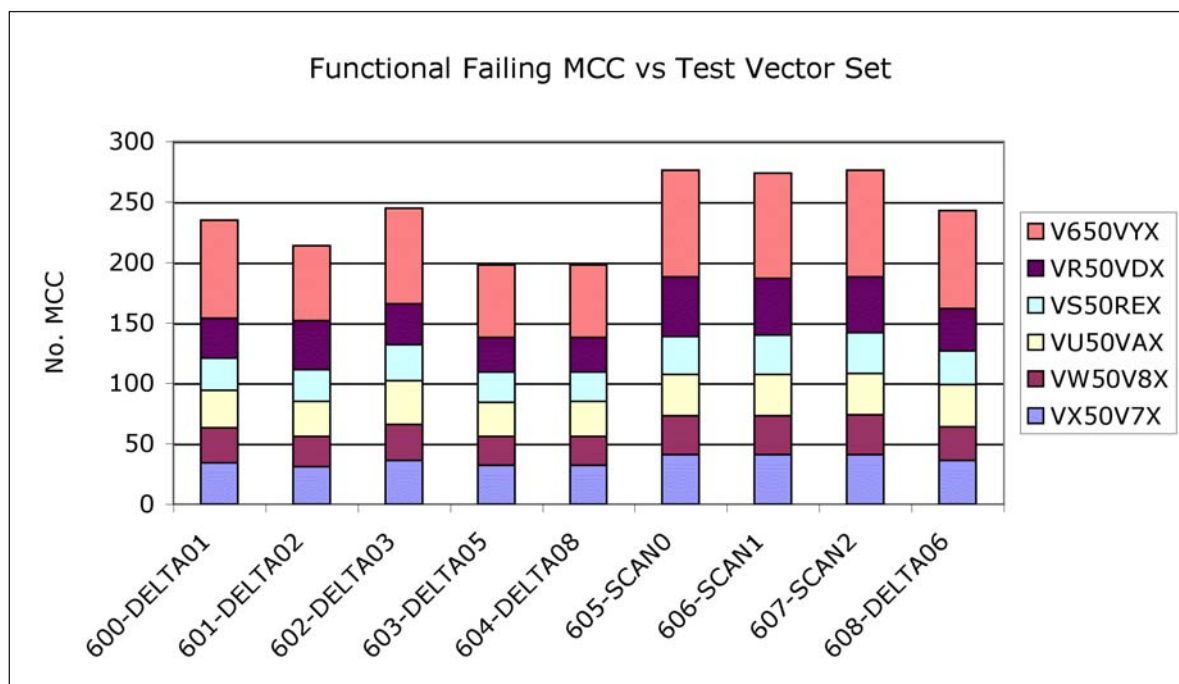


Figure 2-2 Histogram of digital test fails for the different vector files.

Wafer maps of faulty chips are shown in Figure 2-3. Most of the fault classes (bins) have an uniform distribution on wafer. "VDD" faults have a preference for the centre of the wafer.

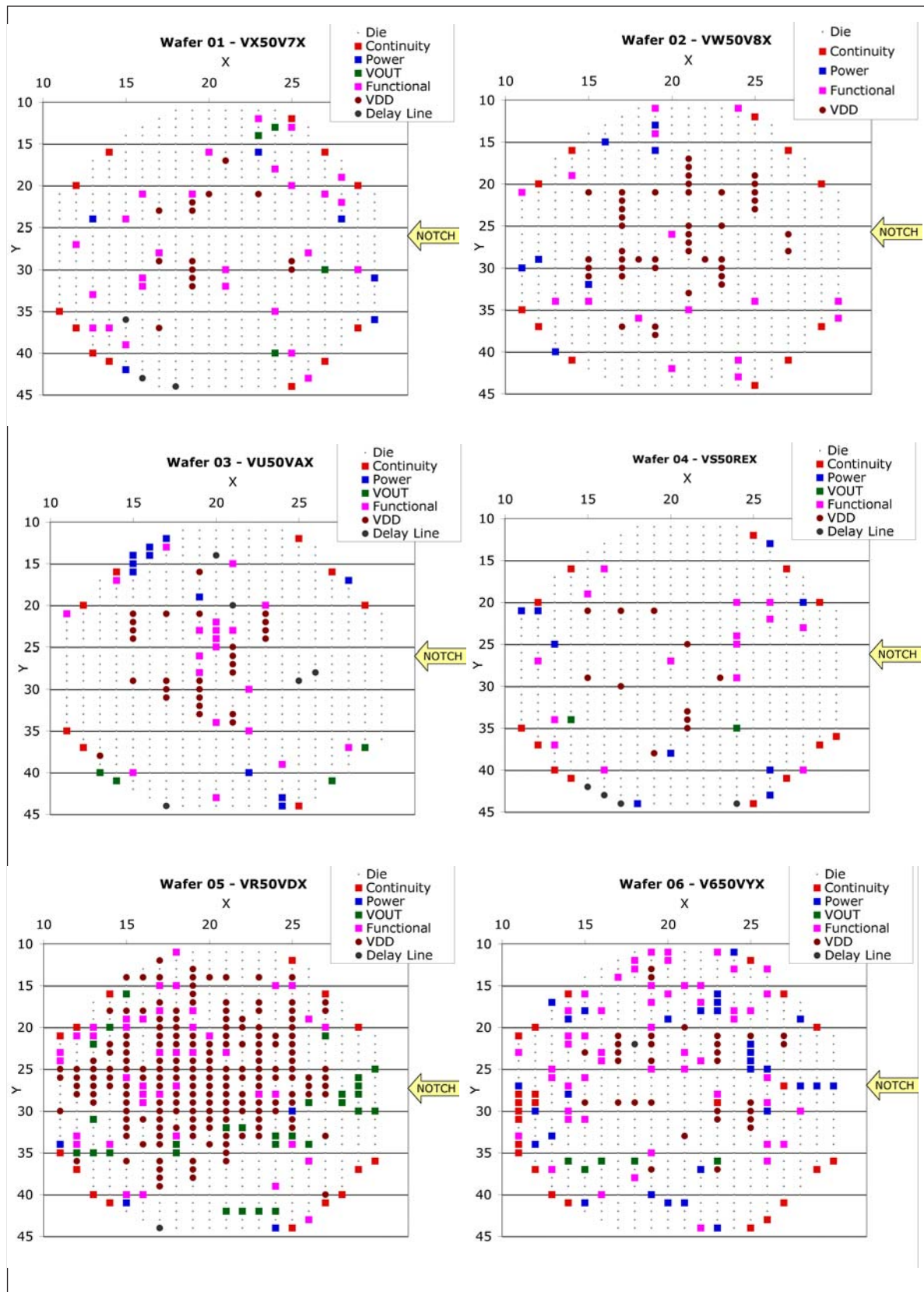


Figure 2-3 Wafer maps of the 6 production wafers.

2.2 Input leakage current.

The 4 CMOS inputs have either pull-up or pull-down resistance. The leakage current (IIH or IIL) detects the presence of the 100 kΩ resistance.

2.3 VDD test.

A critical aspect for the operation of the Pixel Detector module is the VDD supply. The nominal VDD voltage foreseen for ATLAS is 2.0 V. The design of the MCC-I2 have been done using logic synthesys on large extent. The timing library available for the standard cells was characterised for 2.5 V with the inclusion of degrading factors for best/ worst case. During testing of detector modules, built with MCC-I2.0, it was noticed that there are errors generated by the MCC in case of FE calibration with digital injection. Those errors are sensitive to VDD, temperature and possibly to accumulated radiation dose. A set of vectors (Delta-04, see Appendix 3.4) which reproduce such a situation, have been added as qualification test. The value of chip supply voltage (VDD) above which the chip works correctly is recorded in the logfile (see example: Appendix 5.1) and chips with such a value above 1.86 V are inked as bad die..

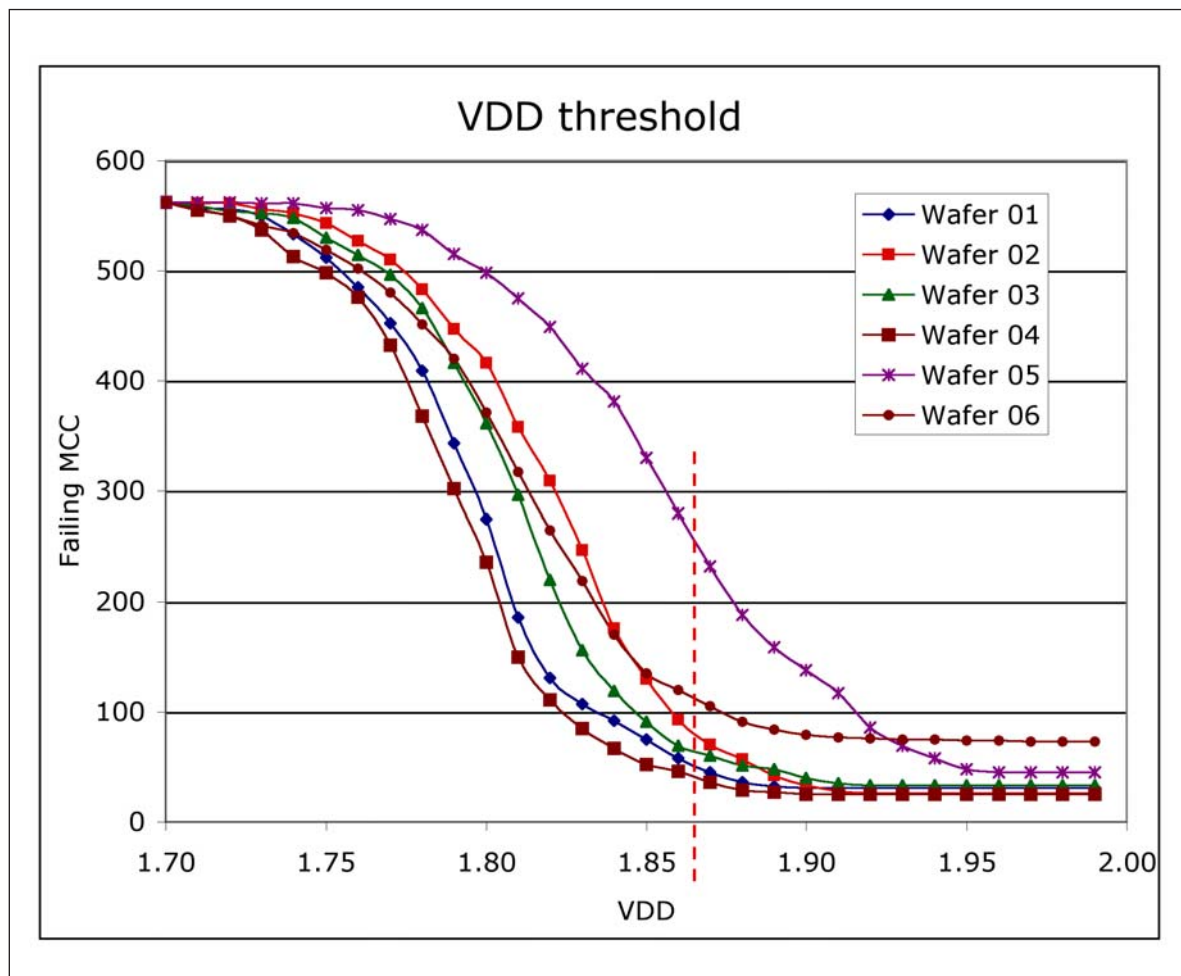


Figure 2-4 VDD scan for the 6 wafers. Curves represent the number of chip failing the test. Chip are accepted if passing the test at VDD supply voltage lower than 1.87 V.

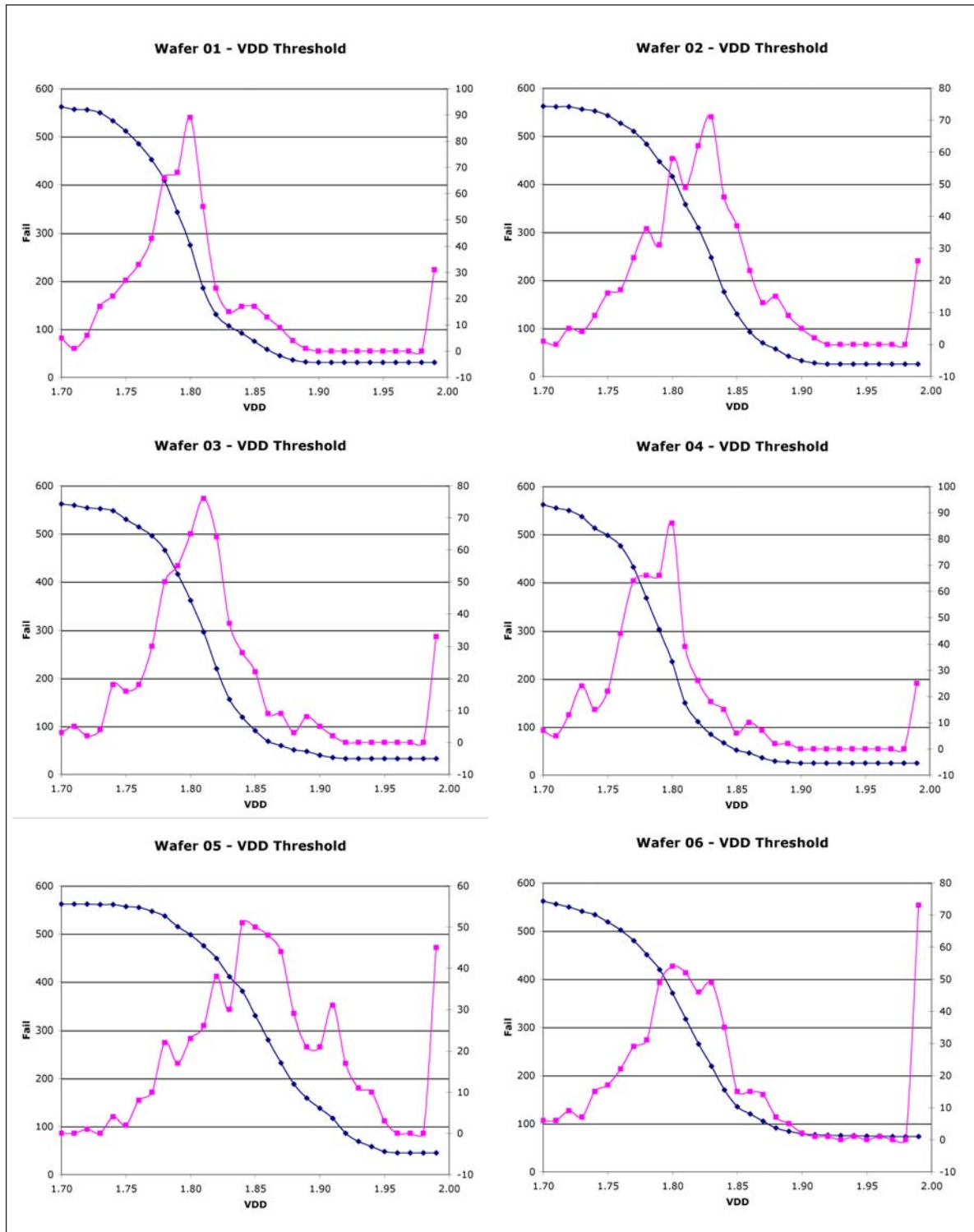


Figure 2-5 VDD threshold for the 6 production wafers. Blue is the cumulative result, while purple is the number of chips non passing the test from that value of VDD and below.

Figure 2-4. shows the number of chip in each of the 6 wafers not passing the test as function of VDD. Figure 2-5 shows both cumulative and differential distribution of the number of faulty chip as function of VDD.

Because each chip will keep his identity during the whole production of the detector, if we find that VDD threshold become critical after irradiation of the module (it has to be correlated its value before and after irradiation), we will put modules with lower threshold in the B_layer.

2.4 Vout test.

Output voltage is measured both for LVDS and CMOS output pins. For CMOS is requested a current sink of 4 mA either to GND (VOH) or to VDD (VOL). LVDS drivers are terminated to 100 Ω between positive and negative outputs.

Test limits are:

- CMOS VOL: > 0.0 V and < 200 mV;
- CMOS VOH: > 1.8 V and < 2.0 V;
- LVDS VOL: > 710 mV and < 810 mV;
- LVDS VOH: > 970 mV and < 1200 mV;

Figure 1-6 shows the measured values VOH-VOL, for the 562 chips on a wafer, of a LVDS out-put (DTop or DTON) as function of the VOL or VOH of that pin.

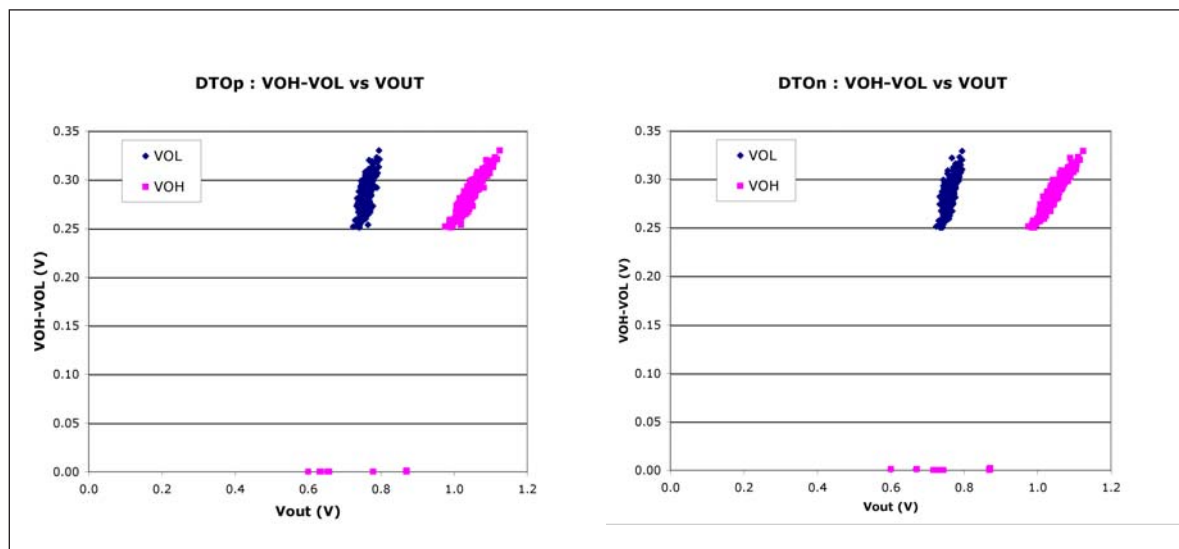


Figure 2-6 Correlation plot of VOH-VOL as function of VOH or VOL for two LVDS drivers (DTop on the right and DTON on the left).

2.5 Delay measurement for the Delay line.

The delay line has been characterised by 100 time's measurements. The data are for 5 of the 16 delay ranges and for each range 20 of the 64 delay steps have been measured. This test is mainly to characterize the component and not for rejecting chips on the base of timing limits.

The limits for the delays are: 375 ns for the lower and 700 ns for the upper. Only defective delay lines are responsible for chip rejection. Delay measurements will be used for chip sorting if it will be found necessary.

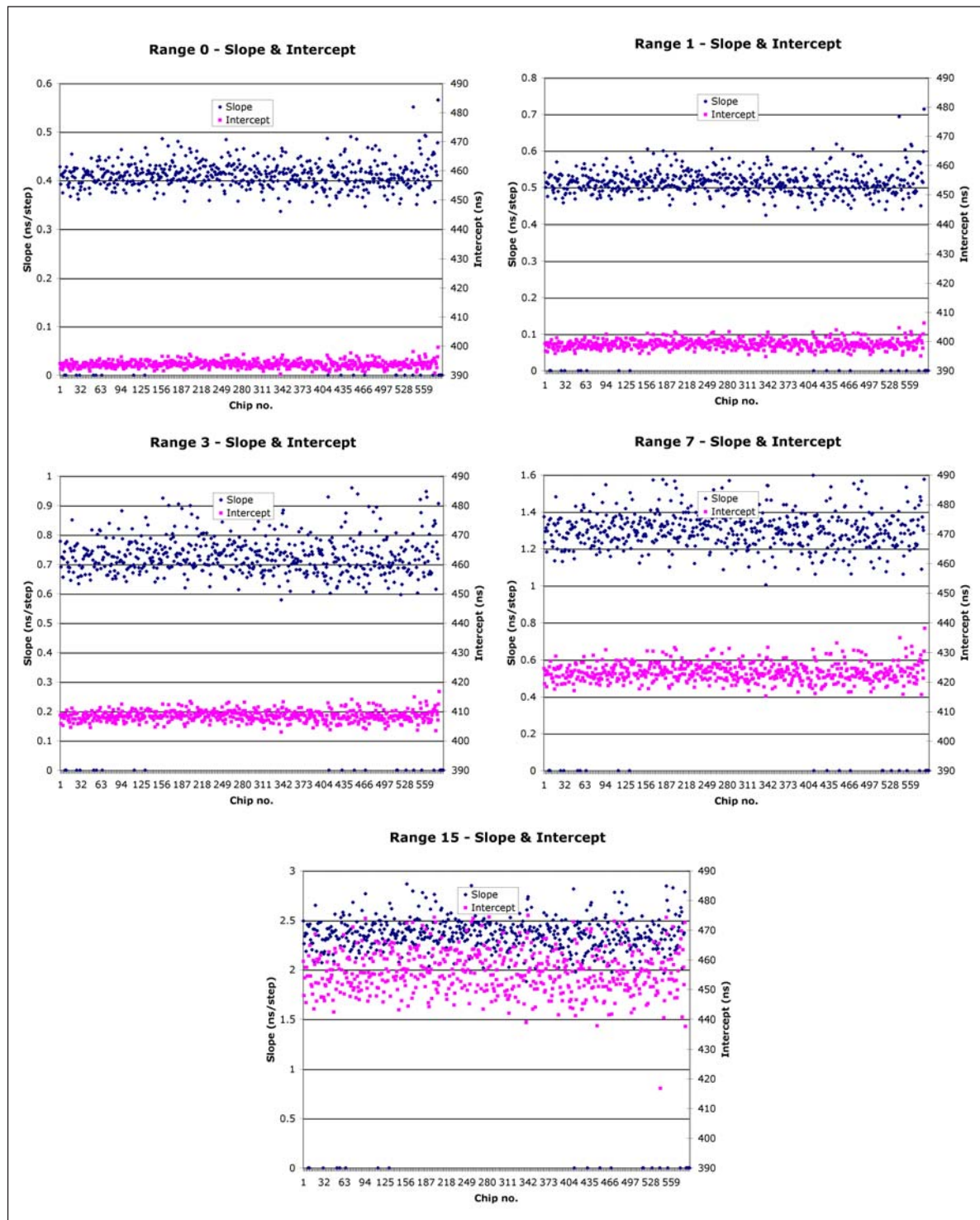


Figure 2-7 Delay line coefficients (Slope/Intercept) for 5 different values of range and 20 delay points per fit.

Figure 2-7 shows the slope and intercept of the linear fits from delays belonging to the same range. All chips from one wafer are in the plot. Defective delay lines usually have a zero slope.

Figure 2-8 shows the linearity plots for the first 200 chips of wafer VX50V7X. The two plots correspond to two delay range settings.

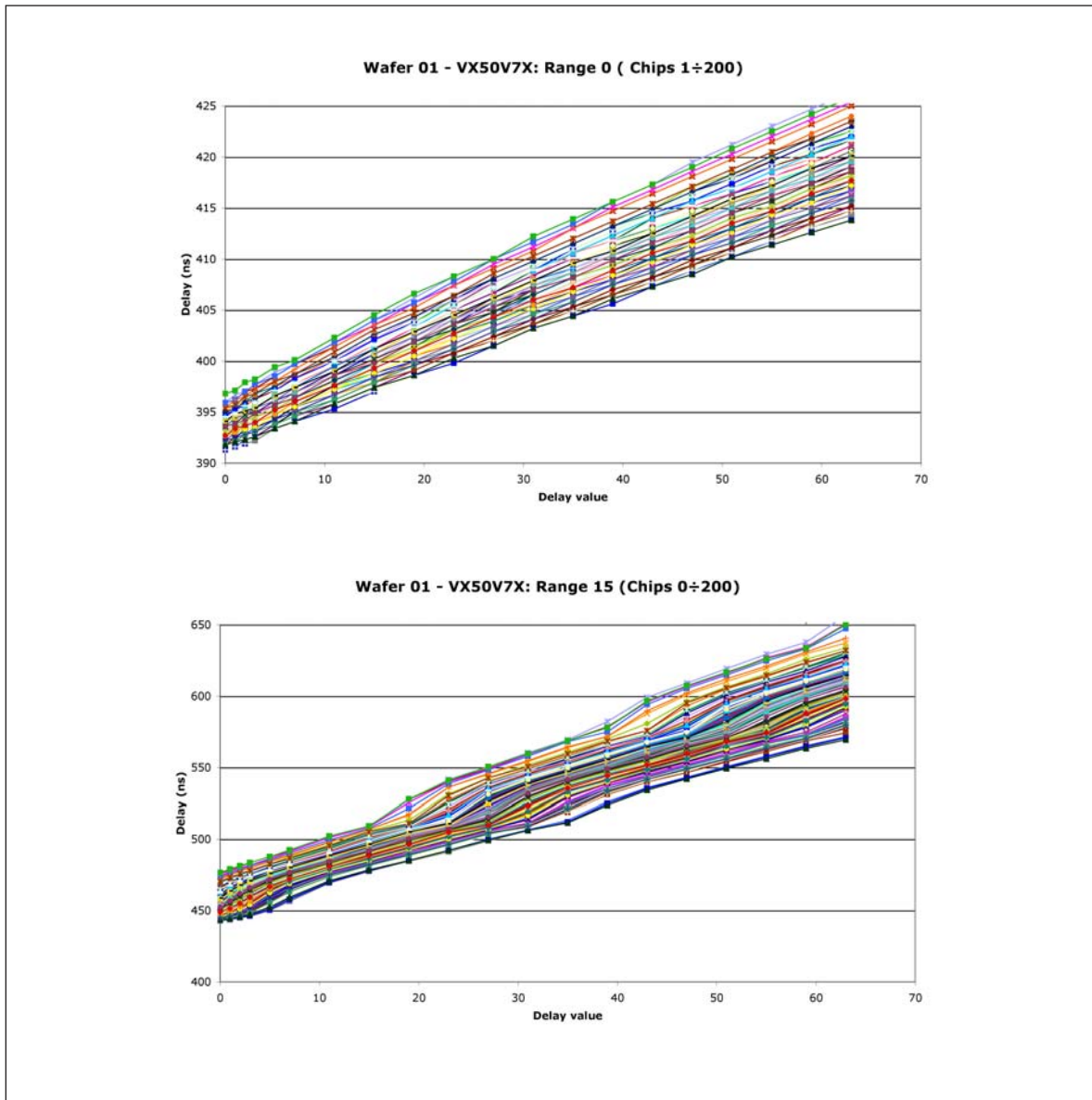


Figure 2-8 Delay Line delays for the first 200 chips of wafer VX50V7X. Delays for range 0 (above) and for range 15 (below).

3 Appendix: Test Vectors - SimPix Source

3.1 Delta-01.spx

```
//  
// TEST #01:  
// Read/Write Registers and FIFOs + PendingEventFIFO  
//  
////////////////////////////////////  
// Variables definitions  
////////////////////////////////////  
  
RdR = 10 // Delay between RdRegister commands  
WrR = 10 // Delay between WrRegister commands  
RdF = 10 // Delay between RdFIFO commands  
WrF = 10 // Delay between WrFIFO commands  
nLV1 = 8 // Half PendingEventFIFO depth  
nRec = 64 // Half ReceiverFIFO depth  
TRAIL 100  
  
////////////////////////////////////  
// SoftReset command Test #1  
////////////////////////////////////  
100 SOFT_RESET  
1030 RD_REGISTER CSR // All zeroes expected  
$RdR RD_REGISTER LV1 // All zeroes expected  
$RdR RD_REGISTER FEEN // All zeroes expected  
$RdR RD_REGISTER WFE // All zeroes expected  
$RdR RD_REGISTER WMCC // All zeroes expected  
$RdR RD_REGISTER CNT // All zeroes expected  
$RdR RD_REGISTER CAL // All zeroes expected  
$RdR RD_REGISTER WBTD // All zeroes expected  
$RdR RD_REGISTER WRECD // All zeroes expected  
  
100 SOFT_RESET  
  
////////////////////////////////////  
// Registers Test  
////////////////////////////////////  
30 WR_REGISTER CSR 5555  
$RdR RD_REGISTER CSR  
$RdR RD_REGISTER CSR  
$WrR WR_REGISTER CSR AA9A // OPAT bit set to zero  
$RdR RD_REGISTER CSR  
$RdR RD_REGISTER CSR  
  
$WrR WR_REGISTER LV1 5555  
$RdR RD_REGISTER LV1  
$RdR RD_REGISTER LV1  
$WrR WR_REGISTER LV1 AAAA  
$RdR RD_REGISTER LV1  
$RdR RD_REGISTER LV1  
  
$WrR WR_REGISTER FEEN 5555  
$RdR RD_REGISTER FEEN  
$RdR RD_REGISTER FEEN  
$WrR WR_REGISTER FEEN AAAA  
$RdR RD_REGISTER FEEN  
$RdR RD_REGISTER FEEN  
  
$WrR WR_REGISTER WFE 5555
```

```
$RdR    RD_REGISTER WFE
$RdR    RD_REGISTER WFE
$WrR    WR_REGISTER WFE      AAAA
$RdR    RD_REGISTER WFE
$RdR    RD_REGISTER WFE

$WrR    WR_REGISTER WMCC    5555
$RdR    RD_REGISTER WMCC
$RdR    RD_REGISTER WMCC
$WrR    WR_REGISTER WMCC    AAAA
$RdR    RD_REGISTER WMCC
$RdR    RD_REGISTER WMCC

$WrR    WR_REGISTER CNT     5555
$RdR    RD_REGISTER CNT
$RdR    RD_REGISTER CNT
$WrR    WR_REGISTER CNT     AAAA
$RdR    RD_REGISTER CNT
$RdR    RD_REGISTER CNT

$WrR    WR_REGISTER CAL     5555
$RdR    RD_REGISTER CAL
$RdR    RD_REGISTER CAL
$WrR    WR_REGISTER CAL     AAAA
$RdR    RD_REGISTER CAL
$RdR    RD_REGISTER CAL

$WrR    WR_REGISTER WBITD   5555
$RdR    RD_REGISTER WBITD
$RdR    RD_REGISTER WBITD
$WrR    WR_REGISTER WBITD   AAAA
$RdR    RD_REGISTER WBITD
$RdR    RD_REGISTER WBITD

$WrR    WR_REGISTER WRECD   5555
$RdR    RD_REGISTER WRECD
$RdR    RD_REGISTER WRECD
$WrR    WR_REGISTER WRECD   AAAA
$RdR    RD_REGISTER WRECD
$RdR    RD_REGISTER WRECD
```

```
////////////////////////////////////
// PendingEventFIFO Test
////////////////////////////////////
// Write Loop #1
LOOP Location 1 $nLV1 1
    $WrR    WR_REGISTER PEF 5555
    $WrR    WR_REGISTER PEF AAAA
END_LOOP

// Read Loop #1
LOOP Location 1 $nLV1 1
    $RdR    RD_REGISTER PEF // 5555 expected
    $RdR    RD_REGISTER PEF // AAAA expected
END_LOOP
LOOP Location 1 $nLV1 1
    $RdR    RD_REGISTER PEF // AAAA expected
    $RdR    RD_REGISTER PEF // AAAA expected
END_LOOP

// Write Loop #2
LOOP Location 1 $nLV1 1
    $WrR    WR_REGISTER PEF AAAA
    $WrR    WR_REGISTER PEF 5555
END_LOOP
```

```
// Read Loop #2
LOOP Location 1 $nLV1 1
  $RdR   RD_REGISTER PEF // AAAA expected
  $RdR   RD_REGISTER PEF // 5555 expected
END_LOOP
LOOP Location 1 $nLV1 1
  $RdR   RD_REGISTER PEF // 5555 expected
  $RdR   RD_REGISTER PEF // 5555 expected
END_LOOP
$RdR   RD_REGISTER PEF // 5555 expected

//
// END-TEST #01:
// Read/Write Registers and FIFOs
//

//
// TEST #01:
// Read/Write Registers and FIFOs
//
// Test PendingEventFifo
//

100 SOFT_RESET

////////////////////////////////////
// PendingEventFIFO Test #1
////////////////////////////////////
//
// Write #1
//
$WrR   WR_REGISTER PEF 0555
$WrR   WR_REGISTER PEF 1AAA
$WrR   WR_REGISTER PEF 2555
$WrR   WR_REGISTER PEF 3AAA
$WrR   WR_REGISTER PEF 4555
$WrR   WR_REGISTER PEF 5AAA
$WrR   WR_REGISTER PEF 6555
$WrR   WR_REGISTER PEF 7AAA
$WrR   WR_REGISTER PEF 8555
$WrR   WR_REGISTER PEF 9AAA
$WrR   WR_REGISTER PEF A555
$WrR   WR_REGISTER PEF BAAA
$WrR   WR_REGISTER PEF C555
$WrR   WR_REGISTER PEF DAAA
$WrR   WR_REGISTER PEF E555
$WrR   WR_REGISTER PEF FAAA
//
// Read Loop #1 (16 commands)
//
100 RD_REGISTER PEF // 0555
$RdR   RD_REGISTER PEF // 1AAA
$RdR   RD_REGISTER PEF // 2555
$RdR   RD_REGISTER PEF // 3AAA
$RdR   RD_REGISTER PEF // 4555
$RdR   RD_REGISTER PEF // 5AAA
$RdR   RD_REGISTER PEF // 6555
$RdR   RD_REGISTER PEF // 7AAA
$RdR   RD_REGISTER PEF // 8555
$RdR   RD_REGISTER PEF // 9AAA
$RdR   RD_REGISTER PEF // A555
$RdR   RD_REGISTER PEF // BAAA
$RdR   RD_REGISTER PEF // C555
$RdR   RD_REGISTER PEF // DAAA
```

```
$RdR    RD_REGISTER PEF // E555
$RdR    RD_REGISTER PEF // FAAA
//
// Read Loop #2 (16 commands)
//
// Now the read pointer points to the next available location
// which is Address 0 and the read address is not incremented anymore.
//
100 RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
$RdR    RD_REGISTER PEF 0x0555 // Value of address 0
//
// Write #2 (opposite values)
//
100 WR_REGISTER PEF FAAA
$WrR    WR_REGISTER PEF E555
$WrR    WR_REGISTER PEF DAAA
$WrR    WR_REGISTER PEF C555
$WrR    WR_REGISTER PEF BAAA
$WrR    WR_REGISTER PEF A555
$WrR    WR_REGISTER PEF 9AAA
$WrR    WR_REGISTER PEF 8555
$WrR    WR_REGISTER PEF 7AAA
$WrR    WR_REGISTER PEF 6555
$WrR    WR_REGISTER PEF 5AAA
$WrR    WR_REGISTER PEF 4555
$WrR    WR_REGISTER PEF 3AAA
$WrR    WR_REGISTER PEF 2555
$WrR    WR_REGISTER PEF 1AAA
$WrR    WR_REGISTER PEF 0555
//
// Read Loop #1 (16 commands)
//
100 RD_REGISTER PEF // FAAA
$RdR    RD_REGISTER PEF // E555
$RdR    RD_REGISTER PEF // DAAA
$RdR    RD_REGISTER PEF // C555
$RdR    RD_REGISTER PEF // BAAA
$RdR    RD_REGISTER PEF // A555
$RdR    RD_REGISTER PEF // 9AAA
$RdR    RD_REGISTER PEF // 8555
$RdR    RD_REGISTER PEF // 7AAA
$RdR    RD_REGISTER PEF // 6555
$RdR    RD_REGISTER PEF // 5AAA
$RdR    RD_REGISTER PEF // 4555
$RdR    RD_REGISTER PEF // 3AAA
$RdR    RD_REGISTER PEF // 2555
$RdR    RD_REGISTER PEF // 1AAA
$RdR    RD_REGISTER PEF // 0555
//
// Read Loop #2 (16 commands)
//
//
```



```
$RdR    RD_REGISTER PEF // 7AA1 expected (Address 7)
$RdR    RD_REGISTER PEF // 8551 expected (Address 8)
$RdR    RD_REGISTER PEF // 9AA1 expected (Address 9)
$RdR    RD_REGISTER PEF // A551 expected (Address A)
$RdR    RD_REGISTER PEF // BAA1 expected (Address B)
$RdR    RD_REGISTER PEF // C551 expected (Address C)
$RdR    RD_REGISTER PEF // DAA1 expected (Address D)
$RdR    RD_REGISTER PEF // E551 expected (Address E)
$RdR    RD_REGISTER PEF // FAA1 expected (Address F)
$RdR    RD_REGISTER PEF // 0551 expected (Address 0)
// Now the read pointer points to the next available location
// which is Address 0 and the read address is not incremented anymore.
$RdR    RD_REGISTER PEF 0x0551 // (Address 0)
$RdR    RD_REGISTER PEF 0x0551 // (Address 0)
$RdR    RD_REGISTER PEF 0x0551 // (Address 0)
$RdR    RD_REGISTER PEF 0x0551 // (Address 0)
//
//
// PEF Test #3 (Write more times than Read)
//
100 SOFT_RESET
//
// Write 16 locations
//
$WrR    WR_REGISTER PEF 0001
$WrR    WR_REGISTER PEF 1001
$WrR    WR_REGISTER PEF 2001
$WrR    WR_REGISTER PEF 3001
$WrR    WR_REGISTER PEF 4001
$WrR    WR_REGISTER PEF 5001
$WrR    WR_REGISTER PEF 6001
$WrR    WR_REGISTER PEF 7001
$WrR    WR_REGISTER PEF 8001
$WrR    WR_REGISTER PEF 9001
$WrR    WR_REGISTER PEF A001
$WrR    WR_REGISTER PEF B001
$WrR    WR_REGISTER PEF C001
$WrR    WR_REGISTER PEF D001
$WrR    WR_REGISTER PEF E001
$WrR    WR_REGISTER PEF F001
//
// 4 additional write commands are ignored
//
100 WR_REGISTER PEF 0FF1 // Dropped
$WrR    WR_REGISTER PEF 1FF1 // Dropped
$WrR    WR_REGISTER PEF 2FF1 // Dropped
$WrR    WR_REGISTER PEF 3FF1 // Dropped
//
// Read Loop #1 (16 commands)
//
100 RD_REGISTER PEF // 0001 expected
$RdR    RD_REGISTER PEF // 1001 expected
$RdR    RD_REGISTER PEF // 2001 expected
$RdR    RD_REGISTER PEF // 3001 expected
$RdR    RD_REGISTER PEF // 4001 expected
$RdR    RD_REGISTER PEF // 5001 expected
$RdR    RD_REGISTER PEF // 6001 expected
$RdR    RD_REGISTER PEF // 7001 expected
$RdR    RD_REGISTER PEF // 8001 expected
$RdR    RD_REGISTER PEF // 9001 expected
$RdR    RD_REGISTER PEF // A001 expected
$RdR    RD_REGISTER PEF // B001 expected
$RdR    RD_REGISTER PEF // C001 expected
$RdR    RD_REGISTER PEF // D001 expected
$RdR    RD_REGISTER PEF // E001 expected
$RdR    RD_REGISTER PEF // F001 expected
```



```
        END_LOOP
END_LOOP

// Write Loop #2
LOOP Location 1 $nRec 1
    $WrF    WR_FIFO 3x2AAAAAA
    $WrF    WR_FIFO 3x5555555
END_LOOP

LOOP i 0 9
    // Read Loop #2
    LOOP Location 1 $nRec 1
        $RdF    RD_FIFO FIFO0$i // AAAAA expected
        $RdF    RD_FIFO FIFO0$i // 155555 expected
    END_LOOP
    LOOP Location 1 $nRec 1
        $RdF    RD_FIFO FIFO0$i // AAAAA expected
        $RdF    RD_FIFO FIFO0$i // 155555 expected
    END_LOOP
    $RdF    RD_FIFO FIFO0$i      // AAAAA expected
END_LOOP
LOOP i 10 15
    // Read Loop #2
    LOOP Location 1 $nRec 1
        $RdF    RD_FIFO FIFO$i  // AAAAA expected
        $RdF    RD_FIFO FIFO$i  // 155555 expected
    END_LOOP
    LOOP Location 1 $nRec 1
        $RdF    RD_FIFO FIFO$i  // AAAAA expected
        $RdF    RD_FIFO FIFO$i  // 155555 expected
    END_LOOP
    $RdF    RD_FIFO FIFO$i      // AAAAA expected
END_LOOP

////////////////////////////////////
// SoftReset command Test #2
////////////////////////////////////
100 SOFT_RESET
30 RD_REGISTER CSR      // All zeroes expected
$RdR  RD_REGISTER LV1   // All zeroes expected
$RdR  RD_REGISTER FEEN  // All zeroes expected
$RdR  RD_REGISTER WFE   // All zeroes expected
$RdR  RD_REGISTER WMCC  // All zeroes expected
$RdR  RD_REGISTER CNT   // All zeroes expected
$RdR  RD_REGISTER CAL   // All zeroes expected
$RdR  RD_REGISTER WBITD // All zeroes expected
$RdR  RD_REGISTER WRECD // All zeroes expected
$RdR  RD_REGISTER PEF   // 5555 expected
LOOP i 0 9
    $RdF    RD_FIFO FIFO0$i // 155555 expected
END_LOOP
LOOP i 10 15
    $RdF    RD_FIFO FIFO$i  // 155555 expected
END_LOOP

100 SOFT_RESET
```

3.2 Delta-02.spx

```
// TEST #05-A:
// WriteReceiver test
//

////////////////////////////////////
// Variables definitions
////////////////////////////////////
    Tr1 = 4000 // Delay between Trigger commands
    N1 = 2     // Number of events
// Tr2 = 200  // Delay between Trigger commands
// N2 = 22    // Number of events
// Tr3 = 100  // Delay between Trigger commands
// N3 = 21    // Number of events
////////////////////////////////////
// Test #1 - 40 Mbit/s test
////////////////////////////////////
100 SOFT_RESET
30 WR_REGISTER CSR 005c // 40 Mbit/s - Playback mode
30 WR_REGISTER FEEN ffff // All FE enabled
30 WR_REGISTER LV1 0000 // No contiguous LEVEL1
//
LOOP i 1 $N1
$Tr1 DELAY
//20 RD_REGISTER CSR
100 WR_RECEIVER
    HITS_EOE fff7 1 0
    HITS_EOE 0008 2 0
    END_WR_RECEIVER
20 RD_REGISTER CSR
20 RD_REGISTER FEEN
200 WR_REGISTER FEEN FFFF
1000 ENABLE_DATATAKE
100 TRIGGER
END_LOOP
//
// END-TEST #05-A:
// WriteReceiver test
//
// -----
//
// TEST #02:
// SYNC Generation, WrFrontEnd/RdFrontEnd
// Delay of DTI nominal is 19.

    active_fe = ffff
    trigd = 7

    FE_0 = 0
    FE_1 = 1
    FE_2 = 2
    FE_3 = 3
    FE_4 = 4
    FE_5 = 5
    FE_6 = 6
    FE_7 = 7
    FE_8 = 8
    FE_9 = 9
    FE_10 = A
    FE_11 = B
    FE_12 = C
    FE_13 = D
    FE_14 = E
    FE_15 = F
```

```
TRAIL 100

4000 GLOBAL_RESET_MCC

30 WR_REGISTER FEEN $active_fe
30 WR_REGISTER CSR 1c
30 WR_REGISTER LV1 0

40 GLOBAL_RESET_FE 0x0
40 GLOBAL_RESET_FE 0x1
40 GLOBAL_RESET_FE 0x2
40 GLOBAL_RESET_FE 0x3
40 GLOBAL_RESET_FE 0x4
40 GLOBAL_RESET_FE 0x5
40 GLOBAL_RESET_FE 0x6
40 GLOBAL_RESET_FE 0x7
40 GLOBAL_RESET_FE 0x8
40 GLOBAL_RESET_FE 0x9
40 GLOBAL_RESET_FE 0xa
40 GLOBAL_RESET_FE 0xb
40 GLOBAL_RESET_FE 0xc
40 GLOBAL_RESET_FE 0xd
40 GLOBAL_RESET_FE 0xe
40 GLOBAL_RESET_FE 0xf

// Some WrFrontEnd and RdFrontEnd
100 WR_REGISTER CNT 0 // 0-CCK, LD, 0-CCK
100 WR_FRONTEND 1 0xa
100 RD_FRONTEND 1 0x5

200 WR_REGISTER CNT 9 // 8-CCK, LD, 1-CCK
200 WR_FRONTEND 2 0x858
200 RD_FRONTEND 2 0x858

200 WR_REGISTER CNT 9
200 WR_FRONTEND 2 0xef8
200 RD_FRONTEND 2 0xef8

200 WR_REGISTER CNT 02
200 WR_FRONTEND 4 0x855
200 RD_FRONTEND 4 0x855
//200 WR_FRONTEND 4 0x85518aaaaa1
//200 RD_FRONTEND 4 0x85518aaaaa1

200 WR_FRONTEND 1 CLOCK_REG 0x123456789
200 RD_FRONTEND 1 CLOCK_REG 0x123456789

// RdFrontRnd from all DTI's
30 WR_REGISTER CNT 41
LOOP i 0 15
    200 WR_FRONTEND $FE_@i 0x5aa5
    200 RD_FRONTEND $FE_@i 0xa55a
END_LOOP

30 WR_REGISTER FEEN 0x0000 // Disable all DTI's
30 WR_REGISTER CNT 41
LOOP i 0 15
    200 WR_FRONTEND $FE_@i 0x5aa5
    200 RD_FRONTEND $FE_@i 0xa55a
END_LOOP

200 GLOBAL_RESET_MCC
//
// End of Test
//
```

3.3 Delta-03.spx

```
//  
// TEST #08-A:  
// L1ID and BCID counters test  
// Consecutive LVL1 test  
// EventCounterReset command test  
//  
////////////////////////////////////  
// Variables definitions  
////////////////////////////////////  
    ATr1 = 300 // Delay before Trigger commands  
    ATr2 = 1000 // Delay between consecutive Trigger commands  
    ATr3 = 7000 // Delay between consecutive Trigger commands  
  
// TRAIL 6000  
////////////////////////////////////  
// MCC initialisation  
////////////////////////////////////  
100 SOFT_RESET  
//30 WR_REGISTER CSR 0013 // 160 Mbit/s  
//30 WR_REGISTER CSR 0012 // 80 Mbit/s double line  
//30 WR_REGISTER CSR 0011 // 80 Mbit/s single line  
30 WR_REGISTER CSR 0010 // 40 Mbit/s  
30 WR_REGISTER FEEN ffff // All FE enabled  
////////////////////////////////////  
// L1ID and BCID counters test #1 - NO contiguous LEVEL1  
////////////////////////////////////  
30 WR_REGISTER LV1 0000 // No contiguous LEVEL1  
10 ENABLE_DATATAKE  
LOOP i 1 24  
    $ATr1 TRIGGER  
    20 EVENT ffff 1 1  
END_LOOP  
3000 ECR  
LOOP i 1 16  
    $ATr1 TRIGGER  
    20 EVENT ffff 1 1  
END_LOOP  
////////////////////////////////////  
// L1ID and BCID counters test #2 - 2 contiguous LEVEL1  
////////////////////////////////////  
4000 WR_REGISTER LV1 0100 // 2 contiguous LEVEL1  
10 ENABLE_DATATAKE  
LOOP i 1 12  
    $ATr2 TRIGGER  
    EVENT  
    LOOP j 20 70 50  
        $j HITS_EOE ffff 1 1  
    END_LOOP  
    END_EVENT  
END_LOOP  
3000 ECR  
LOOP i 1 8  
    $ATr2 TRIGGER  
    EVENT  
    LOOP j 20 70 50  
        $j HITS_EOE ffff 1 1  
    END_LOOP  
    END_EVENT  
END_LOOP  
////////////////////////////////////  
// L1ID and BCID counters test #3 - 14 contiguous LEVEL1  
////////////////////////////////////  
3000 WR_REGISTER LV1 0d00 // 14 contiguous LEVEL1  
10 ENABLE_DATATAKE
```

```
LOOP i 1 3
  $ATr3 TRIGGER
  EVENT
  LOOP j 50 700 50
    $j HITS_EOE ffff 1 1
  END_LOOP
  END_EVENT
END_LOOP
////////////////////////////////////
// L1ID and BCID counters test #4 - 15 contiguous LEVEL1
////////////////////////////////////
10000 WR_REGISTER LV1 0e00 // 15 contiguous LEVEL1
10 ENABLE_DATATAKE
LOOP i 1 3
  $ATr3 TRIGGER
  EVENT
  LOOP j 50 750 50
    $j HITS_EOE ffff 1 1
  END_LOOP
  END_EVENT
END_LOOP
////////////////////////////////////
// L1ID and BCID counters test #5 - 16 contiguous LEVEL1
////////////////////////////////////
10000 WR_REGISTER LV1 0f00 // 16 contiguous LEVEL1
10 ENABLE_DATATAKE
LOOP i 1 3
  $ATr3 TRIGGER
  EVENT
  LOOP j 50 800 50
    $j HITS_EOE ffff 1 1
  END_LOOP
  END_EVENT
END_LOOP
//
// END-TEST #08-A:
// L1ID and BCID counters test
// Consecutive LVL1 test
// EventCounterReset command test
//
// -----
//
// TEST #08-C:
// Close Trigger commands test
//
// TRAIL 1000
// Tr = 10
// RdR = 20
////////////////////////////////////
// MCC initialisation
////////////////////////////////////
6000 SOFT_RESET
//30 WR_REGISTER CSR 0013 // 160 Mbit/s
30 WR_REGISTER CSR 0012 // 80 Mbit/s double line
//30 WR_REGISTER CSR 0011 // 80 Mbit/s single line
//30 WR_REGISTER CSR 0010 // 40 Mbit/s
30 WR_REGISTER FEEN FFFF // All FE enabled
////////////////////////////////////
// Close Trigger commands
////////////////////////////////////
30 WR_REGISTER LV1 00 // No contiguous LEVEL1
10 ENABLE_DATATAKE // Select Run mode
// 16 Trigger and 16 EmptyEvents - 1 clock cycles distance
LOOP i 100 8000 500
  1 TRIGGER
  $i EVENT ffff 1 1
```

```
END_LOOP
// 16 Trigger and 16 EmptyEvents - 0 clock cycles distance
10000 TRIGGER
100 EVENT ffff 1 1
LOOP i 600 8000 500
    0 TRIGGER
    $i EVENT ffff 1 1
END_LOOP
//
// END-TEST #08-C:
// Close Trigger commands test
//
// -----
//
// TEST #08-D:
// Skipped LEVEL1 test
//
// Variables definitions
// Delay before Trigger commands
DTr = 10
// TRAIL 100
// MCC initialisation
//
1000 SOFT_RESET
30 WR_REGISTER CSR 0013 // 160 Mbit/s
//30 WR_REGISTER CSR 0012 // 80 Mbit/s double line
//30 WR_REGISTER CSR 0011 // 80 Mbit/s single line
//30 WR_REGISTER CSR 0010 // 40 Mbit/s
30 WR_REGISTER FEEN FF // All FE enabled
// PEF overflow (4 SkippedLVL1)
30 WR_REGISTER LV1 00 // No contiguous LEVEL1
10 ENABLE_DATATAKE // Select Run mode

LOOP i 400 1900 100 // 16 Trigger and 16 EmptyEvents
    $DTr TRIGGER
    $i EVENT ffff 0 0
END_LOOP
//
LOOP i 1 4 // 4 Trigger (skipped)
    $DTr TRIGGER
        EVENT
        END_EVENT
END_LOOP
// 16th PEF location test
//
200 TRIGGER
1750 EVENT ffff 0 0
110 TRIGGER
1750 EVENT ffff 0 0
110 TRIGGER
1750 EVENT ffff 0 0
// 15th PEF location test
//
220 TRIGGER
1640 EVENT ffff 0 0
110 TRIGGER
1640 EVENT ffff 0 0
110 TRIGGER
1640 EVENT ffff 0 0
//
```

```
// PEF overflow (1 SkippedLVL1)
///////////////////////////////////////////////////////////////////
10 TRIGGER
1740 EVENT ffff 0 0
10 TRIGGER // Skipped
    EVENT
    END_EVENT
80 TRIGGER
1760 EVENT ffff 0 0
10 TRIGGER // Skipped
    EVENT
    END_EVENT
90 TRIGGER
1770 EVENT ffff 0 0
///////////////////////////////////////////////////////////////////
// SkippedLVL1 counter overflow
///////////////////////////////////////////////////////////////////
2000 TRIGGER // 1 Trigger and 1 EmptyEvent
600 EVENT ffff 0 0

LOOP i 650 1350 50 // 15 Trigger and 15 EmptyEvents
    $DTr TRIGGER
    $i EVENT ffff 0 0
END_LOOP

LOOP i 1 18 // 18 Trigger (skipped) -> 15 skipped expected
    $DTr TRIGGER
        EVENT
        END_EVENT
END_LOOP
170 TRIGGER
970 EVENT ffff 0 0
//
// END-TEST #08-D:
// Skipped LEVEL1 test
//
// -----
//
// TEST #08-E:
// Skipped consecutive LEVEL1 test
//
///////////////////////////////////////////////////////////////////
// Variables definitions
///////////////////////////////////////////////////////////////////
    ETr1 = 10 // Delay before Trigger commands
    ETr2 = 20 // Delay before Trigger commands
    TRAIL 500
///////////////////////////////////////////////////////////////////
// MCC initialisation
///////////////////////////////////////////////////////////////////
1000 SOFT_RESET
30 WR_REGISTER CSR 0013 // 160 Mbit/s
//30 WR_REGISTER CSR 0012 // 80 Mbit/s double line
//30 WR_REGISTER CSR 0011 // 80 Mbit/s single line
//30 WR_REGISTER CSR 0010 // 40 Mbit/s
30 WR_REGISTER FEEN FFFF // All FE enabled
///////////////////////////////////////////////////////////////////
// 3 contiguous LEVEL1 (2 SkippedLVL1)
///////////////////////////////////////////////////////////////////
500 WR_REGISTER LV1 0200 // 3 contiguous LEVEL1
10 ENABLE_DATATAKE // Select Run mode
//
LOOP i 1 6 // 18 Trigger (2 skipped)
    $ETr2 TRIGGER
END_LOOP
//
```

```
LOOP i 100 1300 300 // 16 EmptyEvents
$! EVENT
    HITS_EOE ffff 0 0
100 HITS_EOE ffff 0 0
200 HITS_EOE ffff 0 0
    END_EVENT
END_LOOP
//
1400 EVENT
    HITS_EOE ffff 0 0
    END_EVENT
//
1900 WR_REGISTER LV1 0000 // NO contiguous LEVEL1
10 ENABLE_DATATAKE // Select Run mode
$ETr1 TRIGGER
100 EVENT ffff 0 0
////////////////////////////////////
// 2 contiguous LEVEL1 (3 SkippedLVL1)
////////////////////////////////////
500 WR_REGISTER LV1 0000 // NO contiguous LEVEL1
10 ENABLE_DATATAKE // Select Run mode
//
LOOP i 400 1800 100 // 15 Trigger and 15 EmptyEvents
    $ETr1 TRIGGER
    $! EVENT ffff 0 0
END_LOOP
//
30 WR_REGISTER LV1 0100 // 2 contiguous LEVEL1
10 ENABLE_DATATAKE // Select Run mode
//
$ETr1 TRIGGER
$ETr2 TRIGGER
1780 EVENT ffff 0 0
    EVENT
    END_EVENT
//
1900 WR_REGISTER LV1 0000 // NO contiguous LEVEL1
10 ENABLE_DATATAKE // Select Run mode
$ETr1 TRIGGER
100 EVENT ffff 0 0
////////////////////////////////////
// 16 contiguous LEVEL1 (8 SkippedLVL1)
////////////////////////////////////
500 WR_REGISTER LV1 0000 // NO contiguous LEVEL1
10 ENABLE_DATATAKE // Select Run mode
//
LOOP i 400 1100 100 // 8 Trigger and 8 EmptyEvents
    $ETr1 TRIGGER
    $! EVENT ffff 0 0
END_LOOP
//
30 WR_REGISTER LV1 0f00 // 16 contiguous LEVEL1
10 ENABLE_DATATAKE // Select Run mode
//
$ETr1 TRIGGER
    EVENT
        LOOP i 1100 1800 100
$! HITS_EOE ffff 0 0
    END_LOOP
    END_EVENT
$ETr2 TRIGGER
    EVENT
    END_EVENT
//
1900 WR_REGISTER LV1 0000 // NO contiguous LEVEL1
10 ENABLE_DATATAKE // Select Run mode
```

```
$ETr1 TRIGGER
100 EVENT ffff 0 0
////////////////////////////////////
// 8 contiguous and overlapping LEVEL1
////////////////////////////////////
500 WR_REGISTER LV1 0700 // 8 contiguous LEVEL1
10 ENABLE_DATATAKE // Select Run mode
//
20 TRIGGER
0 TRIGGER // Overlap with last 3 triggers
//2 TRIGGER // Overlap with last trigger (PEF overflow)
//
EVENT
  LOOP i 100 500 100
  $i HITS_EOE ffff 0 0
  END_LOOP
  END_EVENT
EVENT
  LOOP i 600 1600 100
  $i HITS_EOE ffff 0 0
  END_LOOP
  END_EVENT
// EVENT
// LOOP i 1400 1600 100
// $i HITS_EOE ffff 0 0
// END_LOOP
// END_EVENT
//
1900 WR_REGISTER LV1 0000 // NO contiguous LEVEL1
10 ENABLE_DATATAKE // Select Run mode
$ETr1 TRIGGER
100 EVENT ffff 0 0
//
// END-TEST #08-E:
// Skipped consecutive LEVEL1 test
//
```

3.4 Delta-04.spx

```
//  
// 16 Consecutive LVL1 40 Mb/sec 90 hits  
// Voltage sweep between 1.7 and 2.2 V  
//  
  
////////////////////////////////////  
// Variables definitions  
////////////////////////////////////  
  
    TRAIL 25000  
  
////////////////////////////////////  
// MCC initialisation  
////////////////////////////////////  
100 SOFT_RESET  
30 WR_REGISTER CSR 0010    // 160 Mbit/s  
30 WR_REGISTER FEEN    0fff    // All FE enabled  
30 WR_REGISTER LV1 0f00    // 16 contiguous LEVEL1  
30 ENABLE_DATATAKE  
  
LOOP i 1 12  
    10 TRIGGER  
    20 EVENT  
        LOOP j 1 5  
            HITS_EOE 0fff    0 0  
        END_LOOP  
        HITS_EOE 0fff 90 0  
        LOOP j 1 10  
            HITS_EOE 0fff    0 0  
        END_LOOP  
    END_EVENT  
    28000 DELAY  
END_LOOP
```

3.5 Delta-05.spx

```
//  
// TEST #09-A:  
// EvBuilder disable and FIFO check  
//  
////////////////////////////////////  
// Variables definitions  
////////////////////////////////////  
  
CSR0    = 0083      // 160 Mbit/s - ev builder disabled  
//CSR0  = 0082      // 80 Mbit/s double line - ev builder disabled  
//CSR0  = 0081      // 80 Mbit/s single line - ev builder disabled  
//CSR0  = 0080      // 40 Mbit/s - ev builder disabled  
  
CSR1    = 0003      // 160 Mbit/s  
//CSR1  = 0002      // 80 Mbit/s double line  
//CSR1  = 0001      // 80 Mbit/s single line  
//CSR1  = 0000      // 40 Mbit/s  
  
nRec    = 128      // ReceiverFIFO depth  
  
TRAIL 1000  
  
////////////////////////////////////  
// SoftReset and FIFO clear  
////////////////////////////////////  
100 SOFT_RESET  
30 WR_REGISTER CSR $CSR0  
  
30 WR_REGISTER FEEN 0xffff  
// FIFO clear  
LOOP Location 1 $nRec 1  
  30 WR_FIFO 3x0000000  
END_LOOP  
  
30 WR_REGISTER FEEN 0x00ff  
  
////////////////////////////////////  
// Event generation  
////////////////////////////////////  
100 ENABLE_DATATAKE  
  
// Trigger  
LOOP Location 1 15 1  
  30 TRIGGER  
END_LOOP  
  
100 EVENT  
  HITS_EOE 000f 2 0  
  HITS_EOE 00f0 2 0  
  HITS_EOE ff00 2 0  
END_EVENT  
  
100 EVENT  
  HITS_EOE 000f 2 0  
  HITS_EOE 00f0 2 0  
  HITS_EOE ff00 2 0  
END_EVENT  
  
100 EVENT  
  HITS_EOE 000f 2 0  
  HITS_EOE 00f0 2 0  
END_EVENT
```

```
100 EVENT
    HITS_EOE      000f    2    0
END_EVENT

////////////////////////////////////
// Scoreboard check #1
////////////////////////////////////
1000    RD_REGISTER SBRD

////////////////////////////////////
// Ev Builder enable #1
////////////////////////////////////
30    WR_REGISTER CSR $CSR1
30    ENABLE_DATATAKE

////////////////////////////////////
// Scoreboard check #2
////////////////////////////////////
1000    RD_REGISTER SBRD

////////////////////////////////////
// Ev Builder enable
////////////////////////////////////
30    WR_REGISTER FEEN    000f
30    WR_REGISTER CSR $CSR1
30    RD_REGISTER SBRD
300   ENABLE_DATATAKE

////////////////////////////////////
// Scoreboard check #3
////////////////////////////////////
1000    RD_REGISTER SBRD

////////////////////////////////////
// FIFO check
////////////////////////////////////
1000    DELAY
// Read Loop #1
LOOP i 0 9
    LOOP Location 1 16 1
        30    RD_FIFO FIFO0$i
    END_LOOP
END_LOOP
LOOP i 10 15
    LOOP Location 1 16 1
        30    RD_FIFO FIFO$i
    END_LOOP
END_LOOP

100    SOFT_RESET

// Read Loop #2
LOOP i 0 9
    LOOP Location 1 16 1
        30    RD_FIFO FIFO0$i
    END_LOOP
END_LOOP
LOOP i 10 15
    LOOP Location 1 16 1
        30    RD_FIFO FIFO$i
    END_LOOP
END_LOOP
```

3.6 Delta-06.spx

```
//-----  
// Test script for SimPix: test_events_nanni.spx  
// Group of events at 40x1, 40x2, 80x1 and 80x2  
//-----  
  
    active_fe = ffff  
    trigd = 7  
  
//          TRAIL 6000 +-> From test_events_nanni.spx  
//  
// Events at 40 x 1  
//  
100 SOFT_RESET  
30 WR_REGISTER FEEN $active_fe  
30 WR_REGISTER CSR 0 // MODE 40 x 1 (default 0)  
30   RD_REGISTER FEEN  
30   WR_REGISTER LV1 0  
  
30 enable_datatake  
  
1000   TRIGGER  
$trigd   EVENT  
        HITS_EOE $active_fe 6 3  
        END_EVENT  
  
        loop i 1 18  
400   TRIGGER  
$trigd   EVENT  
        HITS_EOE $active_fe 6 3  
        END_EVENT  
        end_loop  
        loop i 1 3  
2200   TRIGGER  
$trigd   EVENT  
        HITS_EOE $active_fe 6 3  
        END_EVENT  
        end_loop  
  
//  
// Events at 40 x 2  
//  
  
40000   SOFT_RESET  
30 WR_REGISTER FEEN $active_fe  
30 WR_REGISTER CSR 1 // MODE 40 x 2  
30   RD_REGISTER FEEN  
30   WR_REGISTER LV1 0  
  
30 enable_datatake  
  
1000   TRIGGER  
$trigd   EVENT  
        HITS_EOE $active_fe 6 3  
        END_EVENT  
  
        loop i 1 18  
400   TRIGGER  
$trigd   EVENT  
        HITS_EOE $active_fe 6 3  
        END_EVENT  
        end_loop  
        loop i 1 3
```

```
2200      TRIGGER
$strigd  EVENT
        HITS_EOE $active_fe 6 3
        END_EVENT
        end_loop

//
// Events at 80 x 1
//

20000    SOFT_RESET
30  WR_REGISTER FEEN $active_fe
30  WR_REGISTER CSR 2 // MODE 80 x 1
30      RD_REGISTER FEEN
30      WR_REGISTER LV1 0

30  enable_datatake

1000     TRIGGER
$strigd  EVENT
        HITS_EOE $active_fe 6 3
        END_EVENT

        loop i 1 18
400     TRIGGER
$strigd  EVENT
        HITS_EOE $active_fe 6 3
        END_EVENT
        end_loop
        loop i 1 3
2200     TRIGGER
$strigd  EVENT
        HITS_EOE $active_fe 6 3
        END_EVENT
        end_loop

//
// Events at 80 x 2
//

20000    SOFT_RESET
30  WR_REGISTER FEEN $active_fe
30  WR_REGISTER CSR 3 // MODE 80 x 2
30      RD_REGISTER FEEN
30      WR_REGISTER LV1 0

30  enable_datatake

1000     TRIGGER
$strigd  EVENT
        HITS_EOE $active_fe 6 3
        END_EVENT

        loop i 1 18
400     TRIGGER
$strigd  EVENT
        HITS_EOE $active_fe 6 3
        END_EVENT
        end_loop
        loop i 1 3
2200     TRIGGER
$strigd  EVENT
        HITS_EOE $active_fe 6 3
        END_EVENT
```

```
end_loop

//-----
// TEST #02:
// Output Line and Pattern Generation Test
//-----

// TRAIL 200 +-> From TEST-02.spx

//100 SOFT_RESET
6100 SOFT_RESET

////////////////////////////////////
// Output 101010... pattern at 40 MHz on each line (40x1)
// (20 MHz repetition rate)
////////////////////////////////////
30 WR_REGISTER LV1 0
30 WR_REGISTER CSR 20

////////////////////////////////////
// Output 101010... pattern at 40 MHz on each line (40x2)
// (40 MHz repetition rate)
////////////////////////////////////
100 SOFT_RESET
100 WR_REGISTER CSR 21

////////////////////////////////////
// Output 101010... pattern at 80 MHz on each line (80x1)
// (40 MHz repetition rate)
////////////////////////////////////
100 SOFT_RESET
100 WR_REGISTER CSR 22

////////////////////////////////////
// Output 101010... pattern at 80 MHz on each line (80x2)
// (40 MHz repetition rate)
////////////////////////////////////
100 SOFT_RESET
100 WR_REGISTER CSR 23

200 SOFT_RESET

//
// END-TEST #02:
// Output Line and Pattern Generation Test
//
```

3.7 Delta-07.spx

```
// Test script for SimPix
//
// Delay line

    active_fe = ffff
    trigd = 7

        TRAIL 100

100 SOFT_RESET
30     WR_REGISTER CNT 10
30     RD_REGISTER CNT
//
// range 0
//
30 WR_REGISTER CAL 400
30 ENABLE_DATATAKE
30 CAL // 37 - Range 0, Delay 1x0 (0)
//
30 WR_REGISTER CAL 401
30 ENABLE_DATATAKE
30 CAL // 38 - Range 0, Delay 1x1 (1)
//
30 WR_REGISTER CAL 402
30 ENABLE_DATATAKE
30 CAL // 39 - Range 0, Delay 1x2 (2)
//
30 WR_REGISTER CAL 403
30 ENABLE_DATATAKE
30 CAL // 40 - Range 0, Delay 1x3 (3)
//
30 WR_REGISTER CAL 405
30 ENABLE_DATATAKE
30 CAL // 40 - Range 0, Delay 1x5 (5)
//
30 WR_REGISTER CAL 407
30 ENABLE_DATATAKE
30 CAL // 41 - Range 0, Delay 1x7 (7)
//
30 WR_REGISTER CAL 40b
30 ENABLE_DATATAKE
30 CAL // 42 - Range 0, Delay 1xb (11)
//
30 WR_REGISTER CAL 40f
30 ENABLE_DATATAKE
30 CAL // 43 - Range 0, Delay 1xf (15)
//
30 WR_REGISTER CAL 413
30 ENABLE_DATATAKE
30 CAL // 44 - Range 0, Delay 1x13 (19)
//
30 WR_REGISTER CAL 417
30 ENABLE_DATATAKE
30 CAL // 45 - Range 0, Delay 1x17 (23)
//
30 WR_REGISTER CAL 41b
30 ENABLE_DATATAKE
30 CAL // 46 - Range 0, Delay 1x1b (27)
//
30 WR_REGISTER CAL 41f
30 ENABLE_DATATAKE
30 CAL // 47 - Range 0, Delay 1x1f (31)
//
30 WR_REGISTER CAL 423
```

```
30 ENABLE_DATATAKE
30 CAL // 47 - Range 0, Delay 1x23 (35)
//
30 WR_REGISTER CAL 427
30 ENABLE_DATATAKE
30 CAL // 47 - Range 0, Delay 1x27 (39)
//
30 WR_REGISTER CAL 42b
30 ENABLE_DATATAKE
30 CAL // 47 - Range 0, Delay 1x2b (43)
//
30 WR_REGISTER CAL 42f
30 ENABLE_DATATAKE
30 CAL // 47 - Range 0, Delay 1x2f (47)
//
30 WR_REGISTER CAL 433
30 ENABLE_DATATAKE
30 CAL // 47 - Range 0, Delay 1x33 (51)
//
30 WR_REGISTER CAL 437
30 ENABLE_DATATAKE
30 CAL // 47 - Range 0, Delay 1x37 (55)
//
30 WR_REGISTER CAL 43b
30 ENABLE_DATATAKE
30 CAL // 47 - Range 0, Delay 1x3b (59)
//
30 WR_REGISTER CAL 43f
30 ENABLE_DATATAKE
30 CAL // 48 - Range 0, Delay 1x3f (63)
//
// range 1
//
30 WR_REGISTER CAL 440
30 ENABLE_DATATAKE
30 CAL // 37 - Range 1, Delay 1x0 (0)
//
30 WR_REGISTER CAL 441
30 ENABLE_DATATAKE
30 CAL // 38 - Range 1, Delay 1x1 (1)
//
30 WR_REGISTER CAL 442
30 ENABLE_DATATAKE
30 CAL // 39 - Range 1, Delay 1x2 (2)
//
30 WR_REGISTER CAL 443
30 ENABLE_DATATAKE
30 CAL // 40 - Range 1, Delay 1x3 (3)
//
30 WR_REGISTER CAL 445
30 ENABLE_DATATAKE
30 CAL // 40 - Range 1, Delay 1x5 (5)
//
30 WR_REGISTER CAL 447
30 ENABLE_DATATAKE
30 CAL // 41 - Range 1, Delay 1x7 (7)
//
30 WR_REGISTER CAL 44b
30 ENABLE_DATATAKE
30 CAL // 42 - Range 1, Delay 1xb (11)
//
30 WR_REGISTER CAL 44f
30 ENABLE_DATATAKE
30 CAL // 43 - Range 1, Delay 1xf (15)
//
30 WR_REGISTER CAL 453
```

```
30 ENABLE_DATATAKE
30 CAL // 44 - Range 1, Delay 1x13 (19)
//
30 WR_REGISTER CAL 457
30 ENABLE_DATATAKE
30 CAL // 45 - Range 1, Delay 1x17 (23)
//
30 WR_REGISTER CAL 45b
30 ENABLE_DATATAKE
30 CAL // 46 - Range 1, Delay 1x1b (27)
//
30 WR_REGISTER CAL 45f
30 ENABLE_DATATAKE
30 CAL // 47 - Range 1, Delay 1x1f (31)
//
30 WR_REGISTER CAL 463
30 ENABLE_DATATAKE
30 CAL // 47 - Range 1, Delay 1x23 (35)
//
30 WR_REGISTER CAL 467
30 ENABLE_DATATAKE
30 CAL // 47 - Range 1, Delay 1x27 (39)
//
30 WR_REGISTER CAL 46b
30 ENABLE_DATATAKE
30 CAL // 47 - Range 1, Delay 1x2b (43)
//
30 WR_REGISTER CAL 46f
30 ENABLE_DATATAKE
30 CAL // 47 - Range 1, Delay 1x2f (47)
//
30 WR_REGISTER CAL 473
30 ENABLE_DATATAKE
30 CAL // 47 - Range 1, Delay 1x33 (51)
//
30 WR_REGISTER CAL 477
30 ENABLE_DATATAKE
30 CAL // 47 - Range 1, Delay 1x37 (55)
//
30 WR_REGISTER CAL 47b
30 ENABLE_DATATAKE
30 CAL // 47 - Range 1, Delay 1x3b (59)
//
30 WR_REGISTER CAL 47f
30 ENABLE_DATATAKE
30 CAL // 48 - Range 1, Delay 1x3f (63)
//
// range 3
//
30 WR_REGISTER CAL 4c0
30 ENABLE_DATATAKE
30 CAL // 37 - Range 3, Delay 1x0 (0)
//
30 WR_REGISTER CAL 4c1
30 ENABLE_DATATAKE
30 CAL // 38 - Range 3, Delay 1x1 (1)
//
30 WR_REGISTER CAL 4c2
30 ENABLE_DATATAKE
30 CAL // 39 - Range 3, Delay 1x2 (2)
//
30 WR_REGISTER CAL 4c3
30 ENABLE_DATATAKE
30 CAL // 40 - Range 3, Delay 1x3 (3)
//
30 WR_REGISTER CAL 4c5
```

```
30 ENABLE_DATATAKE
30 CAL // 40 - Range 3, Delay 1x5 (5)
//
30 WR_REGISTER CAL 4c7
30 ENABLE_DATATAKE
30 CAL // 41 - Range 3, Delay 1x7 (7)
//
30 WR_REGISTER CAL 4cb
30 ENABLE_DATATAKE
30 CAL // 42 - Range 3, Delay 1xb (11)
//
30 WR_REGISTER CAL 4cf
30 ENABLE_DATATAKE
30 CAL // 43 - Range 3, Delay 1xf (15)
//
30 WR_REGISTER CAL 4d3
30 ENABLE_DATATAKE
30 CAL // 44 - Range 3, Delay 1x13 (19)
//
30 WR_REGISTER CAL 4d7
30 ENABLE_DATATAKE
30 CAL // 45 - Range 3, Delay 1x17 (23)
//
30 WR_REGISTER CAL 4db
30 ENABLE_DATATAKE
30 CAL // 46 - Range 3, Delay 1x1b (27)
//
30 WR_REGISTER CAL 4df
30 ENABLE_DATATAKE
30 CAL // 47 - Range 3, Delay 1x1f (31)
//
30 WR_REGISTER CAL 4e3
30 ENABLE_DATATAKE
30 CAL // 47 - Range 3, Delay 1x23 (35)
//
30 WR_REGISTER CAL 4e7
30 ENABLE_DATATAKE
30 CAL // 47 - Range 3, Delay 1x27 (39)
//
30 WR_REGISTER CAL 4eb
30 ENABLE_DATATAKE
30 CAL // 47 - Range 3, Delay 1x2b (43)
//
30 WR_REGISTER CAL 4ef
30 ENABLE_DATATAKE
30 CAL // 47 - Range 3, Delay 1x2f (47)
//
30 WR_REGISTER CAL 4f3
30 ENABLE_DATATAKE
30 CAL // 47 - Range 3, Delay 1x33 (51)
//
30 WR_REGISTER CAL 4f7
30 ENABLE_DATATAKE
30 CAL // 47 - Range 3, Delay 1x37 (55)
//
30 WR_REGISTER CAL 4fb
30 ENABLE_DATATAKE
30 CAL // 47 - Range 3, Delay 1x3b (59)
//
30 WR_REGISTER CAL 4ff
30 ENABLE_DATATAKE
30 CAL // 48 - Range 3, Delay 1x3f (63)
//
// range 7
//
30 WR_REGISTER CAL 5c0
```

```
30 ENABLE_DATATAKE
30 CAL // 37 - Range 7, Delay 1x0 (0)
//
30 WR_REGISTER CAL 5c1
30 ENABLE_DATATAKE
30 CAL // 38 - Range 7, Delay 1x1 (1)
//
30 WR_REGISTER CAL 5c2
30 ENABLE_DATATAKE
30 CAL // 39 - Range 7, Delay 1x2 (2)
//
30 WR_REGISTER CAL 5c3
30 ENABLE_DATATAKE
30 CAL // 40 - Range 7, Delay 1x3 (3)
//
30 WR_REGISTER CAL 5c5
30 ENABLE_DATATAKE
30 CAL // 40 - Range 7, Delay 1x5 (5)
//
30 WR_REGISTER CAL 5c7
30 ENABLE_DATATAKE
30 CAL // 41 - Range 7, Delay 1x7 (7)
//
30 WR_REGISTER CAL 5cb
30 ENABLE_DATATAKE
30 CAL // 42 - Range 7, Delay 1xb (11)
//
30 WR_REGISTER CAL 5cf
30 ENABLE_DATATAKE
30 CAL // 43 - Range 7, Delay 1xf (15)
//
30 WR_REGISTER CAL 5d3
30 ENABLE_DATATAKE
30 CAL // 44 - Range 7, Delay 1x13 (19)
//
30 WR_REGISTER CAL 5d7
30 ENABLE_DATATAKE
30 CAL // 45 - Range 7, Delay 1x17 (23)
//
30 WR_REGISTER CAL 5db
30 ENABLE_DATATAKE
30 CAL // 46 - Range 7, Delay 1x1b (27)
//
30 WR_REGISTER CAL 5df
30 ENABLE_DATATAKE
30 CAL // 47 - Range 7, Delay 1x1f (31)
//
30 WR_REGISTER CAL 5e3
30 ENABLE_DATATAKE
30 CAL // 47 - Range 7, Delay 1x23 (35)
//
30 WR_REGISTER CAL 5e7
30 ENABLE_DATATAKE
30 CAL // 47 - Range 7, Delay 1x27 (39)
//
30 WR_REGISTER CAL 5eb
30 ENABLE_DATATAKE
30 CAL // 47 - Range 7, Delay 1x2b (43)
//
30 WR_REGISTER CAL 5ef
30 ENABLE_DATATAKE
30 CAL // 47 - Range 7, Delay 1x2f (47)
//
30 WR_REGISTER CAL 5f3
30 ENABLE_DATATAKE
30 CAL // 47 - Range 7, Delay 1x33 (51)
```

```
//
30 WR_REGISTER CAL 5f7
30 ENABLE_DATATAKE
30 CAL // 47 - Range 7, Delay 1x37 (55)
//
30 WR_REGISTER CAL 5fb
30 ENABLE_DATATAKE
30 CAL // 47 - Range 7, Delay 1x3b (59)
//
30 WR_REGISTER CAL 5ff
30 ENABLE_DATATAKE
30 CAL // 48 - Range 7, Delay 1x3f (63)
//
// range f
//
30 WR_REGISTER CAL 7c0
30 ENABLE_DATATAKE
30 CAL // 37 - Range f, Delay 1x0 (0)
//
30 WR_REGISTER CAL 7c1
30 ENABLE_DATATAKE
30 CAL // 38 - Range f, Delay 1x1 (1)
//
30 WR_REGISTER CAL 7c2
30 ENABLE_DATATAKE
30 CAL // 39 - Range f, Delay 1x2 (2)
//
30 WR_REGISTER CAL 7c3
30 ENABLE_DATATAKE
30 CAL // 40 - Range f, Delay 1x3 (3)
//
30 WR_REGISTER CAL 7c5
30 ENABLE_DATATAKE
30 CAL // 40 - Range f, Delay 1x5 (5)
//
30 WR_REGISTER CAL 7c7
30 ENABLE_DATATAKE
30 CAL // 41 - Range f, Delay 1x7 (7)
//
30 WR_REGISTER CAL 7cb
30 ENABLE_DATATAKE
30 CAL // 42 - Range f, Delay 1xb (11)
//
30 WR_REGISTER CAL 7cf
30 ENABLE_DATATAKE
30 CAL // 43 - Range f, Delay 1xf (15)
//
30 WR_REGISTER CAL 7d3
30 ENABLE_DATATAKE
30 CAL // 44 - Range f, Delay 1x13 (19)
//
30 WR_REGISTER CAL 7d7
30 ENABLE_DATATAKE
30 CAL // 45 - Range f, Delay 1x17 (23)
//
30 WR_REGISTER CAL 7db
30 ENABLE_DATATAKE
30 CAL // 46 - Range f, Delay 1x1b (27)
//
30 WR_REGISTER CAL 7df
30 ENABLE_DATATAKE
30 CAL // 47 - Range f, Delay 1x1f (31)
//
30 WR_REGISTER CAL 7e3
30 ENABLE_DATATAKE
30 CAL // 47 - Range f, Delay 1x23 (35)
```

```
//  
30 WR_REGISTER CAL 7e7  
30 ENABLE_DATATAKE  
30 CAL // 47 - Range f, Delay 1x27 (39)  
//  
30 WR_REGISTER CAL 7eb  
30 ENABLE_DATATAKE  
30 CAL // 47 - Range f, Delay 1x2b (43)  
//  
30 WR_REGISTER CAL 7ef  
30 ENABLE_DATATAKE  
30 CAL // 47 - Range f, Delay 1x2f (47)  
//  
30 WR_REGISTER CAL 7f3  
30 ENABLE_DATATAKE  
30 CAL // 47 - Range f, Delay 1x33 (51)  
//  
30 WR_REGISTER CAL 7f7  
30 ENABLE_DATATAKE  
30 CAL // 47 - Range f, Delay 1x37 (55)  
//  
30 WR_REGISTER CAL 7fb  
30 ENABLE_DATATAKE  
30 CAL // 47 - Range f, Delay 1x3b (59)  
//  
30 WR_REGISTER CAL 7ff  
30 ENABLE_DATATAKE  
30 CAL // 48 - Range f, Delay 1x3f (63)  
//  
// 0, 1, 2, 3, 5, 7, 11, 15, 19, 23, 27, 31, 35, 39, 43, 47, 51, 55, 59, 63  
//  
// 0, 1, 3, 7, f  
//
```

3.8 Delta-08,spx

```
// test vector to test AMS/DSM

    active_fe = ffff
    trigd = 7

        TRAIL 550
//
// Events at 40 x 1
//
// AMSDSM = 1 (AMS pinout)
100 SOFT_RESET
30 WR_REGISTER FEEN $active_fe
30 WR_REGISTER CSR 1c // MODE 40 x 1 (default 1c)
30   RD_REGISTER FEEN
30   WR_REGISTER LV1 0

30 enable_datatake

100 TRIGGER
$trigd EVENT
    HITS_EOE $active_fe 1 0
    END_EVENT

// AMSDSM = 0 (DSM pinout)

600 SOFT_RESET
30 WR_REGISTER FEEN $active_fe
30 WR_REGISTER CSR 1c // MODE 40 x 1 (default 1c)
30   RD_REGISTER FEEN
30   WR_REGISTER LV1 0

30 enable_datatake

100 TRIGGER
$trigd EVENT
    HITS_EOE $active_fe 1 0
    END_EVENT

// End of Script
```

3.9 Delta-09.spx

```
//  
// 09-Delta.spx  
// SYNC Generation, WrFrontEnd/RdFrontEnd  
// Delay of DTI nominal is 19.  
  
    active_fe = ffff  
    trigd = 7  
  
        FE_0 = 0  
        FE_1 = 1  
        FE_2 = 2  
        FE_3 = 3  
        FE_4 = 4  
        FE_5 = 5  
        FE_6 = 6  
        FE_7 = 7  
        FE_8 = 8  
        FE_9 = 9  
        FE_10 = A  
        FE_11 = B  
        FE_12 = C  
        FE_13 = D  
        FE_14 = E  
        FE_15 = F  
  
    TRAIL 10  
  
// Start TEST#1  
// Test inputs: RSIB  
  
// insert RSIB = 0  
30 WR_REGISTER FEEN $active_fe  
// insert RSIB  
30 RD_REGISTER FEEN 0xffff  
30 WR_REGISTER FEEN $active_fe  
  
// End test#1  
  
// Start test#2  
// Test input: DCI  
// Test output: DTO, DTO2, SYNC  
40 GLOBAL_RESET_FE 0x0  
// End test#2  
  
// Start test#3  
// Test input: DCI, DTI<15:0>, DTIa<15:8>, AMSDSM  
// Test output: DTO, DTO2, CCK, CCKa, LD, LDa, DAO  
  
// insert AMSDSM = 1 and generate DTIa<15:8>|DTI<7:0>, DTI<15:8> = 0x00  
  
30 WR_REGISTER CNT 0x0082 // 16-CCK, LD, 16-CCK  
30 RD_FRONTEND 0 0xaaaaaaaa  
  
// insert AMSDSM = 0 and generate DTI<15:0>, DTIa<15:8> = 0x00  
50 WR_REGISTER CNT 0x0082 // 16-CCK, LD, 16-CCK  
30 RD_FRONTEND 0 0xaaaaaaaa  
// End test#3  
  
// Start test#4  
// Test input: DCI  
// Test output: STRO, LV1  
50 WR_REGISTER CNT 8  
30 WR_REGISTER CAL 401  
30 ENABLE_DATATAKE
```

```
30 CAL // 38 - Range 0, Delay 1x1
50 TRIGGER
// End test#4
```

4 Test Vectors: Scan Chain.

4.1 Scan Chain: Fault Coverage.

Logfile from ATPG vector generation. Only main blocks are listed.

```
*****
Report : test
        -coverage
Design : Top
Version: 1999.10
Date   : Thu Aug 28 14:57:00 2003
*****

Test program : `Top'
Depends on   :

Fault Coverage Report :

    Detect   : Number of faults Detected
    Aband   : Number of faults Abandoned
    Tied    : Number of faults Tied High or Low
    Redun   : Number of faults Redundant
    Untest  : Number of faults Untested
    Probl   : Number of faults Probably Detected
    Unpro   : Number of faults Hyperactive/Oscillating
    Total   : Number of faults Total
    Coverage : Detect / (Total - Tied - Redund)

Design/Cell      Detect   Aband   Tied   Redun   Untest   Probl   Unpro   Total   Coverage
-----
Core/Command     21459     0     396     0       37       0       0     21892   99.83%
Core/Evb         44810    1306     700     0     7550       0       0     54366   83.50%
Core/FEport       992         0       35     0     985       0       0      2012   50.18%
Core/Mport       1504         0       36     0       66       0       0      1606   95.80%
Core/Regs        45890     10    1076     0       38       0       0     47014   99.90%
Core/Ttc         4265         0     106     0       13       0       0      4384   99.70%
Core/fe00        11824     200     288     0    1676       0       0    13988   86.31%
Core/fe01        11916     346     288     0    1748       0       0    14298   85.05%
Core/fe02        12268     851     288     0    1299       0       0    14706   85.09%
Core/fe03        11767     196     288     0    1739       0       0    13990   85.88%
Core/fe04        12211     294     288     0    1771       0       0    14564   85.54%
Core/fe05        12128     942     288     0    1220       0       0    14578   84.87%
Core/fe06        12102     240     288     0    1926       0       0    14556   84.82%
Core/fe07        12065     634     288     0    1485       0       0    14472   85.06%
Core/fe08        12176     261     288     0    1877       0       0    14602   85.06%
Core/fe09        11767         0     288     0    2051       0       0    14106   85.16%
Core/fe10        12073     488     288     0    1635       0       0    14484   85.05%
Core/fe11        12057     206     288     0    2035       0       0    14586   84.33%
Core/fe12        11848     181     288     0    1881       0       0    14198   85.18%
Core/fe13        12002     232     288     0    1890       0       0    14412   84.98%
Core/fe14        11983         58     288     0    1969       0       0    14298   85.53%
Core/fe15        11981     329     288     0    1800       0       0    14398   84.91%
Core              312824    6774    6957     0   36839       0       0   363394   87.76%
Top              312858    6774    6957     0   37119       0       0   363708   87.70%
```


5 Test Logfile

5.1 Logfile for a Passed die

Logfile for device device#1 of wafer VX50V7X.

Device: 1 Station: 1 Site: 0 Date: Wed Dec 3 10:30:21 2003

Sequencer: mcc_seq

```

100 Open/Short-          Failing Pins: 0

110 Open/Short+          Failing Pins: 0

200 VDD supply           setup_devic  1.00 v <    2.00 v    <    3.00 v
210 IDD_Static           curr         35.00 ma <   53.24 ma    <   60.00 ma

220 IDD1 @ 1.25MHz       curr         35.00 ma <   56.82 ma    <   70.00 ma
221 IDD1 @ 2.5MHz        curr         35.00 ma <   60.16 ma    <   75.00 ma
222 IDD1 @ 5MHz          curr         40.00 ma <   66.82 ma    <   80.00 ma
223 IDD1 @ 10MHz         curr         45.00 ma <   79.85 ma    <   95.00 ma
224 IDD1 @ 20MHz         curr         50.00 ma <  106.25 ma    <  120.00 ma
225 IDD1 @ 40MHz         curr         80.00 ma <  140.78 ma    <  160.00 ma

300 LEAKAGE Test        Failing Pins: 0
    Pattern: leakage

310 IIH_CMOS_INPUTS     Failing Pins: 0

311 IIH_TSTEN            leak         15.00 ua <   20.82 ua    <   25.00 ua
312 IIH_AMSDSM           leak         -1.00 ua <    0.04 ua    <    1.00 ua
313 IIH_TSTIN            leak         15.00 ua <   20.89 ua    <   25.00 ua
314 IIH_RSIB             leak         -1.00 ua <    0.02 ua    <    1.00 ua

320 IIL_CMOS_INPUTS     Failing Pins: 0

321 IIL_TSTEN            leak         -1.00 ua <    0.01 ua    <    1.00 ua
322 IIL_AMSDSM           leak        -25.00 ua <  -20.84 ua    <  -15.00 ua
323 IIL_TSTIN            leak         -1.00 ua <   -0.06 ua    <    1.00 ua
324 IIL_RSIB             leak        -25.00 ua <  -20.93 ua    <  -15.00 ua

350 DELTA04 Test        Failing Pins: 0
    Pattern: delta04

360 VDD @ DELTA04       voltage_swe  1.70 v <=   1.78 v    <   1.86 v

400 VIN Test            Failing Pins: 0
    Pattern: inputs

410 VTN_DTI_0_TO_7       vin          0.000 v <=   0.000 v    <   0.040 v
411 VTN_DTI_8_TO_15      vin          0.000 v <=   0.000 v    <   0.040 v
412 VTN_DTIA_8_TO_15     vin          0.000 v <=   0.000 v    <   0.040 v
413 VTN_XCKIN            vin          0.000 v <=   0.000 v    <   0.040 v
414 VTN_DCI              vin          0.000 v <=   0.000 v    <   0.040 v
415 VTN_CK               vin          0.000 v <=   0.000 v    <   0.040 v

420 VTP_DTI_0_TO_7       vin          0.000 v <=   0.000 v    <   0.040 v
421 VTP_DTI_8_TO_15      vin          0.000 v <=   0.000 v    <   0.040 v
422 VTP_DTIA_8_TO_15     vin          0.000 v <=   0.000 v    <   0.040 v
    
```

423	VTP_XCKIN	vin	0.000	v <=	0.000	v	<	0.040	v
424	VTP_DCI	vin	0.000	v <=	0.000	v	<	0.040	v
425	VTP_CK	vin	0.000	v <=	0.000	v	<	0.040	v
500	VOUT Test	Failing Pins: 0							
	Pattern:	outputs							
510	VOL_LD	vout	0.000	v <	0.062	v	<	0.200	v
511	VOL_LDA	vout	0.000	v <	0.069	v	<	0.200	v
512	VOL_CCK	vout	0.000	v <	0.064	v	<	0.200	v
513	VOL_CCKA	vout	0.000	v <	0.067	v	<	0.200	v
514	VOL_DAO	vout	0.000	v <	0.066	v	<	0.200	v
515	VOL_DTOP	vout	0.700	v <	0.760	v	<	0.810	v
516	VOL_DTO2P	vout	0.700	v <	0.760	v	<	0.810	v
517	VOL_LV1P	vout	0.700	v <	0.759	v	<	0.810	v
518	VOL_SYNCN	vout	0.700	v <	0.758	v	<	0.810	v
519	VOH_DTON	vout	0.970	v <	1.043	v	<	1.200	v
520	VOH_DTO2N	vout	0.970	v <	1.042	v	<	1.200	v
521	VOH_LV1N	vout	0.970	v <	1.032	v	<	1.200	v
522	VOH_SYNCNP	vout	0.970	v <	1.036	v	<	1.200	v
530	VOH_DTOP	vout	0.970	v <	1.045	v	<	1.200	v
531	VOH_DTO2P	vout	0.970	v <	1.043	v	<	1.200	v
532	VOL_DTON	vout	0.700	v <	0.761	v	<	0.810	v
533	VOL_DTO2N	vout	0.700	v <	0.760	v	<	0.810	v
540	VOH_LD	vout	1.800	v <	1.864	v	<	2.000	v
541	VOH_LDA	vout	1.800	v <	1.854	v	<	2.000	v
542	VOH_CCK	vout	1.800	v <	1.859	v	<	2.000	v
543	VOH_CCKA	vout	1.800	v <	1.855	v	<	2.000	v
544	VOH_DAO	vout	1.800	v <	1.855	v	<	2.000	v
545	VOH_SYNCN	vout	0.970	v <	1.034	v	<	1.200	v
546	VOL_SYNCNP	vout	0.700	v <	0.756	v	<	0.810	v
550	VOH_LV1P	vout	0.970	v <	1.028	v	<	1.200	v
551	VOL_LV1N	vout	0.700	v <	0.755	v	<	0.810	v
600	DELTA01 Test	Failing Pins: 0							
	Pattern:	delta01							
601	DELTA02 Test	Failing Pins: 0							
	Pattern:	delta02							
602	DELTA03 Test	Failing Pins: 0							
	Pattern:	delta03							
603	DELTA05 Test	Failing Pins: 0							
	Pattern:	delta05							
604	DELTA08 Test	Failing Pins: 0							
	Pattern:	delta08							
605	SCAN0 Test	Failing Pins: 0							
	Pattern:	scan0							
606	SCAN1 Test	Failing Pins: 0							
	Pattern:	scan1							
607	SCAN2 Test	Failing Pins: 0							
	Pattern:	scan2							
608	DELTA06 Test	Failing Pins: 0							
	Pattern:	delta06							
700	Delay Line #01	delay_line	375.0 ns	<=	393.6 ns		<=	400.0 ns	

701	Delay Line #02	delay_line	375.0 ns	<=	393.9 ns	<=	700.0 ns
702	Delay Line #03	delay_line	375.0 ns	<=	394.7 ns	<=	700.0 ns
703	Delay Line #04	delay_line	375.0 ns	<=	394.9 ns	<=	700.0 ns
704	Delay Line #05	delay_line	375.0 ns	<=	396.1 ns	<=	700.0 ns
705	Delay Line #06	delay_line	375.0 ns	<=	396.9 ns	<=	700.0 ns
706	Delay Line #07	delay_line	375.0 ns	<=	398.6 ns	<=	700.0 ns
707	Delay Line #08	delay_line	375.0 ns	<=	400.7 ns	<=	700.0 ns
708	Delay Line #09	delay_line	375.0 ns	<=	402.4 ns	<=	700.0 ns
709	Delay Line #10	delay_line	375.0 ns	<=	404.1 ns	<=	700.0 ns
710	Delay Line #11	delay_line	375.0 ns	<=	405.8 ns	<=	700.0 ns
711	Delay Line #12	delay_line	375.0 ns	<=	407.4 ns	<=	700.0 ns
712	Delay Line #13	delay_line	375.0 ns	<=	409.1 ns	<=	700.0 ns
713	Delay Line #14	delay_line	375.0 ns	<=	410.8 ns	<=	700.0 ns
714	Delay Line #15	delay_line	375.0 ns	<=	412.5 ns	<=	700.0 ns
715	Delay Line #16	delay_line	375.0 ns	<=	414.2 ns	<=	700.0 ns
716	Delay Line #17	delay_line	375.0 ns	<=	415.5 ns	<=	700.0 ns
717	Delay Line #18	delay_line	375.0 ns	<=	417.2 ns	<=	700.0 ns
718	Delay Line #19	delay_line	375.0 ns	<=	418.9 ns	<=	700.0 ns
719	Delay Line #20	delay_line	375.0 ns	<=	420.6 ns	<=	700.0 ns
720	Delay Line #21	delay_line	375.0 ns	<=	398.6 ns	<=	700.0 ns
721	Delay Line #22	delay_line	375.0 ns	<=	399.2 ns	<=	700.0 ns
722	Delay Line #23	delay_line	375.0 ns	<=	400.2 ns	<=	700.0 ns
723	Delay Line #24	delay_line	375.0 ns	<=	400.4 ns	<=	700.0 ns
724	Delay Line #25	delay_line	375.0 ns	<=	401.9 ns	<=	700.0 ns
725	Delay Line #26	delay_line	375.0 ns	<=	402.9 ns	<=	700.0 ns
726	Delay Line #27	delay_line	375.0 ns	<=	405.4 ns	<=	700.0 ns
727	Delay Line #28	delay_line	375.0 ns	<=	407.9 ns	<=	700.0 ns
728	Delay Line #29	delay_line	375.0 ns	<=	409.9 ns	<=	700.0 ns
729	Delay Line #30	delay_line	375.0 ns	<=	411.9 ns	<=	700.0 ns
730	Delay Line #31	delay_line	375.0 ns	<=	414.4 ns	<=	700.0 ns
731	Delay Line #32	delay_line	375.0 ns	<=	416.5 ns	<=	700.0 ns
732	Delay Line #33	delay_line	375.0 ns	<=	418.5 ns	<=	700.0 ns
733	Delay Line #34	delay_line	375.0 ns	<=	420.5 ns	<=	700.0 ns
734	Delay Line #35	delay_line	375.0 ns	<=	422.6 ns	<=	700.0 ns
735	Delay Line #36	delay_line	375.0 ns	<=	424.6 ns	<=	700.0 ns
736	Delay Line #37	delay_line	375.0 ns	<=	426.6 ns	<=	700.0 ns
737	Delay Line #38	delay_line	375.0 ns	<=	428.7 ns	<=	700.0 ns
738	Delay Line #39	delay_line	375.0 ns	<=	430.7 ns	<=	700.0 ns
739	Delay Line #40	delay_line	375.0 ns	<=	432.8 ns	<=	700.0 ns
740	Delay Line #41	delay_line	375.0 ns	<=	408.6 ns	<=	700.0 ns
741	Delay Line #42	delay_line	375.0 ns	<=	409.8 ns	<=	700.0 ns
742	Delay Line #43	delay_line	375.0 ns	<=	410.5 ns	<=	700.0 ns
743	Delay Line #44	delay_line	375.0 ns	<=	411.1 ns	<=	700.0 ns
744	Delay Line #45	delay_line	375.0 ns	<=	412.7 ns	<=	700.0 ns
745	Delay Line #46	delay_line	375.0 ns	<=	414.4 ns	<=	700.0 ns
746	Delay Line #47	delay_line	375.0 ns	<=	417.4 ns	<=	700.0 ns
747	Delay Line #48	delay_line	375.0 ns	<=	420.5 ns	<=	700.0 ns
748	Delay Line #49	delay_line	375.0 ns	<=	423.6 ns	<=	700.0 ns
749	Delay Line #50	delay_line	375.0 ns	<=	426.2 ns	<=	700.0 ns
750	Delay Line #51	delay_line	375.0 ns	<=	429.4 ns	<=	700.0 ns
751	Delay Line #52	delay_line	375.0 ns	<=	432.5 ns	<=	700.0 ns
752	Delay Line #53	delay_line	375.0 ns	<=	435.6 ns	<=	700.0 ns
753	Delay Line #54	delay_line	375.0 ns	<=	438.8 ns	<=	700.0 ns
754	Delay Line #55	delay_line	375.0 ns	<=	442.4 ns	<=	700.0 ns
755	Delay Line #56	delay_line	375.0 ns	<=	444.6 ns	<=	700.0 ns
756	Delay Line #57	delay_line	375.0 ns	<=	447.7 ns	<=	700.0 ns
757	Delay Line #58	delay_line	375.0 ns	<=	451.4 ns	<=	700.0 ns
758	Delay Line #59	delay_line	375.0 ns	<=	455.2 ns	<=	700.0 ns
759	Delay Line #60	delay_line	375.0 ns	<=	458.9 ns	<=	700.0 ns
760	Delay Line #61	delay_line	375.0 ns	<=	425.2 ns	<=	700.0 ns
761	Delay Line #62	delay_line	375.0 ns	<=	426.4 ns	<=	700.0 ns
762	Delay Line #63	delay_line	375.0 ns	<=	427.6 ns	<=	700.0 ns
763	Delay Line #64	delay_line	375.0 ns	<=	428.7 ns	<=	700.0 ns
764	Delay Line #65	delay_line	375.0 ns	<=	431.4 ns	<=	700.0 ns
765	Delay Line #66	delay_line	375.0 ns	<=	434.0 ns	<=	700.0 ns
766	Delay Line #67	delay_line	375.0 ns	<=	438.7 ns	<=	700.0 ns

767	Delay Line #68	delay_line	375.0 ns	<=	443.9 ns	<=	700.0 ns
768	Delay Line #69	delay_line	375.0 ns	<=	448.1 ns	<=	700.0 ns
769	Delay Line #70	delay_line	375.0 ns	<=	454.9 ns	<=	700.0 ns
770	Delay Line #71	delay_line	375.0 ns	<=	462.4 ns	<=	700.0 ns
771	Delay Line #72	delay_line	375.0 ns	<=	468.8 ns	<=	700.0 ns
772	Delay Line #73	delay_line	375.0 ns	<=	474.7 ns	<=	700.0 ns
773	Delay Line #74	delay_line	375.0 ns	<=	479.6 ns	<=	700.0 ns
774	Delay Line #75	delay_line	375.0 ns	<=	484.4 ns	<=	700.0 ns
775	Delay Line #76	delay_line	375.0 ns	<=	489.9 ns	<=	700.0 ns
776	Delay Line #77	delay_line	375.0 ns	<=	494.8 ns	<=	700.0 ns
777	Delay Line #78	delay_line	375.0 ns	<=	499.7 ns	<=	700.0 ns
778	Delay Line #79	delay_line	375.0 ns	<=	505.2 ns	<=	700.0 ns
779	Delay Line #80	delay_line	375.0 ns	<=	509.0 ns	<=	700.0 ns
780	Delay Line #81	delay_line	375.0 ns	<=	456.7 ns	<=	700.0 ns
781	Delay Line #82	delay_line	375.0 ns	<=	461.5 ns	<=	700.0 ns
782	Delay Line #83	delay_line	375.0 ns	<=	465.7 ns	<=	700.0 ns
783	Delay Line #84	delay_line	375.0 ns	<=	468.8 ns	<=	700.0 ns
784	Delay Line #85	delay_line	375.0 ns	<=	474.4 ns	<=	700.0 ns
785	Delay Line #86	delay_line	375.0 ns	<=	479.0 ns	<=	700.0 ns
786	Delay Line #87	delay_line	375.0 ns	<=	487.4 ns	<=	700.0 ns
787	Delay Line #88	delay_line	375.0 ns	<=	495.9 ns	<=	700.0 ns
788	Delay Line #89	delay_line	375.0 ns	<=	505.6 ns	<=	700.0 ns
789	Delay Line #90	delay_line	375.0 ns	<=	511.3 ns	<=	700.0 ns
790	Delay Line #91	delay_line	375.0 ns	<=	527.7 ns	<=	700.0 ns
791	Delay Line #92	delay_line	375.0 ns	<=	539.5 ns	<=	700.0 ns
792	Delay Line #93	delay_line	375.0 ns	<=	548.4 ns	<=	700.0 ns
793	Delay Line #94	delay_line	375.0 ns	<=	556.7 ns	<=	700.0 ns
794	Delay Line #95	delay_line	375.0 ns	<=	565.6 ns	<=	700.0 ns
795	Delay Line #96	delay_line	375.0 ns	<=	572.7 ns	<=	700.0 ns
796	Delay Line #97	delay_line	375.0 ns	<=	586.9 ns	<=	700.0 ns
797	Delay Line #98	delay_line	375.0 ns	<=	599.3 ns	<=	700.0 ns
798	Delay Line #99	delay_line	375.0 ns	<=	607.8 ns	<=	700.0 ns
799	Delay Line #100	delay_line	375.0 ns	<=	617.0 ns	<=	700.0 ns

Bin: 1 Wafer Coordinates: (18 , 11)