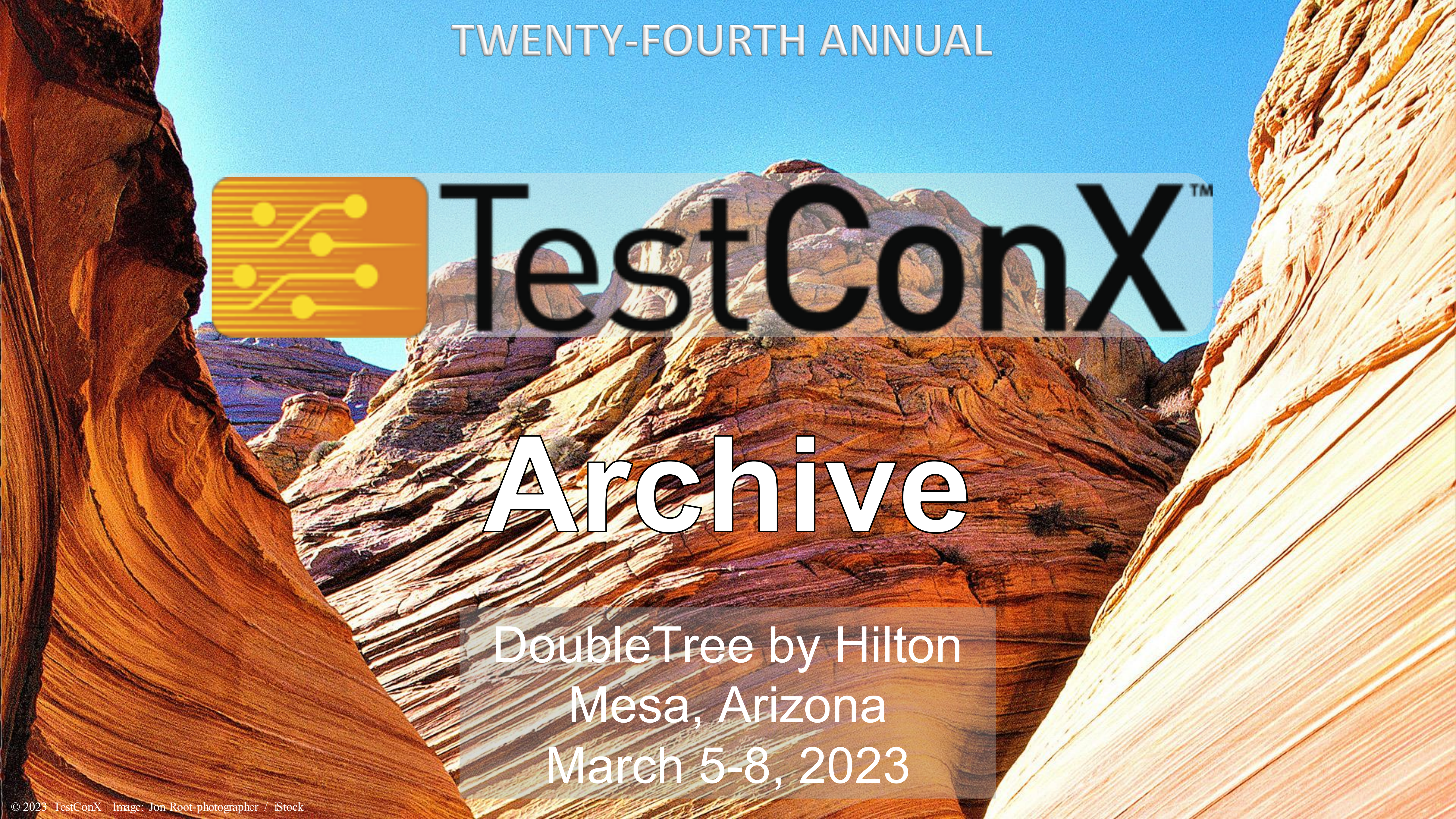


TWENTY-FOURTH ANNUAL

TestConX™

Archive

DoubleTree by Hilton
Mesa, Arizona
March 5-8, 2023

With Thanks to Our Sponsors!

Premier

HiCon

Honored

GLOBAL INNOVATION ISC

WIN SOLUTION

LEENO Total Interface Solutions

TSE

TestConX™

2023

With Thanks to Our Sponsors!

Distinguished

Johnstech™

smiths interconnect

MICRO CONTROL COMPANY SINCE 1972

TFE Align & Smart Contact

ADVANTEST®

Industry Partners

SWTEST

MEPTEC THE NEXT GENERATION

An Integrated Approach to Testing Analog Sub-Systems in Large Digital “Cheap” SoCs

Thecla Chomicz
NXP Semiconductors

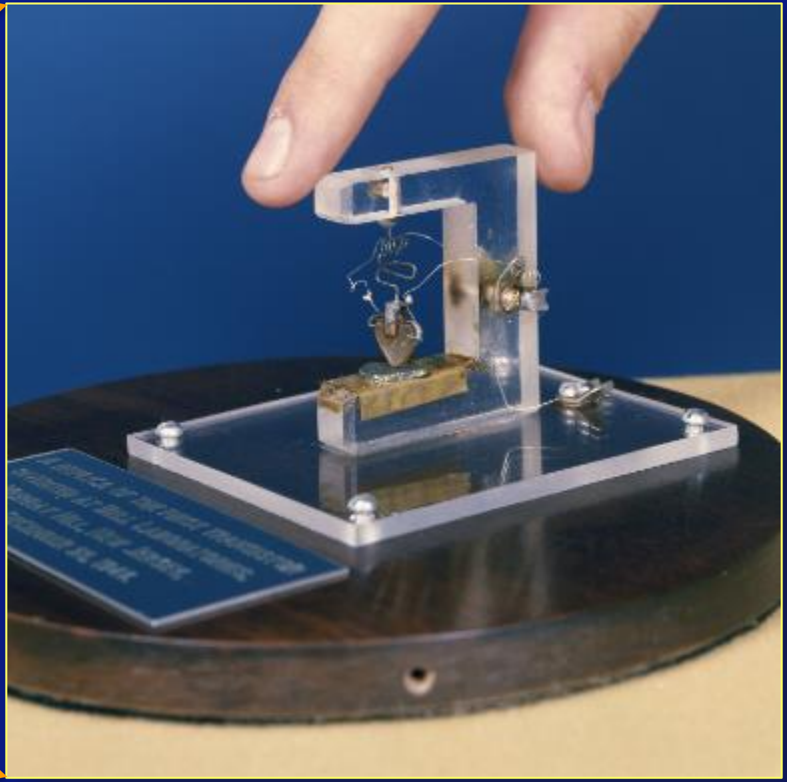
TestConX™

Mesa, Arizona ● March 5-8, 2023

Contents

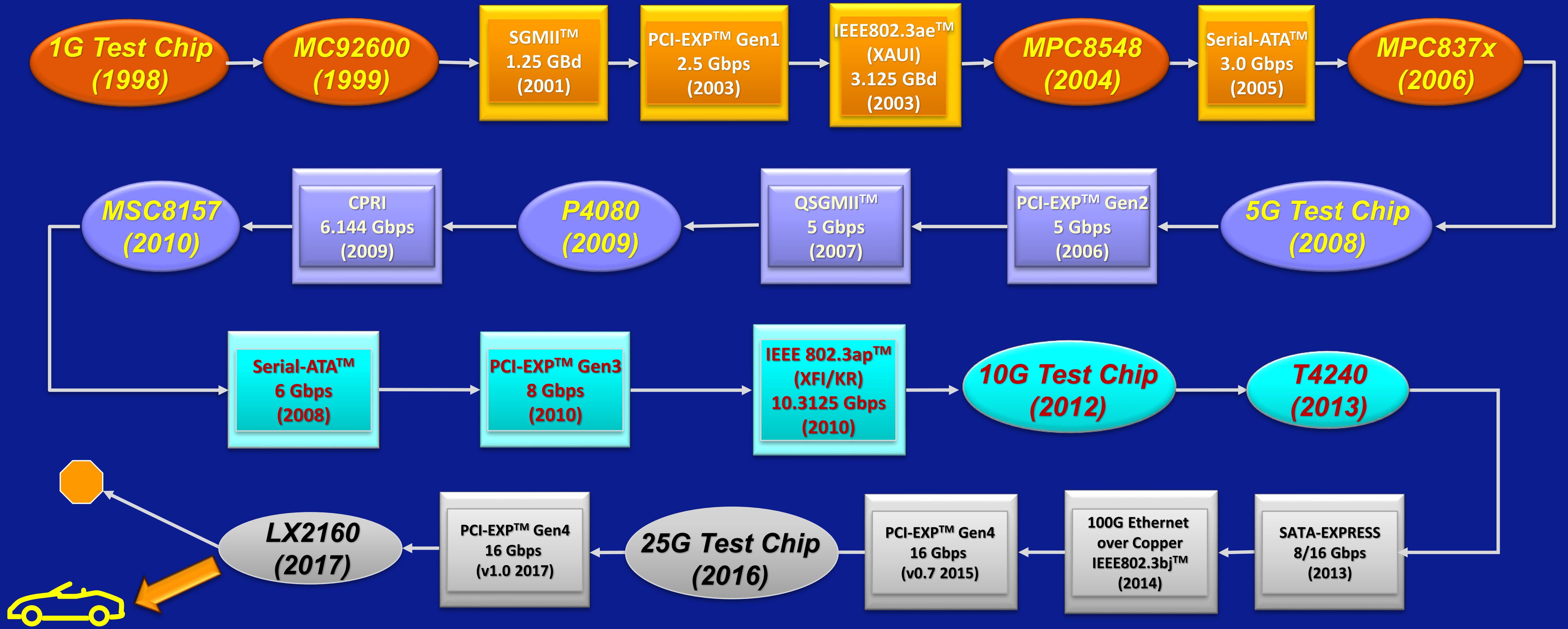
A look back…
(device count = 1)

John Bardeen, left, William Shockley, middle (sitting), and Walter Brattain
Nokia Corporation / AT&T Archives

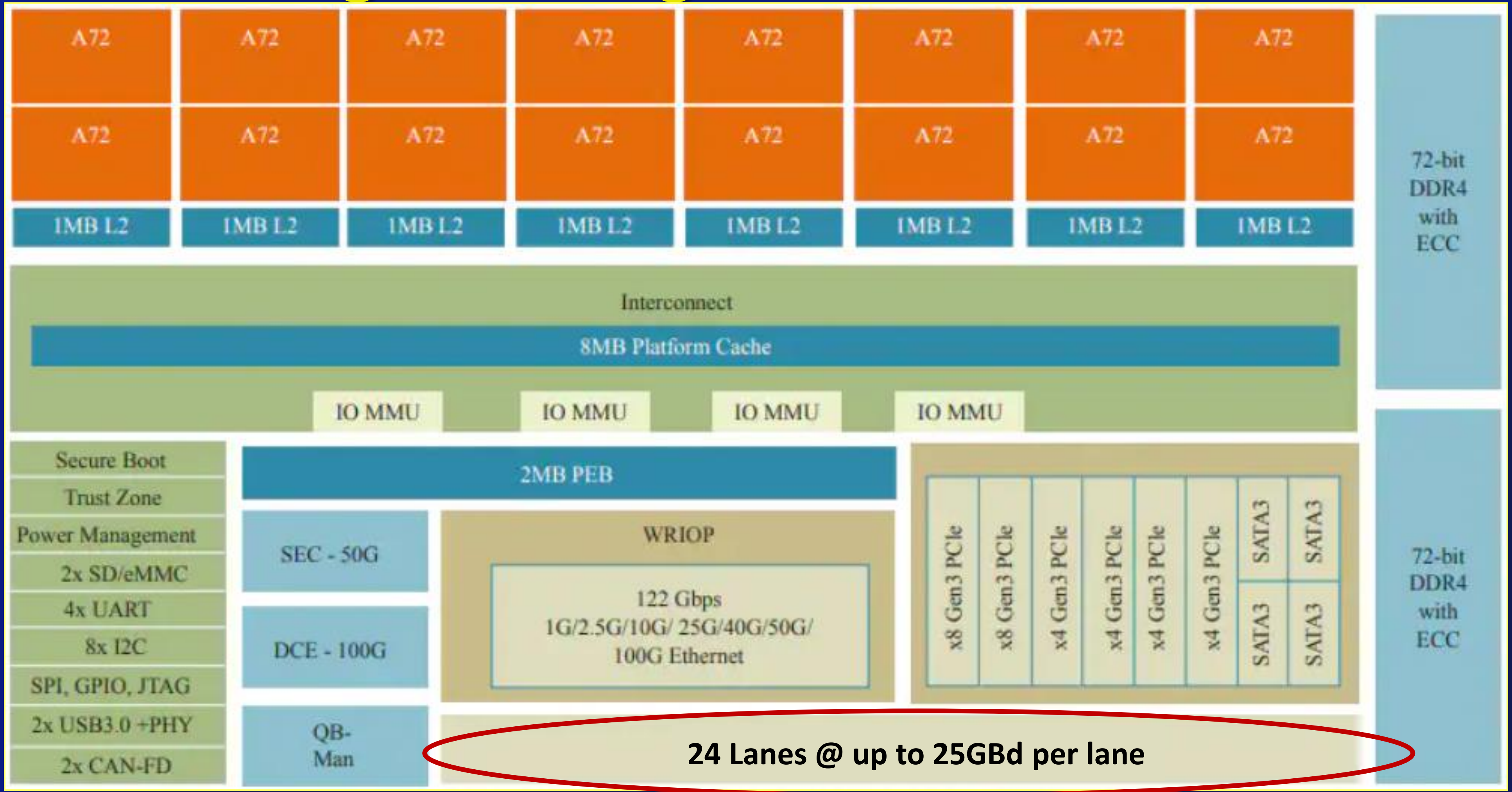


Science & Society Picture Library/SSPL

High Speed Serial Evolution

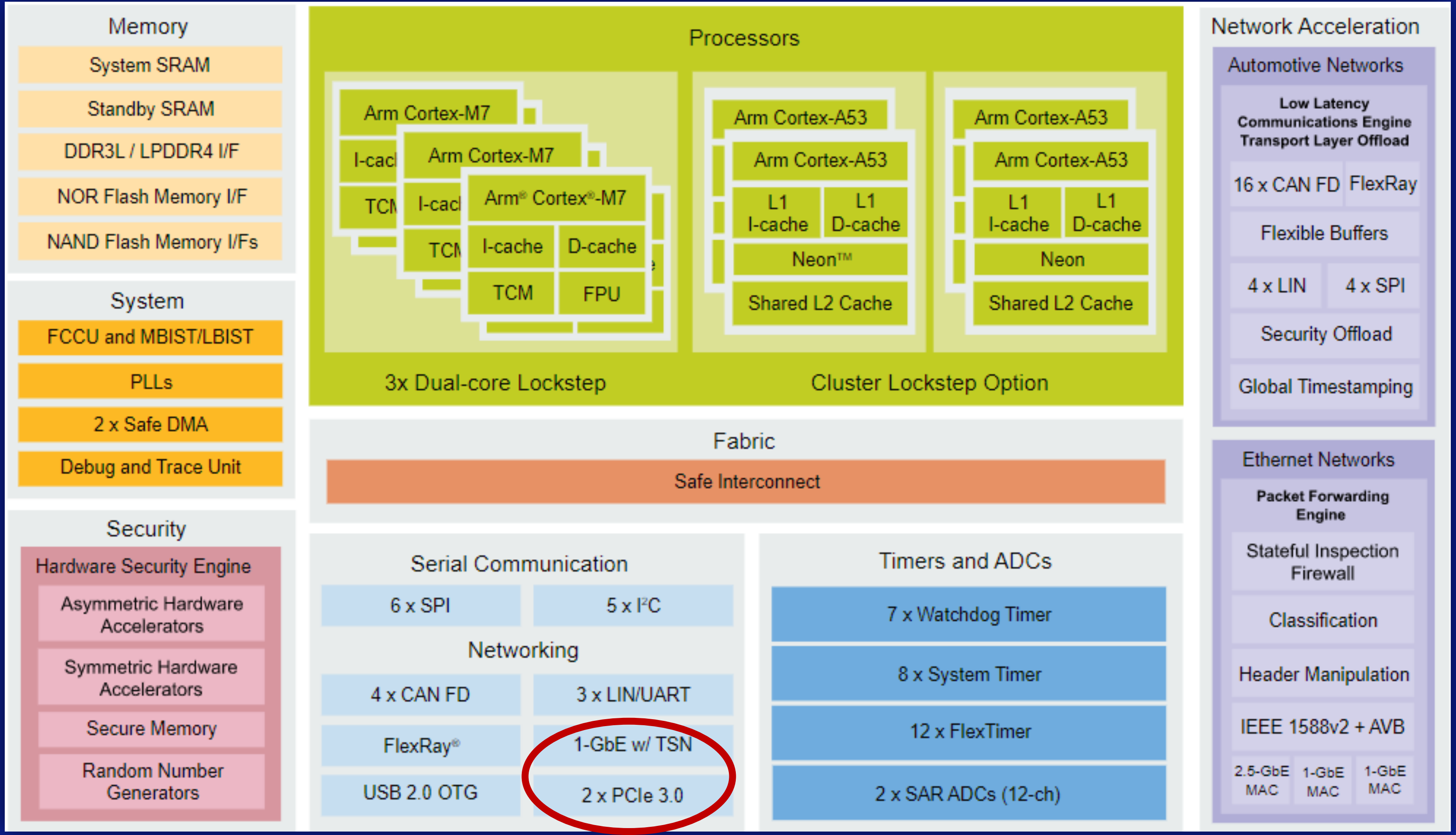

1G Test Chip (1998)
MC92600 (1999)
SGMII™ 1.25 GBd (2001)
PCI-EXP™ Gen1 2.5 Gbps (2003)
IEEE802.3ae™ (XAUI) 3.125 GBd (2003)
MPC8548 (2004)
Serial-ATA™ 3.0 Gbps (2005)
MPC837x (2006)
5G Test Chip (2008)
PCI-EXP™ Gen2 5 Gbps (2006)
QSGMII™ 5 Gbps (2007)
P4080 (2009)
CPRI 6.144 Gbps (2009)
MSC8157 (2010)
Serial-ATA™ 6 Gbps (2008)
PCI-EXP™ Gen3 8 Gbps (2010)
IEEE 802.3ap™ (XFI/KR) 10.3125 Gbps (2010)
10G Test Chip (2012)
T4240 (2013)
SATA-EXPRESS 8/16 Gbps (2013)
100G Ethernet over Copper IEEE802.3bj™ (2014)
PCI-EXP™ Gen4 16 Gbps (v0.7 2015)
25G Test Chip (2016)
PCI-EXP™ Gen4 16 Gbps (v1.0 2017)
LX2160 (2017)

Large Analog Block in LX2160



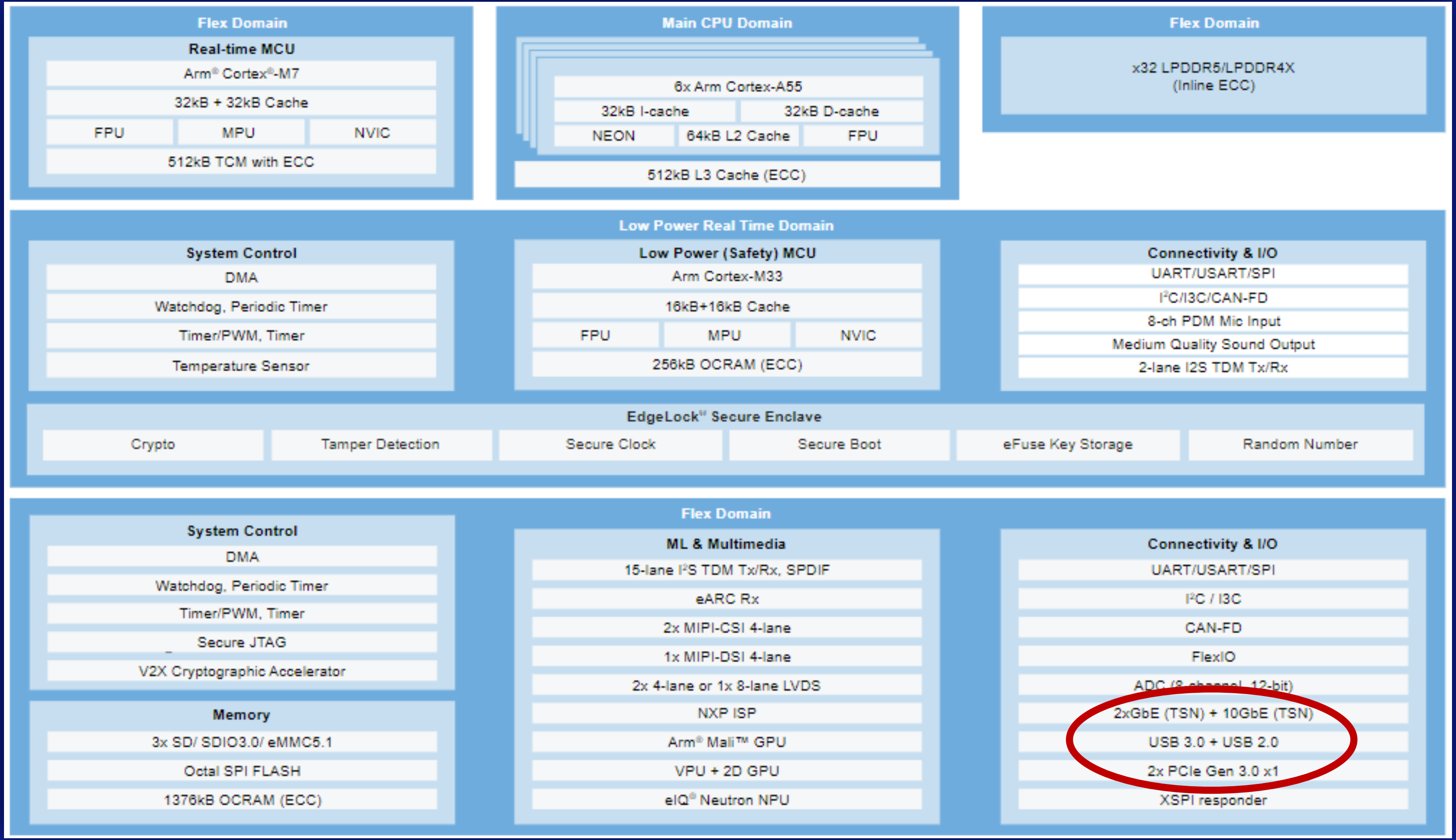
https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/layerscape-processors/layerscape-lx2160a-lx2120a-lx2080a-processors:LX2160A

Analog Blocks in S32G2



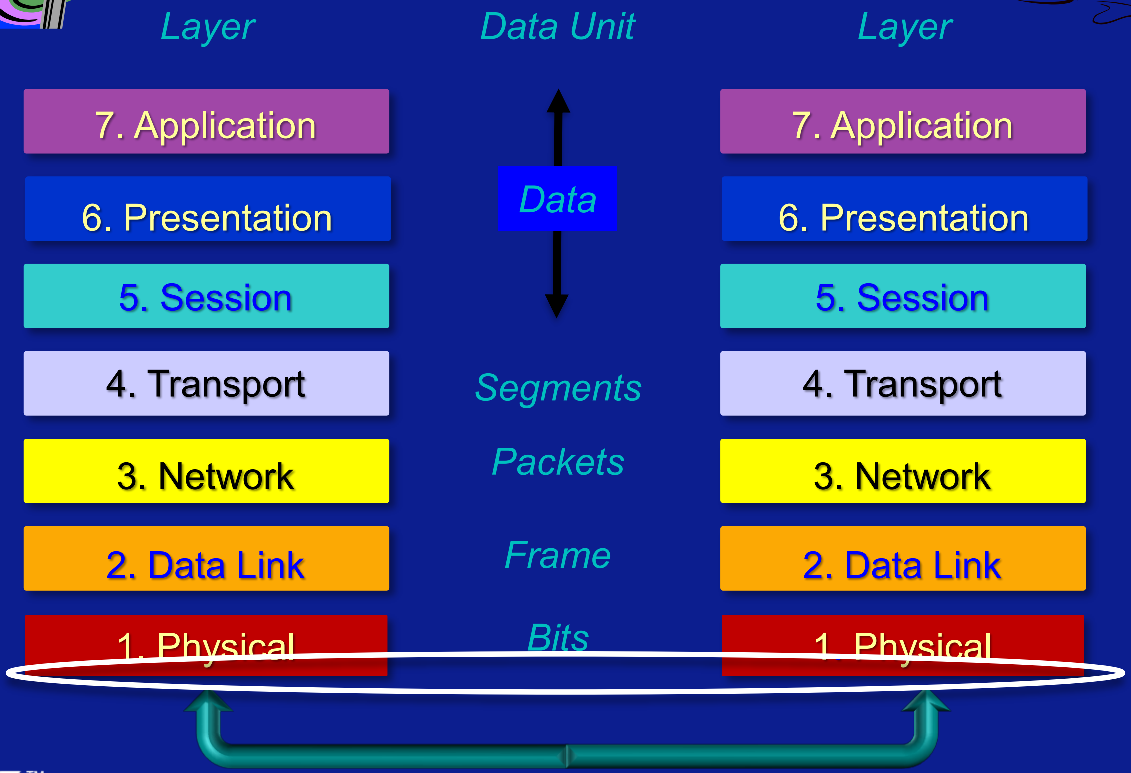
https://www.nxp.com/products/processors-and-microcontrollers/s32-automotive-platform/s32g-vehicle-network-processors/s32g2-processors-for-vehicle-networking:S32G2

Analog in I.MX95 (Pre-Production)

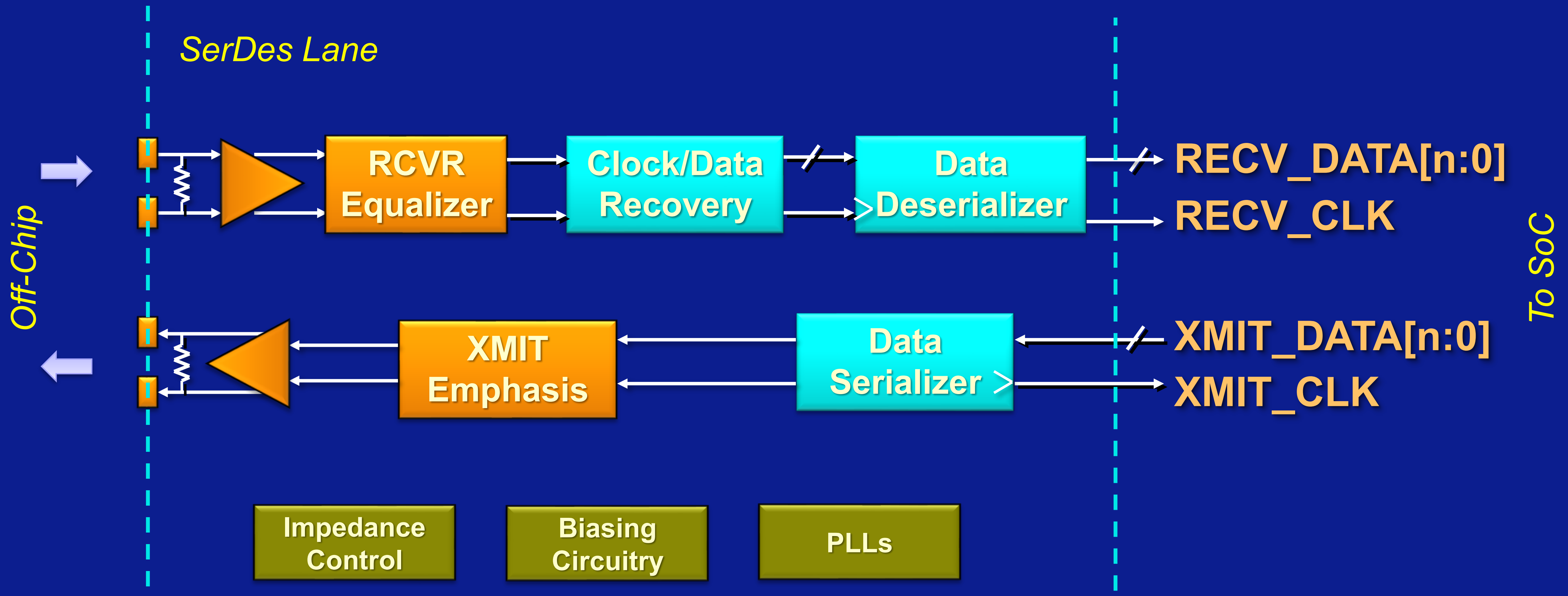


https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/i-mx-applications-processors/i-mx-9-processors/i-mx-95-applications-processor-familyhigh-performance-safety-enabled-platform-with-eiq-neutron-npu:iMX95

Open System Interconnect (OSI) 7-Layer Reference Model



A Closer Look



Starter Test Criteria

- Correct by Design
- Test to Specification
- Test *Quickly* for Manufacturing Anomalies
- *Provide for Debug*

Test Expectations Refined
(you can't sell if you can't get it off the Tester)

- Design to Protocol and SoC specifications using the FULL range of manufacturing tolerances and test conditions
- Characterize for Compliance in Lab
- Production Test for Basic Functionality and Manufacturing Anomalies
- Signal Integrity
- Stress Testing
- Probe
- Multi-Site Testing

➢ ***PROVE IT'S NOT THE ANALOG!!!***

Understanding the Test Environment

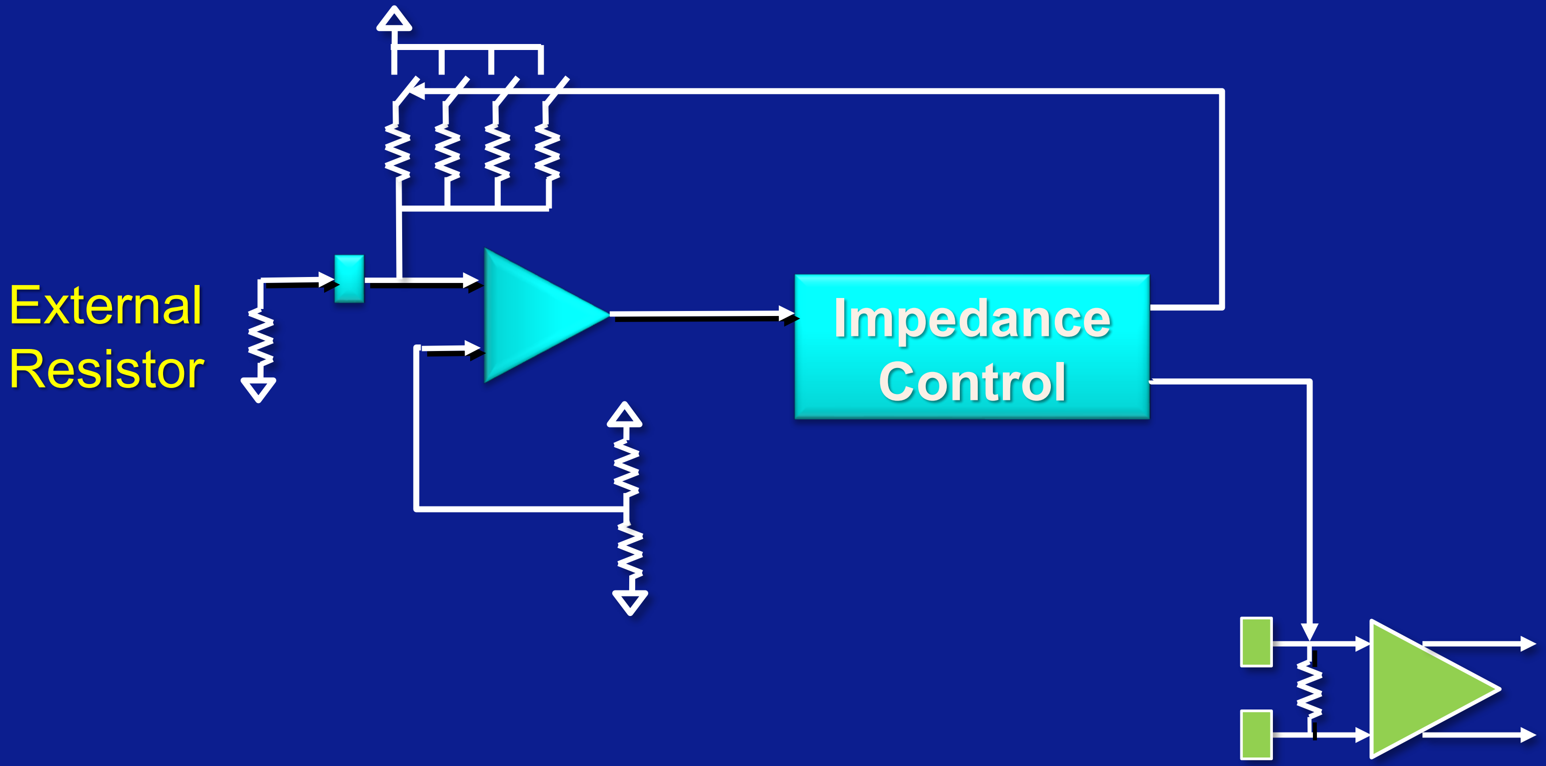
- Manufacturing Tolerances
- Testing with Large Digital Testers
  - Fixed Voltage
  - Fixed Clock
  - Deterministic Results
  - Load Board, Pogo Pins, Socket, ***Signal Integrity….***
  - Probe vs. Final Test
  - Multi-site Testing
  - $$$ → $

Every additional second on the tester is just taping dollars to the top of the package as it goes out the door.

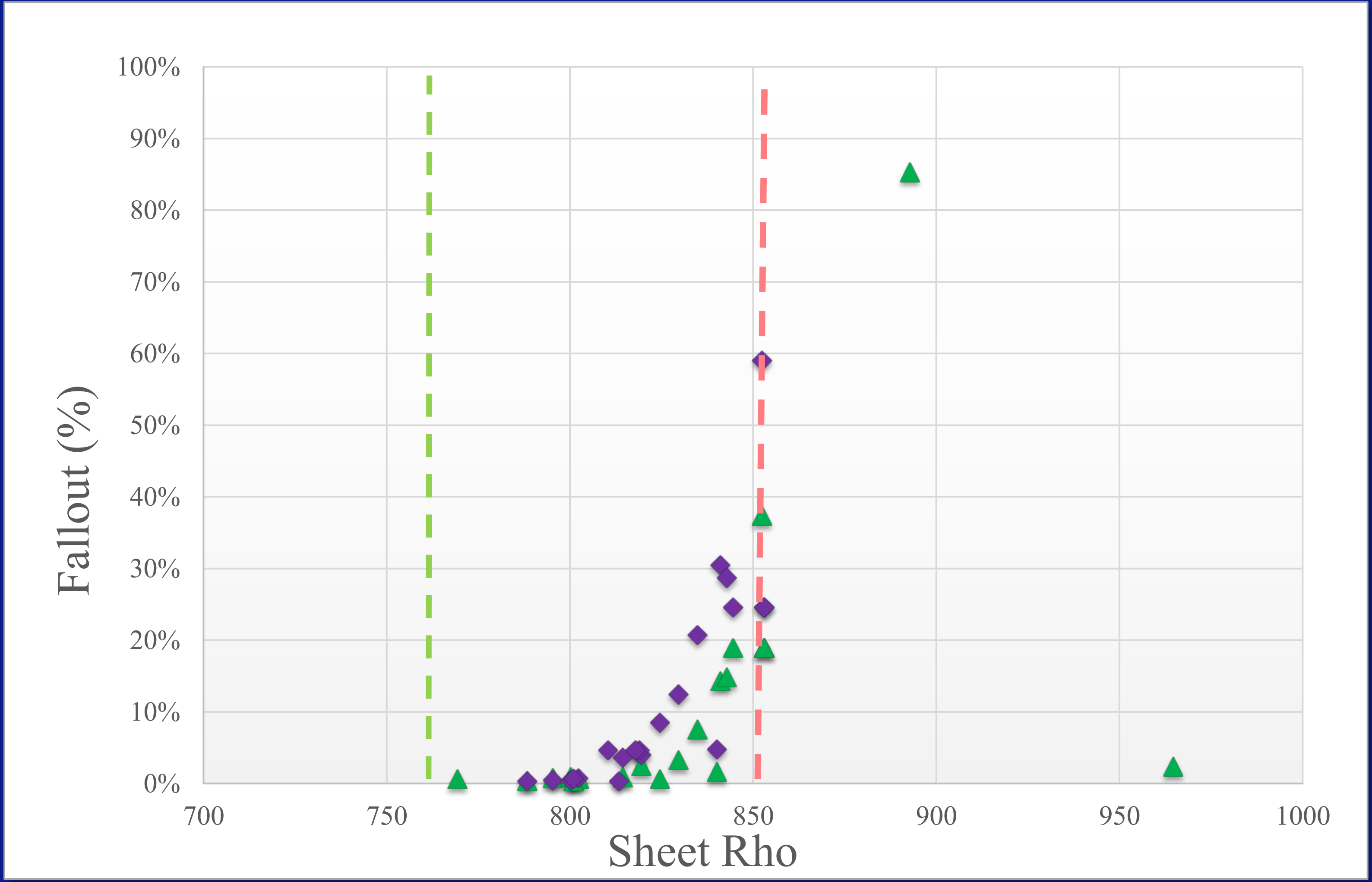
HOW

- ANALYZE AND ARCHITECT
- ISOLATE
- ACCESS
- OBSERVE

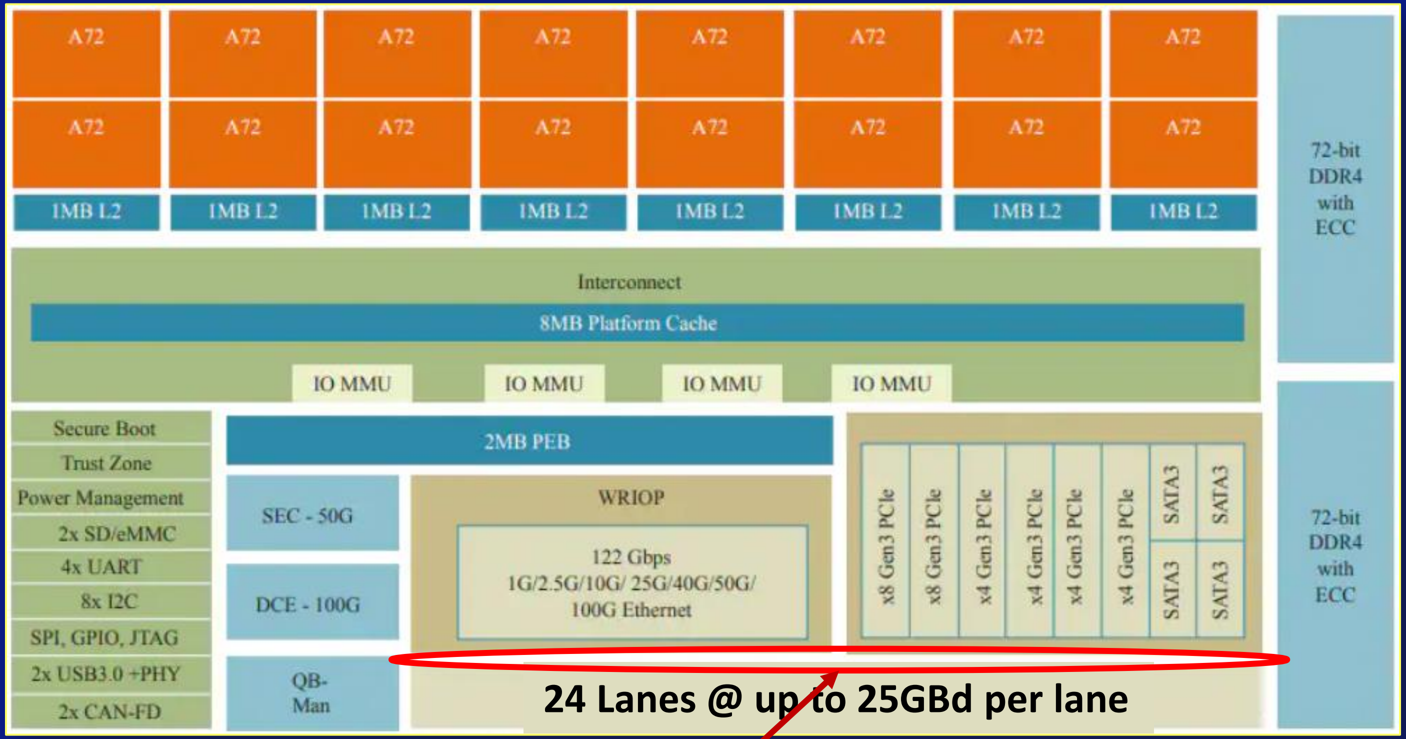
Example



Example (cont.)

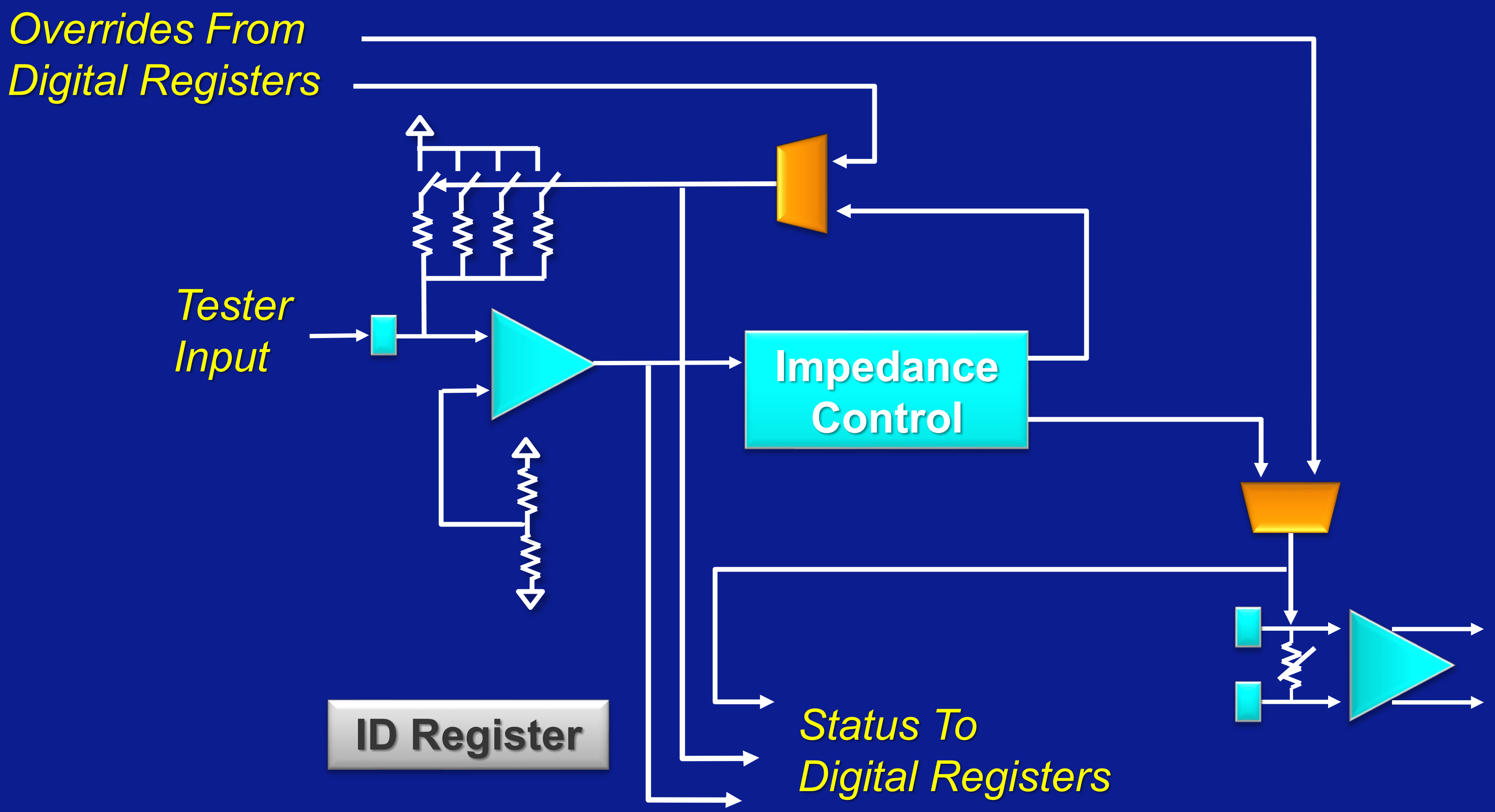


Wrap the Analog

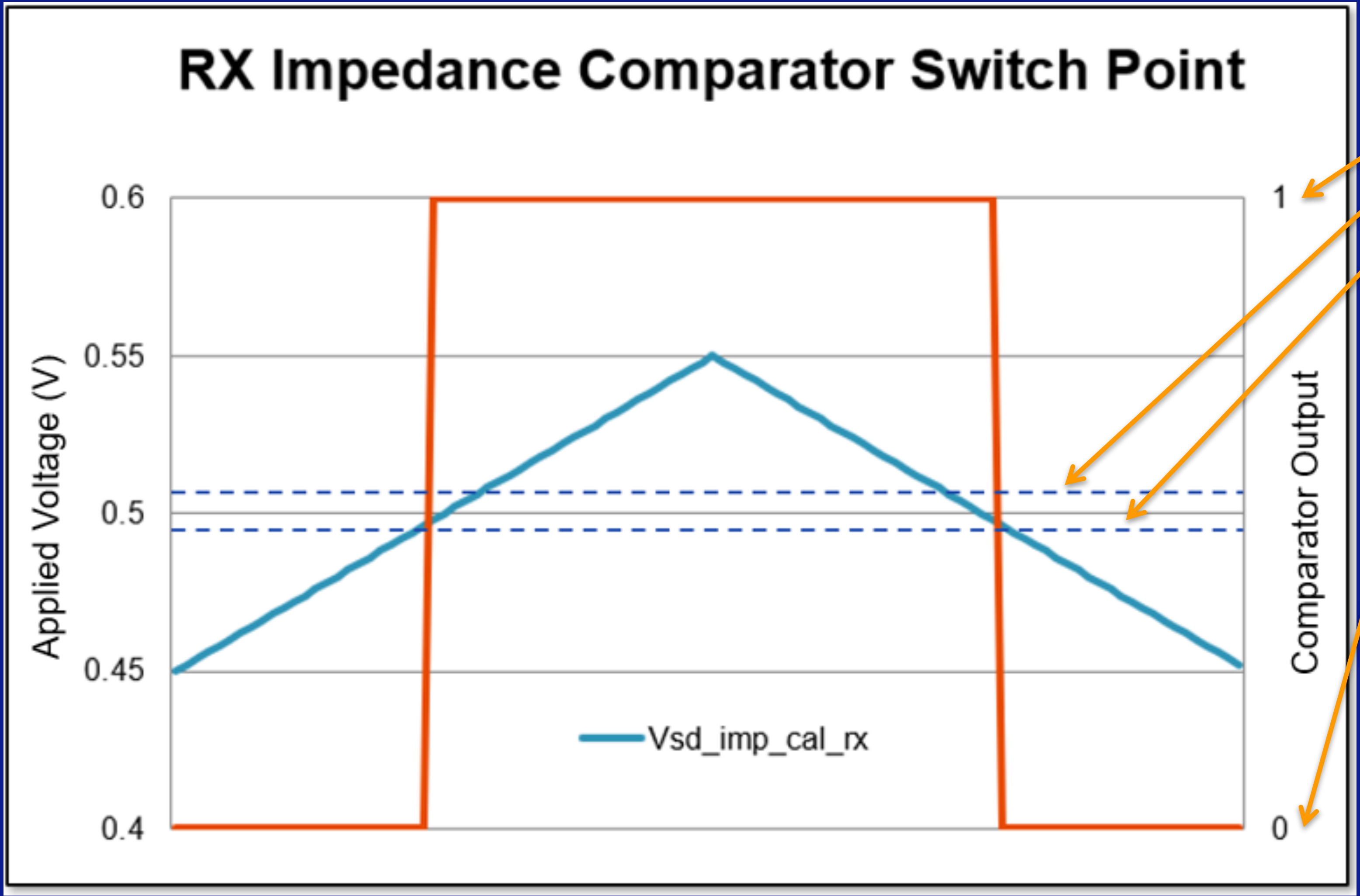


- **Every Analog Design Team should include Digital Design Specialists in:**
  - ✓ RTL Design
  - ✓ Verification
  - ✓ DFT
  - ✓ System Software

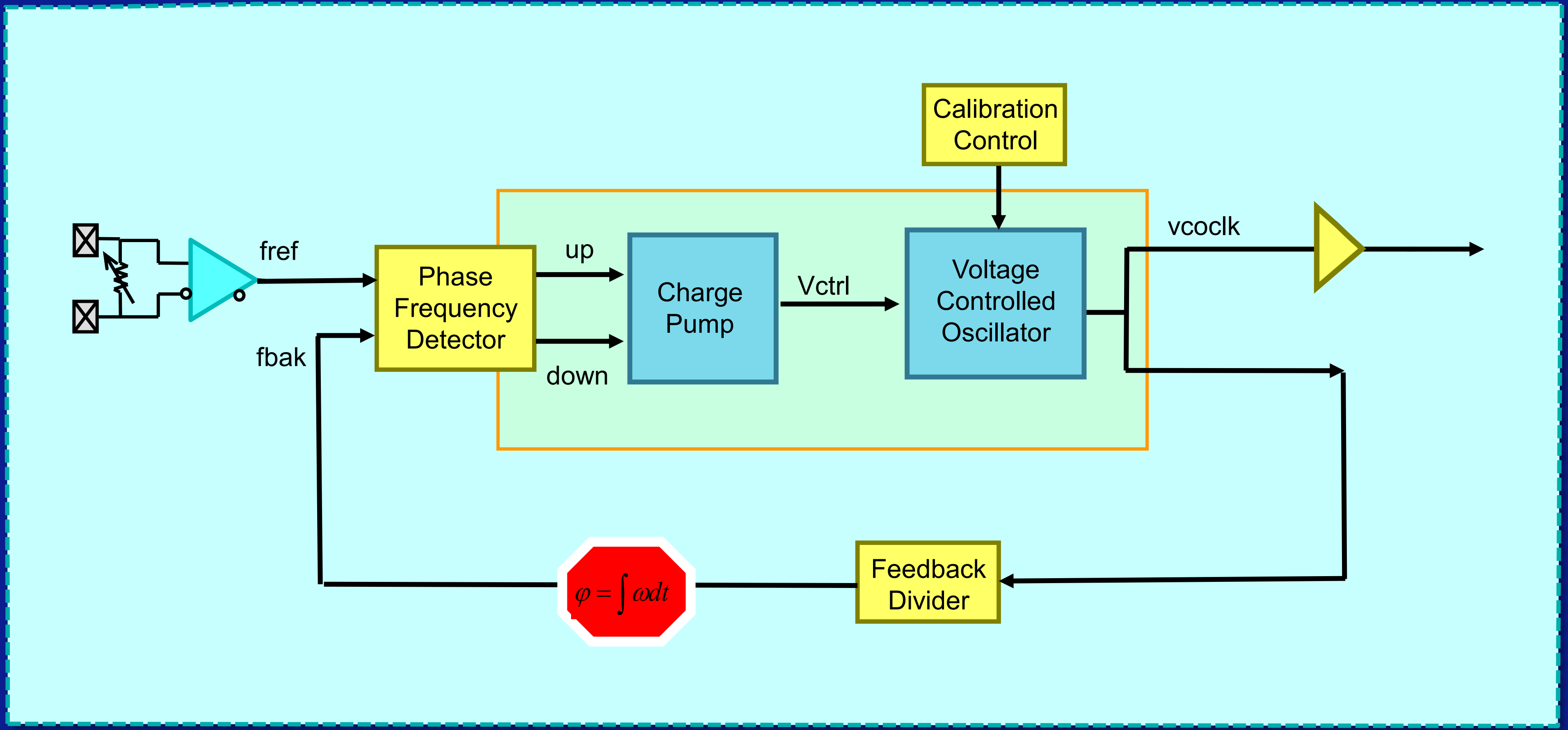
Architecting for Test



Manufacturing Test for Comparators/Resistors

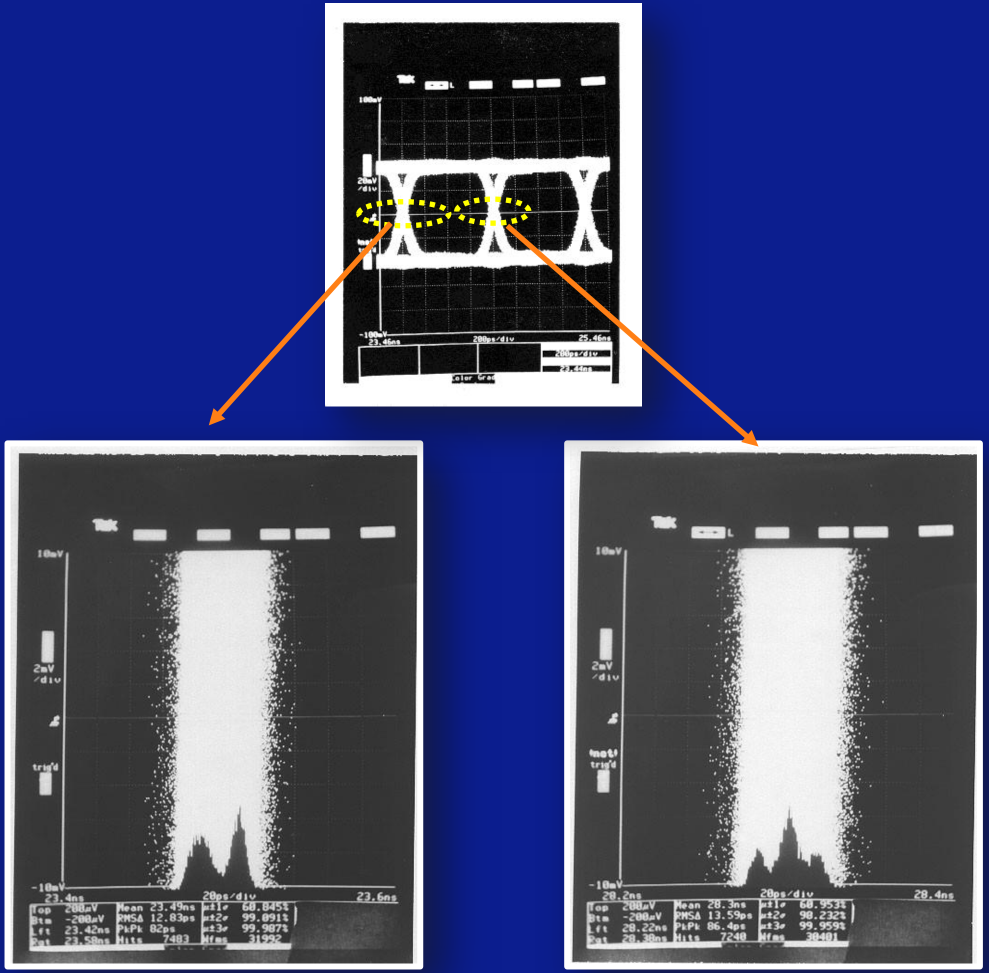


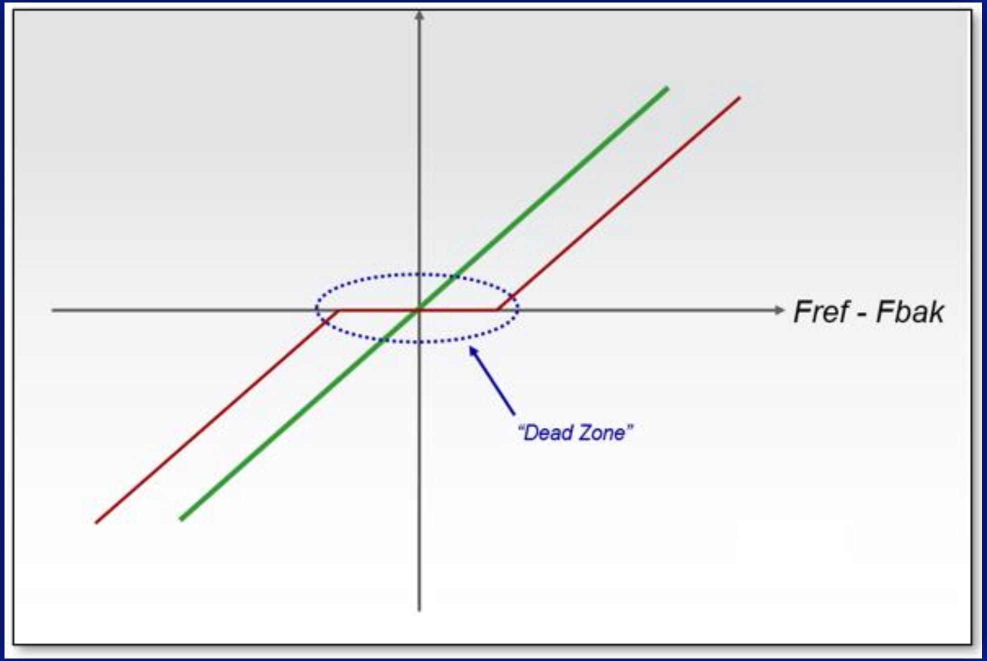
PLL Testing

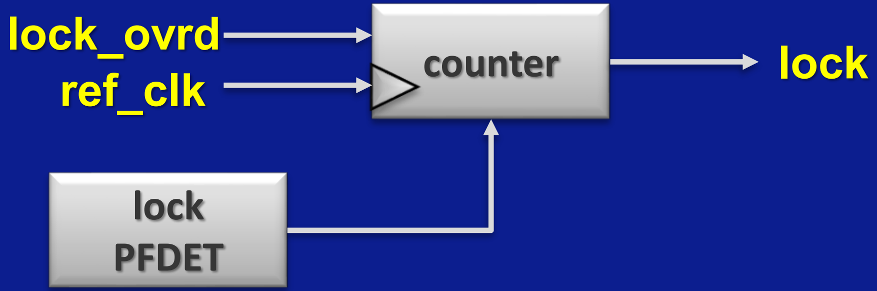


Calibration Control
fref
up
Vctrl
vcoclk
Phase Frequency Detector
Charge Pump
Voltage Controlled Oscillator
fbak
down
$\varphi = \int \omega dt$
Feedback Divider

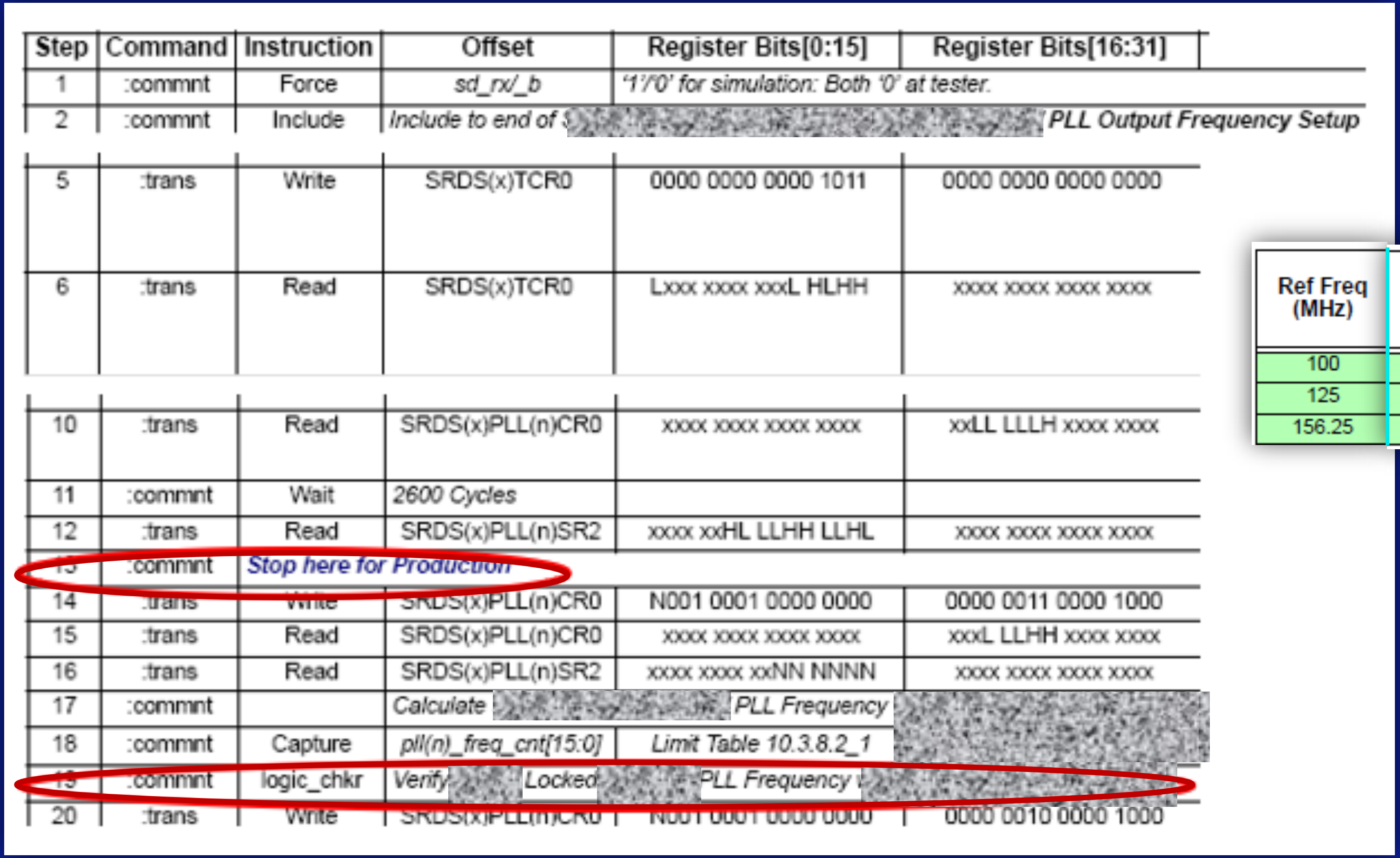
LOCK DETECTION



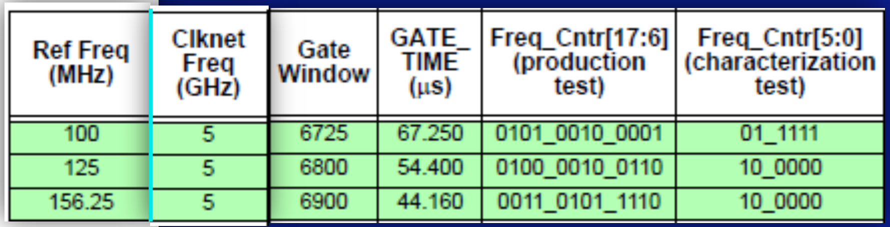




Frequency Counting

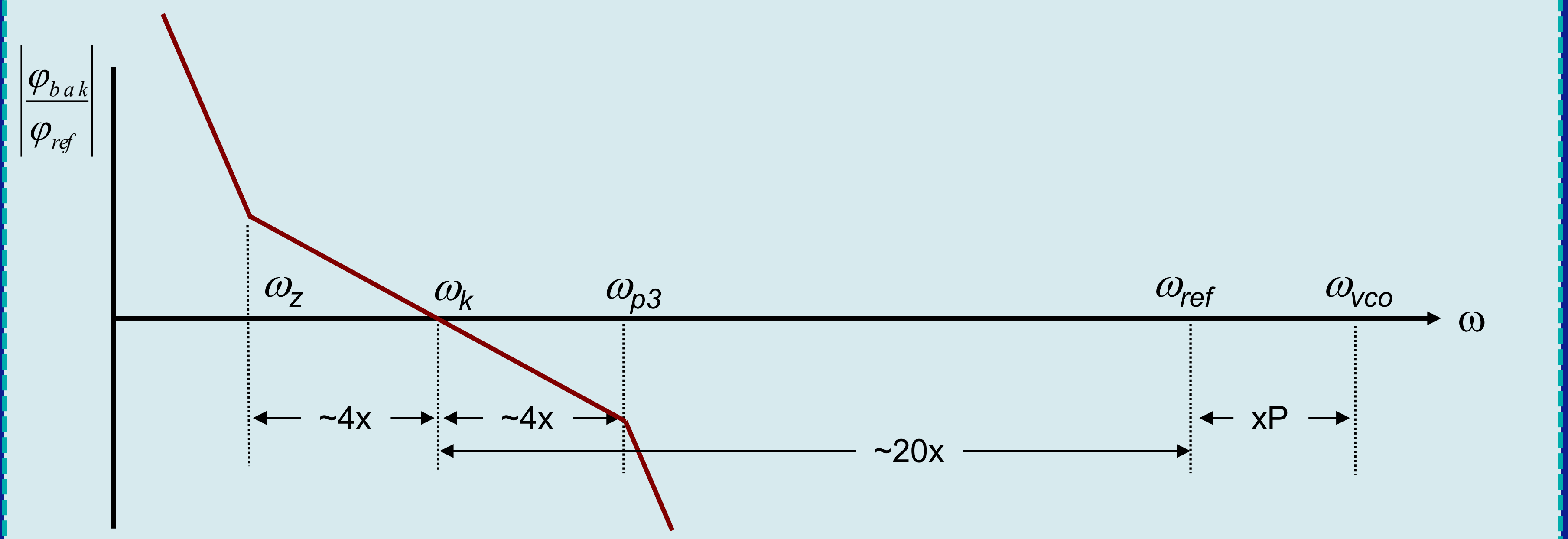


Step	Command	Instruction	Offset	Register Bits[0:15]	Register Bits[16:31]
1	:commnt	Force	*sd_rx/_b*	*'1'/'0' for simulation: Both '0' at tester.*	
2	:commnt	Include	*Include to end of* [redacted] *PLL Output Frequency Setup*		
5	:trans	Write	SRDS(x)TCR0	0000 0000 0000 1011	0000 0000 0000 0000
6	:trans	Read	SRDS(x)TCR0	Lxxx xxxx xxxL HLHH	xxxx xxxx xxxx xxxx
10	:trans	Read	SRDS(x)PLL(n)CR0	xxxx xxxx xxxx xxxx	xxLL LLLH xxxx xxxx
11	:commnt	Wait	*2600 Cycles*		
12	:trans	Read	SRDS(x)PLL(n)SR2	xxxx xxHL LLHH LLHL	xxxx xxxx xxxx xxxx
13	:commnt	***Stop here for Production***			
14	:trans	Write	SRDS(x)PLL(n)CR0	N001 0001 0000 0000	0000 0011 0000 1000
15	:trans	Read	SRDS(x)PLL(n)CR0	xxxx xxxx xxxx xxxx	xxxL LLHH xxxx xxxx
16	:trans	Read	SRDS(x)PLL(n)SR2	xxxx xxxx xxNN NNNN	xxxx xxxx xxxx xxxx
17	:commnt		*Calculate* [redacted] *PLL Frequency*		[redacted]
18	:commnt	Capture	*pll(n)_freq_cnt[15:0]*	*Limit Table 10.3.8.2_1*	[redacted]
19	:commnt	logic_chkr	*Verify* [redacted] *Locked* [redacted] *PLL Frequency* [redacted]		
20	:trans	Write	SRDS(x)PLL(n)CR0	N001 0001 0000 0000	0000 0010 0000 1000



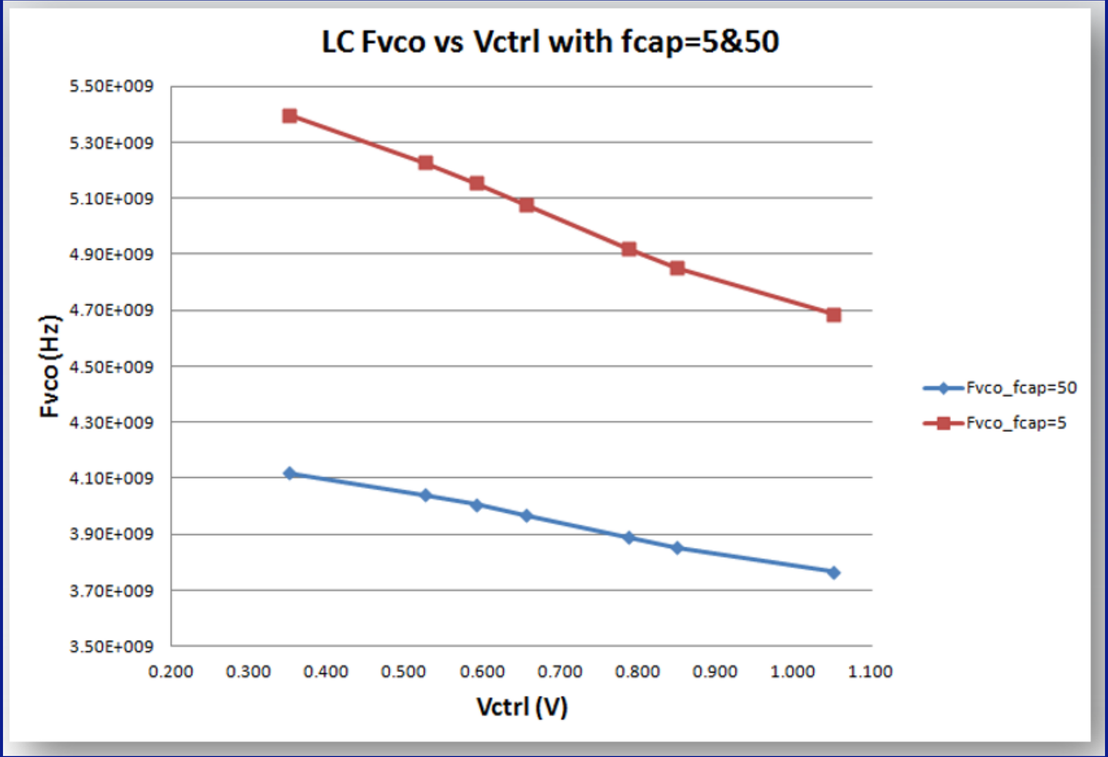
Ref Freq (MHz)	Clknet Freq (GHz)	Gate Window	GATE_ TIME (µs)	Freq_Cntr[17:6] (production test)	Freq_Cntr[5:0] (characterization test)
100	5	6725	67.250	0101_0010_0001	01_1111
125	5	6800	54.400	0100_0010_0110	10_0000
156.25	5	6900	44.160	0011_0101_1110	10_0000

A Low Noise PLL



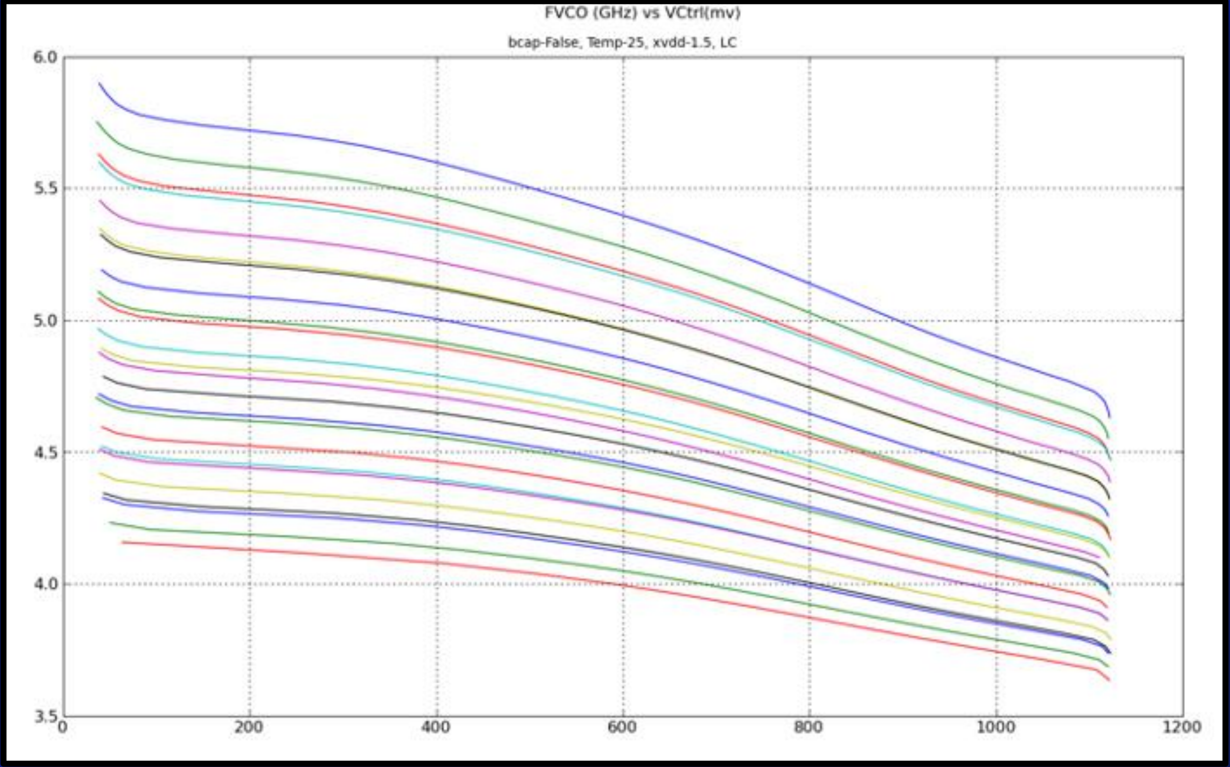
$$\frac{\phi_{bak}}{\phi_{in}} \approx \left(\frac{K_i b}{P}\right)\left[\frac{\left(\frac{I_{\text{int}}}{2\pi b C_{\text{int}}}\right)+s}{s^2}\right]$$

VCO Range of Operation Simulation vs Silicon



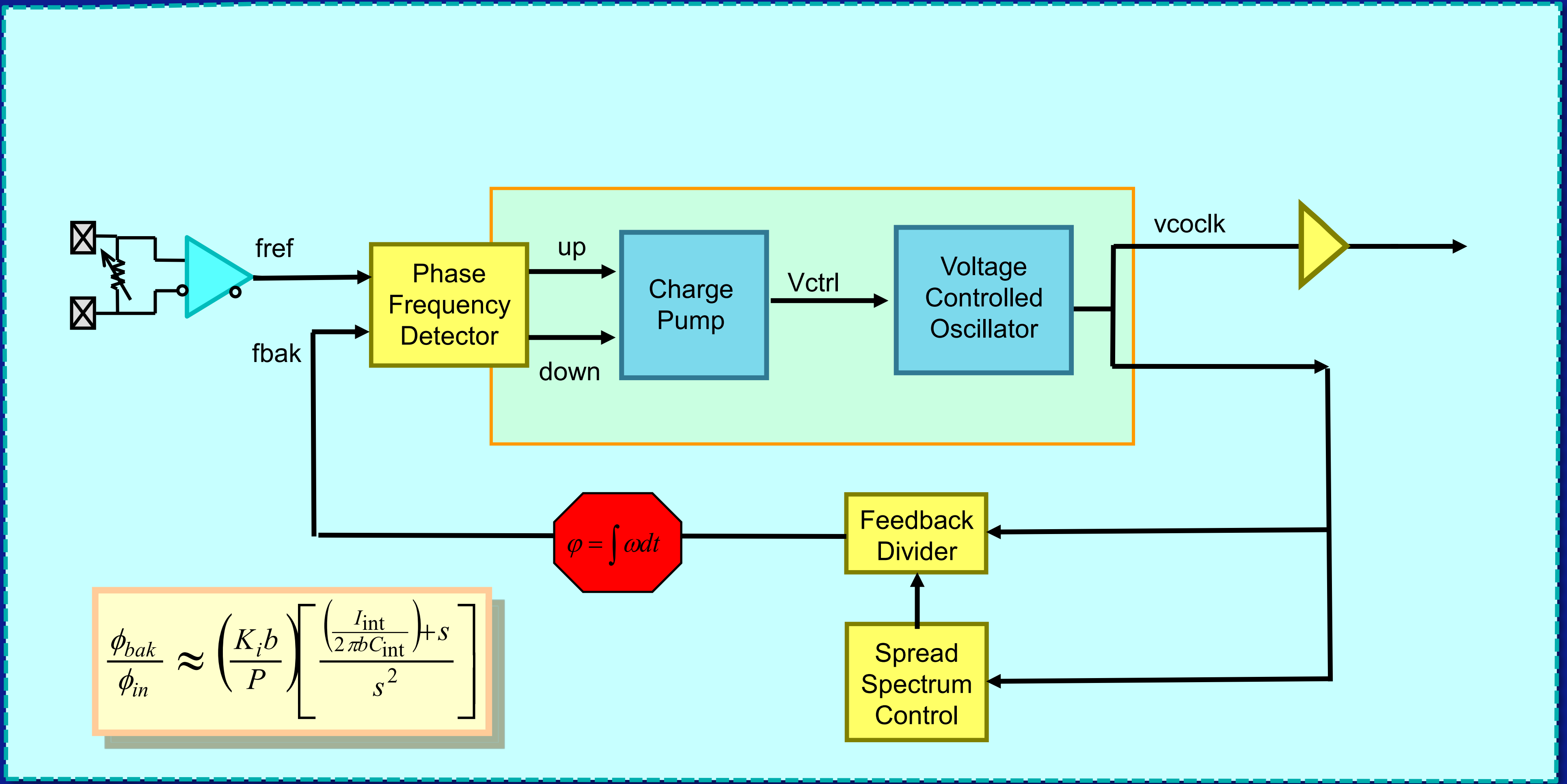
Simulation

Set Fcap
Sweep Fref to Change Fvco
Measure Vctrl



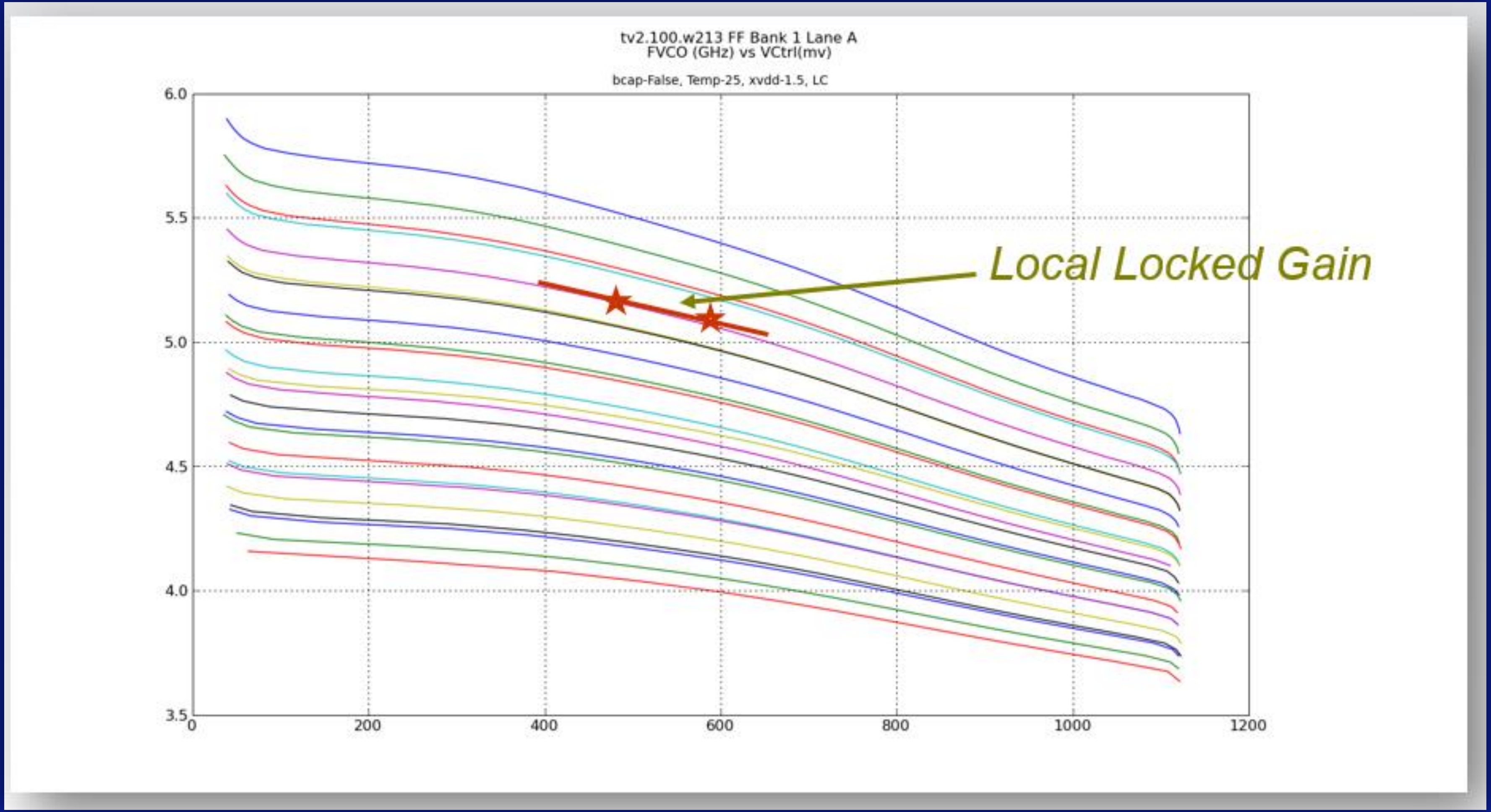
Lab Measurement

PLL Testing

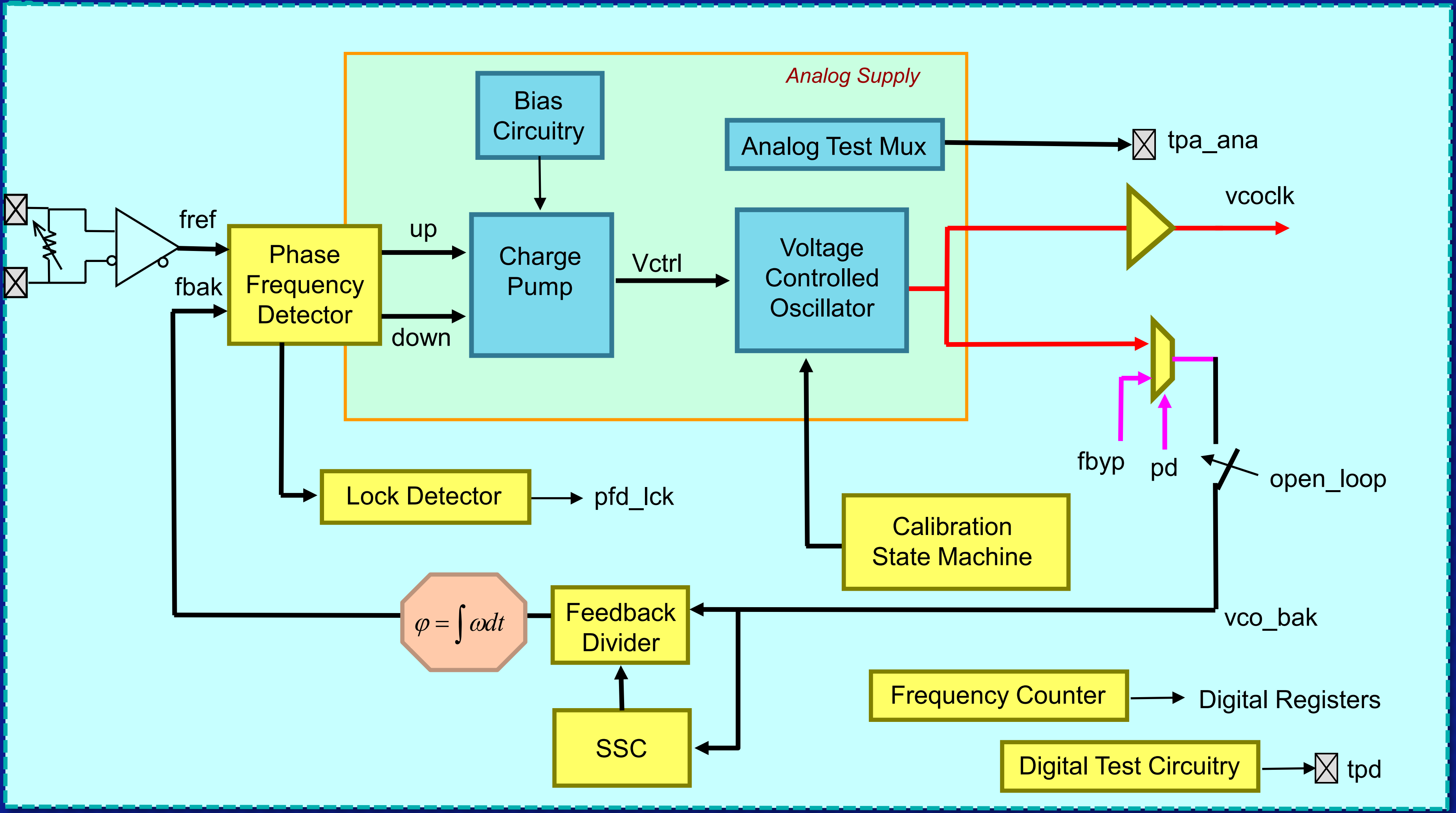


$$\frac{\phi_{bak}}{\phi_{in}} \approx \left(\frac{K_i b}{P}\right)\left[\frac{\left(\frac{I_{\text{int}}}{2\pi b C_{\text{int}}}\right)+s}{s^2}\right]$$

Localized Gain Tests on the Tester

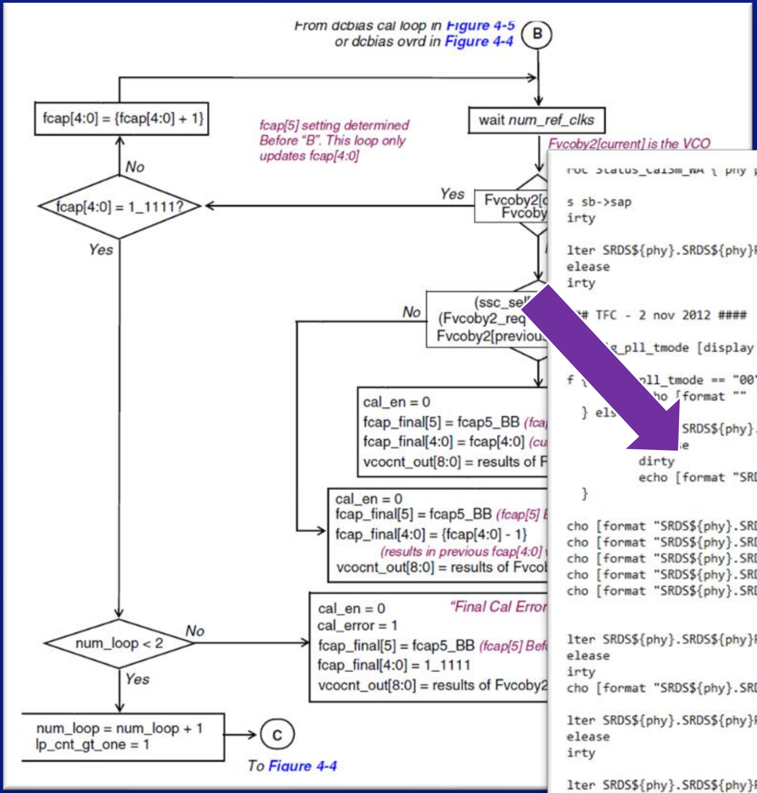


PLL Testing

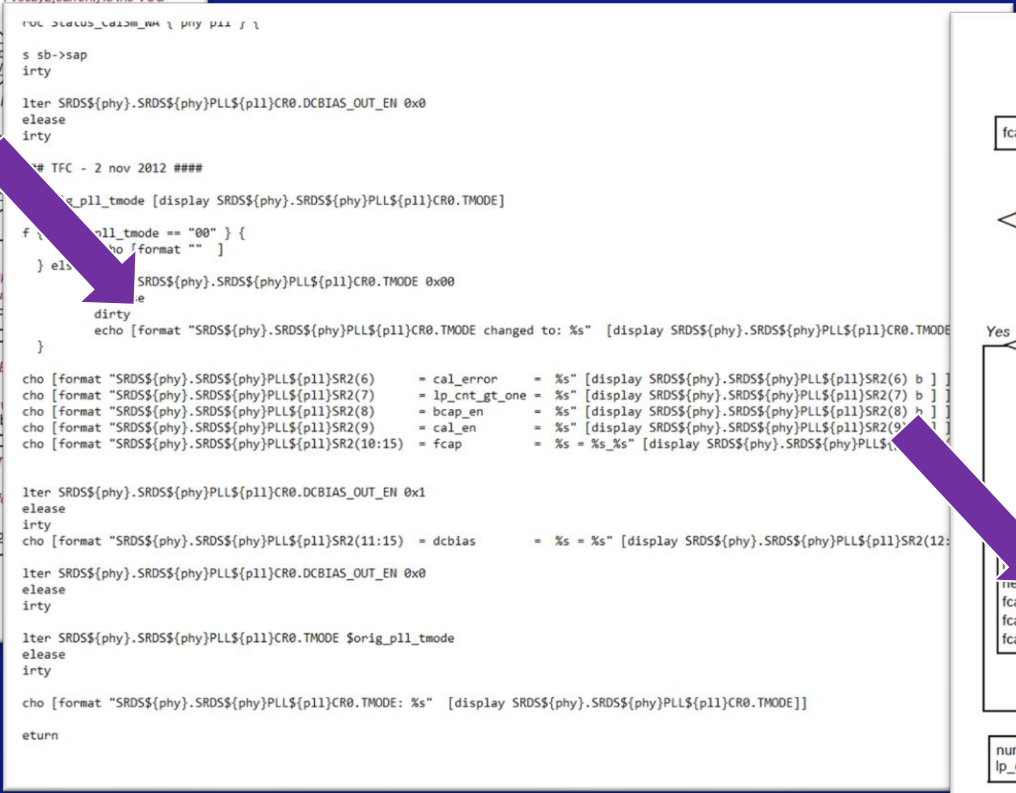


- Lock Observation
- Lock Override
- Bypass
- Frequency Counter
- Fmin/Fmax/Recover
- Calibration Overrides
- Calibration Status
- Open Loop
- Control Voltage
- Gain Tests
- Voltage Regulators
- Current Bias
- Digital Logic Testing

Calibration Status and Overrides

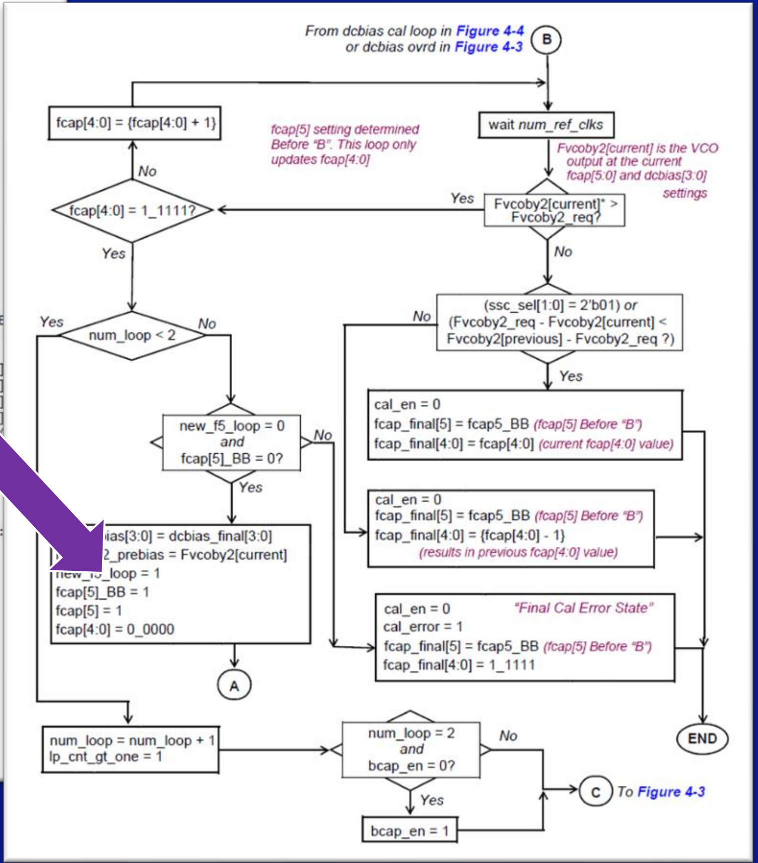

From dcbias cal loop in Figure 4-5
or dcbias ovrd in Figure 4-4
B
fcap[4:0] = {fcap[4:0] + 1}
fcap[5] setting determined Before "B". This loop only updates fcap[4:0]
wait num_ref_clks
Fvcoby2[current] is the VCO
No
fcap[4:0] = 1_1111?
Yes
Yes
No
cal_en = 0
fcap_final[5] = fcap5_BB
fcap_final[4:0] = fcap[4:0]
vcocnt_out[8:0] = results of F
cal_en = 0
fcap_final[5] = fcap5_BB
fcap_final[4:0] = {fcap[4:0] - 1}
(results in previous fcap[4:0]
vcocnt_out[8:0] = results of Fvcol
"Final Cal Error
cal_en = 0
cal_error = 1
fcap_final[5] = fcap5_BB
fcap_final[4:0] = 1_1111
vcocnt_out[8:0] = results of Fvcoby2
num_loop < 2
No
Yes
num_loop = num_loop + 1
lp_cnt_gt_one = 1
C
To Figure 4-4

Original Logic



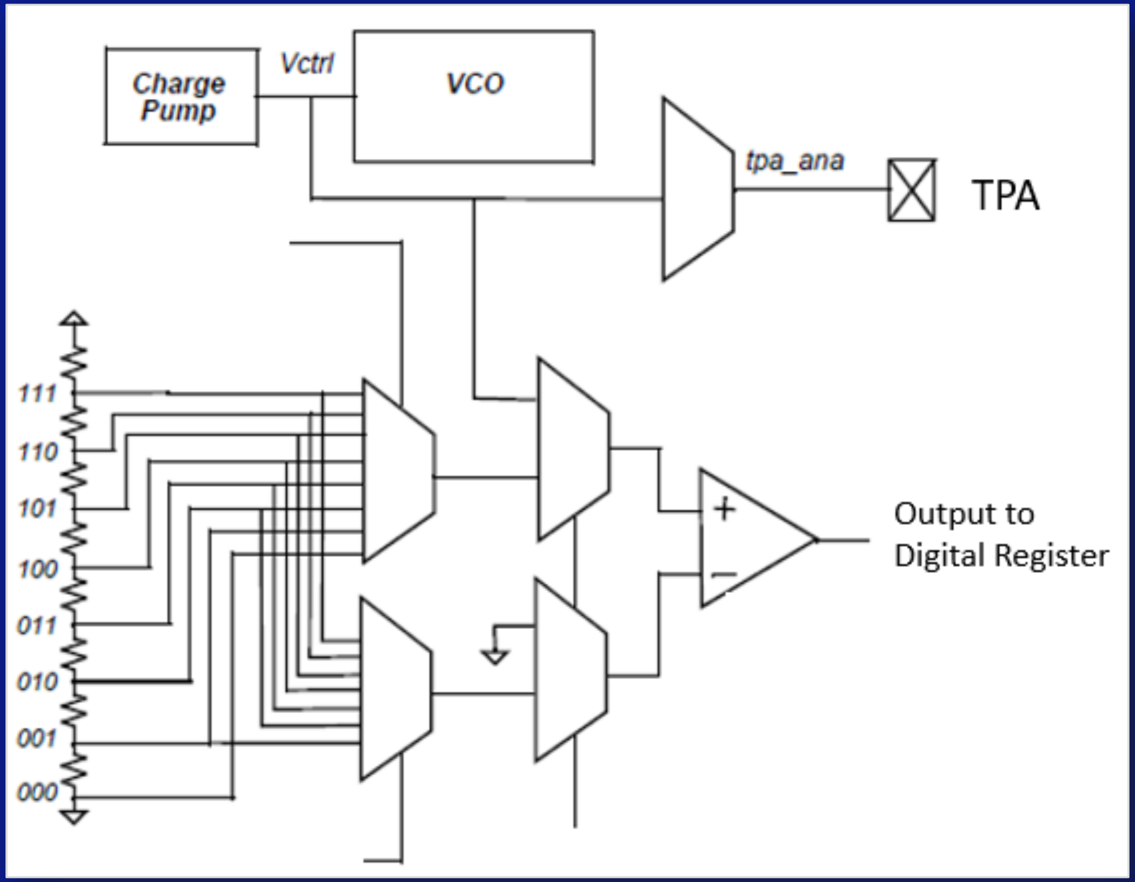
```
roc Status_CalSM_WA { phy pll } {

s sb->sap
irty

lter SRDS${phy}.SRDS${phy}PLL${pll}CR0.DCBIAS_OUT_EN 0x0
elease
irty

## TFC - 2 nov 2012 ####

  g_pll_tmode [display SRDS${phy}.SRDS${phy}PLL${pll}CR0.TMODE]

f {      pll_tmode == "00" } {
          ho [format ""  ]
  } els
          SRDS${phy}.SRDS${phy}PLL${pll}CR0.TMODE 0x00
          e
        dirty
        echo [format "SRDS${phy}.SRDS${phy}PLL${pll}CR0.TMODE changed to: %s"  [display SRDS${phy}.SRDS${phy}PLL${pll}CR0.TMODE
  }

cho [format "SRDS${phy}.SRDS${phy}PLL${pll}SR2(6)      = cal_error     =  %s" [display SRDS${phy}.SRDS${phy}PLL${pll}SR2(6) b ] ]
cho [format "SRDS${phy}.SRDS${phy}PLL${pll}SR2(7)      = lp_cnt_gt_one =  %s" [display SRDS${phy}.SRDS${phy}PLL${pll}SR2(7) b ] ]
cho [format "SRDS${phy}.SRDS${phy}PLL${pll}SR2(8)      = bcap_en       =  %s" [display SRDS${phy}.SRDS${phy}PLL${pll}SR2(8) b ] ]
cho [format "SRDS${phy}.SRDS${phy}PLL${pll}SR2(9)      = cal_en        =  %s" [display SRDS${phy}.SRDS${phy}PLL${pll}SR2(9
cho [format "SRDS${phy}.SRDS${phy}PLL${pll}SR2(10:15)  = fcap          =  %s = %s_%s" [display SRDS${phy}.SRDS${phy}PLL${


lter SRDS${phy}.SRDS${phy}PLL${pll}CR0.DCBIAS_OUT_EN 0x1
elease
irty
cho [format "SRDS${phy}.SRDS${phy}PLL${pll}SR2(11:15)  = dcbias        =  %s = %s" [display SRDS${phy}.SRDS${phy}PLL${pll}SR2(12:

lter SRDS${phy}.SRDS${phy}PLL${pll}CR0.DCBIAS_OUT_EN 0x0
elease
irty

lter SRDS${phy}.SRDS${phy}PLL${pll}CR0.TMODE $orig_pll_tmode
elease
irty

cho [format "SRDS${phy}.SRDS${phy}PLL${pll}CR0.TMODE: %s"  [display SRDS${phy}.SRDS${phy}PLL${pll}CR0.TMODE]]

eturn
```

Coded Errata

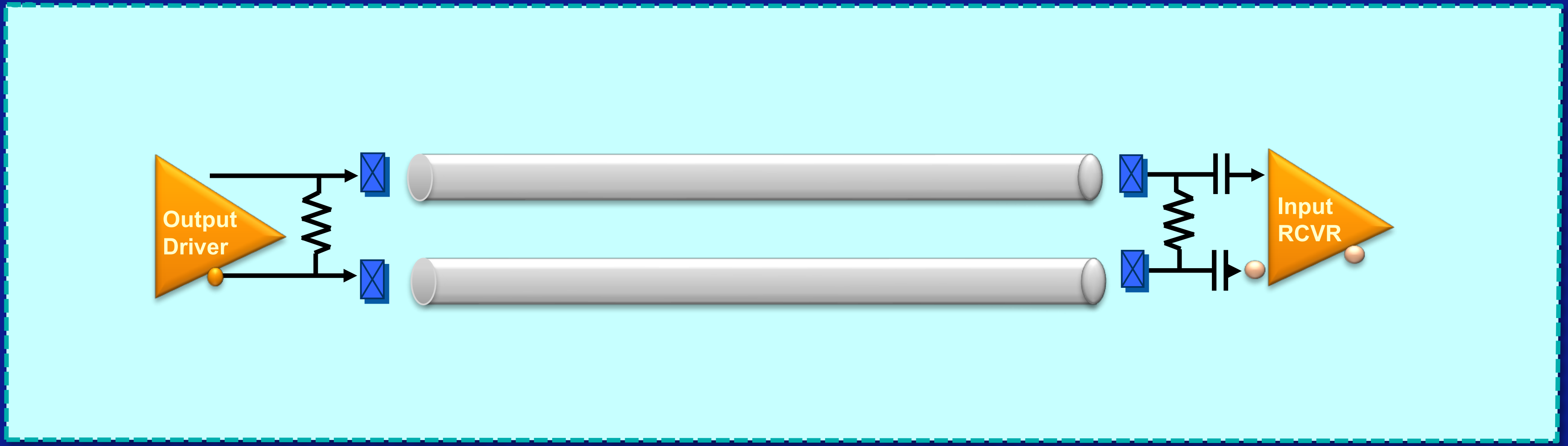

From dcbias cal loop in Figure 4-4
or dcbias ovrd in Figure 4-3
B
fcap[4:0] = {fcap[4:0] + 1}
fcap[5] setting determined Before "B". This loop only updates fcap[4:0]
wait num_ref_clks
Fvcoby2[current] is the VCO output at the current fcap[5:0] and dcbias[3:0] settings
No
fcap[4:0] = 1_1111?
Yes
Fvcoby2[current]* > Fvcoby2_req?
Yes
No
Yes
num_loop < 2
No
No
(ssc_sel[1:0] = 2'b01) or (Fvcoby2_req - Fvcoby2[current] < Fvcoby2[previous] - Fvcoby2_req ?)
Yes
cal_en = 0
fcap_final[5] = fcap5_BB (fcap[5] Before "B")
fcap_final[4:0] = fcap[4:0] (current fcap[4:0] value)
new_f5_loop = 0 and fcap[5]_BB = 0?
No
Yes
cal_en = 0
fcap_final[5] = fcap5_BB (fcap[5] Before "B")
fcap_final[4:0] = {fcap[4:0] - 1}
(results in previous fcap[4:0] value)
ias[3:0] = dcbias_final[3:0]
2_prebias = Fvcoby2[current]
new_f5_loop = 1
fcap[5]_BB = 1
fcap[5] = 1
fcap[4:0] = 0_0000
"Final Cal Error State"
cal_en = 0
cal_error = 1
fcap_final[5] = fcap5_BB (fcap[5] Before "B")
fcap_final[4:0] = 1_1111
A
num_loop = num_loop + 1
lp_cnt_gt_one = 1
num_loop = 2 and bcap_en = 0?
No
Yes
bcap_en = 1
END
C
To Figure 4-3

Revised Logic

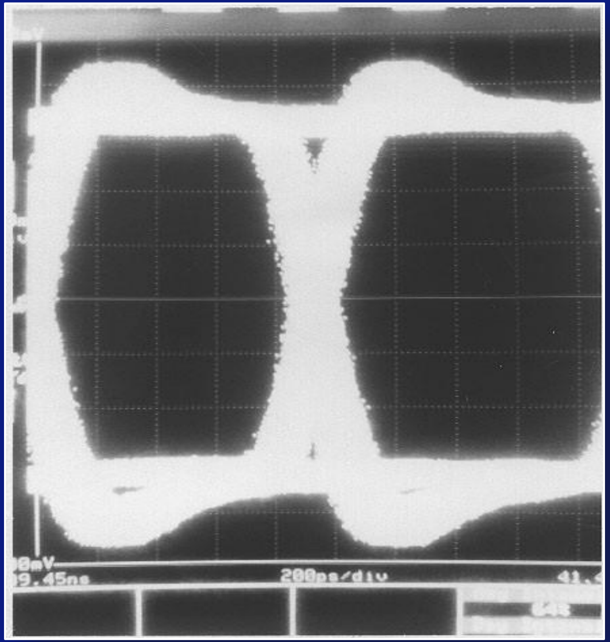
Converting To Digital Result

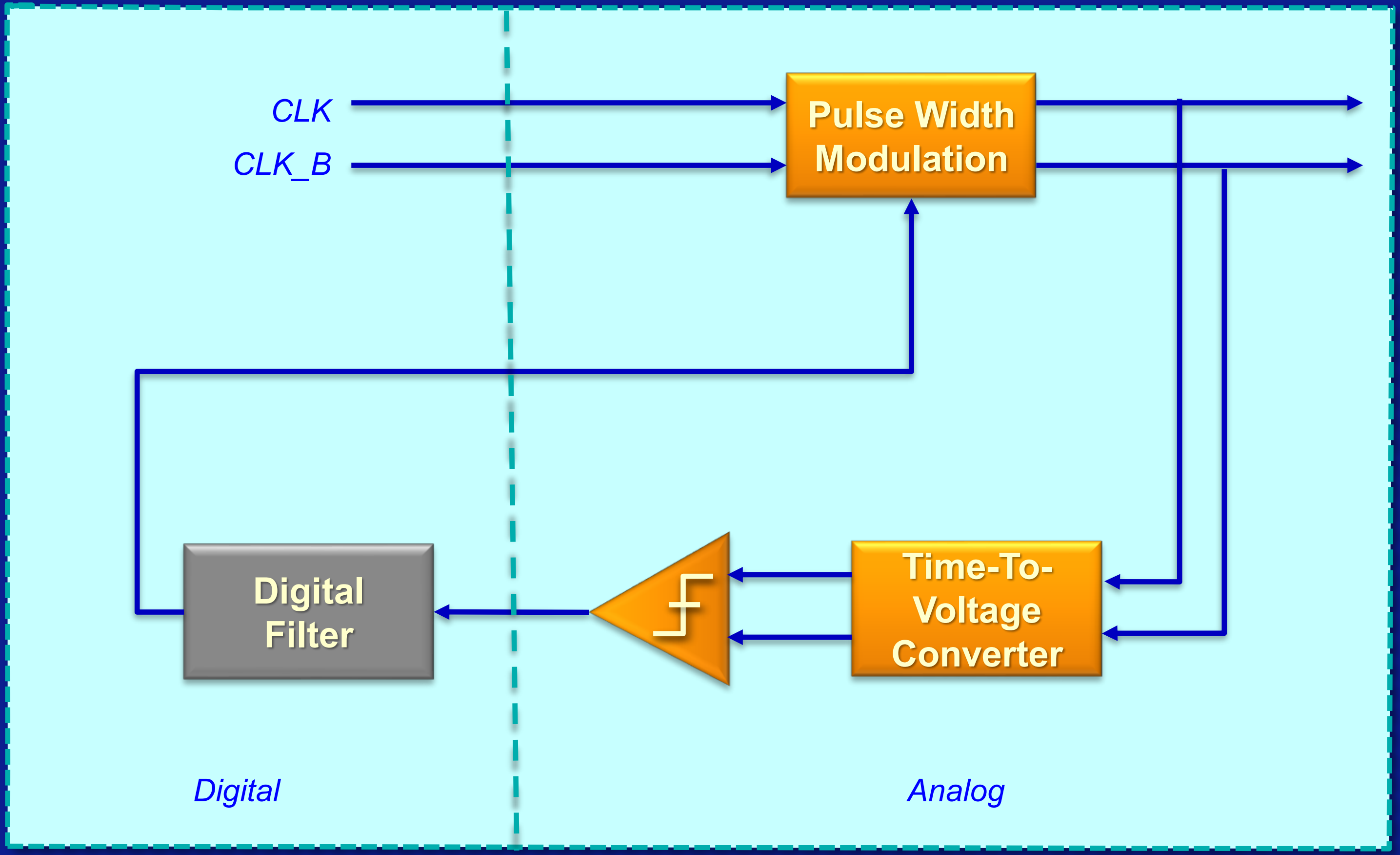


High Speed I/O

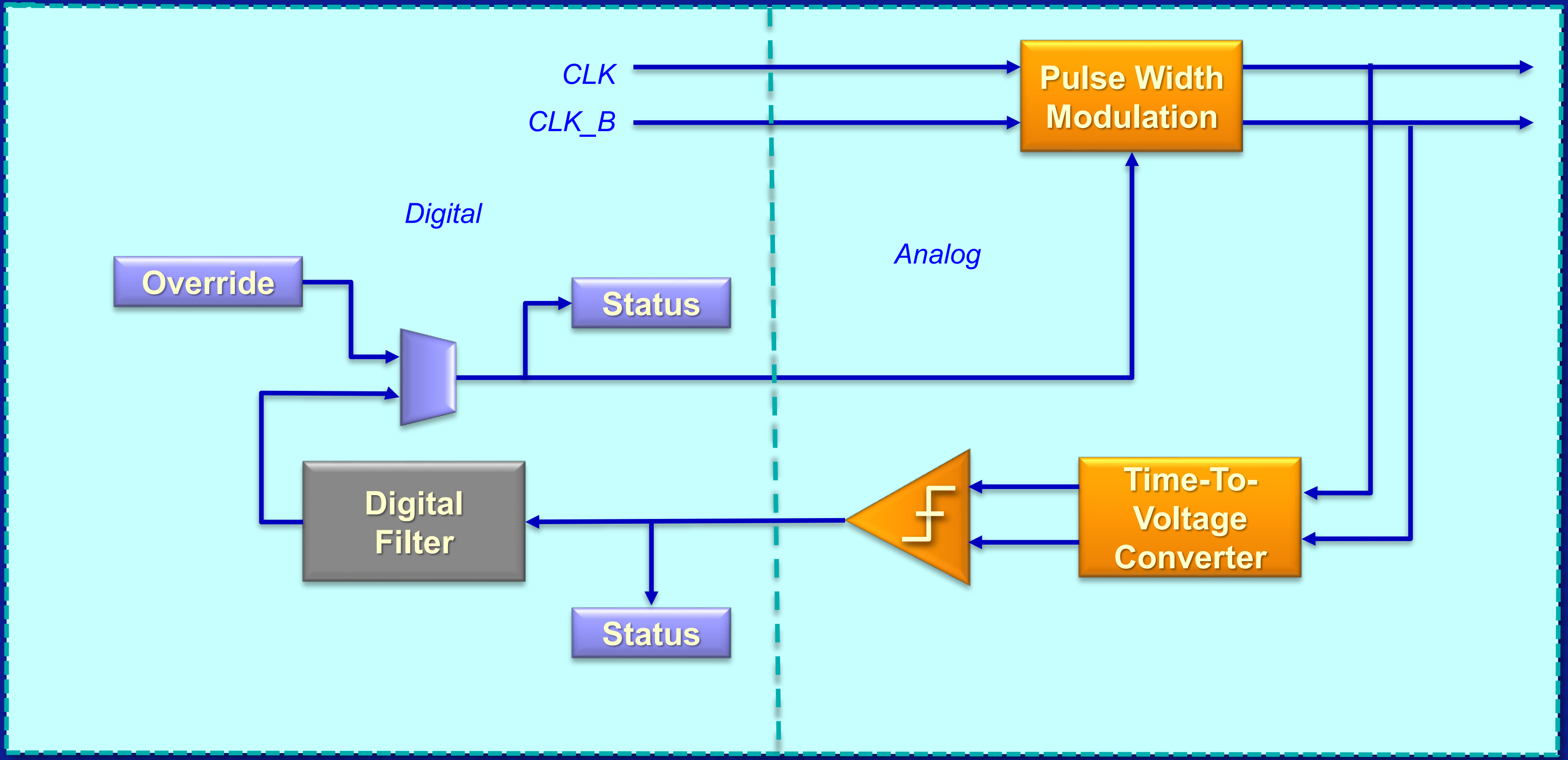


Duty Cycle Correction

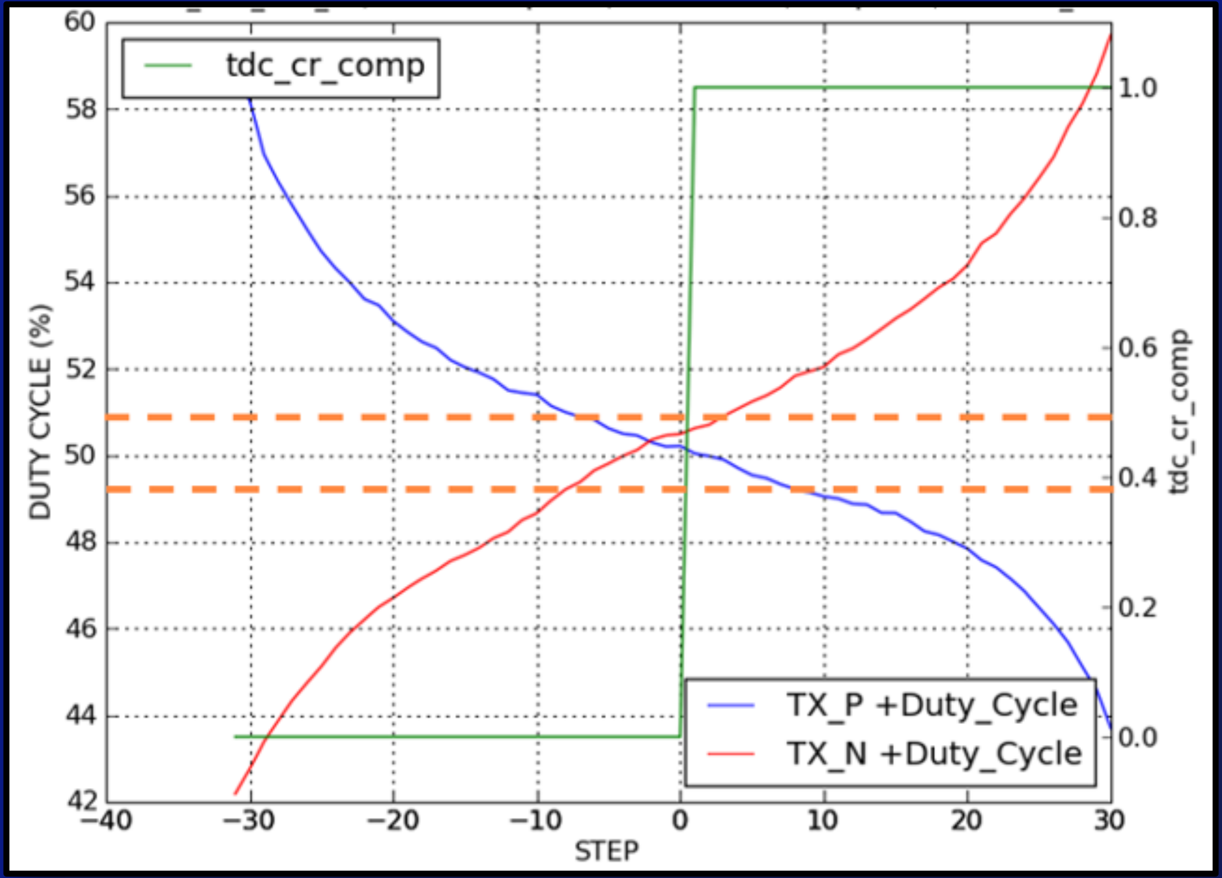




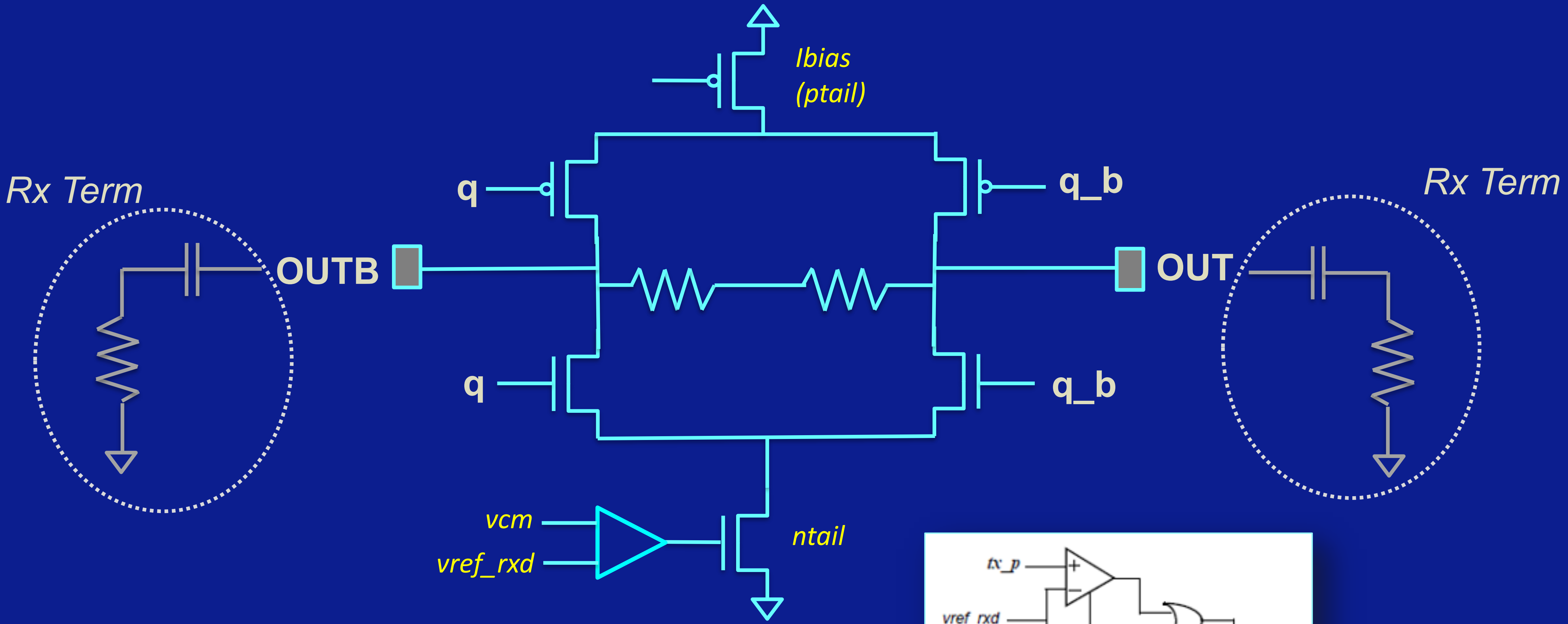
Duty Cycle Correction

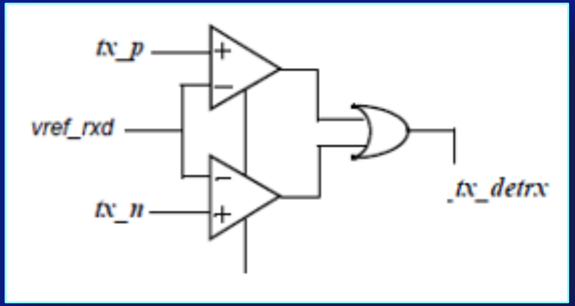


Duty Cycle Correction Testing

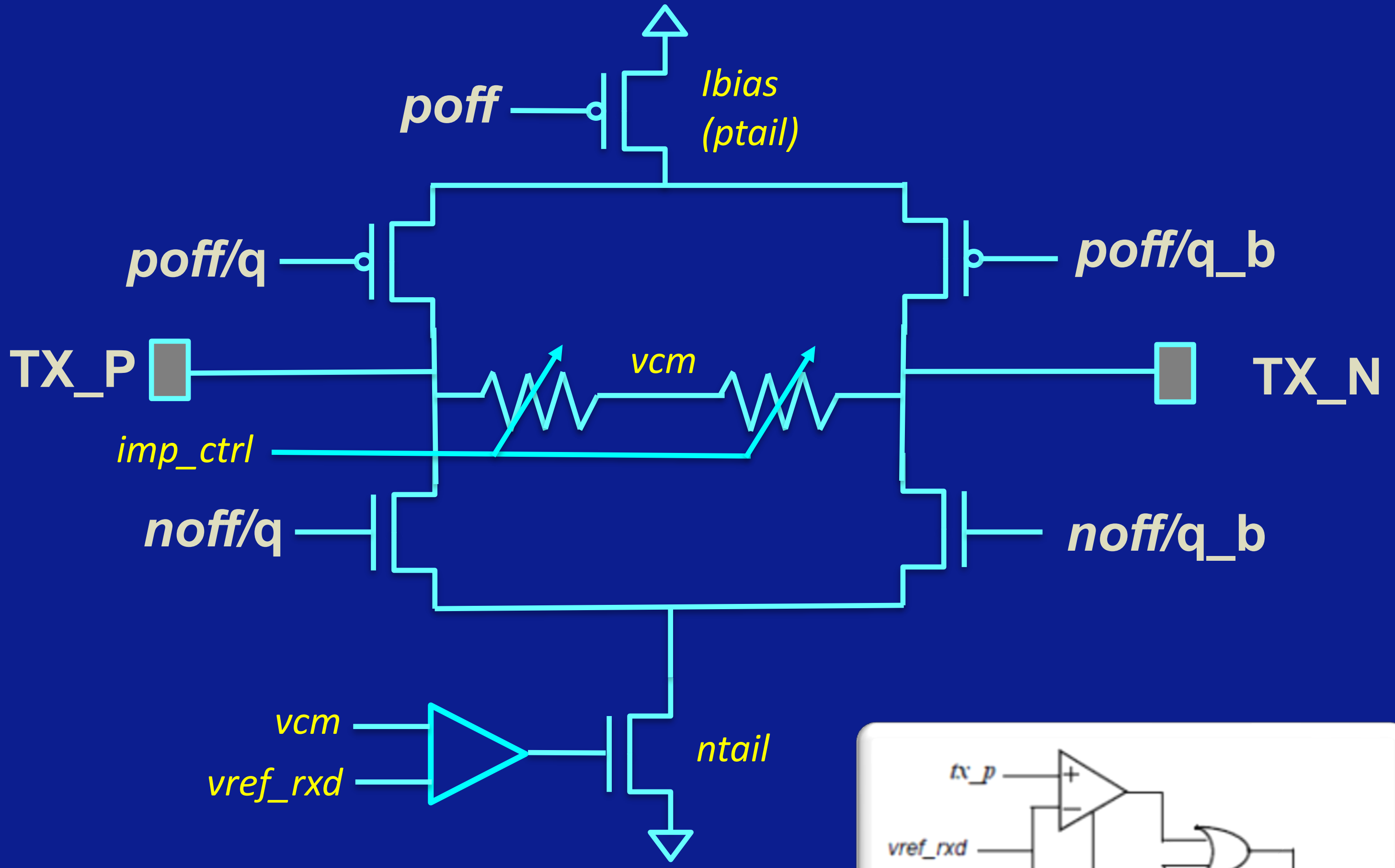


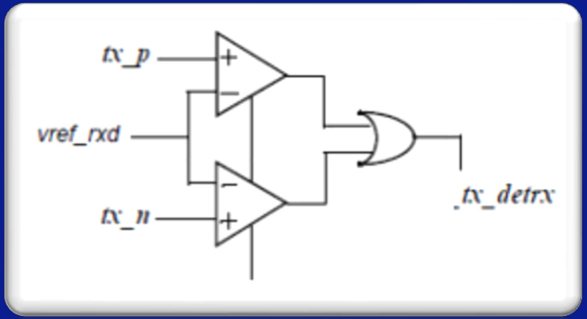
Transmitter Example





Transmitter Tests

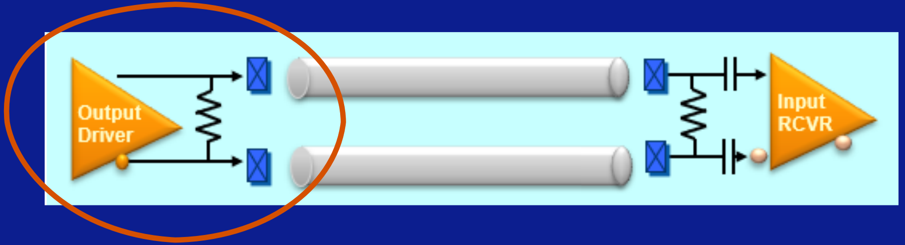


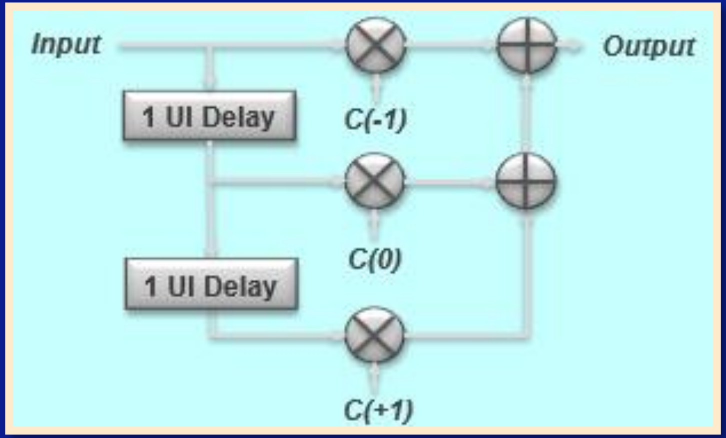


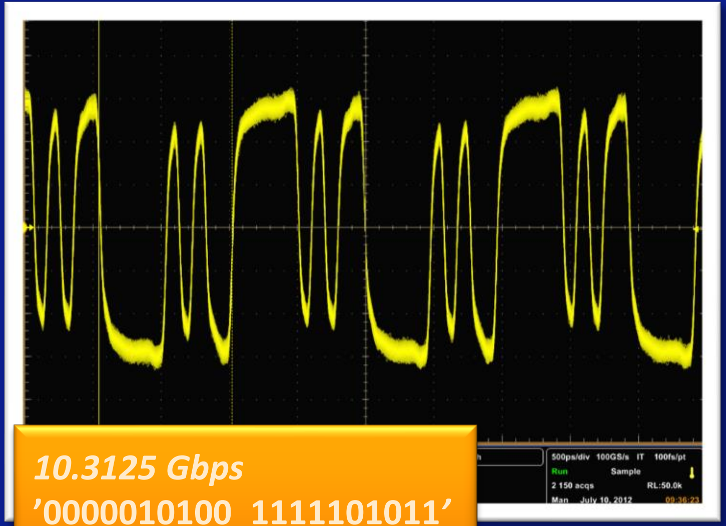
- Output Impedance Measurements
- DC Output Amplitudes and Ratios
- RX Detect Comparator Threshold
- Output Voltage Self Test
- Common Mode

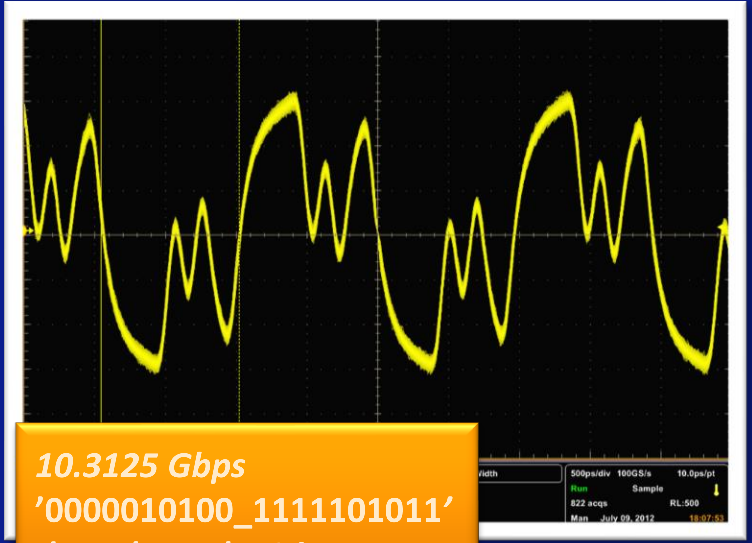
Test_Ctrl[1:0]	Comparator Threshold
2'b00	n/a
2'b01	Vref_1
2'b10	Vref_2
2'b11	Vref_3

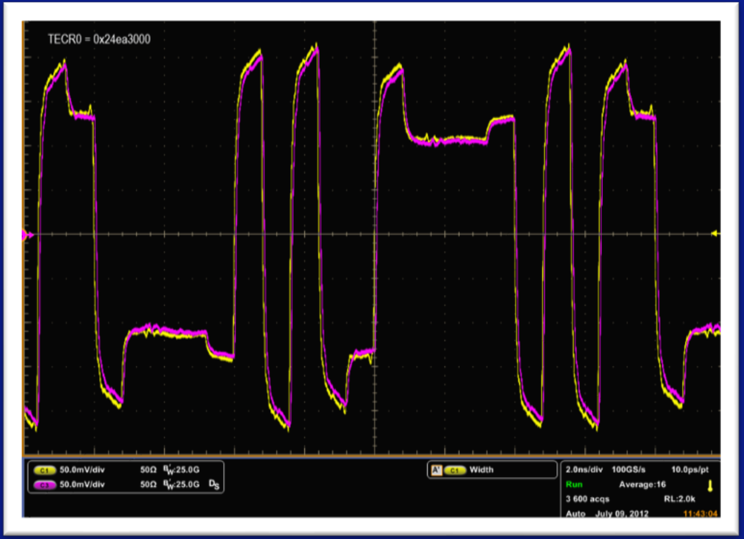
Adapting Functions for Test





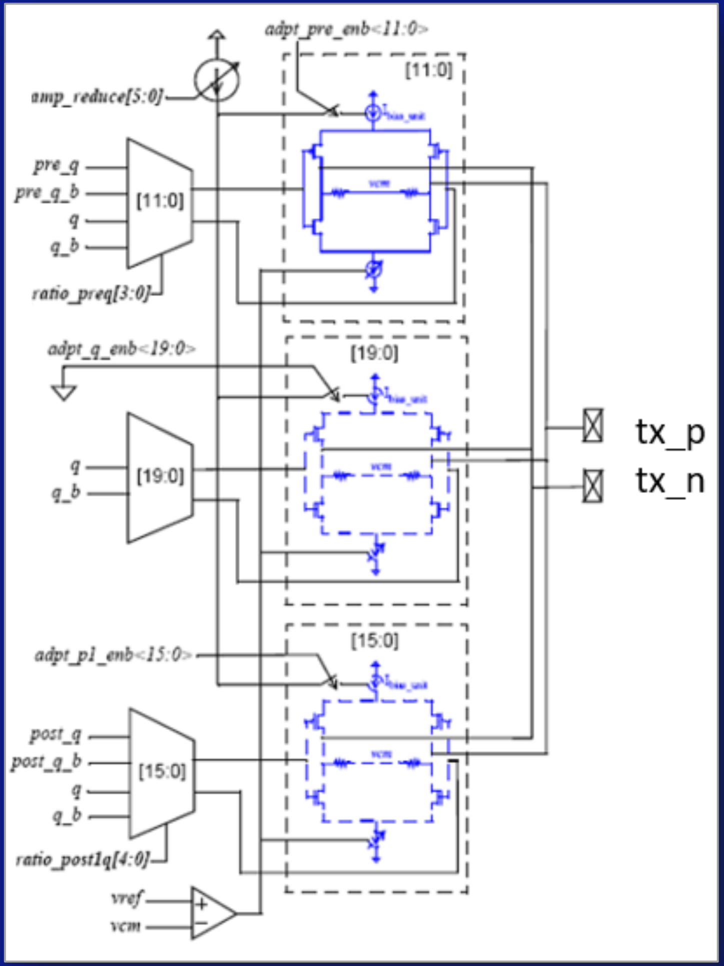

10.3125 Gbps
'0000010100_1111101011'
data through 9 in FR4

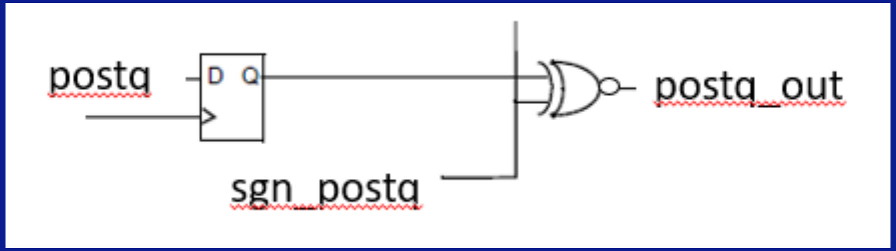

10.3125 Gbps
'0000010100_1111101011'
data through 40 in FR4

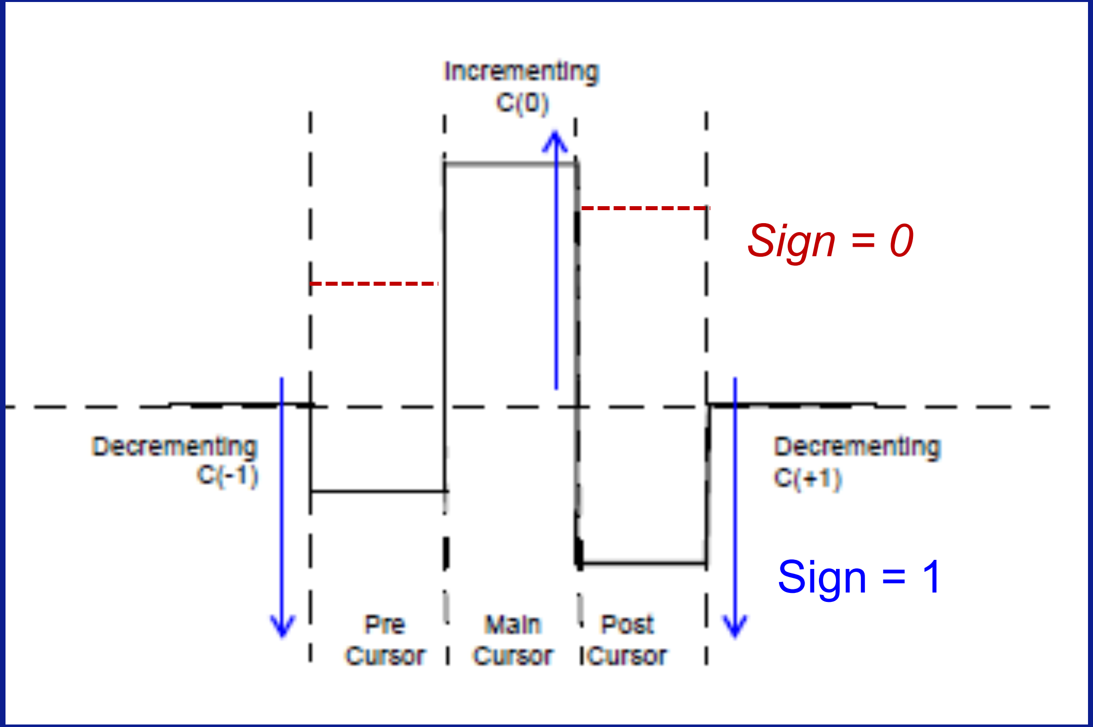


Ref: IEEE802.3^{TM} Clause 72

Transmitter Equalization Testing



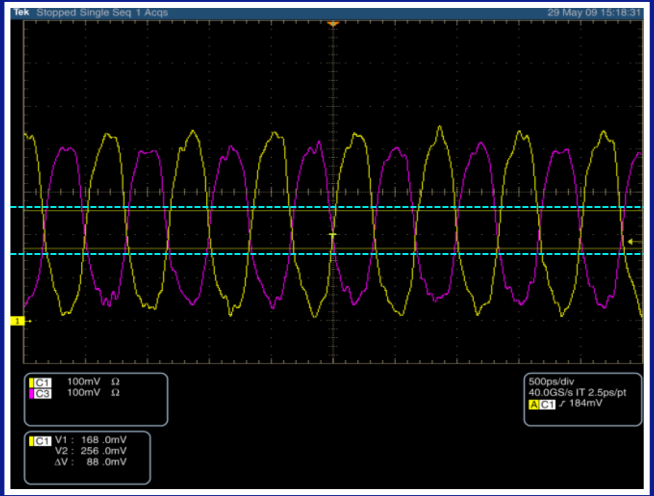




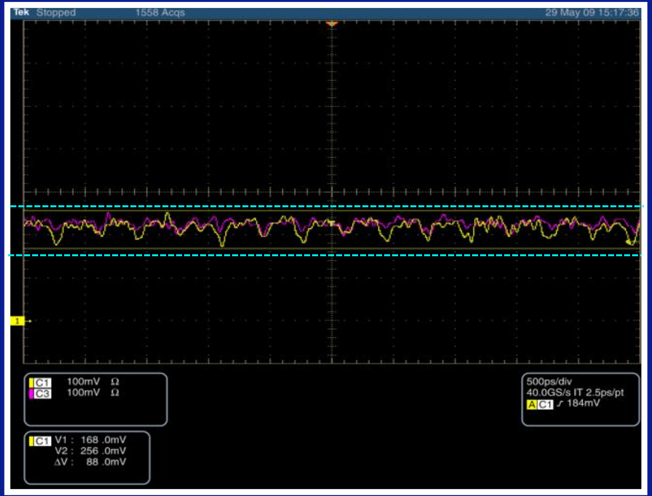
App Note: NXP AN5119

Transmitter At-Speed Cursor Tests

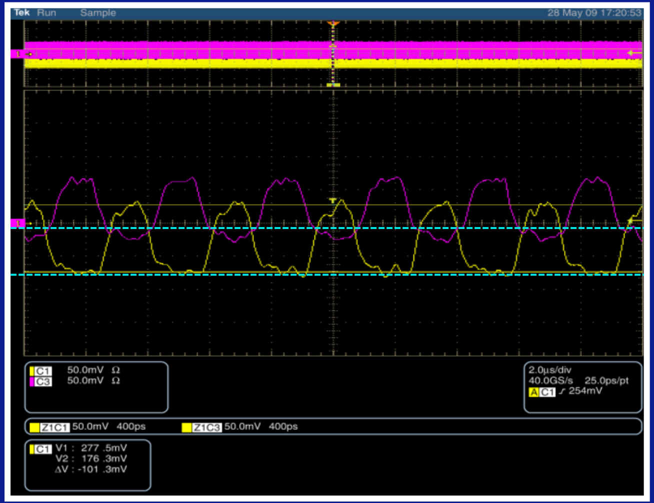
Sign = 1



Sign = 0



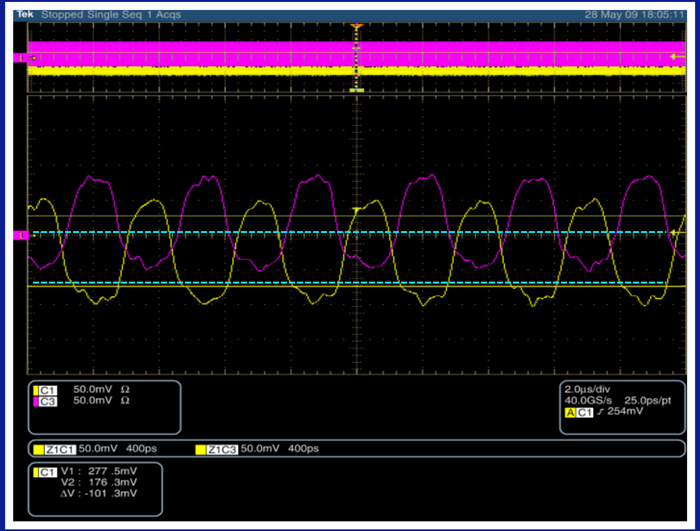
Sign bit broken



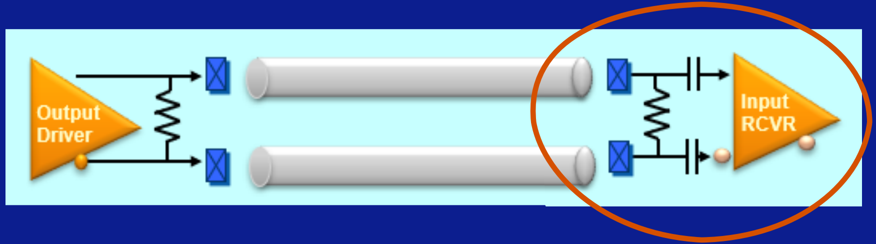
Stuck

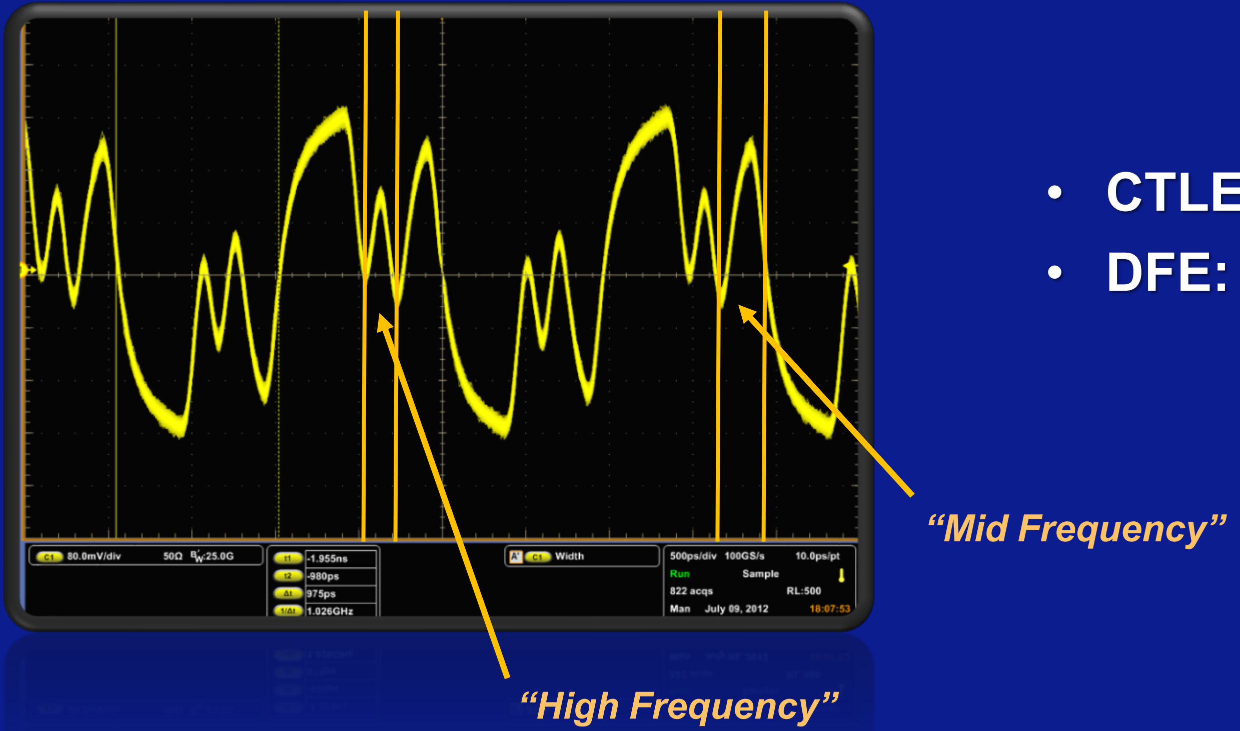
Weighting

MSB



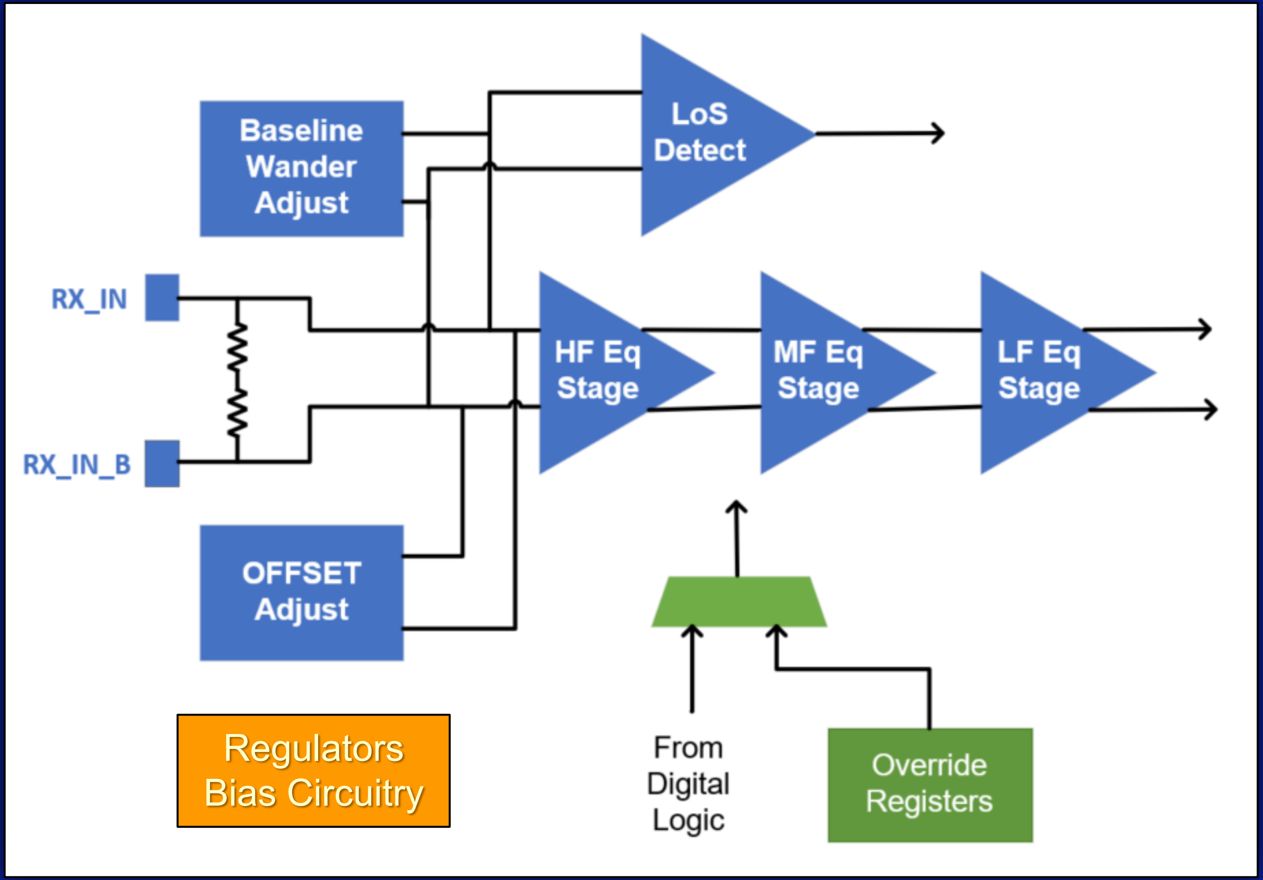
Receive Equalization

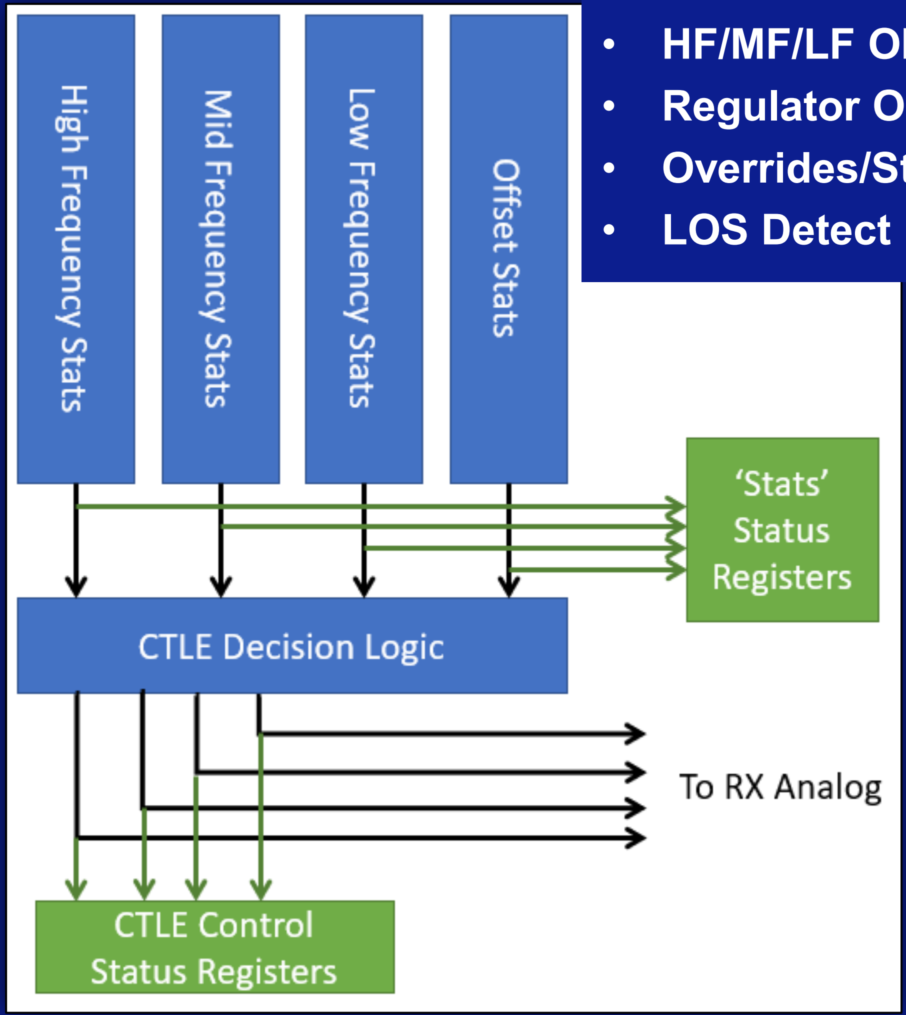




- **CTLE: Continuous Time Linear Equalization**
- **DFE: Decision Feedback Equalization**

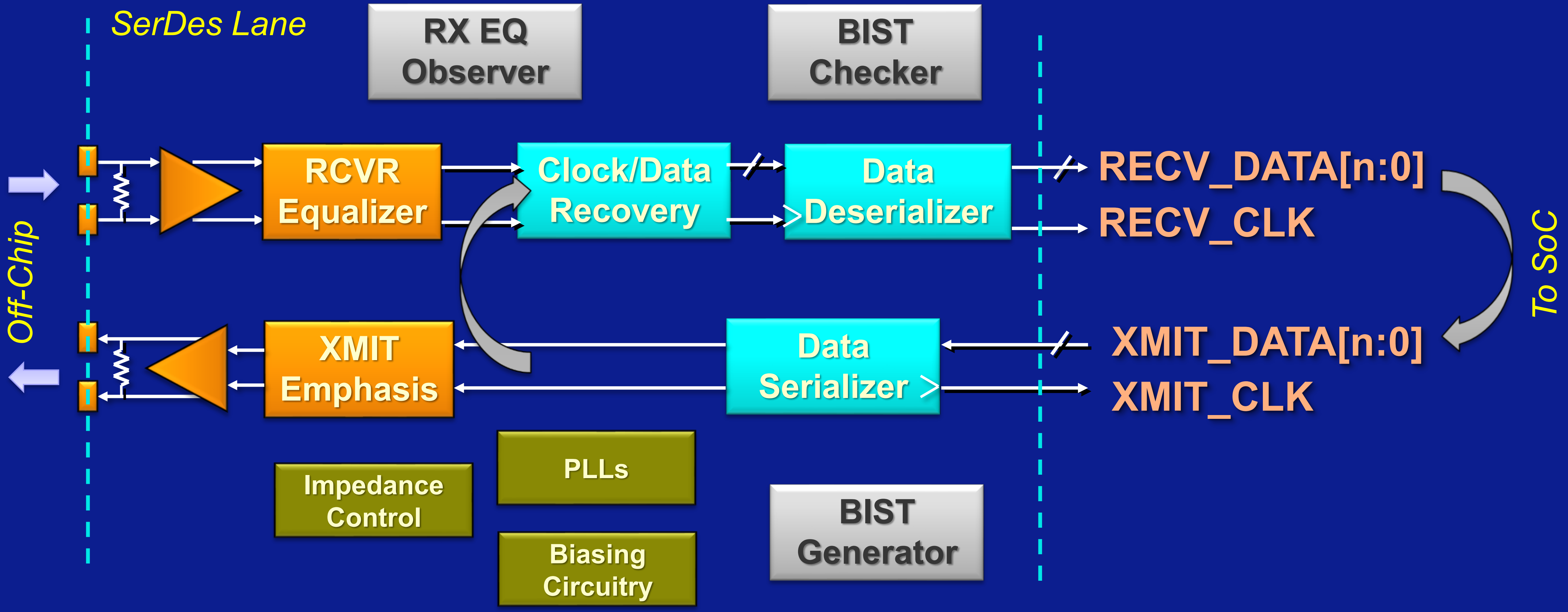
Receiver CTLE Testing





- Control Decode Tests
- HF/MF/LF Observation
- Regulator Observation
- Overrides/Status
- LOS Detect

Things that Became "Standard"

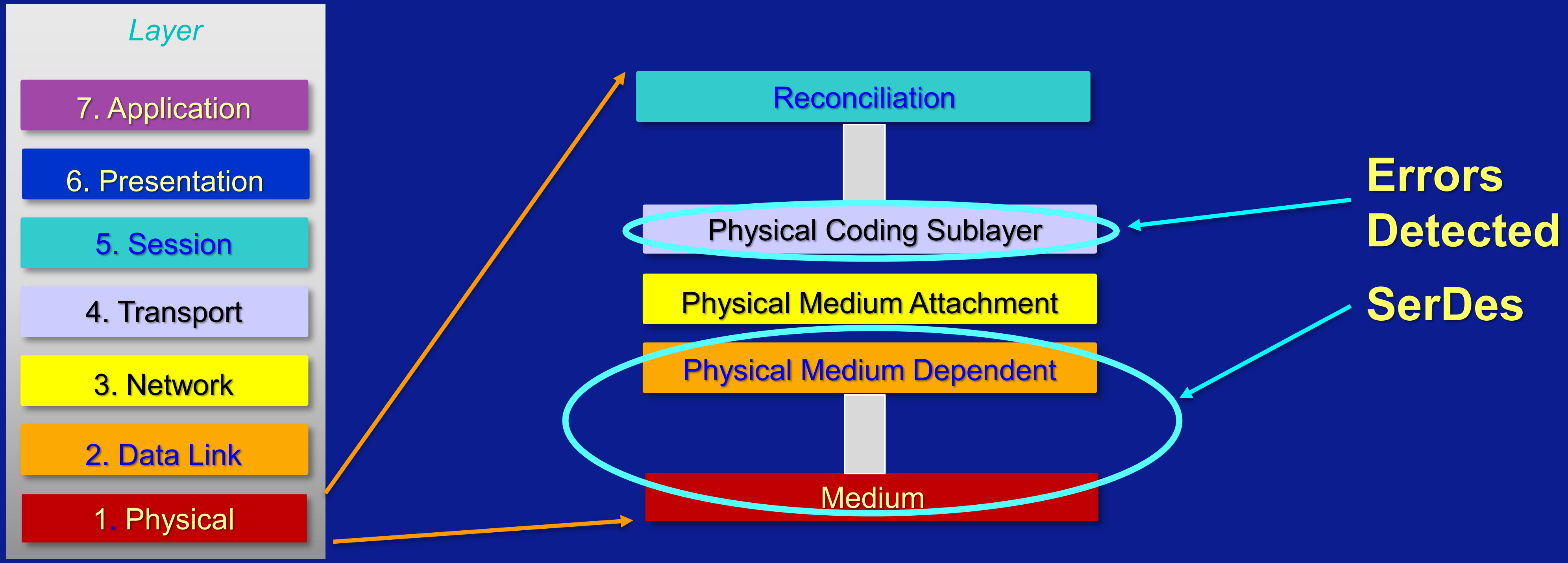


How to get what you need from Design

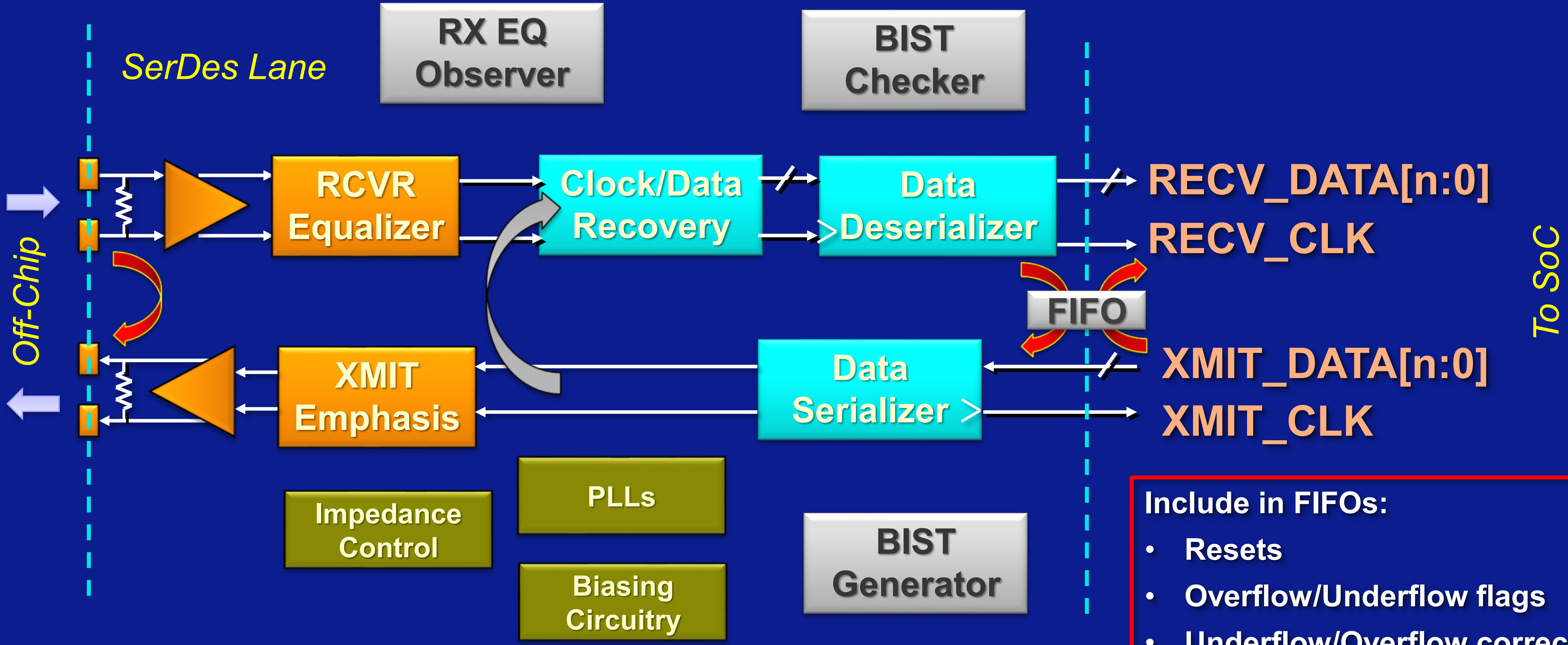
How to prove it’s not the Analog

How to get Test to go Away

Detecting if Things Go Wrong



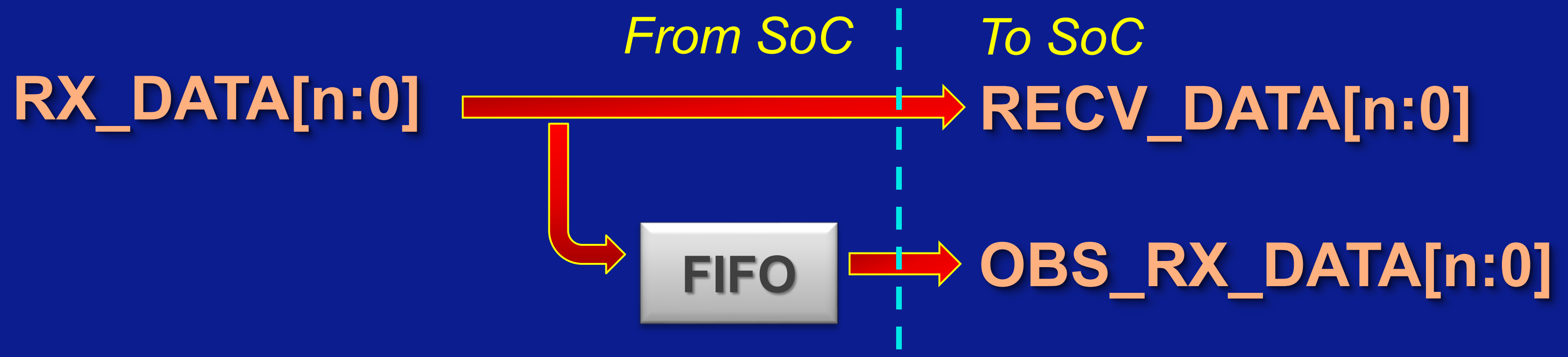
Isolation



Include in FIFOs:

- Resets
- Overflow/Underflow flags
- Underflow/Overflow correction

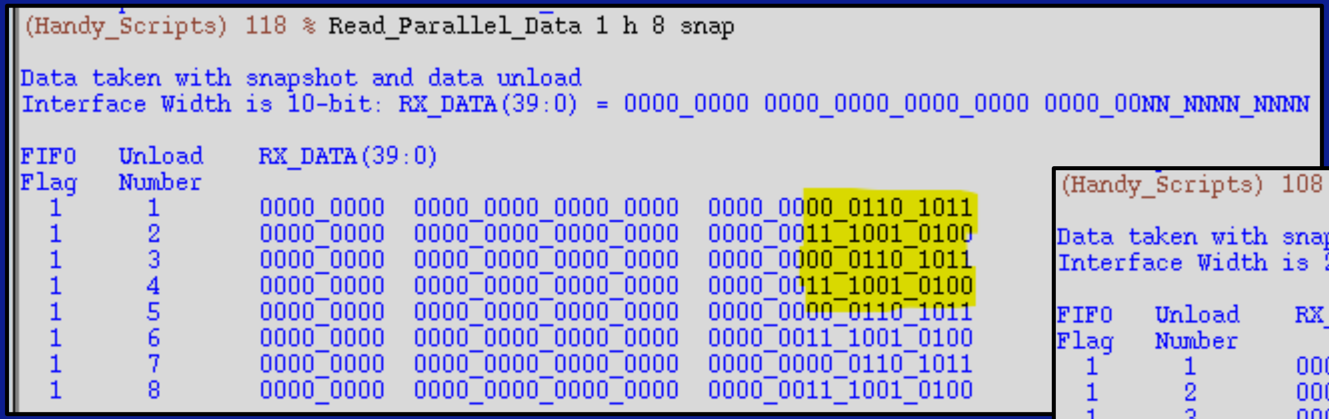
Multi-purpose FIFOs



Include in FIFOs:

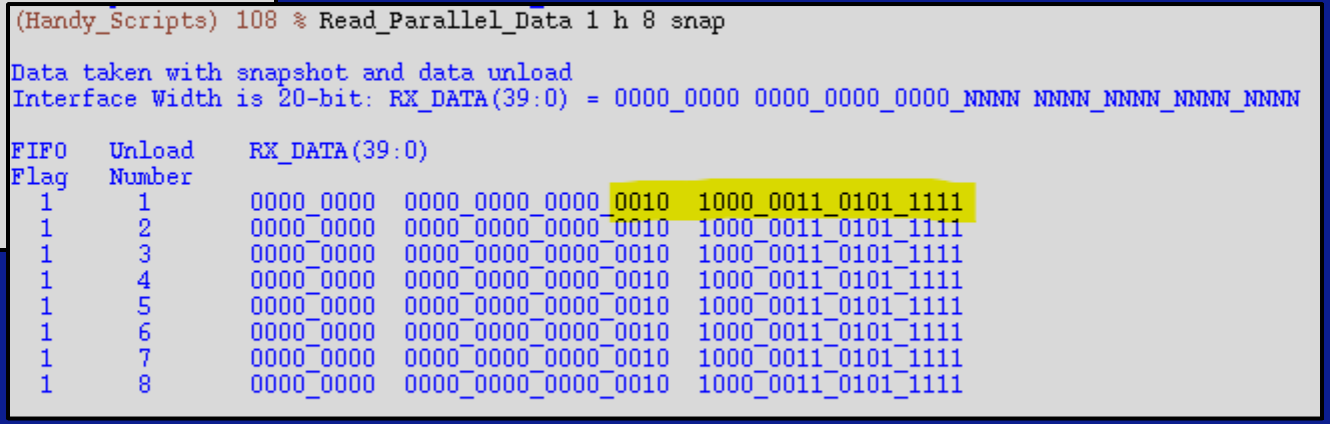
- **Resets**
- **Overflow/Underflow flags**
- **Underflow/Overflow correction**

Capturing K28.5 (10bit interface):



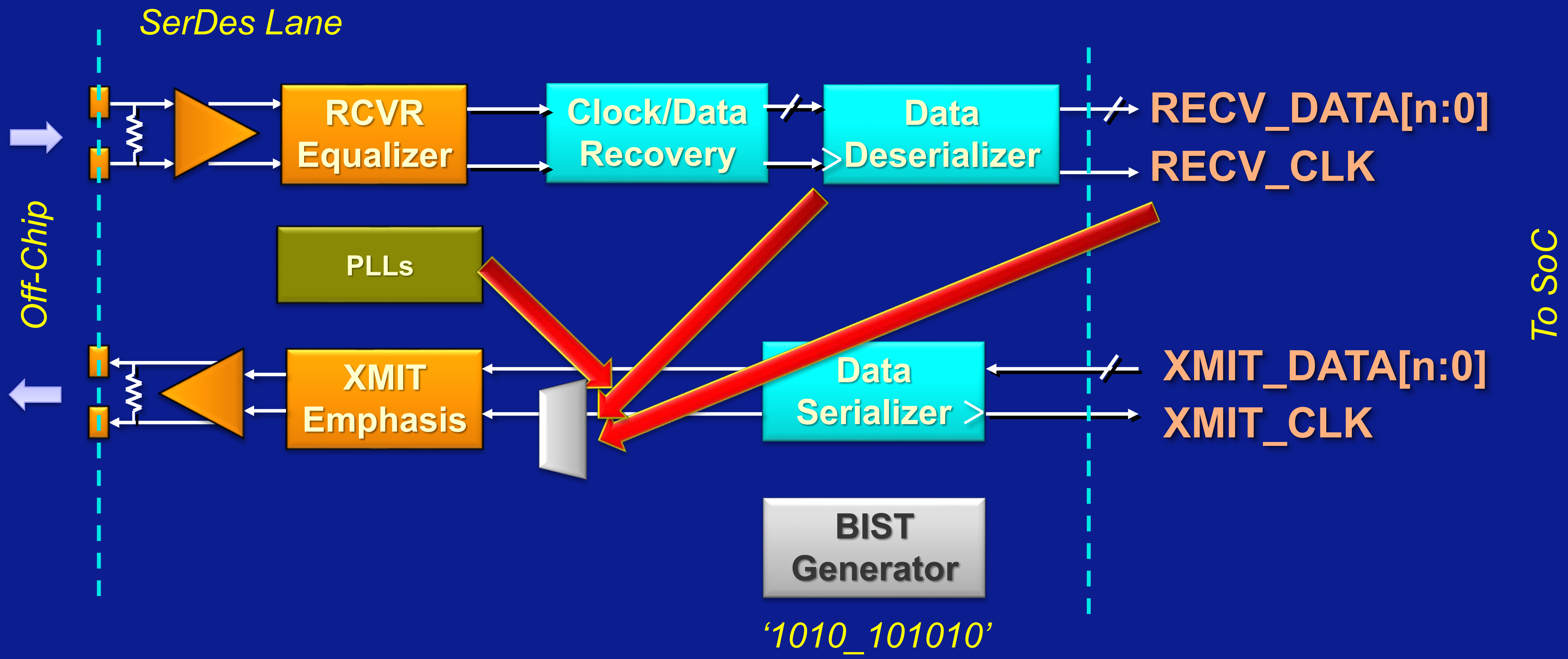
```
(Handy_Scripts) 118 % Read_Parallel_Data 1 h 8 snap

Data taken with snapshot and data unload
Interface Width is 10-bit: RX_DATA(39:0) = 0000_0000 0000_0000_0000_0000 0000_00NN_NNNN_NNNN

FIFO   Unload    RX_DATA(39:0)
Flag   Number
 1       1       0000_0000  0000_0000_0000_0000  0000_0000_0110_1011
 1       2       0000_0000  0000_0000_0000_0000  0000_0011_1001_0100
 1       3       0000_0000  0000_0000_0000_0000  0000_0000_0110_1011
 1       4       0000_0000  0000_0000_0000_0000  0000_0011_1001_0100
 1       5       0000_0000  0000_0000_0000_0000  0000_0000_0110_1011
 1       6       0000_0000  0000_0000_0000_0000  0000_0011_1001_0100
 1       7       0000_0000  0000_0000_0000_0000  0000_0000_0110_1011
 1       8       0000_0000  0000_0000_0000_0000  0000_0011_1001_0100
```

Capturing K28.5 (20bit interface):

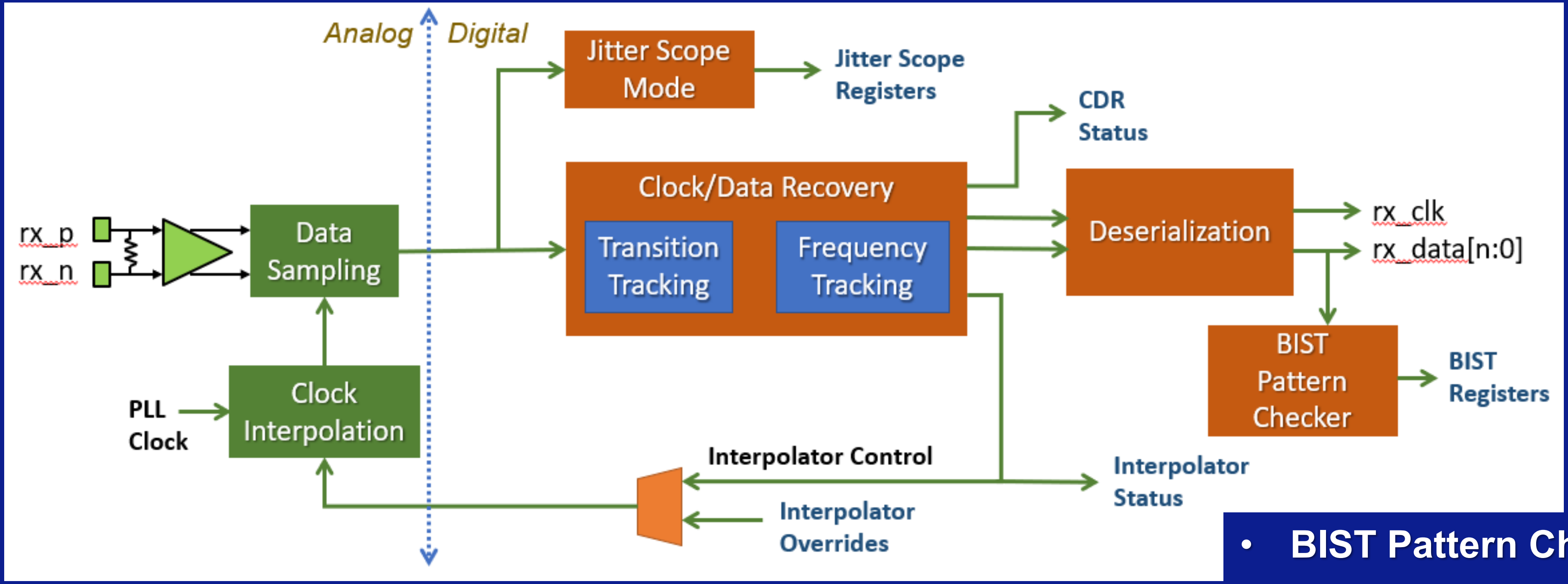


```
(Handy_Scripts) 108 % Read_Parallel_Data 1 h 8 snap

Data taken with snapshot and data unload
Interface Width is 20-bit: RX_DATA(39:0) = 0000_0000 0000_0000_0000_NNNN NNNN_NNNN_NNNN_NNNN

FIFO   Unload    RX_DATA(39:0)
Flag   Number
 1       1       0000_0000  0000_0000_0000_0010  1000_0011_0101_1111
 1       2       0000_0000  0000_0000_0000_0010  1000_0011_0101_1111
 1       3       0000_0000  0000_0000_0000_0010  1000_0011_0101_1111
 1       4       0000_0000  0000_0000_0000_0010  1000_0011_0101_1111
 1       5       0000_0000  0000_0000_0000_0010  1000_0011_0101_1111
 1       6       0000_0000  0000_0000_0000_0010  1000_0011_0101_1111
 1       7       0000_0000  0000_0000_0000_0010  1000_0011_0101_1111
 1       8       0000_0000  0000_0000_0000_0010  1000_0011_0101_1111
```

Clock Observation

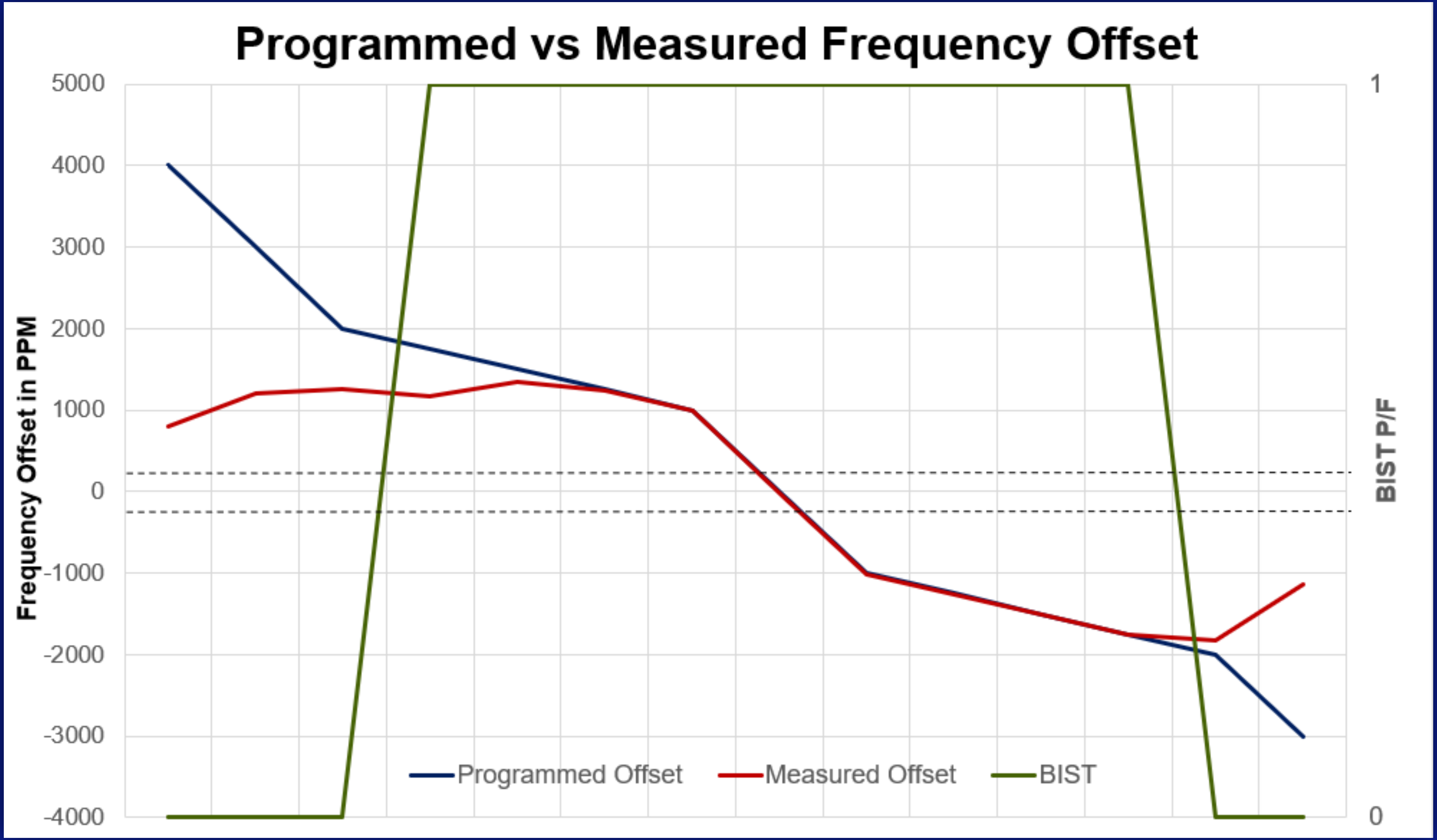


RX Clock/Data Recovery



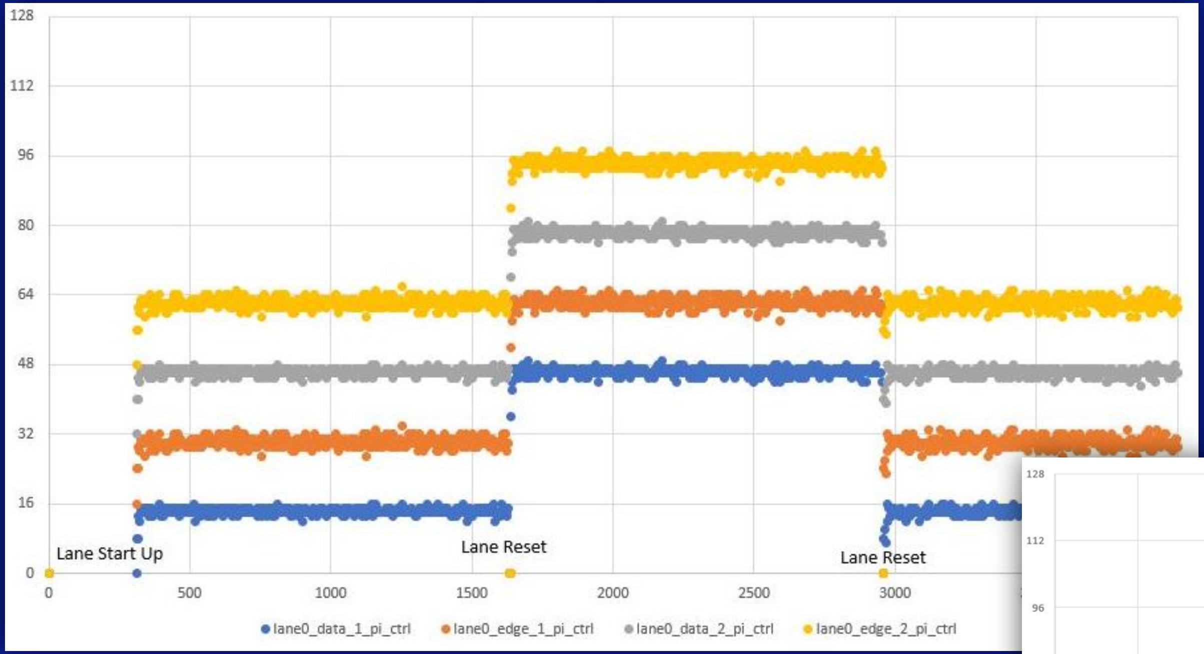
- BIST Pattern Checker (NXP AN5119)
- Internal Eye Monitor (NXP AN5119)
- Frequency Offset Status
- Interpolator Status/Overrides
- CDR Stall Detector
- False Lock Watchdog

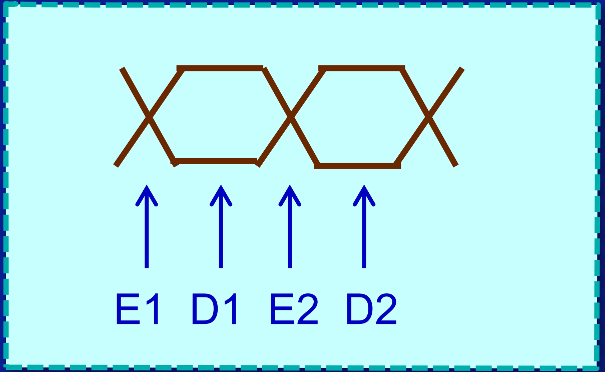
Frequency Offset

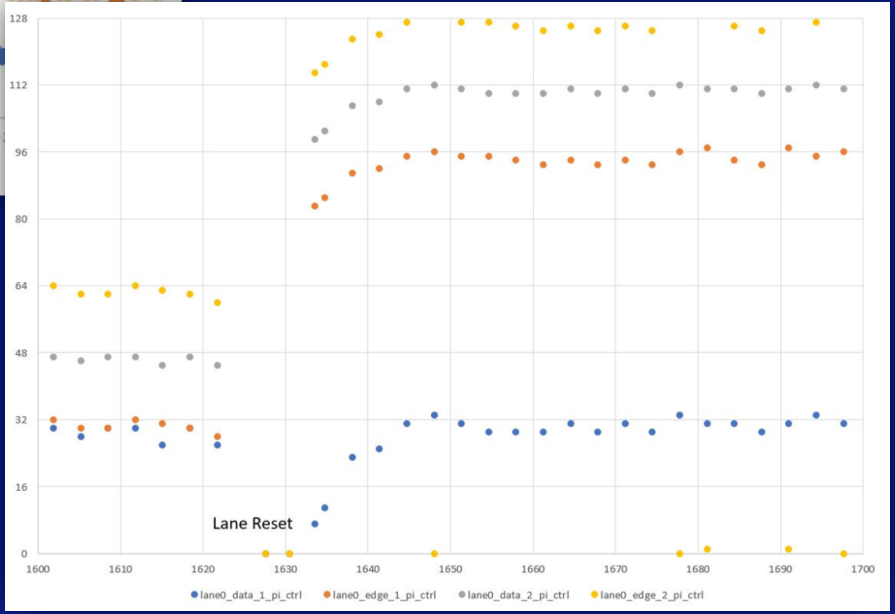


Interpolator Observation

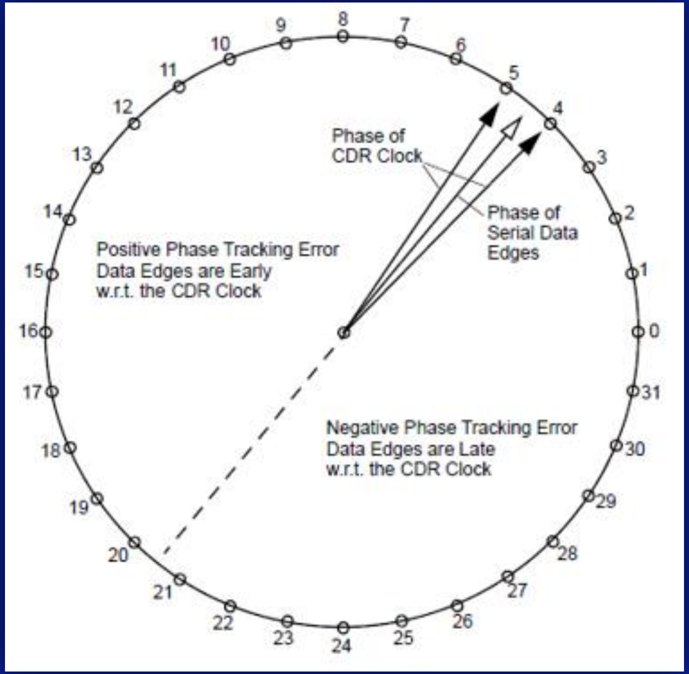
Interpolator Position



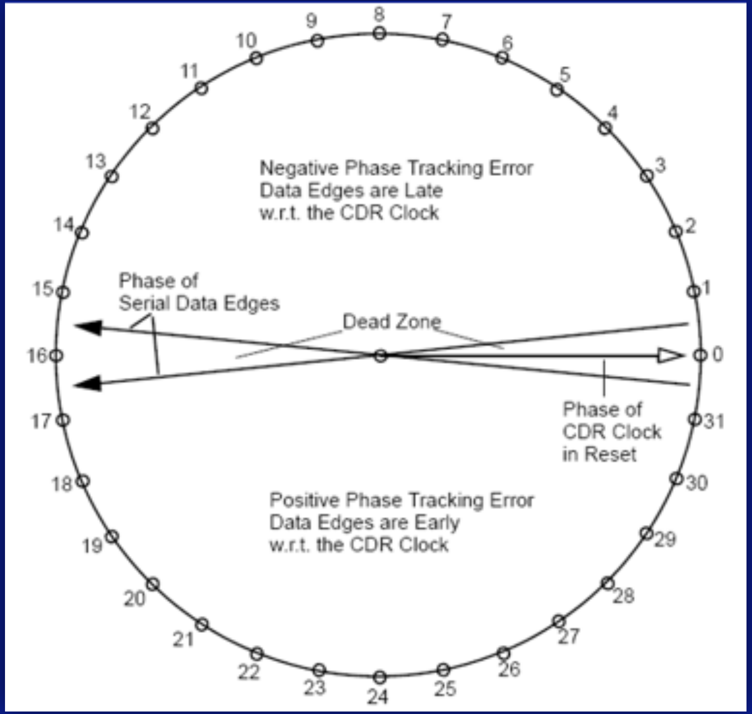




Every Loop Needs an 'Unstick' Plan



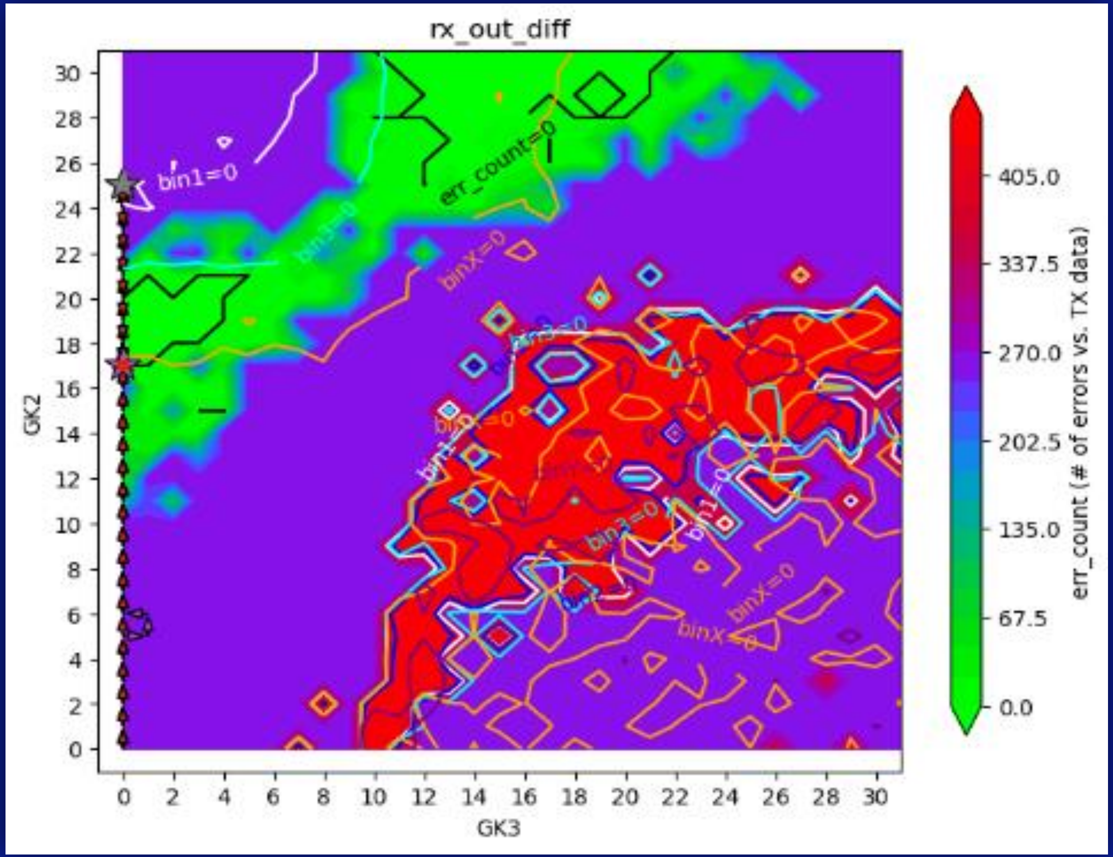
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Phase of CDR Clock
Phase of Serial Data Edges
Positive Phase Tracking Error Data Edges are Early w.r.t. the CDR Clock
Negative Phase Tracking Error Data Edges are Late w.r.t. the CDR Clock



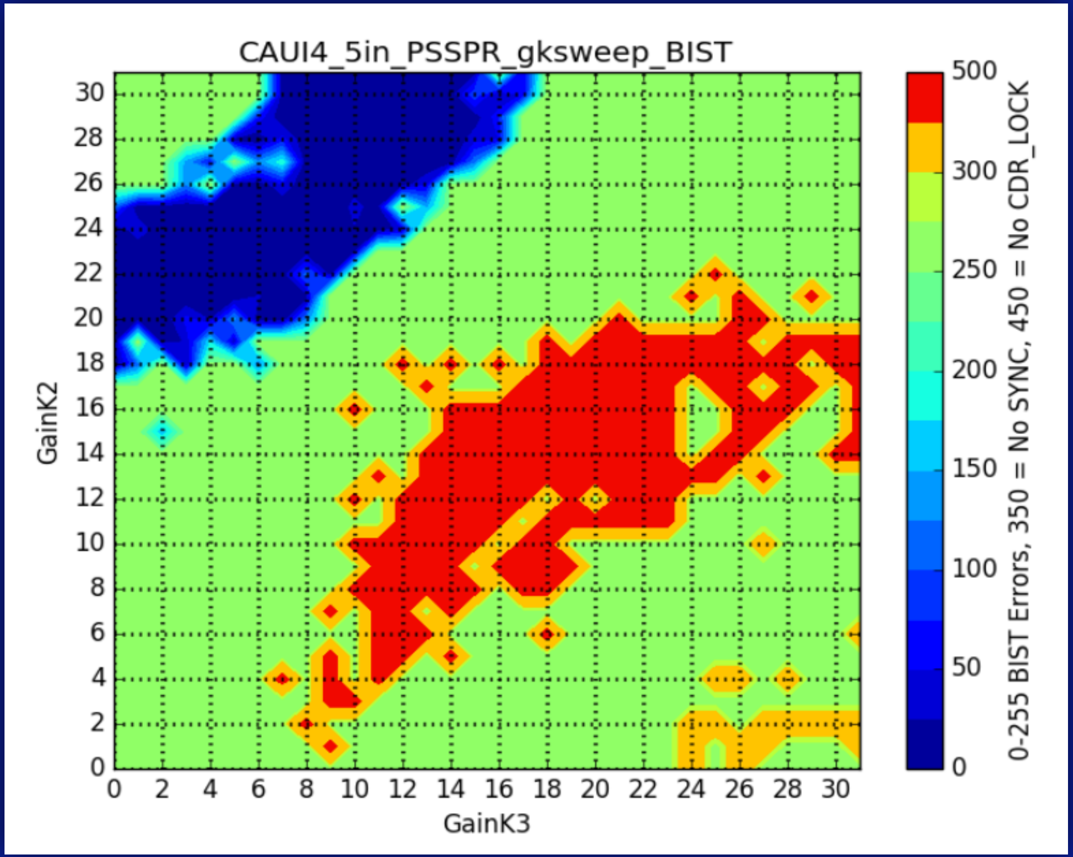
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Negative Phase Tracking Error Data Edges are Late w.r.t. the CDR Clock
Phase of Serial Data Edges
Dead Zone
Phase of CDR Clock in Reset
Positive Phase Tracking Error Data Edges are Early w.r.t. the CDR Clock

- Tester is 'clean' environment
- Suppose startup position ends up 180deg out from initial state
- Correct by overriding and releasing interpolator position
- Implement a watchdog timer

RX Equalization Design Confirmation

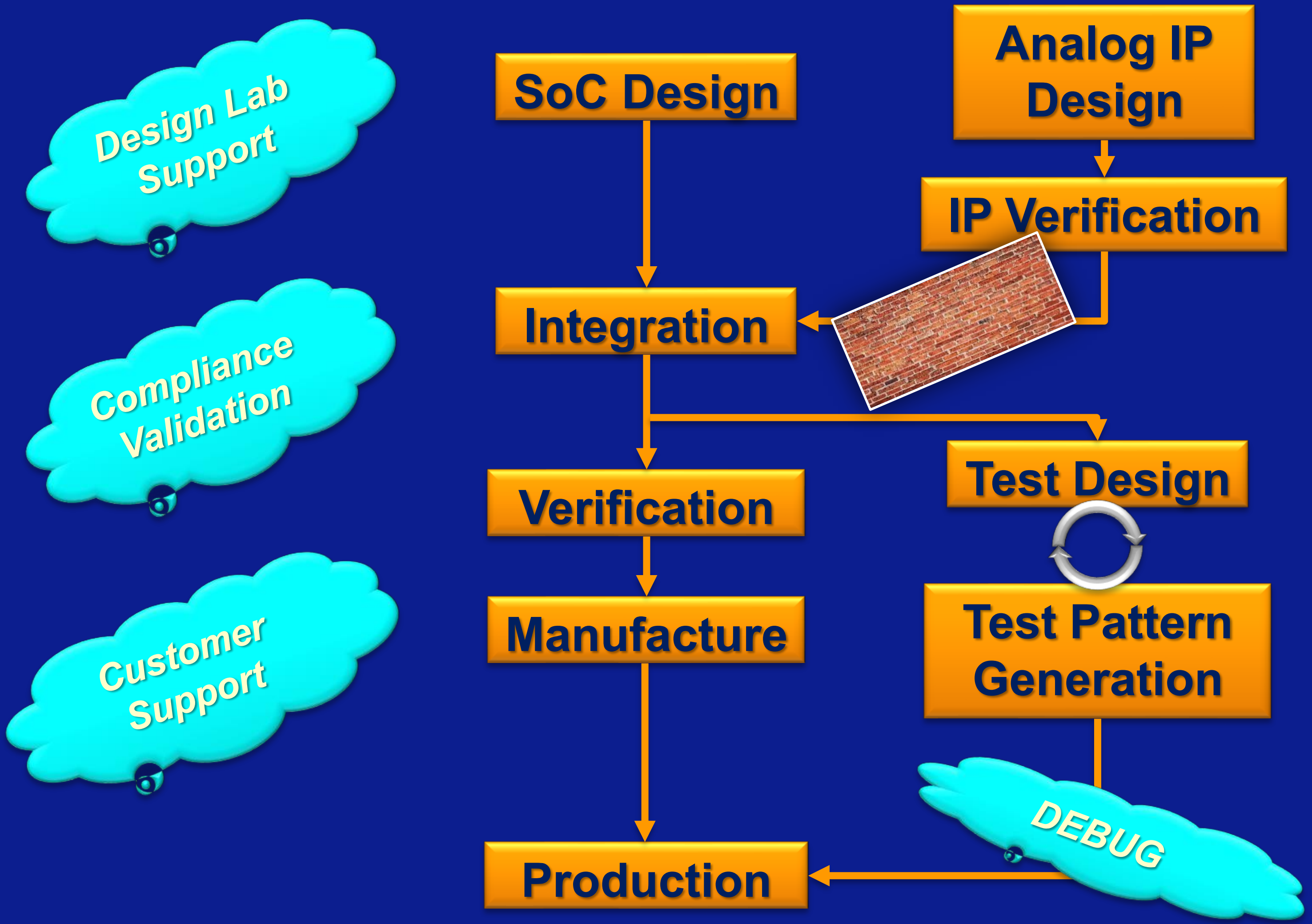


Simulation – 6.75in channel

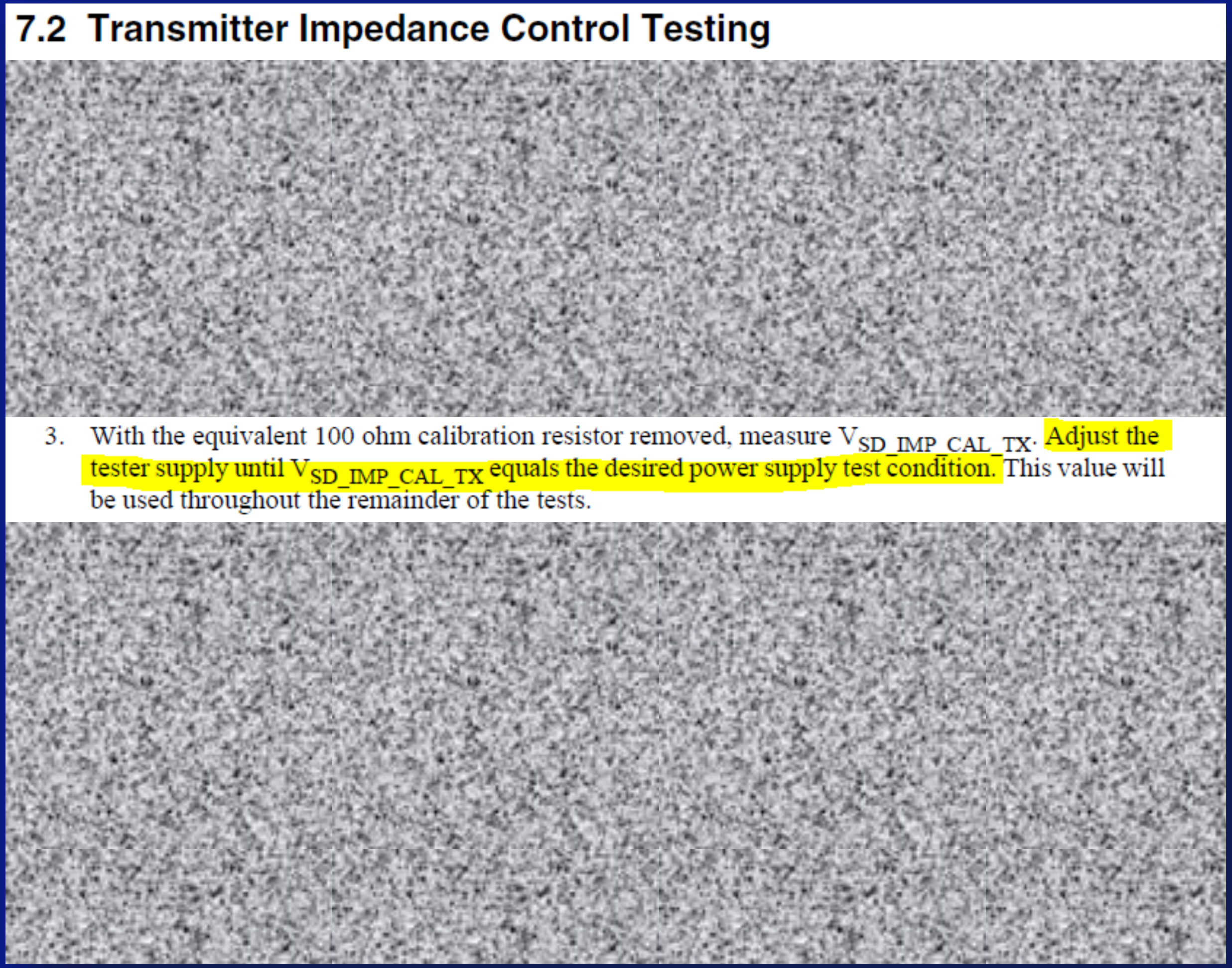


Silicon – 5in channel

“Over The Wall” Design for Test



How Not to Write a Test Guide



7.2 Transmitter Impedance Control Testing

3. With the equivalent 100 ohm calibration resistor removed, measure $V_{SD\_IMP\_CAL\_TX}$. Adjust the tester supply until $V_{SD\_IMP\_CAL\_TX}$ equals the desired power supply test condition. This value will be used throughout the remainder of the tests.

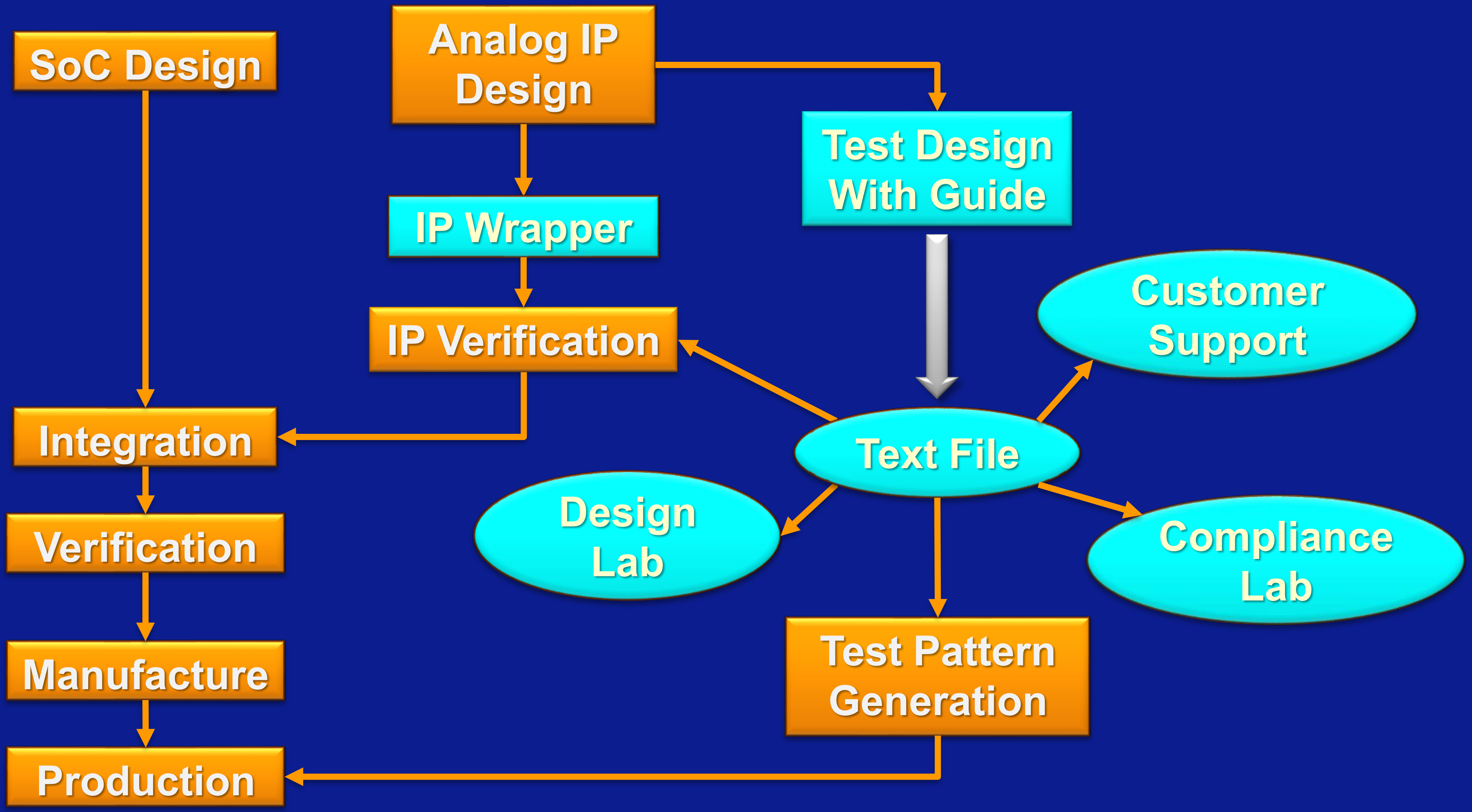
Understanding the Test Environment

- **ELIF**
- Manufacturing Tolerances
- Testing with Digital Testers
  - All conditions known at the START of the test
  - Fixed Voltage
  - Fixed Clock
  - Deterministic 1's and 0's
  - Load Board, Pogo Pins, Socket….***SIGNAL INTEGRITY!!!***
- ***Fully*** defined stress test modes

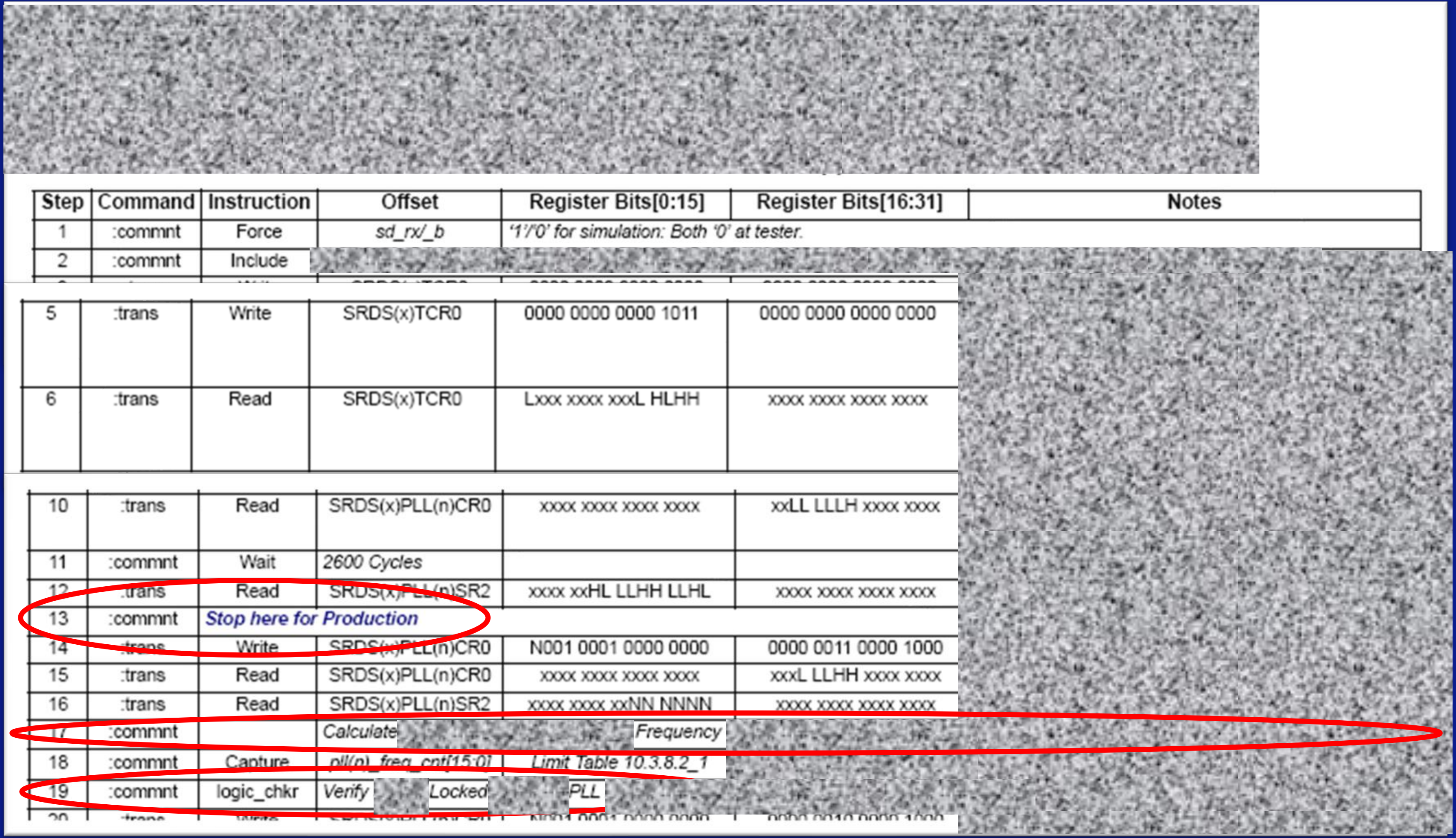
Fully Defined Stress Test Modes

- Voltages applied during Stress Testing
- Clock Frequencies used during Stress Testing
- Device Control Method
- Multi-Site? IDDQ Limitations?
- ***Suggest:***
  - Simplified Access: **Stress_En[3:0]**
  - Internal Biasing
  - Internal Control of Activity Levels
  - Internal Regulators Bypassed as Needed
  - Simplified Status: **Stress_Result[3:0]**

Integrated Design for Test



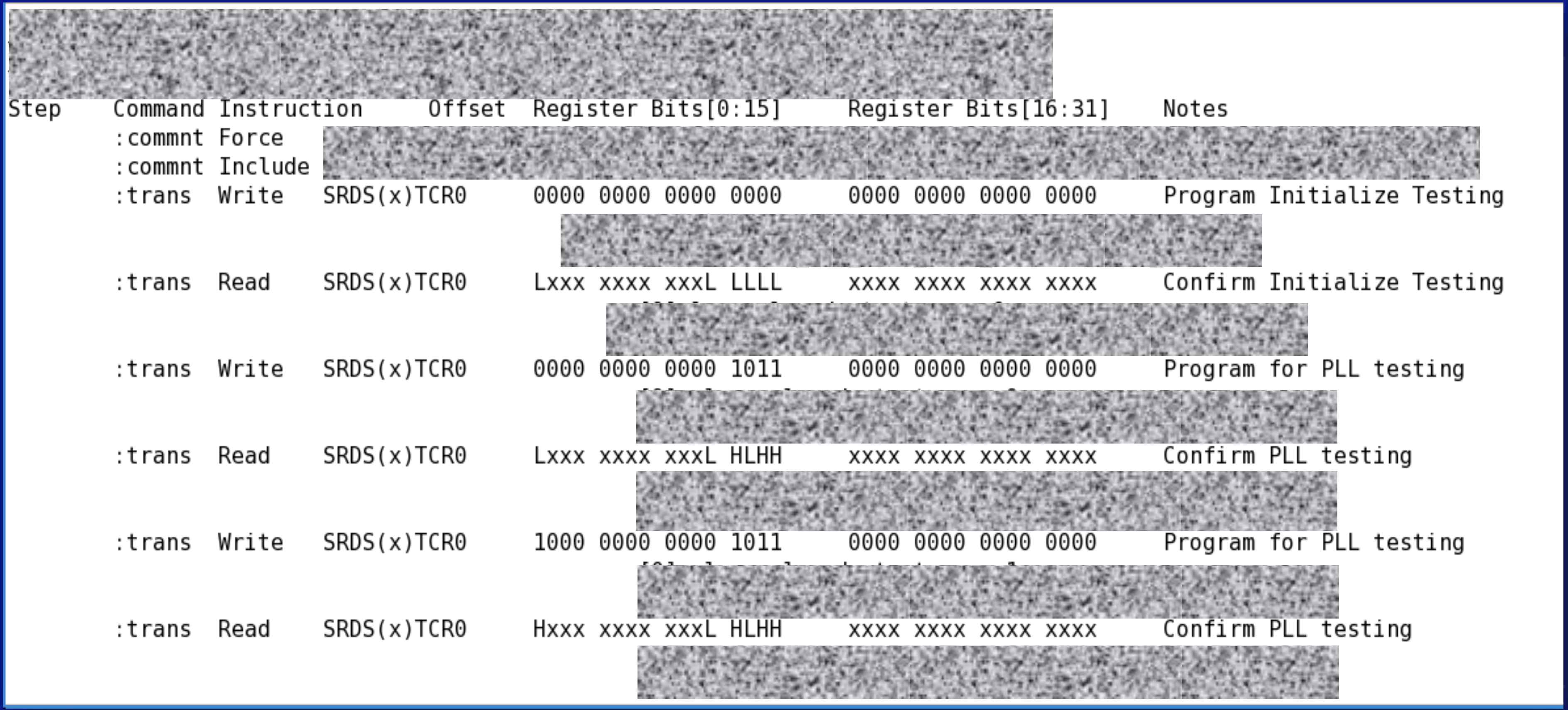
Human Readable Format



[redacted]

Step	Command	Instruction	Offset	Register Bits[0:15]	Register Bits[16:31]	Notes
1	:commnt	Force	*sd_rx/_b*	*'1'/'0' for simulation: Both '0' at tester.*		
2	:commnt	Include	[redacted]			
5	:trans	Write	SRDS(x)TCR0	0000 0000 0000 1011	0000 0000 0000 0000	[redacted]
6	:trans	Read	SRDS(x)TCR0	Lxxx xxxx xxxL HLHH	xxxx xxxx xxxx xxxx	[redacted]
10	:trans	Read	SRDS(x)PLL(n)CR0	xxxx xxxx xxxx xxxx	xxLL LLLH xxxx xxxx	[redacted]
11	:commnt	Wait	*2600 Cycles*			[redacted]
12	:trans	Read	SRDS(x)PLL(n)SR2	xxxx xxHL LLHH LLHL	xxxx xxxx xxxx xxxx	[redacted]
13	:commnt	***Stop here for Production***				[redacted]
14	:trans	Write	SRDS(x)PLL(n)CR0	N001 0001 0000 0000	0000 0011 0000 1000	[redacted]
15	:trans	Read	SRDS(x)PLL(n)CR0	xxxx xxxx xxxx xxxx	xxxL LLHH xxxx xxxx	[redacted]
16	:trans	Read	SRDS(x)PLL(n)SR2	xxxx xxxx xxNN NNNN	xxxx xxxx xxxx xxxx	[redacted]
17	:commnt		*Calculate* [redacted] *Frequency* [redacted]			
18	:commnt	Capture	*pll(n)_freq_cnt[15:0]*	*Limit Table 10.3.8.2_1* [redacted]		
19	:commnt	logic_chkr	*Verify* [redacted] *Locked* [redacted] *PLL* [redacted]			
20	:trans	Write	SRDS(x)PLL(n)CR0	N001 0001 0000 0000	0000 0010 0000 1000	[redacted]

Text File



[redacted]

```
Step      Command  Instruction     Offset  Register Bits[0:15]     Register Bits[16:31]     Notes
          :commnt  Force
          :commnt  Include
          :trans   Write   SRDS(x)TCR0     0000 0000 0000 0000     0000 0000 0000 0000     Program Initialize Testing
```

[redacted]

```
          :trans   Read    SRDS(x)TCR0     Lxxx xxxx xxxL LLLL     xxxx xxxx xxxx xxxx     Confirm Initialize Testing
```

[redacted]

```
          :trans   Write   SRDS(x)TCR0     0000 0000 0000 1011     0000 0000 0000 0000     Program for PLL testing
```

[redacted]

```
          :trans   Read    SRDS(x)TCR0     Lxxx xxxx xxxL HLHH     xxxx xxxx xxxx xxxx     Confirm PLL testing
```

[redacted]

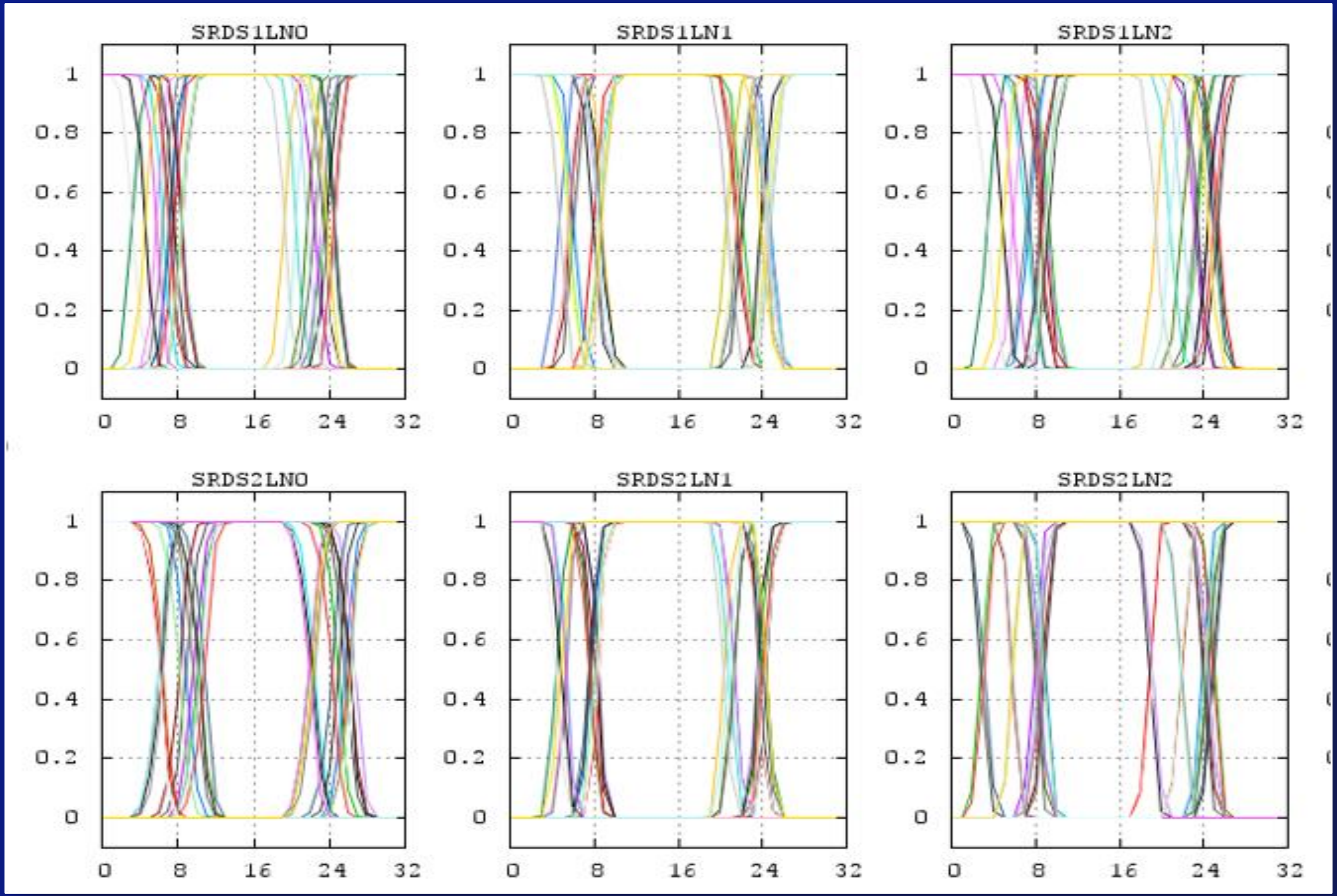
```
          :trans   Write   SRDS(x)TCR0     1000 0000 0000 1011     0000 0000 0000 0000     Program for PLL testing
```

[redacted]

```
          :trans   Read    SRDS(x)TCR0     Hxxx xxxx xxxL HLHH     xxxx xxxx xxxx xxxx     Confirm PLL testing
```

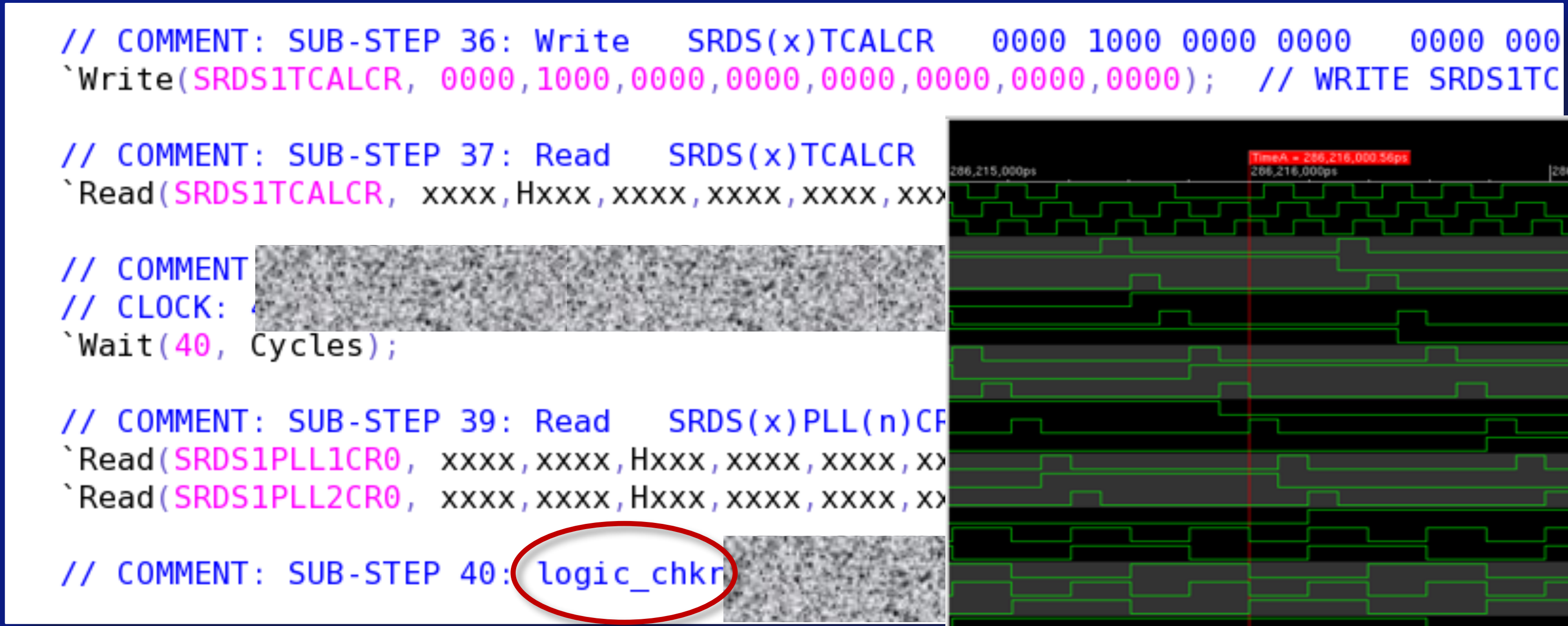
[redacted]

To Tester

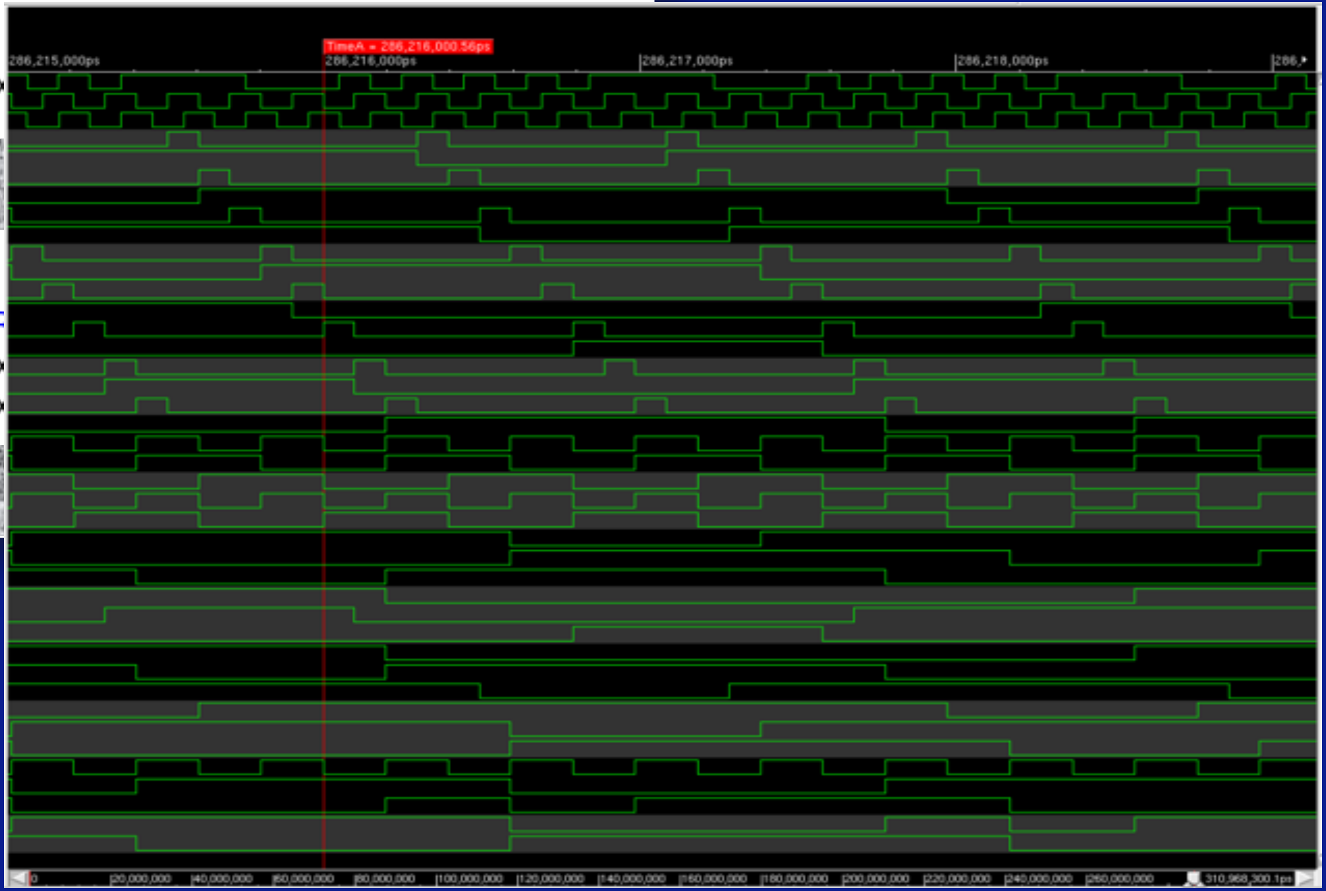


SRDS1LN0
SRDS1LN1
SRDS1LN2
SRDS2LN0
SRDS2LN1
SRDS2LN2
1
0.8
0.6
0.4
0.2
0
0
8
16
24
32

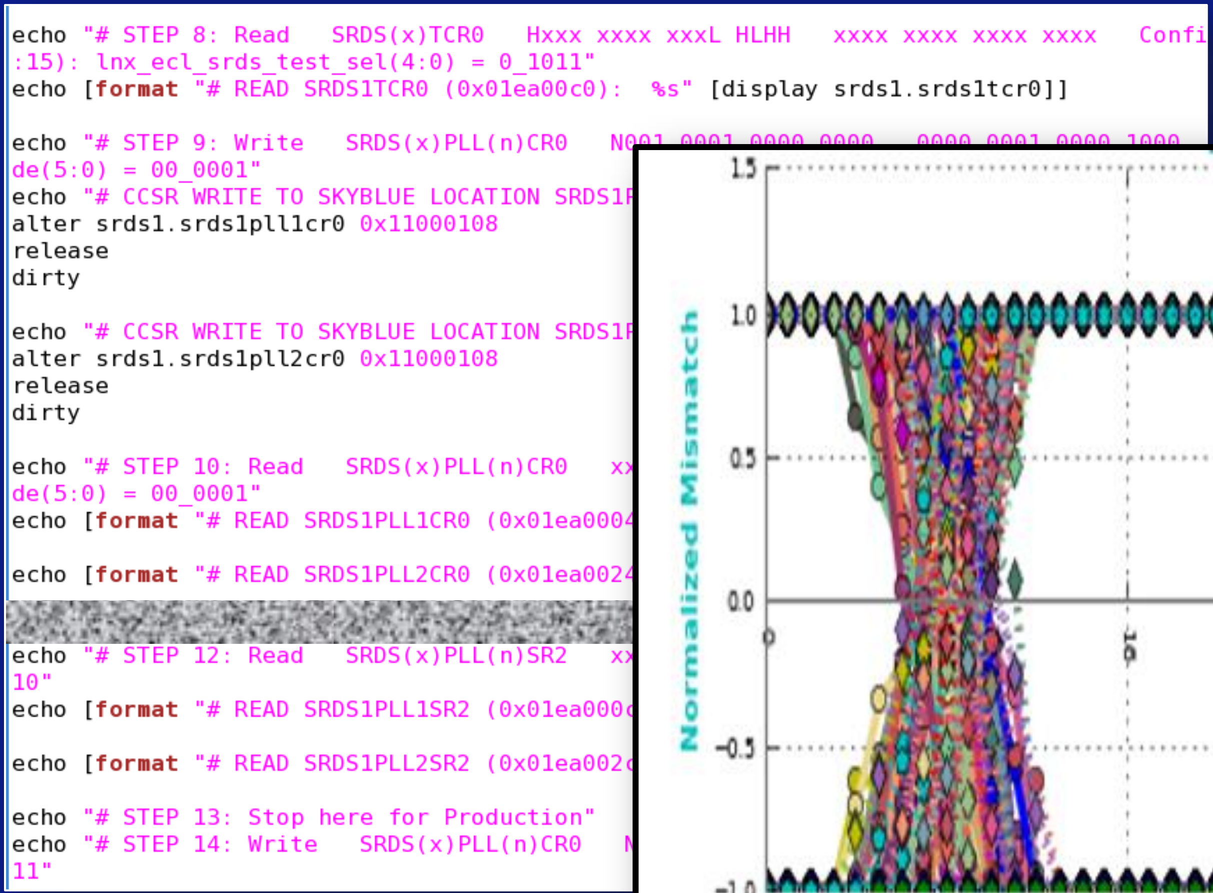
RTL Stimulus



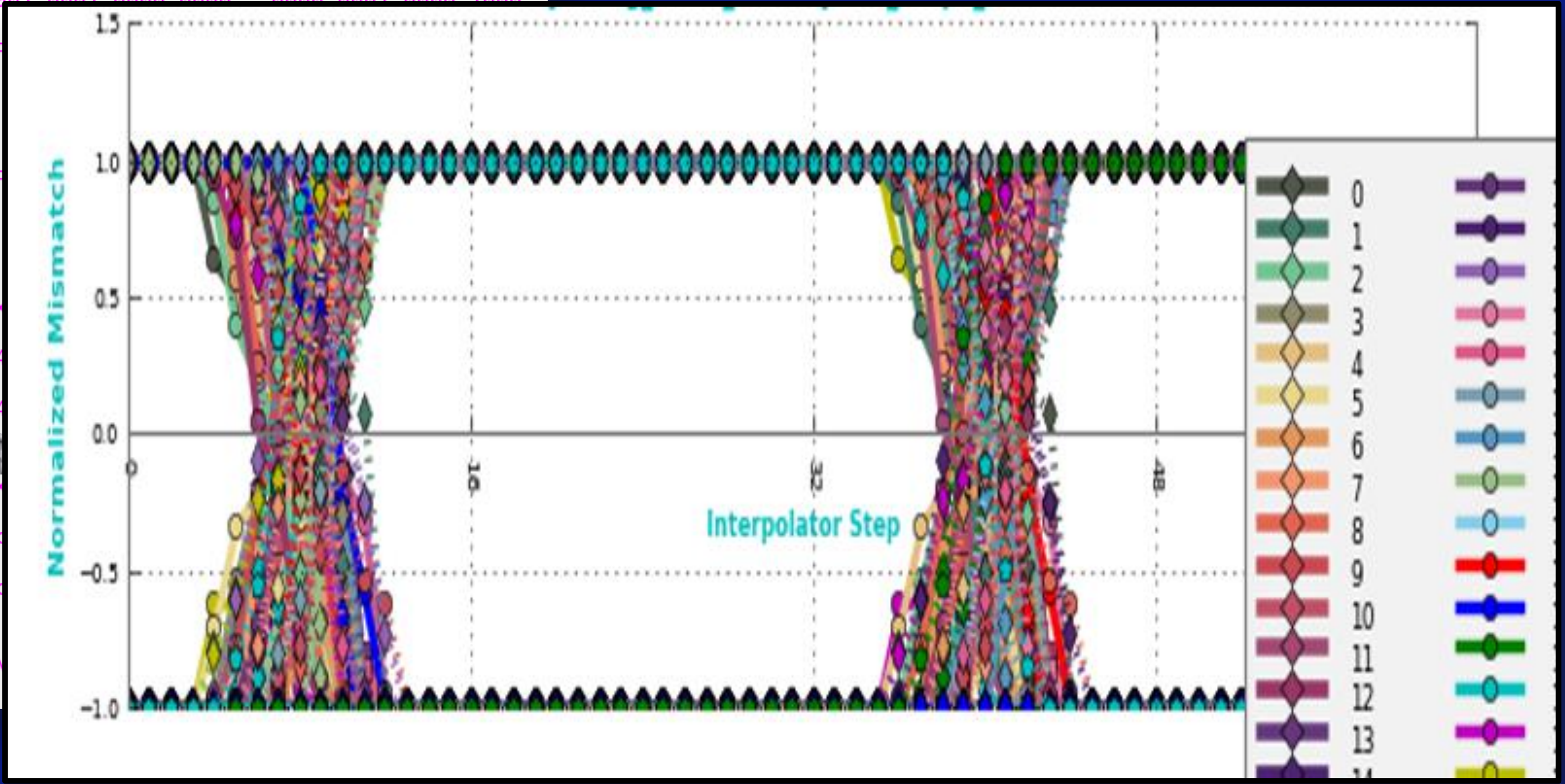
```
// COMMENT: SUB-STEP 36: Write   SRDS(x)TCALCR   0000 1000 0000 0000   0000 000
`Write(SRDS1TCALCR, 0000,1000,0000,0000,0000,0000,0000,0000);  // WRITE SRDS1TC

// COMMENT: SUB-STEP 37: Read   SRDS(x)TCALCR
`Read(SRDS1TCALCR, xxxx,Hxxx,xxxx,xxxx,xxxx,xxx

// COMMENT [redacted]
// CLOCK: [redacted]
`Wait(40, Cycles);

// COMMENT: SUB-STEP 39: Read   SRDS(x)PLL(n)CR
`Read(SRDS1PLL1CR0, xxxx,xxxx,Hxxx,xxxx,xxxx,xx
`Read(SRDS1PLL2CR0, xxxx,xxxx,Hxxx,xxxx,xxxx,xx

// COMMENT: SUB-STEP 40: logic_chkr [redacted]
```



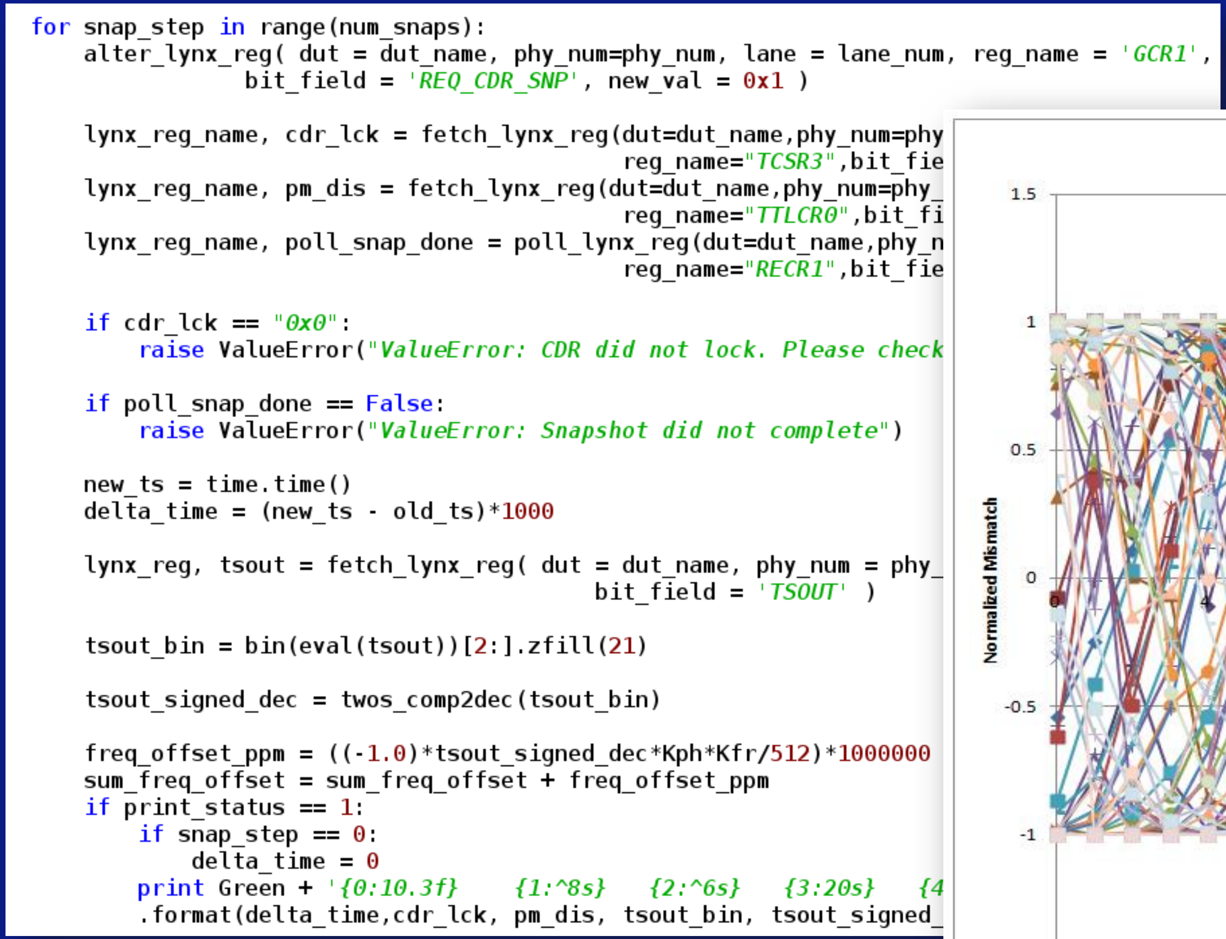
TCL Files for Design/Compliance Lab



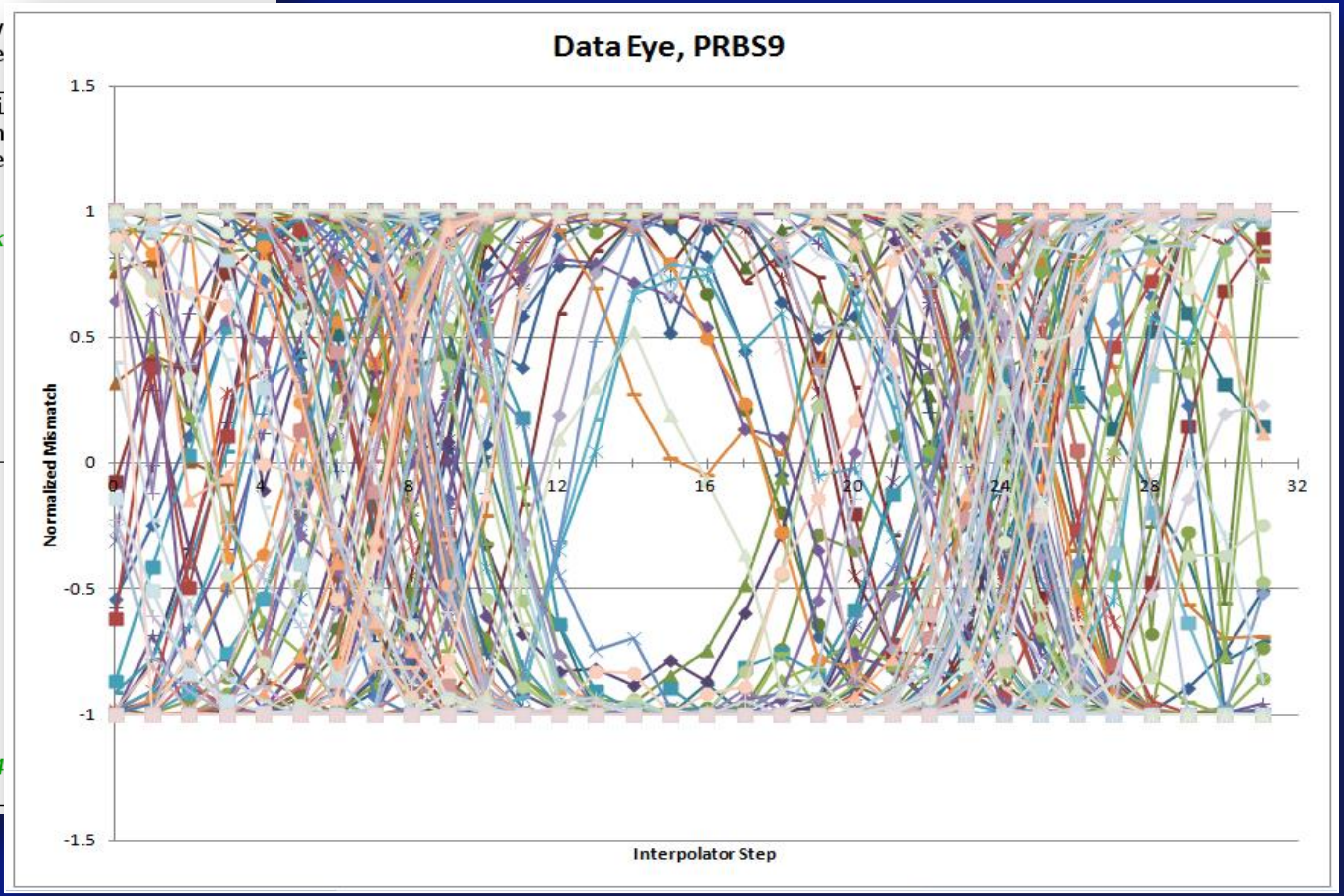
```
echo "# STEP 8: Read   SRDS(x)TCR0   Hxxx xxxx xxxL HLHH   xxxx xxxx xxxx xxxx   Confi
:15): lnx_ecl_srds_test_sel(4:0) = 0_1011"
echo [format "# READ SRDS1TCR0 (0x01ea00c0):  %s" [display srds1.srds1tcr0]]

echo "# STEP 9: Write   SRDS(x)PLL(n)CR0   N001 0001 0000 0000   0000 0001 0000 1000
de(5:0) = 00_0001"
echo "# CCSR WRITE TO SKYBLUE LOCATION SRDS1P
alter srds1.srds1pll1cr0 0x11000108
release
dirty

echo "# CCSR WRITE TO SKYBLUE LOCATION SRDS1P
alter srds1.srds1pll2cr0 0x11000108
release
dirty

echo "# STEP 10: Read   SRDS(x)PLL(n)CR0   xx
de(5:0) = 00_0001"
echo [format "# READ SRDS1PLL1CR0 (0x01ea0004

echo [format "# READ SRDS1PLL2CR0 (0x01ea0024

echo "# STEP 12: Read   SRDS(x)PLL(n)SR2   xx
10"
echo [format "# READ SRDS1PLL1SR2 (0x01ea000c

echo [format "# READ SRDS1PLL2SR2 (0x01ea002c

echo "# STEP 13: Stop here for Production"
echo "# STEP 14: Write   SRDS(x)PLL(n)CR0   N
11"
```



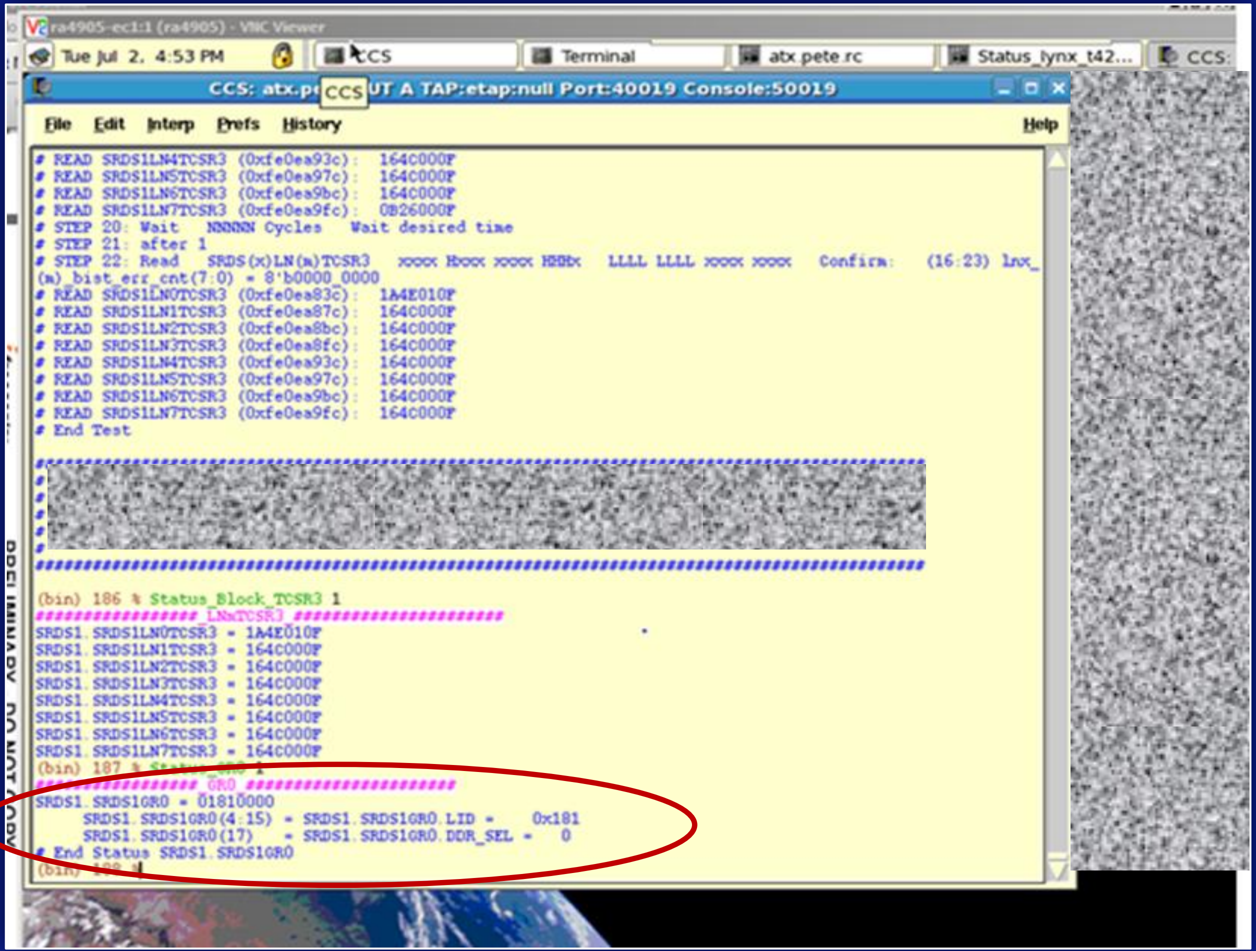
Python For The Validation Lab



```
for snap_step in range(num_snaps):
    alter_lynx_reg( dut = dut_name, phy_num=phy_num, lane = lane_num, reg_name = 'GCR1',
                bit_field = 'REQ_CDR_SNP', new_val = 0x1 )

    lynx_reg_name, cdr_lck = fetch_lynx_reg(dut=dut_name,phy_num=phy
                                        reg_name="TCSR3",bit_fie
    lynx_reg_name, pm_dis = fetch_lynx_reg(dut=dut_name,phy_num=phy_
                                        reg_name="TTLCR0",bit_fi
    lynx_reg_name, poll_snap_done = poll_lynx_reg(dut=dut_name,phy_n
                                        reg_name="RECR1",bit_fie

    if cdr_lck == "0x0":
        raise ValueError("ValueError: CDR did not lock. Please check

    if poll_snap_done == False:
        raise ValueError("ValueError: Snapshot did not complete")

    new_ts = time.time()
    delta_time = (new_ts - old_ts)*1000

    lynx_reg, tsout = fetch_lynx_reg( dut = dut_name, phy_num = phy_
                                      bit_field = 'TSOUT' )

    tsout_bin = bin(eval(tsout))[2:].zfill(21)

    tsout_signed_dec = twos_comp2dec(tsout_bin)

    freq_offset_ppm = ((-1.0)*tsout_signed_dec*Kph*Kfr/512)*1000000
    sum_freq_offset = sum_freq_offset + freq_offset_ppm
    if print_status == 1:
        if snap_step == 0:
            delta_time = 0
        print Green + '{0:10.3f}    {1:^8s}   {2:^6s}   {3:20s}   {4
        .format(delta_time,cdr_lck, pm_dis, tsout_bin, tsout_signed_
```



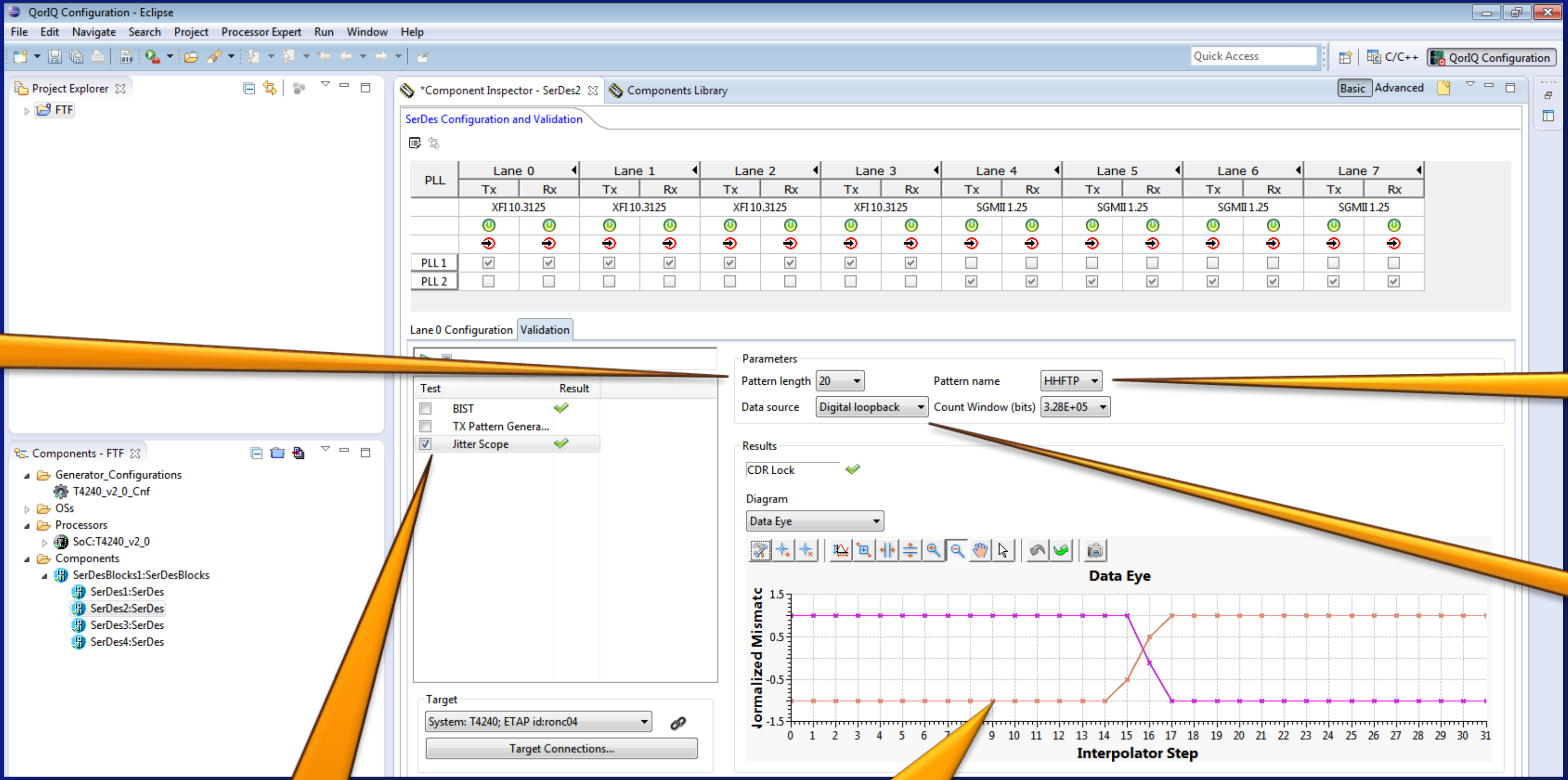
Even More From The Tester



ra4905-ec1:1 (ra4905) - VNC Viewer
Tue Jul 2, 4:53 PM
CCS
Terminal
atx pete.rc
Status_lynx_t42...
CCS:
CCS: atx.pe CCS UT A TAP:etap:null Port:40019 Console:50019
File
Edit
Interp
Prefs
History
Help

```
# READ SRDS1LN4TCSR3 (0xfe0ea93c):  164C000F
# READ SRDS1LN5TCSR3 (0xfe0ea97c):  164C000F
# READ SRDS1LN6TCSR3 (0xfe0ea9bc):  164C000F
# READ SRDS1LN7TCSR3 (0xfe0ea9fc):  0B26000F
# STEP 20: Wait   NNNNN Cycles   Wait desired time
# STEP 21: after 1
# STEP 22: Read   SRDS(x)LN(m)TCSR3   xxxx Hxxx xxxx HHHx   LLLL LLLL xxxx xxxx   Confirm:   (16:23) lnx_
(m)_bist_err_cnt(7:0) = 8'b0000_0000
# READ SRDS1LN0TCSR3 (0xfe0ea83c):  1A4E010F
# READ SRDS1LN1TCSR3 (0xfe0ea87c):  164C000F
# READ SRDS1LN2TCSR3 (0xfe0ea8bc):  164C000F
# READ SRDS1LN3TCSR3 (0xfe0ea8fc):  164C000F
# READ SRDS1LN4TCSR3 (0xfe0ea93c):  164C000F
# READ SRDS1LN5TCSR3 (0xfe0ea97c):  164C000F
# READ SRDS1LN6TCSR3 (0xfe0ea9bc):  164C000F
# READ SRDS1LN7TCSR3 (0xfe0ea9fc):  164C000F
# End Test

[redacted]
##########################################################################################

(bin) 186 % Status_Block_TCSR3 1
################### LNmTCSR3 ########################
SRDS1.SRDS1LN0TCSR3 = 1A4E010F
SRDS1.SRDS1LN1TCSR3 = 164C000F
SRDS1.SRDS1LN2TCSR3 = 164C000F
SRDS1.SRDS1LN3TCSR3 = 164C000F
SRDS1.SRDS1LN4TCSR3 = 164C000F
SRDS1.SRDS1LN5TCSR3 = 164C000F
SRDS1.SRDS1LN6TCSR3 = 164C000F
SRDS1.SRDS1LN7TCSR3 = 164C000F
(bin) 187 % Status_GR0 1
################### GR0 ########################
SRDS1.SRDS1GR0 = 01810000
     SRDS1.SRDS1GR0(4:15) = SRDS1.SRDS1GR0.LID =     0x181
     SRDS1.SRDS1GR0(17)   = SRDS1.SRDS1GR0.DDR_SEL =   0
# End Status SRDS1.SRDS1GR0
(bin) 188 %
```

Customer Tools



PLL	Lane 0		Lane 1		Lane 2		Lane 3		Lane 4		Lane 5		Lane 6		Lane 7	
	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx
	XFI 10.3125		XFI 10.3125		XFI 10.3125		XFI 10.3125		SGMII 1.25		SGMII 1.25		SGMII 1.25		SGMII 1.25	
PLL 1	☑	☑	☑	☑	☑	☑	☑	☑	☐	☐	☐	☐	☐	☐	☐	☐
PLL 2	☐	☐	☐	☐	☐	☐	☐	☐	☑	☑	☑	☑	☑	☑	☑	☑

Test	Result
☐ BIST	✓
☐ TX Pattern Genera...	
☑ Jitter Scope	✓

Parameters
Pattern length: 20
Pattern name: HHFTP
Data source: Digital loopback
Count Window (bits): 3.28E+05

Results
CDR Lock ✓
Diagram: Data Eye

Target
System: T4240; ETAP id:ronc04
Target Connections...

Select pattern length

Select Test Pattern

Run internal Loopback

Select Jitter Scope

Select Data Eye or Recovered Data Stream

App Note: NXP AN5119

During That Inevitable Debug…

1. Check your power supplies
2. Check your clocks
   a. Frequency?
   b. Spread Spectrum?
3. Are you still in reset?
4. Check the programmed configuration
5. Repeat steps 1-4 at least twice
6. Check your PLLs
7. Do you have data? Does it make sense?
8. Back up to simpler test/setup
9. Document steps 1-8 in something other than email

In Summary….

- Be sure ALL disciplines understand ALL Test Environments
- Make the Analog look "Digital"
  - Configurations in one group of registers
  - Status in another group of registers
  - Power Downs and Resets in a third group of registers
- Make Test Description Modular, Scriptable and Portable
- Provide for Debug Hooks
- IEEE 1687TM - 2014: IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device, November 2014

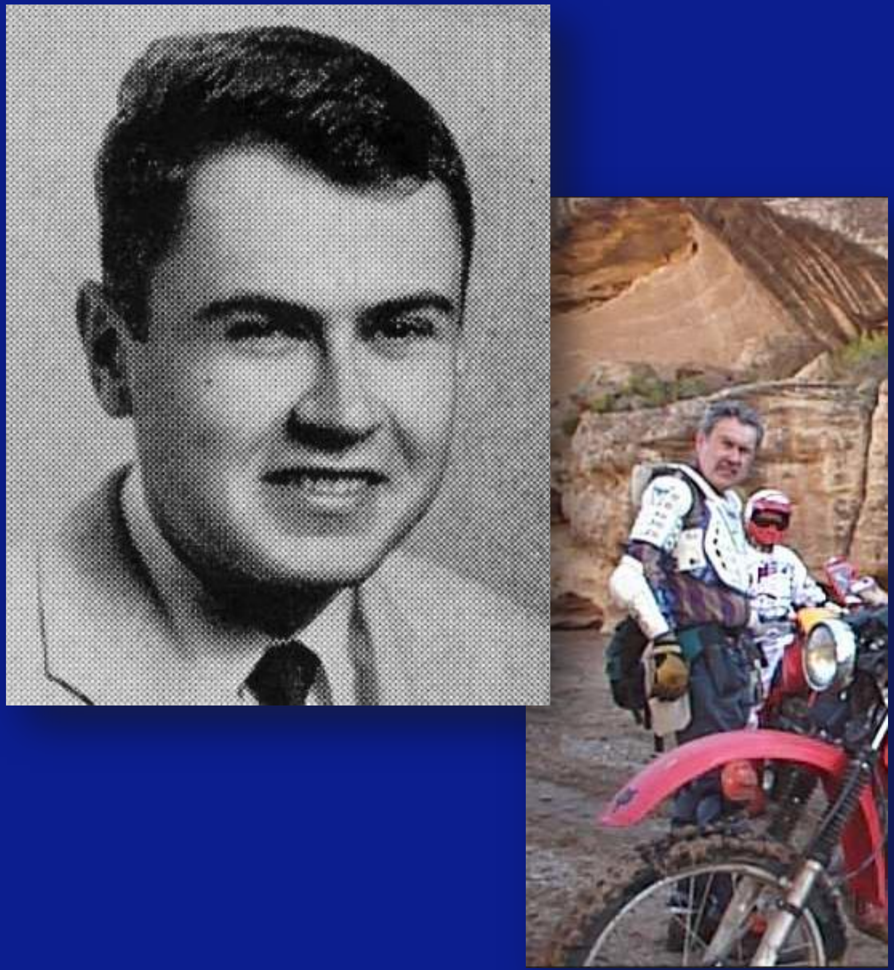
Conclusion

	First Generation	Second Generation	Third Generation
IP test control	Non-standard	Standard PHY wrapper, fully register based	Standard PHY wrapper, fully register based, improved internal test capability
Test Documentation	Verbal, difficult translation to SoC svc reg transactions, multiple test guides and limits per process node and fab site	Tabular, scalable, unified test guide and limits	Tabular, scalable, unified test guide and limits, standard setups
Test Documentation to RTL Verification	Non-Existent	Manual	Fully Automated from Test Documentation
Test pattern development	Manual – slow, error-prone (months)	Automated – fairly fast, accurate, simulated	Automated – very fast, accurate, scripted (week)
Tester code development	Manual, SoC-specific	Automated, SoC-generic	Automated, SoC-generic
Tester CZ development	Manual, SoC-specific	Separate, significant development effort	Integrated, minimal development effort
Tester CZ capability	Good	Better	Best, Tx Eq, Jitter Scope Mode
Lab Validation capability	Manual, SoC-specific	Standard Test Guide Based	Standard Test Guide Based, More Labs On-Line
Data analysis	ASCII datalogs, manual summaries, lengthy analysis	Semi - automated summaries, quicker analysis	STDF, automated summaries, fast analysis
Test issue resolution	Poor tester-to-lab correlation, slow, iterative	Excellent tester-to-lab correlation, fast, few iterations	Excellent tester-to-lab correlation, fast, few iterations
Propagating changes	Lengthy for both pattern development and bring-up	Reasonably fast	Acceptably Quick

Acknowledgements

- SerDes Design Team
- SerDes IP Wrapper Team
- Test and Product Development Teams
- Validation and Compliance Lab Teams
- Software Teams
- Applications Teams

Dedication



Richard (Dick) Adlhoch
Nov 1939 – Dec 2022

References

1. https://www.digikey.com/es/maker/blogs/2018/71st-anniversary-of-the-transistor
2. Common Public Radio Interface Specification 4.2, September 2010
3. Fibre Channel - Methodologies for Jitter and Signal Quality Specification – MJSQ, T11.2 / Project 1316-DT/ Rev 14.1, June 5, 2005
4. IEEE 1149.6TM: IEEE Standard for Boundary Scan Testing of Advanced Digital Networks Std IEEE 1149.6, 2003 and 2015
5. IEEE 1687TM - 2014: IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device, November 2014
6. IEEE 802.3zTM - 1998, June 1998
7. IEEE 802.3aeTM - 2002, August 2002
8. IEEE Std 802.3apTM - 2007, May 2007
9. IEEE Std 802.3baTM - 2010, June 2010
10. IEEE Std 802.3bjTM - 2014, June 2014
11. IEEE Std 802.3TM - 2022, May 2022
12. OIF-CEI-03.1: Common Electrical I/O (CEI) - Electrical and Jitter Interoperability agreements for 6G+ bps, 11G+ bps and 25G+ bps I/O, February 18, 2014
13. PCI-ExpressTM Base Specification 2.0, December 2006
14. PCI-ExpressTM Base Specification 3.0, November 2010
15. PCI-ExpressTM Base Specification 4.0 v1.0, September 2017
16. RapidIOTM Interconnect Specification Part 6: 1x/4x LP-Serial Physical Layer Specification 1.3, June 2005
17. Serial ATATM Revision 2.6, February 2007
18. Serial ATATM Revision 3.0, June 2009
19. Serial ATATM Revision 3.2, 2013