SERDES Rx CDR Verification using Jitter, Spread-spectrum clocking (SSC) stimulus

Somasunder Sreenath, Raghuram Kolipaka, Chirag Shah, Parag Lonkar

Cadence, Bangalore

accellera SYSTEMS INITIATIVE

cadence

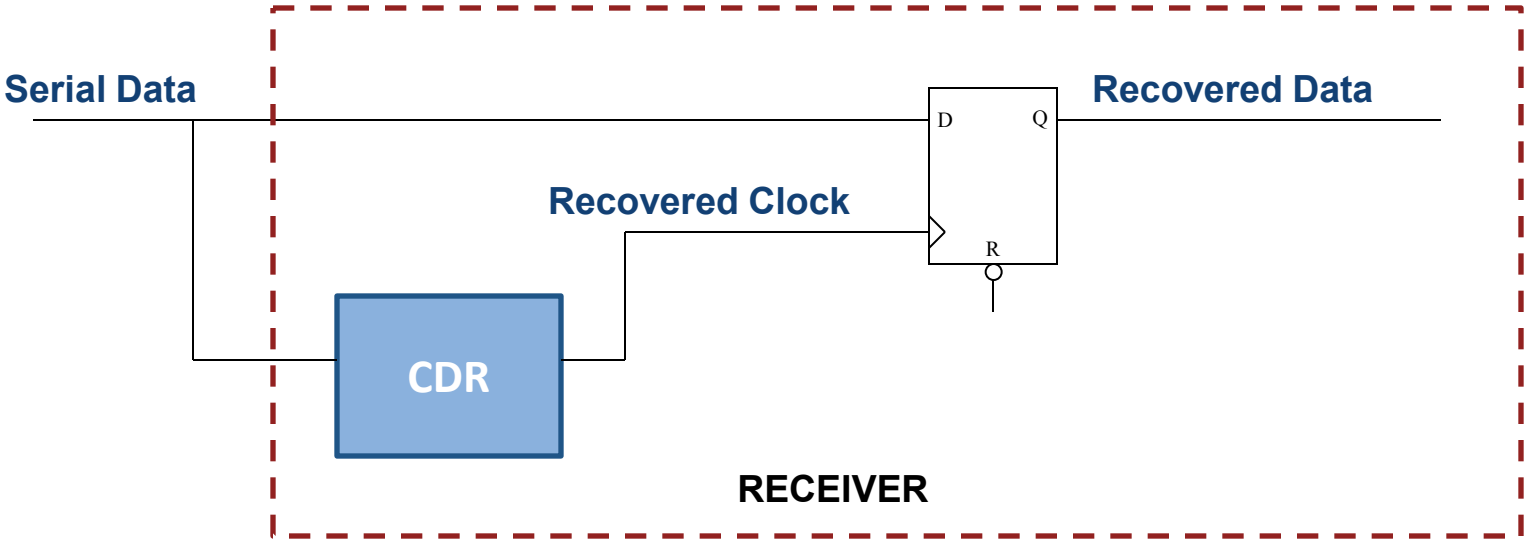
2014 DESIGN AND VERIFICATION DVCON CONFERENCE AND EXHIBITION INDIA

Outline

- CDR Overview
  - Jitter Tolerance
  - SSC and ppm Offset
- Verification Environment
- Verification Methodology
  - Deterministic Jitter (DJ): Period Jitter ; Absolute Jitter
  - Random Jitter (RJ)
- Results

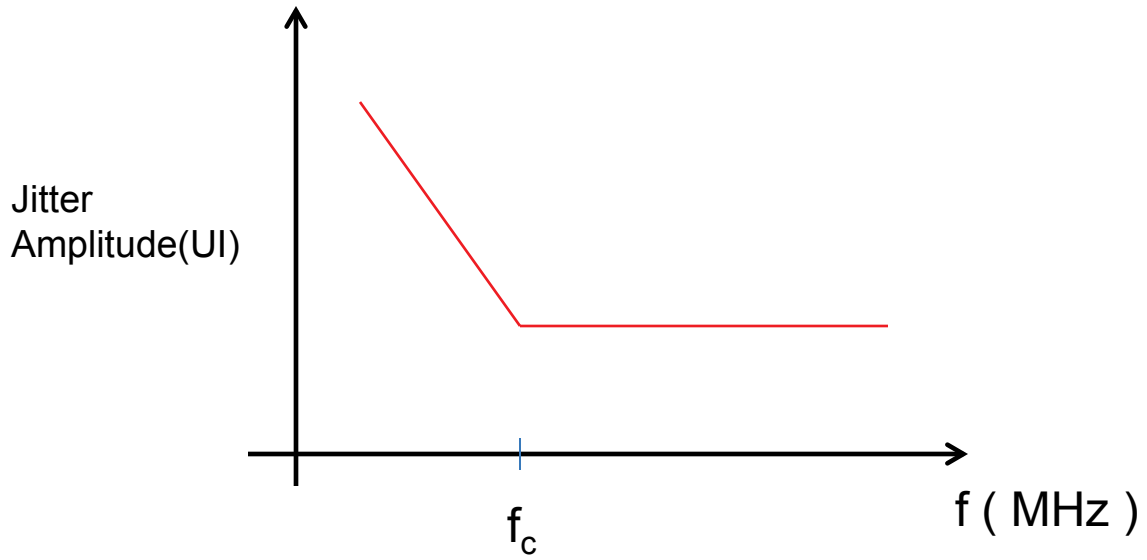
Clock Data Recovery (CDR)

- SERDES (USB3.0, PCIe, SATA, DP, MPHY) use embedded clock transmission
- CLOCK recovered from data at the far-end - CDR
- 8b-10b encoding – Guarantees transition density



Jitter Tolerance

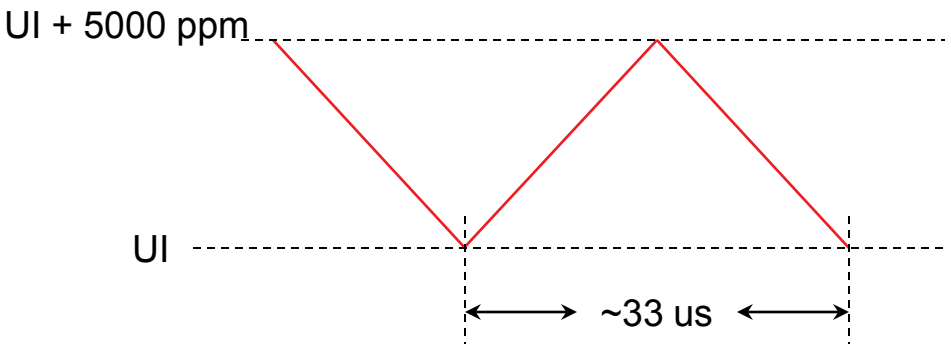
- CDR – Low Pass characteristic.
  - Tracks Low Frequency Jitter.
- Corner Frequency (f_c) – ~ 5-10 MHz range



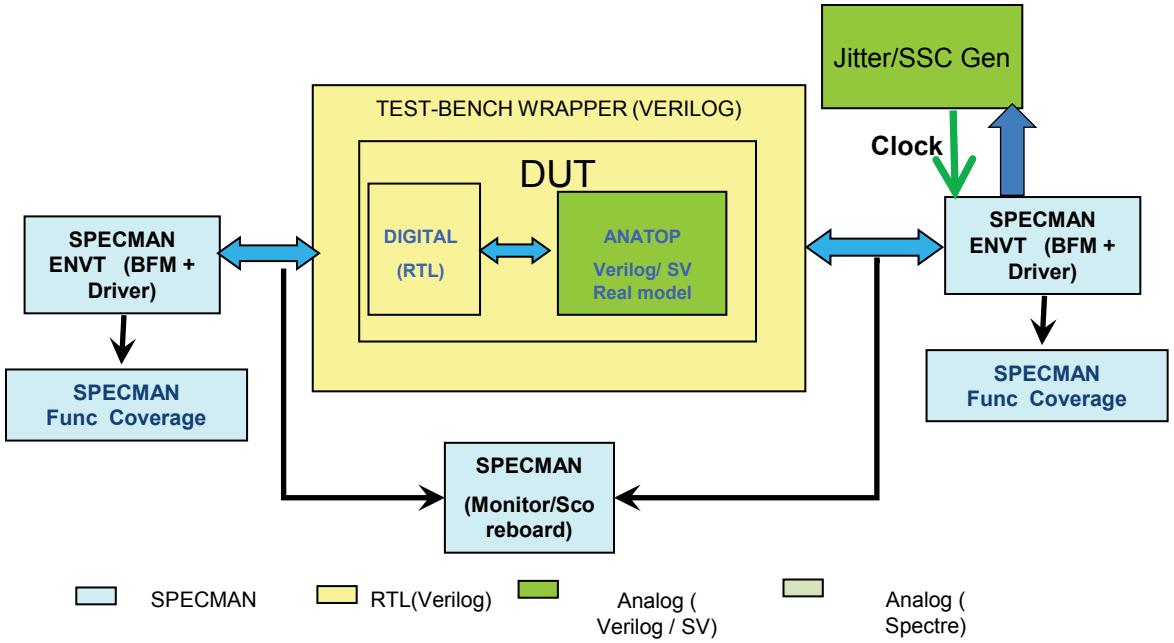
USB3.0, PCIe, SATA, MPHY have similar requirements

SSC and ppm Offset

- ppm offset – Frequency variation across 2 sides of the link
  ➔ 300 ppm : 300* Freq/10^6
- SSC – Modulation Rate : 30 to 33 kHz
  Frequency Deviation : -4000 to -5000 ppm (**DOWN-SPREAD)**
- USB3 : +/- 600 ppm offset, upto -5000 ppm SSC
- PCIE (common refclk) : +/- 600 ppm offset
- PCIE (Separate refclk) : +/- 600 ppm offset, upto -5000 ppm SSC
- SATA : +/- 700 ppm offset, upto -5000 ppm SSC



Verification Environment



- Serial-line UVC uses clock with Jitter + SSC
- AMS : Analog portion replaced by Spectre Netlist

Verification Methodology

- Clock Generation
  - Jitter – Deterministic (DJ) sine-wave jitter and RJ
  - SSC – Spread spectrum triangular wave with Downspread
  - ppm offset on the clock w.r.t Internal PLL clock
- DJ Flavours
  - Absolute Jitter
  - Period Jitter
- Verification/Checking
  - Data Integrity
  - Recovered clock sampling position w.r.t centre of the EYE.

DJ-Period Jitter

- Sine-wave generated using SV DPI
- CLOCK period modulated based on the sine-wave
- Basic Equations:
  - **jitter = AMPL * sin(2π*freq*time_sec)**
  - var_period = UI/2 + jitter
  - CLOCK = #(var_period) ~ CLOCK

\+ Easy Implementation , less no of parameters

\- Inconsistency in jitter addition in sine-wave cycles

\- Slow ppm drift with time

\- Cannot add multiple-tones easily

DJ - Period Jitter

- No of samples in the +ve and –ve half-cycles **NOT SAME**
- **Non-zero Residue** at the end of each sine-wave cycle
- **Variable Step Sizes** ➔ difference in no of of samples ➔ ppm accumulation

Figure A: Jitter Samples and Residue at Zero

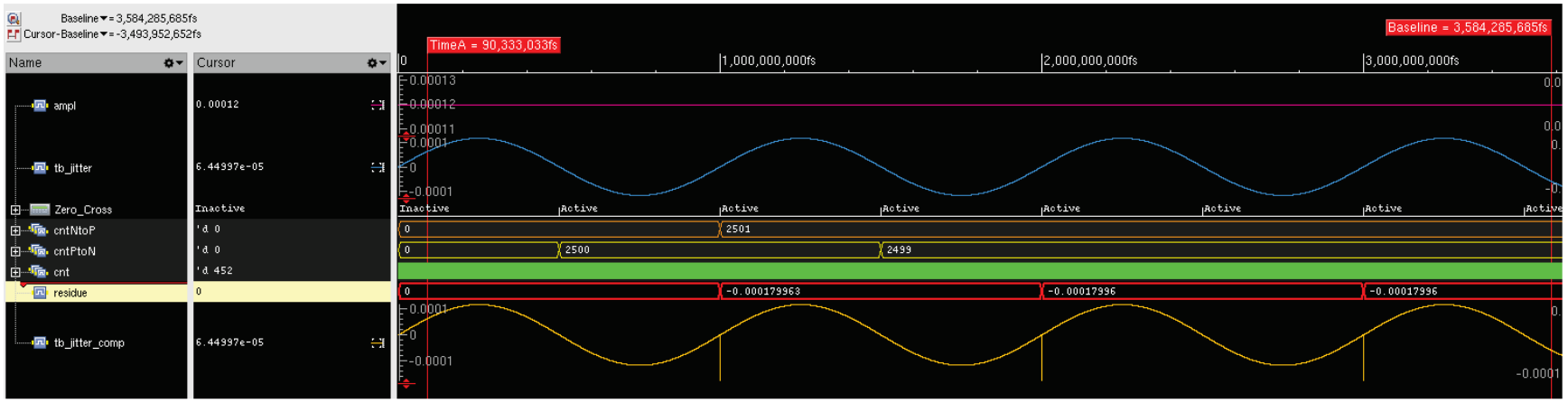
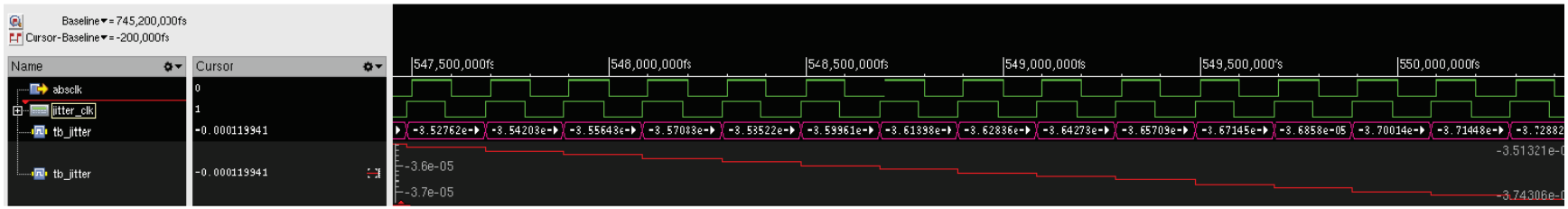
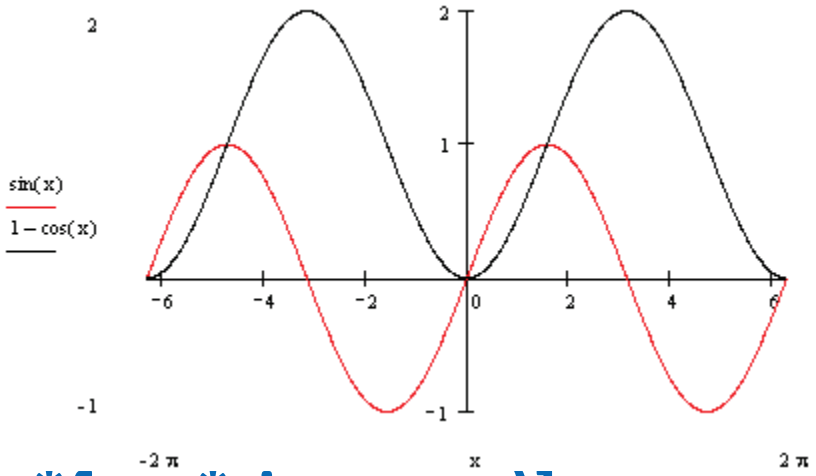


Figure B: Jitter Step Sizes



DJ- Absolute Jitter

- Use [1- cos(x)] function
  - Jitter is always >= 0
- Use Transport Delay to generate Jitter clock



- **jitter = UI*AMPL * [1- cos(2π*freq*time_sec)]**
- **jitter_clk <= #(jitter) abs_clk;**
  - abs_clk : Free Running clock with ppm Offset/SSC
- For multiple-tones, generate jitter1,2,..
  jitter_clk <= #(jitter1 + jitter2+…) abs_clk;

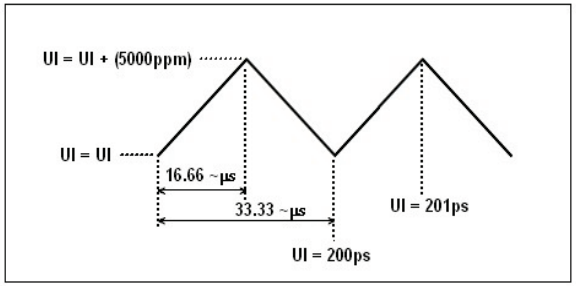
Random Jitter(RJ)

- Based on Gaussian Distribution
  - $f(x,\mu,\sigma) = (1 / \sigma \sqrt{2\pi})* e^{-(x-\mu)^2/2\sigma^2}$
  - x = random seed
  - μ = mean
- Generated using $dist_normal system task
  - **$dist_normal** (seed, mean, standard_deviation) ;
- **RJ = $dist_normal(myseed, 0, σ);**
  jitter_clk <= #(jitter1 + jitter2+… + RJ) abs_clk;

Clock Tolerance Compensation

USB3.0			
UI(ps)	200		
ppm	5600		
1SKP OS	3540		19.824
4 SKP OS	14160		79.296
PCIE Gen2			
UI(ps)	200		
ppm	600		
1SKP OS	15360		9.216

Worst-case pointer movement in terms of UI



USB3 : SKP Ordered set : K28.1 K28.1
PCIE : SKP Ordered set : K28.5 K28.0 K28.0 K28.0

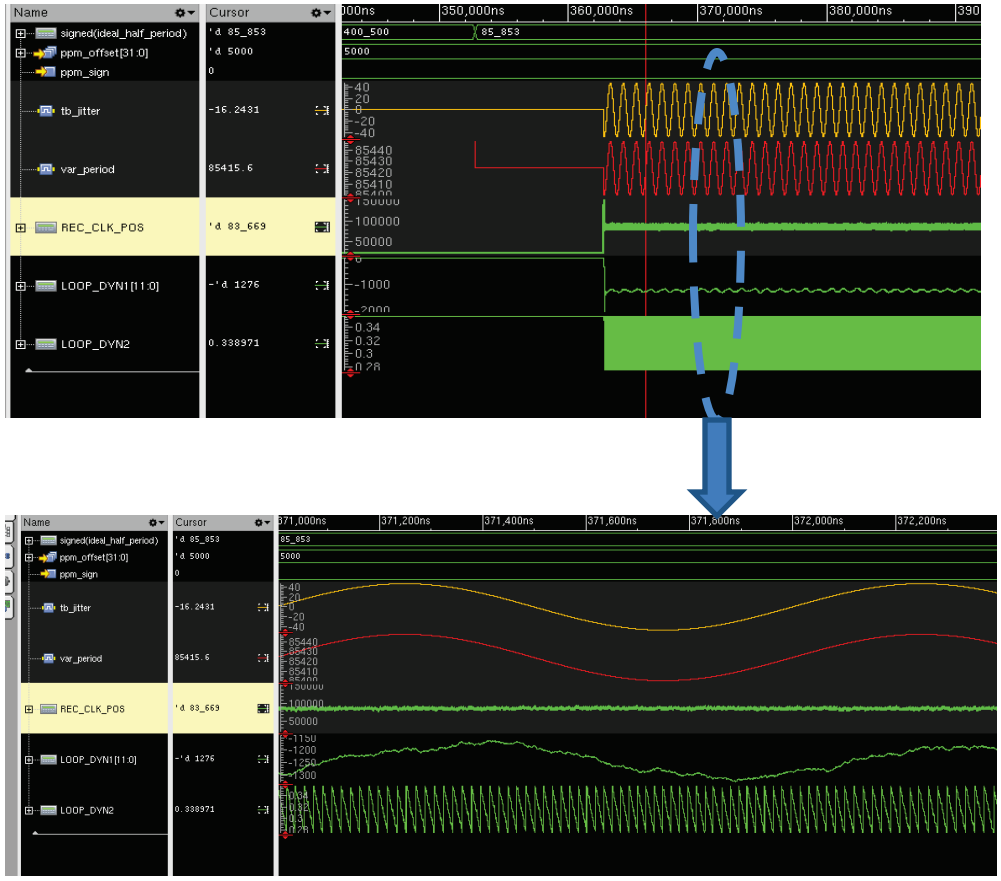
USB3 PHY can add/remove SKP OS when it receives an SKP OS in half-full EB mode
USB3 PHY can add SKP OS whenever it is close to underflow in nominal empty mode

PCIE : PHY can add/remove one SKP(K28.0) when it receives an SKP OS
PHY input : SKP OS may have COM followed by 1 to 5 SKP

RESULTS

- Coverage/Test-scenarios  **> 100000 simulations**
  - **ppm Offset**
  - **JITTER :** NO JITTER, SINGLE-TONE JITTER, 2-TONE JITTER, 2-TONE-JITTER+RJ
  - **JITTER FREQ :** 1, 2,4,6,8,10,100,500,1200 MHz
  - **JITTER AMPLITUDE :** Steps of 0.1UI (better resolution for higher freq range)
  - **SSC Modulation Rate** : 30 KHz , 33 kHz
  - **SSC Frequency Deviation** : NO SSC, -4000 ppm, -5000 ppm
  - **DATA PATTERN** : TSEQ, TS1, TS2, CP0, Clock pattern(D10.2),MPHY Sync pattern(D10.5,D26.5), SATA ALIGN, PRBS7, PRBS31,CRPAT,CJTPAT, Random data
  - **SSC PHASE Difference** between the internal and external clocks

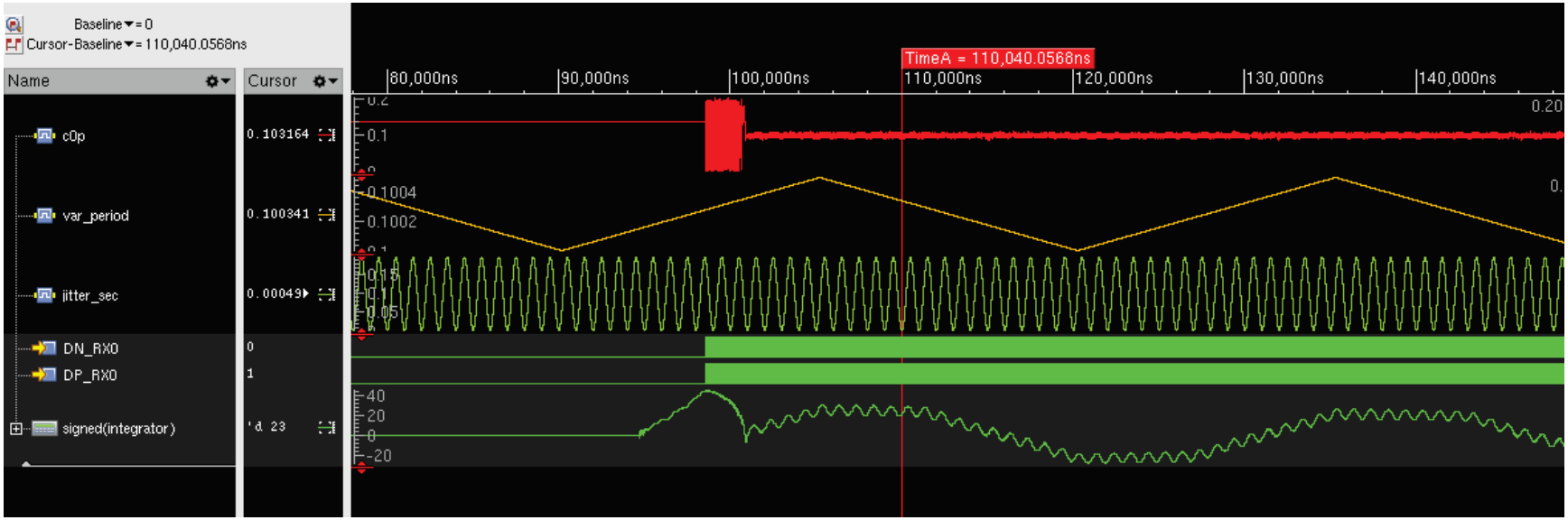
RESULTS



- Scenario
  - Offset : 5000 ppm
  - 1MHz single tone sine-jitter
  - Data rate of ~5.8 Gbps
- Loop tracking the offset and sine wave(LOOP_DYN1)
- Recovered clock at the centre of the eye(REC_CLK_POS)
- ppm offset – Recovered clock position sweeps like a ramp across quadrants(LOOP_DYN2)

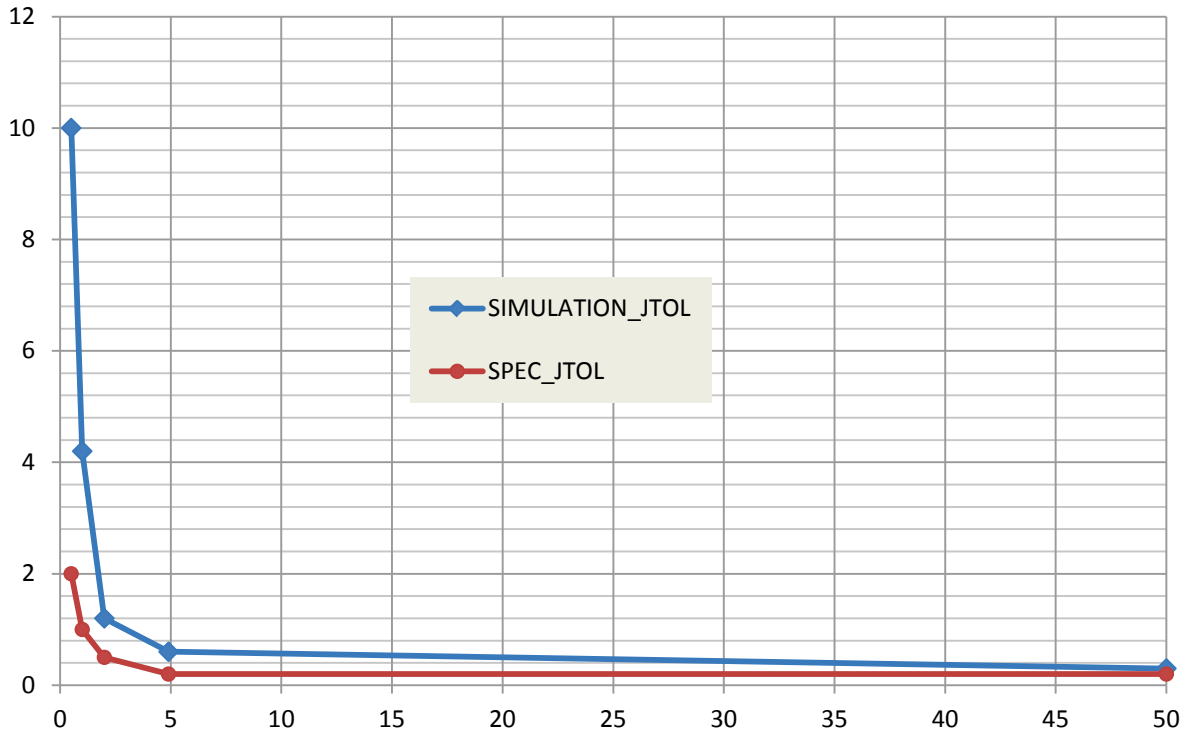
Simulation with SSC+Jitter

- Simulation Scenario
  - 5 Gbps Rata Rate
  - -5000 ppm SSC Down-spread
  - Single-tone DJ : 1*UI @ 1MHz



JITTER TOLERANCE (JTOL)

- Simulations run with SSC, ppm offset, random phase
- For each jitter-frequency – Sweep the jitter amplitude(UI) to determine the failure point
- Multiple simulations for the same jitter UI to get more confidence



Questions

Spread Spectrum Clocking (SSC)

- Triangular in nature with definite SSC Modulation Frequency and Deviation.
- Expression :
  - ssc_os = step_ssc * ((ssc_ppm)/(10000000.0*(ssc_freq/4)));
    - ssc_ppm : -5000 ppm for SATA,-5000 ppm , -4000 ppm for USB3
    - ssc_freq : 30 KHz – 33 KHz
    - step_ssc : varies from 1 to 1000 in a loop
    - var_period = UI/2 + (offset_ppm/(10000000.0)+(ssc_os)) ;
  - abs_clk = #(var_period) ~ abs_clk;