2019
DESIGN AND VERIFICATION
DVCON
CONFERENCE AND EXHIBITION
INDIA

Challenges, Complexities and Advanced Verification Techniques in Stress Testing of Elastic Buffer in High Speed SERDES IPs

Kamesh Velmail, Samsung, Bangalore, India (*kamesh.vel@samsung.com*)

Suvadeep Bose, Samsung, Bangalore, India (*suvadeep.b@samsung.com*)

Parag Lonkar, Samsung, Bangalore, India (*parag.lonkar@samsung.com*)

Somasunder Sreenath, Samsung, Bangalore, India (*soma.ks@samsung.com*)

Ankit Garg, Samsung, Bangalore, India (*ankit.garg@samsung.com*)

Abstract— **High Speed Interface (HSI) protocols like USB, PCIE, SATA, etc. are used in a wide range of applications today from Mobiles to PCs. Hence design and verification of these IPs constitute a fair share of market in the semiconductor industry. All of these serial IPs would generally have a dedicated Physical layer (PHY). Among other tasks, the PHY is also responsible for managing Clock Tolerance Compensation (CTC) [2] between the far-end and receiver clock domains to ensure data-integrity in the Rx data-path. To address this issue, a special buffer called the Elastic Buffer (EB) is generally implemented in the receiver path of the PHY to handle the frequency difference between the bit rates at the two ends of a channel. The EB plays a vital role in maintaining correct and uninterrupted communication between the two link partners. As a result special attention needs to be given while designing and verifying EB in order to ensure efficient functionality of the RX Datapath. The paper provides a brief understanding about the functionality of Elastic Buffer and then focuses on the verification challenges, strategies, generation & choice of stimulus needed for stress testing the elastic buffer. It includes the methodology used to create frequency deviation between the read and write pointers of the buffer by using SSC/PPM/jitter and covers various SKP handling scenarios and impact of regular/irregular SKP insertion on EB operation.**

Keywords—HSI; USB; PCIe; SATA; PHY; PCS; PMA; CTC; Elastic Buffer; SKP;

I. INTRODUCTION

The Physical layer (PHY) of High Speed SERDES IPs would typically comprise of Transceiver, clock generation and protocol handling blocks. The PHY is generally partitioned into two blocks- the **P**hysical **C**oding **S**ub-layer (PCS) which is the protocol dependent portion and the **P**hysical **M**edia **A**ttachment (PMA) Layer which handles serial data and clock generation. Most HSI links operate on independent clocks on Host/Device ports which result in slight variations in the frequencies. So it becomes responsibility of the receiver PHY to accommodate any such variations in the incoming bit streams from the far end. This is taken up by PCS within PHY. The PCS utilizes special symbols (E.g.: SKP for USB, PCIe, and ALIGNs for SATA) [2] to manage the EB read/write pointers dynamically.

The paper has been organised as follows: section II describes in brief, the working of the elastic buffer and provides some information on the range of clock tolerance and SKP insertion parameters as discussed in PIPE and USB specs; section III talks about challenges seen in verifying the elastic buffer logic and verification approach adopted to deal with those challenges. Section IV showcases the results obtained from various EB verification scenarios.

II. ELASTIC BUFFER FUNCTIONALITY AND MODES

A. *Elastic Buffer Functionality*

The Elastic Buffer (EB) is essentially an asynchronous FIFO which is maintained in the PCS of PHY layer. It acts as a bridge between the 2 separate clock domains – the Recovered Rx clock from serial bit stream on one side and the local PLL clock on the other. It is usually the job of PMA to deserialize the serial data and recover the clock out of it.

This recovered clock would have the characteristics of the far end PLL or transmitter's PLL clock which was used to generate the bit stream. It can also be influenced by other factors like jitter which comes into play while the bit stream is being transmitted through the link. Due to these and other similar factors a slight frequency variation in the de-serialized recovered data is observed which can offset the data throughput to the link.

The de-serialized data is converted to aligned symbols within PCS and stored into the Elastic Buffer at the write clock edges. This write clock is basically a symbol clock derived from recovered bit clock. The data which is latched into the elastic buffer is read, decoded and provided to the link at a rate which is determined by the read clock from local PLL.

The EB, just like any other FIFO, has the possibilities of running into overflow or underflow. When Overflow or Underflow occurs, the Link layer may go through a process of recovery and Re-training as specified by the protocol. This process takes up a considerable time ultimately resulting in reducing bandwidth and throughput of the PHY than expected. Hence effective bandwidth of a PHY often depends on the careful and meticulous design of the Elastic Buffer.

Protocol mandates use of a special skip symbols (SKP for USB and PCIe, Align for SATA) which are periodically inserted in the data stream by the transmitter. These symbols are often not used by link partner and are usually dropped at the receiver's end. The EB plays a specific role in handling these SKP patterns within PCS. It can either add or remove SKPs as stipulated by the protocol and can adjust its write and read pointers. So a proper implementation of EB, along with guaranteed SKP patterns by the transmitter, together can ascertain that no data is lost due to overflows and underflows.

As per PIPE protocol, in USB, the Elastic Buffer can operate in 2 modes Nominal Half-full Buffer mode (the PHY tends to keep the EB as close to half-filled as possible) and Nominal Empty Buffer mode (the PHY attempts to keep the elasticity buffer as close to empty as possible)

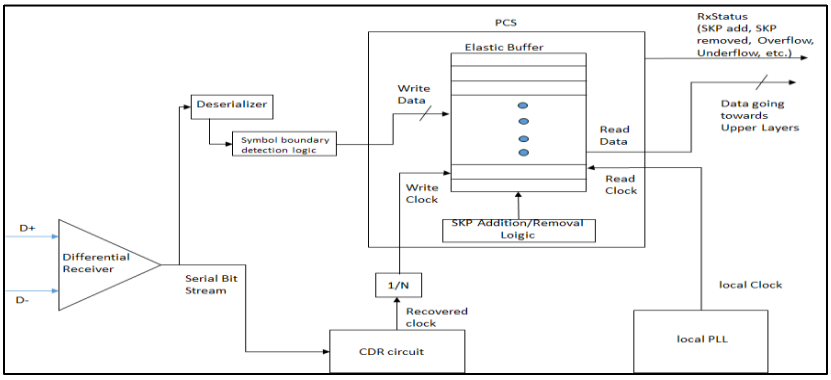


Fig.2.1. Elastic Buffer Functional Diagram

B. Protocol limits for clock frequency variation and skip insertion/removal logic

The number of special symbols which are inserted into the data stream by the transmitting device is dictated by the protocol specified maximum SSC, PPM offset and jitter margins between the two clocks on the opposite sides of the link.

The below table shows these values for the 3 protocols- USB, PCIe and SATA

Protocol	SSC Modulation rate	SSC frequency Deviation	PPM offset	Max. frequency deviation from ideal frequency
USB	30 to 33 KHz	upto -5000ppm	-300ppm to +300ppm	+300 to -5300ppm
PCIe	30 to 33 KHz	upto -5000ppm	-300ppm to +300ppm	+300 to -5300ppm
SATA	30 to 33 KHz	upto -5000ppm	-350ppm to +350ppm	+350 to -5350ppm

Table 2.1. Frequency deviation for different HSI protocols

As table shows, the Worst case frequency difference for USB among the transmitter and receiver clocks can be as high as 5600 ppm because the clocks on both sides can vary between 300ppm to -5300ppm from the ideal frequency. Apart from SSC and fixed PPM offset, there can be various sources of Jitter in the end to end connection which will also affect the bit clock period slightly from the actual UI width. The amount of tolerable jitter is limited by the desired bit error rate performance of the channel.

In order to account for the maximum CTC range, the USB spec.[1] mandates the following:

- For Gen1, (5 Gbps speed) on an average, for every 354 symbols, a gen1 SKP ordered set (containing 2 SKP symbols) should be inserted by the transmitter which can go to 8 SKPs per 1416 symbols.

- For Gen2, (10 Gbps speed), on an average for every 40 data blocks, a SKP Ordered Set should be transmitted which can go to 3 SKP Ordered Sets per 120 data blocks.

However, SKP ordered sets cannot be transmitted in all places like while transmitting training ordered sets or data packets, which create worst case scenarios for Elastic Buffers in receivers.
Related specifications can be referred for greater details on SKP symbols and rules.

III. VERIFICATION APPROACH

To ensure robust verification of EB (including Receiver data-path) multi-pronged approach including modelling and enhancements in DV Infrastructure is necessary. Elastic Buffer, like any other FIFO, needs to be tested with various real time scenarios which can create maximum variations in the write and read pointers, thus stress testing it. This can be achieved with either one or the combination of the following factors:

- Write Clk of Elastic Buffer (Wclk_EB): As this clock is derived directly out of the recovered clock from serial data, any change in the bit widths of incoming data can alter the period of Wclk_EB, making write pointers move fast or slow accordingly.

- Read Clk of Elastic Buffer (Rclk_EB): This clock is derived directly out of the local PLL output clock. PLL are behavioural models with usually having an optional configurations to enable internal SSC. Using this features different values can be tried to make the Rclk_EB clock slower than usual

- SKP Symbols in the received data stream: This factor differentiates EB from a normal FIFO. As discussed above, the movement of the read and write pointers can be controlled by varying number, length and position of SKP symbols/Ordered sets.

The following section, provides more insight on how to employ these 3 factors to create best suited tests for EB

A. TB Details

The UVM based Test-bench under discussion here is created primarily for verifying the USB based PHY. This can easily be extended in principle to other PHYs with similar features. It can either be configured as Host side or Device Side. The PHY talks to Link on one side using PIPE interface and with the Far-End Receiver on the other side using serial interface.

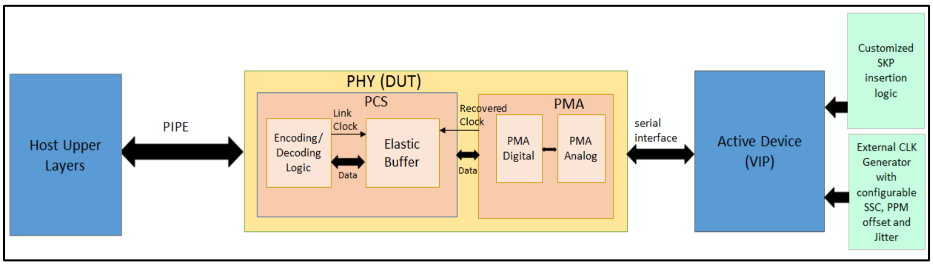


Fig.3.1. Test-Bench Setup for Elastic Buffer Stress Testing

Major components:

- External Clk Generator: This block has the capability to alter the ideal UI width of the transmitted serial bit data from the Active Device.
- Customized SKP Insertion Logic: This block has the required capability to adjust position, number & length of SKPs as per protocol and insert it accordingly to the serial data stream of PHY.

B. External Clk Generator

The External Clk Generator [3] is a portable and reusable module which has been developed in System Verilog (SV). This block provides a bit clock which can be used by the Active Device to serialize and transmit the data. The clock can be modulated to change its period. Hence the UI width of the serial bit data transmitted over it also gets modulated accordingly. This variation in the UI of bits eventually translates into recovered clock at the receiver and hence impacts the write pointers of EB.
The External clock is designed taking into account the following factors:

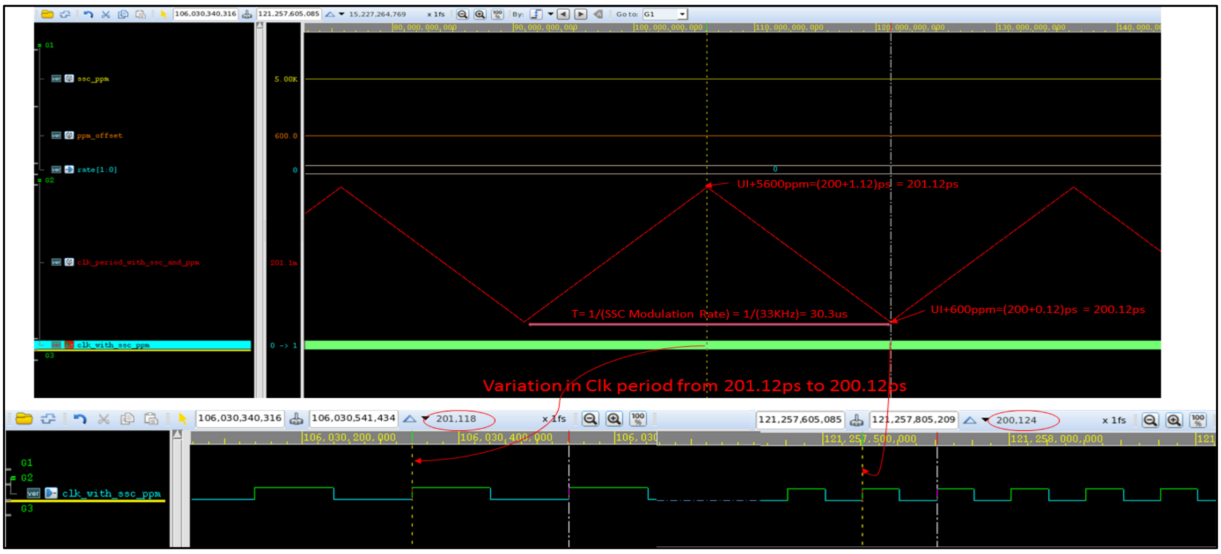
- **S**pread-**S**pectrum **C**locking (SSC) : The SSC modulation rate can be randomized between 30 to 33 KHz and the max frequency deviation can be randomized between 4000ppm to 5000ppm. If SSC is OFF on the

DUT side during simulation, then in order to account for the DUT's SSC(which will be present in reality) , we can either deviate the External CLK SSC in positive or negative direction, so that the worst case net frequency deviation between the read and the write clocks is achieved.

- PPM offset : As per protocol requirement, absolute value of PPM offset has been randomized in between 0 to 300 ppm in either direction. However in simulation, the DUT side may not have a PPM offset modelled. Hence in order to create the effect of PPM offset on frequency difference between read and write clocks, the external clk may be configured to have an offset up to 600ppm in either direction.

Here is a small example to understand the impact. If maximum frequency deviation due to SSC =5000ppm (down spread), then the UI should rise from 200ps(Ideal UI) to 201ps [Ideal UI + 200*(5000/10^6)] and then again fall back to 200ps.

In Gen1 speed , suppose during simulation, there is no ppm offset on DUT side, them in order to account for a possible -300ppm offset on the read clk and +300ppm offset on write clk, we can configure the external clk with a fixed PPM offset of +600ppm(or 200*(600/10^6) = 0.12ps. So the UI profile due to the effect of SSC + ppm offset would rise from 200.12ps to 201.12ps



UI+5600ppm=(200+1.12)ps = 201.12ps

T= 1/(SSC Modulation Rate) = 1/(33KHz)= 30.3us

UI+600ppm=(200+0.12)ps = 200.12ps

Variation in Clk period from 201.12ps to 200.12ps

Fig. 3.2. External Clk UI profile with SSC and PPM offset

- Jitter: In this case, Periodic Jitter (Pj) [5] has been modelled since it is easy to deploy and test. One can also model the other components of both Deterministic and Random Jitter according to Jitter budgeting requirements.

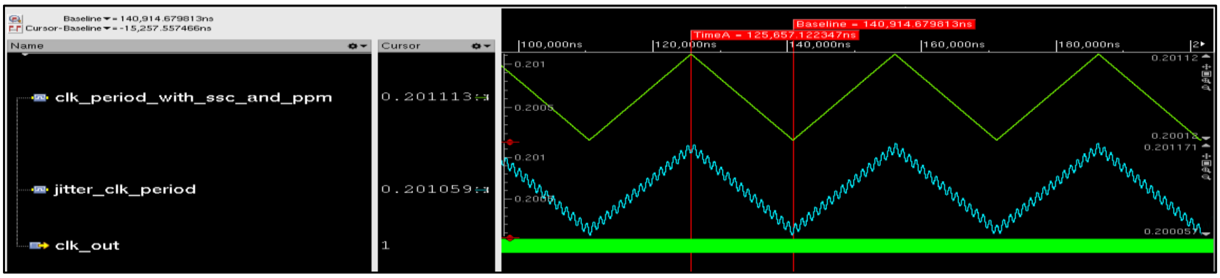


Fig.3.3. External Clk UI profile with added Jitter

All these factors are parametrized in such a way that they can be easily tuned from the Test bench sequence.

C. *Customizable Skip Insertion Logic*

The Skip insertion logic follows the respective protocol for inserting SKP symbols or ordered sets. The transmitter sends SKP ordered sets at an average of every 354 symbols in Gen1 and 40 blocks in Gen2, while the worst case scenario will be inserting 4 SKP Ordered sets after 1416 symbols in Gen1 and three SKP Ordered Sets after 120 blocks in Gen2.

The worst case scenario coupled with maximum frequency deviation between read and write clock may be used to stress test the Elastic Buffer.

Hence the tests that are written to test the Elastic Buffer capability targets to cover different numbers and positions /lengths of SKP symbols/Ordered sets.

USB Gen1

No. of symbols before SKP ordered set	No. of SKP symbol inserted
353, 354	2
707, 708	4
1061, 1062	6
1415, 1416	8

USB Gen2

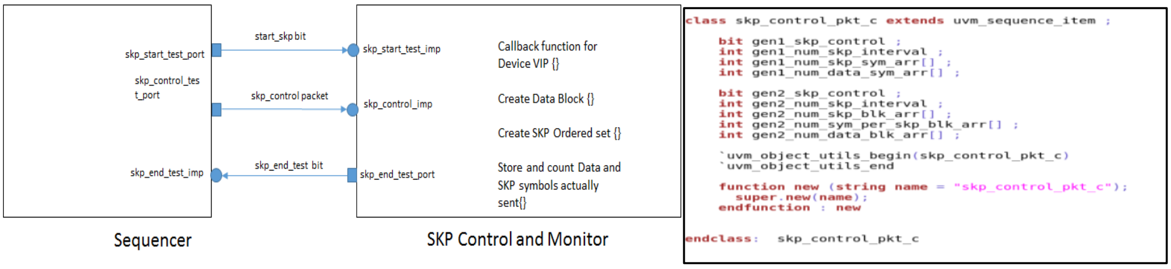
No. of data blocks before SKP Ordered sets(d)	No. of SKP ordered sets inserted (s)
39 ,40	1
79.80	2
119,120	3

Table 3.1. SKP insertion cases for USB Gen1 and Gen2

In the case of USB Gen1, symbols coming before SKP symbols+ SKP symbols constitute a Skip Interval. For USB Gen2, the receiver may receive multiple SKP ordered sets with variable lengths at the end of a certain number of data blocks– the number of SKP symbols inside a SKP ordered set received at the DUT has been randomized in the range of all the numbers from 12 to 36 in multiples of 4. We consider a chunk of Blocks followed by single or multiple SKP ordered sets of variable length as a Skip Interval in the case of Gen2. E.g. 3 Skip intervals means there will be 3*(d Data blocks + s SKP ordered sets) where each SKP ordered set may vary in length. The parameters relating to SKP insertion viz. no. of data blocks before SKP, No. of SKP ordered sets, No. of SKP symbol in each SKP ordered set, and No. of Skip Intervals can either be completely randomized in the test, or can be partially or completely fixed according to the test requirement. By following this approach, all the possible combinations can be covered.

Callbacks have been used to customize the size of the SKP ordered sets and their positions in the data stream. The starting point for counting SKP intervals is controlled from the sequence. The sequencer and the SKP control_monitor are connected through multiple analysis ports as illustrated in fig.

The calculation of SKP intervals is started as soon as start_skp bit is made 1 from the sequence. Inside the SKP control and monitor block, using Device Callbacks, we are monitoring the number of symbols/blocks and SKP symbols/Ordered sets sent till now and initializing our parameters accordingly. Then we create symbols/blocks using custom function create_data_block{} and start injecting them into the Transmission Queue of the VIP. Also we keep a count of the symbols/blocks being injected. When sufficient number of blocks/Symbols for a specific SKP interval has been sent, we inject desired number of SKP symbols/Ordered sets (as targeted or randomized) from the sequence into the Transmission Queue of the VIP. The SKP ordered set of required length is made using the custom function create_skp_block{} All the information regarding no. of skip intervals, no. of SKPs in each SKP ordered set, No. of data symbols/blocks coming before SKP ordered SET are all assigned to the skp_control packet which is ported from the sequencer to the SKP Control and Monitor block through skp_control_test analysis port.

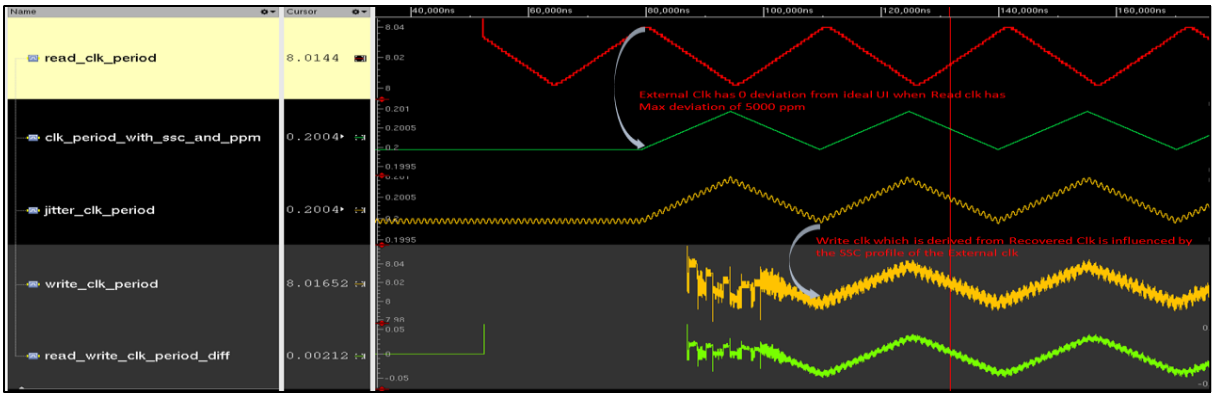


```
class skp_control_pkt_c extends uvm_sequence_item ;

    bit gen1_skp_control ;
    int gen1_num_skp_interval ;
    int gen1_num_skp_sym_arr[] ;
    int gen1_num_data_sym_arr[] ;

    bit gen2_skp_control ;
    int gen2_num_skp_interval ;
    int gen2_num_skp_blk_arr[] ;
    int gen2_num_sym_per_skp_blk_arr[] ;
    int gen2_num_data_blk_arr[] ;

    `uvm_object_utils_begin(skp_control_pkt_c)
    `uvm_object_utils_end

    function new (string name = "skp_control_pkt_c");
      super.new(name);
    endfunction : new


endclass:  skp_control_pkt_c
```

Fig. 3.4. SKP insertion methodology

D. Corner Scenarios and Coverage Space

Some of the corner scenarios for testing the capability of EB include shifting the UI profile of the recovered clk or write clk of the EB with respect to the local clk of the DUT or read clk of the elastic buffer. For this case as soon as required condition(such as RxValid goes high) is satisfied on the DUT's end, we start monitoring the DUT's local clk's UI profile and start generating increasing the time period of external clk from required position of the DUT clk. The UI profile of the Device VIP's transmitted data in term affects the recovered clk at the DUT or the write clk and this happens approximately at the same time. This technique is especially useful when SSC is "ON" on both the Link Partners (DUT and Device VIP) during simulation. This technique can also help us in reaching the worst case frequency difference of 5600ppm between the two clocks. This process is further illustrated in the figure below:



Name	Cursor
read_clk_period	8.0144
clk_period_with_ssc_and_ppm	0.2004▸
jitter_clk_period	0.2004▸
write_clk_period	8.01652
read_write_clk_period_diff	0.00212

External Clk has 0 deviation from ideal UI when Read clk has Max deviation of 5000 ppm

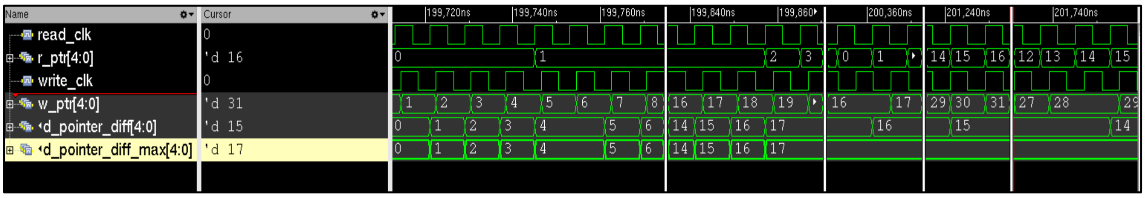
Write clk which is derived from Recovered Clk is influenced by the SSC profile of the External clk

Figure 3.5. Phase shifting of write_clk period profile w.r.t. the read_clk period profile to create max. deviation

Also during silicon testing, lot of simulation request for receiver path tend to come to DV team. So flexibility of running long tests, with highest or at times 'custom' values of SSC/Jitter and SKP combination is needed to re-create scenarios.
In order to thoroughly test all possible scenarios, functional Covergroup including coverpoints for configurable parameters related to External clock and SKP insertion were created. Also to test all the related CLK and SSC combinations, proper cross coverages were also created. Since the Verification environment tends to achieve complete flexibility in terms of all the configurable parameters proper switches were also created so that one or multiple CLK features such as SSC, PPM or Jitter can be turned on or off for particular directed or corner scenarios.

E. Debugging Techniques

- Monitoring movement of Read and Write pointers of Elastic Buffer and EB depth: The Read and write pointers of the Elastic Buffer are being probed and their difference is also being plotted in order to know at a certain instant how many buffer positions are filled and are left to be read. We can also compare this parameter with the Difference in Time period of the read and write clock in order to obtain a correlation between them and understand the buffer behaviour. Another parameter, the read-write pointer difference max is also being plotted. This helps us in identifying what was the maximum difference between read and write pointers reached throughout the simulation. If we are running a simulation with worst case frequency deviation between the read and write clks



Name	Cursor
read_clk	0
r_ptr[4:0]	'd 16
write_clk	0
w_ptr[4:0]	'd 31
d_pointer_diff[4:0]	'd 15
d_pointer_diff_max[4:0]	'd 17

Figure 3.6. Variation of read-write pointers and associated difference

- Plotting Recovered clock's UI width with time and local clock's UI width with time; Taking difference of these values and plotting it with time, we get the net Deviation in between the clock frequencies. This helps us to track down the point where we have maximum net frequency deviation. This can help us reaching worst case scenario by tuning Transmitter's SSC profile accordingly
- Monitoring no. of SKP symbols entering the EB, no. of SKP symbols getting added/removed by PCS and finally how many SKP symbols are reaching the MAC: This can help n understanding whether our elastic buffer is behaving as expected. By monitoring the flow of SKP through DUT we will be able to know this. We will get the number of SKPs coming from partner device from our Serial monitor. Similarly, the monitor on the pipe-interface gives us the number of SKPs which are being transmitted from PHY to Link. By subtracting these two numbers, we can get the number of SKPs added, or removed by DUT and also check whether SKP removed or added status is correctly reflected on the RxStatus.

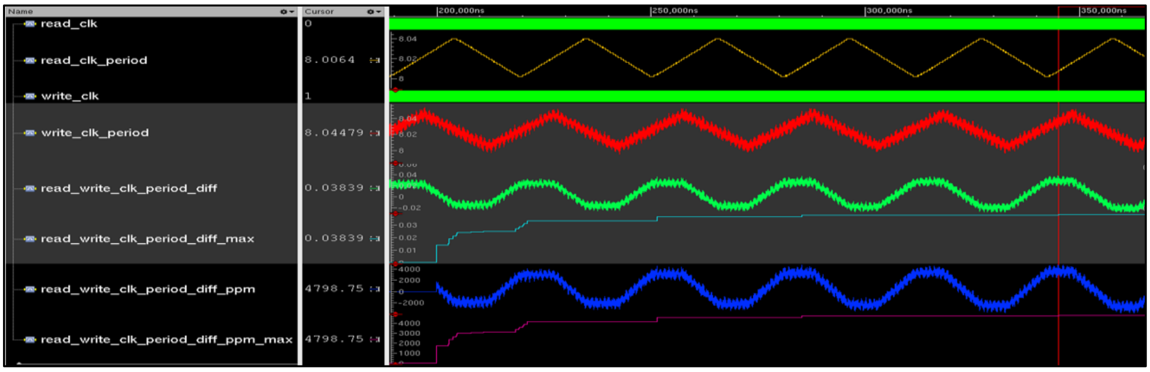


Figure 3.7. Variation of read-write clks and associated deviation

- SKP prediction methodology :

According to USB spec, there can be a worst case frequency deviation of 5600ppm between the read and write clks. Therefore, if this worst case frequency deviation is maintained throughout, after 1/ (5600 ppm) =178 clks, there will be a difference of 1 clk between the read and write clk.

Since the minimum Link width can be 1 symbol (8 bits) long, in that case, for every clk one symbol will be read or written into the EB. Now, in the worst case, SKP may not be inserted in (DPP + overhead) +354 symbols = (1024+32+354) symbols =1410 symbols

Now 1410 symbols will be written or read from the buffer in 1410 clk if link width and hence Elastic Buffer width is 1 symbol long.

Now, number of symbols which can be missed in this period= 1410/178 = 7.877 = 8 symbols (approx.)

Hence the minimum elastic Buffer width for Nominal Empty Buffer mode should be 8 and for Nominal Half-full buffer mode should be 16.

Following the above calculation, one can develop a SKP prediction logic which will be dependent on the following factors:

- Frequency deviation between the read and write clks
- Elastic Buffer depth
- Protocol dependent special symbol insertion and removal logic
- Other factors dependent on elastic buffer implementation

IV. RESULTS

In this section, we try to describe the results and implications of some of the typical scenarios that were tested in while verifying the Elastic Buffer of USB Gen1(5Gbps) PHY

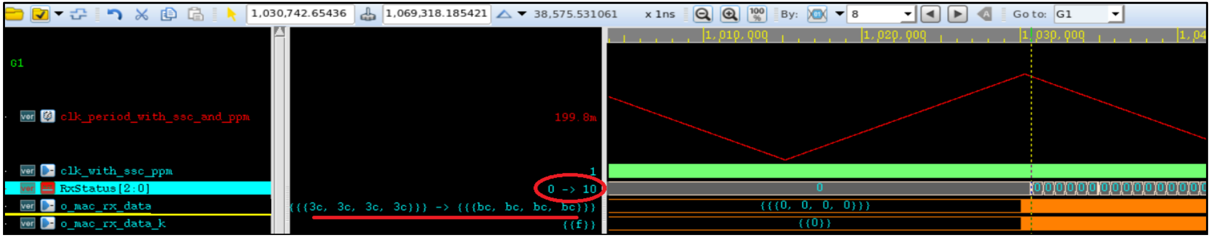
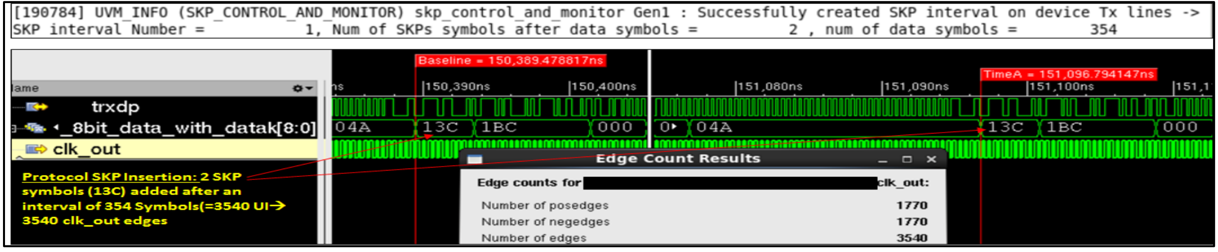


Figure 4.1. PCS- SKP removal to avoid overflow

The Figure 2.1. Shows a clock signal, clk_period_with_ssc_ppm (Rx bit clock), where period is modulated (reduced in this case from the ideal clock period value) using maximum SSC and PPM offset allowable by USB 3.2 specification [1]. The RxStatus signal reported as `b10 shows that SKPs (symbol '3C' shown in signal o_mac_rx_data) have been removed by the PCS in order to avoid an overflow in the elastic buffer.



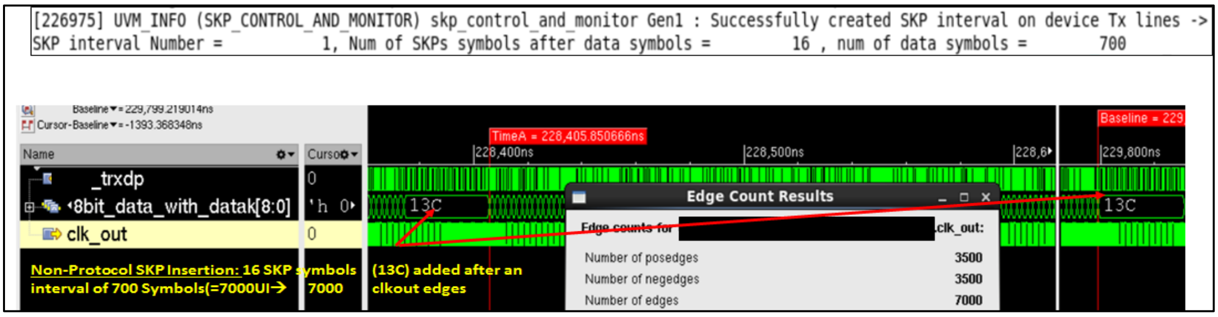
```
[190784] UVM_INFO (SKP_CONTROL_AND_MONITOR) skp_control_and_monitor Gen1 : Successfully created SKP interval on device Tx lines ->
SKP interval Number =          1, Num of SKPs symbols after data symbols =          2 , num of data symbols =          354
```

Baseline = 150,389.478817ns
TimeA = 151,096.794147ns
Name
trxdp
_8bit_data_with_datak[8:0]
clk_out
150,390ns
150,400ns
151,080ns
151,090ns
151,100ns
04A
13C
1BC
000
0
04A
13C
1BC
000
Protocol SKP Insertion: 2 SKP symbols (13C) added after an interval of 354 Symbols(=3540 UI-> 3540 clk_out edges
Edge Count Results

Edge counts for [redacted] clk_out:

Number of posedges 1770
Number of negedges 1770
Number of edges 3540

Fig.4.2. SKP insertion within Protocol limits



```
[226975] UVM_INFO (SKP_CONTROL_AND_MONITOR) skp_control_and_monitor Gen1 : Successfully created SKP interval on device Tx lines ->
SKP interval Number =           1, Num of SKPs symbols after data symbols =          16 , num of data symbols =         700
```

Non-Protocol SKP Insertion: 16 SKP symbols (13C) added after an interval of 700 Symbols(=7000UI→ 7000 clkout edges

Edge Count Results

Edge counts for [redacted] clk_out:

```
Number of posedges     3500
Number of negedges     3500
Number of edges        7000
```

Fig. 4.3. Customised SKP insertion violating protocol limits

Figures 4.2 and 4.3 show log file snippets along with sample waveforms for illustrating SKP symbol insertion on Serial TX lines of Device BFM within protocol limits and outside protocol limits

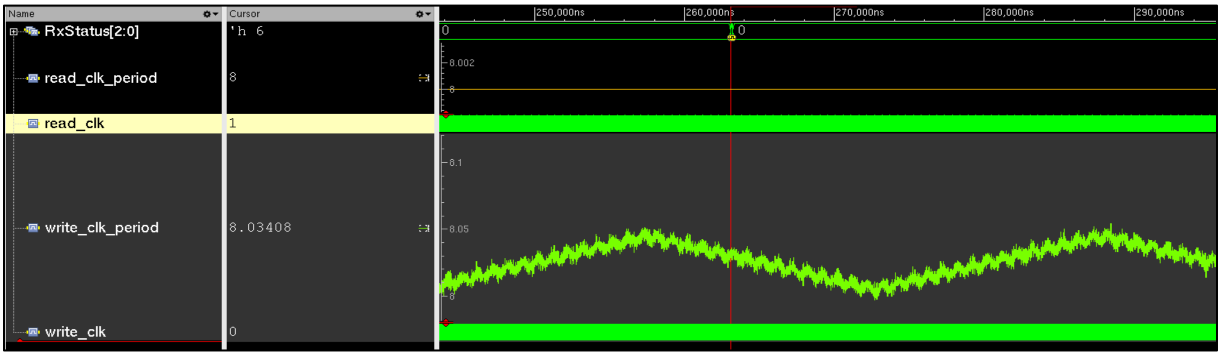


Fig.4.4. Simulation for achieving Underflow (error scenario)

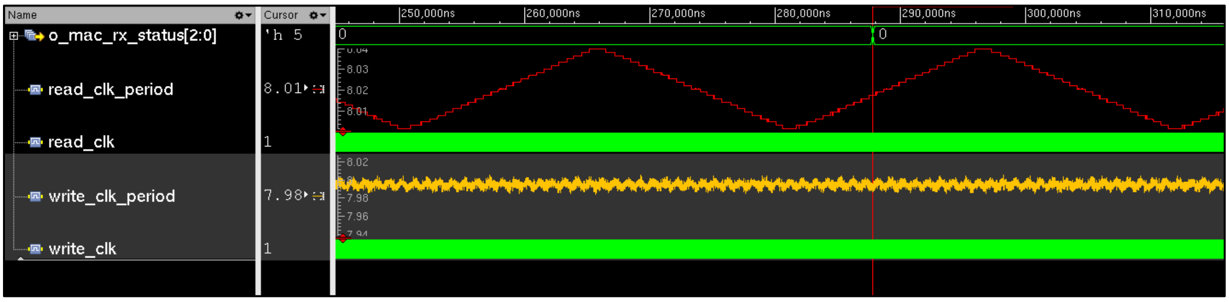


Fig.4.5. Simulation for achieving Overflow (error scenario)

Figures 4.4 and 4.5 show how Underflow and Overflow has been achieved respectively in the simulation by providing, max frequency deviation of 5600ppm between the read and write clks.

For Underflow, the Parameters were configured as follows: DUT Clk (read_clk) is SSC disabled, External Clk modulates Device TX UI with a fixed offset of 600ppm, SSC of 33 KHZ ranging from 0 to 5000ppm which ultimately reflects on write_clk.

While for Overflow, the Parameters were configured as follows: DUT (Read) Clk has SSC of 33 KHZ ranging from 0 to 5000ppm, External Clk modulates Device TX UI with a fixed offset of -600ppm and hence the write_clk also has only -600ppm offset.

In both the cases, SKP ordered sets were being sent after more than 1416 symbols have been transmitted.If like in normal scenario, SKPs are transmitted by data stream after 353 symbols, this Overflow/Underflow could have been avoided in the above cases.

REFERENCES

[1] Universal Serial Bus 3.2 Specification, September 22 2017

[2] PHY Interface For the PCI Express, SATA, and USB 3.1 Architectures Version 4.3, 2014

[3] Somasunder Sreenath, Raghuram Kolipaka, Chirag Shah, Parag Lonkar. " SERDES Rx CDR Verification using Jitter, Spread-spectrum clocking (SSC) stimulus". DVCON INDIA, September 2014

[4] Richard A Prasad, Madhusudan Kulkarni. "DESIGN AND VERIFICATION OF PHY INTERFACE FOR PCIe GEN 3.0 AND USB GEN 3.1 USING UVM METHODOLOGY". IRJET, October 2017

[5] Amlan Chakrabarti, Malathi Chikkanna. "Parameterized and Re-usable Jitter Model for Serial and Parallel Interfaces". DVCON INDIA, September 2014