2014 DESIGN AND VERIFICATION DVCON CONFERENCE AND EXHIBITION INDIA
Sept 25-26, 2014
Hotel Park Plaza
Bangalore, India

Performance Verification of a 6 Gbps HSlink Receiver Including Equalization and Clock Recovery Using Mixed Signal Simulations

Aashish Ram Bhide, Abhishek Chowdhary, Alok Kaushik, Vivek Uppal
STMicreolectronics, Greater Noida, UP, India
(*aashish.rambhide, abhishek.chowdhary, alok.kaushik, vivek.uppal*)@st.com

***Abstract*— A key figure of merit of receivers in serial link standards is the Jitter Tolerance (JTOL) curve and it depends on the performance of both Equalizer and CDR and their interaction. These components are mixed signal designs and hence verifying JTOL is a mixed signal verification challenge and requires a platform to co-simulate HDL, HDL-A, spice netlist and channel s-parameters. In this paper, we present the mixed signal verification environment that was created for a 6Gbps receiver. We explain the strategy that was used to verify JTOL in the presence of channel attenuation. These results from the environment provide the necessary confidence in signing off JTOL of the receiver.**

Keywords— Jitter Tolerance , Equalization, Clock Data Recovery, Mixed Signal Simulation

I. INTRODUCTION

Multi Gbps High Speed Serial Link Physical Layers (PHYs) are a key differentiating IP in the portfolio of any semiconductor IP provider. Serial link standards like USB3, PCIe, SATA, MPHY and DP are now targeting speeds in the range of 5-10Gbps. Reliable communication between a transmitter and receiver is affected by channel degradations and by accuracy of placement of sampling edge in the middle of eye diagram inside the receiver. Inside a receiver, Equalizer is responsible for compensating for channel degradations to open the eye both vertically and horizontally and Clock Data Recovery (CDR) circuit is responsible for placing the sampling edges in the middle of data bits .Figure of merit of a receiver is the Jitter Tolerance (JTOL) curve at the target Bit Error Rate (BER), and it depends on the performance of both Equalizer and CDR and their interaction [1]. Equalizers usually have analog implementation while it is more popular nowadays to have CDR algorithm in digital implementation [2,3]. Thus, verifying JTOL is a mixed signal verification challenge and requires a platform which can co-simulate HDL, HDL-A, spice netlist and channel scattering parameters (S parameters). In this paper, we explain the mixed signal verification environment that was created for a 6Gbps receiver. The paper is organized as follows. Section II will explain the overall verification environment and steps in verifying JTOL on CAD including the effects of channel. Section III will explain the modelling of key components of stimuli generation and receiver. Section IV will show a result and explain value additions of this environment. Section V will conclude the paper

II. VERIFICATION STRATEGY

The 6 Gbps receiver considered in this work is shown as Device Under Test (DUT) in Fig-1. It consists of a Continuous Time Linear Equalizer (CTLE) and CDR based on phase interpolation. CTLE is used to equalize the inter-symbol interference (ISI) introduced due to channel and is an analog circuit which can be tuned to change the location of the zero that is responsible for amplification .The equalized CTLE output is sampled at twice baud rate to capture center and edge samples using two opposite phases of 6 Ghz provided by a Phase Mixer. The phase mixer interpolates between two phases of 6 GHz coming from a Phase Locked Loop (PLL) .The samples are collected through a Serial Input Parallel Output (SIPO) block and processed to realize parallel Bang Bang Phase Detectors (BBPD) on multiple bits. Based on the phase error computations of the BBPD, the Phase Controller decides the next phase adjustment in the phase mixer that produces the sampling edges. Phase controller is responsible for lock acquisition i.e. initial phase lock and tracking the phase movements in the incoming data edges.

A. *Simulation Environment*

JTOL is defined as the tolerance of receiver to jitter in incoming data bits. There are 2 types of jitter that need to be applied i.e. sinusoidal jitter and random jitter. Sinusoidal jitter is characterized by it's frequency and amplitude and random jitter is characterized by it's standard deviation (assumed Gaussian distribution).

Our simulation environment can be divided into 2 parts, namely Stimuli Generation and Receiver Model as shown in Fig-1. Stimuli Generation consists of all the components that are required to produce a calibrated input signal for the receiver with different types of jitter and channel inter symbol interference. Data source produces data to be sent across the link .This block is followed by an 8B10B encoder which is followed by a Parallel In Serial Out (PISO). Output of PISO goes to Jitter addition block which adds programmed amounts of random jitter and sinusoidal jitter. The signals upto this point are digital .The jittered signal goes to De-emphasis and Driver block that converts the digital 1's and 0's to required analog voltages on the line with de-emphasis. The analog signal is then fed into the channel, which is modeled by it's scattering parameters (S-parameters). The s-parameters capture the required characteristics of the channel such as insertion loss, return loss, mode conversion etc in form of 4 port parameters. The output of channel is an analog signal that has been attenuated and has inter-symbol interference.

Receiver model consists of the CTLE followed by samplers and SIPO. The PLL provides the raw clock edges to phase mixer which in turn produces the interpolated edges based on the commands coming from Phase Controller. After locking, the samples on center edges from SIPO are sent to Word alignment block. The word aligned output is fed to 8B10B decoder which produces the final received output. This received output is then compared with data sent from the Data Source. If there are zero received errors over a defined amount of transmission time, the test is considered as pass for that combination of SJ and RJ. Starting from a large amplitude and using a bisection method, a max passing amplitude is found for the value of SJ frequency under consideration. Frequency points for SJ are decided based on standard requirements.

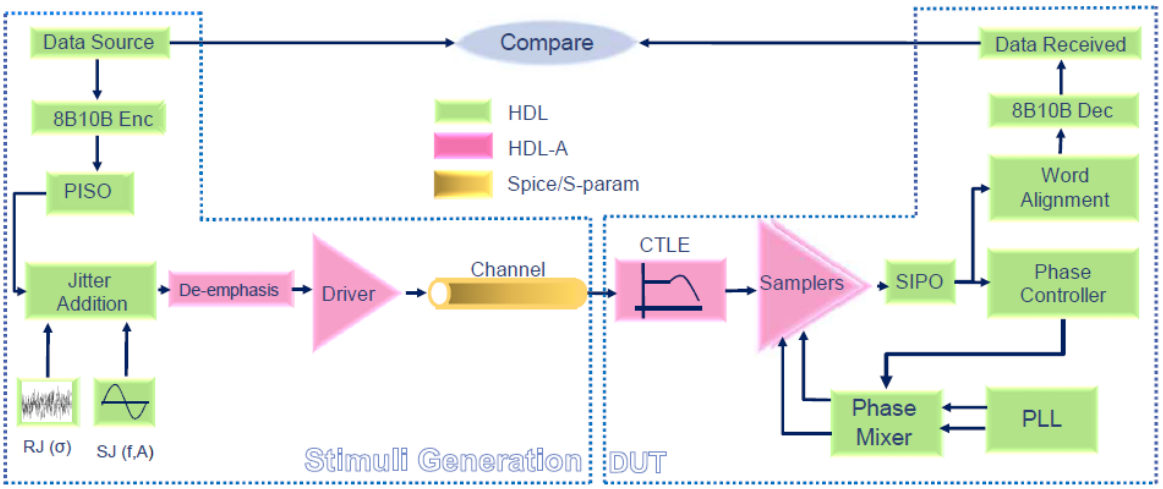


Figure 1. Simulation Environment

B. *Two Step Jitter Tolerance*

Fig-2 shows the two steps in arriving at JTOL of the receiver. Step 1 consists of bypassing HDL-A and spice models (including channel s-parameters) in stimuli generation and receiver. They are replaced by their equivalent simplified HDL models which model their functionality in digital domain e.g. De-emphasis and Driver are replaced by a simple buffer model. Channel and CTLE are also replaced by simple buffer model. Use of these simplified models enables a faster first pass JTOL derivation without the impact of channel and equalizer. JTOL with channel ISI and equalization effects can be derived by reducing this first pass JTOL at each frequency point

by the residual ISI. Residual ISI can be derived by considering the worst case channel (usually defined by the standard) and equalizer reference function, which is also defined in the standard. This can be considered as the final JTOL from digital simulations. By the use of mixed signal environment, we can actually simulate the channel, equalizer and it's interaction with samplers and get a JTOL with equalization considered directly from

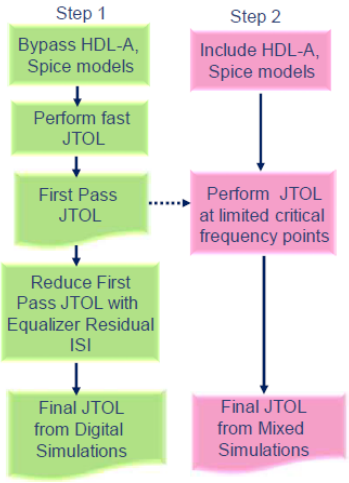


Figure 2. Two Step Jitter Tolerance

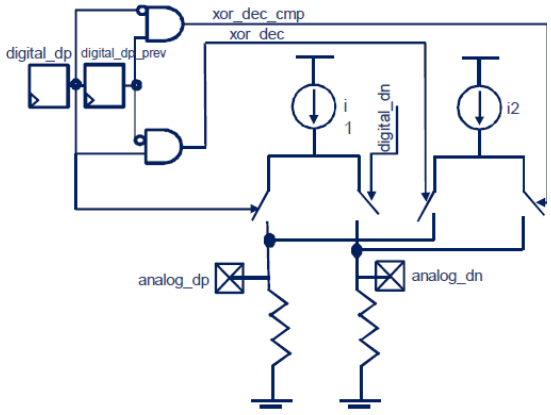
simulation. Since mixed signal simulations are usually more time consuming, JTOL in mixed signal environment is performed at a smaller number of critical frequency points.

III. KEY COMPONENT MODELS

In this section we will explain the model of key elements used in our mixed simulation setup. The code is from VHDL-A.

A. De-Emphasis and Driver

De-Emphasis circuit is shown in Fig-3 and it's corresponding VHDL-A implementation is shown in Fig-3.




```
--Quantity definitions
terminal v_d2a,v_d2a2  : electrical;
quantity v_through_I_SRC_in across i1 through v_d2a to electrical_ref;
quantity v_through_I_SRC_in2 across i2 through v_d2a2 to electrical_ref;

--Check for transition in serial data and type
xor_dec      <= digital_dp and (not(digital_dp_prev));
xor_dec_comp <= (not(digital_dp)) and (digital_dp_prev) ;

--Main Branch Current
I_SWITCH_DP : SWITCH
port map (switch_in => v_d2a,
          switch_out => analog_dp,
          enable => digital_dp);

I_SWITCH_DN : SWITCH
port map (switch_in => v_d2a,
          switch_out => analog_dn,
          enable => digital_dn);

--De-Emphasis Current
I_SWITCH_xor : SWITCH
port map (switch_in => v_d2a2,
          switch_out => analog_dp,
          enable => xor_dec);

I_SWITCH_xor_comp : SWITCH
port map (switch_in => v_d2a2,
          switch_out => analog_dn,
          enable => xor_dec_comp);

--50 ohm terminations to ground
LD_DP_termination : resistor generic map (R => termination_resistance)
port map ( vdd => analog_dp, gnd => electrical_ref);

LD_DN_termination : resistor generic map (R => termination_resistance)
port map ( vdd => analog_dn, gnd => electrical_ref);
```

Figure 3. De-emphasis and Driver

B. *Jitter Addition*

Jitter is added by passing the data edges through a programmable delay element whose delay is derived by the amount of sinusoidal Jitter and random jitter to be applied. This is shown in Fig-4. The sinusoidal jitter delays are generated using the VHDL *sine* function. The random jitter delays are generated by summing the output of 8 *uniform* functions.

```
--Sine Jitter
period1_SinValue := ( SIN ( (real ( (2.0 * MATH_PI * PERIOD1_freq_MHz * sine_time_mod ))/1000000.0)
                 + ((real(POLARITY) + JIT_INIT_PHASE)*MATH_PI) ) );

PERIOD1_delay <= ((integer(jitter_amp) + 1) * UI_unit ) + jitter_amp * period1_SinValue * UI_unit / 2;

--Random Jitter
UNIFORM ( seed0 , seed1 , RANDOM0_d ) ;
UNIFORM ( seed2 , seed3 , RANDOM1_d ) ;
UNIFORM ( seed4 , seed5 , RANDOM2_d ) ;
UNIFORM ( seed6 , seed7 , RANDOM3_d ) ;
UNIFORM ( seed8 , seed9 , RANDOM4_d ) ;
UNIFORM ( seedA , seedB , RANDOM5_d ) ;
UNIFORM ( seedC , seedD , RANDOM6_d ) ;
UNIFORM ( seedE , seedF , RANDOM7_d ) ;
RANDOM_delay <= RANDOM_ampl * UI_unit *
                (RANDOM0_d + RANDOM1_d + RANDOM2_d + RANDOM3_d +
                 RANDOM4_d + RANDOM5_d + RANDOM6_d + RANDOM7_d) * 1.25 / 8.0 ;

--Total Jitter
Jitter_delay <= PERIOD1_delay + PERIOD2_delay + RANDOM_delay + PULSE_delay + (DC_OFFSET*UI_unit);
```

Figure 4. Jitter Addition

C. *Channel*

Channel is modelled as 4 port s-parameters in touchstone format. These s-parameters can be called in the environment using a spice circuit file.

D. *CTLE*

Continuous Time Linear Equalizer is modeled using the *ltf* function of VHDL-A as in Fig-5. This function uses the pole and zero locations of CTLE which are obtained directly from analog designer.

```
entity equalizer is
        port (
                terminal dp_in  : electrical;
                terminal dn_in  : electrical;
                terminal dp_out : electrical;
                terminal dn_out : electrical;
                terminal common_mode_voltage : electrical;
                HSGEAR                  : in std_logic_vector ( 1 downto 0) ;
                HS_RX_EQ_FC             : in std_logic_vector ( 1 downto 0) ;
                HS_RX_EQ_OFFC   : in std_logic_vector (2 downto 0) ;
                IB30U                   : in std_logic := '1';
                HS_RX_EN                : in std_logic := '1' ;
                VDD , GND               : in std_logic ;
                GNDSUB, GNDIND  : in std_logic ;
                STAGGER_ON      : in std_logic ;
                IND_BYPASS              : in std_logic := '0';
                STAGGER_DEL_CTRL: in std_logic_vector(1 downto 0)
        );
end entity equalizer;

architecture analog_model of equalizer is
quantity v_diff_in across dp_in to dn_in;
quantity v_diff_out_dp across i_diff_out_dp through dp_out to electrical_ref;
quantity v_diff_out_dn across i_diff_out_dn through dn_out to electrical_ref;
quantity v_diff_out : voltage;

begin

if (HS_RX_EN = '1') and (IB30U = '1') and (IND_BYPASS = '0') use
        v_diff_out == v_diff_in'ltf((N50, N51, N52, N53, N54, N55), (D50, D51, D52, D53, D54, D55)
        v_diff_out_dp == v_diff_out/2.0+ common_mode_voltage;
        v_diff_out_dn == -v_diff_out/2.0+ common_mode_voltage;
else
        v_diff_out == 0.0;
        v_diff_out_dp == 0.0;
        v_diff_out_dn == 0.0;
end use;
break on HS_RX_EN;
break on IB30U;
break on IND_BYPASS;
end analog_model;
```

Figure 5. CTLE

E. Sampler with Offset

Sampler model with offset is shown in Fig-6. The model has the option to produce random data or always '1' or always '0' when sampling in offset region.

```
process (RN, CK)

begin

        if (RN = '0') then
                rx_data_s <= '0';
                prbs <= "11111";
        elsif (CK'event and CK = '1') then
                if TEST_DATA_IN_EN = '0' then
                        if ( v_diff_in > v_th_1 ) then
                                rx_data_s <= not SWAP_DPDN;--'1';
                        elsif ( v_diff_in < v_th_0 ) then
                                rx_data_s <= SWAP_DPDN;--'0';
                        else
                                if data_type = "random" then
                                        prbs <= prbs(3 downto 0) & (prbs(4) xor prbs(3)) ;
                                        rx_data_s <= prbs(4);
                                elsif data_type = "always_1" then
                                        rx_data_s <= '1';
                                elsif data_type = "always_0" then
                                        rx_data_s <= '0';
                                else
                                        rx_data_s <= 'X';
                                end if;
                        end if;
                else
                        rx_data_s <= TEST_DATA_IN;
                end if;
        end if;
end process;
```

Figure 6. Sampler with offset

IV. RESULTS

Fig-7 shows the first pass JTOL (red curve) obtained from digital models. Since in these simulations, there is no channel or equalizer, the residual ISI is accounted for in JTOL by subtracting it from first pass JTOL at each frequency point (purple curve). The residual ISI estimate is given in the serial link standard spec, where a worst case channel and an equalizer reference transfer function is assumed. Fig-3 also shows the JTOL curve obtained from setup of step 2, simulated at reduced frequency points (black curve). In this simulation, actual worst case channel s-parameter are incorporated along with the CTLE model of the implementation. Hence we get a better estimate of JTOL with residual ISI of channel and our equalizer performance. Moreover, this setup can be used to extract JTOL with channels of different quality.

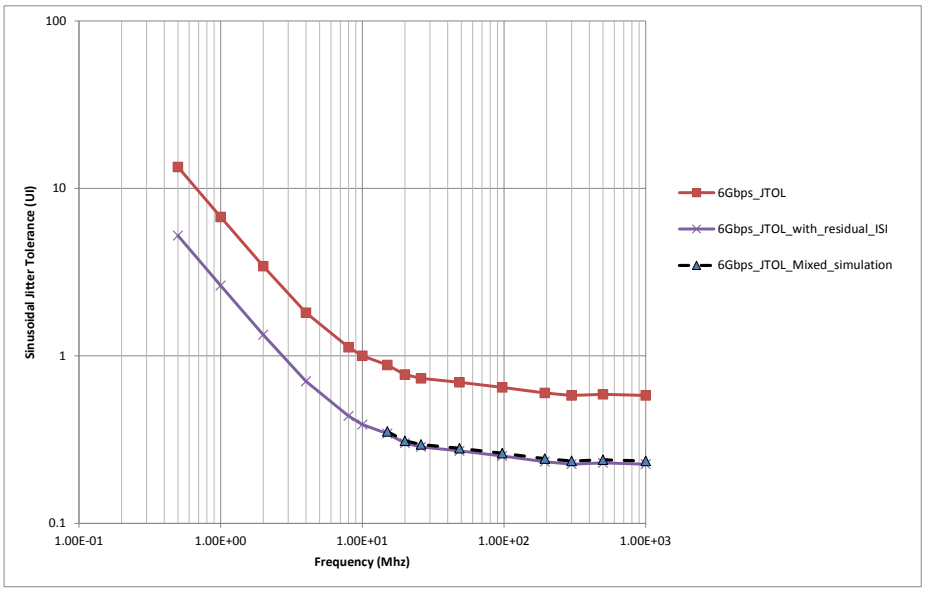


Figure 7. JTOL result

V. CONCLUSION

In this paper, we discussed how a mixed signal verification environment was created for performance (JTOL) signoff of a high speed receiver capable of reception at 6 Gbps. This environment provides confidence in signing off JTOL since it enables simulation of channel ISI, equalizer and CDR together. It is also useful in estimating JTOL with different channels or at different bit rates than specified in serial link standard. It provides a method of checking JTOL with components closest to implementation. Since the components are pin compatible with implementation, it is easy to replace any analog sub-block model by it's actual circuit file at any stage of project. This is particularly useful to do correlation between CAD and silicon performance e.g. to understand the reason of failure, if it is deep rooted in the interaction between equalizer and CDR. Another very significant advantage of this environment is that it provides a way of creating corner cases for verification of CDR algorithm i.e. phase controller, which helps to catch potential bugs in the phase controller algorithm. Going forward, as bit rates increase and there is need to support wide range of channels, we believe this kind of environment will continue to be useful. In particular, it can easily incorporate Decision Feedback Equalizer and effect of crosstalk from adjacent channels through s-parameters.

ACKNOWLEDGMENT

The authors wish to acknowledge the support of Mentor Graphics in using their Questa ADMS flow for mixed signal simulations. The authors also wish to thank the designers of respective blocks inside STMicroelectronics for their support and useful discussions around the development of HDL-A models.

REFERENCES

[1] Vladimir Stojanovic et al, "Accurate Sytem Voltage and Timing Margin Simulation in CDR based high speed designs" ,IEEE Electrical Performance of Electronic Packaging ,2006

[2] Jeff Sonntag et al, "A Digital Clock and Data Recovery Architecture for Multi-Gbps Binary Links", IEEE Journal of Solid State Circuits, vol 41, No8 , 2006

[3] Gerrit W. Den Besten et al , "A Robust High Speed Serial Phy Architecture with Feed Forward Correction Clock and Data Recovery", IEEE Journal of Solid State Circuits , vol 44, No 7 , 2009

[4] Mentor Questa ADMS User Guide