

Equivalence Checking using Assertion based Technique

Shailesh Kumar
NIT Bhopal

Sameer Arvikar
DAVV Indore

Saurabh Jha
STMicroelectronics,
Greater Noida

Tarun K. Gupta, PhD
Asst. Professor NIT
Bhopal

ABSTRACT

This paper presents approach to equivalence checking methodology for large analog/mixed signal systems such as HDMI-PHY, USB-PHY (Transceiver). We verify the equivalence between a behavioral model and its electrical equivalent (Spice netlist) by applying same inputs to both representations. Inputs to spice are given through a D2A. The output waveforms are then compared to find the equivalence. We have implemented SystemVerilog assertions for critical timing checks. We are using digital-on-top (DoT) approach, SystemVerilog assertions are applied in the testbench, means the same assertions are applied to behavioral output as well as to the output of spice which is converted from analog to digital. We have given margin (rise/fall time) in assertions to ensure that assertion should comply with the allowed difference in spice and model output.

The proposed methodology is then applied for equivalence checking of USB-PHY transceiver as a test case.

Keywords

Transceiver, Equivalence, Assertion.

1. INTRODUCTION

The recent advances in semiconductor technology and continued transistor scaling have allowed designers to integrate increasingly more functionality on the same chip. This has resulted in development of complex mixed signal system on chip (SoC) designs. As the designs are becoming more and more complex and also interfacing analog circuitry with digital one there is a need of efficient methodology for verification of mixed signal system to avoid costly respins and design errors.

Current technology is efficient enough to verify the complex digital circuits but the same cannot be said for analog or mixed signal systems. Mixed signal systems SoC contains some digital blocks and some analog blocks.

To verify whole SoC we have to run simulation at SPICE level because there are large number of complex designs interfacing with each other, it takes a lot of time due to huge computational complexity. So instead of using SPICE netlist of analog block we prefer its behavioral equivalent while doing SoC level verification, to reduce the time taken for verification. But at the same time we need to verify the equivalence of analog block with its behavioral model to ensure that the verification at SoC will be done in less time .But this will be at cost of accuracy of verification results as the behavioral model is the highest abstraction level representation of a circuit having its own limits to represent exact analog behavior. Assertion based verification provides effective way to verify and improves overall quality of verification. Sequences and properties of the system are easily represented using SV assertions ,which are obviously same for spice and behavioral model ,so checking consistency of

both representations through assertions proves to be time saving.

2. PRELIMANIRIES

2.1 Problem Realization

Testbench has been written using system Verilog which is applied to both Spice representation and Behavioral model. So the problem comes how to convert digital signals from Verilog (digital inputs) to analog signals which will act as input to spice representation, which should be as close as possible to original inputs. Secondly the checkers in test bench can create problem during transition of analog signals, because analog signals have rise and fall time whereas digital signals change instantly. Firstly we have to decide which inputs (valid) and outputs should be matched to prove the equivalence because using all inputs which are applied on behavioral will take time to simulate with spice.

For equivalence checking using assertions we need to face following challenges.

Which assertions should be applied? So that to cover all possible properties.

How to reuse same Assertions for spice and model?

How the timing checks can be executed using Assertions.

Our method emphasizes on two things, accurate conversion of output analog signals to digital signals because assertions are written in Verilog testbench, and giving allowable margin to assertions because spice output conversion can make the output shifted at different times than model output because of conversion process.

To make digital to analog conversions accurate we have added rise time and fall time to the conversion depending upon the signal specifications given. These converters are called boundary converters, similarly at output of spice and test bench boundary we have used analog to digital converters.

```
a2d_default A2D MODE = std_logic vth=0.5  
d2a_default D2A MODE = std_logic vlc=0.0 vhi=1.0 trise=30p tfall=30p
```

Fig.1 Boundary converters

Converters shown in Figure are default converters, i.e. will be applied to every signal, but can be changed for specific inputs (custom converters).

2.2 Normal equivalence flow:

Since we are using digital on top approach (DoT), a digital Testbench developed in hardware descriptive language (Verilog/systemVerilog) having specific test cases applied on behavior model and spice of TRANSCEIVER.

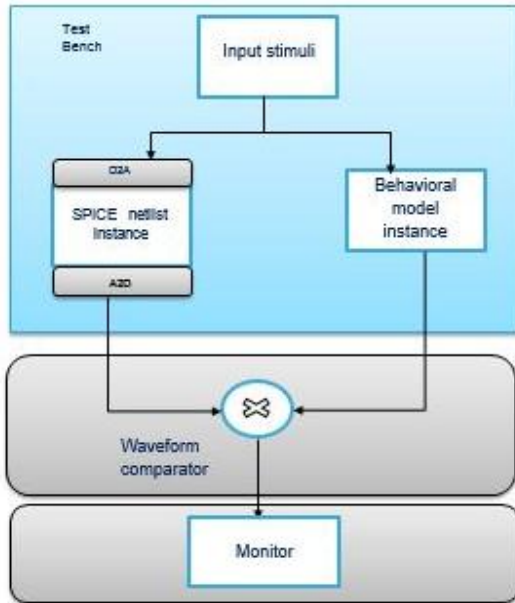


Fig.2 Equivalence flow

In Figure.2 flow same inputs are given to both representations .After the simulation the waveforms are compared in a comparator and the results are checked manually. We can check the result after the completion of simulation that is what not acceptable. We can't wait to complete the simulation as it will take time, we must look an intermediate solution which enable us to analyze our result in between the simulation if output are incorrect.

2.3 Assertion based equivalence flow

Assertion based equivalence flow has an advantage as we can assert same property for the outputs of behavior as well as spice. If any of the assertion failed, the simulation stops and intimate the cause of error.

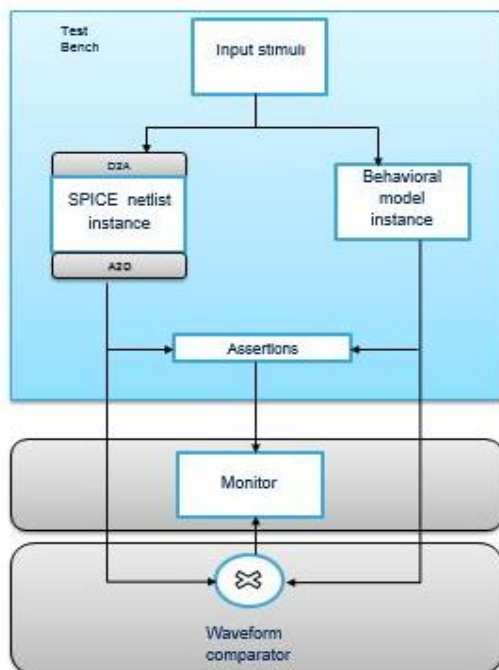


Fig.3 Assertion based equivalence flow

In Figure.3 outputs of both spice and behavioral are given to comparator as well as assertions, if assertion fails we will get the exact time of mismatch quickly rather than comparing them manually.

3. USB PHY AS A TEST CASE

Universal serial bus (USB) is one of the industry-standard extension to the PC architecture with a focus on PC peripherals that enable consumer and business applications. USB PHY, well known as transceiver (XCVR), is one of the Analog block of USB consists of all three modes (LS, FS, HS) of transmit and receive operations.

We have written SystemVerilog assertion for timing (3.1) and functionality (3.2) checks of the Transceiver block. Both the assertions are unique to themselves such that one checks the response time of transceiver disconnect block and other one simply verify functionality as our design working according to specifications.

3.1 USB Transceiver Disconnect Block

The USB specifications [4] defines the physical layer of high speed serial links. In USB PHY data transfer occurs through D+ and D- pads using differential signaling. The PHY therefore consists of a high speed transmitter and a differential high speed receiver. In addition to the differential receiver, there is also a 'Disconnection envelope detector'. A downstream facing transceiver operating in high-speed mode detects disconnection of a high-speed device by sensing the doubling in differential signal amplitude across the D+ and D-lines that can occur when the device terminations are removed. The Disconnection Envelope Detector output (HSDISCONNECTOUT) goes high when the downstream facing transceiver transmits and positive reflections from the open line return with a phase which is additive with the transceiver driver signal. Signals with differential amplitudes ≥ 625 mV must reliably activate the Disconnection Envelope Detector. Signals with differential amplitudes ≤ 525 mV must never activate the Disconnection Envelope Detector.

The Disconnect scenario can be triggered manually in the functional simulations using an internal register based mechanism implemented in the behavioral model. However due to the inability of manipulating the voltage swing dynamically in a DoT based environment, a disconnect test cannot be emulated during normal high speed receiver (HSRX) mode. For the same we have defined custom converter to change the swing on DP/DN pads.

I. Implementation of SVA showing timing check for USB Transceiver Disconnect Block

```

property HSRX_DISCONNECT_MODE;
realtime clk1,clk2;
realtime x_time;
@(posedge clock)($rose(VBGIN && IREXT50U1 && INBIASENABLE_VDDX && INPROTENABLE_VDDX && INCHIRPMODE_VDDX && INHSDISCONENABLE_VDDX
&& !INHSRECNABLE_VDDX && INHSENVENABLE_VDDX),clk1=$realtime)|->first_match(#[0:155]((HSDISCONNECTOUT_VDDX === 1'b1
),clk2=$realtime,x_time = clk2-clk1)) ##1 (x_time)>=144 && (x_time)<=146)
endproperty
assertion_HSRX_DISCONNECT_MODE : assert property (HSRX_DISCONNECT_MODE) $display ("ASSERTION OF DISCONNECT MODE PASS @
%t", $realtime);
else $warning("PROPERTY NOT SATISFIED, ASSERTION FAILED @ %t ", $realtime);
    
```

Fig.4 Assertion defined for HSDISCONNECT MODE

In above Figure.4 assertion describe that HSDISCONNECTOUT will be triggered after 145ns of HSDISCONENABLE enable. It means 145ns is the rise time of HSDISCONNECTOUT. Since we have to prove equivalent of behavior and spice, then we will assert time taken for HSDISCONNECTOUT to rise in both representations with some allowable margin.

II. Simulation result of behavioral model of Transceiver

Figure.5 shows the time required for HSDISCONNECTOUT to become high after all input conditions are met.

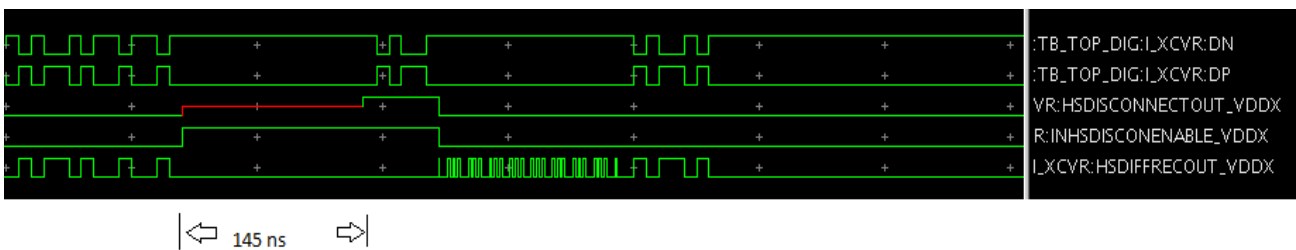


Fig.5 Output of behavioral model of TRANSCEIVER

Figure.6 is an output log, shows that the assertion we have written are working fine for behavioral model.

```

TB_INFO :DISCONNECT_CHECK ON, 4585.00000 ns
ASSERTION OF DISCONNECT MODE PASS @ 4585.50000 ns
TB_INFO :DISCONNECT_CHECK OFF, 4645.00000 ns
    
```

Fig.6 Assertion passed for behavioral model

III. Simulation result of spice of Transceiver

Design is analog while we are applying digital stimuli, we are applying boundary convertors (DAC) to convert digital input stimuli into analog stimuli

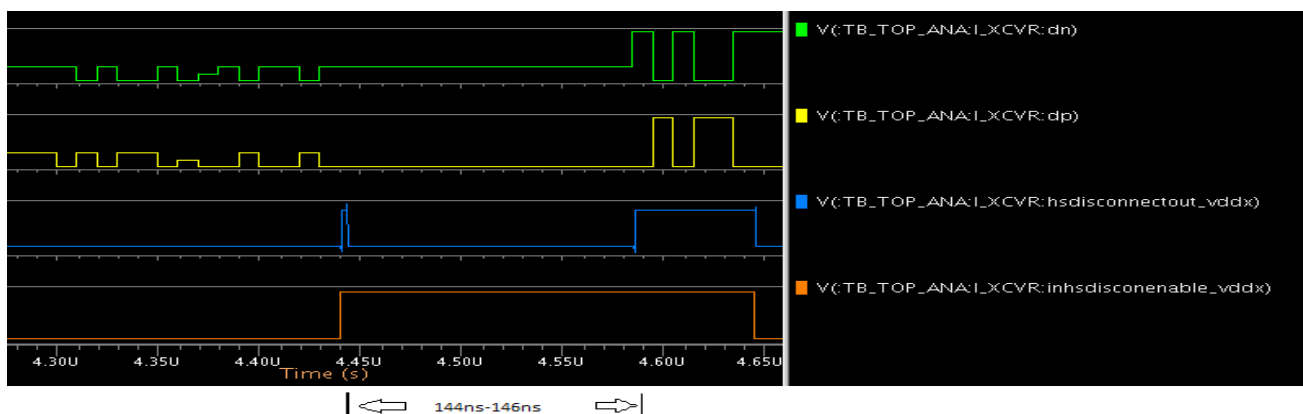


Fig.7 Output of Spice

The output from spice netlist are analog in nature, so we need analog to digital convertor at the boundary of spice and digital testbench.

IV. Analog converted digital wave where the assertions being applied.

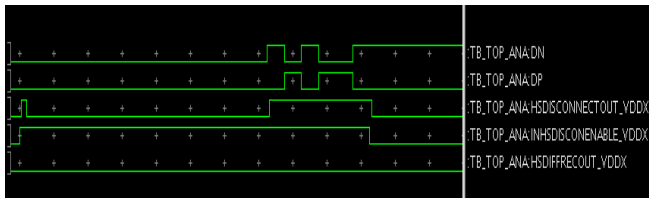


Fig.8 A2D converted output of spice

In Figure.9, the output log showing that Aassertion is getting failed for spice netlist.

```
TB_INFO :DISCONNECT CHECK ON, 4440.00000 ns
** Warning: PROPERTY NOT SATISFIED, ASSERTION FAILED @ 4442.50000 ns
Time: 4442500 ps Started: 4440500 ps Scope: TB_TOP_ANA.assertion_H
ALIDATION/SQUELCH_CHECK_HSRX/DIGITAL_RUN/../../TB/TB_AMS_u2custmac_c
```

Fig.9 output log showing assertion failed for spice

V. Glitch encounter

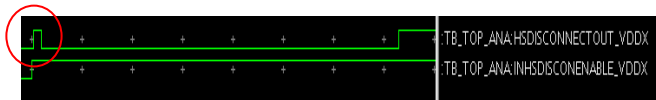


Fig.10 Glitch in HSDISCONNECTOUT

Since according to specification HSDISCONNECTOUT (output) must be triggered after 145ns of INHSDISCONENABLE but HSDISCONNECTOUT goes up before the specified time and hence assertion failed. This is the bug that we have observed without doing any manual

```
property HSRX_NORMAL;
@(posedge clock) disable iff (HSDIFFRECOU_VDDX === 1'bX) (VBGIN && IREXT50U1 && INBIASENABLE_VDDX && INPROTENABLE_VDDX &&
!INHSDRIVERENABLE_VDDX && !INHSTERMENABLE_VDDX && INHSRECEENABLE_VDDX && INHSENVENABLE_VDDX && !INLOOPENABLE_VDDX &&
!INHSDISCONENABLE_VDDX && !INCHIRPMODE_VDDX ) |-> (HSDIFFRECOU_VDDX === DP && HSSQUELCHOUT_VDDX === 1'b0 );
endproperty
assertion_HSRX_NORMAL : assert property (HSRX_NORMAL) $display ("ASSERTION FOR HIGH SPEED RECEPTION PASSES @ %t", $realtime);
else $warning("PROPERTY NOT SATISFIED, ASSERTION FAILED ");
```

Fig.11 SystemVerilog assertion for High Speed Differential Receiver

II. Simulation output of Behavior model

As the waveforms showing the output follow the input. Assertion property get passed without any hassle.

inspection of the output waveform. There is a glitch of 3.16ns encountered during simulation of spice. It may be severe from design prospective. It has saved our time since no need to wait for the completion of the simulation. This is a successful implementation of assertion in equivalence flow.

3.2 USB Transceiver High Speed differential receiver mode

In high speed differential receiver mode, differential signalling of 200mv applied on DP/DN pads. The output of differential receiver is HSDIFFRECOU simply follows the signal value at DP pad. The SystemVerilog assertion for the same has been shown in Figure.9

I. Implementation of SVA showing functionality check for Transceiver differential receiver mode

Virtual clock has been defined because transceiver is clock less circuit, We have triggered the property check at the posedge of clock (virtual). Analog outputs having startup time and rise/fall time delay, according to specification rise time of output signals can vary inbetween 500ps. For the same we have defined clocking block in our testbench So that the sampling of the output HSDIFFRECOU will take place after skew delay of 500ps.

```
clocking cb @(posedge clock);
default input # (0.5) output #0;
output DP, DN;
input HSDIFFRECOU_VDDX;
endclocking
```

Fig.10 Clocking block

The clocking block cb trigger at the posedge of clock and samples DP/DN at the same posedge while sample HSDiffrecoU with the skew delay of 500ps.

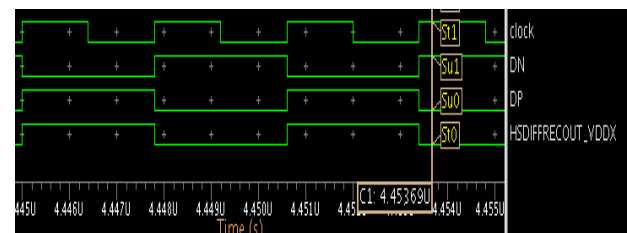


Fig.12 Behavioral output of differential receiver in HSRX

The property and data (DP/DN) are applied at posedge of clock.

III. Simulation output of Spice

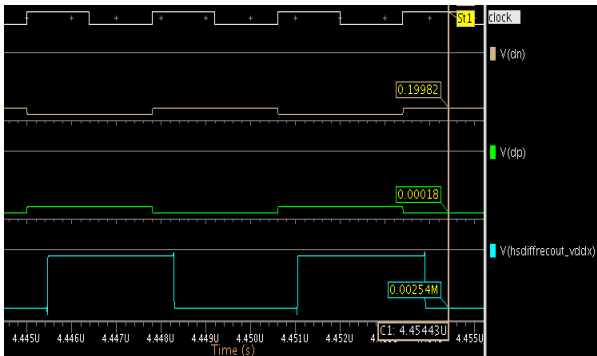


Fig.13 analog output of differential receiver in HSRX

The above analog output **HSDIFFRECOUT** is following input on DP pad but with some delay of rise/fall time delay.

IV. Analog converted digital waves of spice

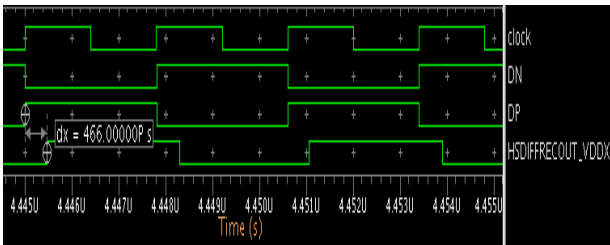


Fig.14 analog converted digital output of differential receiver in HSRX

Assertion are applied to analog converted digital wave, since we have defined clocking block to synchronise our inputs and outputs. The property get satisfied and assertion is passed for spice as well.

V. Assertion result in output log file

```
# ASSERTION FOR HIGH SPEED RECEPTION PASSES @ 4447.80000 ns
# ASSERTION FOR HIGH SPEED RECEPTION PASSES @ 4450.60000 ns
# ASSERTION FOR HIGH SPEED RECEPTION PASSES @ 4453.40000 ns
# ASSERTION FOR HIGH SPEED RECEPTION PASSES @ 4456.20000 ns
# ASSERTION FOR HIGH SPEED RECEPTION PASSES @ 4459.00000 ns
```

Fig.15 output log showing assertions are passed for digital and spice

4. SIMULATION RESULTS

We have followed a Digital-on-Top (DoT) approach for equivalence validation as suggested in [3]. However, the proposed technique is valid for any mixed signal co-simulation environment. The simulations are performed using

QuestaADMS [6] platform on USB PHY Transceiver. The simulation results of Transceiver high speed HSDISCONNECT mode shown that there was mismatch between spice and behavioral model .As soon as assertion updates log file we found the glitch and started reviewing the spice and model before the completion of spice simulation hence saving the time .We have automated the process of equivalence checking by writing the assertions for timing checks.

The second assertion (figure.9) shown, is simply check the functionality as output follow to the input or not. We have successfully ran the assertion and it get passed in high speed differential receiver case.

5. CONCLUSION

We have successfully implemented the SystemVerilog assertions for equivalence methodology. It gives us surety that our behavioral and spice are performing according to the specifications. Assertions make the equivalence environment more automated and less time consuming such that we need not to analyze the waveform manually or if any of the design misbehaves it stops the simulation. The main advantages of developing the equivalence checking flow is to ensure our behavior model equivalent to that of spice netlist, so that at SoC level we can save simulation time of system while replacing spice with its behavior model. This will faster our verification process at the SoC level.

6. REFERENCES

- [1] M.H. Zaki et al, "Formal verification of analog and mixed signal designs: a survey", *Microelectronics Journal*, 39(2008), pp. 1395-1404.
- [2] Amandeep Singh and Peng Li, "On behavioral model equivalence checking for large analog/mixed signal systems", *Proceedings of the IEEE International Conference on Computer-Aided Design*, San Jose, California (2010).
- [3] H. Parekh et al, "Equivalence Validation of Analog Behavioral Models", *DVCon United States* (2014).
- [4] Universal Serial Bus 2.0 Specification, USB Implementers Forum, (2000).
- [5] R. Mekkadan, "Functional Verification of CSI-2 Rx-PHY using AMS Co-simulations", *DVCon India* (2014).
- [6] Questa ADMS Reference Manual, Mentor Graphics.
- [7] EZWave Reference Manual, Mentor Graphics
- [8] C. Ming song and P. Mishra, 'Assertion-Based Functional Consistency Checking between TLM and RTL Models', in *26th International Conference on VLSI Design and the 12th International Conference on Embedded Systems*, 2013, pp. 320-325