DDR5 Equalization Options with IBIS

Arpad Muranyi
Mentor Graphics, A Siemens Business

IBIS Summit, DesignCon
February 2, 2018
Santa Clara, CA
Background

- DDR Equalization is a hot topic as data rates keep going up
- Engineers are looking for simulation solutions
  - What kind of equalizations are needed?
    - Tx: FIR; Rx: CTLE, FFE, DFE, etc.
  - What types of effects need to be simulated?
    - Simultaneously Switching Output (SSO), Power Delivery (PDN), Single Ended, Differential signaling effects, Crosstalk, etc...
  - What kind of models can include all these effects?
  - Is the statistical modeling/simulation approach needed?
- A good example for such conversations is an [SI-List] email thread with the subject line “EQ for DDR5” that started in October 2017
  — https://www.freelists.org/post/si-list/EQ-for-DDR5#footer
Is IBIS-AMI a viable option?

- **IBIS-AMI** can simulate practically any filter
  - Simulations are extremely fast, millions of bits in seconds or minutes

- Its fundamental assumption is that the channel (including the buffers on the ends) is Linear and Time Invariant (LTI)
  - This excludes SSO, PDN and other time varying effects

- **IBIS-AMI** also assumes that rising and falling edges are symmetric

- It was conceived with SerDes topologies in mind
  - SerDes is point-to-point, DDR has a controller and several memory chips
  - SerDes uses embedded clocks (and CDR), DDR uses a separate clock input

- **IBIS-AMI** is tailored for differential signaling
  - DDR uses single ended and differential signals
If not IBIS-AMI, what else could be considered?

- Good old SPICE transistor level models
  - Don’t even think about it, they are so slow… 😊

- StatEye, Matlab, Python, etc…
  - Your mileage may vary, lack of standardization, etc…

- At least one “very large IC vendor” started releasing an increasing number of Verilog-A behavioral buffer models in recent times

- Most major EDA vendors provide support for Verilog-A models
  - Transparent to the user, no need to manually compile the models

- Verilog-A is available as a modeling language extension in IBIS since v4.1 (2004)
  - [External Model] or [External Circuit] keywords
Let’s look at a Verilog-A example

- **Tx:** conventional IBIS model from Micron
- **Rx:** clocked DFE model using IBIS [External Circuit] with Verilog-A

Rx with DFE

Clock input for the DFE in the Rx model
DFE code snippet (declarations, initializations)

```verilog
module VA_DFE_Sngl_Clocked(In, Out, ClkInP, ClkInN, PCref, GCref);
  electrical  In, Out, ClkInP, ClkInN, PCref, GCref;
  branch (In,GCref) BrCcomp;
  branch (PCref,In) BrRterm;
  parameter real Rterm = 48;
  parameter real Ccomp = 0.88e-12;
  parameter real BitInterval = 0.25e-9;
  parameter real Tap0 =  0.12099;
  parameter real Tap1 =  0.0181078;
  parameter real Tap2 =  0.00692665;
  parameter real Tap3 =  -0.0133208;
  localparam integer NoOfTaps = 4;
  real Taps[0:NoOfTaps-1] = {Tap0, Tap1, Tap2, Tap3};
  real CurTime = 0;
  integer Tenable = 0;

  integer i = 0;
  integer Bits[0:NoOfTaps-1] = {0, 0, 0, 0};
  real Sum = 0;
endmodule
```

$$z_k = y_k - w_1 \tilde{d}_{k-1} \cdots - w_{n-1} \tilde{d}_{k-(n-1)} - w_n \tilde{d}_{k-n}$$
DFE code snippet (sampling logic)

```vhdl
// analog begin
// ----------------------------------------
@cross(V(ClkInP, ClkInN), 0) begin
  CurTime = $time;
  if (Tenable == 0)
    Tenable = 1;
  for (i = NoOfTaps - 1; i > 0; i = i - 1)
    Bits[i] = Bits[i - 1];
  // Store most recent sample in Bits[0]
  // If input is above the threshold, store +1
  // If input is below the threshold, store -1
  // No change otherwise
  if (V(Out, GCref) > Threshold)
    Bits[0] = 1;
  else if (V(Out, GCref) < Threshold)
    Bits[0] = -1;
end
// ----------------------------------------
@timer(CurTime + BitInterval/2, , Tenable) begin
  // Calculate the value that will be subtracted from the input waveform
  Sum = 0;
  for (i = 0; i < NoOfTaps; i = i + 1)
    Sum = Sum + Taps[i] * Bits[i];
end
// ----------------------------------------
```

This code executes once when the differential clock signals cross each other.

This is a timer event:
This code executes once, ½ UI after each clock.

This is the clock event:
This code executes once when the differential clock signals cross each other.
DFE code snippet (analog output equations)

```
// --analog begin

// Termination resistor equation
V(BrRterm) <-> Rterm * I(BrRterm);

// Ccomp equation
I(BrCcomp) <-> Ccomp * ddt(V(BrCcomp));

// Output waveform equation
V(Out,GCref) <-> V(In,GCref) - Sum;

// --end

endmodule
```

Analog equations execute at each time step
More equations can be added for
- additional termination resistors
- “split C_comp”
- etc…
The “AMI-like” reference simulator

- The Verilog-A simulation results are compared against the results obtained from an IBIS-AMI-like simulator.
- This simulator works very much like IBIS-AMI, except it uses its own, built-in algorithmic models instead of IBIS-AMI models.
- It has some features which are not supported by IBIS-AMI — e.g. it supports asymmetric rising and falling edges.
- We will refer to this simulator as “RefSim” on the remaining slides.
Results: IBIS + Verilog-A

Verilog-A model
Bit rate: 4 Gbit/s
Bit pattern: $4 \times PRBS7$
Total bits: 512
Time step: 7.8125 ps
Sim. time: 12.5 sec
Results: RefSim

RefSim model
Bit rate: 4 Gbit/s
Bit pattern: 4 * PRBS7
Total bits: 512
Samples/bit: 32
Ch. char. time: 10.4 sec
Alg. sim. time: 2.4 sec

Red = before DFE
Green = after DFE
Overlaying IBIS and RefSim at DFE input

4 * PRBS7 = 512 bits

IBIS_pad

RefSim_pad
Overlaying IBIS and RefSim at DFE output

4 * PRBS7 = 512 bits
Results: IBIS + Verilog-A

Verilog-A model
Bit rate: 4 Gbit/s
Bit pattern: 1 * PRBS12
Total bits: 4096
Time step: 7.8125 ps
Sim. time: 1 min 10 sec

1 * PRBS12 = 4096 bits
Results: RefSim

RefSim model
Bit rate: 4 Gbit/s
Bit pattern: 1 * PRBS12
Total bits: 4096
Samples/bit: 32
Ch. char. time: 12.1 sec
Alg. sim. time: 2.0 sec

Red = before DFE
Green = after DFE
Overlaying IBIS and RefSim at DFE input

1 * PRBS12 = 4096 bits
Overlaying IBIS and RefSim at DFE output

1 * PRBS12 = 4096 bits
Adding jitter to the stimulus of U1.D (DQ0)

Advanced Jitter Settings:

- Add Gaussian jitter
  - Standard Deviation: 8 % of UI
  - Frequency: 1 % of rate
  - Peak-to-peak: 42.21 % of UI at BER of 10^-12

- Add uniform jitter
  - Peak-to-peak: 0 % of UI
  - Mean: 0 % of interval

- Add sinusoidal jitter
  - Half of peak-to-peak: 0 % of UI
  - Initial phase: 0 degrees
  - Frequency: 1 % of rate

- Add Ta duty cycle distortion
  - Peak to peak: 0 % of UI

Jitter template:

[Load] [Save] [Save As]
Results: IBIS + Verilog-A with jitter

1 * PRBS12 = 4096 bits

Before DFE

After DFE
Results: RefSim with jitter

Red = before DFE
Green = after DFE
Overlaying IBIS and RefSim at DFE input

1 * PRBS12 = 4096 bits
Overlaying IBIS and RefSim at DFE output

1 * PRBS12 = 4096 bits
Half way point summary

- RefSim was used to synthesize the DFE tap settings
- These tap settings were used for all IBIS-Verilog-A and RefSim simulations
- Up to this point, the IBIS-Verilog-A and RefSim results match well
  - These simulations did not include any non-LTI effects
- The simulation time for 512 bits were about the same
  - RefSim spent most of its time on channel characterization
  - The algorithmic simulation time remains very short (almost constant) for few thousand bit simulations
  - The IBIS-Verilog-A simulation time is comparable to normal IBIS simulations, and grows proportionally with the number of bits simulated
Adding a non-ideal PDN model

Rx with DFE

Clock input for the DFE in the Rx model

A very crude PDN model
IBIS + Verilog-A with non-ideal PDN

Power, ground and signal (before DFE) waveforms with and without PDN
IBIS + Verilog-A with non-ideal PDN

Note the asymmetry in the rising and falling edges and the high and low signal levels.

$1 \times \text{PRBS12} = 4096$ bits
RefSim with non-ideal PDN

The channel characterization was performed with step and pulse waveforms

Red = before DFE
Green = after DFE
RefSim with non-ideal PDN

The channel characterization was performed with a PRBS5 pattern

Red = before DFE
Green = after DFE
Overlaying IBIS and RefSim at DFE input (with Step & Pulse channel characterization)

$1 \times \text{PRBS12} = 4096 \text{ bits}$
Overlaying IBIS and RefSim at DFE input (with PRBS5 channel characterization)

1 * PRBS12 = 4096 bits

IBIS_pad
RefSim_pad
Final summary

- The simulation results are significantly different when non-LTI effects are present.

- This happens because the channel characterization is a “snapshot” of the channel’s behavior at a certain time.
  - The assumption is that the channel’s characteristics do not vary with time.
  - In reality, the channel’s characteristics may vary with time.
  - The PDN effects are just one such example.

- For this reason, the IBIS-Verilog-A simulations are more accurate.

- Note: IBIS-Verilog-A models are not limited to DFE modeling only.
  - What you can model is mostly limited by your imagination and programming experience only.
Is the slower simulation time a problem?

- DDR designs may not need multi-million bit simulations
  - DDR and SerDes protocols are different
  - SerDes channels are unidirectional and can have very long bit streams
  - DDR channels are bi-directional, and the length of a bit stream is limited by alternating read/write cycles

- DDR channels tend to have more non-LTI related challenges which need to be studied by simulations

- IBIS-Verilog-A models may be reasonably fast for DDR simulations
  - They are definitely many orders of magnitudes faster than full transistor level SPICE models 😊
How about Intellectual Property protection?

■ It is commonly believed that Verilog-A models cannot be encrypted
  — I was under that impression too until a few weeks ago...
  — It turns out that most major EDA vendors can generate and use encrypted Verilog-A models
  — Encrypted versions of the models used in this presentation have been tested successfully in two EDA tools

■ To make the model maker’s life easier, it would be useful to have an industry wide common encryption mechanism
  — The Accellera P1735 effort addressed that, but a team from the University of Florida found security flaws in it
  — Need to investigate whether these problems are addressed
  — Need to investigate what, if anything, the IBIS Open Forum needs to do in order to use it in IBIS contexts