Welcome to

DesignCon® 2022

WHERE THE CHIP MEETS THE BOARD

Conference
April 5 – 7, 2022

Expo
April 6 – 7, 2022

Santa Clara Convention Center

Validation Shift-Left: Enabling Early SerDes Mixed-Signal Validation

David Halupka (SeriaLink Systems)

David Halupka, Aleksey Tyshchenko (SeriaLink Systems)
Richard Allred, Marc Erickson, Tripp Worrell, Barry Katz, Jesson John (The MathWorks)
Venu Balasubramonian, Lenin Patra (Marvell Semiconductor), Ranjan Sahoo (NXP)

SPEAKER

David Halupka, PhD

Co-Founder, SeriaLink Systems

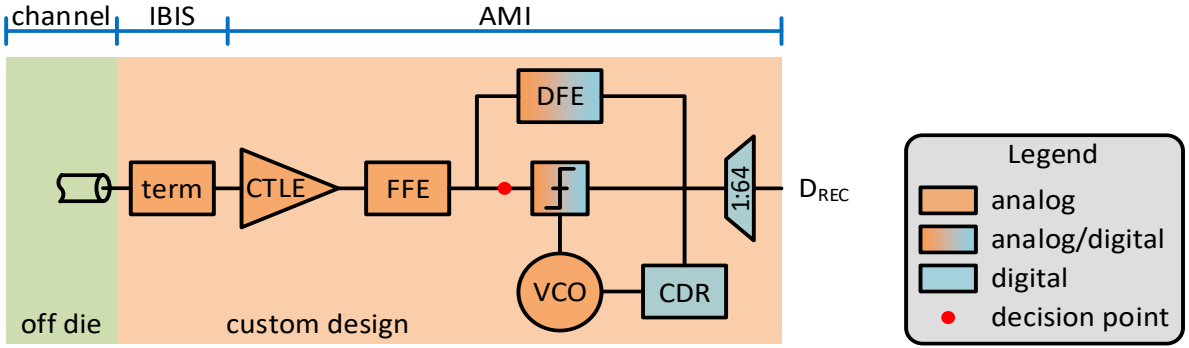
david@serialinksystems.com | www.serialinksystems.com

David has over 20 years of experience in mixed-signal and embedded system design. He was with Kapik for 11 years, where he served as Senior System-Architect and Principal Engineer and led the digital design team. In 2018, he joined Intel's Mixed Signal-IP Group as Senior Systems Engineer, where he was responsible for adaptation algorithm development for the multi-standard SerDes. Ph.D, M.A.Sc., and B.A.Sc. from the University of Toronto.

SERIALINK SYSTEMS

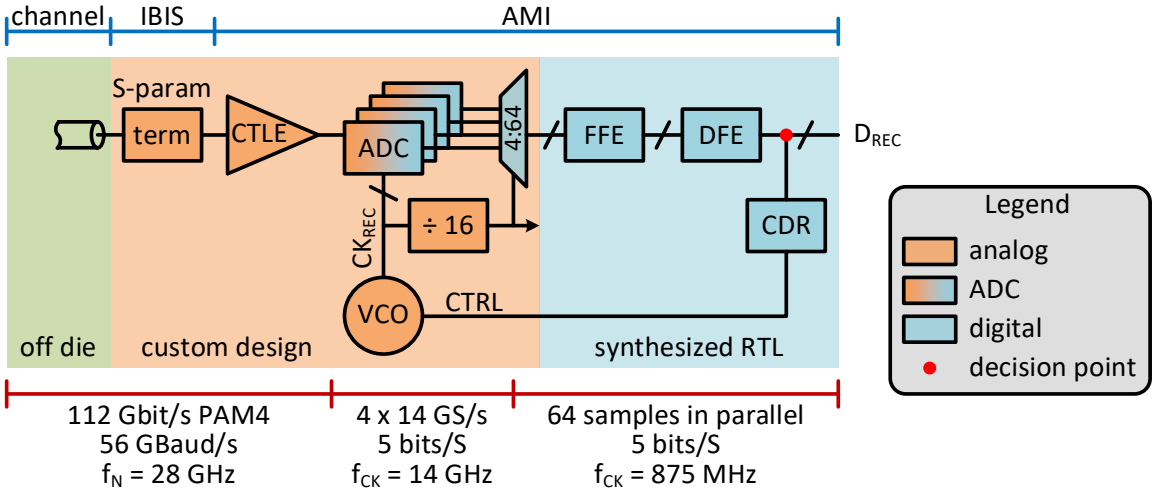
SeriaLink Systems is a consulting team focusing on system modeling of high-speed serial links, IBIS AMI modeling, model correlation and system validation. SeriaLink Systems is working on building a configurable modeling flow to support SerDes projects through their entire life cycles: from architecture definition, through analog and digital design, to design validation

Motivation: Legacy SerDes



- Legacy-based SerDes used analog circuits for RX equalization
- Digital subsystem used for post-equalization deserialization, adaptation control, analog calibration
- Digital/analog interaction mainly one-way: consumer/producer type of interaction

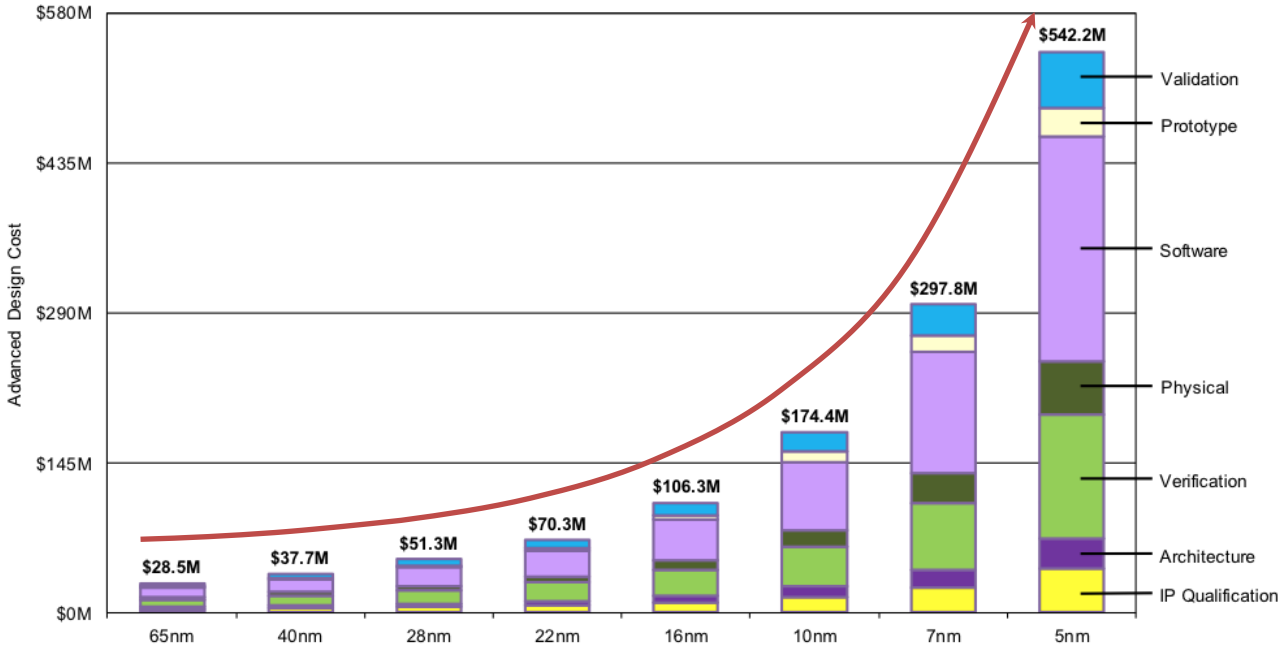
Motivation: SerDes Complexity is Increasing



- Modern SerDes are increasingly mixed-signal: equalization is split between analog and digital domains
- Closed-loop analog/digital interaction is increasing, e.g., CDR loop crosses digital/analog boundary 2×
- Digital systems (not shown) control calibration tuning of analog sub-systems as well
- Digital/Analog validation becomes increasingly more important

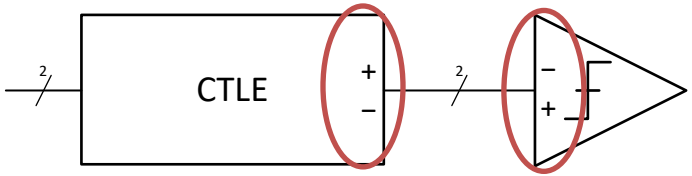
Increasing Design Costs

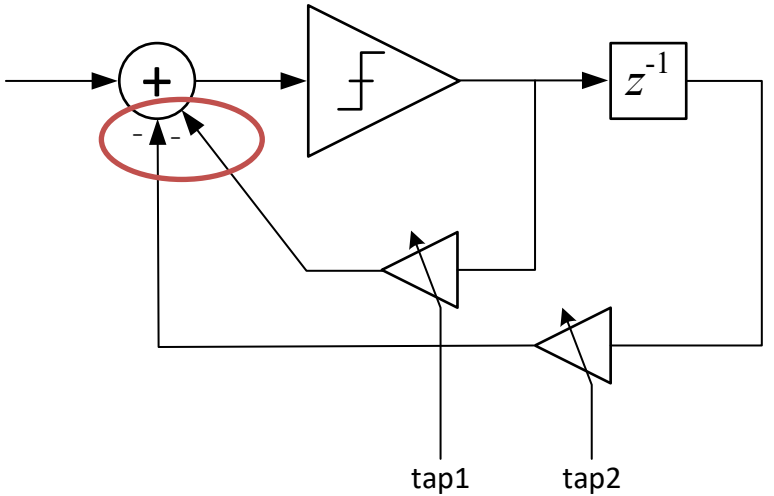
- **Exponentially increasing design costs**
- **Nano-scale technologies are not analog friendly**
  - Reducing voltage headroom
  - Increasing process variation
  - Increasing layout-dependency
- **1st-time success is critical, hence verification is critical**



IBS, "As Chip Design Costs Skyrocket, 3 nm Process Node Is in Jeopardy," 2020. https: //www.extremetech.com/computing/272096-3nm-process-node (accessed on 11 March 2022).

Motivation: Tiny Mistakes → Huge Problems

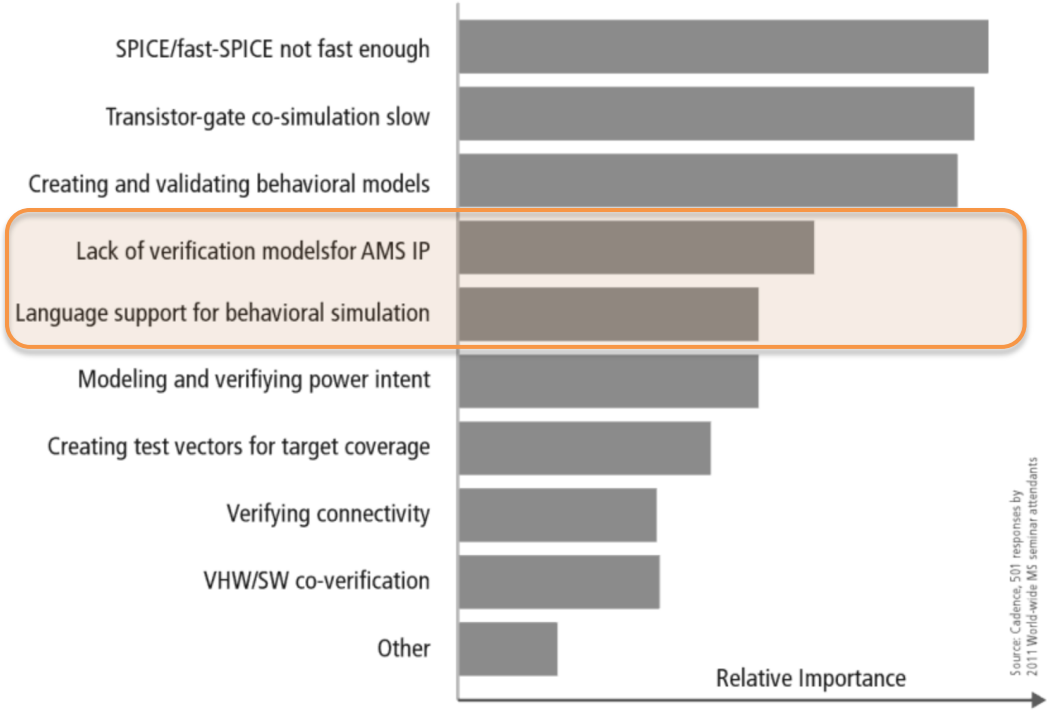




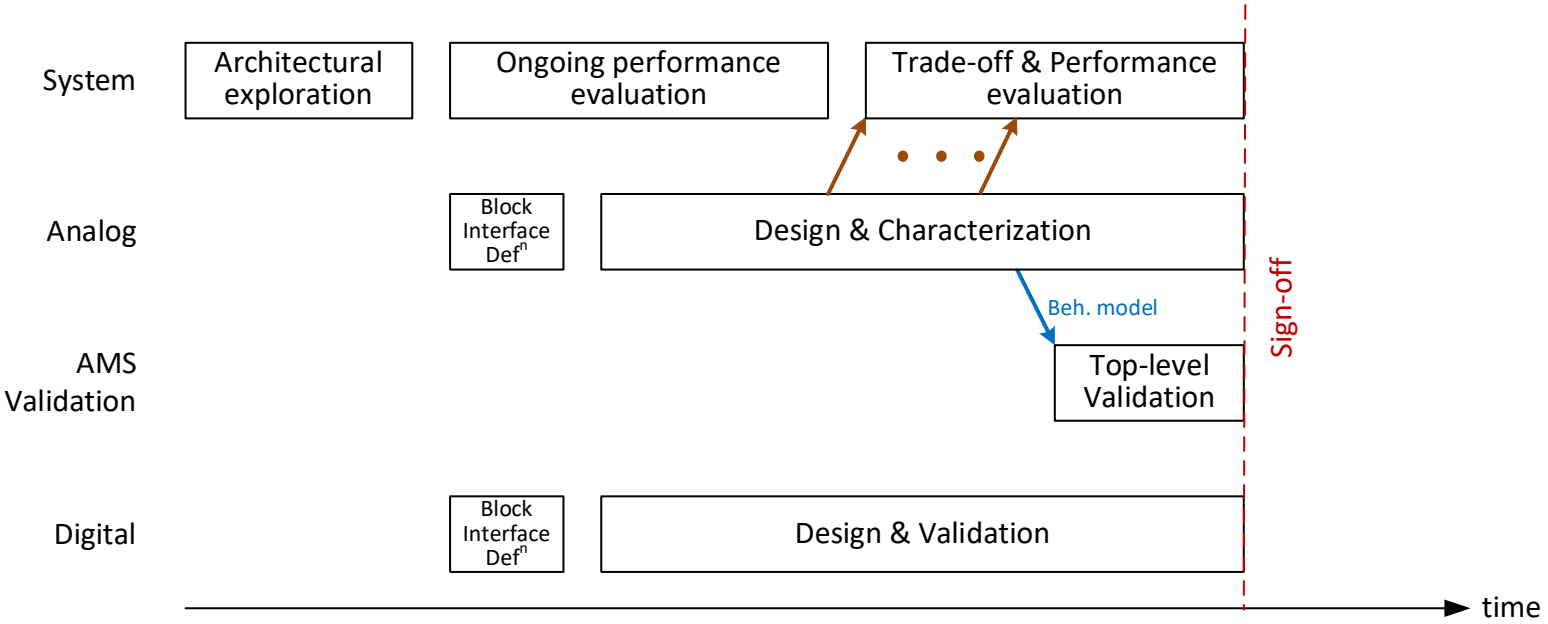
- "The devil is in the details" – Failures often occur at digital/analog boundaries
- "If it's not tested, it doesn't work" – Mixed-signal simulations test domain crossing interfaces
- Failures are costly
  - Low-end: metal mask change to fix interface issues (optimistic)
  - High-end: missed market window, squandered investment, unhappy customers, and tarnished reputation

Motivation: Solution is to Test Everything

- **Unit testing**
  - Block functionality verified against specifications
- **In-situ testing**
  - Blocks tested in groups within the same domain
- **Interface testing**
  - Exercises inter-domain connectivity/functionality
- **Validation time is limited**
  - Analog circuit simulations are CPU intensive
  - Behavioral models accelerate M/S validation

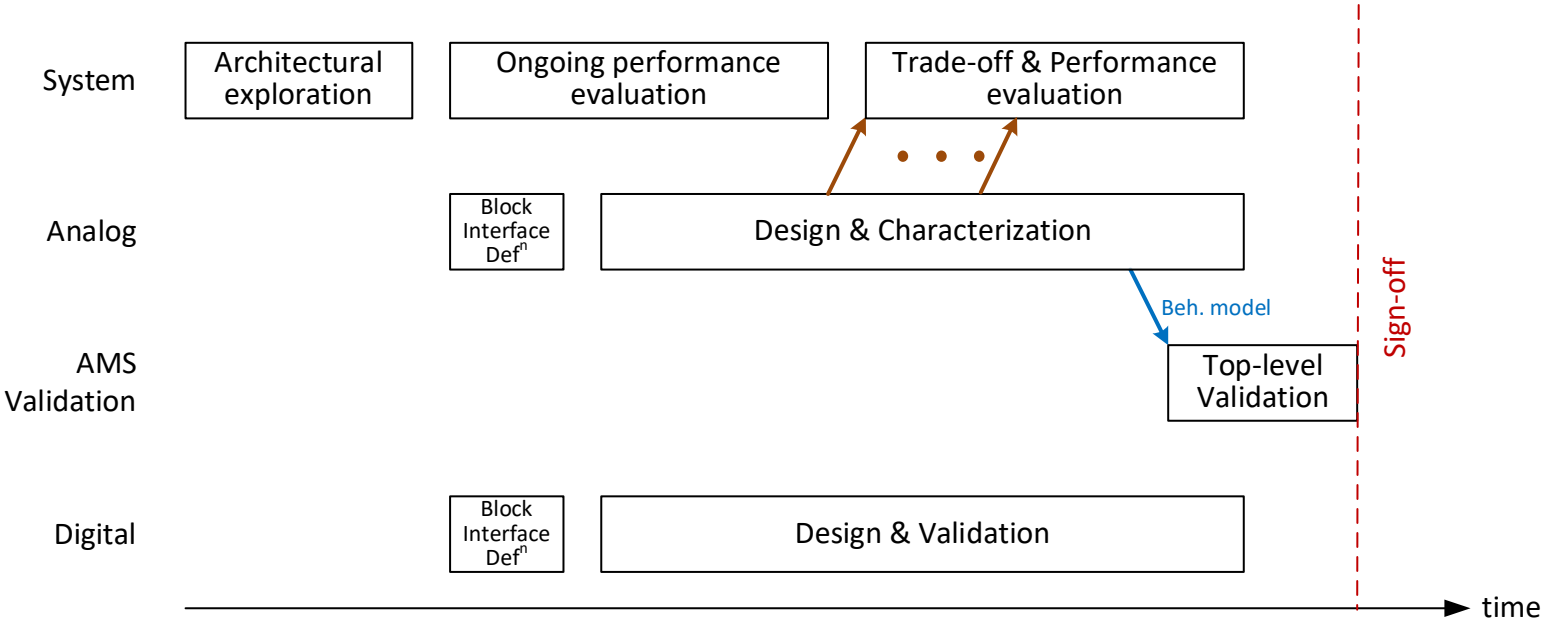


Motivation: Typical SerDes Project Lifetime



- **Behavioral models are created late in the project lifecycle or are insufficiently representative**
- **Behavioral models require manual design in a low-level language: SystemVerilog, C/C++, etc.**

Motivation: Typical SerDes Project Lifetime



- **Behavioral models are created late in the project lifecycle or are insufficiently representative**
- **Behavioral models require manual design in a low-level language: SystemVerilog, C/C++, etc.**

Outline

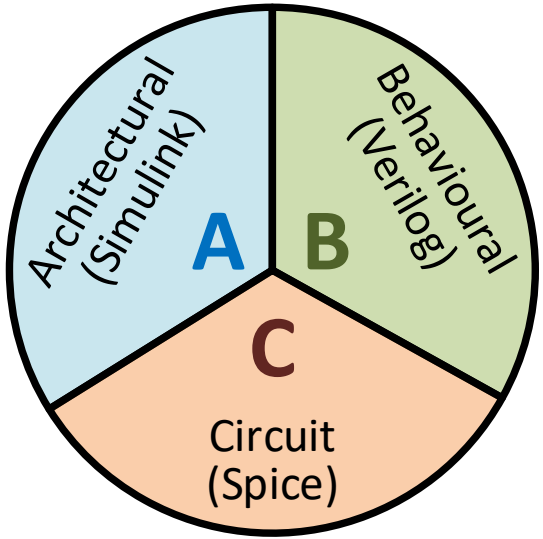
- Model use throughout a SerDes development life-cycle and the ABCs of SerDes models
- Generating B-models from A-models
- C-model design and characterization
- Updating A-model based on C-model characterization
- Automatic B-model refresh from A-model

Outline

- **Model use throughout a SerDes development life-cycle and the ABCs of SerDes models**
- Generating B-models from A-models
- C-model design and characterization
- Updating A-model based on C-model characterization
- Automatic B-model refresh from A-model

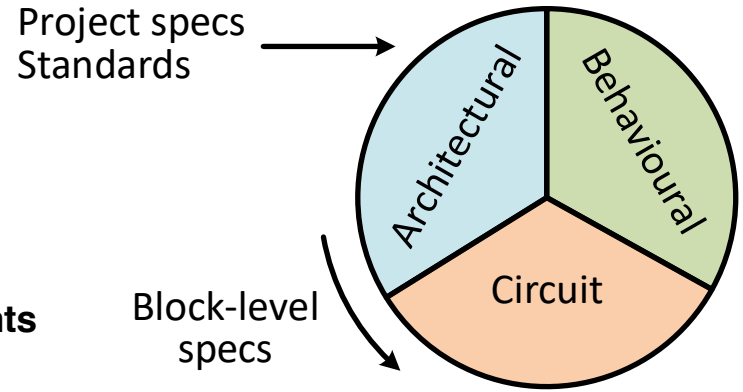
ABC Models

- **Architectural** models validate system functionality with different channels
- **Behavioral** models are used to stand-in for C-models to accelerate validation
- **Circuit** models make use of circuit simulators and embody the low-level details of the SerDes design
- **ABC** models are (re)used throughout a SerDes development



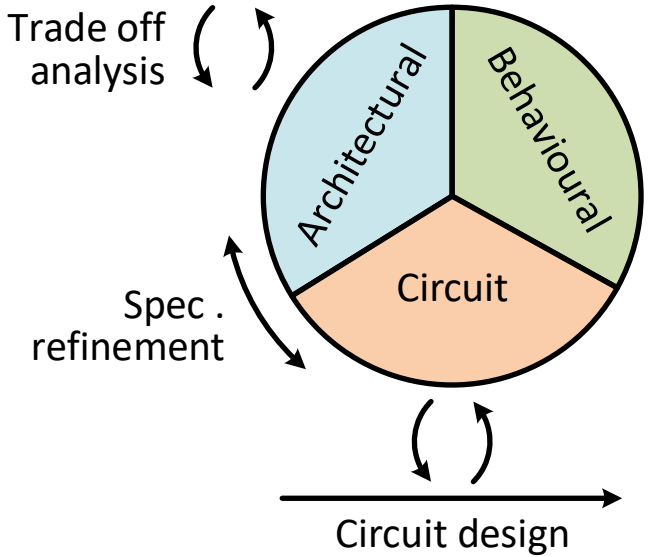
Models Usage Scenarios

- **Requirements and standards drive architectural definition**
- **Architectural choices are explored using A models**
- **A models embody design requirements and target functionality/behavior**
- **Specifications are provided to circuit designers for implementation**



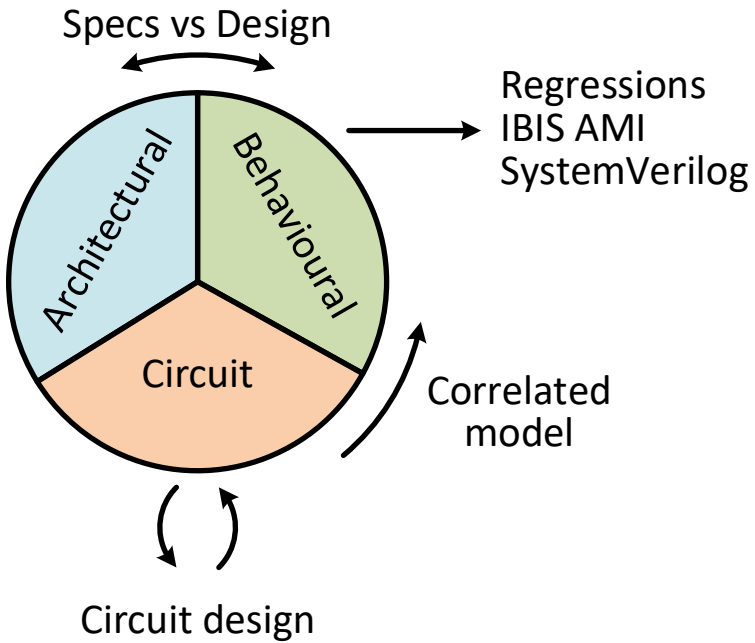
Models Usage Scenarios (2)

- **Meeting specs may not be feasible**
  - Voltage headroom limitations
  - Technology limitations
  - Power/area limitations
- **Design limitations are encountered during circuit design, but design tradeoffs are explored using A models**
- **For tradeoff analysis, A model is updated to match circuit performance**

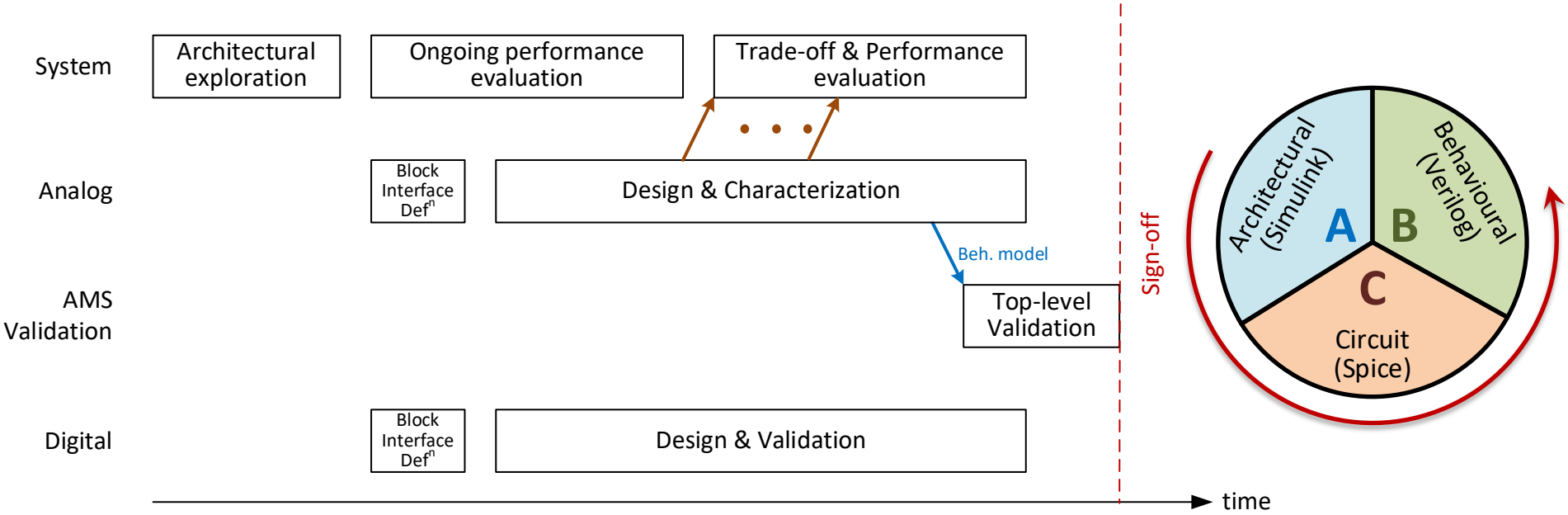


Models Usage Scenarios (3)

- **Behavioral models are manually created**
  - Susceptible to human errors
  - Questionable model quality (what's important vs what is not)
  - Requires manual maintenance
- **Correlation to circuit models is required**
- **Enables top-level design vs spec. validation**
- **IBIS-AMI models = behavioral models**

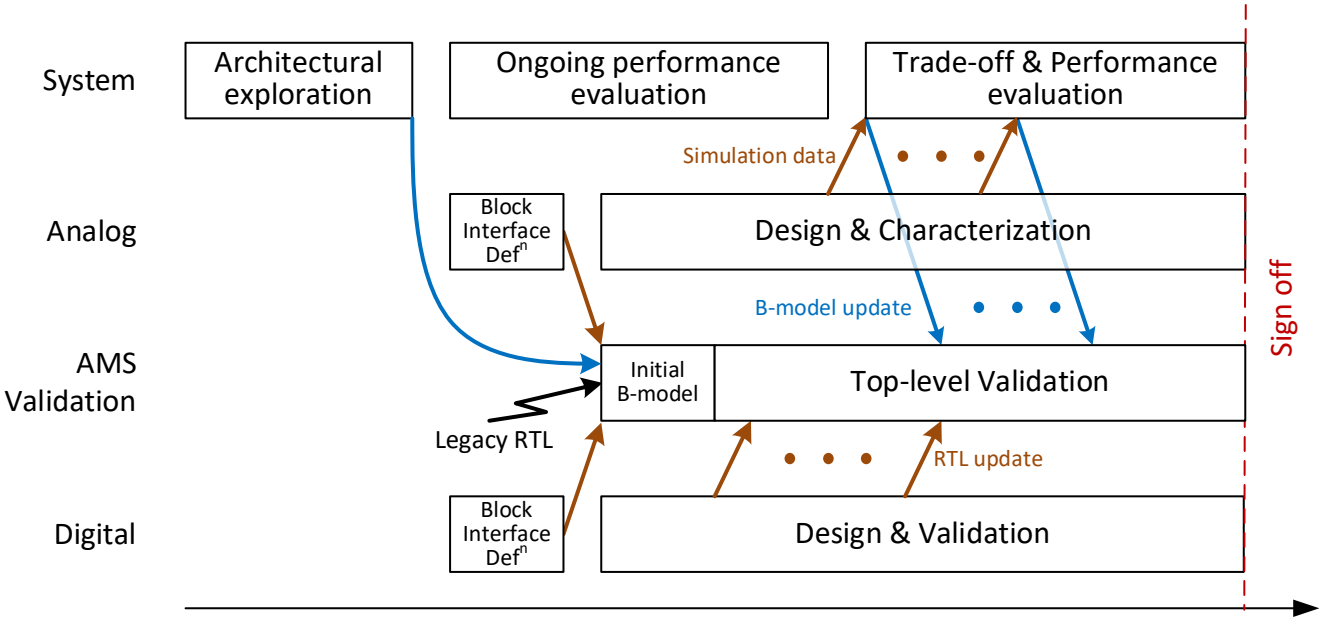


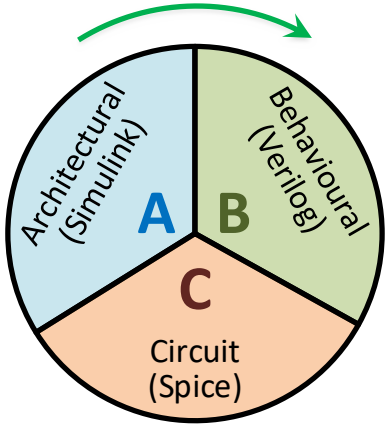
Validation Shift-Left: Early B-Models



- **Generated near project start based on specifications embodied in architectural models**
- **B-models updated and refined as project evolves based on C-model characterization**

Validation Shift-Left: Early B-Models



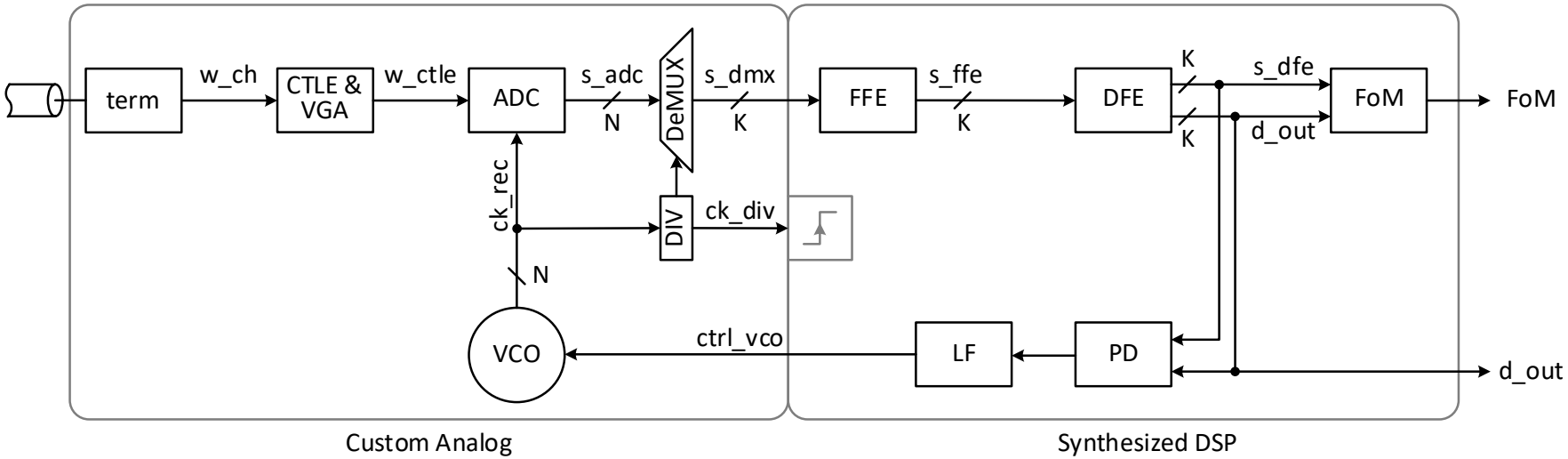


- **Generated near project start based on specifications embodied in architectural models**
- **B-models updated and refined as project evolves based on C-model characterization**

Outline

- Model use throughout a SerDes development life-cycle and the ABCs of SerDes models
- **Generating B-models from A-models**
- C-model design and characterization
- Updating A-model based on C-model characterization
- Automatic B-model refresh from A-model

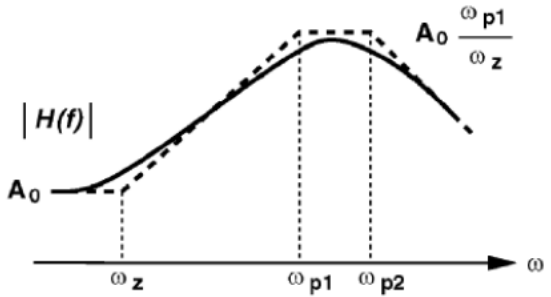
A-Model Hierarchy: Receiver Example



- **Matches C-model hierarchy to allow for future block refinement and export**

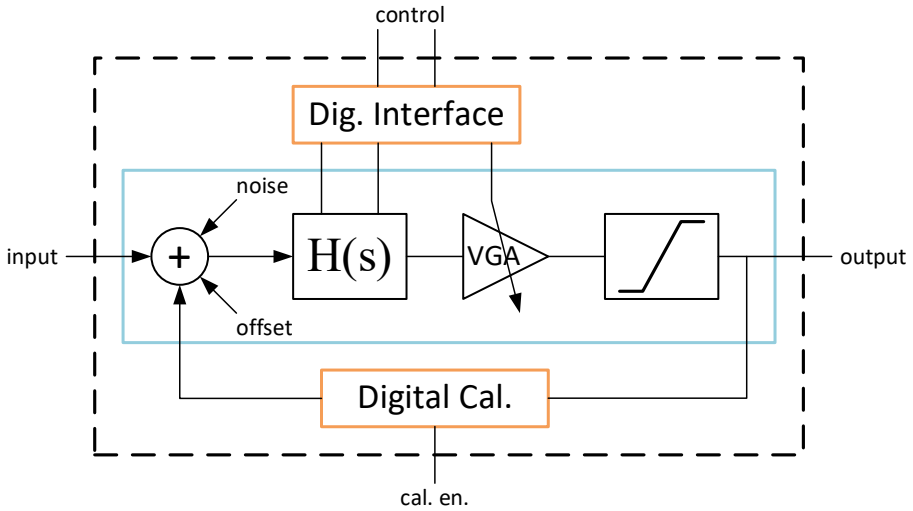
COM-Based CTLE Definition

- **An LTI filter – from the system POV**
- **A continuous-time filter is fully parametrized by pole and zero location, and its gain**
  - Approach used by COM
  - Tuning affected by changing pole or zero locations
- **Represented in discrete-time via**
  - bilinear transform → IIR implementation
  - Sampled impulse response → FIR implementation



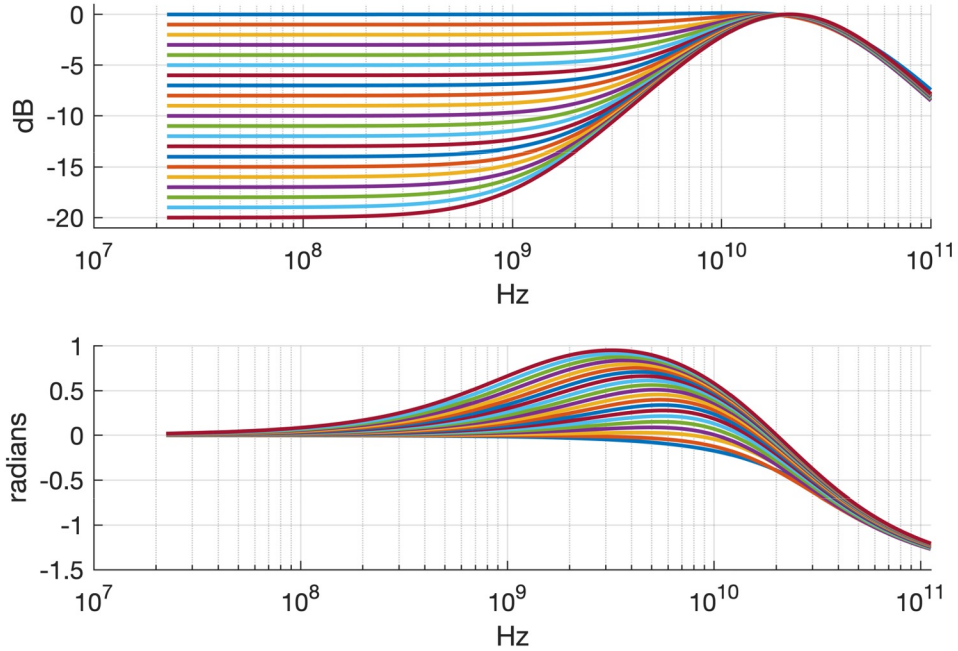
CTLE Circuit Block Diagram

- **Circuit implementation is complex**
  - Continuous-time filtering via analog circuits
  - Tuning via digitally controllable resistor/capacitor banks
- **Imperfect and variable circuit characteristics**
  - Additive circuit noise
  - Dynamic range compression due to voltage headroom limits
  - Transistor mismatches lead to unwanted offsets
  - Offsets can be removed via digital trimming
- **Required behavior is captured by A-model**
  - Some effects do not need to be modeled: e.g., offset and offset-compensation → yet, residual offset is important



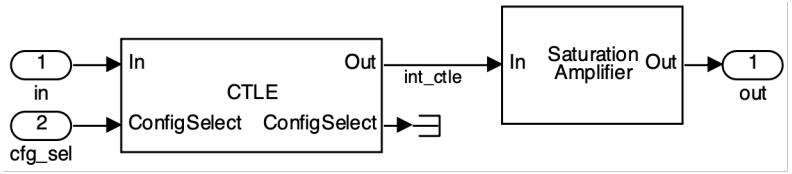
A-Model Target CTLE Frequency Response

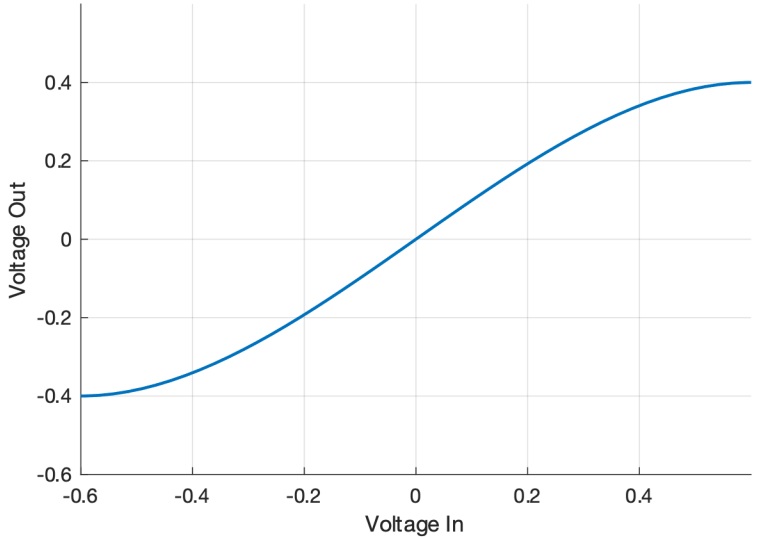
- **Family of digitally selectable filter responses**
  - Boost at 20GHz
  - DC attenuation: -20 to 0dB in 1dB steps
- **Input referred noise needs to be estimated**
  - It is shaped by the CTLE frequency response
- **LTI behavior is captured. What about non-linear effects?**



CTLE Output Nonlinearity Model

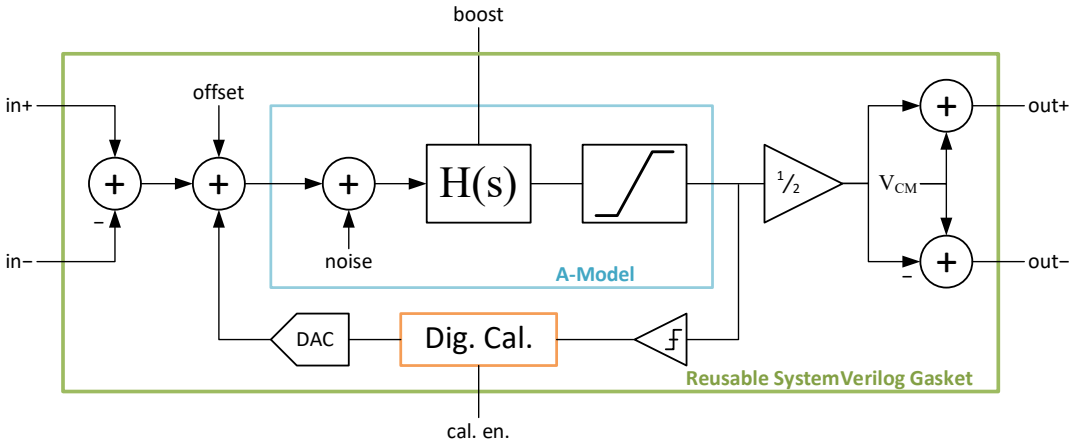
- **CTLE can be input, or output voltage limited**
  - Depending circuit topology chosen, biasing levels, etc.
  - Limiting impacts voltage input/output swing
- **CTLE designed as a differential circuit**
  - Soft limiting – as differential pair becomes less linear
  - Approximated by arctangent function
- **Expected to be output limited**





Complete CTLE B-Model

- **3 constituent pieces**
  - Core functionality – based on A-model
  - Interface gasket – re-usable RTL
  - Digital calibration engine – synthesizable RTL
- **Single-ended to differential conversion performed by re-usable SystemVerilog**
- **DAC and slicer abstracted in example B-model – can be exported A-models**
- **If circuit implementation changes, then A-model functionality will change**



SystemVerilog Gasket

- **Instantiates**
  - Digital calibration engine – synthesizable RTL
  - Core filter behavior – exported A-model
- **SE-to-diff. conversion – 3-lines of code**
- **DAC and comparator modelled directly**
- **The rest is port and net definitions**
- **Offset & common-mode are parameters**

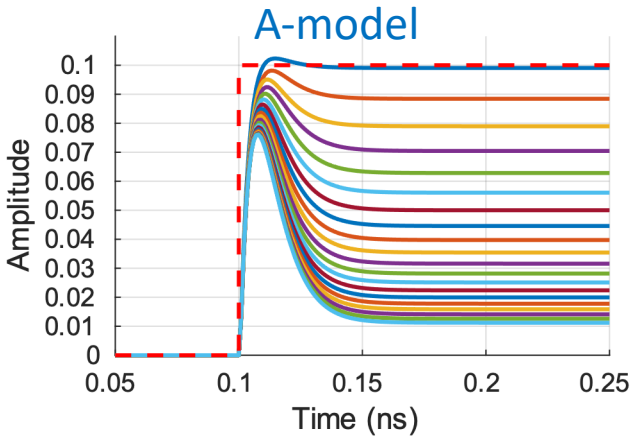
```
module ctle #(
    parameter offset = 0.0,
    parameter cm = 0.5
) (
    input arst_b, // Asynchronous reset
    input cm_clk, // common-mode compensation clock
    input cm_cal_en, // common-mode calibration enable
    input real inp, // CTLE differential input +
    input real inm, // CTLE differential input -
    input byte unsigned boost, // boost setting
    output real outp, // CTLE differential output +
    output real outm  // CTLE differential output -
);

    real in, offset_comp, out;
    reg out_q;
    wire signed [7:0] offset_comp_dig;

    always @(*) begin: CTLE_step
        offset_comp = offset_comp_dig * 0.125/128; // offset DAC model

        in = inp - inm + offset - offset_comp;
        outp = cm + out/2;
        outm = cm - out/2;
    end

    // Clocked comparator
    always @(posedge cm_clk)
            out_q <= cm_cal_en? outp > outm: 1'b0;

// CTLE digital offset calibration
ctle_cm_cal #(.M(8), .N (12)) ctle_cm_cal (.arst_b(arst_b), .clk(cm_clk),
    .en(cm_cal_en), .sense(out_q), .comp(offset_comp_dig));

// CTLE B-model core (based on exported A-model)
CTLE_dpi ctle_core(.ctle_in(in), .cfg_sel(boost), .out(out));

endmodule
```

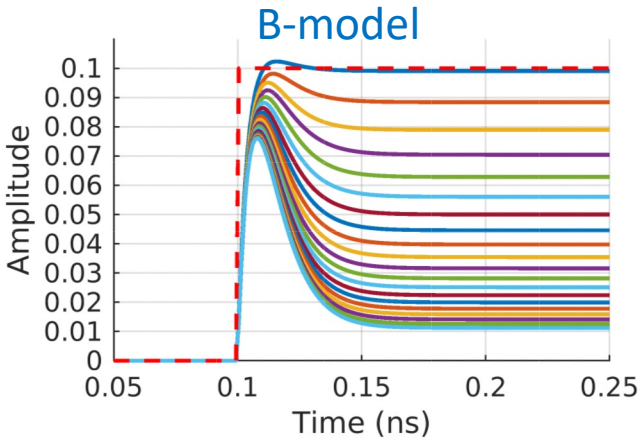
Outline

- Model use throughout a SerDes development life-cycle and the ABCs of SerDes models
- **Generating B-models from A-models**
  - Correlation of A- and B-model
- C-model design and characterization
- Updating A-model based on C-model characterization
- Automatic B-model refresh from A-model

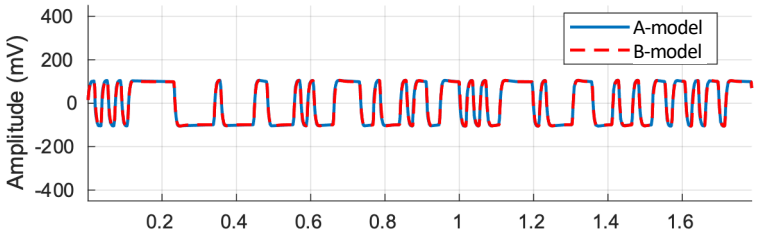
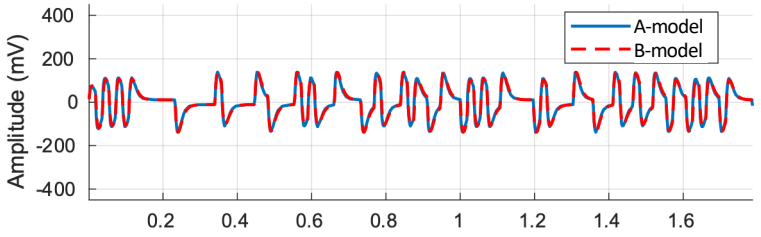
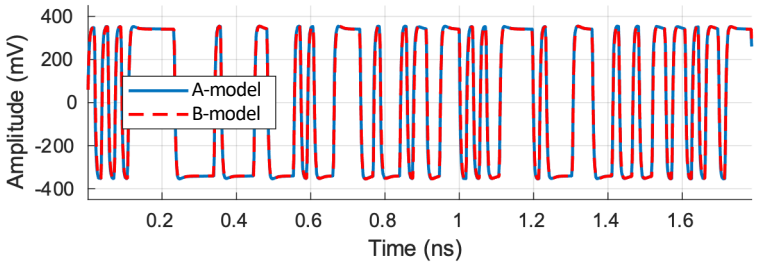
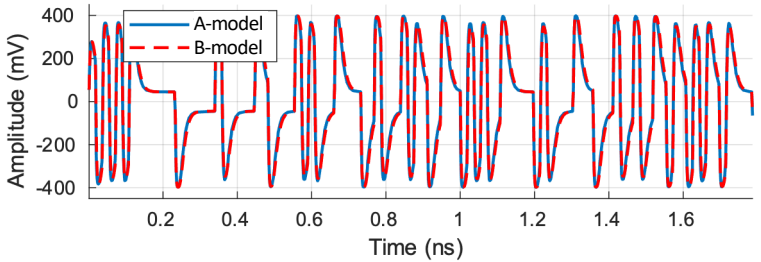
A- vs B-model: Step Response

- Step response fully characterizes CTLE
- 100mV step does not result in output-limiting
- Responses for all CTLE boost settings shown
- No distinguishable difference between A and B models





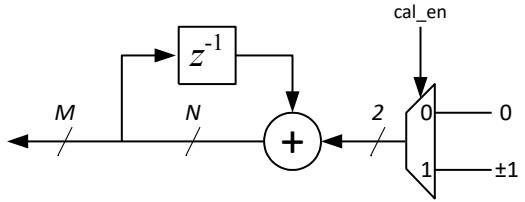
A- vs B-model: Large/Small Amplitude PRBS

	Low-boost setting	High-boost setting
100mV PRBS		
400mV PRBS		

- **Simulink and Verilog outputs overlaid. Matched across amplitude and boost settings.**

CTLE Offset Calibration Engine

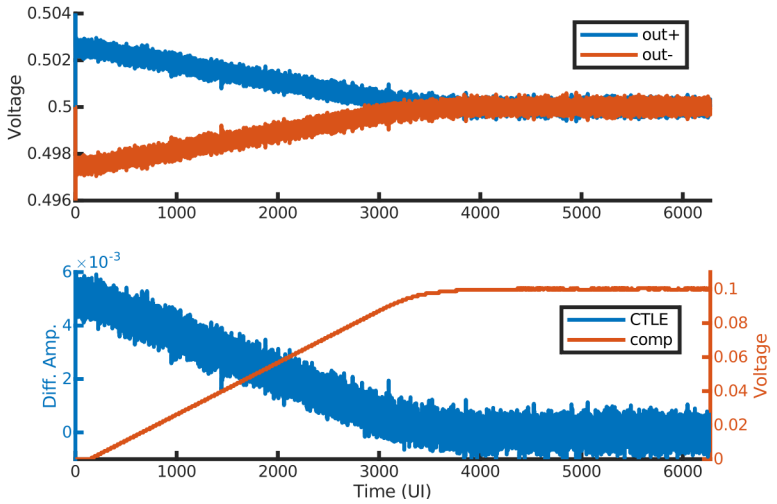
- Input-referred offset is a DC phenomenon
- Measured at CTLE output with shorted inputs
- CTLE output is asynchronously sampled
- ±1 sample decisions are digitally integrated
- Top *N* integrator bits drive compensation DAC
- Compensated when ±1 are equally probable




```
module ctle_cm_cal #(
    parameter M = 8,
    parameter N = 16
) (
    input arst_b, // Asynchronous reset
    input clk,    // clock
    input cal_en, // engine enable
    input sense,  // comparator sense output
    output signed [M-1:0] comp); // compensation code

reg signed [N-1:0] control;

assign comp = control[N – 1 -: M];

always @(posedge clk, negedge arst_b)
    if (~arst_b)
        control = {N{1'd0}};
    else
        control = control + (cal_en? (sense? 1'sd1: -1'sd1): 1'd0);
endmodule
```

CTLE Offset Calibration

- **Verilog-based testbench**
  - Supplies a random offset to CTLE gasket
  - Drives CTLE inputs to common mode
  - CTLE input-referred noise is added
- **Ironically:**
  - Got the feedback polarity wrong!
  - Didn't account for boost dependent DC attenuation
  - Both debugged with early CTLE B-model

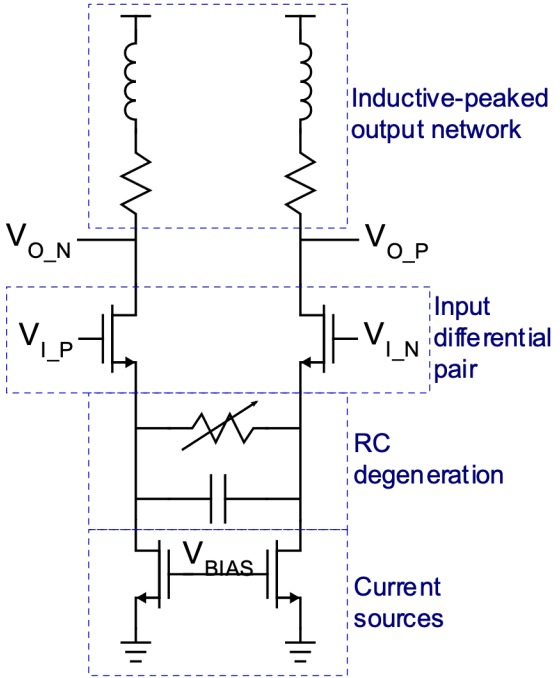


Outline

- Model use throughout a SerDes development life-cycle and the ABCs of SerDes models
- Generating B-models from A-models
- **C-model design and characterization**
- Updating A-model based on C-model characterization
- Automatic B-model refresh from A-model

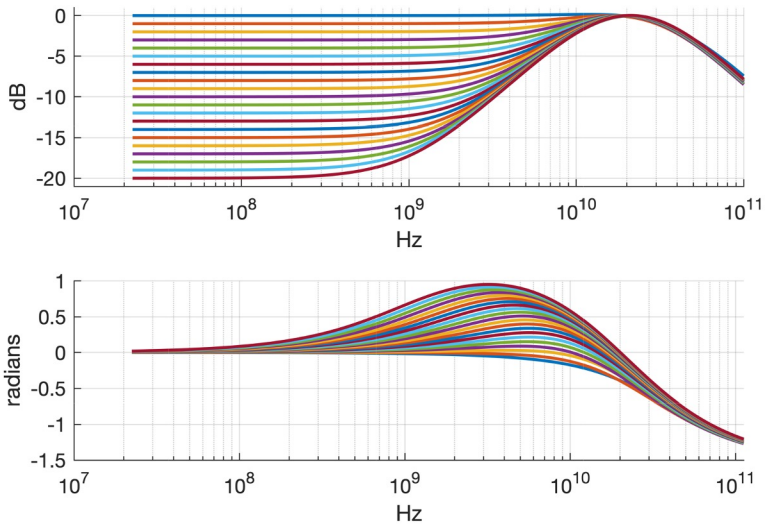
CTLE C-Model

- **Flow demonstrated via simplified CTLE circuit**
- **C-model characterized via simulations**
  - AC simulations to get frequency dependent response
  - DC simulations to get large-signal compression
  - MC simulations to measure expected input-offset
  - Noise simulations to determine input-referred noise
  - etc.
- **Analog simulation results are used to refine A-model**

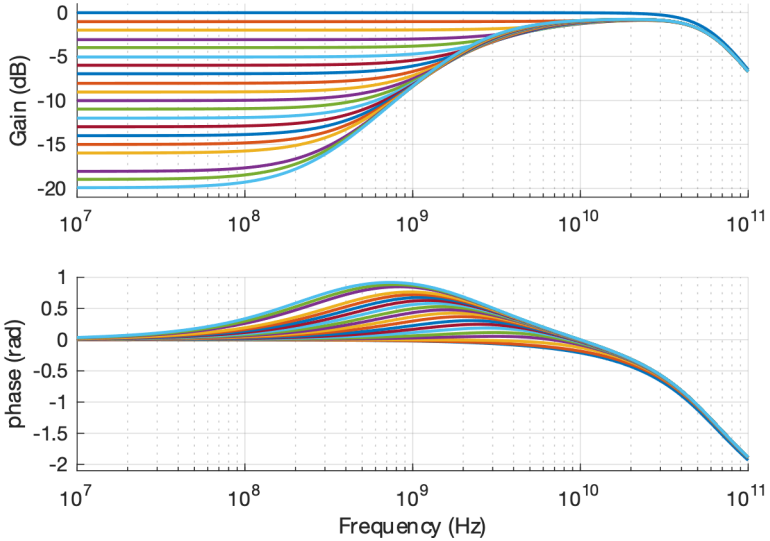


Desired vs Simulated Freq. Response

COM Target Response



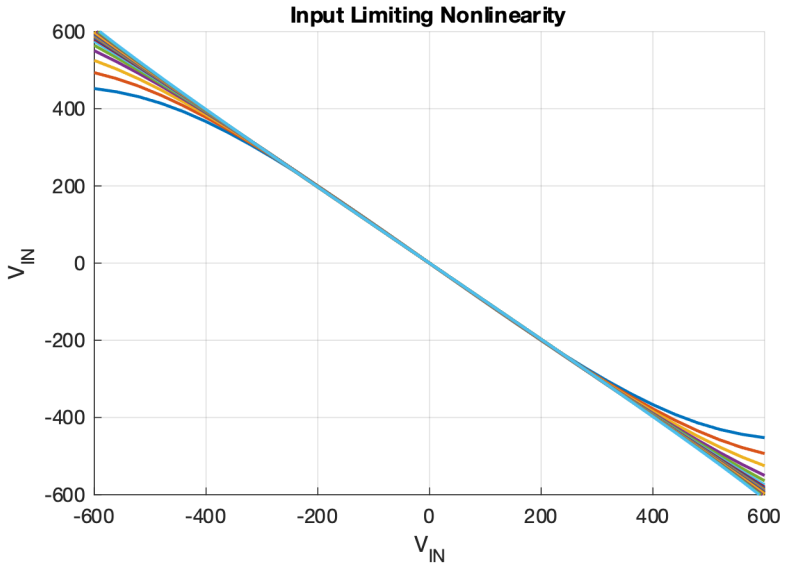
Simulated C-model



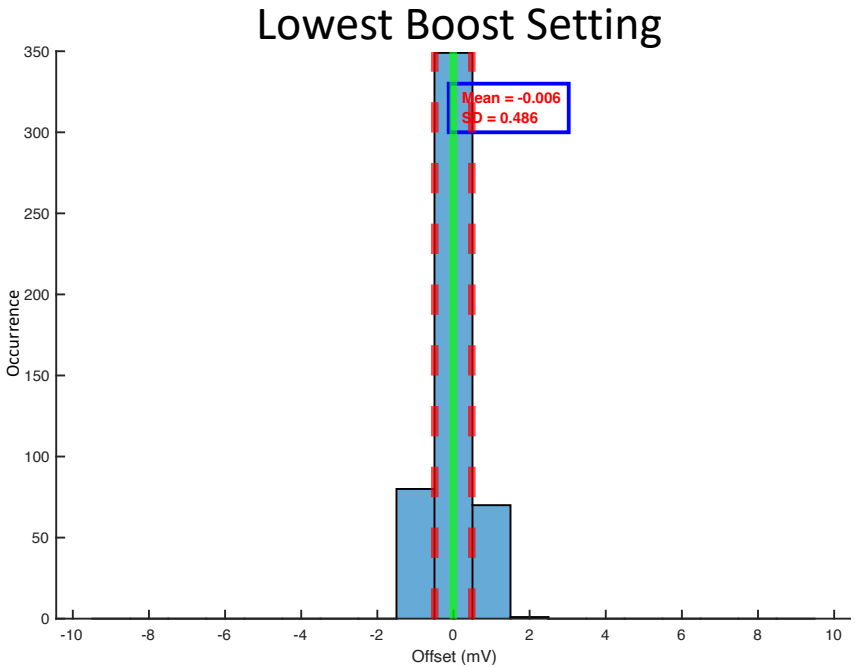
- **Visible disparity used to highlight change due to A-model refresh**

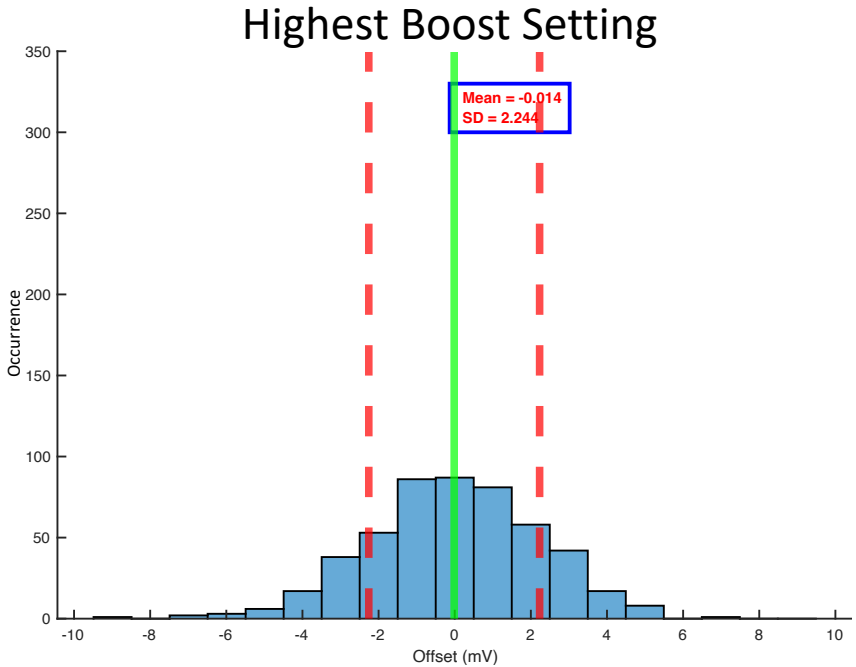
Simulated Large Signal Compression

- Differential DC sweep about common mode
- Simulated V_{OUT} vs V_{IN} curve normalized by DC gain to get input or output limiting
- Plot shows CTLE is input limited, rather than output limited



Simulated CTLE Input-Referred Offset





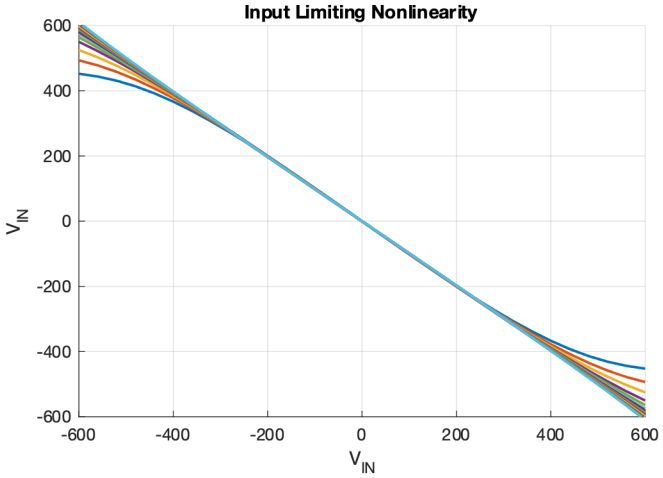
- **The simulated CTLE offset is within the offset compensation range (±100mV)**
- **Information can used to decrease the feedback DAC dynamic range: no impact on digital**

Outline

- Model use throughout a SerDes development life-cycle and the ABCs of SerDes models
- Generating B-models from A-models
- C-model design and characterization
- **Updating A-model based on C-model characterization**
- Automatic B-model refresh from A-model

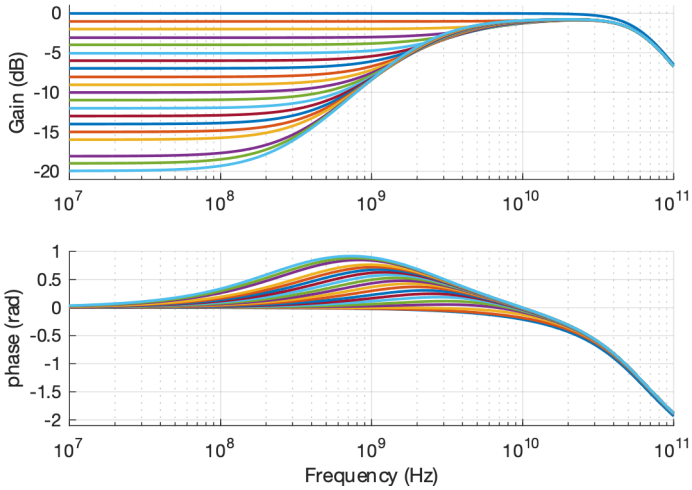
A-Model Update

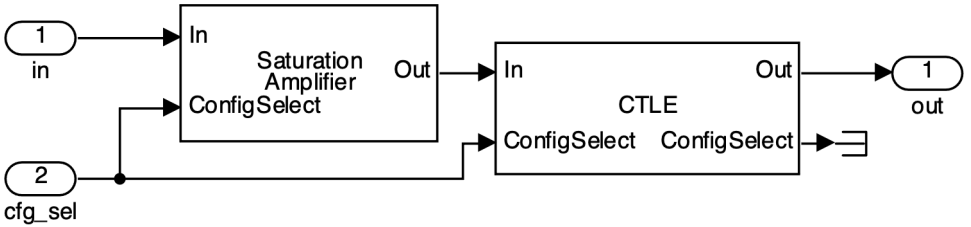
- **Simulated input limiting used to configure saturating amplifier**
- **Rational fit used to match simulated frequency response**
  - Compact gain, pole & zero representation
- **GPZ matrix used to configure CTLE model**



A-Model Update

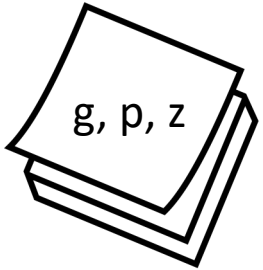
- **Simulated input limiting used to configure saturating amplifier**
- **Rational fit used to match simulated frequency response**
  - Compact gain, pole & zero representation
- **GPZ matrix used to configure CTLE model**

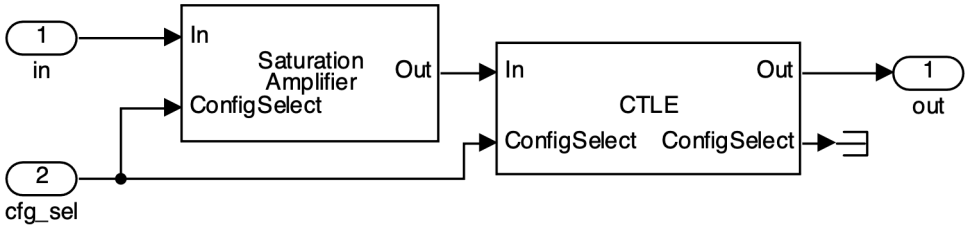




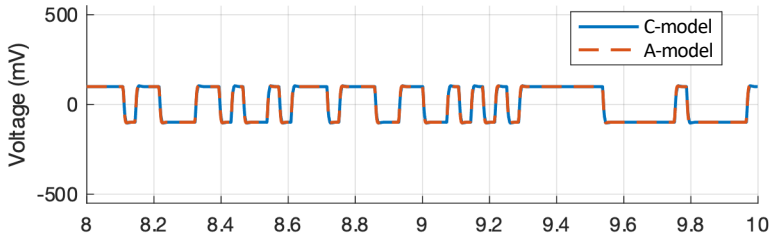
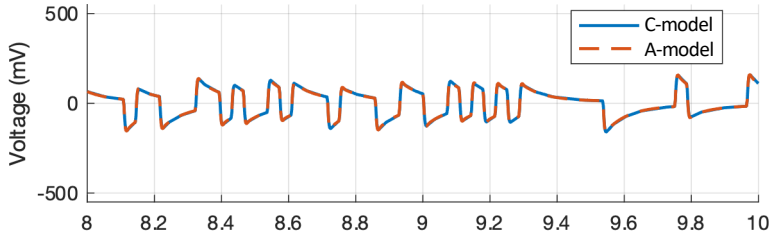
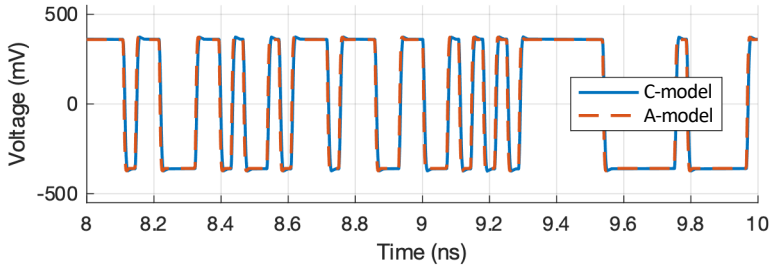
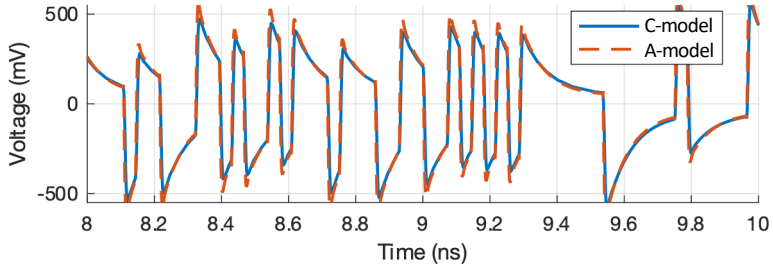
A-Model Update

- **Simulated input limiting used to configure saturating amplifier**
- **Rational fit used to match simulated frequency response**
  - Compact gain, pole & zero representation
- **GPZ matrix used to configure CTLE model**





C-to-A model Correlation

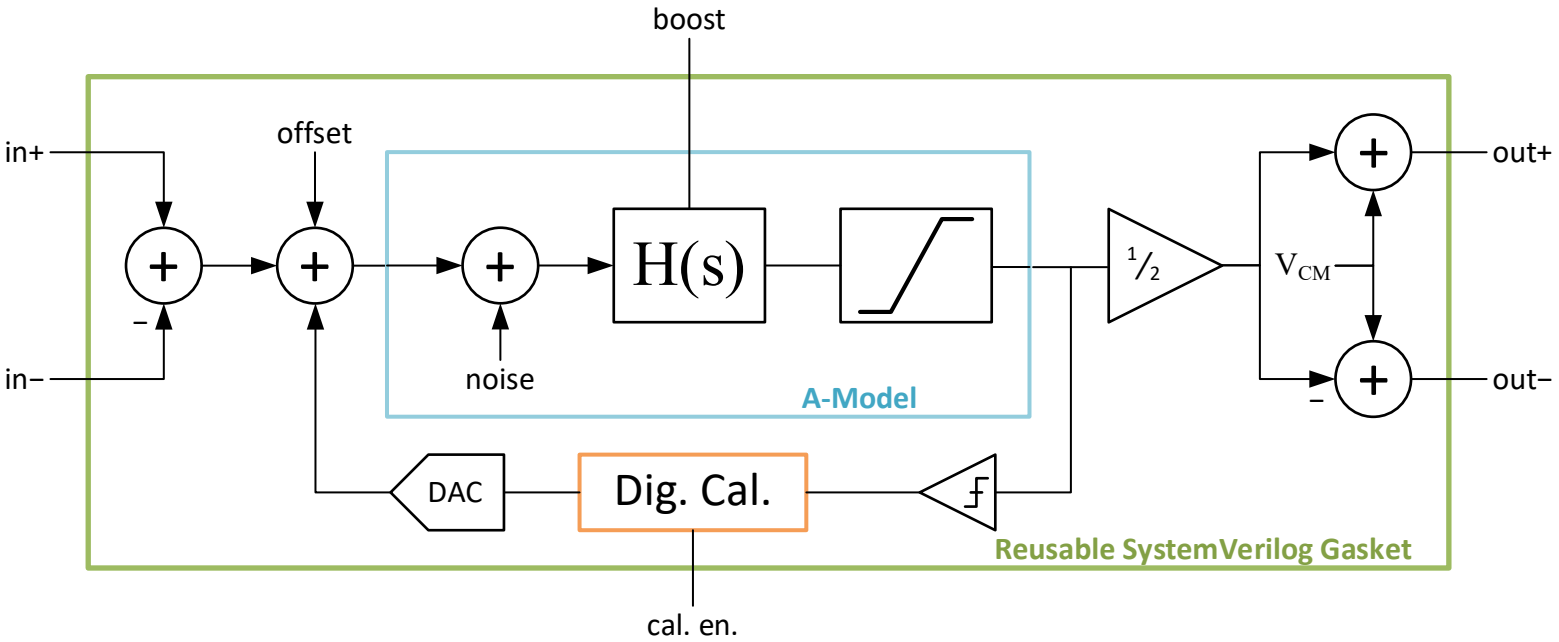
	Low-boost setting	High-boost setting
100mV PRBS		
400mV PRBS		

- **PRBS sequence used to correlate C-model to A-model behavior**
- **Well matched, but could do better by modeling output limiting behavior as well**

Outline

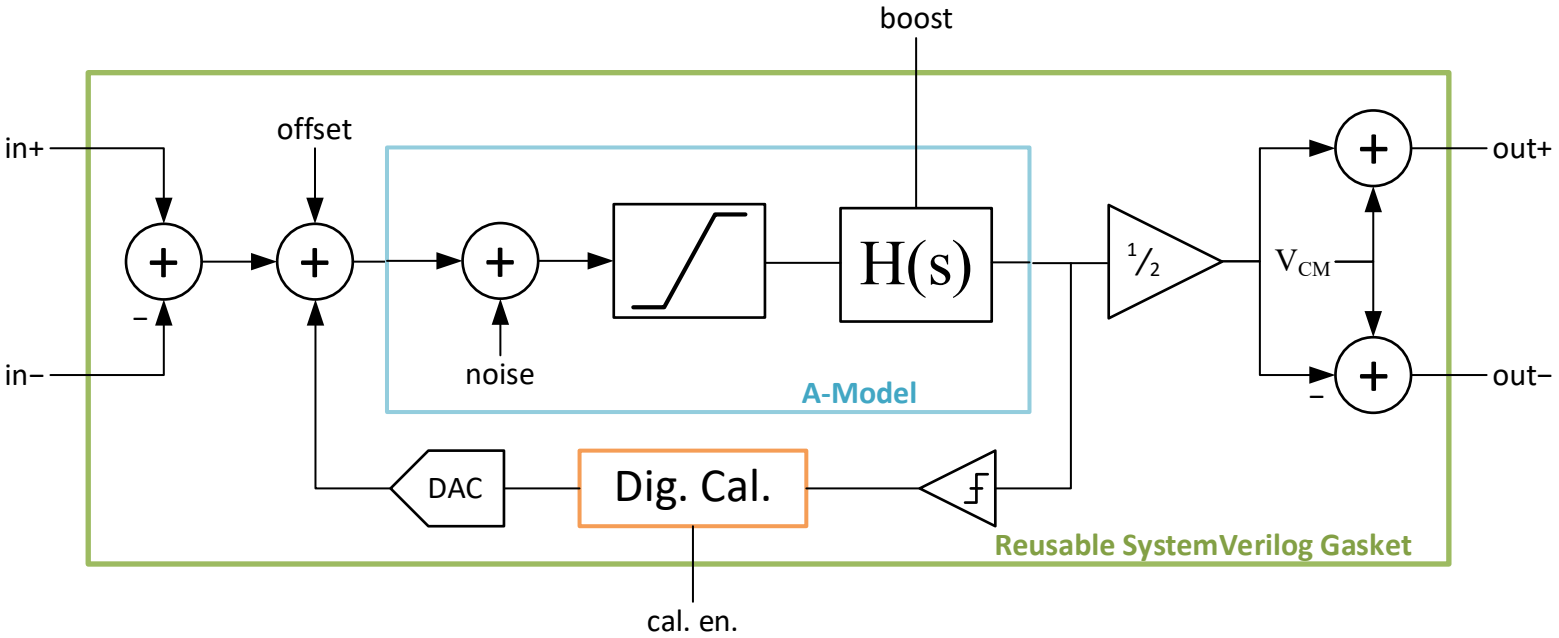
- Model use throughout a SerDes development life-cycle and the ABCs of SerDes models
- Generating B-models from A-models
- C-model design and characterization
- Updating A-model based on C-model characterization
- **Automatic B-model refresh from A-model**

How do we refresh the B-Model



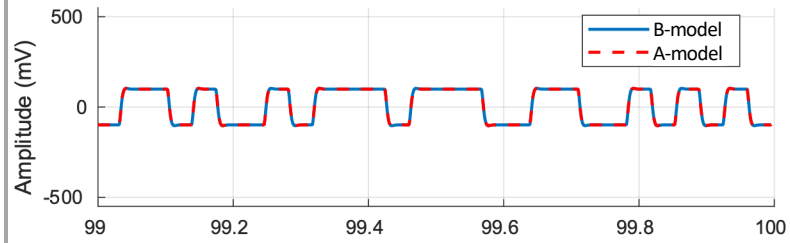
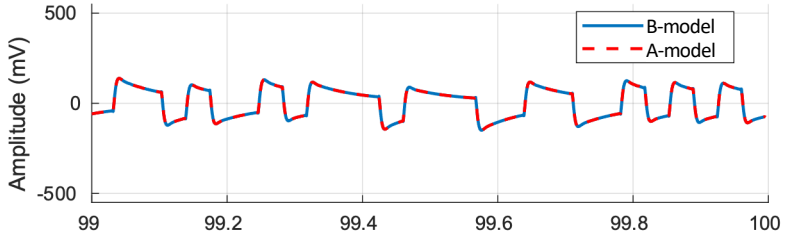
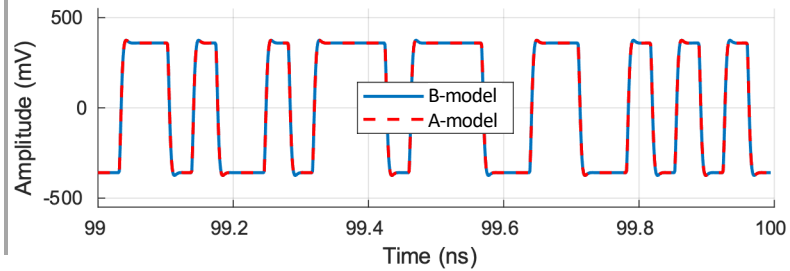
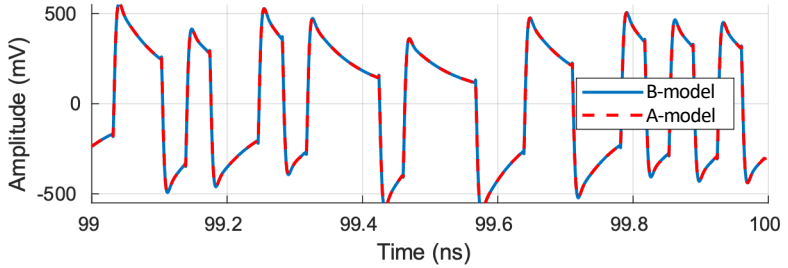
- **Simple! Just re-export the DPI, based on the updated Simulink model**

How do we refresh the B-Model



- **Simple! Just re-export the DPI, based on the updated Simulink model**

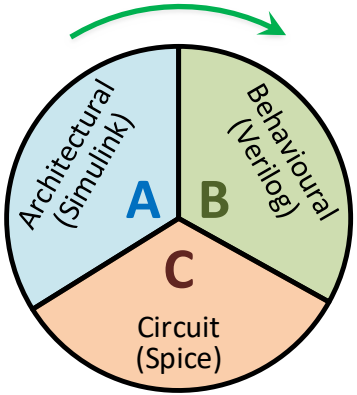
B-to-A Model Correlation

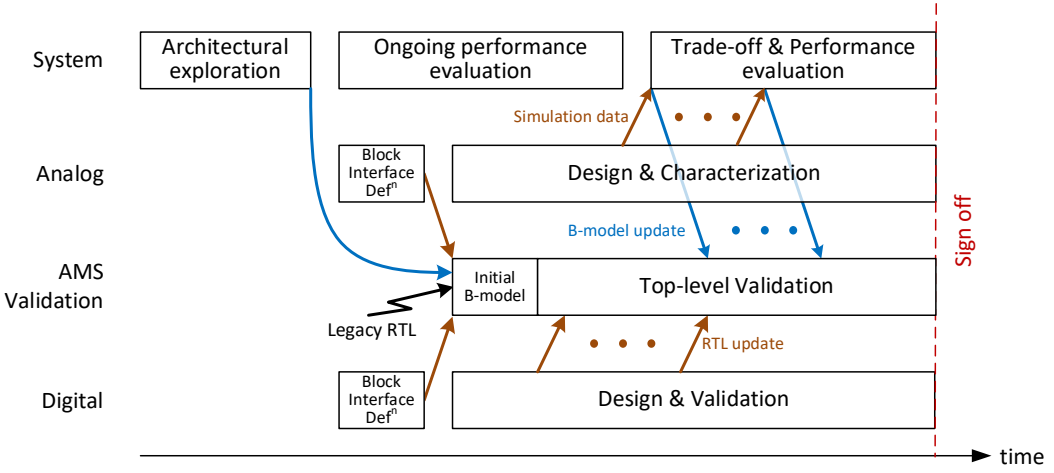
	Low-boost setting	High-boost setting
100mV PRBS		
400mV PRBS		

- **Very well-matched behavior**
- **Improvements in C-to-A correlation would be manifested in B-model as well**

Conclusion

- **Initial A-models embody circuit specs.**
- **Demonstrated flow for a CTLE**
  - Early B-model based on architectural models
  - C-model was designed and characterized
  - Updated A-model based on simulation data
  - Regenerated updated B-model
- **Enables validation shift-left**
  - Earlier start
  - Better test coverage
  - Higher sign-off confidence





Follow up with us after the conference

- Engage with us info@serialinksystems.com
- Visit us at www.serialinksystems.com