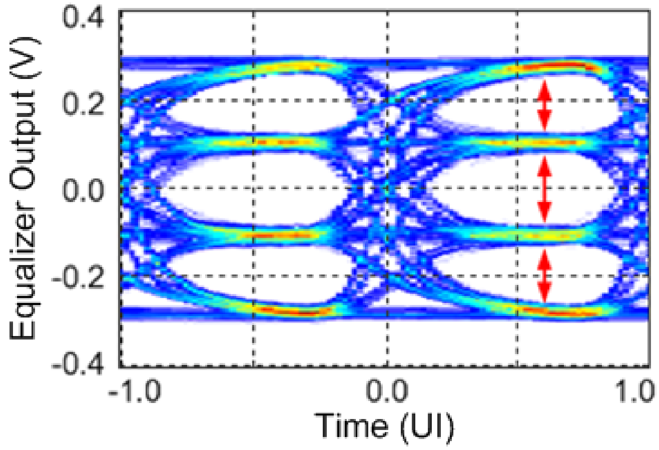
An Event-Driven Simulation of Analog/Mixed-Signal Systems and Its Implications to Property Assertions

Prof. Jaeha Kim
Mixed-Signal IC and System Group
Seoul National University

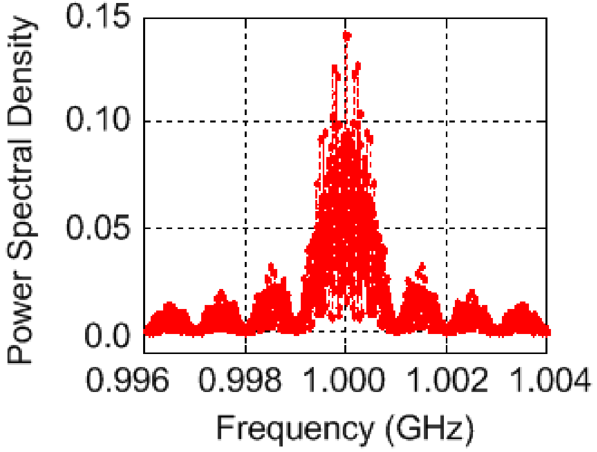
XMODEL and Property Assertions

- XMODEL is a SystemVerilog-based event-driven simulator for analog and mixed-signal systems
  - With 100~1000x speed-up over Verilog-AMS or SPICE
- This talk addresses ways to assert properties in XMODEL using SystemVerilog Assertions

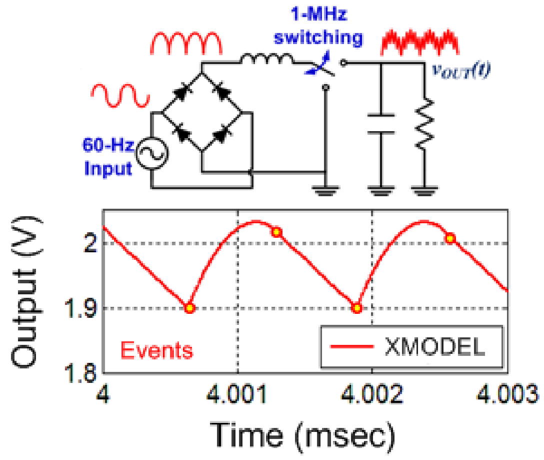
High-Speed I/O Interfaces



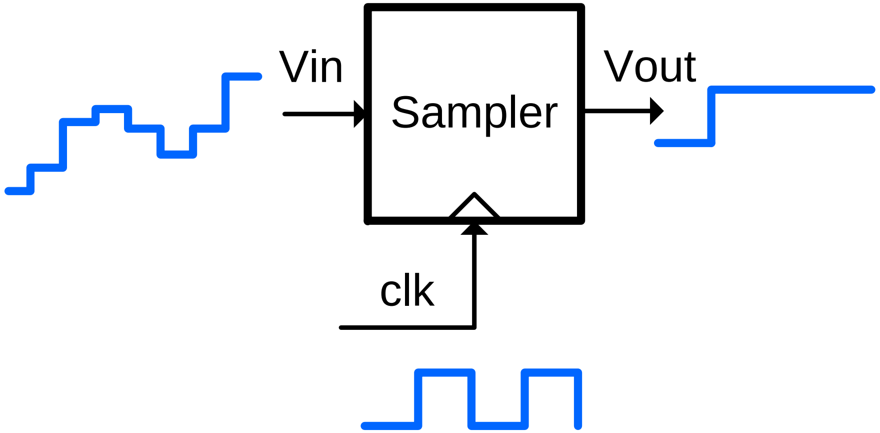
RF/Wireless Transceivers



Power Converters



Problems with Straight Verilog

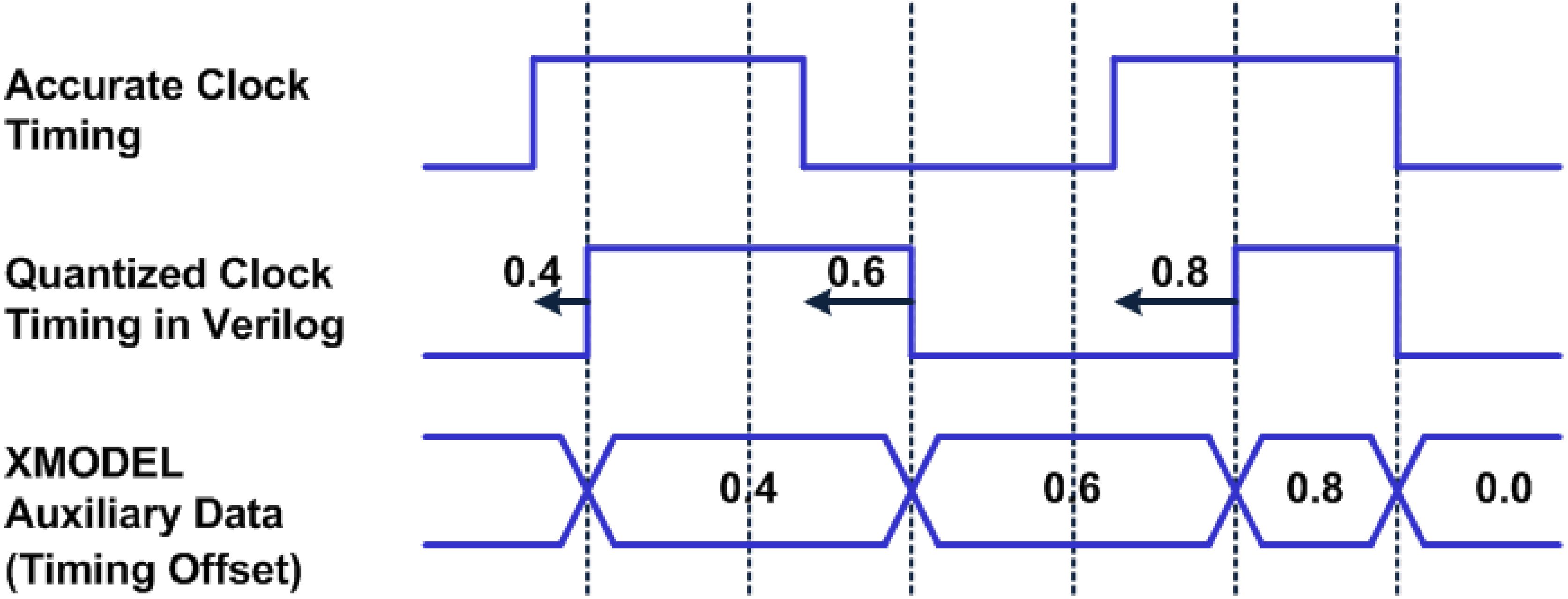


```
`timescale 10ps/10ps

module sampler(clk, Vin, Vout);
   always @(posedge clk) begin
      Vout <= (Vin>Vth) ? 1:0;
   end
endmodule
```

- In case we want to model the effects of jitter and ISI on BER of a high-speed receiver
  - Sub-ps simulation time-step is required to express clock jitter and input data waveform
  - Simulation speed has to be sacrificed to retain accuracy

Case 1: Expressing Clock Timing

- Normally, to express accurate clock timing in PLLs one needs very fine simulation time steps
- XMODEL removes this dependency by adding an explicit timing information to the signal (***xbit***)


Accurate Clock Timing
Quantized Clock Timing in Verilog
0.4
0.6
0.8
XMODEL Auxiliary Data (Timing Offset)
0.4
0.6
0.8
0.0

Data Supplementation

- The first idea was to add auxiliary information to the signal in order to express it accurately
  - Without relying on fine simulation time steps
  - For clock signals, it is the time offset
- The ***struct*** type in SystemVerilog is handy in keeping the language simple

```
typedef struct {
    bit value;
    real t_offset;
} xbit;
```

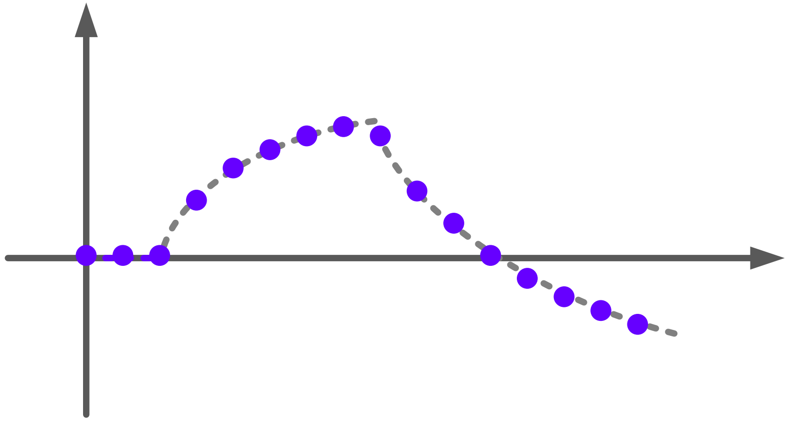
```
module divider(
    input xbit clkin
    output xbit clkdiv
);
```

Case 2: Expressing Analog Signals

- Instead of using a series of time-value pairs, XMODEL expresses signals in a functional form:

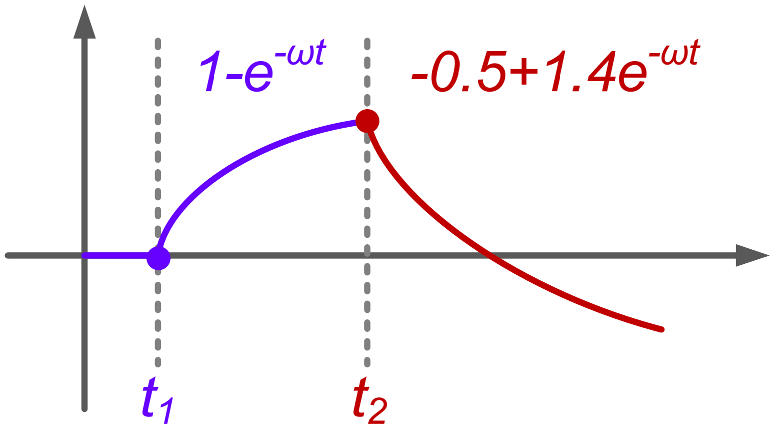
$$x(t) = \sum_i c_i t^{m_i} e^{-a_i t} u(t - t_i)$$

SPICE



Accuracy relies on fine time step

XMODEL

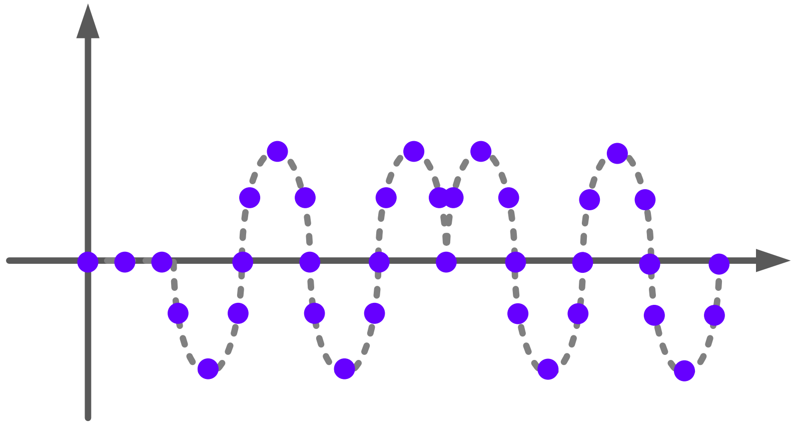


Events occur only when the coefficients are updated

Expressing Analog Signals (2)

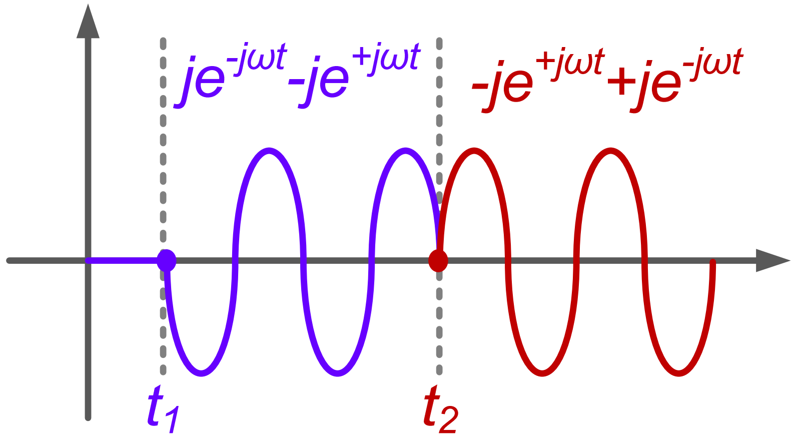
- The form can accurately express all possible outputs of linear systems requiring far fewer events
  - Including sinusoids, exponentials, ramps, steps, etc.

SPICE



# points must scale with the carrier frequency

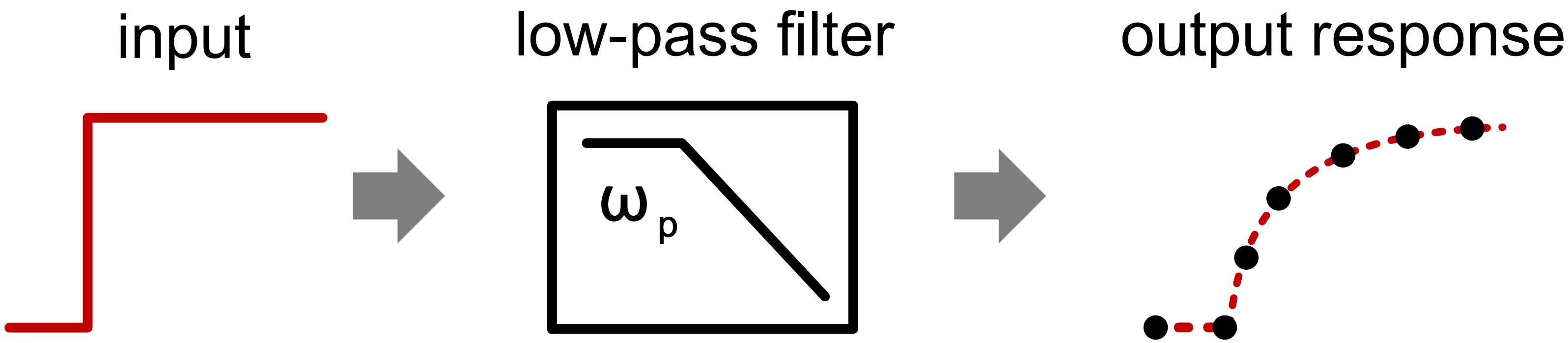
XMODEL


$je^{-j\omega t}-je^{+j\omega t}$
$-je^{+j\omega t}+je^{-j\omega t}$
t_1
t_2

# events can scale with the modulation frequency

Simulating System Responses

- Numerically integrating an ODE results in a trade-off between speed and accuracy



input

low-pass filter

output response

$$\dot{x}(t) \approx \frac{x(t+h)-x(t)}{h}$$

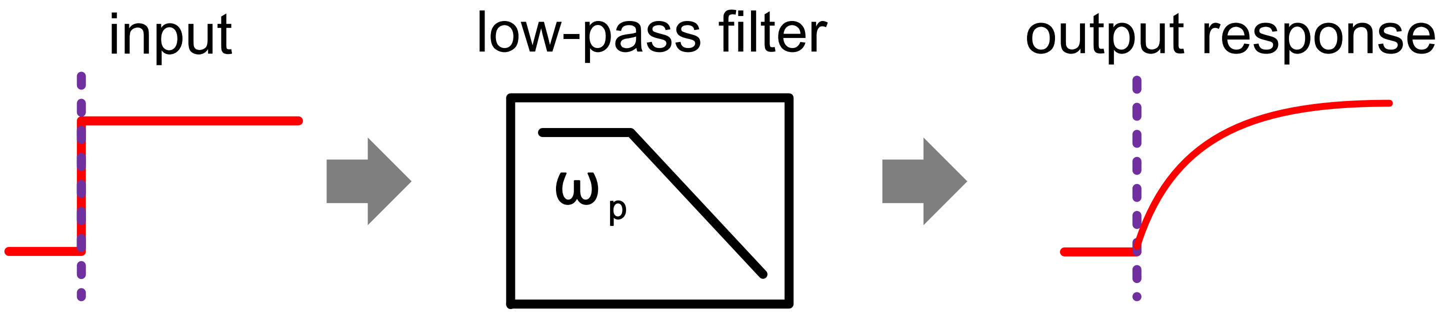
Speed and accuracy depends on the time step *h*

Simulating System Responses (2)

- Signals have an alternate s-domain representation:

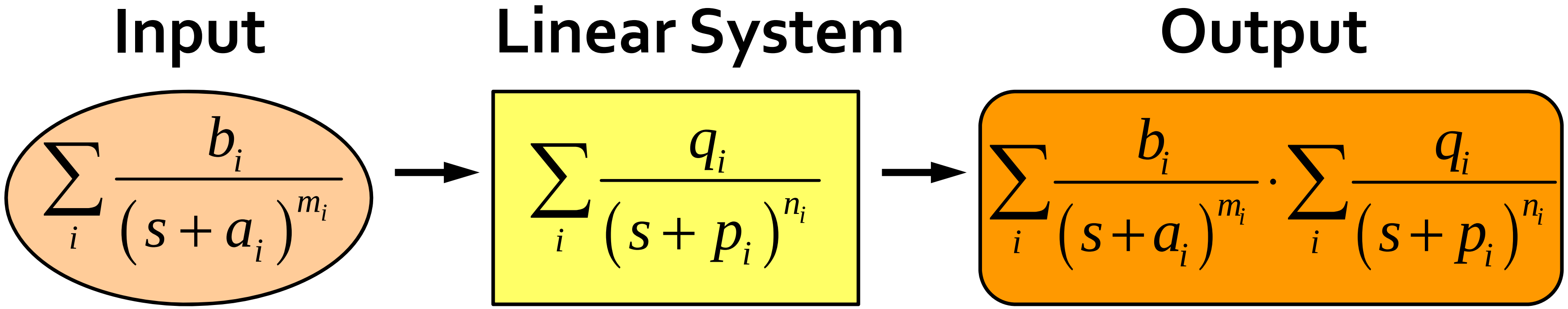
$$x(t) = \sum_i c_i t^{m_i - 1} e^{-a_i t} u(t) \xrightarrow{\mathcal{L}} X(s) = \sum_i \frac{b_i}{(s + a_i)^{m_i}}$$

- With this, the response of a linear system can be computed algebraically without integrating ODE:



$$\frac{1}{s} \times \frac{\omega_p}{\omega_p + s} = \frac{1}{s} - \frac{1}{\omega_p + s}$$

Event-Driven Simulation of Analog



$$\sum_i \frac{b_i}{(s+a_i)^{m_i}} \rightarrow \sum_i \frac{q_i}{(s+p_i)^{n_i}} \rightarrow \sum_i \frac{b_i}{(s+a_i)^{m_i}} \cdot \sum_i \frac{q_i}{(s+p_i)^{n_i}}$$

Input Event: change in parameters

$$\{(a_i, b_i, m_i)'s\}$$

System Model: s-domain TF

$$\{(p_i, q_i, n_i)'s\}$$

Output Event: computed by partial fraction decomposition

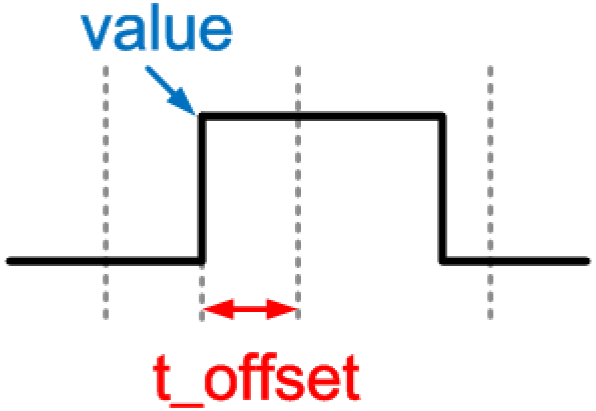
$$= \sum_i \frac{c_i}{(s+a_i)^{m_i}} + \sum_j \frac{d_j}{(s+p_j)^{n_j}}$$

$$\{(a_i, c_i, m_i), (p_j, d_j, n_j)\}$$

xbit and *xreal*

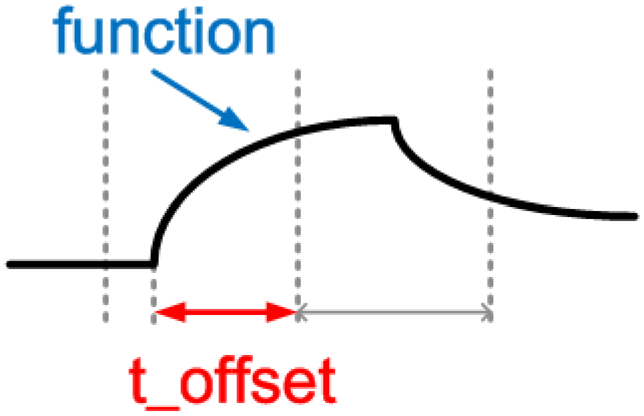
- ***xbit*** for digital signals

```
typedef struct {
        bit value;
        real t_offset;
} xbit;
```

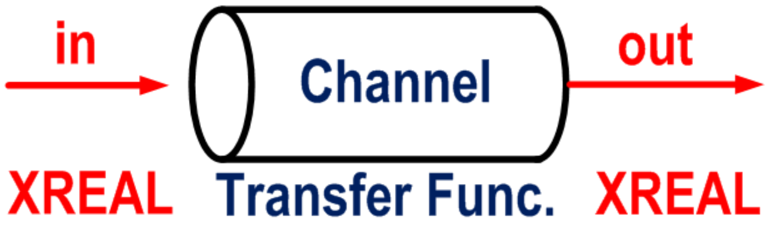


- ***xreal*** for analog signals

```
typedef struct {
        chandle param_set;
        real t_offset;
        event flag;
} xreal;
```



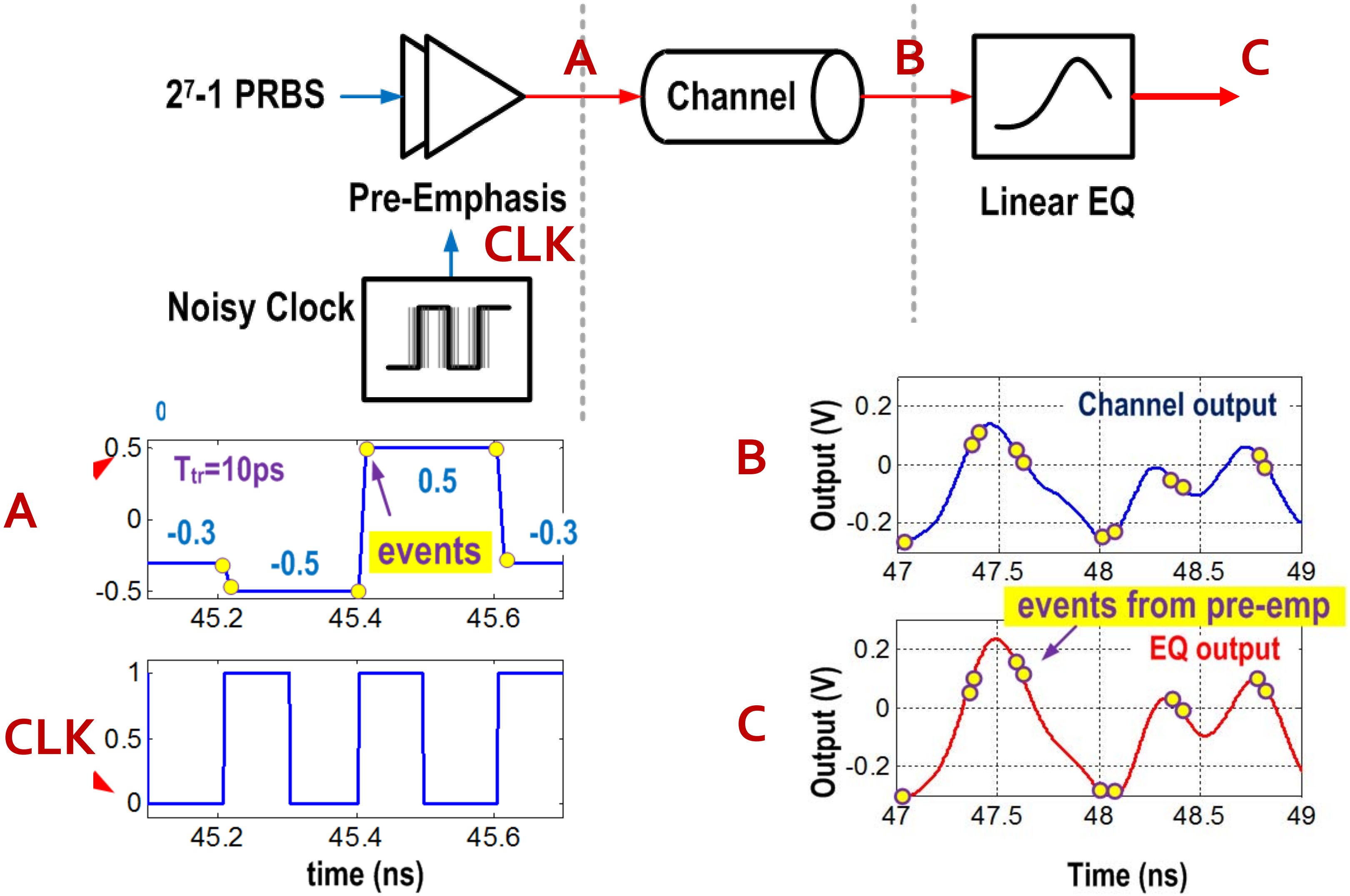
Channel Model Example



- ***in.flag*** indicates that ***in.param_set*** is changed
- ***eval_system()*** calculates s-domain output
- ***out.flag*** notifies subsequent blocks of the change events

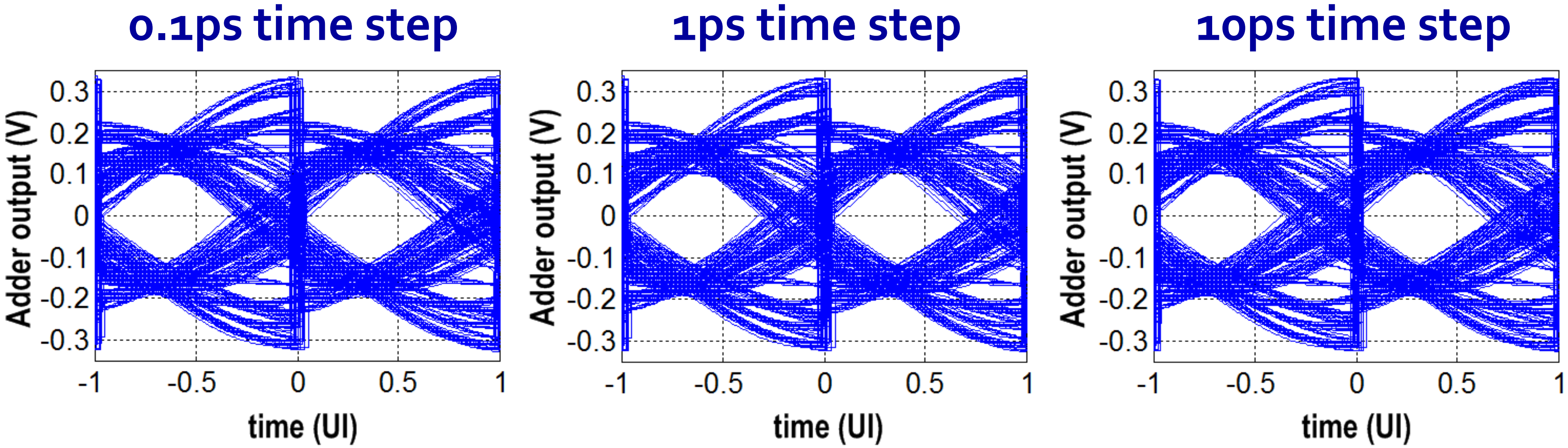
```
module channel (
  input xreal in,
  output xreal out);

  // channel transfer function
  chandle TF_channel;

  always @(in.flag) begin
    out.param_set = eval_system(
         in.param_set,TF_channel);
    out.t_offset = in.t_offset;
    -> out.flag;
   end
endmodule
```

Channel Simulation Results



XMODEL Simulation Accuracy

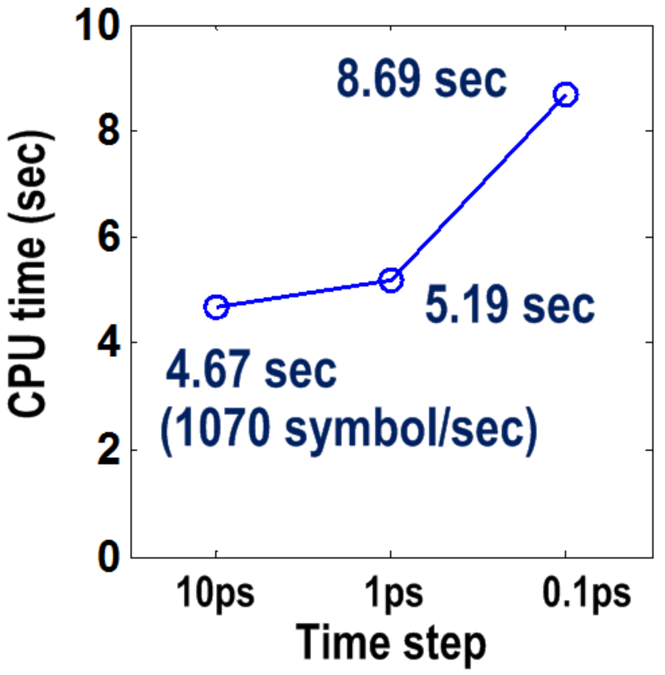
- Since the waveforms are expressed in functional forms, accuracy depends weakly on the time step
- The eye diagrams of DFE receiver vs. time steps:

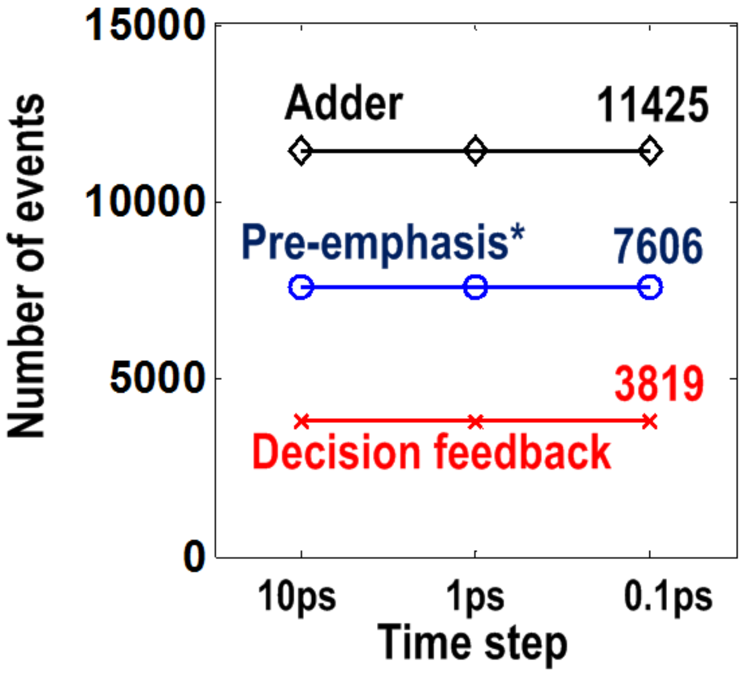


0.1ps time step **1ps time step** **10ps time step**

XMODEL Simulation Speed

- Simulation speed is also very weakly dependent on the time step
- 100~1000x faster than Verilog-AMS or SPICE





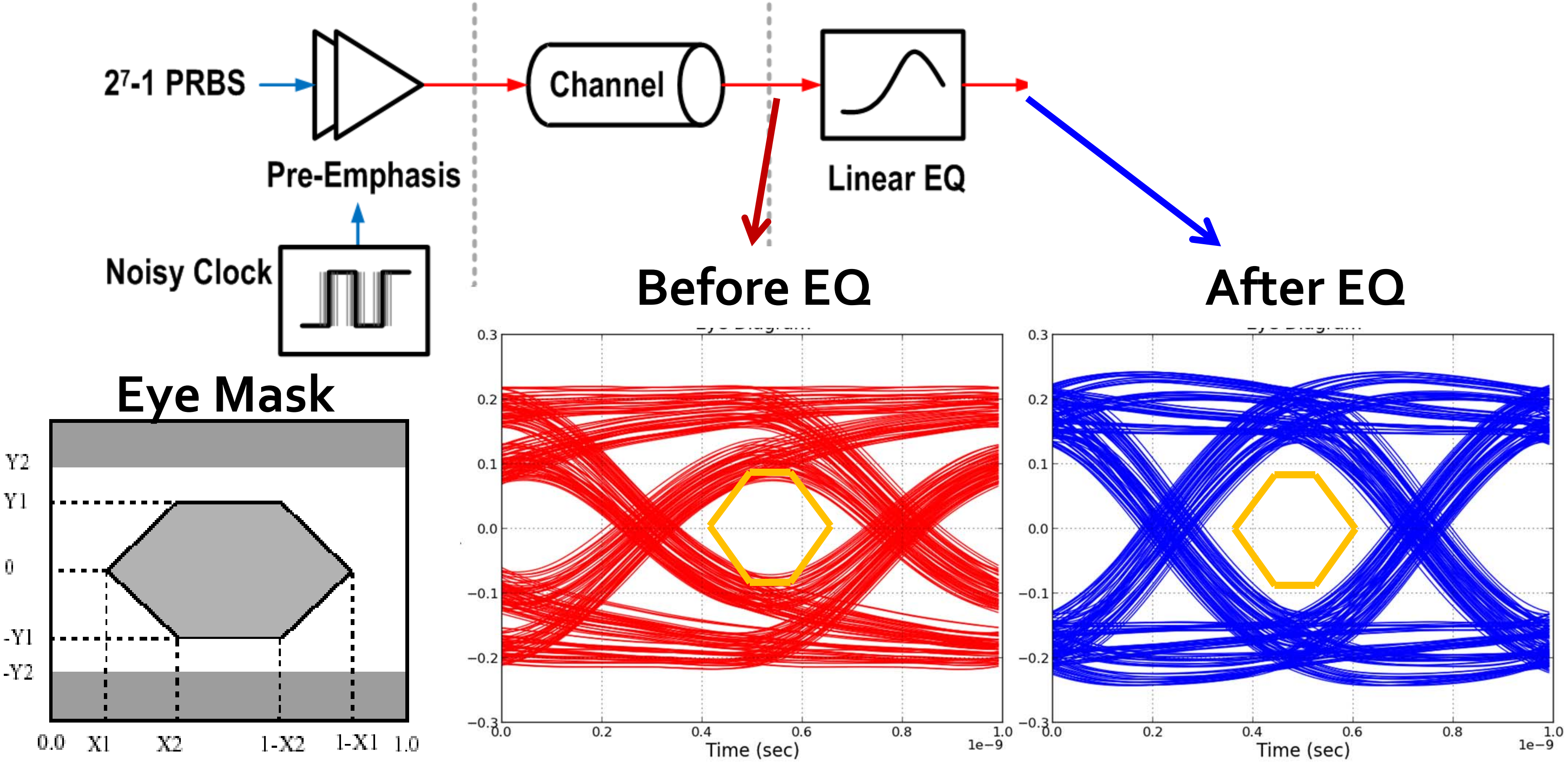
SystemVerilog Assertions (SVA)

- SystemVerilog Assertions (SVA) is a language extension to describe property assertion checks
- Since XMODEL is based on SystemVerilog, it can use SVA to check AMS properties via simulation

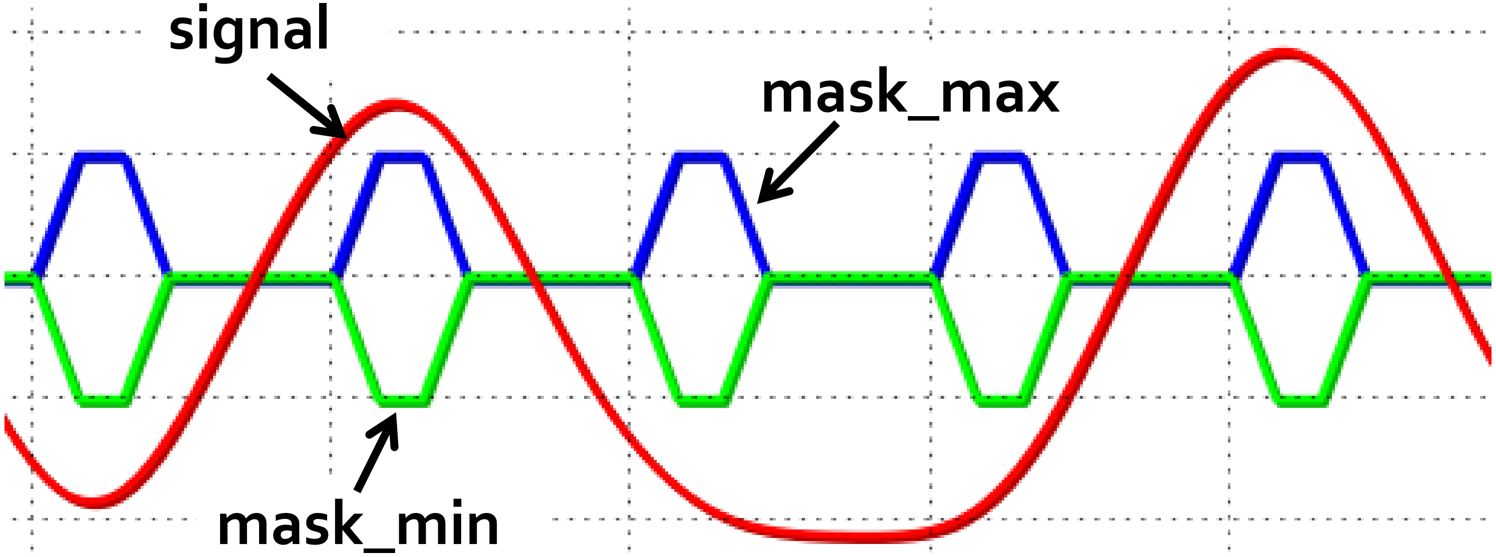
```
property handshake;
   @(posedge clk) REQ |-> ##[1:2] ACK;
endproperty

assert property (handshake);
cover property (handshake);
```

Time-domain Property Assertion

- Check the eye opening of the received signal



Eye-Opening Assertion

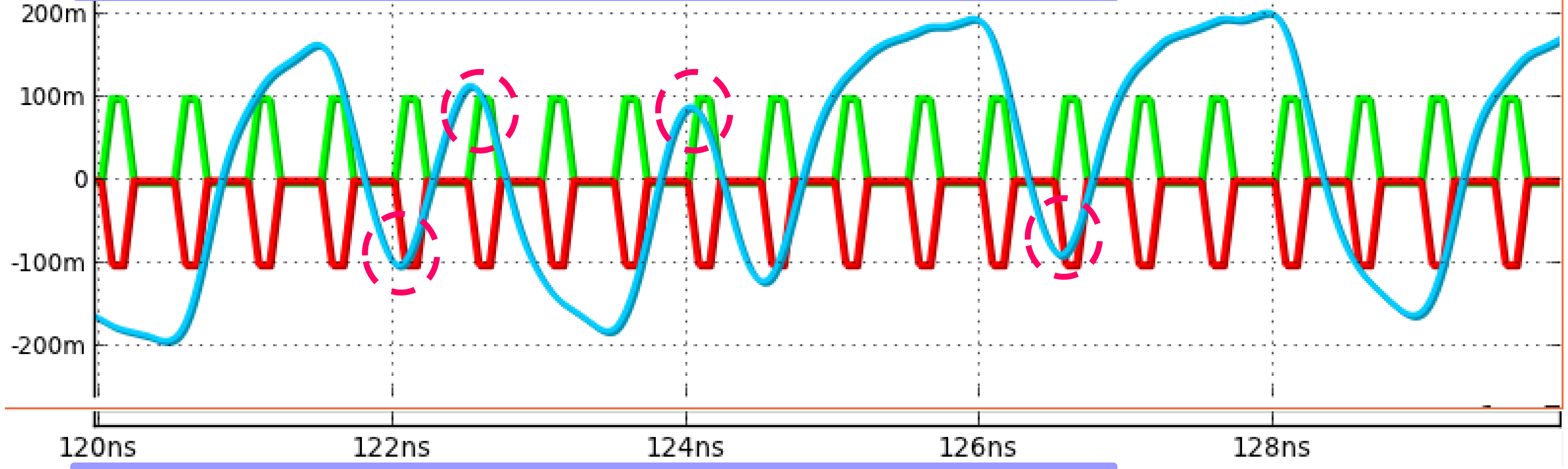


```
slice check_max(.in_pos(signal), .in_neg(mask_max),.out(flag_max));
slice check_min(.in_pos(signal), .in_neg(mask_min),.out(flag_min));

property eye_is_open;
   flag_max || ~flag_min;
endproperty

assert property (eye_is_open);
```

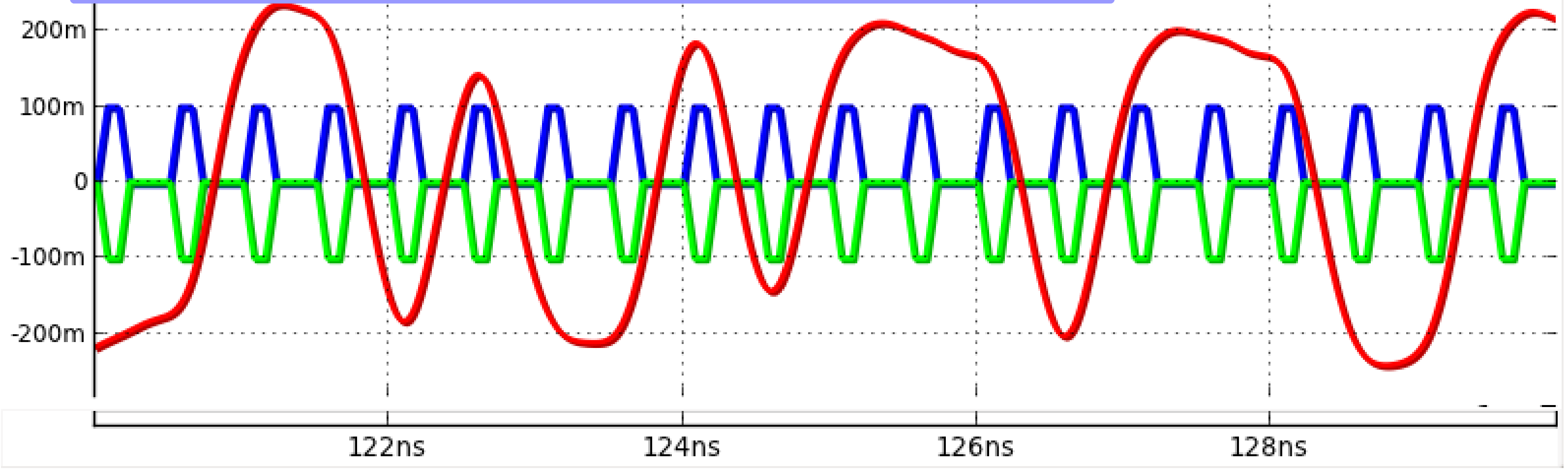
signal >= mask_max or
signal <= mask_min
at all times

Assertion Check Results

Before EQ: Violations Found



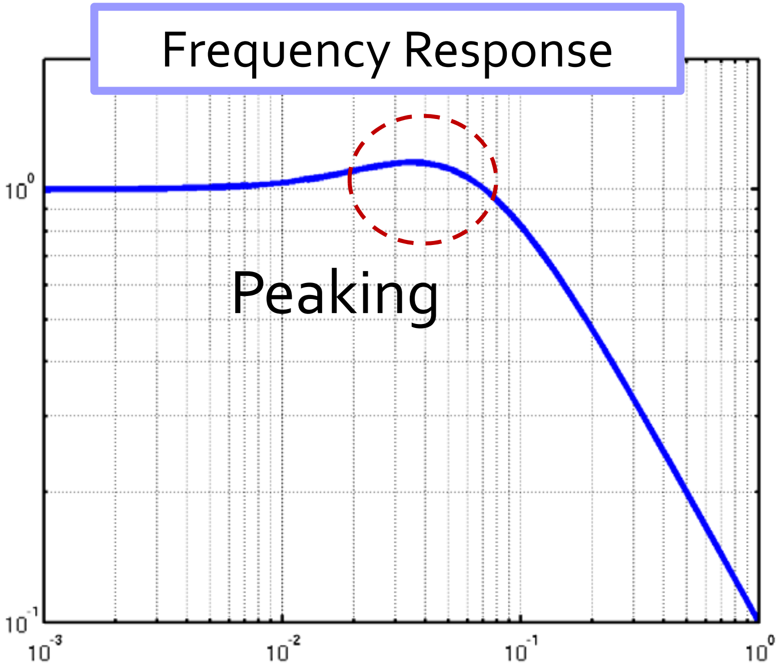
After EQ: No Violations Found

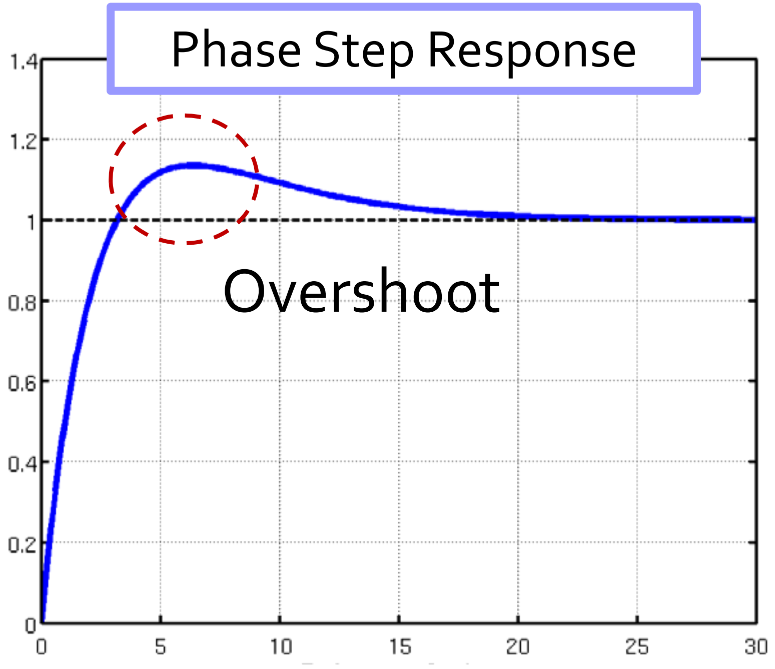


Frequency-domain Property Assertion

- Check the amount of jitter peaking in a phase-locked loop due to the presence of a closed-loop zero

$$H(s) = N\frac{2\zeta s/\omega_n + 1}{s^2/\omega_n^2 + 2\zeta s/\omega_n + 1}$$

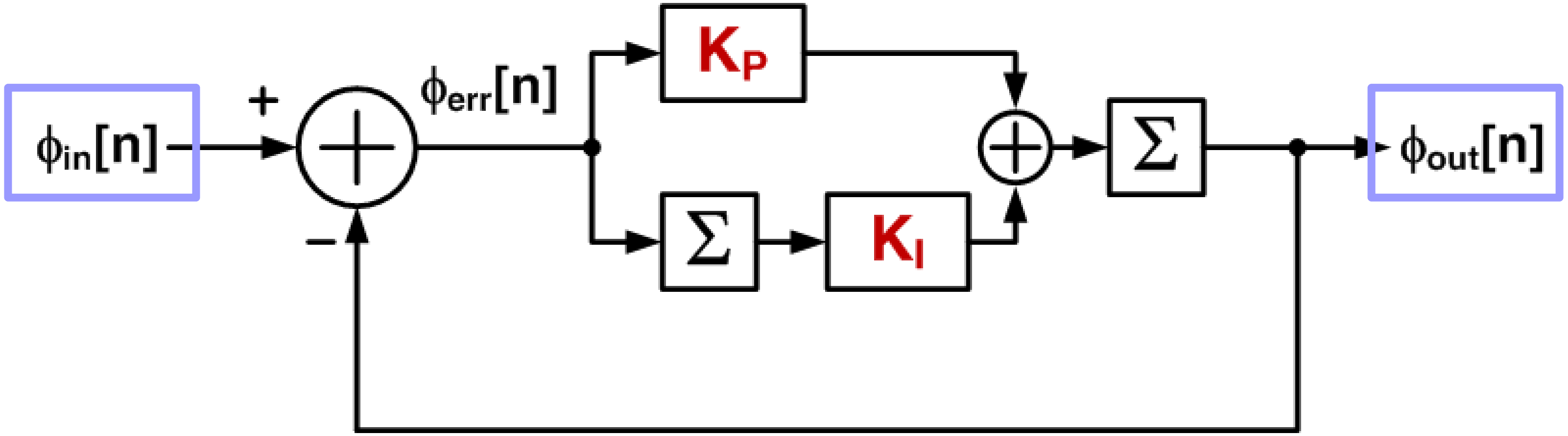




PLL Parameter Estimation

- From the traces of ϕ_{in} and ϕ_{out}, estimate K_P and K_I and verify whether ω_n and ζ meet the spec.
  - What input stimulus is eligible for such estimation?

Unknown Parameters



$$\omega_n = \frac{\sqrt{K_I}}{T_{REF}}$$

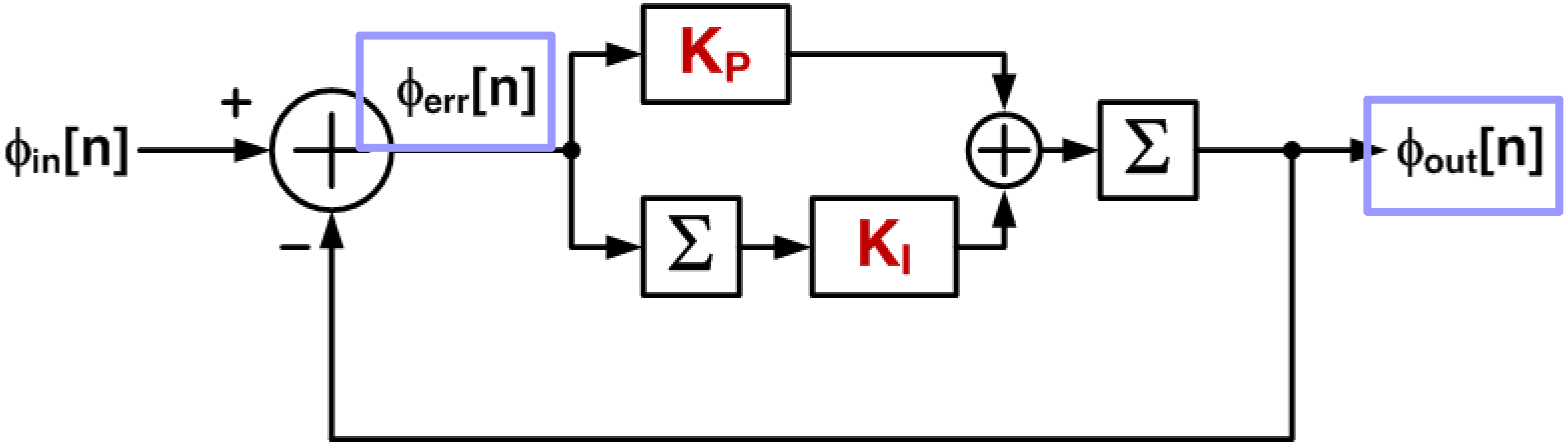
$$\zeta = \frac{K_P}{2\sqrt{K_I}}$$

Open-Loop Approach

- K_P and K_I can be found by solving this open-loop EQ:

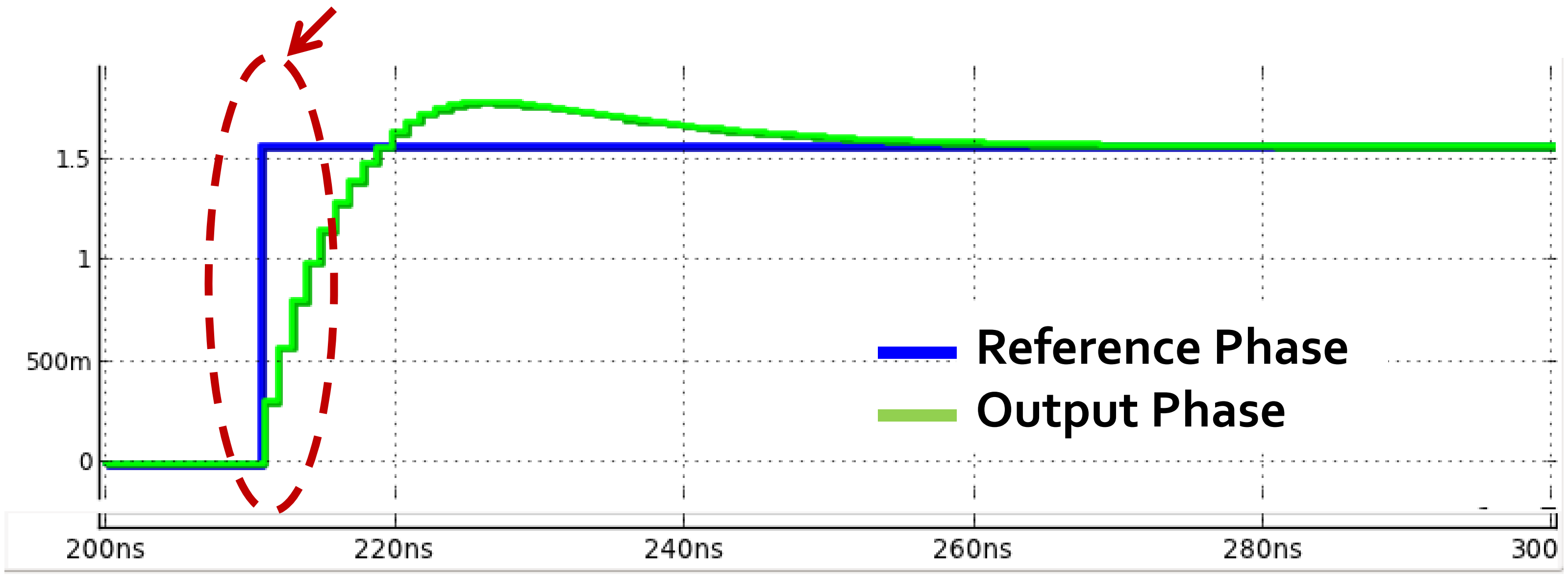
$$\boxed{\begin{bmatrix} \phi_{err}[\text{n}] - \phi_{err}[\text{n-1}] & \phi_{err}[\text{n}] \\ \phi_{err}[\text{n-1}] - \phi_{err}[\text{n-2}] & \phi_{err}[\text{n-1}] \end{bmatrix}} \begin{bmatrix} K_P \\ K_I \end{bmatrix} = \begin{bmatrix} \phi_{out}[\text{n}] - 2\phi_{out}[\text{n-1}] + \phi_{out}[\text{n-2}] \\ \phi_{out}[\text{n-1}] - 2\phi_{out}[\text{n-2}] + \phi_{out}[\text{n-3}] \end{bmatrix}$$

- By examining the condition number of this matrix, we can tell whether the assertion check is covered

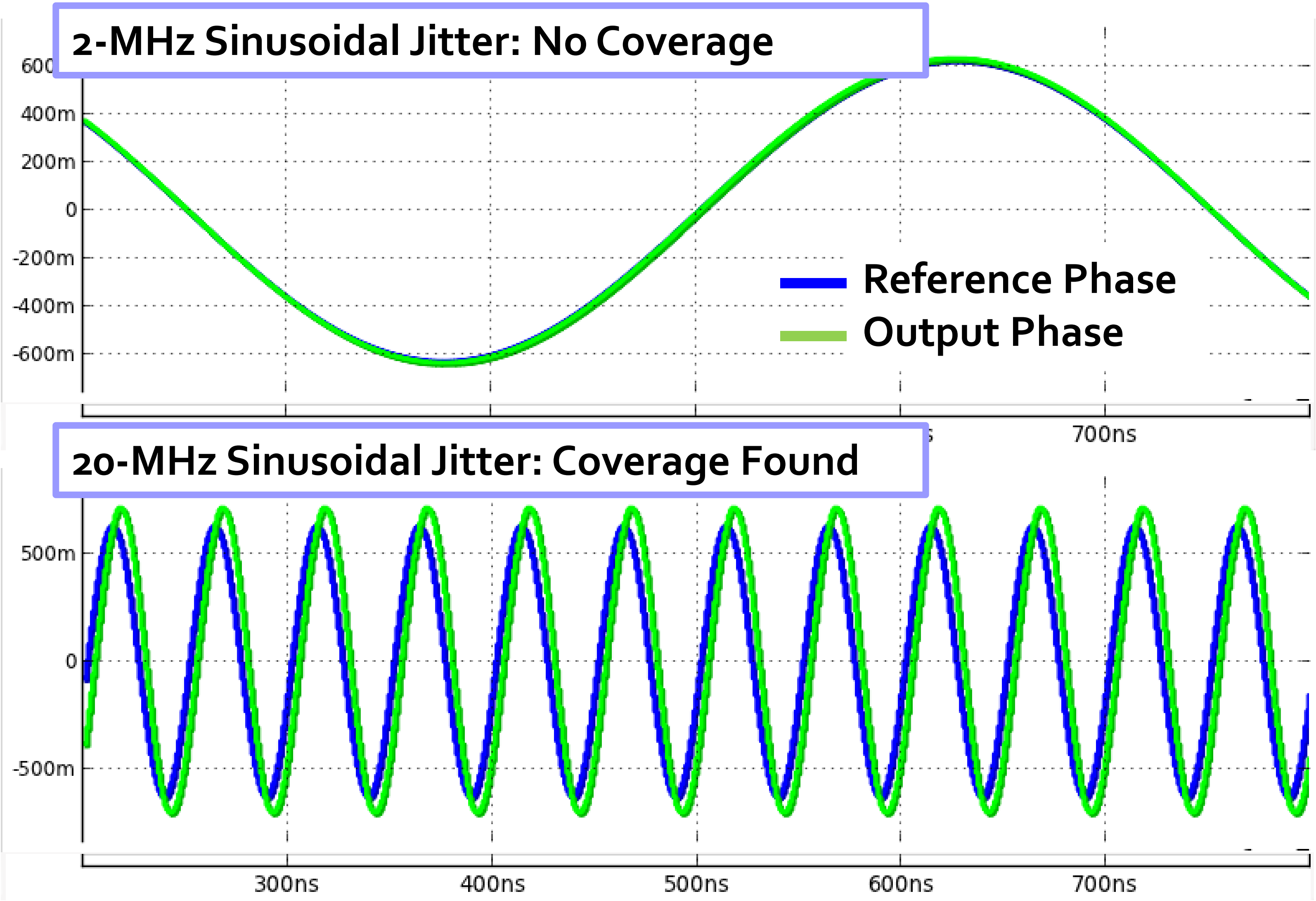


With Step Excitation

Coverage Triggered!



With Sinusoidal Excitations



2-MHz Sinusoidal Jitter: No Coverage

20-MHz Sinusoidal Jitter: Coverage Found

Summary

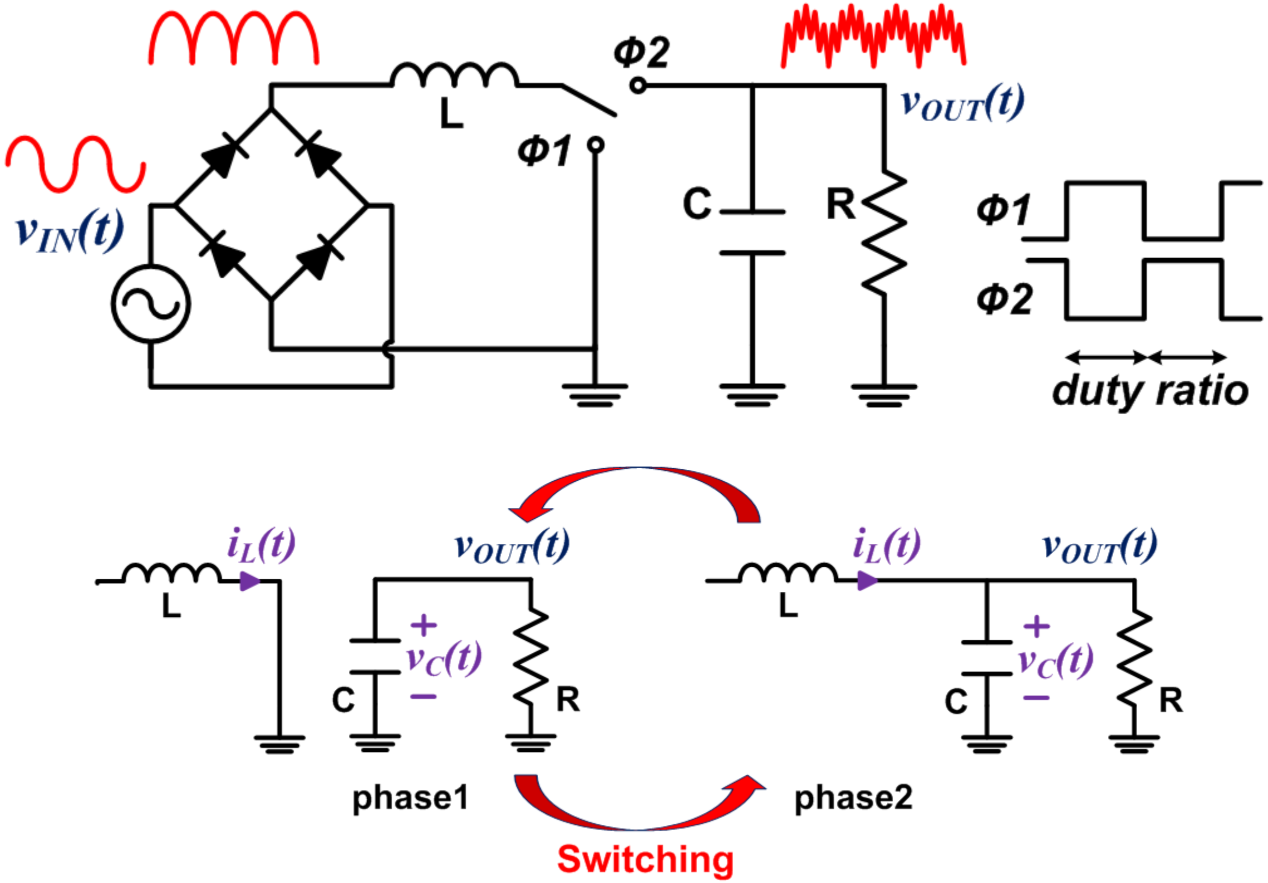
- XMODEL is a truly event-driven behavioral simulator that can model analog and mixed-signal systems in SystemVerilog

- Its compatibility with SystemVerilog allows the use of SystemVerilog Assertions (SVA) to implement time-domain and frequency-domain assertion checks for analog/mixed-signal systems

Backup Slides

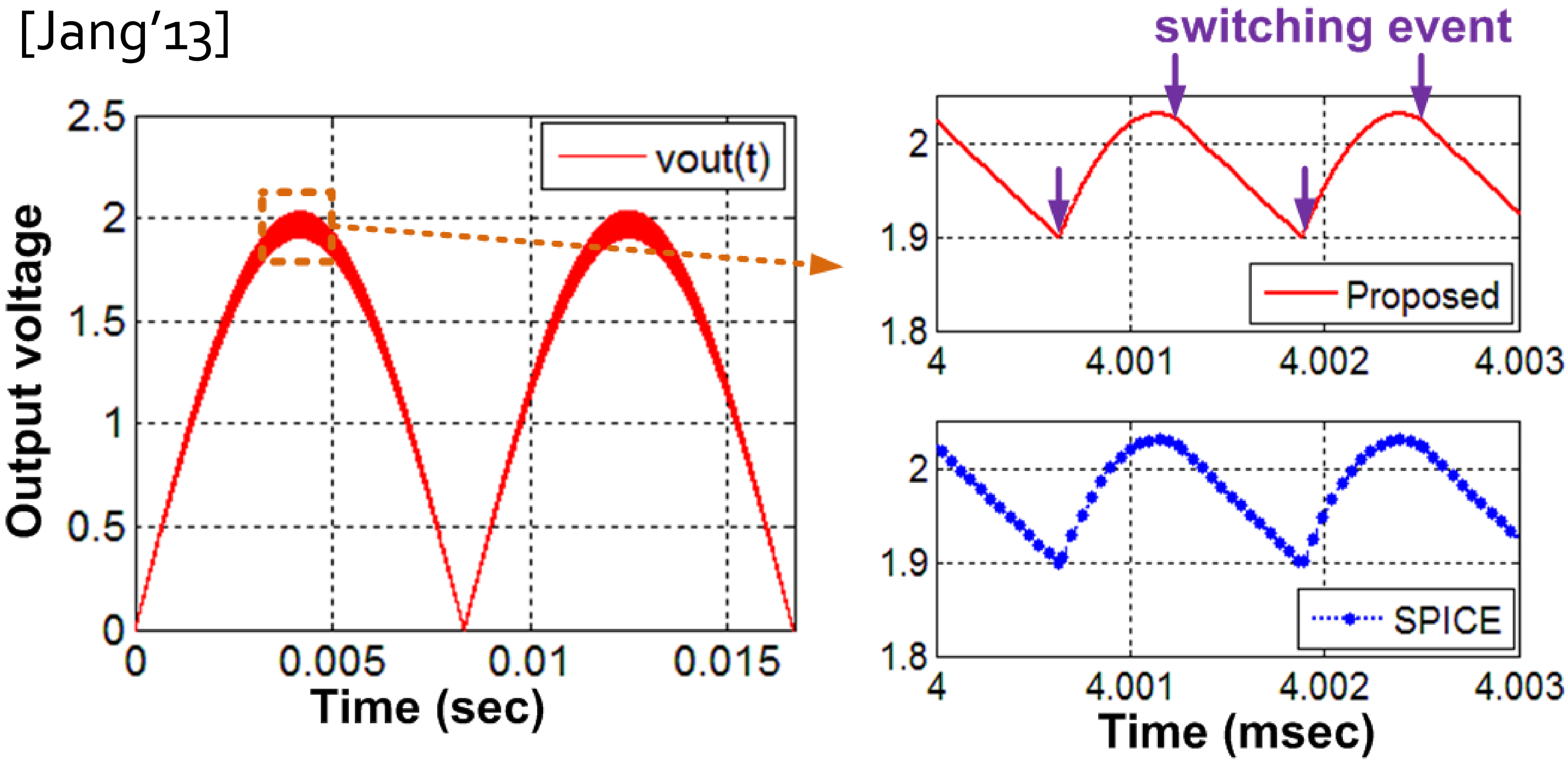
Switched-Linear Circuits

- Many switching supplies can be modeled as a system switching among multiple linear systems



Simulation Results: Boost Converter

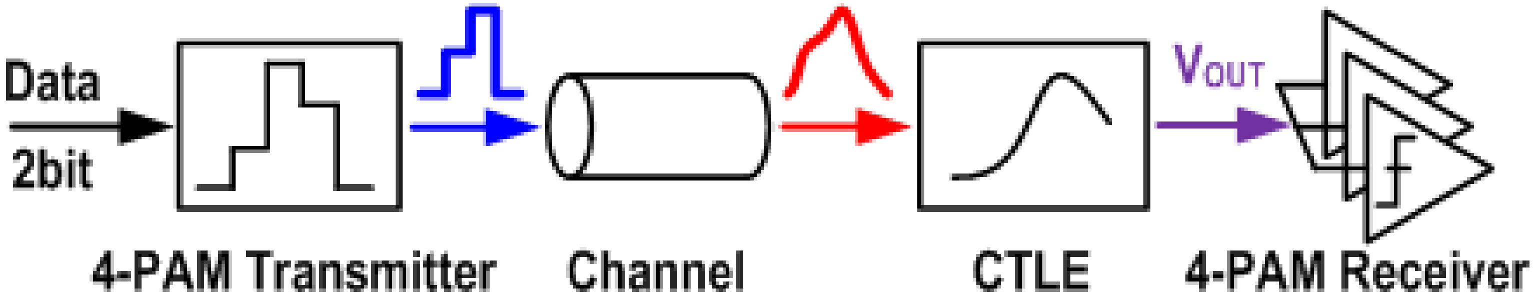
[Jang'13]



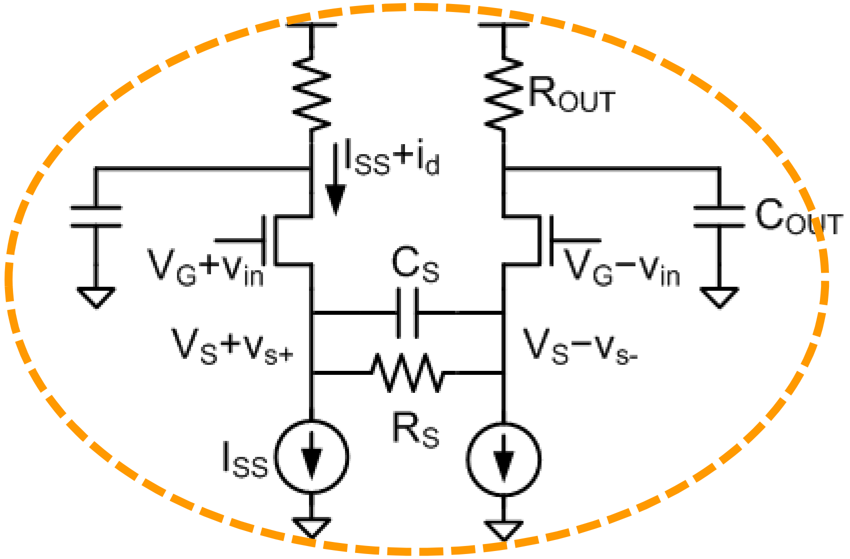
- Runtime is 8.2sec for 0.1sec simulation time (~110X faster than SPICE)

Weakly Nonlinear Circuits

- CTLE of high-speed link receiver may exhibit limiting or saturating behavior



- Its nonlinearity can be modeled as Volterra series



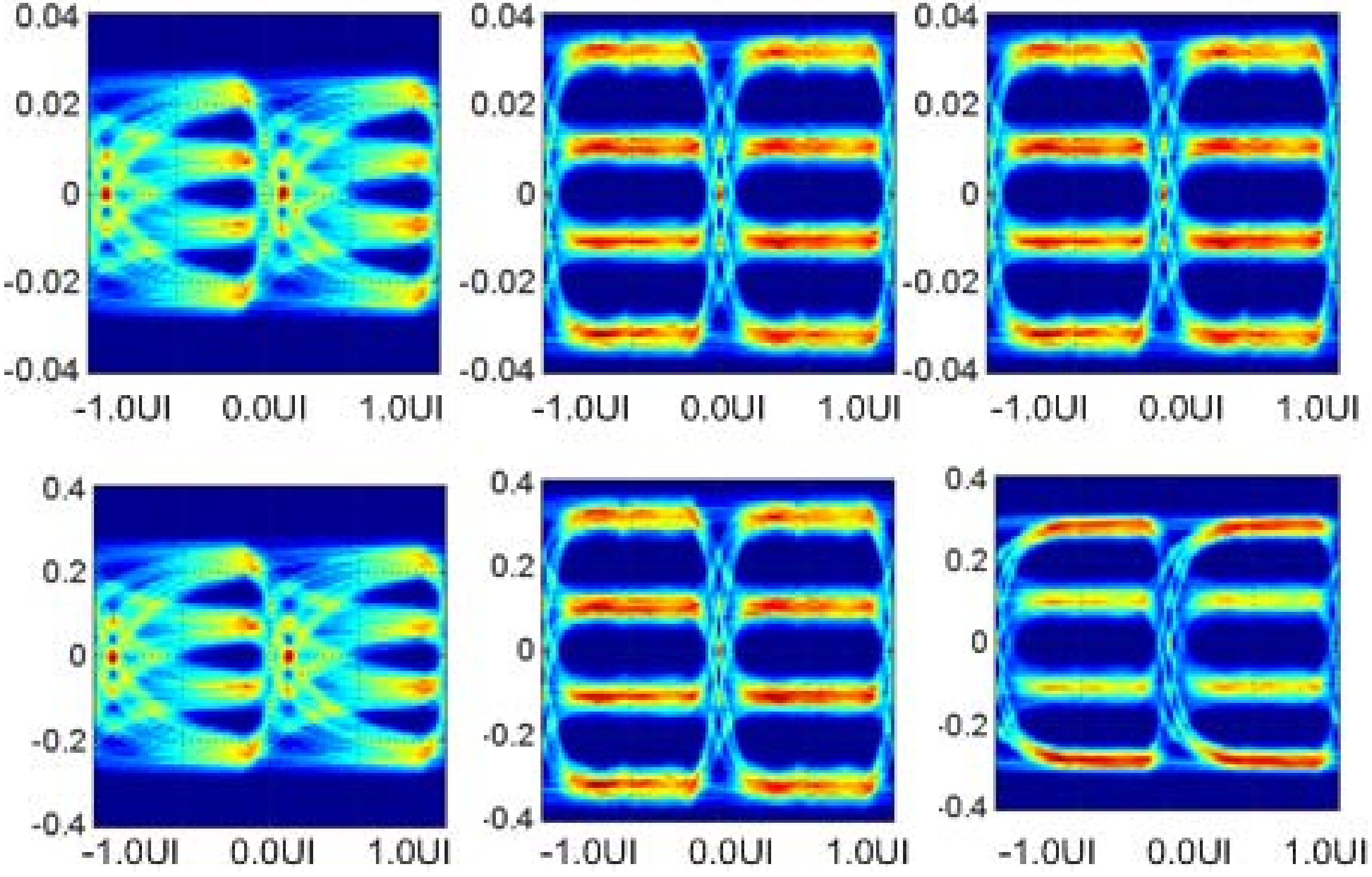
Simulation Results: Eye-Diagrams

[Jang'13]

Channel output | CTLE output (1st) | CTLE output (1st+3rd)

Signal swing: ±30mV$_{dpp}$

Signal swing: ±300mV$_{dpp}$


-1.0UI
0.0UI
1.0UI