



Case Study of Behavioral Modeling for Mixed-Signal Systems

Prof. Chien-Nan Liu
Institute of Electronics
National Chiao-Tung Univ.

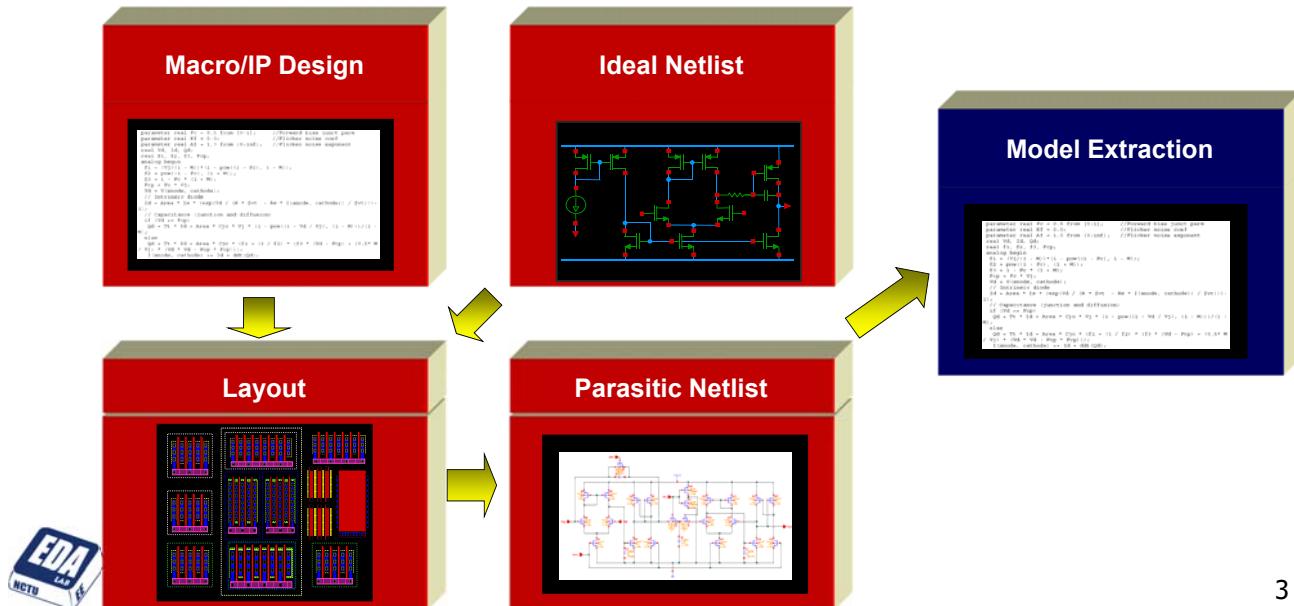
Tel: (03)5712121 ext:31211
E-mail: jimmyliu@nctu.edu.tw
<http://mseda.ee.nctu.edu.tw/jimmyliu>

Outline

- Charge-pump phase lock loop (CPPLL)
- Delta-sigma A/D converter
- Delta-sigma D/A converter
- Transmission link system

Bottom-Up Modeling Approach

- For **existing** blocks, bottom-up **behavior extraction** is more proper to build their behavioral models
 - Non-ideal effects can also be captured



3

Accurate Behavior Extraction

- Bottom-up extraction from simulation results**
 - More accurate
 - Still useful for flattened designs
- Do not separate into sub-blocks**
 - Correctly deal with timing information, loading, parasitics and interactions
- Do not measure from normal operations**
 - Develop a **special characterization mode**
 - Easily send special patterns to trigger the circuit
 - Long extracting time can be avoided



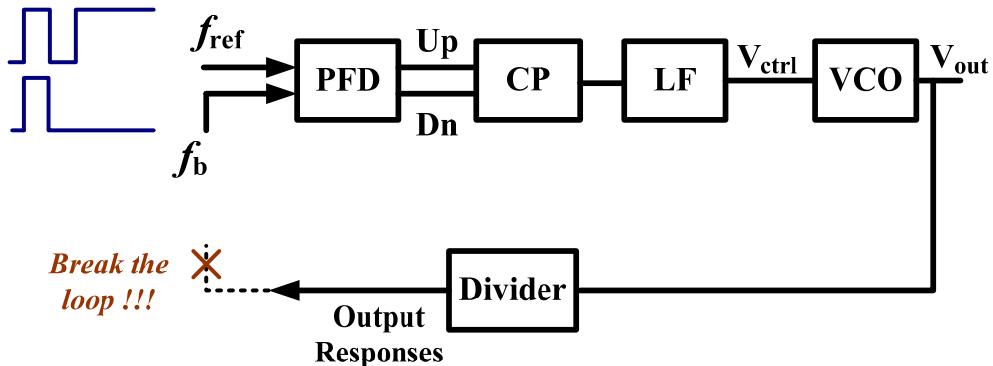
4

Characterization Mode

- Example: Charge Pump Phase-Locked Loop (CPPLL)

(Extraction patterns)

PFD: Phase Frequency Detector, CP: Charge Pump
 LPF: Low-Pass Filter, DIV: frequency DIVider
 VCO: Voltage-Controlled Oscillator



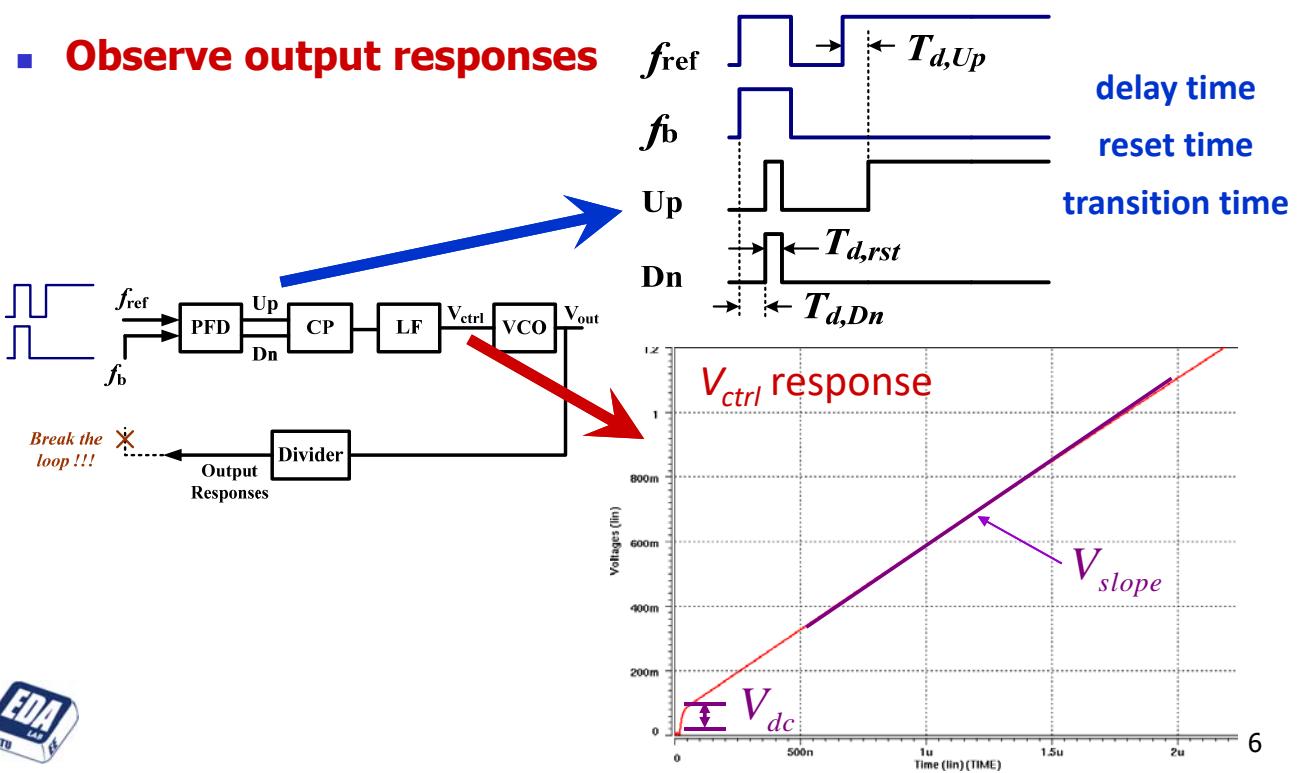
C.C. Kuo, Y.C. Wang, and C.N. Liu, "An Efficient Approach to Build Accurate PLL Behavioral Models of PLL Designs", IEICE Trans. on Fundamentals (SCI), vol. E89-A, no. 2, pp. 391-398, Feb. 2006.

C.C. Kuo, Y.C. Wang, and C.N. Liu, "An Efficient Bottom-Up Extraction Approach to Build Accurate PLL Behavioral Models for SOC Designs", ACM/IEEE GLSVLSI, pp. 286-290, Apr. 2005.

5

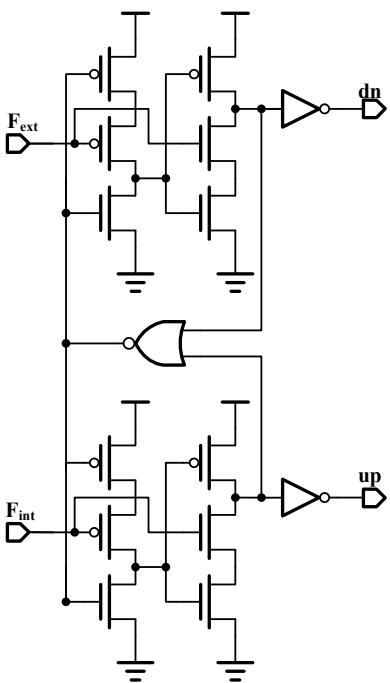
Extract Circuit Properties

- Observe output responses

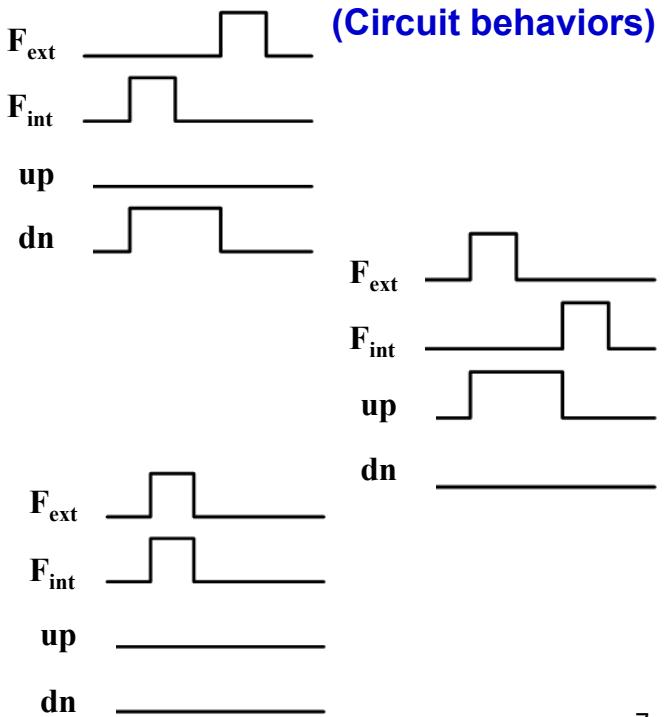


Phase Frequency Detector

(Implementation)



(Circuit behaviors)



7

Verilog-A Code for PFD

```

parameter real vlogic_high = 1.2;
parameter real vlogic_low = 0;

parameter real vtrans = 0.6;

parameter real up_delay = 238p;
parameter real dn_delay = 238p;

parameter real tr_up = 20p;
parameter real tf_up = 25p;
parameter real tr_dn = 23p;
parameter real tf_dn = 27p;

parameter real rst_delay = 93p;
parameter real tr_rst = 10p;
parameter real tf_rst = 12p;

```

```

@ (cross(V(fext) - vtrans, +1))
  tpos_on_A = 1;
@ (cross(V(fint) - vtrans, +1))
  tpos_on_B = 1;
if (tpos_on_A && tpos_on_B)
  reset_val = vlogic_high;
else
  reset_val = vlogic_low;
@ (cross(V(reset) - vtrans, +1))
begin
  tpos_on_A = 0; tpos_on_B = 0;
end
if (tpos_on_A)
  sigout_A_val = vlogic_high;
else
  sigout_A_val = vlogic_low;
if (tpos_on_B)
  sigout_B_val = vlogic_high;
else
  sigout_B_val = vlogic_low;
V(up) <+ transition(sigout_A_val, up_delay, tr_up, tf_up);
V(dn) <+ transition(sigout_B_val, dn_delay, tr_dn, tf_dn);
V(reset) <+ transition(reset_val,  rst_delay, tr_rst, tf_rst);

```

Extracted timing behaviors

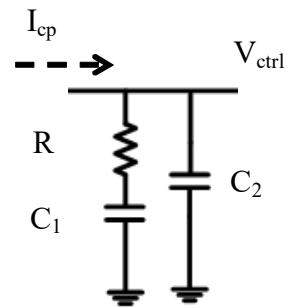


8

Ideal Loop Filter Modeling

$$Z_{LF} = \frac{\left(R + \frac{1}{sC_1}\right) \times \frac{1}{sC_2}}{\left(R + \frac{1}{sC_1}\right) + \frac{1}{sC_2}} = \frac{1 + sRC_1}{s(C_1 + C_2) + s^2 RC_1 C_2} = \frac{\frac{1}{C_1} + sR}{s\left(\frac{1+\alpha}{\alpha} + \frac{sRC_1}{\alpha}\right)}$$

where $\alpha = \frac{C_1}{C_2}$



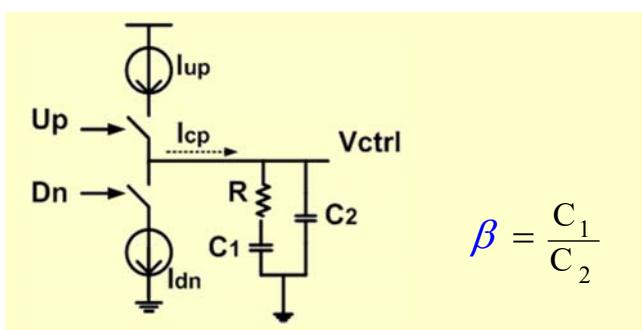
$$\text{laplace_nd}(s) \frac{\sum_k n_k s^k}{\sum_k d_k s^k}$$

$V(V_{ctrl}) <+ \text{laplace_nd}(I(I_{cp}), \{1/C_1, R\}, \{0, (1+\alpha)/\alpha, RC_1/\alpha\});$



9

Model CP+LF Together



$$\beta = \frac{C_1}{C_2}$$

Key:

Translating the current form to the voltage form.

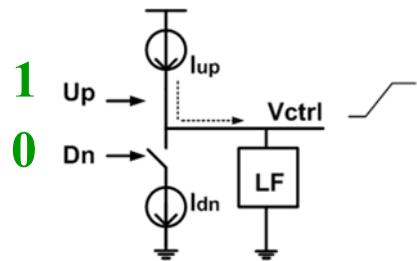
$$V_{ctrl} = I_{CP} Z_{LF} = \frac{s I_{CP} R + \frac{I_{CP}}{C_1}}{\frac{s^2 R C_1}{\beta} + s \left(\frac{1 + \beta}{\beta} \right)}$$

$$\approx \frac{s I_{CP} R + \frac{I_{CP}}{C_1}}{s} = I_{CP} R + \frac{I_{CP}}{s C_1} = \textcolor{red}{a} + \frac{\textcolor{red}{b}}{s}$$



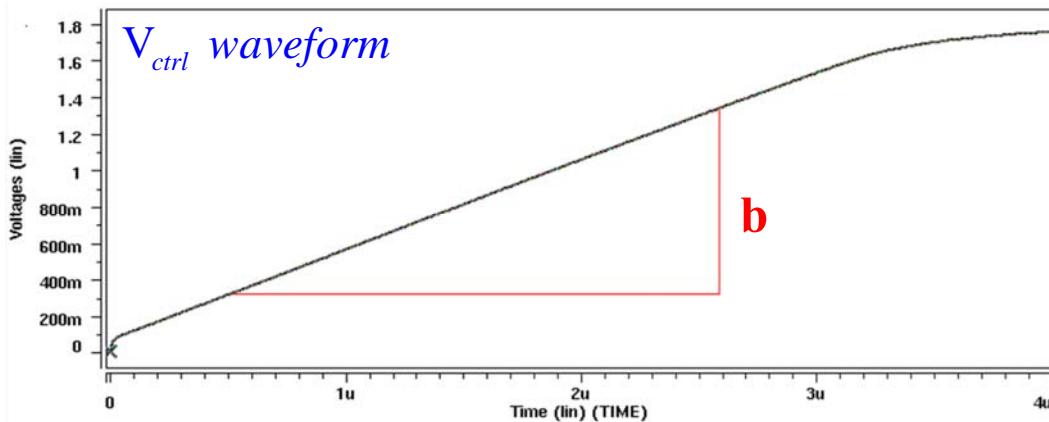
10

CP and LF Response (1/2)



Extract the information of I_{CP} and Z_{LF}

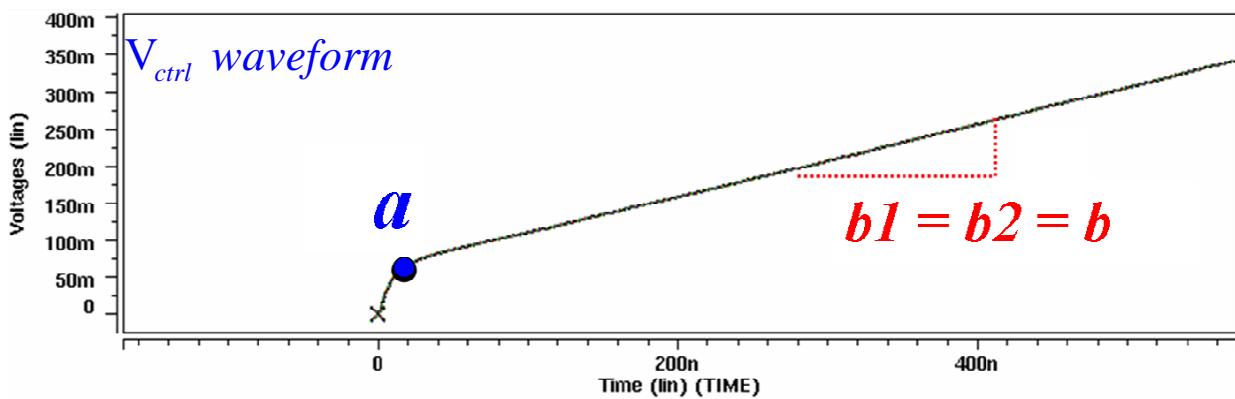
$$b = \frac{I_{CP}}{C_1} \Rightarrow b = \frac{\Delta I}{C_1} = \frac{\Delta V_{ctrl}}{\Delta T}$$



11

CP and LF Response (2/2)

$$a = I_{CP} R \Rightarrow a = \Delta I \cdot R = \Delta V_{ctrl}$$



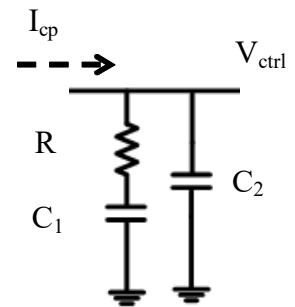
Extract the information of I_{CP} and Z_{LF}

12

CP+LF Modeling

$$Z_{LF} = \frac{\left(R + \frac{1}{sC_1}\right) \times \frac{1}{sC_2}}{\left(R + \frac{1}{sC_1}\right) + \frac{1}{sC_2}} = \frac{1 + sRC_1}{s(C_1 + C_2) + s^2 RC_1 C_2} = \frac{\frac{1}{C_1} + sR}{s\left(\frac{1+\alpha}{\alpha} + \frac{sRC_1}{\alpha}\right)}$$

$$= \frac{(ab) + s \cdot (\alpha a)}{s \left\{ (1+\alpha) + s \cdot \frac{a}{b} \right\}}$$



where $\alpha = \frac{C_1}{C_2}$, $a = I_{CP} \cdot R$, $b = \frac{I_{CP}}{C_1}$

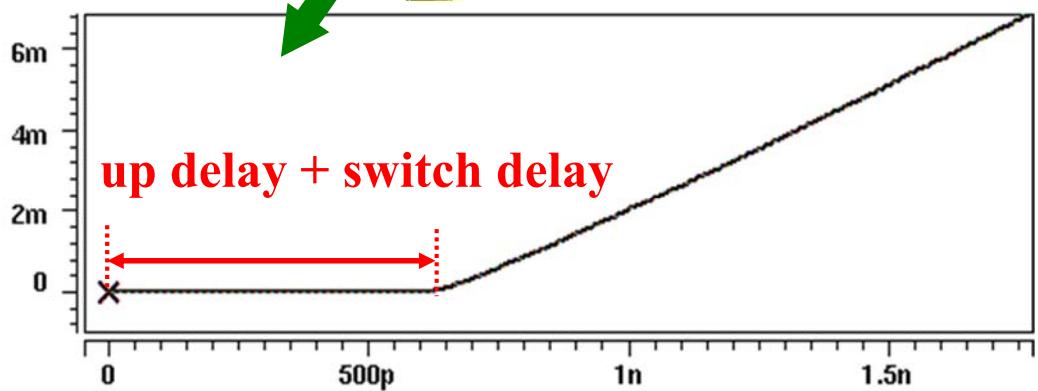
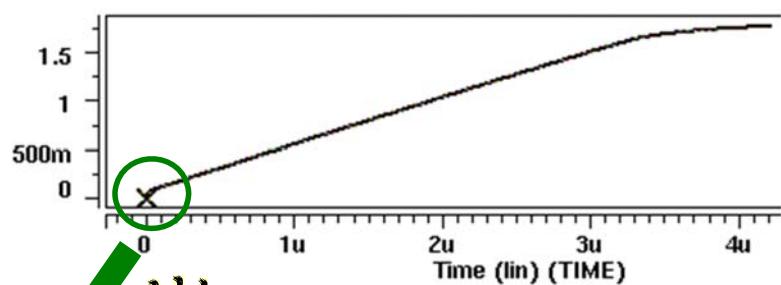
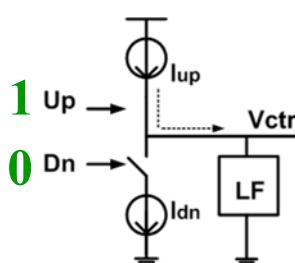
$\text{laplace_nd}(s) \frac{\sum_k n_k s^k}{\sum_k d_k s^k}$

$V(V_{ctrl}) <+ \text{laplace_nd}(I(I_{cp}), \{\alpha \cdot b, \alpha \cdot a\}, \{0, 1+\alpha, a/b\});$



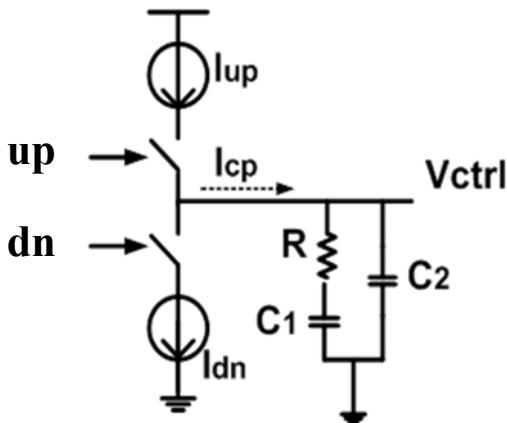
13

Switch Delay



14

Charge Pump Modeling



```

inc_high = up > vtrans;
dec_high = dn > vtrans;
i_mult = 0.0;

if (inc_high == dec_high) begin
    i_mult = 0.0;
end else if (inc_high) begin
    i_mult = 1.0;
end else if (dec_high) begin
    i_mult = -1.0;

icp = iamp*i_mult;

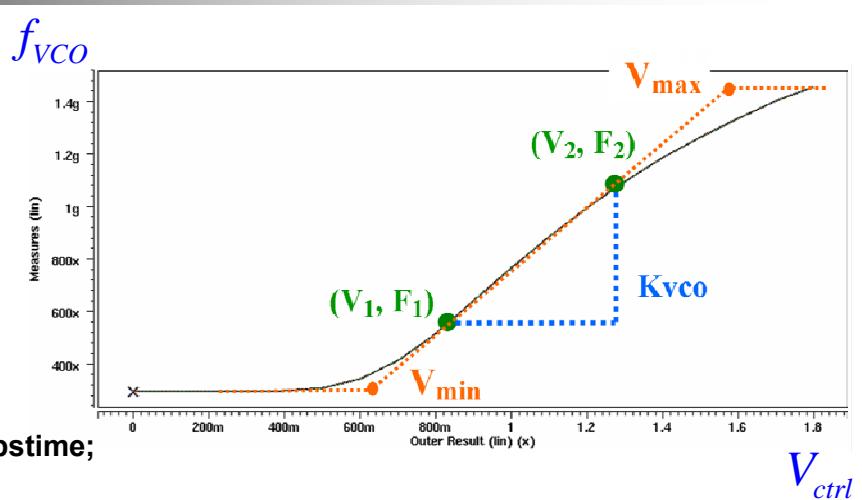
I(gnd, vctrl) <+ transition(icp, tdel, tr, tf);

```

Extract the output current responses

15

VCO Modeling



```

`define PI 3.14159
phase_lin = (2 * `PI * fmin) * $abstime;
if (V(vctrl) > vmin )
    dv = V(vctrl) - vmin;
else
    dv =0;
phase_nonlin = 2 * `PI * kvco * idt(dv, 0);
V(vout) <+ amp * cos(phase_lin + phase_nonlin) + vdc;

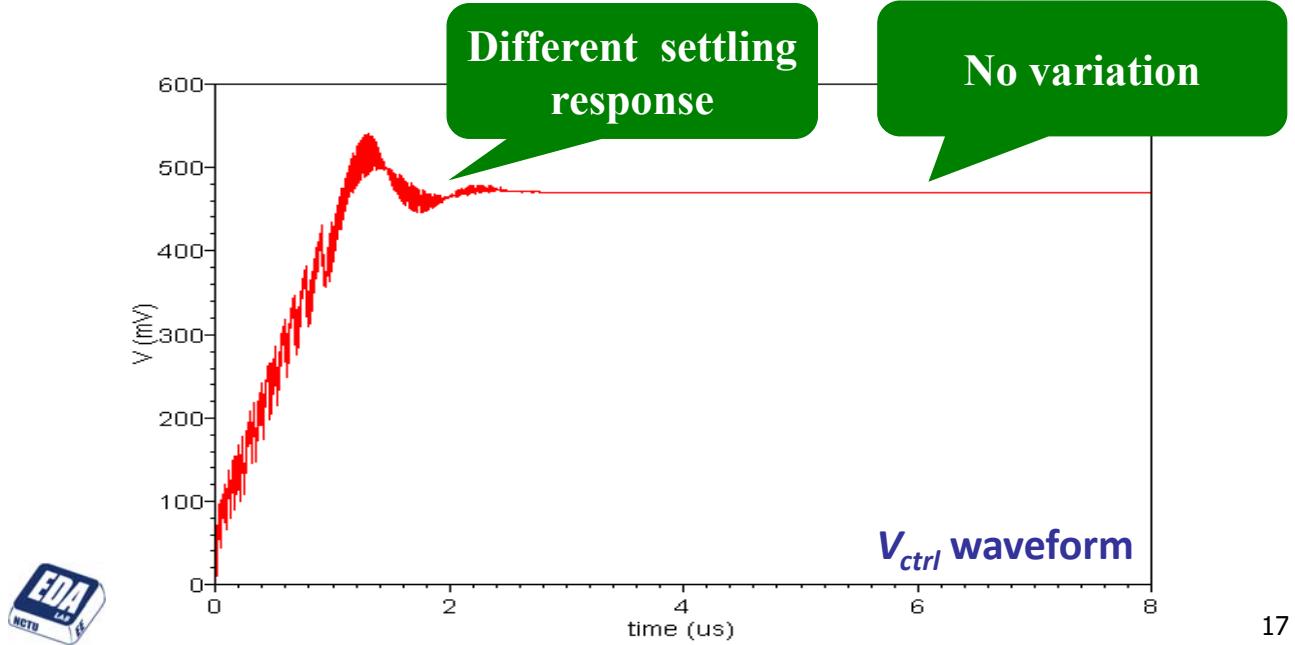
```

Extract the information of
 V_{ctrl} - f_{out} transfer function

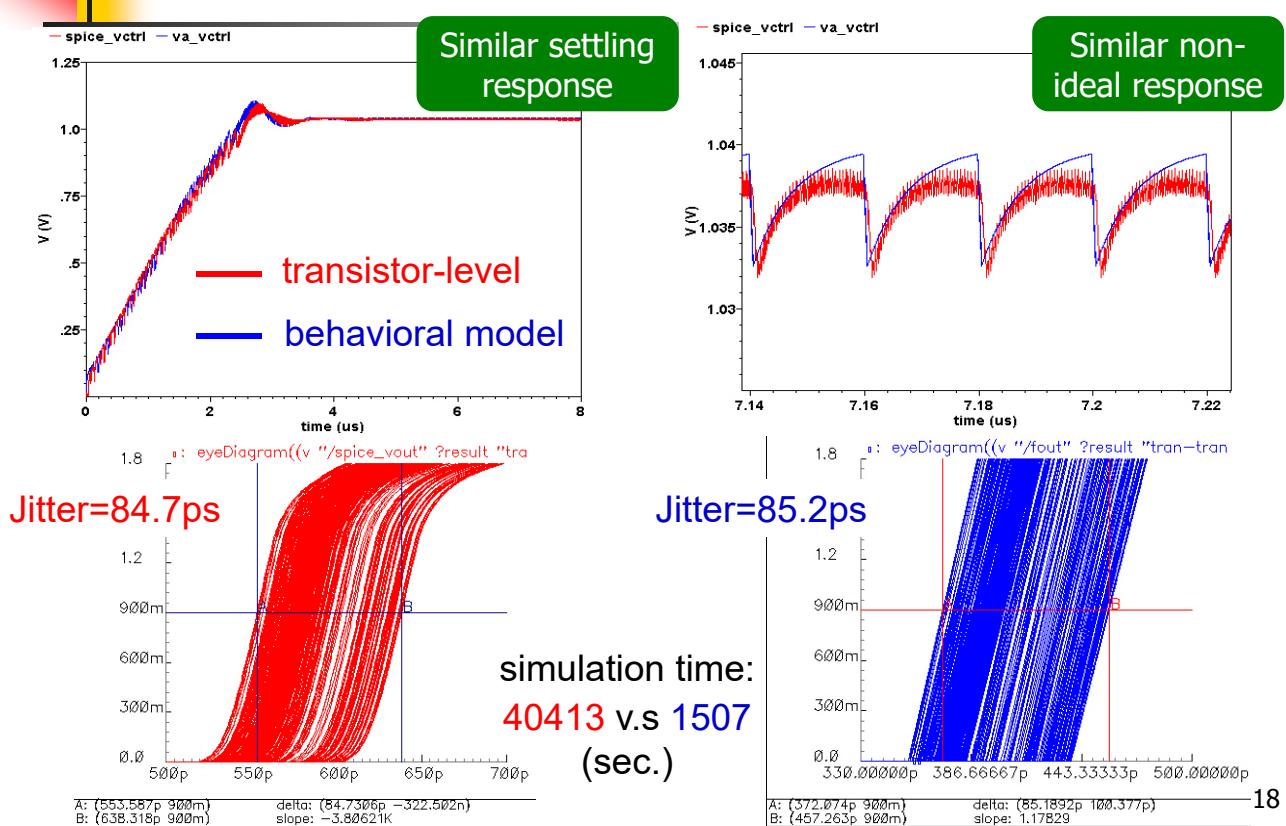
16

Ideal Behavioral Model

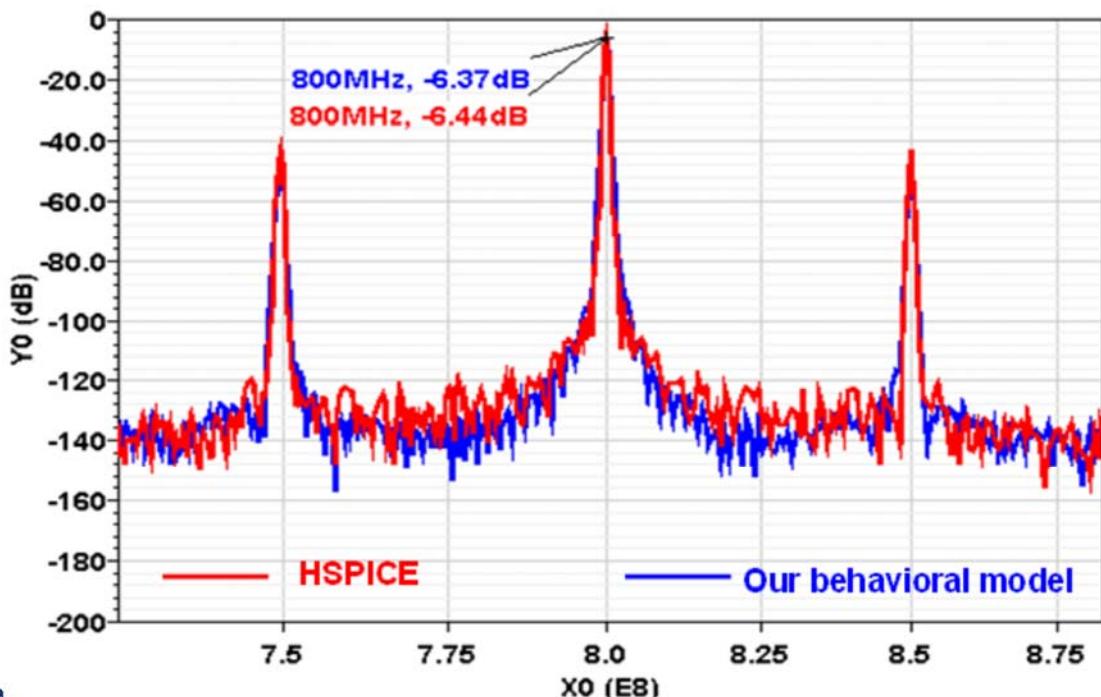
- Use the embedded behavioral blocks from **Cadence's AHDL library**



Extracted Behavioral Model



Similar Frequency Responses



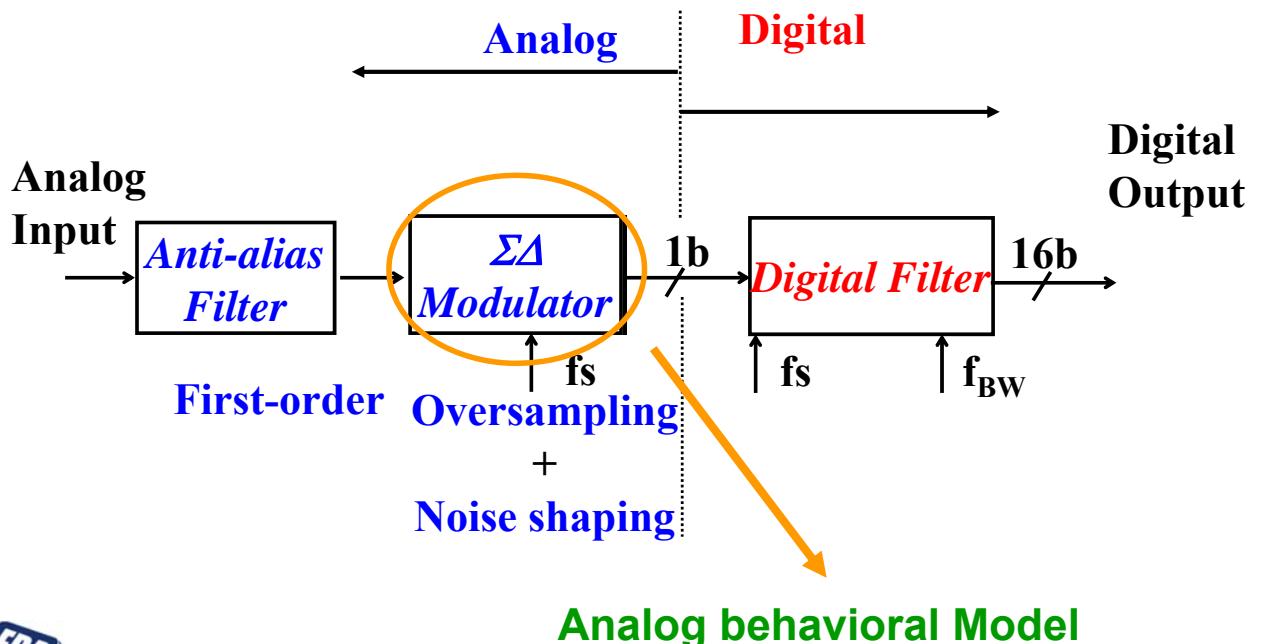
19

Outline

- Charge-pump phase lock loop (CPPLL)
- Delta-sigma A/D converter
- Delta-sigma D/A converter
- Transmission link system

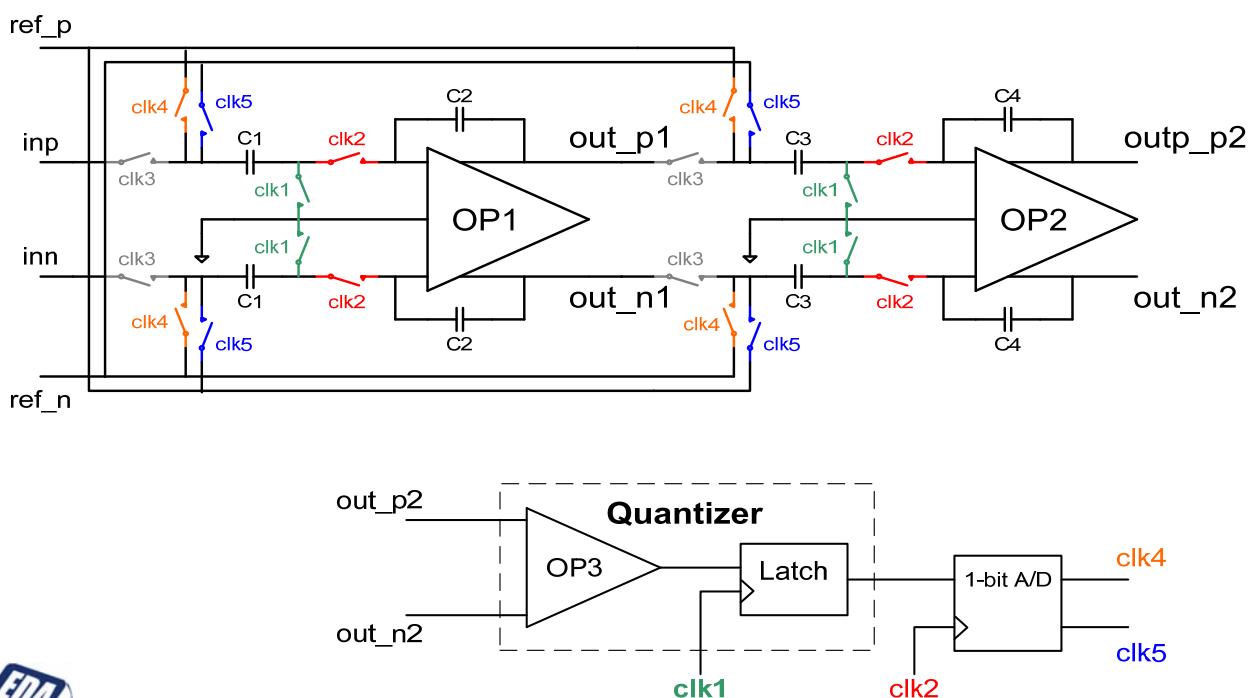
20

Sigma-Delta A/D Converter



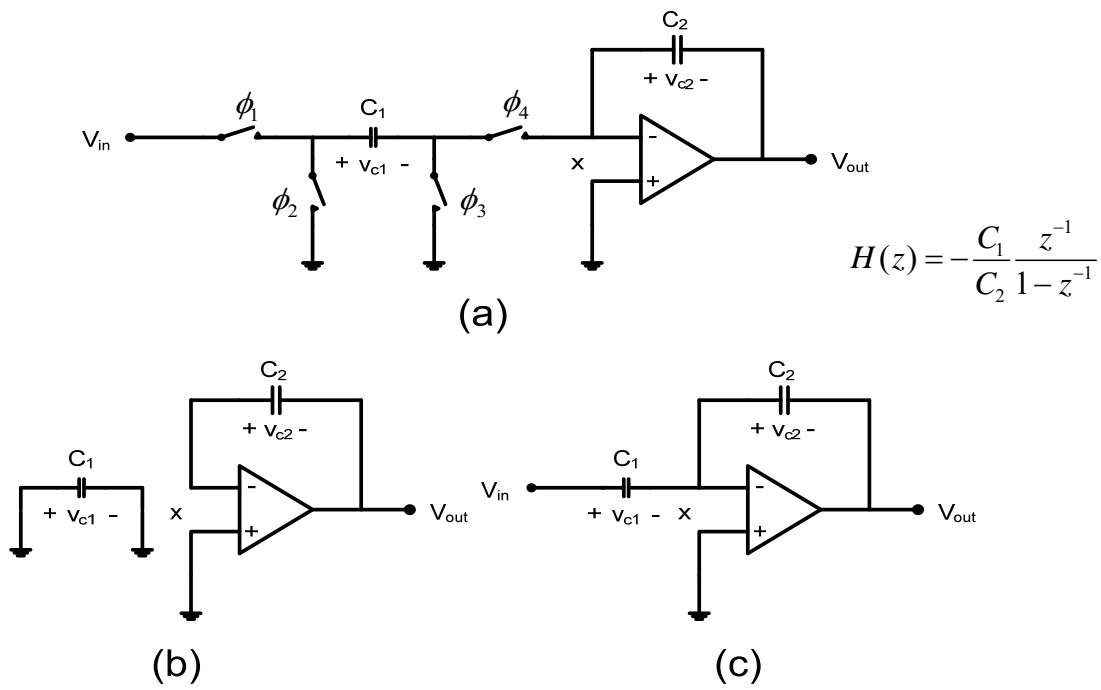
21

2nd-Order Sigma-Delta Modulator



22

SC Integrator



(a) Circuit (b) Sample mode (c) Integration mode

23

SC Integrator in Verilog-A (1/2)

```
parameter real vth=1.65,vcm=1.65,vrp=2.2,vrn=1.1;
// vth: transition threshold voltage
// vrp, vrn: reference voltage
// vcm = VDD/2 in this case
```

```
parameter real T=80n,Tf=0,Td=43n; // for z-transform
```

```
real a1=0.4; // DC gain: a1=C1/C2
```

```
parameter real voffset=0.27342; // DC-level offset
```

//Sampling mode

```
if(V(ck1)>vth && V(ck3)>vth)
```

```
    vi_tmp=V(vi);
```

```
else
```

```
    vi_tmp=vi_tmp;
```



24

SC Integrator in Verilog-A (2/2)

//Integration mode

```

if(V(ck2)>vth && V(ck4)>vth && V(ck5)<vth)
    vi_in=vi_tmp - vrp;

else if(V(ck2)>vth && V(ck4)<vth && V(ck5)>vth)
    vi_in=vi_tmp - vrn;

else
    vi_in=vi_in;

```

//z-domain transfer function of SC integrator

```

scout_tmp = zi_nd(vip_in,[0, a1],[1,-1],T,Tf,Td) + (vcm-voffset);
V(scout) <+ transition(scout_tmp, 0, tr, tf);

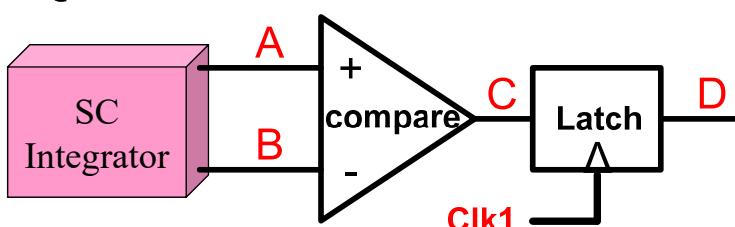
```



25

Quantizer and 1-bit ADC

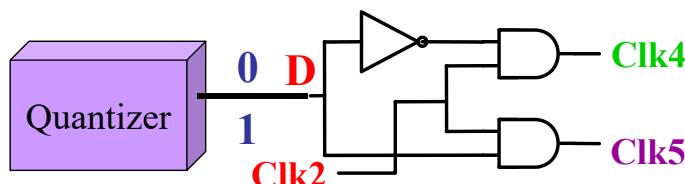
■ Quantizer



If A > B, C = Vdd,
else C = 0

If clk1 on, D = C,
else D = D

■ 1-bit ADC



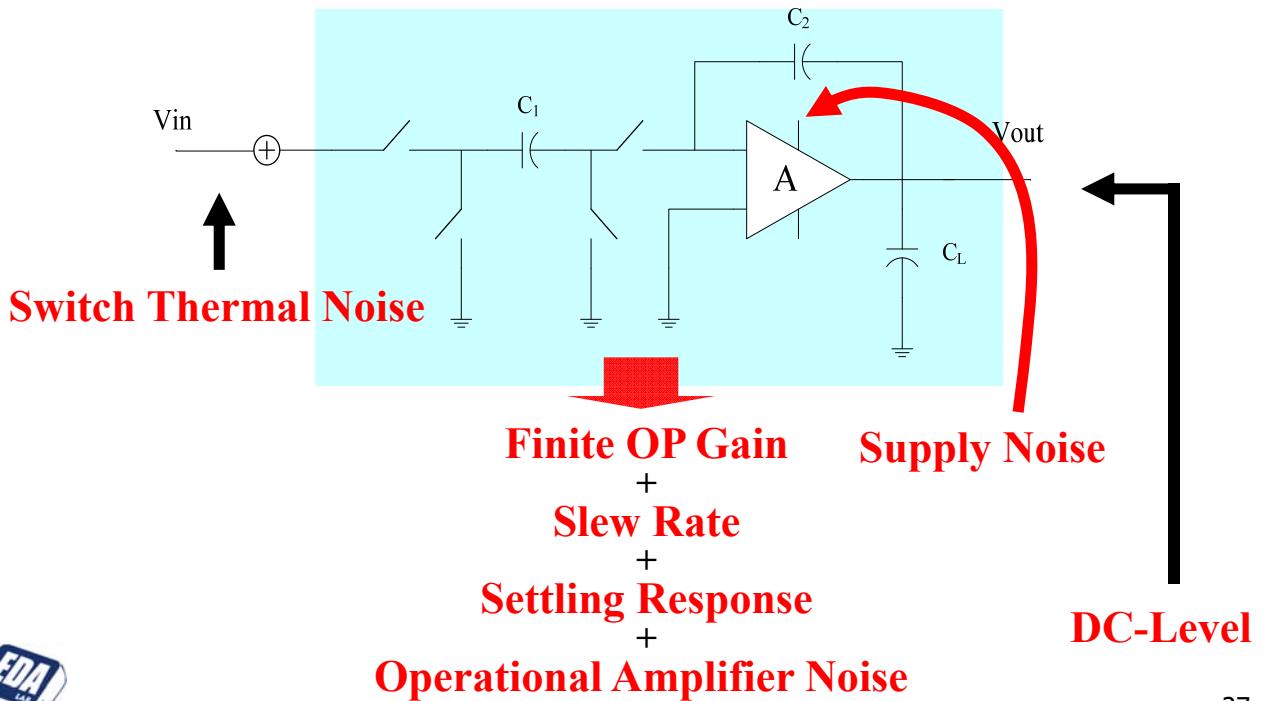
If D = 0, clk2 on,
clk4 = 1

If D = 1, clk2 on,
clk5 = 1



26

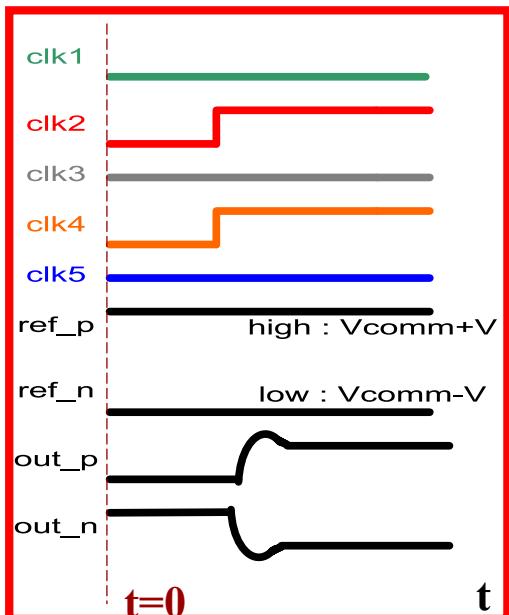
Non-ideal Effects



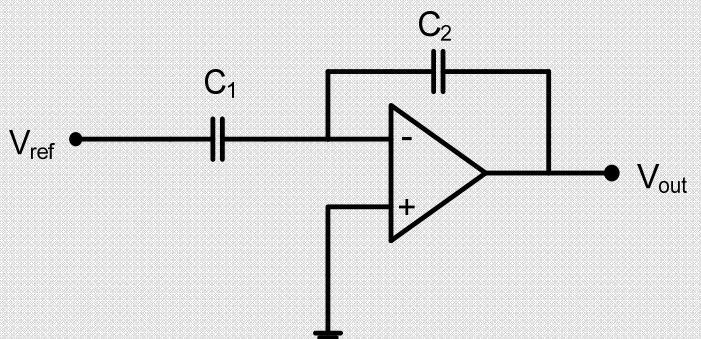
27

Characterization Mode

- Break the feedback path by controlling the clocks



➤ Operated as an inverting-integrator



$$V_{comm} = \frac{V_{DD}}{2} \quad V: \text{given voltage difference}$$

W.H. Cheng, C.C. Kuo, P.J. Chen, Y.M. Wang, and C.N. Liu,
"An Efficient Bottom-Up Extraction Approach to Build the
Behavioral Model of Switched-Capacitor Sigma-Delta
Modulator", IEEE BMAS, pp. 17-21, Sep. 2007.

28

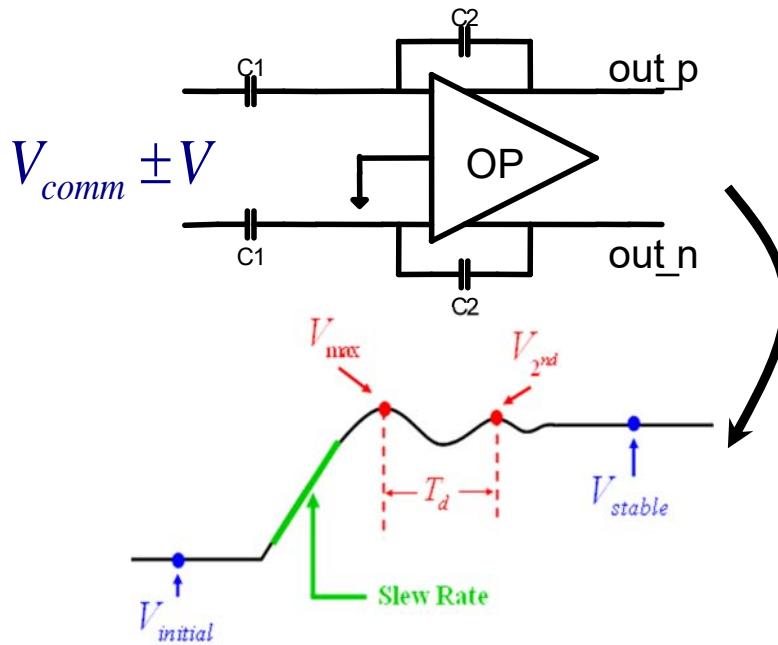
Extraction from Simulation

- Operate the modulator as the **SC integrator**

$$(V_{comm} = V_{DD}/2)$$

3 Patterns

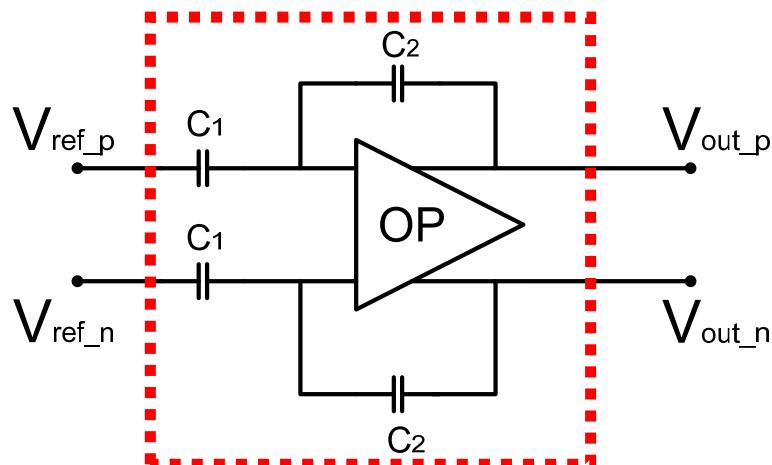
1. $V = V_1$
DC Gain
2. $V = V_2$
Slew Rate
3. $V = V_{in,max}$
Max Slew Rate
Settling Response



29

DC Gain

- Measure the input/output voltages to get real value of OP DC gain

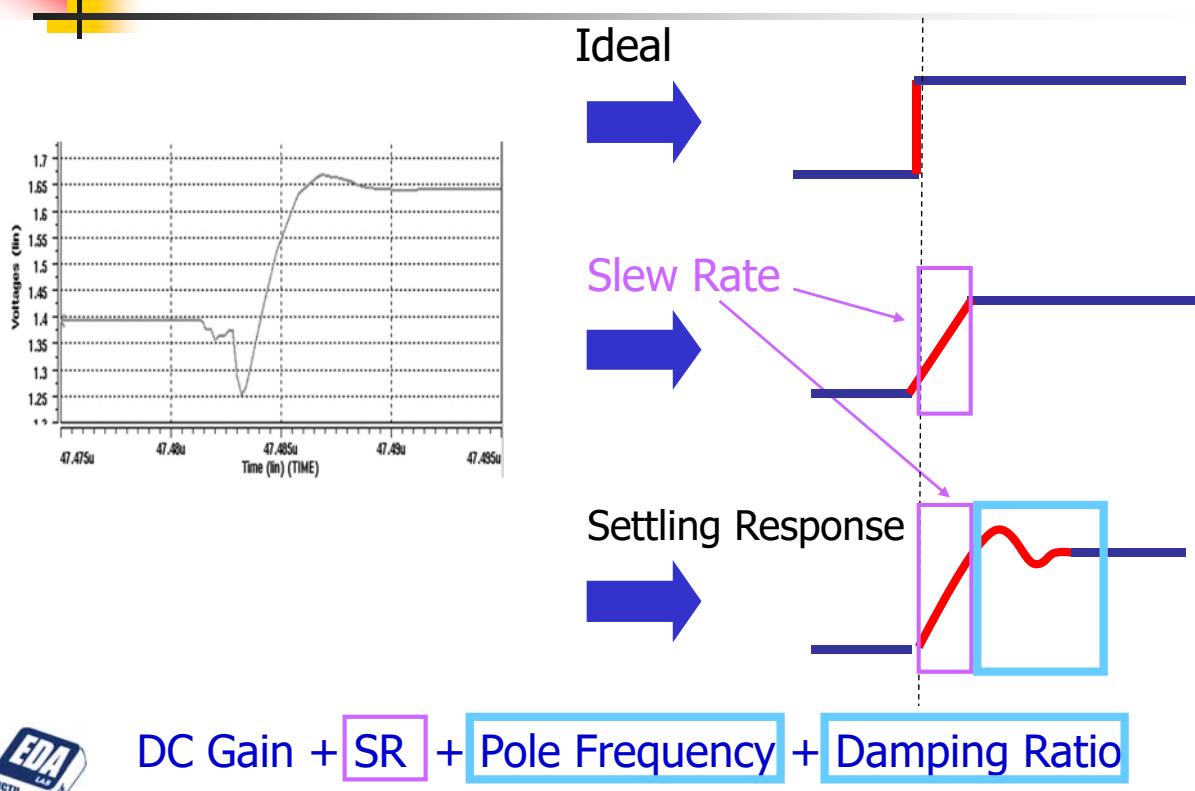


$$V_{out_n} - V_{out_p} = \boxed{DC\ Gain} \times (V_{ref_p} - V_{ref_n})$$



30

Settling Response



31

Output Slew Rate

$$SR = \frac{I_{out}}{C_{out}} = \frac{I_{Vdd} - I_{int}}{C_{out}}$$

$$V_{ref1} \Rightarrow SR_1 = \frac{I - \frac{1}{2}\beta(V_{ref1} - V_{tp} - V_m)^2}{C}$$

$$V_{ref2} \Rightarrow SR_2 = \frac{I - \frac{1}{2}\beta(V_{ref2} - V_{tp} - V_m)^2}{C}$$

$$V_{ref} \Rightarrow SR = \frac{I - \frac{1}{2}\beta(V_{ref} - V_{tp} - V_m)^2}{C}$$

$$\Leftrightarrow \frac{SR_1 - SR_2}{SR_2 - SR} \approx \frac{V_{ref1} - V_{ref2}}{V_{ref2} - V_{ref}} \Rightarrow SR \approx SR_2 - \frac{V_{ref2} - V_{ref}}{V_{ref1} - V_{ref2}} \times (SR_1 - SR_2)$$



$(V_{ref1}, V_{ref2} : \text{Extraction Pattern}; SR_1, SR_2 : \text{Extraction Value})$

32

- Measure the output slew rate of two different inputs to find the slew rate equation
- Should add an upper bound on the output slew rate

Output Ripple Modeling

- Use the standard 2-order transform function to model ripple response

$$T(s) = \frac{\omega_0^2}{s^2 + 2\xi\omega_0 s + \omega_0^2}$$

$$\text{laplace_nd}(s) \frac{\sum_k n_k \cdot s^k}{\sum_k d_k \cdot s^k}$$

$H(s) <+ \text{laplace_nd}(\text{, } \{\omega_0^2\}, \{\omega_0^2, 2\cdot\xi\cdot\omega_0, 1\})$;



33

Extract Damping Response

➤ Damping Ratio - ξ

$$OS \% = 100 \times e^{\frac{-\pi}{\sqrt{4Q^2-1}}}$$

$$= \left(\frac{V_{\max} - V_{stable}}{V_{stable} - V_{initial}} \right)$$

$$\Rightarrow \boxed{\xi = \frac{1}{2 \cdot \sqrt{\left(\frac{-\pi}{\ln(OS)} \right)^2 + 1}}}$$

$OS \%$: OverShoot Percent

Q (pole quality factor)

$$\Rightarrow Q = \frac{1}{2\xi}$$

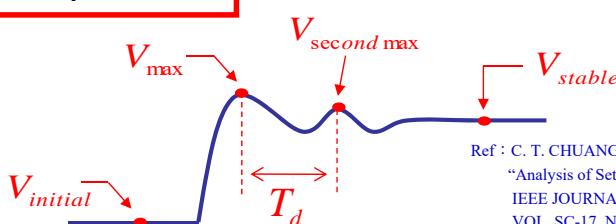
➤ Pole Frequency - ω_0

$$T_d = \frac{2\pi}{\omega_d} = \frac{2\pi}{\omega_0 \sqrt{1-\xi^2}}$$

$$\Rightarrow \boxed{\omega_0 = \frac{2\pi}{T_d \sqrt{1-\xi^2}}}$$

T_d : period of oscillation

ω_d : damped frequency



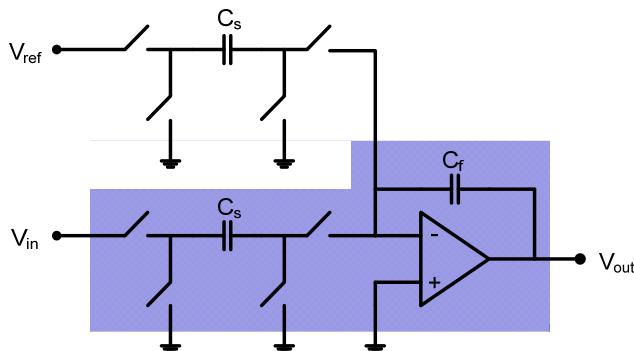
Ref : C. T. CHUANG, STUDENT MEMBER, IEEE
 "Analysis of Settling Behavior of an Operational Amplifier"
 IEEE JOURNAL OF SOLID-STATE CIRCUITS,
 VOL. SC-17, NO. 1, FEBRUARY 1982

34



Thermal Noise

➤ Switch Thermal Noise



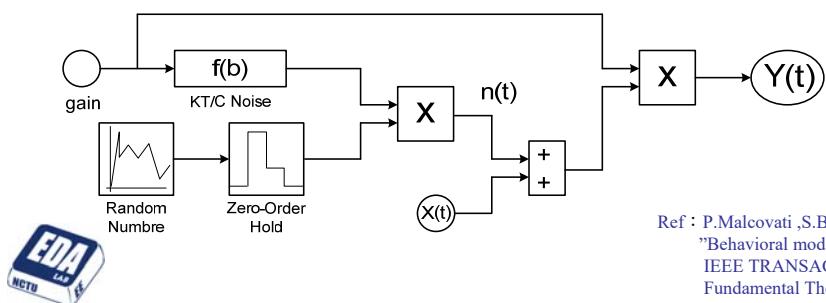
$$e_T^2 = \int_0^\infty \frac{4kTR_{on}}{1 + (2\pi f R_{on} C_s)^2} df = \frac{kT}{C_s}$$

$$y(t) = [x(t) + e_T(t)] \times \text{gain}$$

Switch Thermal Noise

$$y(t) = \left[x(t) + \sqrt{\frac{kT}{C_s} \cdot n(t)} \right] \times \text{gain}$$

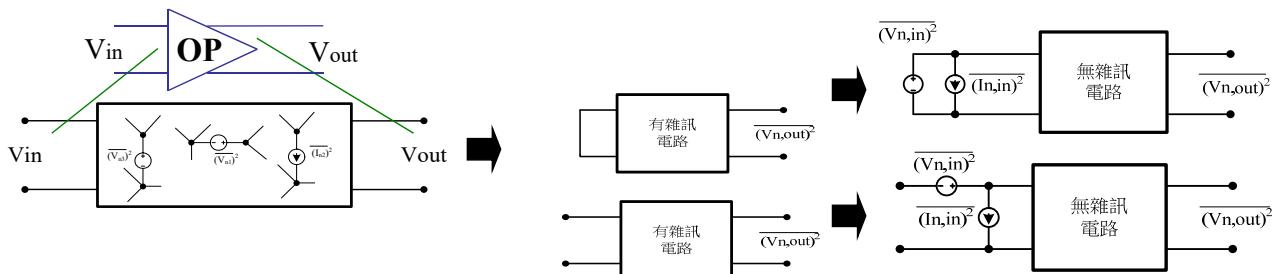
$n(t)$: Gaussian random process with $\mu=0, \sigma=1$



Ref : P.Malcovati ,S.Brigati, F.Francesconi, F.Maloberti, P.Cusinato, A.Baschirrotto
"Behavioral modeling of switched-capacitor sigma-delta modulators"
IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS – I:
Fundamental Theory and Applications, VOL. 50, NO. 3, MARCH 2003 35

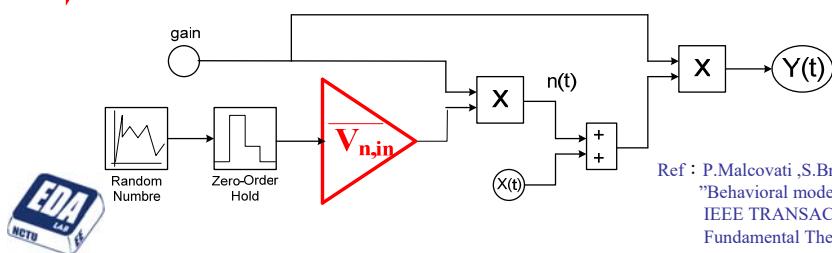
OP Noise

➤ Operational Amplifier (OP) Noise



$$V_{in} = 0 \Rightarrow \overline{V_{n,out}^2} = \text{Thermal Noise} + \text{Flicker Noise} + \text{RC Noise}$$

→ Obtained by Hspice command (.noise), ex: .noise Vout Vin 10



Ref : P.Malcovati ,S.Brigati, F.Francesconi, F.Maloberti, P.Cusinato, A.Baschirrotto
"Behavioral modeling of switched-capacitor sigma-delta modulators"
IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS – I:
Fundamental Theory and Applications, VOL. 50, NO. 3, MARCH 2003 36

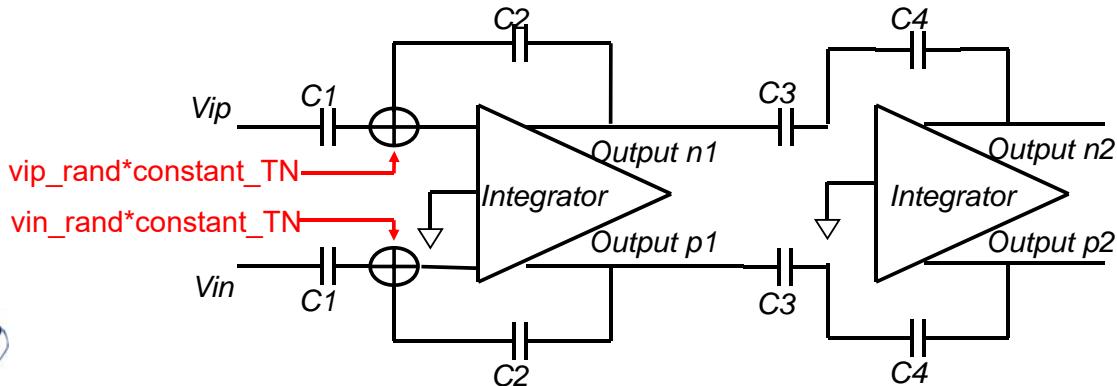
Random Noise Modeling

- Thermal noise and OP noise are random behavior
 - Cannot be measured
- Those noises can be modeled as random variables only

```

analog begin
    // assign random seed
    @(initial_step) begin
        seed_p = xxx; seed_n = ooo;
    end
    // Gauss Random Generation for thermal noise
    vip_rand=$rdist_normal(seed_p, 0 , 1);
    vin_rand=$rdist_normal(seed_n, 0 , 1);
    .....
    vip = vip + vip_rand * constant_TN
    vin = vin + vin_rand * constant_TN
end

```



37

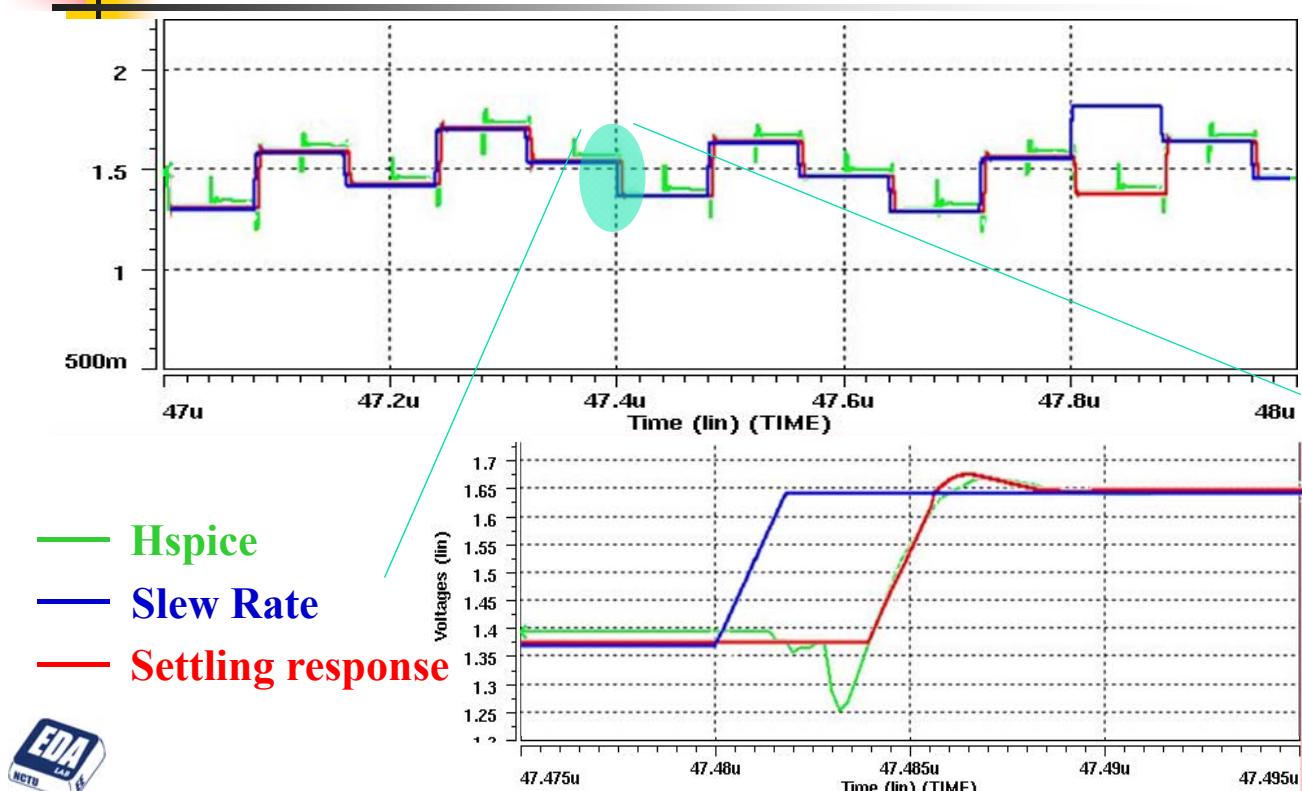
Case Study

TABLE I
Parameter of The Second-Order $\Sigma\Delta$ Modulator Model

Parameter	Value
<i>Sample frequency</i>	12.5MHz
<i>Signal frequency</i>	20.599kHz
<i>Signal frequency range</i>	19.073k~21.362kHz
<i>Noise frequency range</i>	1.5259k~40.054kHz
<i>Oversampling ratio(OSR)</i>	303
<i>Number of samples considered</i>	32768
<i>Time step</i>	200ps
<i>Window</i>	Blackman

38

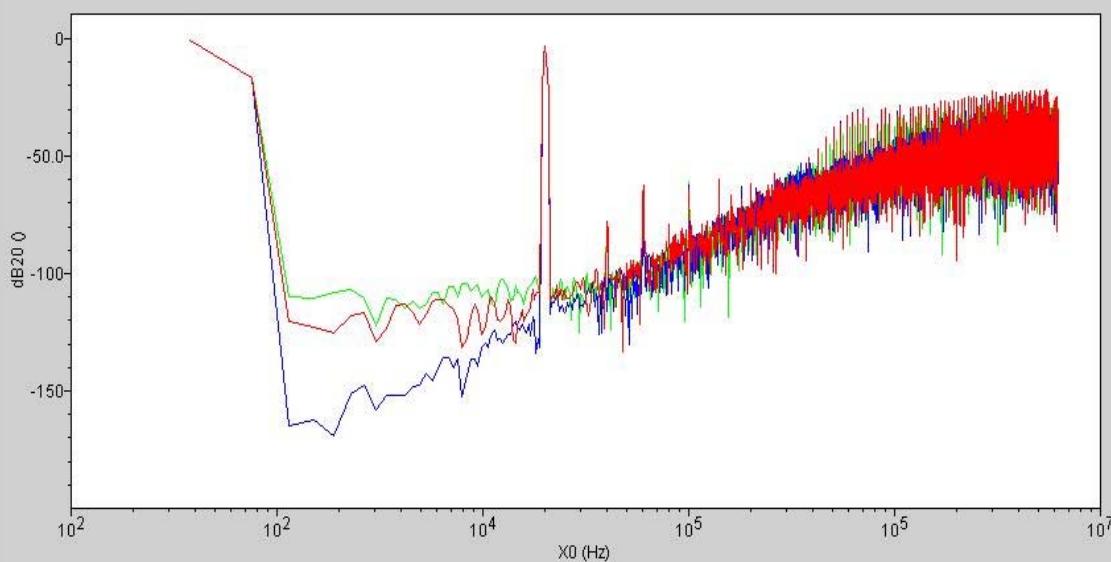
Time-Domain Comparison Results



Comparison Results - FFT

HSPICE	Ideal model	Our model with actual non-ideal effects
SNR: 85.62 dB	SNR: 91.64 dB	SNR: 86.83 dB

$\text{fft(quant, (800n), (2622.24u), (32768), (window='blackman'))}$ $\text{fft(V(dout), (800n), (2622.24u), (32768), (window='blackman'))}$
 $\text{fft(V(out), (800n), (2622.24u), (32768), (window='blackman'))}$



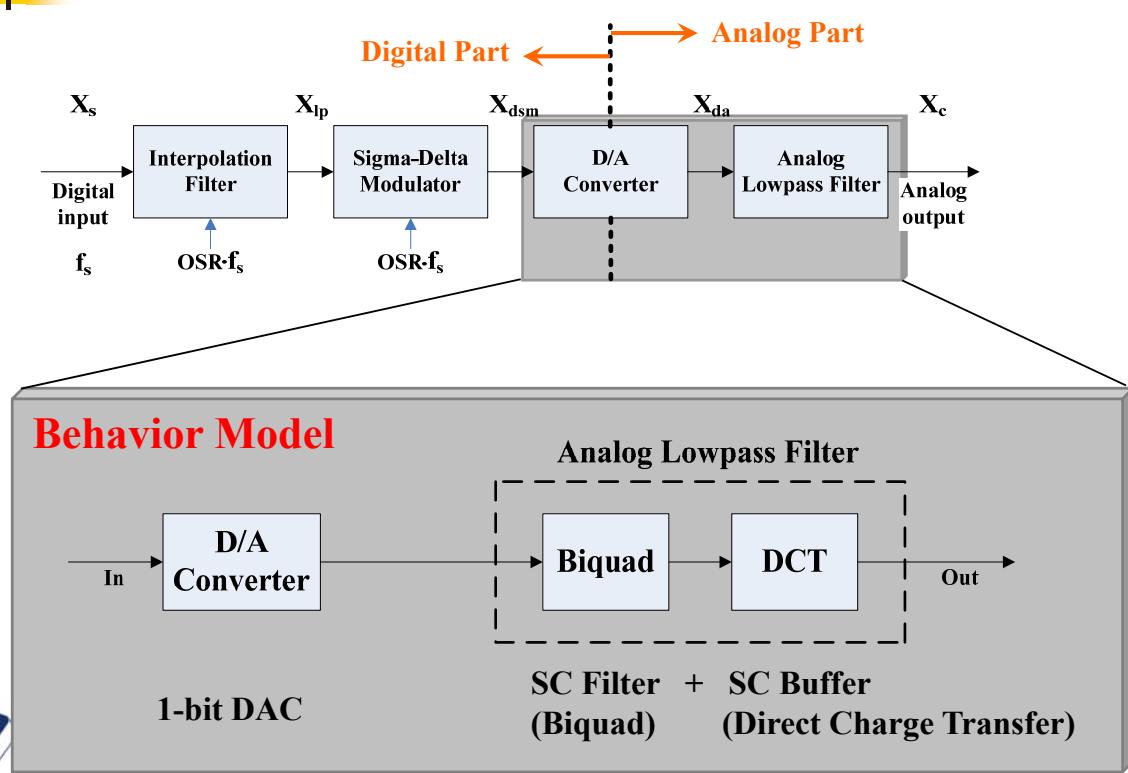
Outline

- Charge-pump phase lock loop (CPPLL)
- Delta-sigma A/D converter
- Delta-sigma D/A converter
- Transmission link system



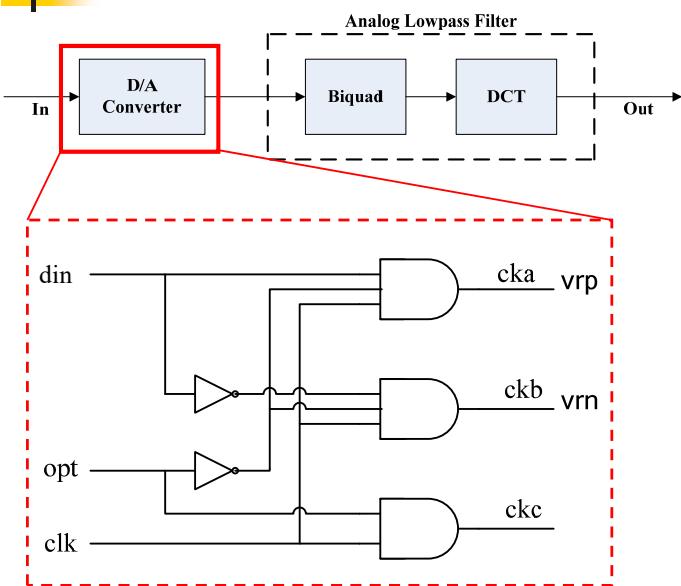
41

Sigma-Delta D/A Converter



42

1-bit D/A Converter Model



- $\text{OPT} = 0$
 - Normal mode
 - cka 、 ckb trigger
- $\text{OPT} = 1$
 - **Test mode**
 - ckc trigger

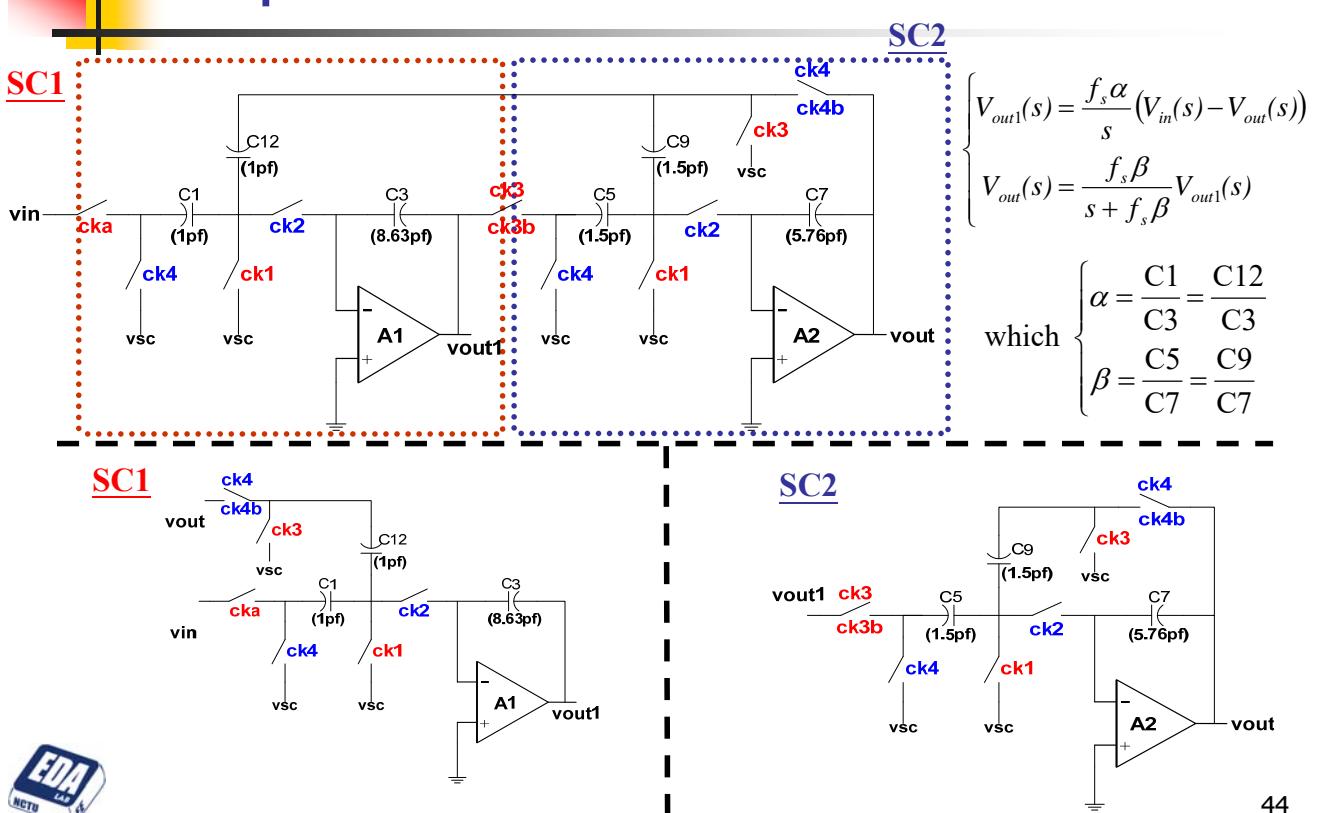
$$\text{cka} = \text{clk} \cdot \text{din} \cdot \overline{\text{opt}}$$

$$\text{ckb} = \text{clk} \cdot \overline{\text{din}} \cdot \overline{\text{opt}}$$

$$\text{ckc} = \text{clk} \cdot \text{opt}$$

43

Biquad Circuit Model



44

Biquad in Verilog-A Language

```
module da_biquad_mod2(cka,ckb,ckc,vip,vin,ck1,ck2,ck3,ck4,bop,bon,bout);
    input cka, ckb, ckc, vip, vin, ck1, ck2, ck3, ck4;
    output bop, bon, bout;
    electrical cka, ckb, ckc, vip, vin, bop, bon, bout, ck1, ck2, ck3, ck4;

    real t_vop1, t_von1, t_bop, t_bon;
    real vip_tmp, vin_tmp, vip_tmp2, vin_tmp2, vop1_tmp, von1_tmp;
    real bop_tmp, bon_tmp;

    parameter real fc=4M;
    parameter real a=0.114391, b=0.263762; // capacitor ratio
    parameter real vcm=1.5, vrp=2.25, vrn=0.75;
    parameter real td=0p, tr=100p, tf=100p;

analog begin

    @(initial_step) begin // setup initial condition
        vip_tmp=vin_tmp=vip_tmp2=vin_tmp2=vop1_tmp=von1_tmp=vcm;
        bop_tmp=bon_tmp=t_vop1=t_von1=t_bop=t_bon=vcm;
    end
```



45

```
if (V(cka) > vcm && V(ck1) > vcm) begin
    vip_tmp = vrp; vin_tmp = vrn;
end else if (V(ckb) > vcm && V(ck1) > vcm) begin
    vip_tmp = vrn; vin_tmp = vrp;
end else if (V(ckc) > vcm && V(ck1) > vcm) begin
    vip_tmp = V(vip)-bop_tmp; vin_tmp = V(vin)-bon_tmp;
    vip_tmp2 = vop1_tmp; vin_tmp2 = von1_tmp;
end else begin
    vip_tmp = vip_tmp; vin_tmp = vin_tmp; vip_tmp2 = vip_tmp2;
    vin_tmp2 = vin_tmp2;
end

t_vop1 = vcm + laplace_nd(vip_tmp, [fc*a], [0,1]); // SC1 transfer function
t_von1 = vcm + laplace_nd(vin_tmp, [fc*a], [0,1]); // differential
t_bop = vcm + laplace_nd(vip_tmp2, [fc*b], [fc*b,1]); // SC2 transfer function
t_bon = vcm + laplace_nd(vin_tmp2, [fc*b], [fc*b,1]);

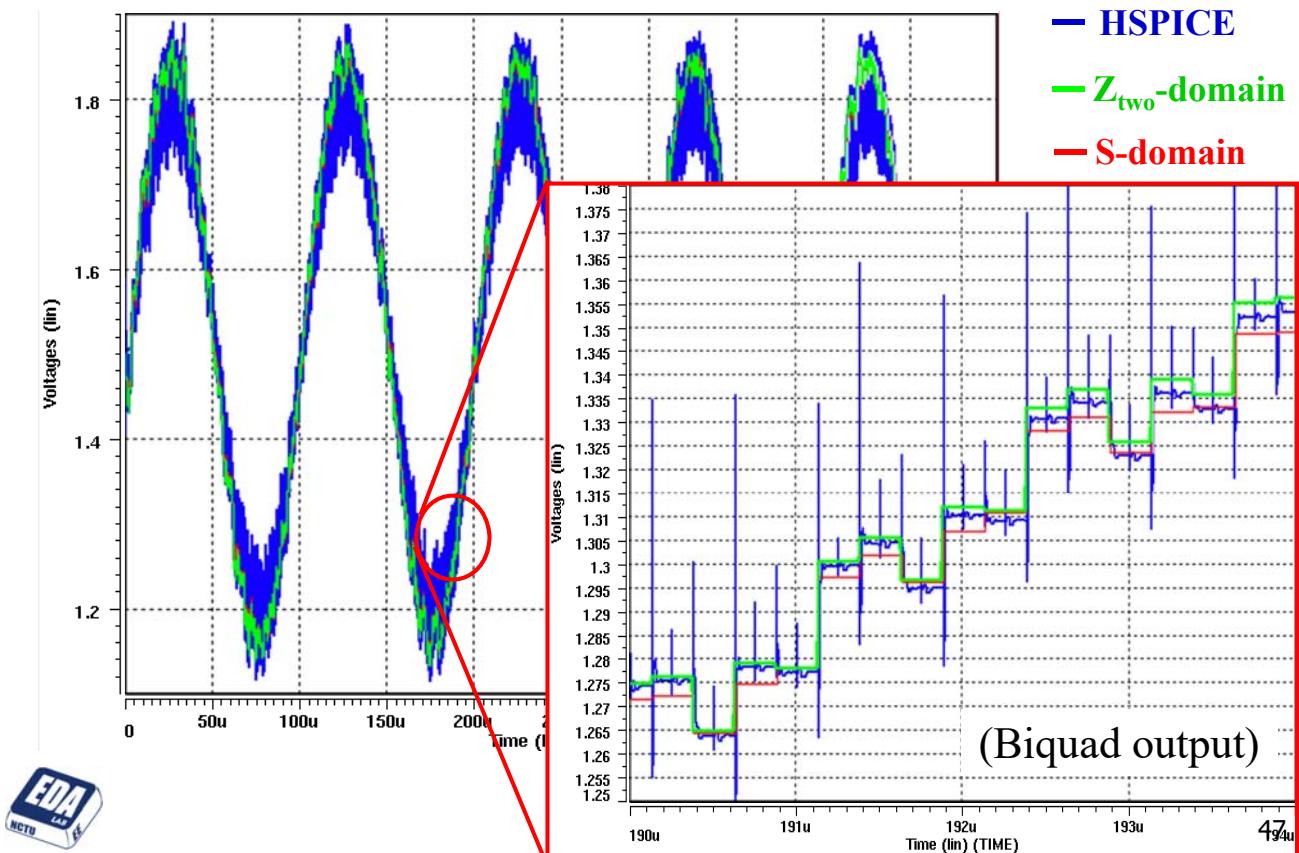
@(cross(V(ck4)-vcm,+1)) begin
    vop1_tmp = t_vop1; vop1_tmp = t_vop1; bop_tmp = t_bop; bon_tmp = t_bon;
end

V(bop) <+ transition(bop_tmp,td,tr,tf); V(bon) <+ transition(bon_tmp,td,tr,tf);
V(bout) <+ V(bop) - V(bon); // biquad output
end
endmodule
```

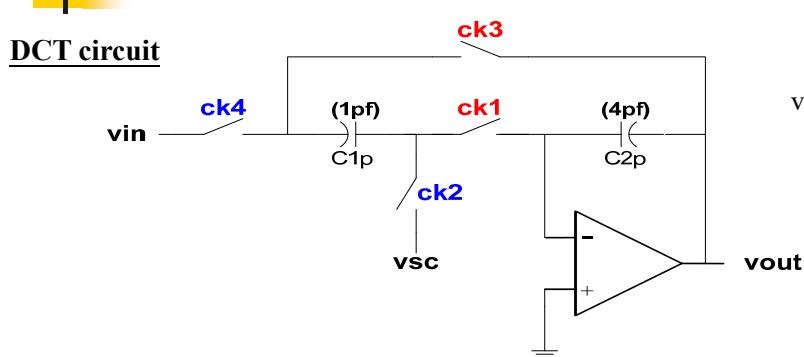


46

Waveforms of the Biquad Model

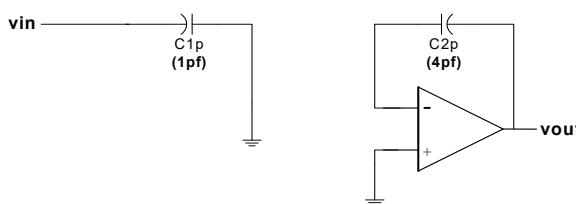


Direct Charge Transfer (DCT)

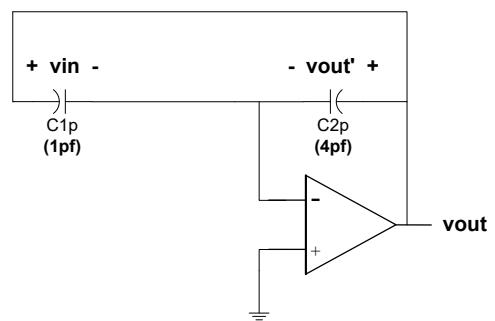


$$\begin{aligned} v_{\text{out}} &= \frac{v_{\text{in}} \cdot C1p + v_{\text{out}}' C2p}{C1p + C2p} \\ &= \frac{C1p}{C1p + C2p} v_{\text{in}} + \frac{C2p}{C1p + C2p} v_{\text{out}}' \end{aligned}$$

ck2 , ck4 on : Sample

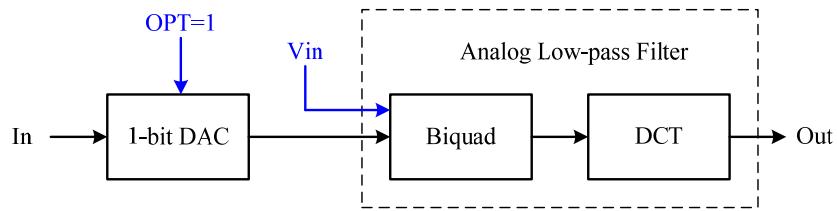


ck1 , ck3 on : Integration

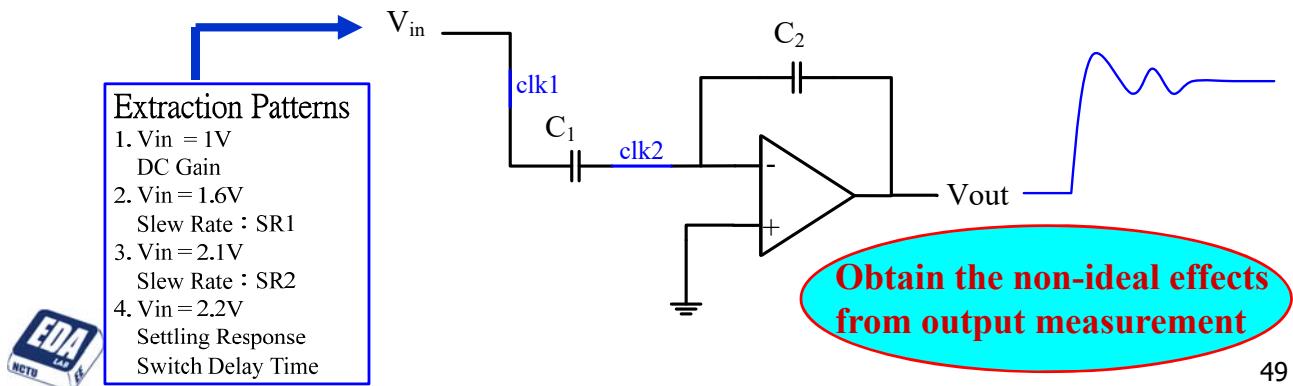


Characterization Mode

- Set OPT=1 (test mode), input become V_{in}

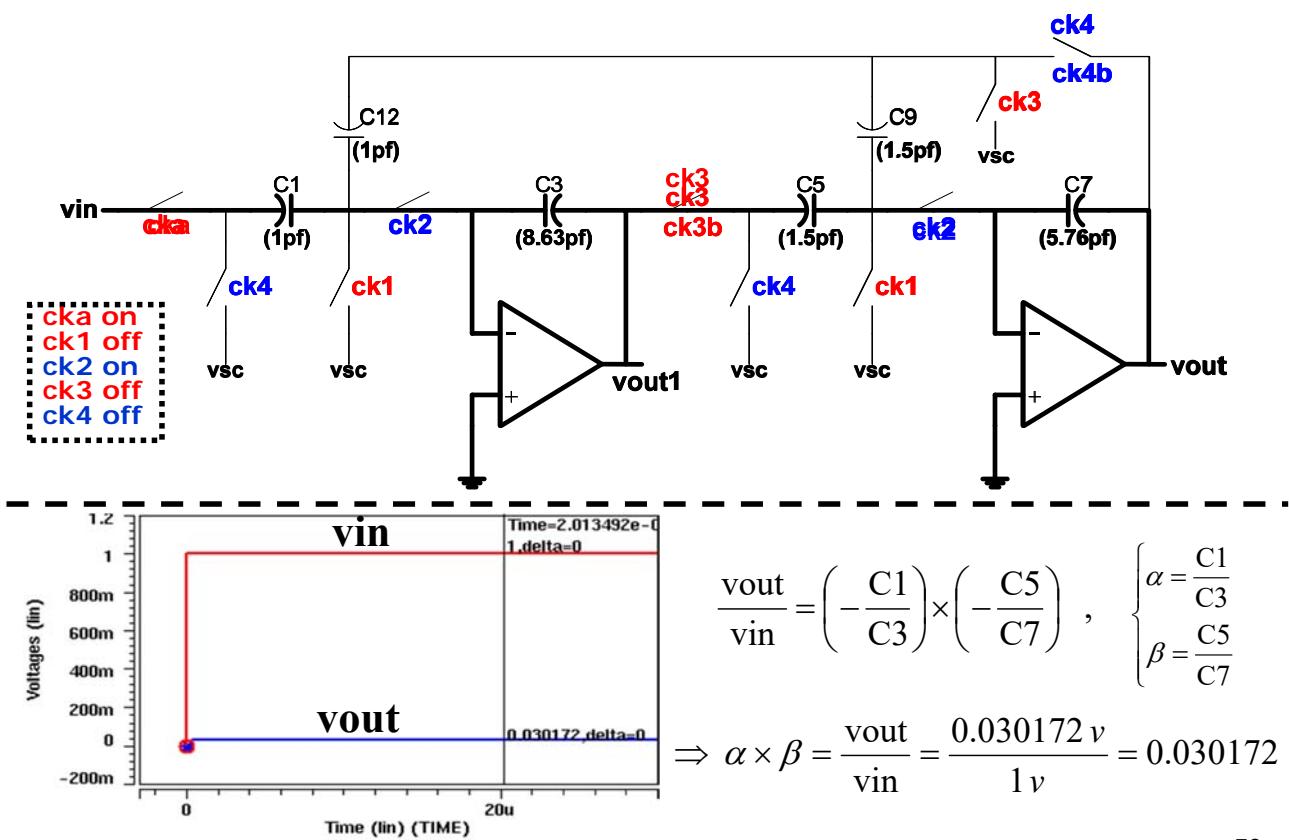


- Change clk, circuit operation in Extraction Mode



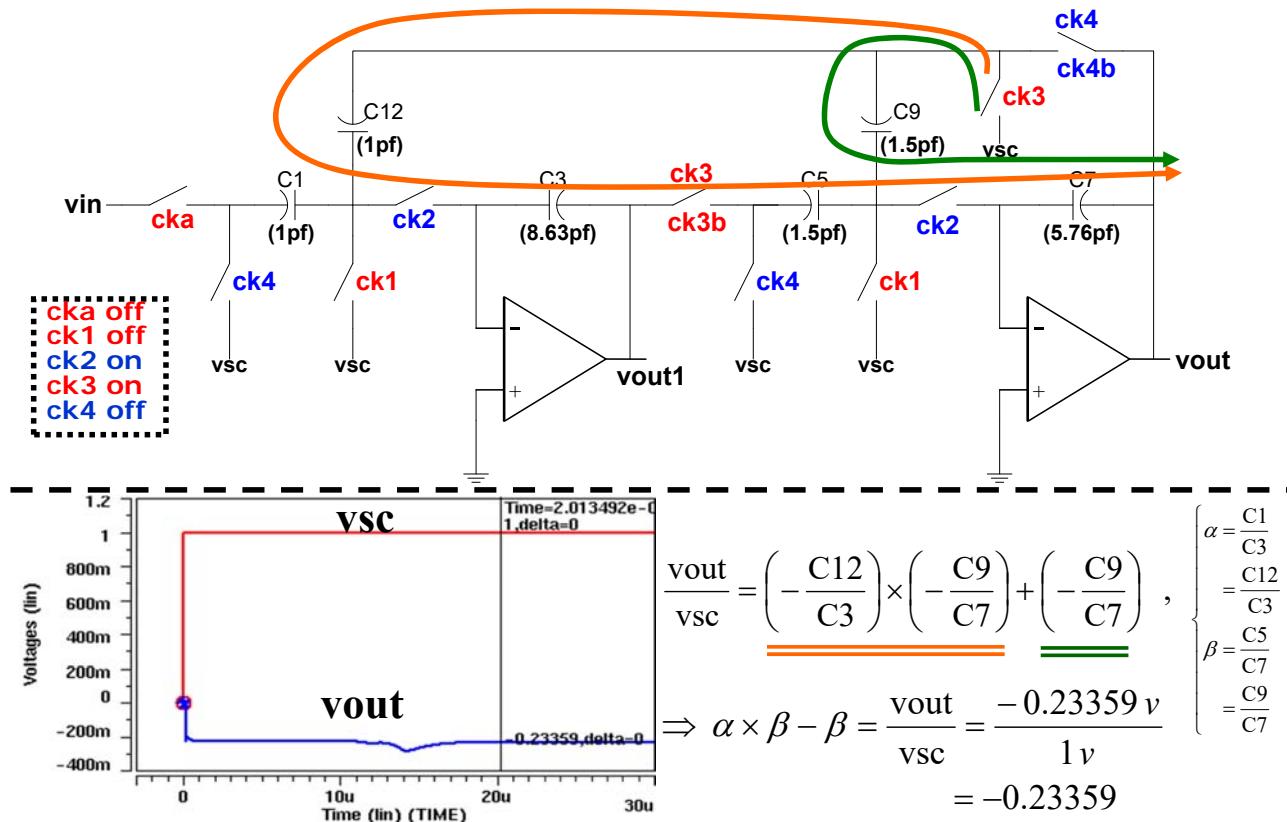
49

DC Gain Extraction – Biquad (1/3)



50

DC Gain Extraction – Biquad (2/3)



51

DC Gain Extraction – Biquad (3/3)

- Calculate the DC gain (ratio of the capacitors)

$$\begin{cases} \alpha \times \beta = 0.030172 \\ \alpha \times \beta - \beta = -0.23359 \end{cases} \Rightarrow \begin{cases} \beta = \alpha \times \beta + 0.23359 = 263.762m \\ \alpha = \alpha \times \beta / \beta = 114.391m \end{cases}$$

$\text{ideal} = \frac{1.5p}{5.76p} = 260.416m$

$\text{ideal} = \frac{1p}{8.63p} = 115.874m$

- Then add them into the Verilog-A code

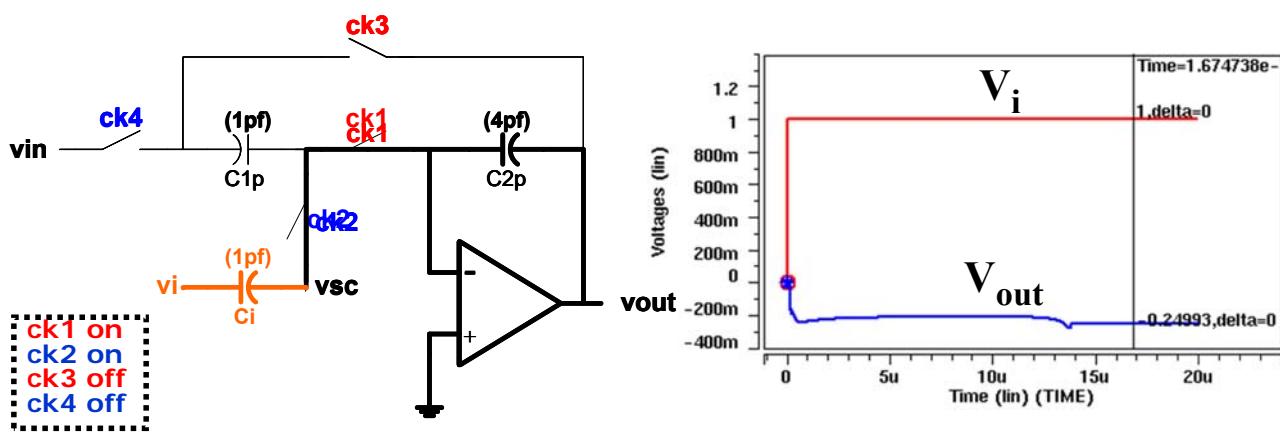
```

parameter real a=0.114391, b=0.263762;
parameter real vcm=1.5, vrp=2.25, vrn=0.75;
parameter real td=0p, tr=100p, tf=100p;

analog begin
    :
    v1 = laplace_nd(vip_tmp,{fc*fc*a*b},{fc*fc*a*b,fc*b,1});
    v2 = laplace_nd(vin_tmp,{fc*fc*a*b},{fc*fc*a*b,fc*b,1});

```

DC Gain Extraction – DCT (1/3)



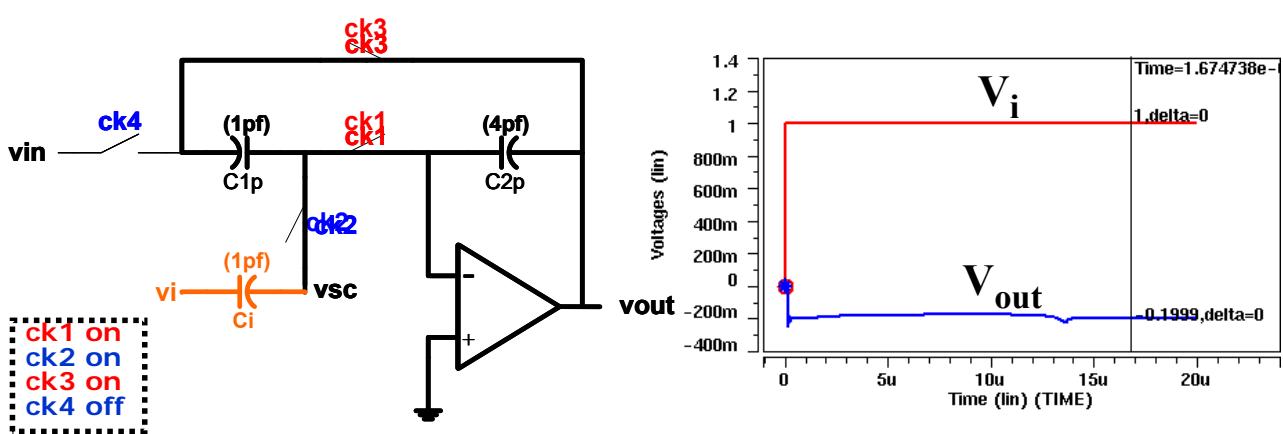
$$\frac{V_{out}}{V_i} = -\frac{C_i}{C_{2p}}$$

$$\Rightarrow \frac{C_i}{C_{2p}} = -\frac{V_{out}}{V_i} = -\frac{(-0.24993V)}{1V} = 0.24993$$



53

DC Gain Extraction – DCT (2/3)



$$\frac{V_{out}}{V_i} = -\frac{C_i}{C_{1p} + C_{2p}}$$

$$\Rightarrow \frac{C_i}{C_{1p} + C_{2p}} = -\frac{V_{out}}{V_i} = -\frac{(-0.1999)}{1} = 0.1999$$



54

DC Gain Extraction – DCT (3/3)

- Calculate the gain (ratio of the capacitors)

$$\left\{ \begin{array}{l} \frac{C_i}{C_{2p}} = 0.24993 \\ \frac{C_i}{C_{1p}+C_{2p}} = 0.1999 \end{array} \right. \Rightarrow \left\{ \begin{array}{l} \frac{C_{2p}}{C_{1p}+C_{2p}} = \frac{C_i}{C_{1p}+C_{2p}} / \frac{C_i}{C_{2p}} = 799.824m \\ \frac{C_{1p}}{C_{1p}+C_{2p}} = 1 - \frac{C_{2p}}{C_{1p}+C_{2p}} = 200.176m \end{array} \right.$$

ideal = $\frac{4p}{1p+4p} = 800m$
ideal = $\frac{1p}{1p+4p} = 200m$

- Then add them into the Verilog-A code

```

parameter real ci_c2p = 0.24993, ci_c1pc2p = 0.1999;
parameter real vcm=1.5,
parameter real td=0p, tr=100p, tf=100p;

analog begin
    :
    @(cross(V(ck3)-vcm,+1))
    vip_tmp = V(vip); //actually sampled at ck2,ck4
    vin_tmp = V(vin); //the static value was sampled
    vop_pre = vop_pre;
    von_pre = von_pre;
    vop_tmp = (vip_tmp* (1-ci_c1pc2p) )+(vop_pre* ci_c1pc2p/ci_c2p );
    von_tmp = (vin_tmp* (1-ci_c1pc2p) )+(von_pre* ci_c1pc2p/ci_c2p );
end

```



55

Slew Rate Extraction – Overall Filter

Given V_{in1} , we can get ΔV_1 , SR_1

Given V_{in2} , we can get ΔV_2 , SR_2

Then use

$$\frac{SR_2 - SR_1}{SR - SR_2} = \frac{\Delta V_2 - \Delta V_1}{\Delta V - \Delta V_2}, \text{ where } \Delta V = V_{in} - V_{out}$$

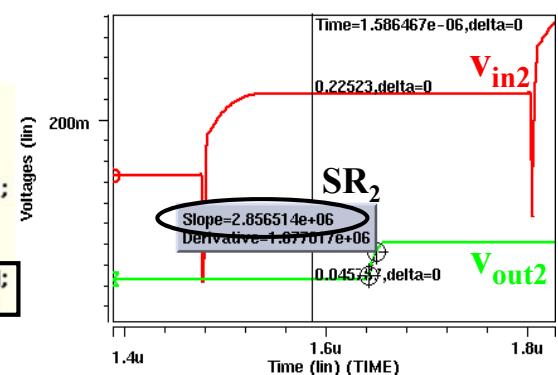
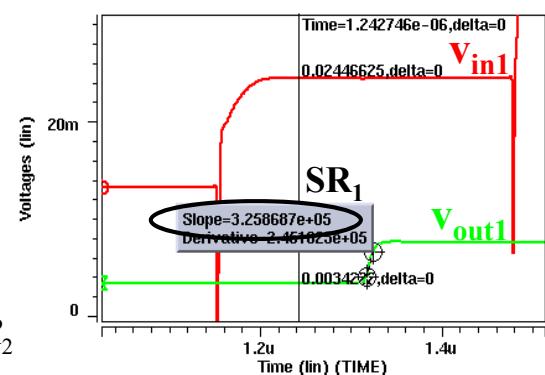
$$\Rightarrow SR(V_{in}) = \frac{(V_{in} - V_{out}) - \Delta V_2}{\Delta V_2 - \Delta V_1} (SR_2 - SR_1) + SR_2$$

- Adding it in the Verilog-A code

```

module dct_sr(dop,don,dout,vip,vin,ck1,ck2,ck3,ck4);
  input vip, vin, ck1, ck2, ck3, ck4;
  output dop, don, dout;
  electrical vip, vin, ck1, ck2, ck3, ck4, dop, don, dout;
  real vip_tmp, vin_tmp, vop_tmp, von_tmp, vop_pre, von_pre;
  parameter real ci_c2p = 0.24993, ci_c1pc2p = 0.1999;
  parameter real vth=1.5, vcm=1.5;
  parameter real td=0n, trtf=2n;
  parameter real v1=(0.02446625 - 0.003422), SR1=0.3258687M;
  parameter real v2=(0.22523 - 0.045757), SR2=2.856514M;

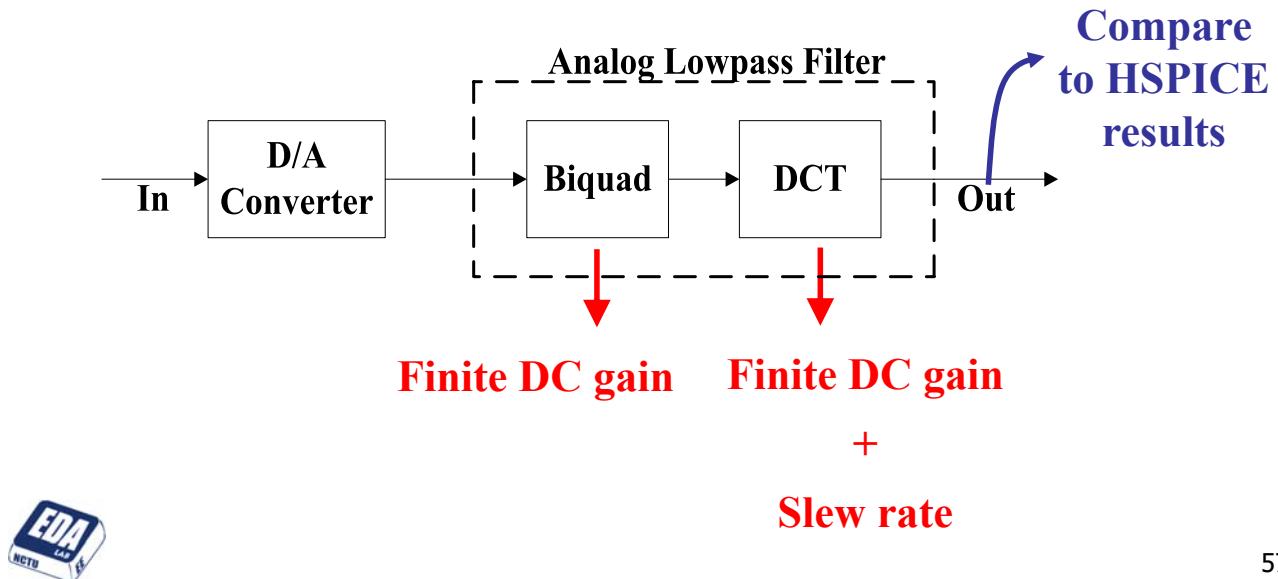
```



56

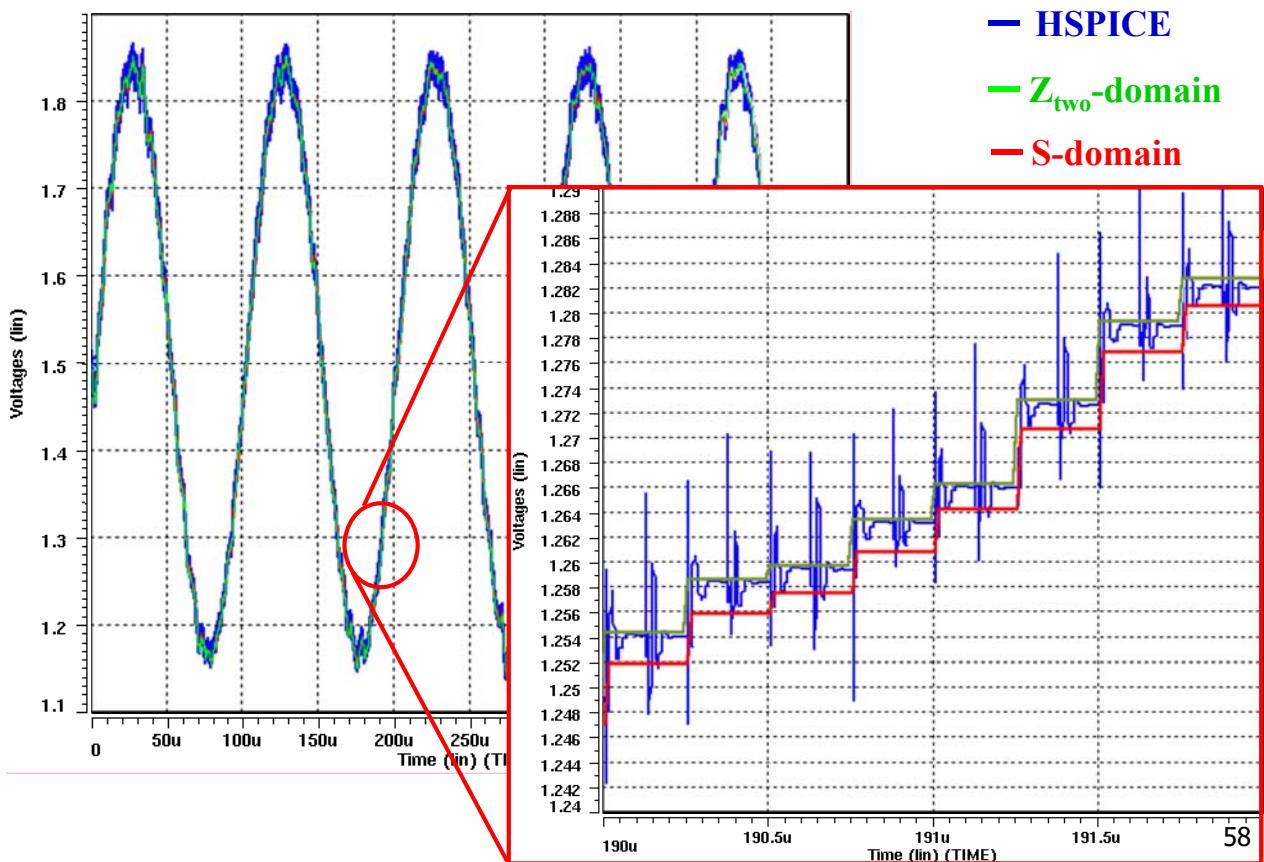
Experiments

- Our behavioral model:
 - Add finite DC gain & Slew rate



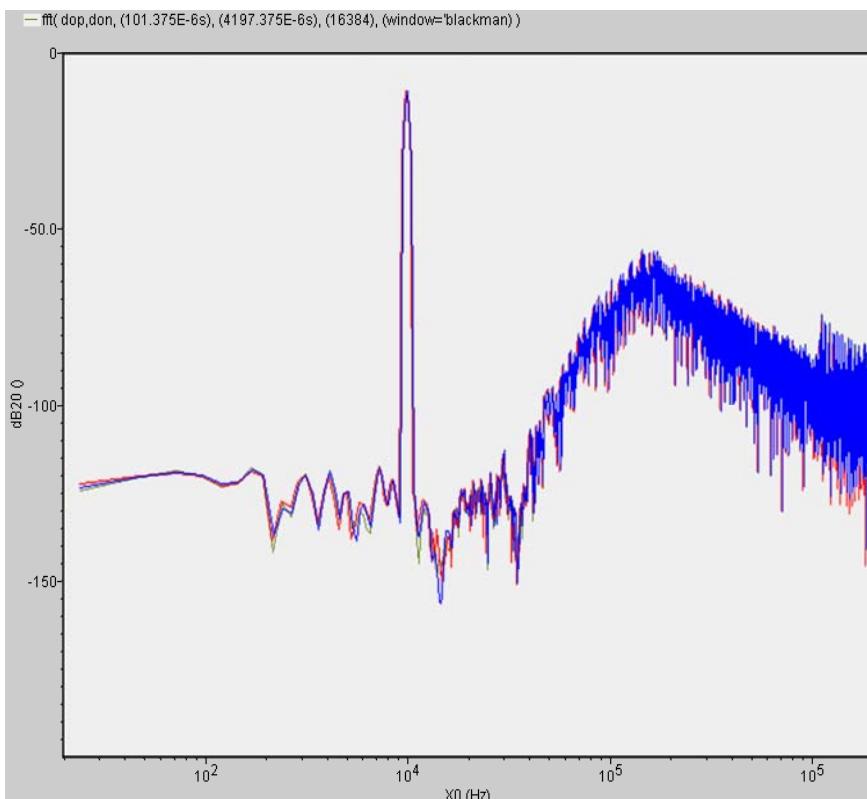
57

Time Domain Comparison Waveforms



58

Comparison Waveforms – FFT

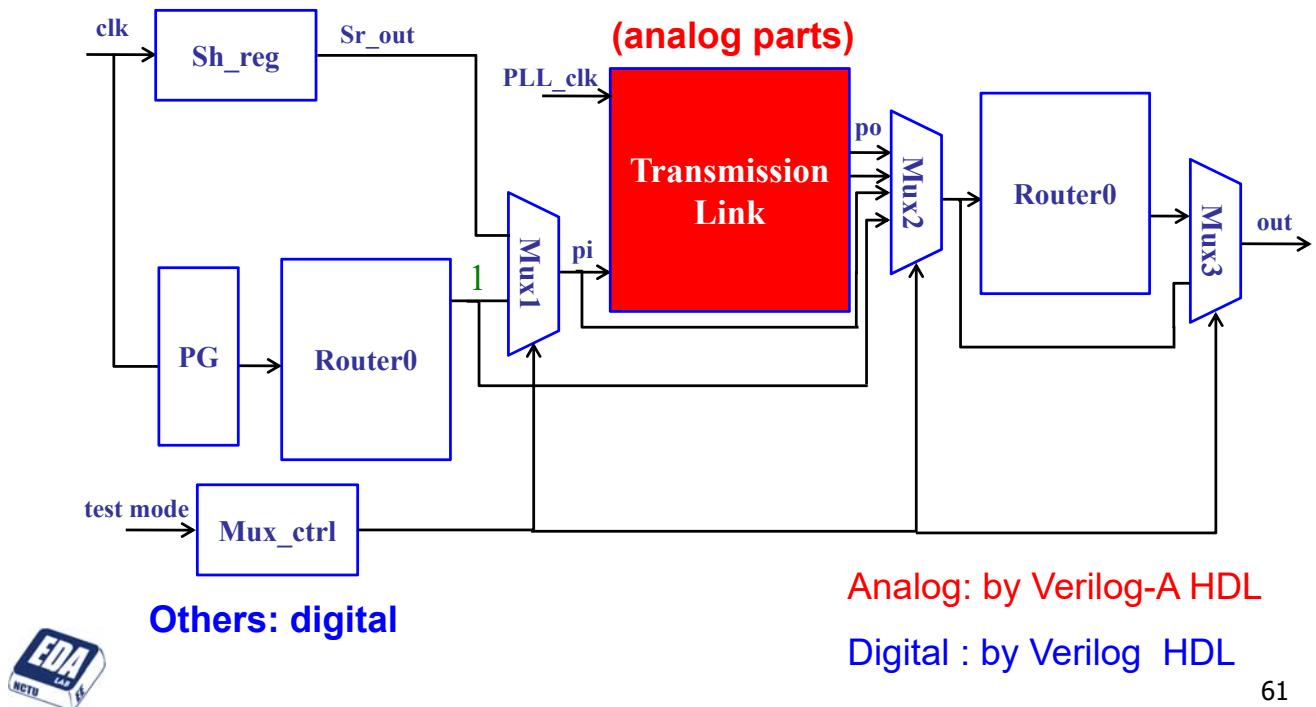


59

Outline

- Charge-pump phase lock loop (CPPLL)
- Delta-sigma A/D converter
- Delta-sigma D/A converter
- Transmission link system

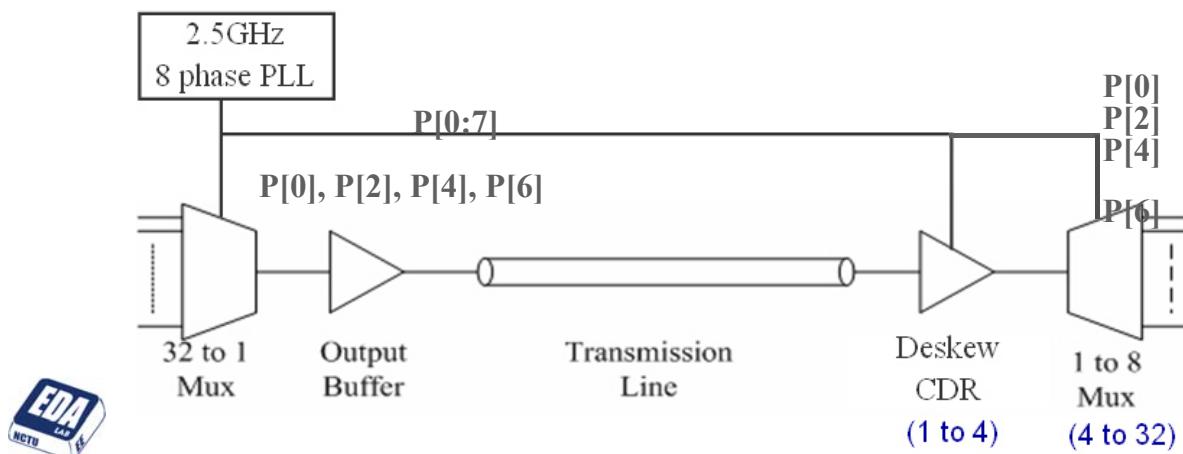
Transmission Link System



61

Analog Part in Transmission Link

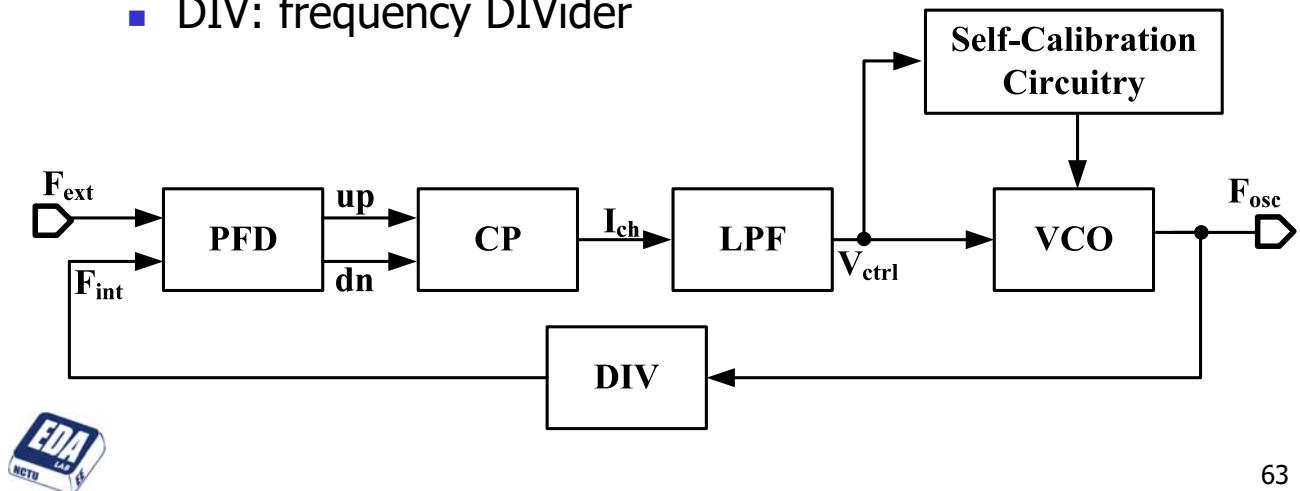
- Transmitter: 8 phase 2.5GHz **PLL** & **Serializer**
- Receiver: **CDR** & **Deserializer**
- Transmission line: RLCG Model



62

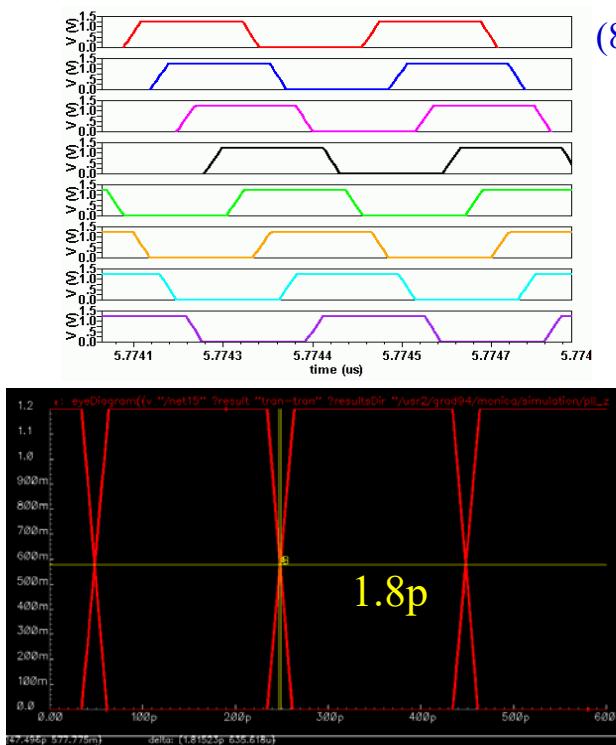
2.5GHz PLL Design

- PFD: Phase Frequency Detector
- CP: Charge Pump
- LPF: Low-Pass Filter
- VCO: Voltage-Controlled Oscillator
- DIV: frequency DIVider



63

PLL Modeling Results

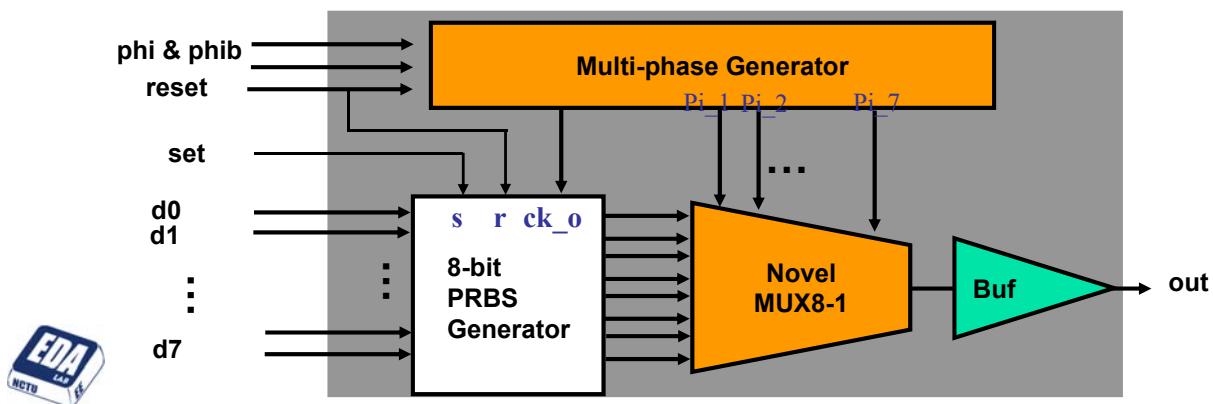
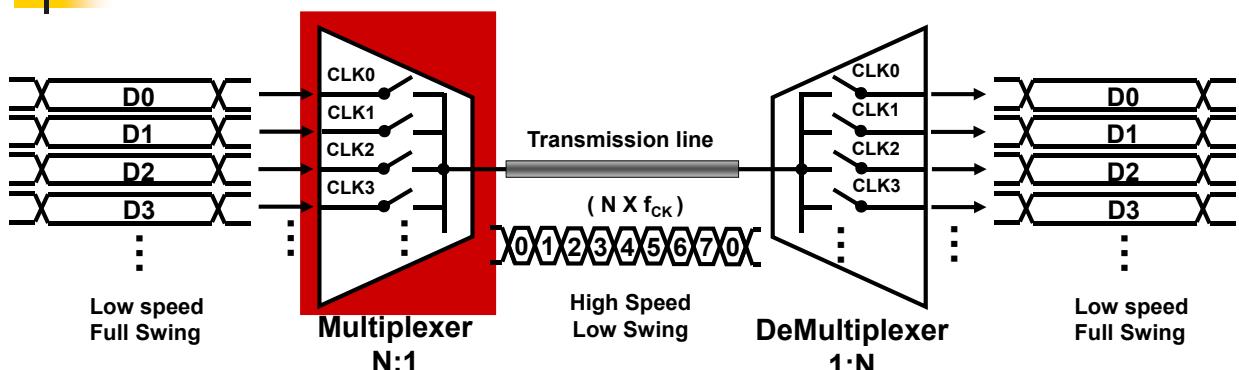


(8 phase outputs)

	HSPICE	PLL Model
Process	TSMC 0.13um 1P8M	
Input Frequency	312.5MHz	
Output Frequency	2.5GHz	
Phase	8	
Locking Time	1.3us	0.9us
Jitter	2.3ps	1.8ps
T_{sim}	99218sec	194sec

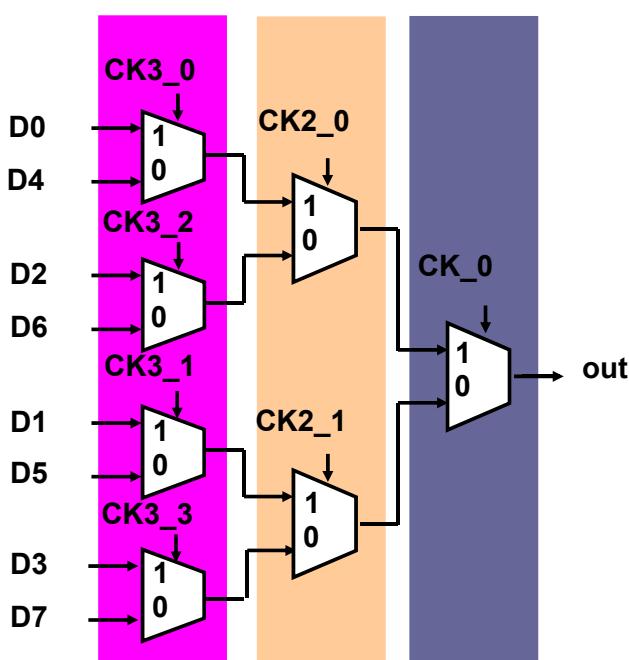
64

Serializer



65

Mux8-1 Model



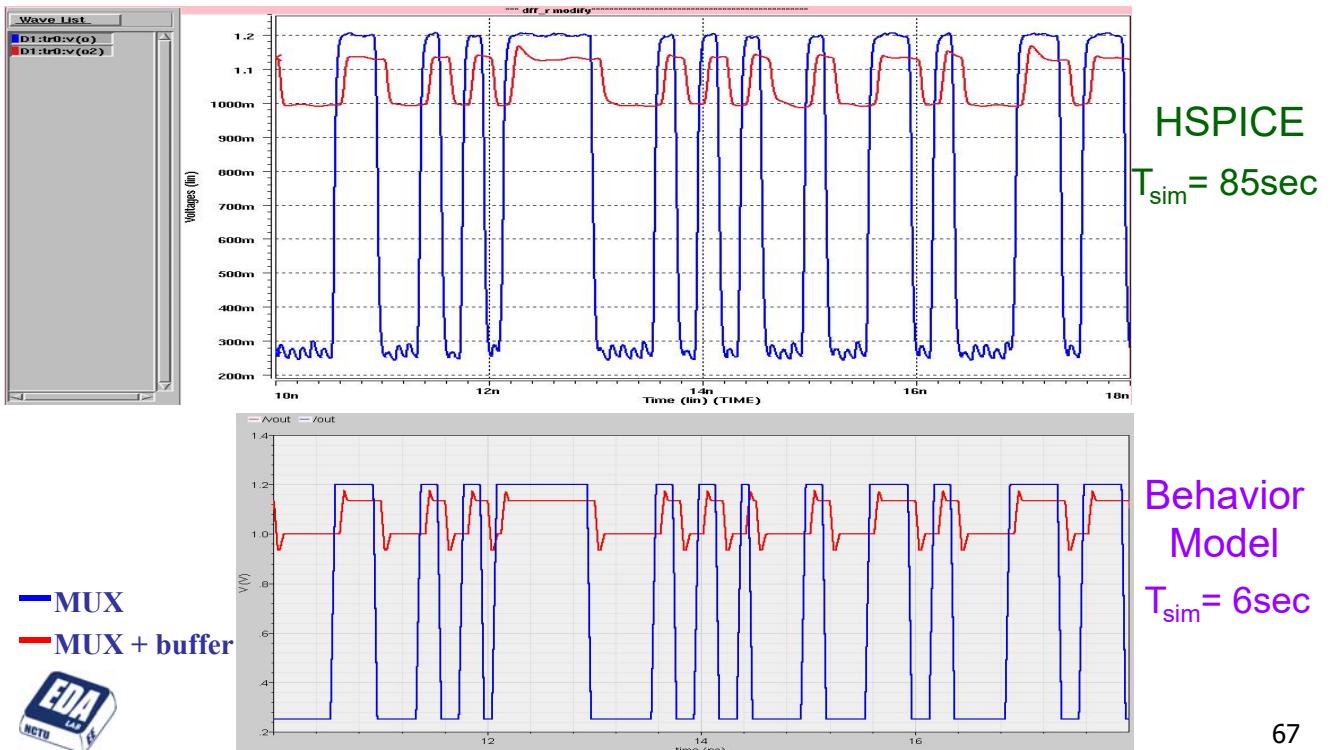
Mux2-1:

```
if (sel > vtrans)
    V(out) <+ V(in1);
else
    V(out) <+ V(in2);
```

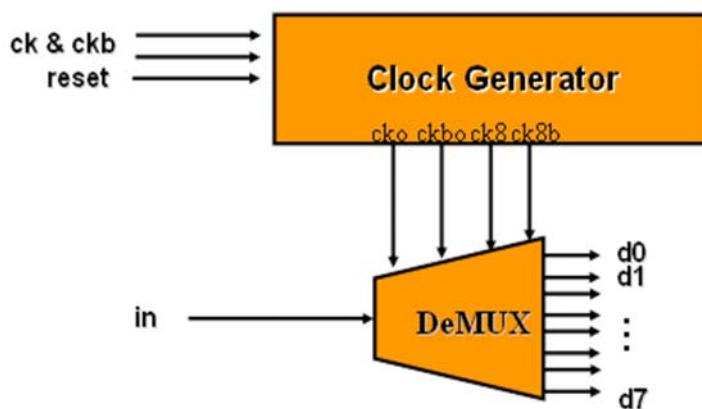
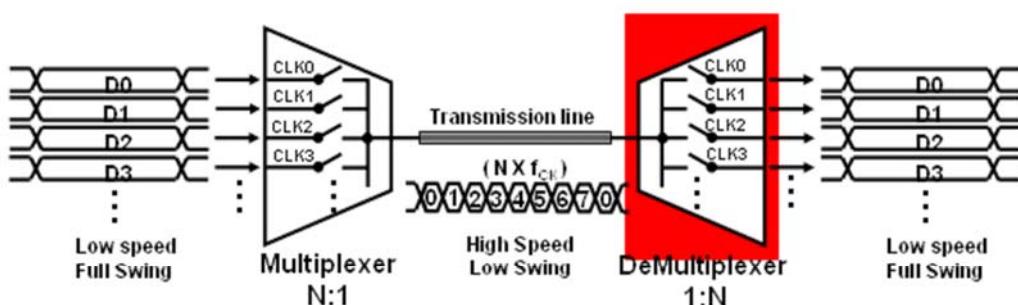


66

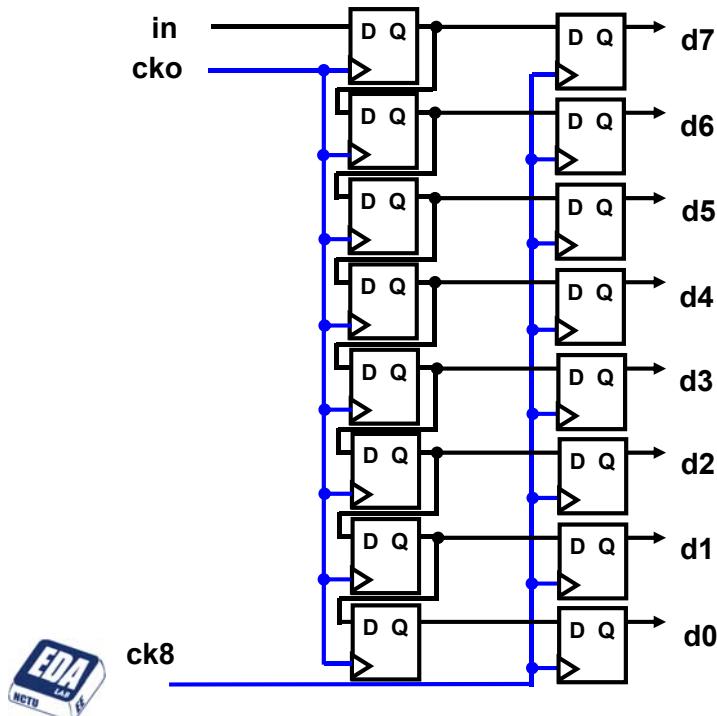
Serializer Simulation Results



DeSerializer



DeMultiplexer Model



```

if(V(ck8)==VL) begin
  o7o = in7;
  o6o = in6;
  o5o = in5;
  o4o = in4;
  o3o = in3;
  o2o = in2;
  o1o = in1;
  o0o = in0;
end
else begin
  o7o = o7o;
  o6o = o6o;
  o5o = o5o;
  o4o = o4o;
  o3o = o3o;
  o2o = o2o;
  o1o = o1o;
  o0o = o0o;
end

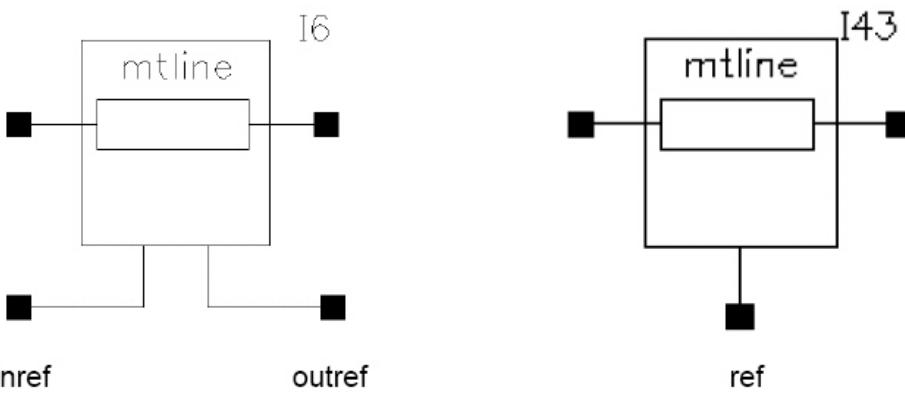
@(cross(V(ck8)-vtrans,+1)) begin
  d7 = o7o;
  d6 = o6o;
  d5 = o5o;
  d4 = o4o;
  d3 = o3o;
  d2 = o2o;
  d1 = o1o;
  d0 = o0o;
end

```

69

Transmission Line Model

- **Multi-Conductor Transmission Line (*mtline*)**
- Characterized by **constant RLCG** matrices or **frequency dependent RLCG** data

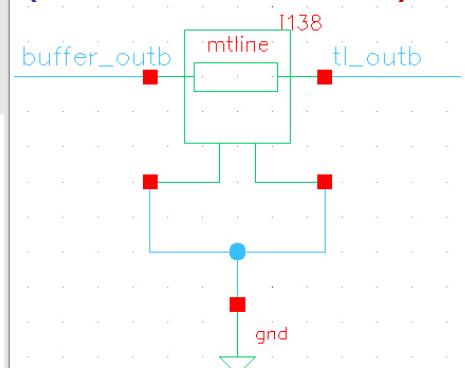


70

CDF Parameter	Value	Display
Num of lines (excluding ref.)	1	off
RLCG data file		off
use Img subckt	<input type="checkbox"/>	off
Invoke 'LMG' parameter extraction tool		
Physical length	5m	length = 5m
Enter RLCG etc. matrices	<input checked="" type="checkbox"/>	off
R matrix per unit length	49.549	$R_{total} = 49.549$
L matrix per unit length	7.859n	$L_{total} = 7.859n$
G matrix per unit length	0.0	$G_{total} = 0.0$
C matrix per unit length	3.0705p	$C_{total} = 3.0705p$
Skin effect res matrix per uni	0.00005	$Rs = 0.00005$
Dielectric loss cond matrix pe	1e-12	$Gd = 1e-12$
Frequency scale factor		off
Max signal frequency	10G	off
No. of Harmonics for PSS		off
Multiplicity factor	1	off



(Parameters of *mtline*)

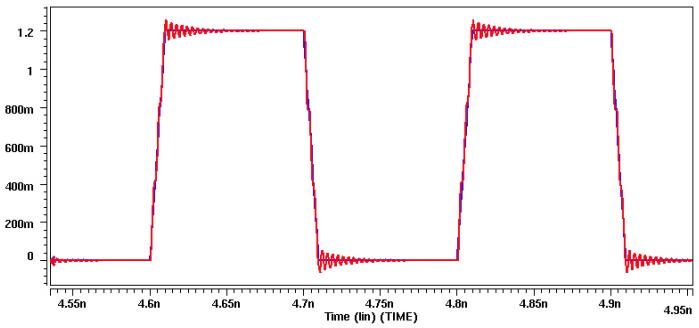


71

Transmission Line Outputs

- 10GHz data in

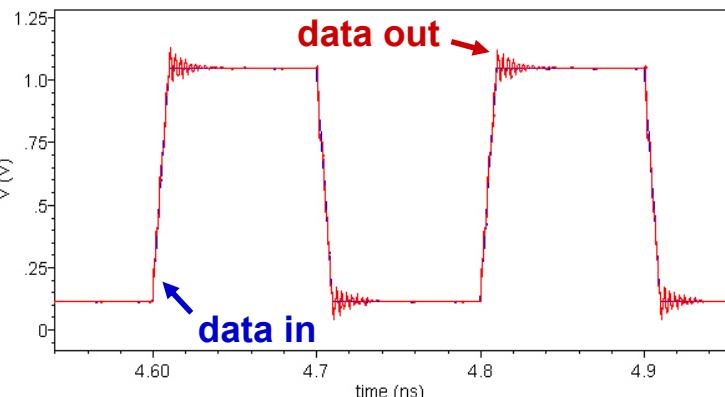
Transistor level:
HSPICE (w-element)



Analog model:

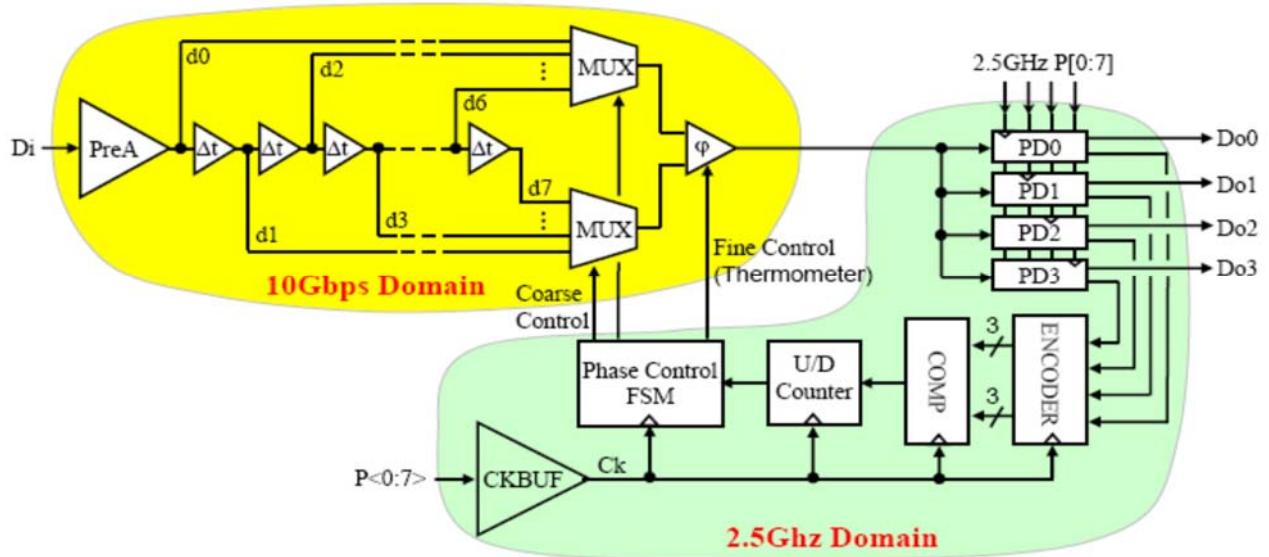
Spectre (mtline)

T_{sim} : 4x faster



DeSkew CDR

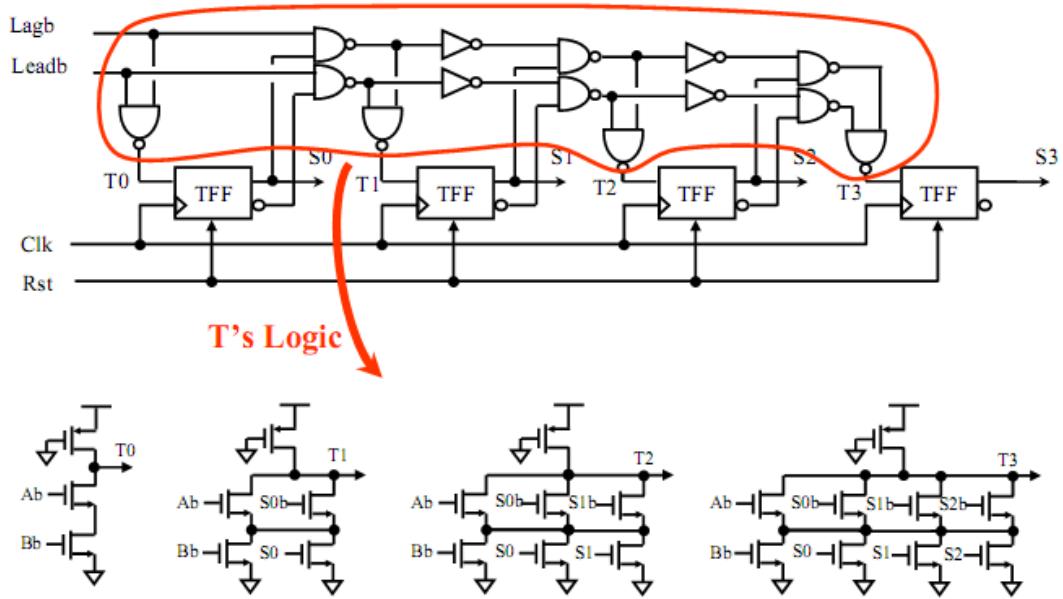
- 2.5Ghz Tree-Type-Adder Structure



73

Confidence Counter (CC) in CDR

TFF Up/Down Counter (Signed 4-bit Up/Dn Counter)



74

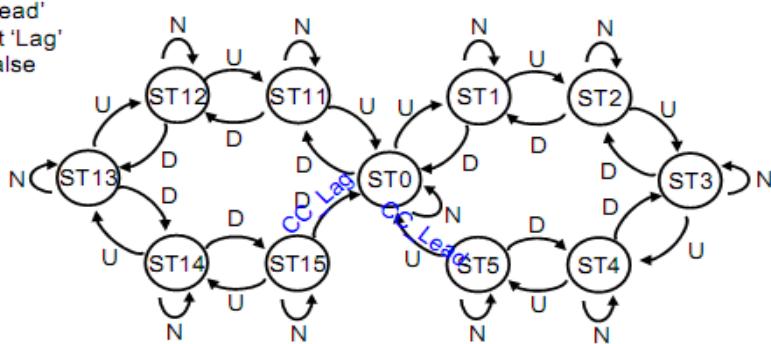
CC Behavior Model

Definition

U: Up, that is input 'Lead'
 D: Down, that is input 'Lag'
 N: Both U & D are False

Lim+ = +6
 Lim- = -6

State Transition Diagram



```

@(cross(V(clk) - vtrans, +1)) begin
  if(lead == 1 && lag == 0)
    count = count + 1;
  else if(lag == 1 && lead == 0)
    count = count - 1;
  else
    count = count;
  
```

```

if (count == +6) begin
  count = 0; cc_lead = 1;
end
else if (count == -6) begin
  count = 0; cc_lag = 1;
end
else begin
  count = count; cc_lead = 0; cc_lag = 0;
end
  
```



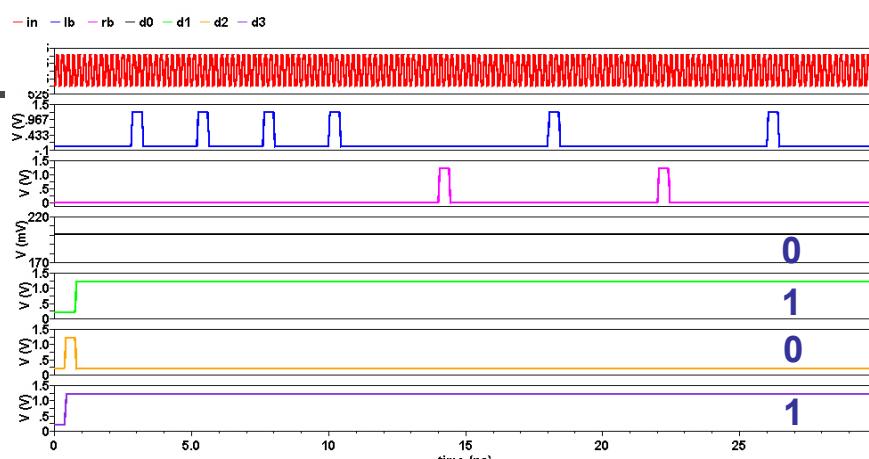
75

CDR Simulation Results

(Data in)

cc_lead

cc_lag



CDR Model

T_{sim}

= 25sec

cc_leadbar

cc_lagbar

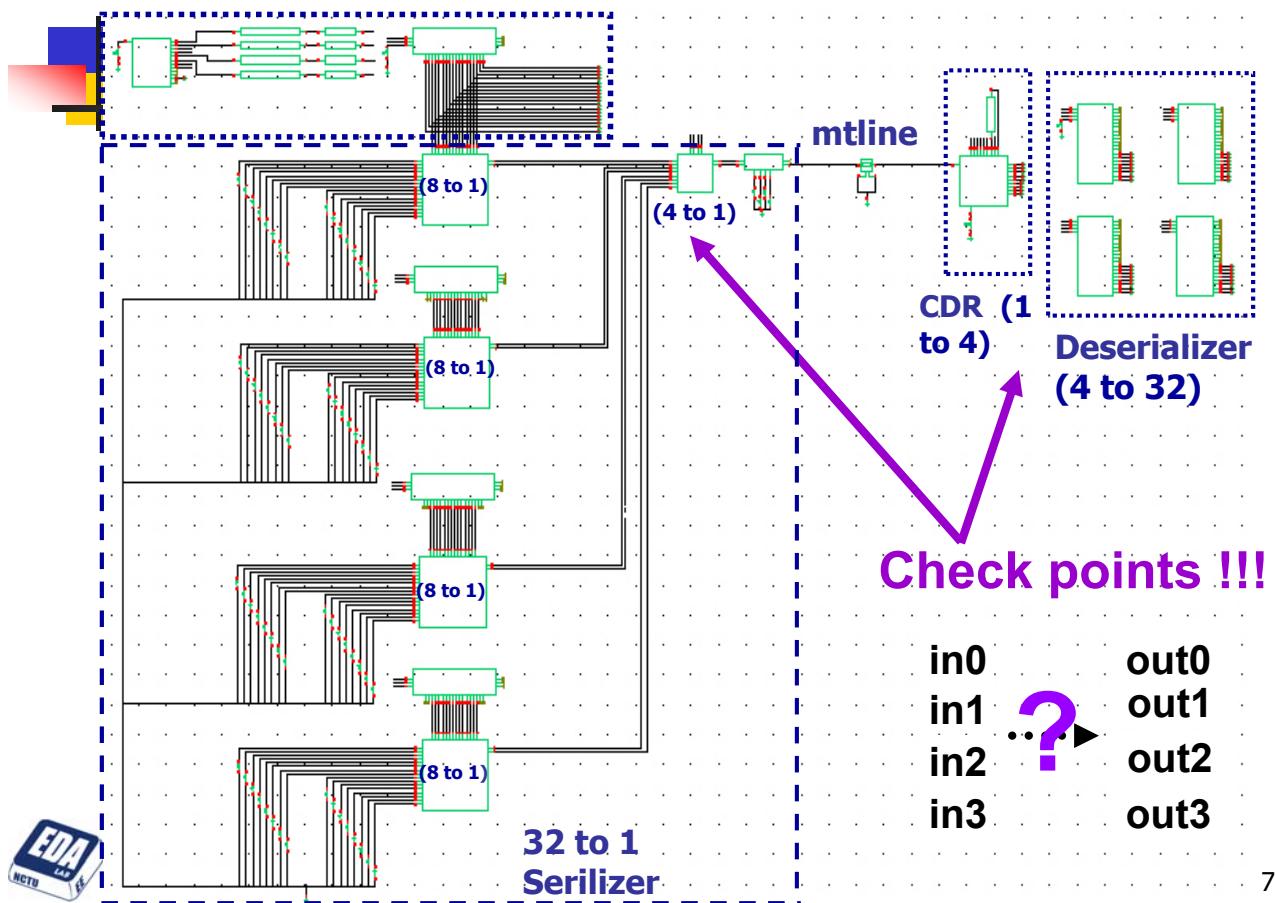
cc_lead

cc_lag

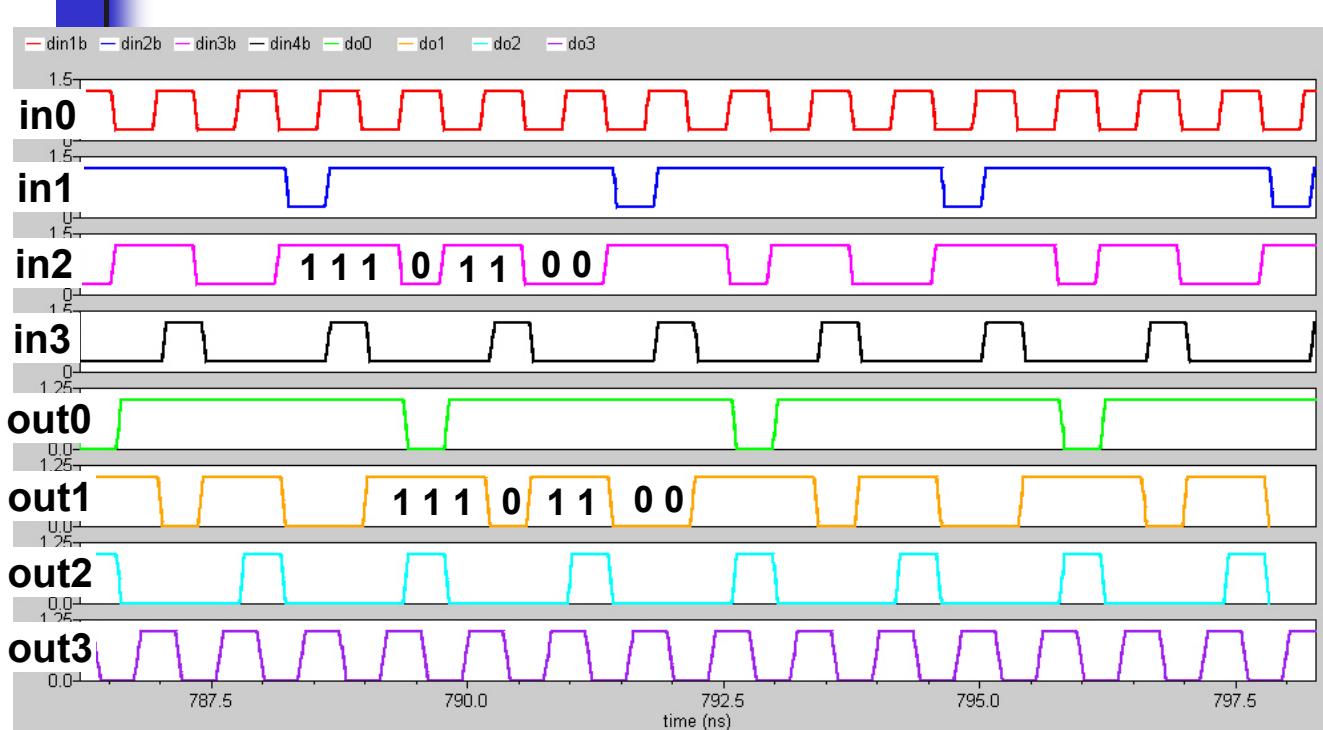
Voltage [mV]

PLL & CK Generator

(Integration of Analog Part)



Date Rotation/Shift



Data in/out mapping:

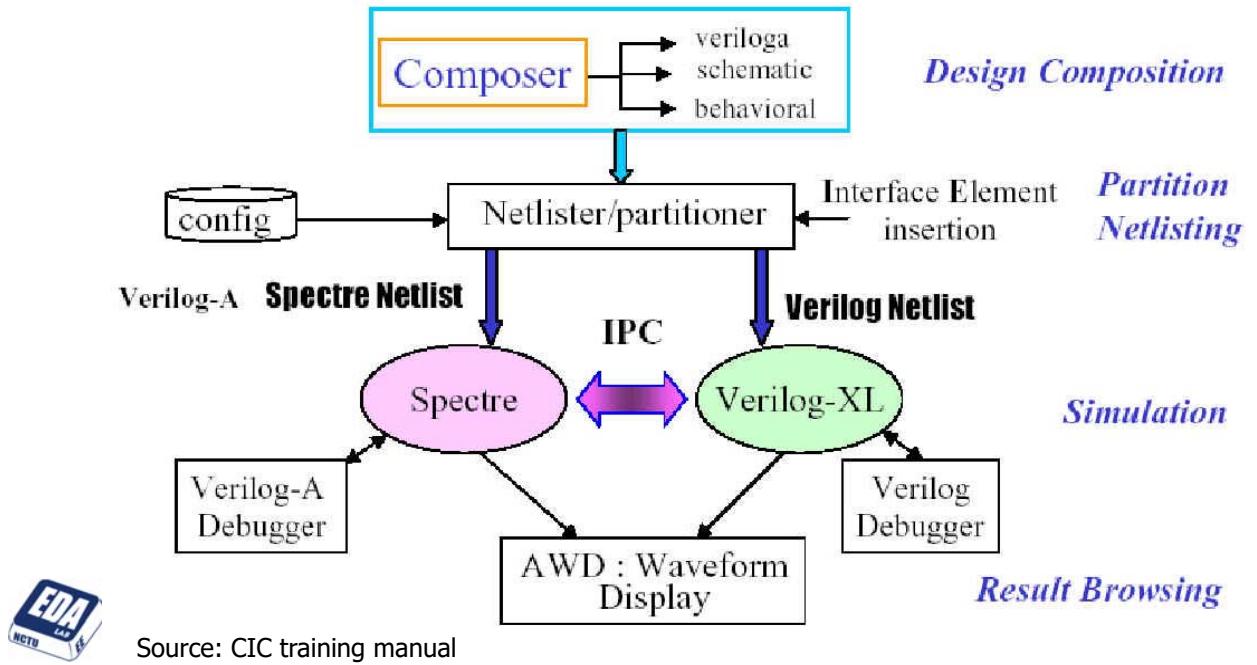
In0 → out3
In1 → out0

In2 → out1
In3 → out2

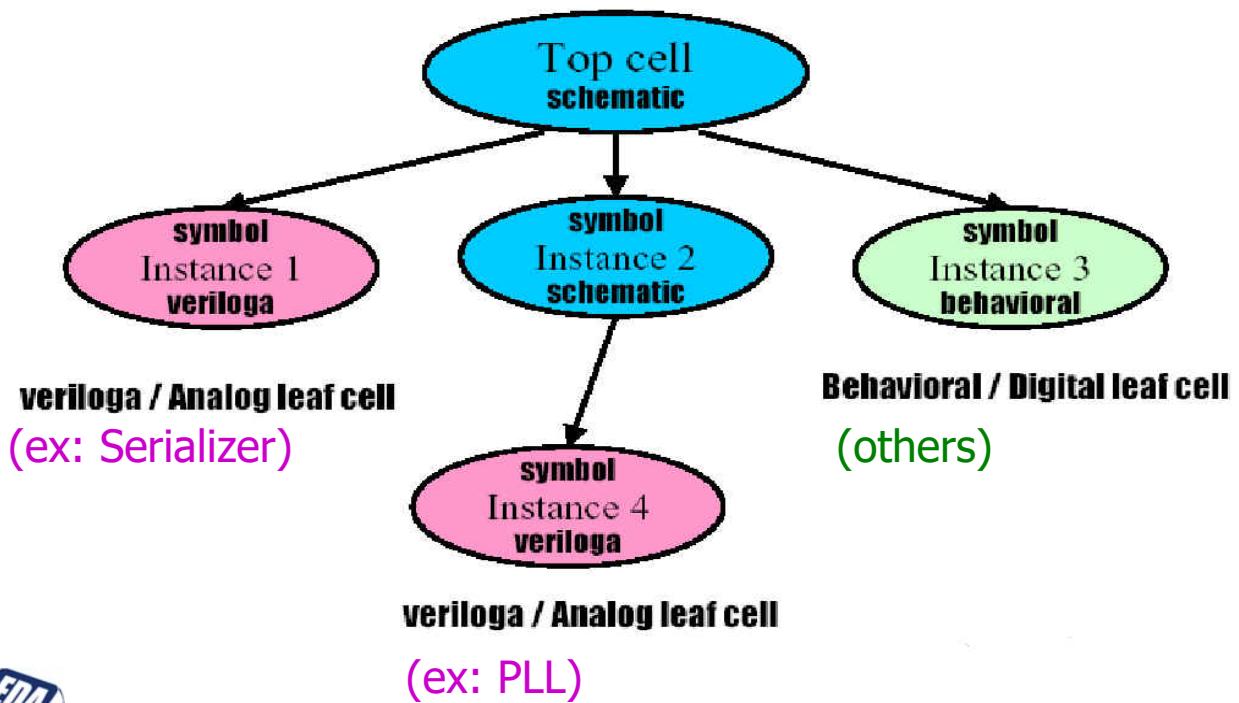


Cadence Co-simulation Platform

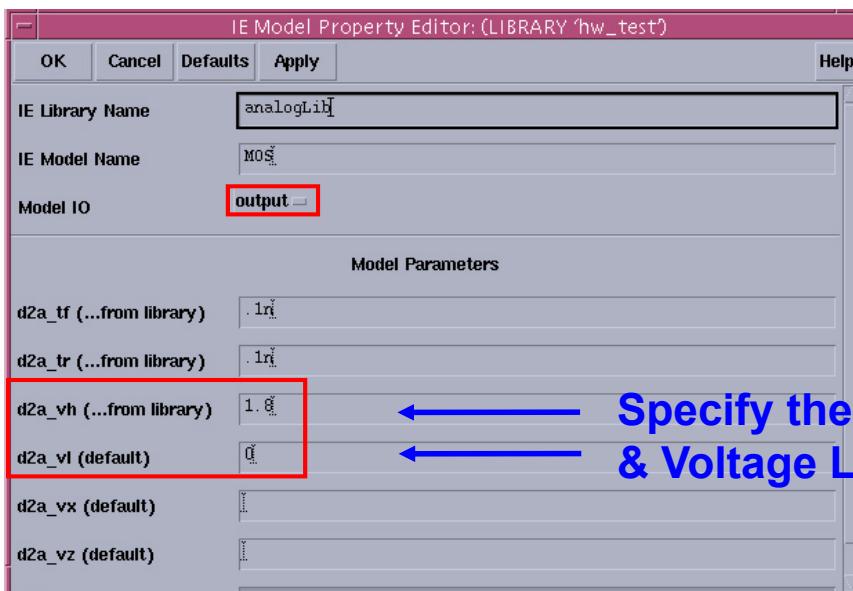
- Integrate digital & analog behavioral models



Modeling Design Hierarchy

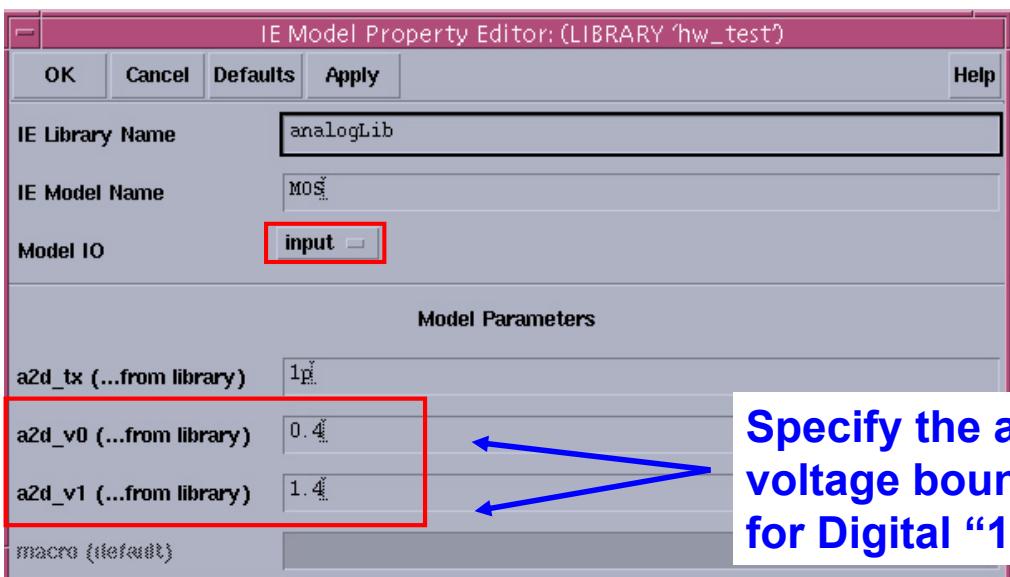


D/A Interface Element



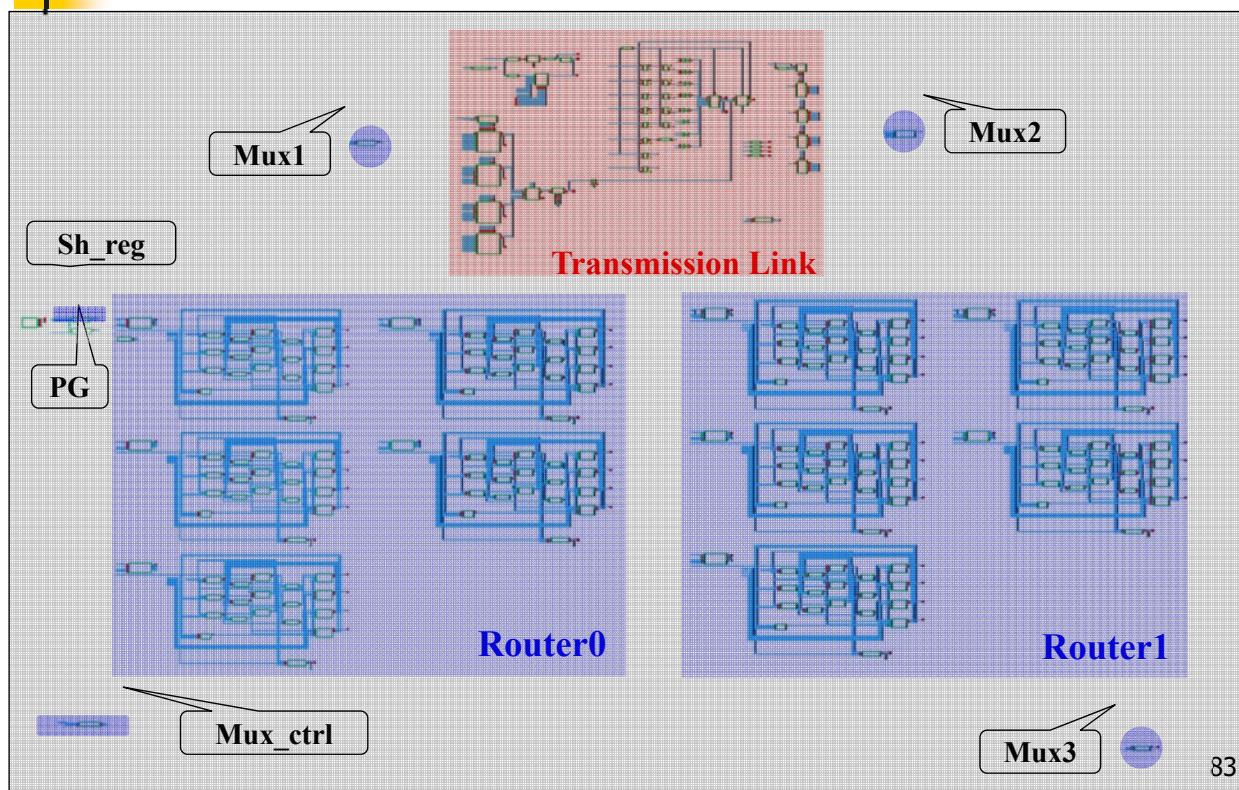
81

A/D Interface Element



82

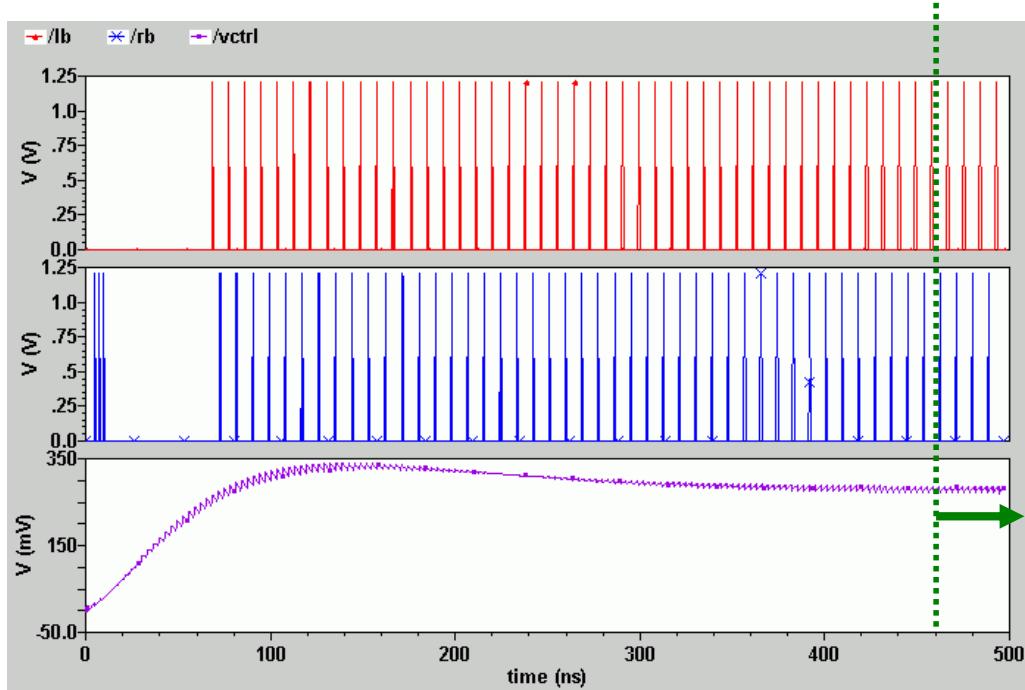
Whole System Co-Simulation



83

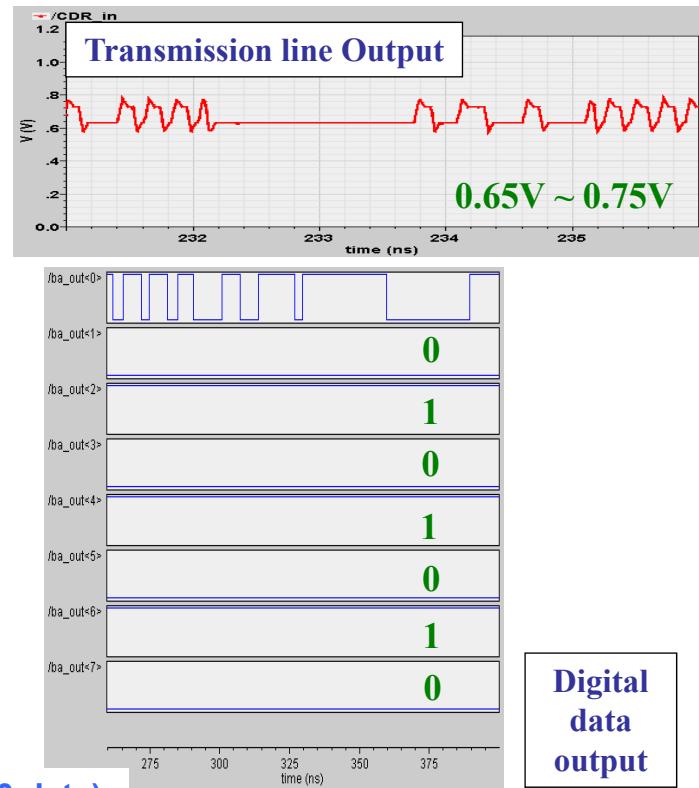
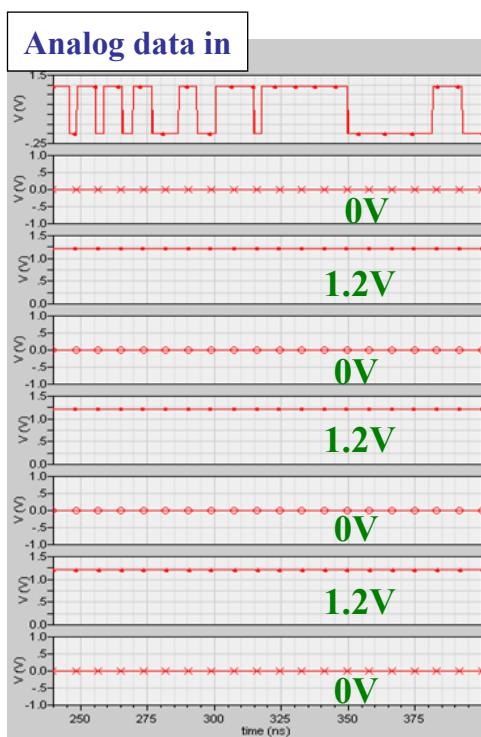
Mixed-Signal Simulation Results (1)

- First check PLL locked and CDR tracked



84

Mixed-Signal Simulation Results (2)



Simulation Time : 2727.10 sec. (5000 data)