وَمَا أُوتِيتُمْ مِنَ الْعِلْمِ إِلَّا قَلِيلًا

Ain Shams University – Faculty of Engineering – EECE Dept. – Integrated Circuits Lab.
Dr. Hesham Omran

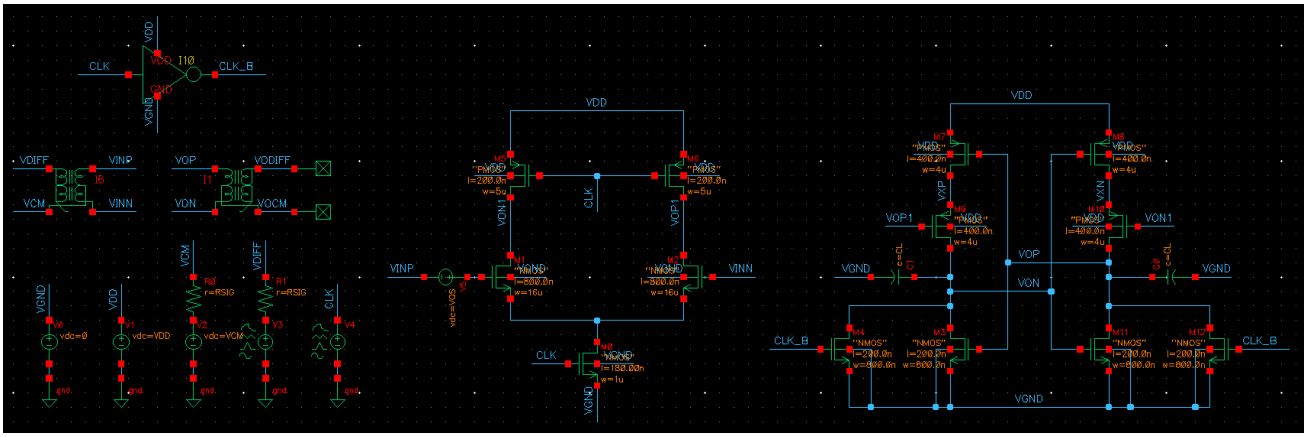
Analog Integrated System Design – Cadence Tools
Lab 07
Dynamic Comparator

Intended Learning Objectives

1) To be familiar with the operation of dynamic comparator.
2) To be familiar with the simulation and characterization of dynamic comparator.

PART 1: Basic Dynamic Comparator Simulation

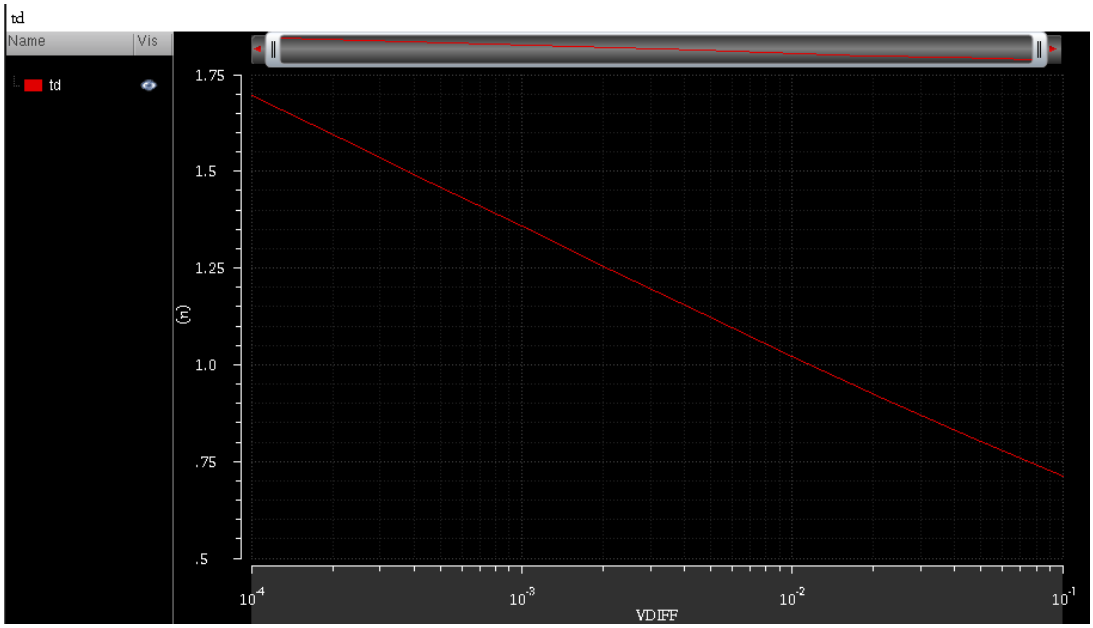
1) Carefully read the description of the dynamic comparator reported in the following paper:
   P. Harpe *et. al.*, A 26 uW 8 bit 10 MS/s asynchronous SAR ADC for low energy radios, Solid-State Circuits, IEEE Journal of, 2011, 46, 1585-1595
2) Create a testbench for the above-mentioned dynamic comparator. Scale the sizing of the transistors to meet your technology minimum channel length.



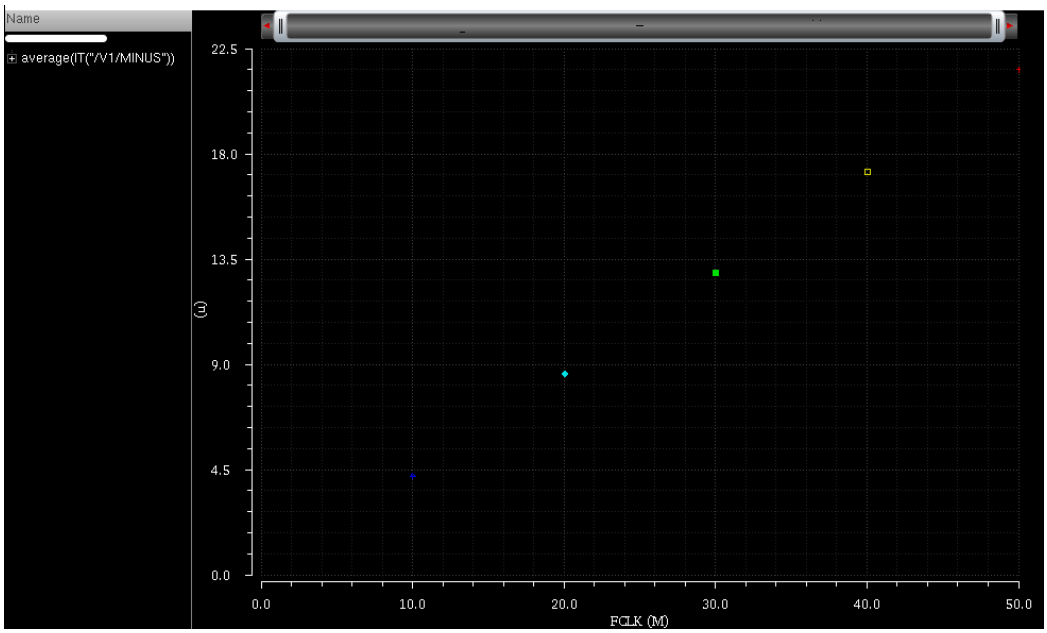
3) Set the clock as follows: one value (initial) = 0, zero value (pulse) = VDD, period = TCLK, rise/fall time = TRF, delay = TCLK/2, pulse width = TCLK/2. The remaining sources are simple DC sources.
4) The differential input signal is set as DC source (we will apply different stimuli in the next parts).
5) Set the global variables as below.

CL	50f
TCLK	20n
TRF	100p
RSIG	100
VOS	0
VDD	2
VCM	VDD/2
VDIFF	10m

6) Run transient simulation for 2*VAR(“TCLK”) conservative. Examine the transient waveforms at different points of the design.
7) Calculate the delay using the calculator.
```
delay(?wf1 VT("/CLK") ?value1 1.0 ?edge1 "rising" ?nth1 1 ?td1 0.0 ?wf2 abs(VT("/VODIFF"))
?value2 1.0 ?edge2 "either" ?nth2 1 ?td2 nil ?stop nil ?multiple nil)
```
8) Plot the power consumption vs time. Calculate the average power consumption using the calculator.
```
VAR(“VDD”)*average(IT("/V1/MINUS"))
```
9) Set a logarithmic parametric sweep for the differential input voltage. Set VDIFF = 100uV to 100mV, 10 steps, logarithmic.
10) Run transient analysis and plot the delay vs VDIFF. Use log axis for VDIFF. Comment on the results.



11) Set VDIFF to its original value. Set TCLK = 1/FCLK and set a linear sweep for FCLK 10M:10M:50M. Plot the average power consumption vs FCLK. Comment on the results.

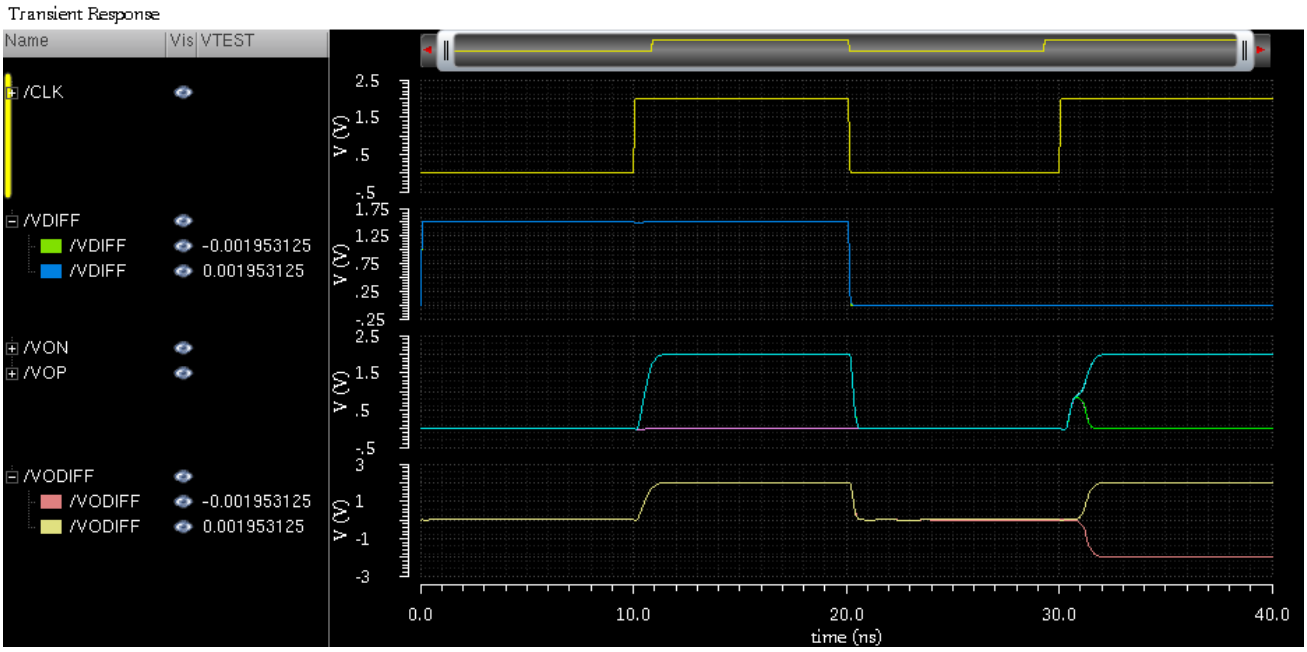


PART 2: Overdrive (Overload) Recovery Test

1) We will apply small input (±0.5 LSB) is applied to the comparator input in a cycle right after a full-scale input is applied; the comparator should be able to resolve to the right output in either case → memoryless.
1) Change the differential input to be a pulse source with the following parameters: one value (initial) = VREFP, zero value (pulse) = VTEST, period = 2*TCLK, rise/fall time = TRF, delay = 0, pulse width = TCLK.
2) Assume the comparator will be used in an 8-bit ADC. Set global parameters as shown below.

CL	50f
TCLK	20n
TRF	100p
RSIG	100
VOS	0
VDD	2
VCM	0.5*(VREFP + VREFN)
VDIFF	10m
VREFP	0.75*VDD
VREFN	0.25*VDD
NBIT	8
VLSB	(VREFP – VREFN) / 2**NBIT
VTEST	VLSB/2, -VLSB/2

3) Run transient analysis and plot the output for the two extreme cases.



PART 3: Offset Voltage Simulation

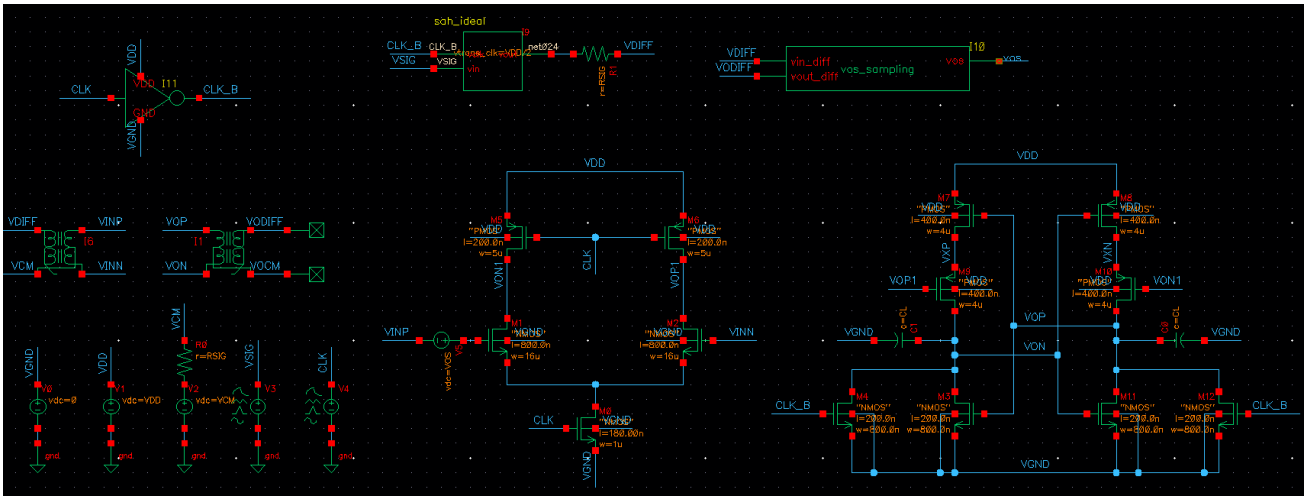
1) We will create two Verilog-A models to help in offset voltage simulation[1]. Read and analyze the code below. The first model is an ideal sample and hold. The second model samples the offset voltage and stops the simulation afterwards.

```
`include "constants.vams"
`include "disciplines.vams"
// sah_ideal
// ideal sample and hold amplifier
//
module sah_ideal(vin, vout, vclk);
input vin, vclk;
output vout;
electrical vin, vout, vclk;
parameter real vth = 1;       // set it to VDD/2
real vout_val;

        analog begin
                @ (cross(V(vclk) - vth, 1.0) or initial_step)
                        vout_val = V(vin);
                V(vout) <+ vout_val ;
        end
endmodule
```

```
`include "constants.vams"
`include "disciplines.vams"

module vos_sampling(vin_diff,vout_diff,vos);
input vin_diff, vout_diff;
output vos;
electrical vin_diff, vout_diff, vos;

parameter real TCLK = 20n;
parameter real VDD = 2.0;

// real variables are automatically initialized to zero
real vin_diff_delayed;
real vos_val;
integer done;

        analog begin

                @(initial_step)  done = 0;

                vin_diff_delayed = absdelay(V(vin_diff), TCLK/4);

                // Do not use the `cross' analog operator in a conditionally-executed statement or
in an expression
                @ (cross(V(vout_diff) - VDD/2, +1)) begin
                        if (done == 0) begin
                                vos_val = -vin_diff_delayed;
                                done = 1;
                        end
                        else begin
                                done = 2;
                        end
                end
                V(vos) <+ vos_val ;
                if (done == 2) $finish_current_analysis;
        end
endmodule
```

[1] A fast and accurate technique for comparator offset voltage simulation is reported here: Hesham Omran, Fast and accurate technique for comparator offset voltage simulation, Microelectronics Journal, Volume 89, 2019, Pages 91-97.

2) Copy the testbench to a new one and modify it as below. Practically Monte Carlo simulation should be used to simulate the offset voltage. But we artificially add an offset voltage using vdc source as a simple test case.



3) Set the input signal to be a PWL source: (0, -VDIFF) (NPTS*TCLK, VDIFF)
4) Set global variables as shown below. Set the properties of the Verilog-A blocks as a function of the global variables.

CL	50f
TCLK	20n
TRF	100p
RSIG	100
VOS	5.4m
VDD	2
VCM	VDD/2
VDIFF	10m
NPTS	100

5) Run conservative transient analysis with tstop = VAR("NPTS")*VAR("TCLK").
6) Use the following expression in the calculator to evaluate the offset voltage.
`value(VT("/vos") ymax(xval(VT("/vos"))))`
7) Compare the evaluated offset with the value added in the global variables.