EE288 – Final Project

10 bit – 1.5 bit/stage Pipeline ADC

(using 45nm CMOS Technology)

Muhammad Aldacher

Enrique Hernandez

Overview

1) Project Target
2) Circuits from HW6
   - Comparator
   - Sub-ADC Logic
   - Switches
3) OpAmp Design for the MDAC
   - Circuit
   - AC responses
4) Whole System simulations
   - Normal Mode
   - 2x Gain Mode
5) Corner Simulations
6) VerilogA Blocks

(1)
Project Target

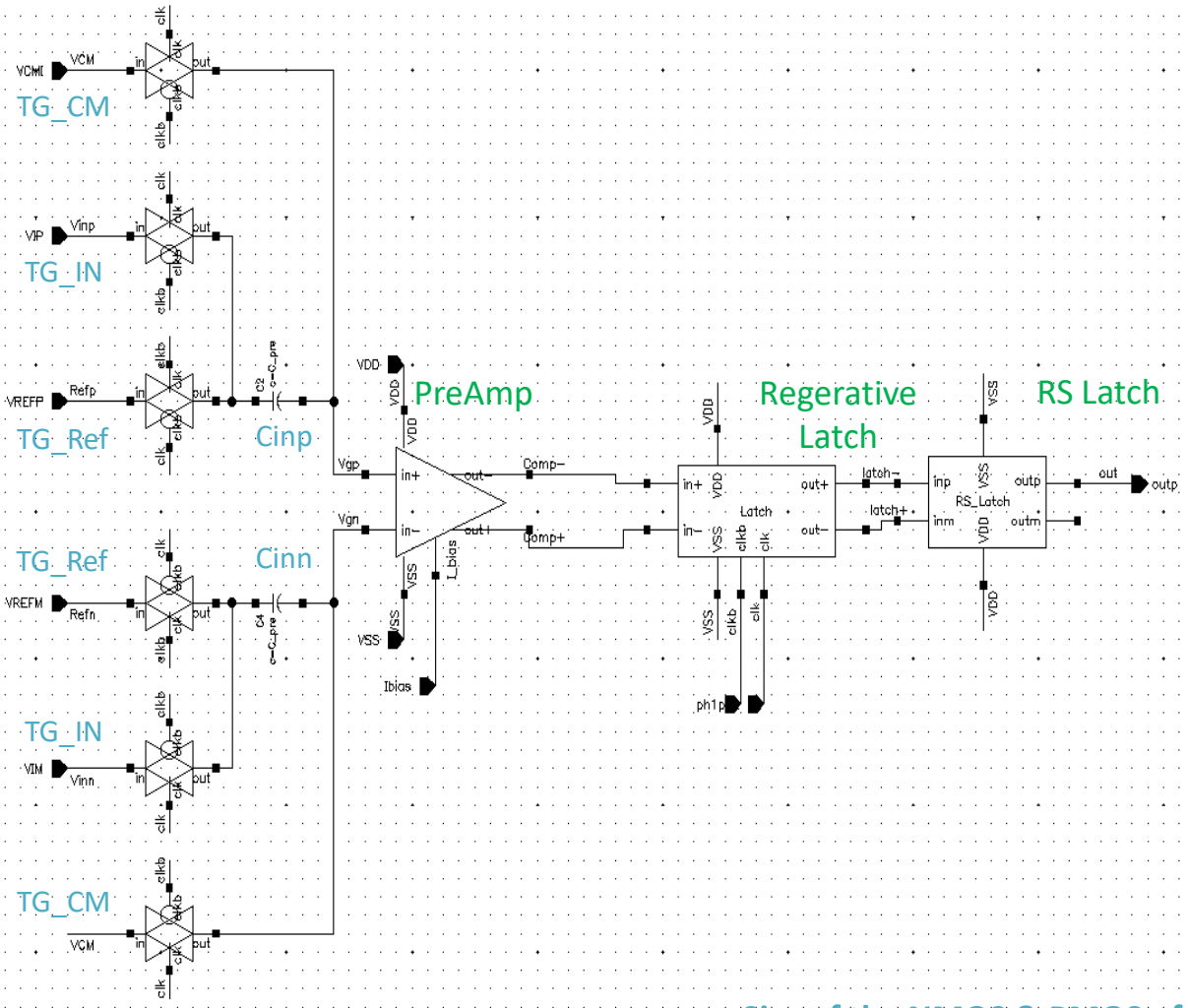
Target

Typical corner (TT, 1.8V, 27C)
Vin = 0.8Vpp (Full scale signal)
fin = (cycles/N) * fs where cycles=7 and N=64
Fs = 50 MSPS
Real switches & Real opamp
ideal_clock, ideal_bias, and ideal voltages
Target: 10-bit ENOB at 2x gain mode with Vin=0.8Vpp

Worst case ENOB > 7-bit for 2x gain mode, Vin=0.8Vpp
TT, 1.8V, 0C, 27C, 70C
TT, 27C, 1.7V, 1.8V, 1.9V
27C, 1.8V, TT, FF, SS
SS, 1.7V, 70C

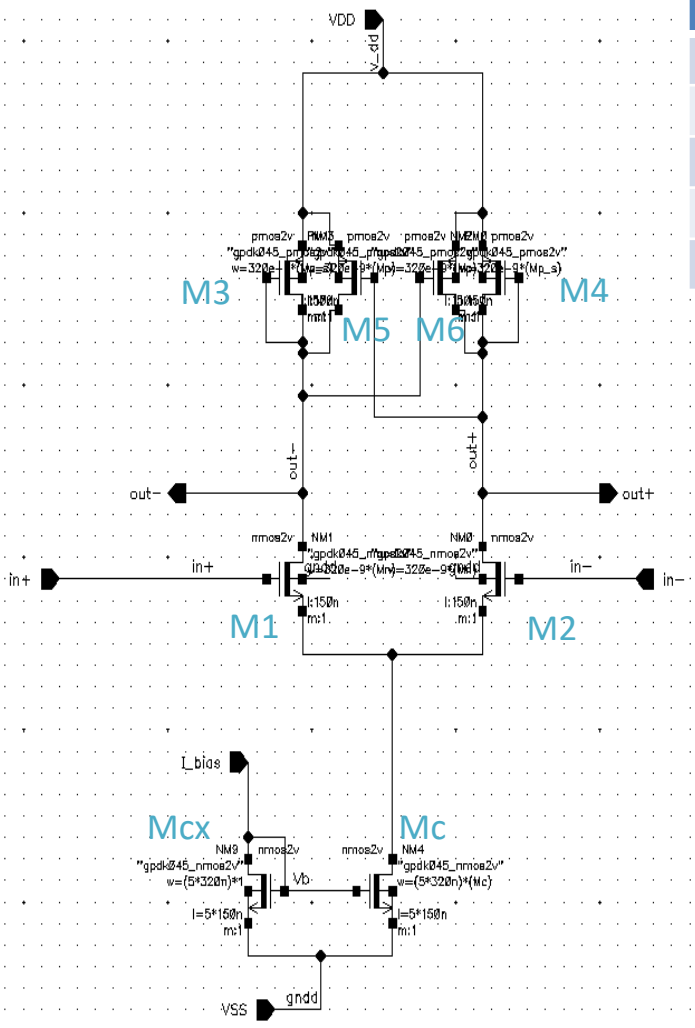
(2)
Circuits from HW6

A- Comparator



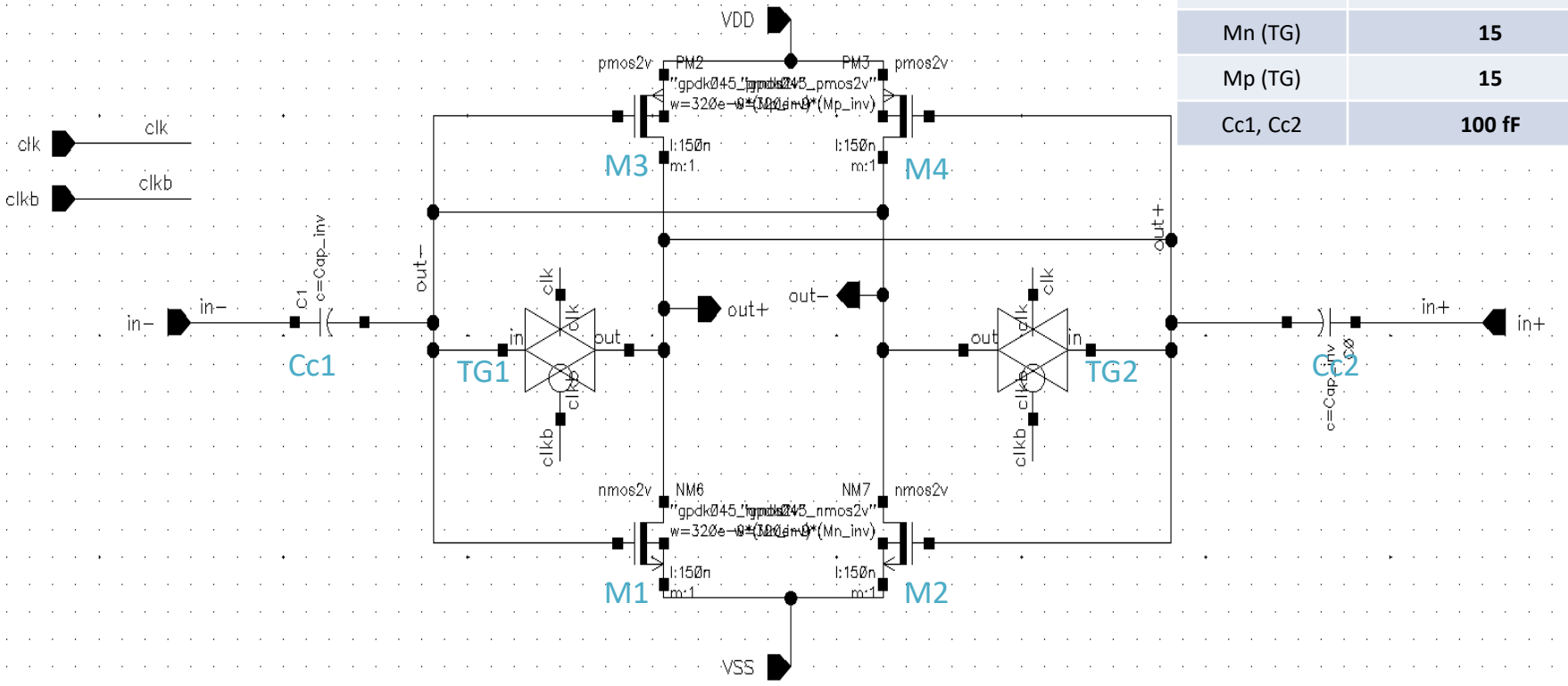
Size of the NMOS & PMOS of the TGs here (m) = 3

PreAmp



Device	Size (in terms of fingers, m)
M1, M2	**5**
M3, M4, M5, M6	**1**
Mc	**2**
Mcx	**1**
Ideal Current Source	**8.8 uA**

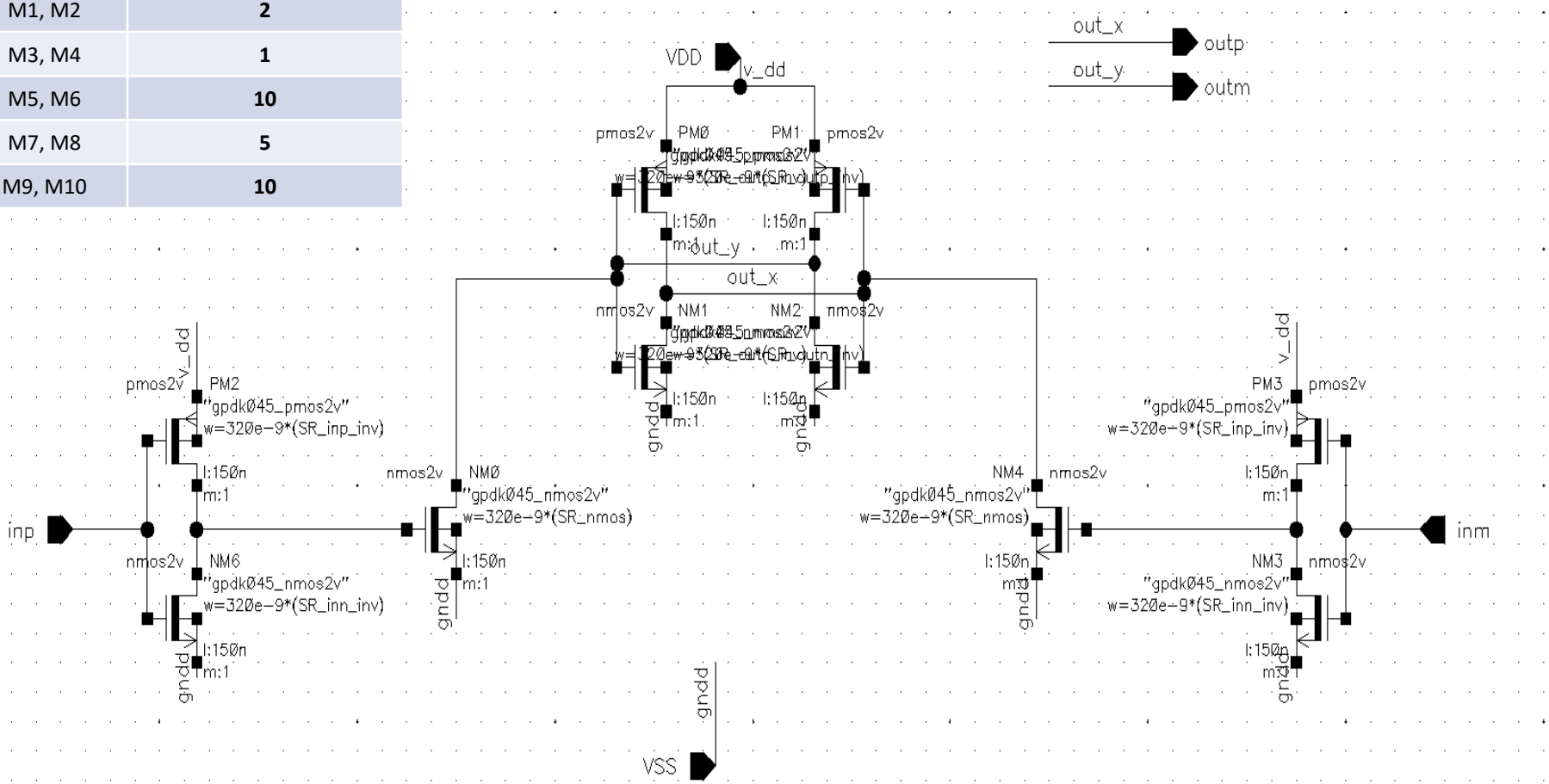
Regenerative Latch



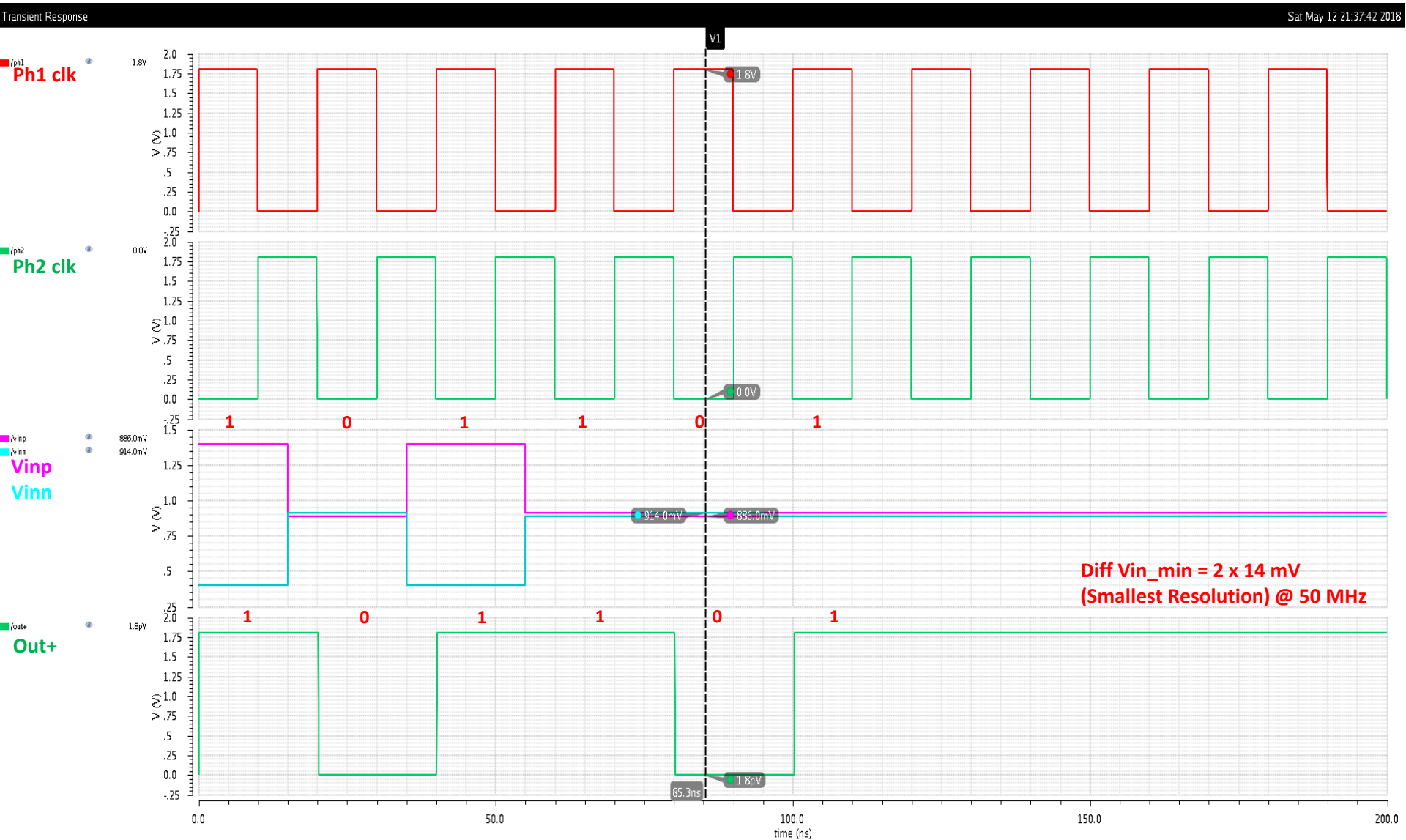
Device	Size (in terms of fingers, m)
M1, M2	**5**
M3, M4	**10**
Mn (TG)	**15**
Mp (TG)	**15**
Cc1, Cc2	**100 fF**

RS Latch

Device	Size (in terms of fingers, m)
M1, M2	**2**
M3, M4	**1**
M5, M6	**10**
M7, M8	**5**
M9, M10	**10**

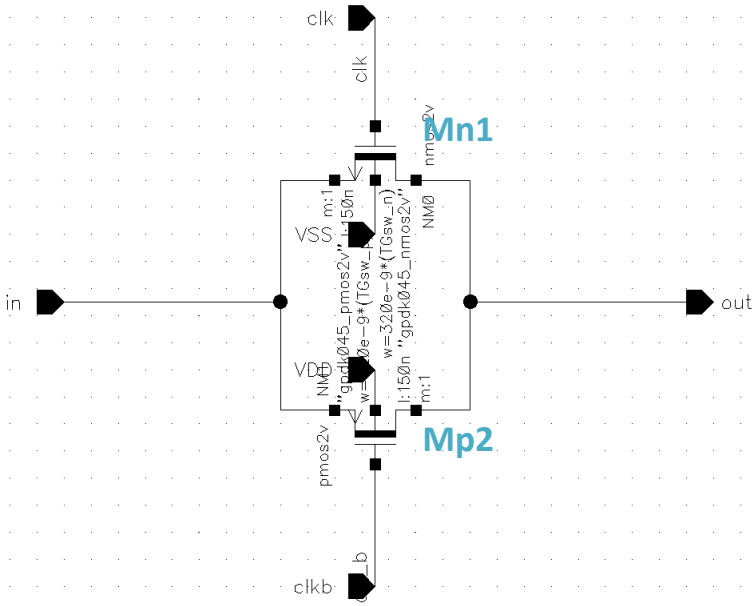


Overdrive test Waveforms



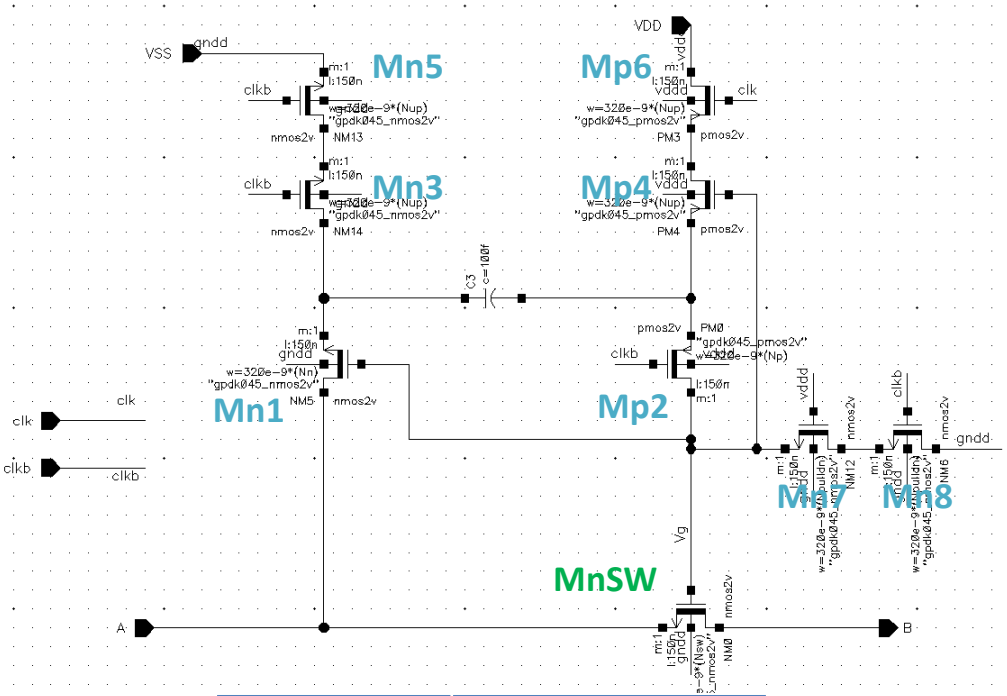
B- Switches

Transmission Gate



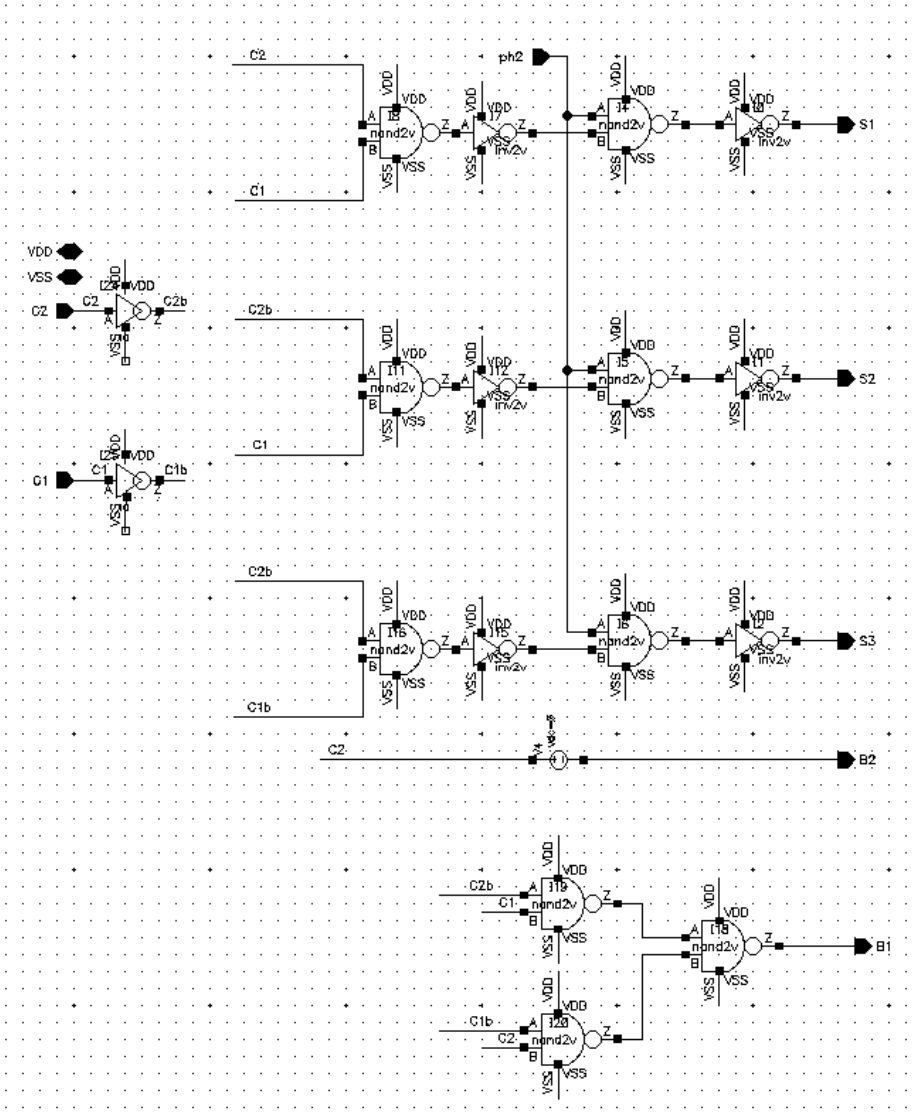
Device	Size (in terms of fingers, m)
Mn1	**7**
Mp2	**7**

Bootstrap Switch



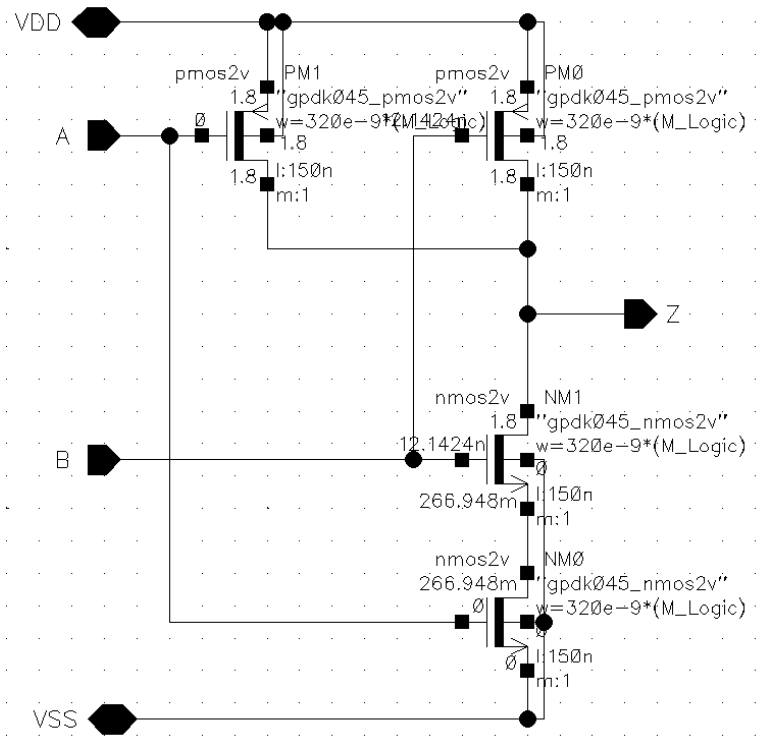
Device	Size (in terms of fingers, m)
MnSW	**30**
Mn1	**20**
Mp2	**1**
Mn3,Mn5 Mp4,Mp6	**1**
Mn7,Mn8	**28**

C- Sub ADC Logic

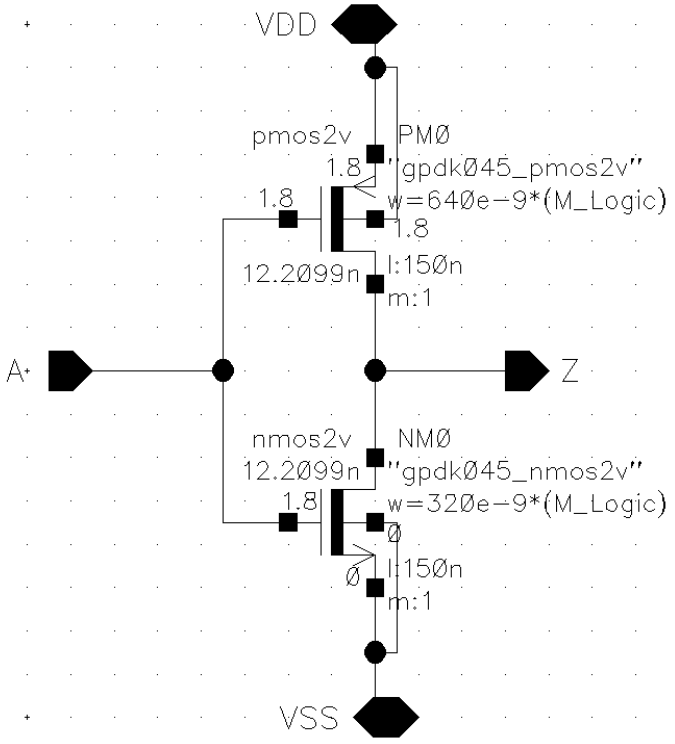


Logic Circuits used

NAND Gate Circuit



Inverter Circuit



M_Logic = 1

(3)
OpAmp Design

Target

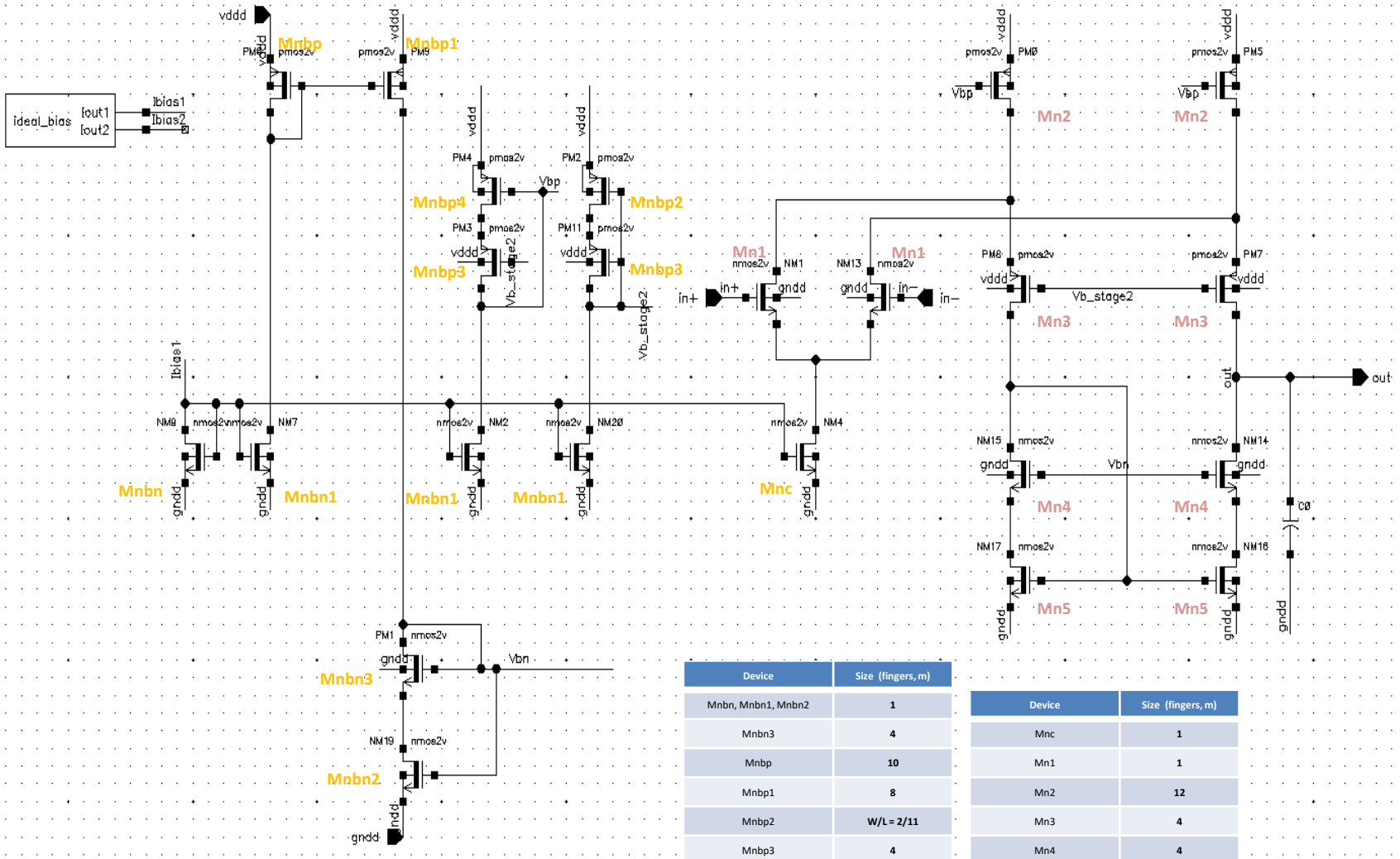
- For the Gain Error < ¼ LSB
  (Gain Requirement):

Resolution	Full Scale	Beta	Gain (dB)	Gain (dB)
N	V_{FS} (Volt)	β	$2 \cdot 2^{(N)} / (\beta \cdot V_{FS})$	20log(x)
10	0.8	1	5120	74
10	0.8	0.5	10240	80
11	0.8	1	10240	80
11	0.8	0.5	20480	86

- For the Settling Error < ½ LSB
  (Unity BW Requirement):

Resolution	Full Scale	Beta	Sampling Rate	UGB
N	V_{FS} (Volt)	β	fs	fu
10	0.8	1	5.00E+07	1.25E+08
10	0.8	0.5	5.00E+07	2.50E+08
11	0.8	1	5.00E+07	1.36E+08
11	0.8	0.5	5.00E+07	2.72E+08

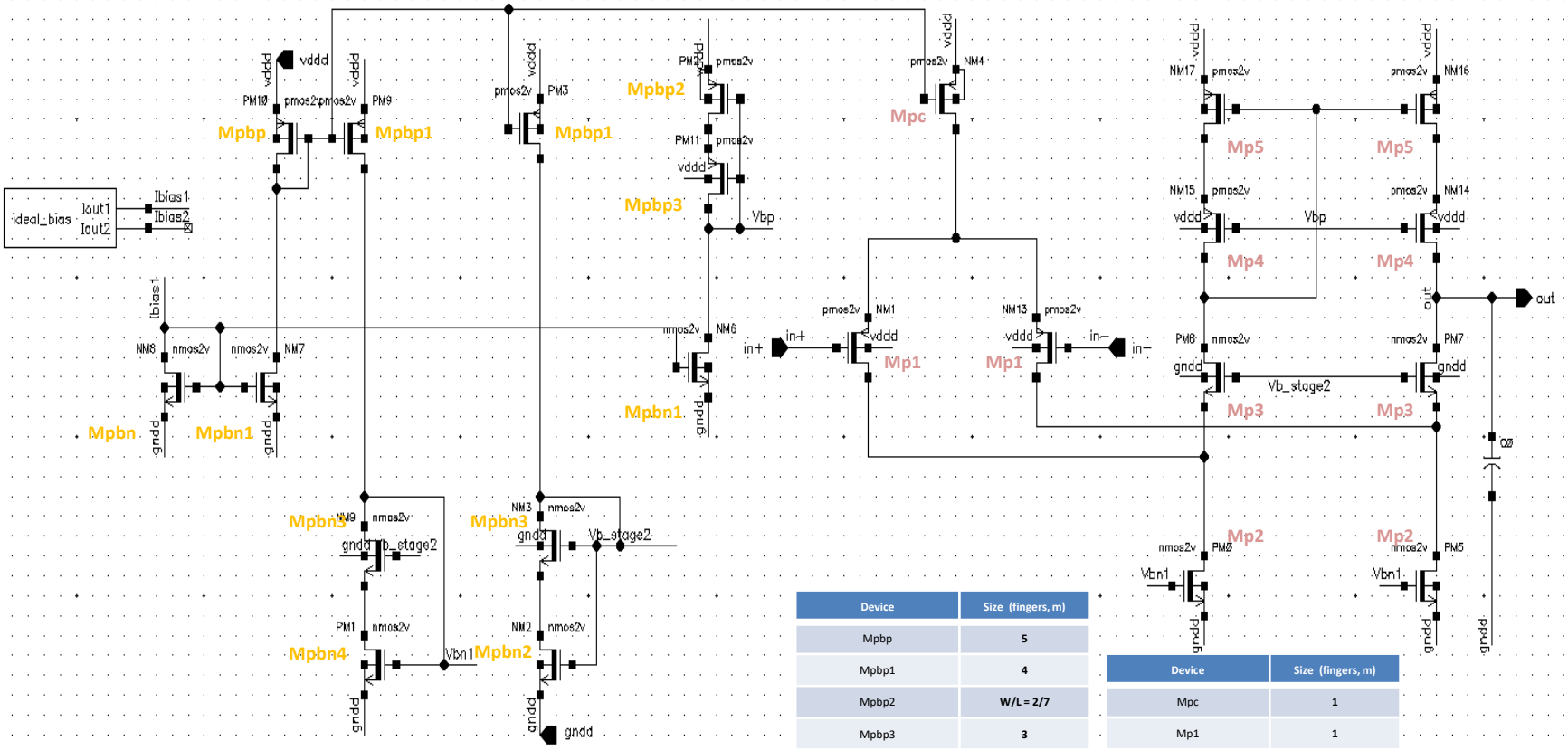
NMOS-input, Single-Ended Output, Folded-Cascode OpAmp



Device	Size (fingers, m)
Mnbn, Mnbn1, Mnbn2	1
Mnbn3	4
Mnbp	10
Mnbp1	8
Mnbp2	W/L = 2/11
Mnbp3	4
Mnbp4	31

Device	Size (fingers, m)
Mnc	1
Mn1	1
Mn2	12
Mn3	4
Mn4	4
Mn5	5

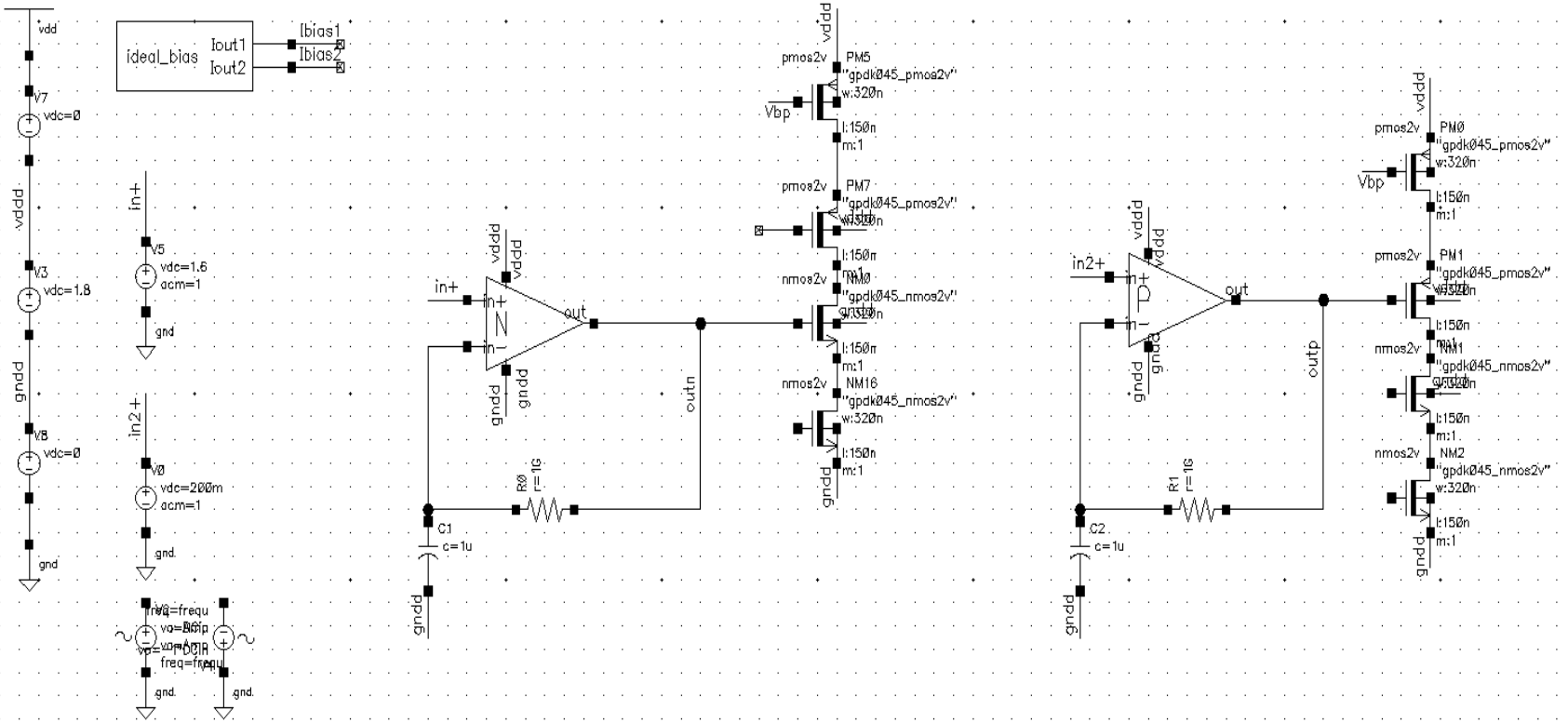
PMOS-input, Single-Ended Output, Folded-Cascode OpAmp



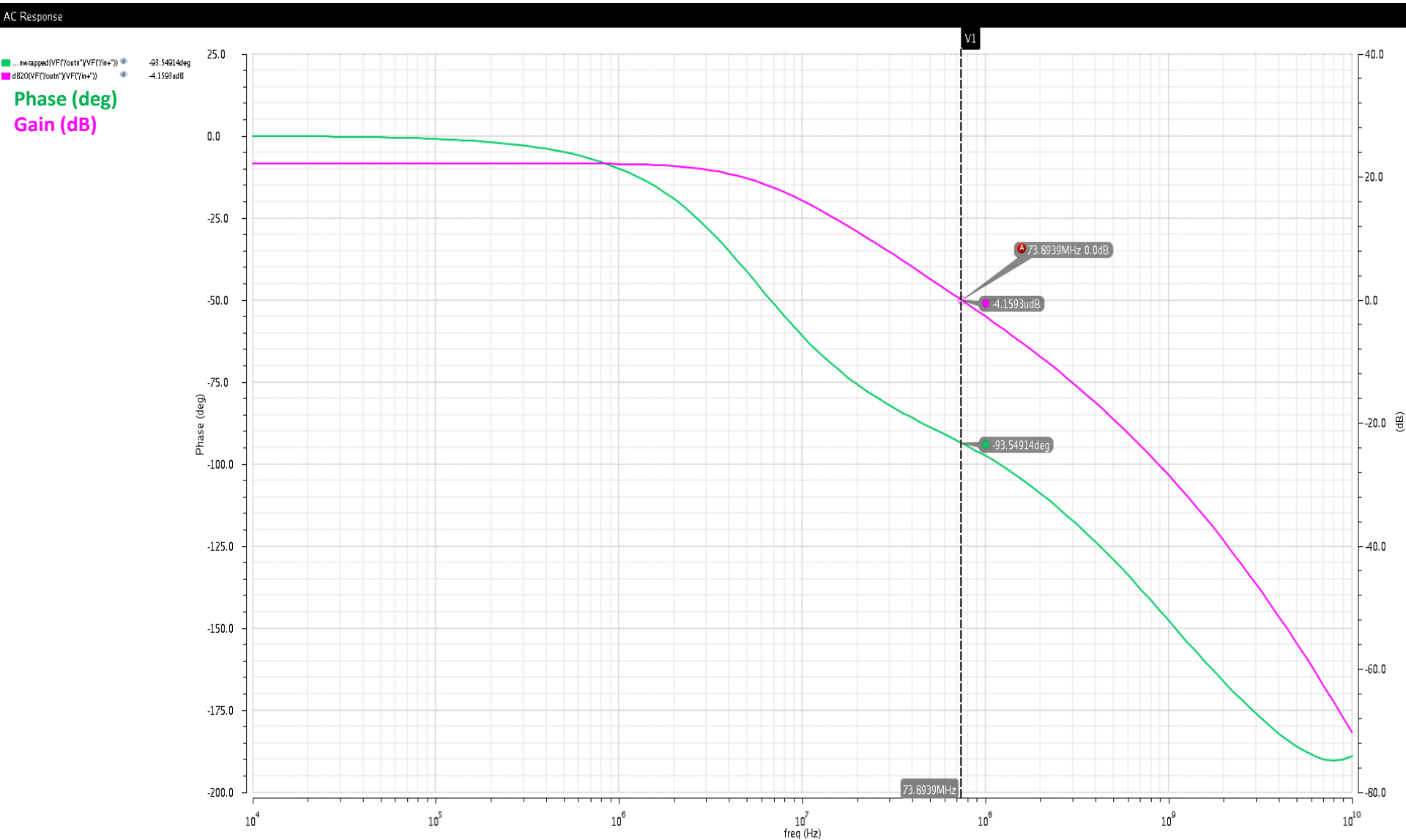
Device	Size (fingers, m)
Mpbp	5
Mpbp1	4
Mpbp2	W/L = 2/7
Mpbp3	3
Mpbn, Mpbn1	1
Mpbn2	1
Mpbn3	2
Mpbn4	20

Device	Size (fingers, m)
Mpc	1
Mp1	1
Mp2	4
Mp3	2
Mp4	3
Mp5	5

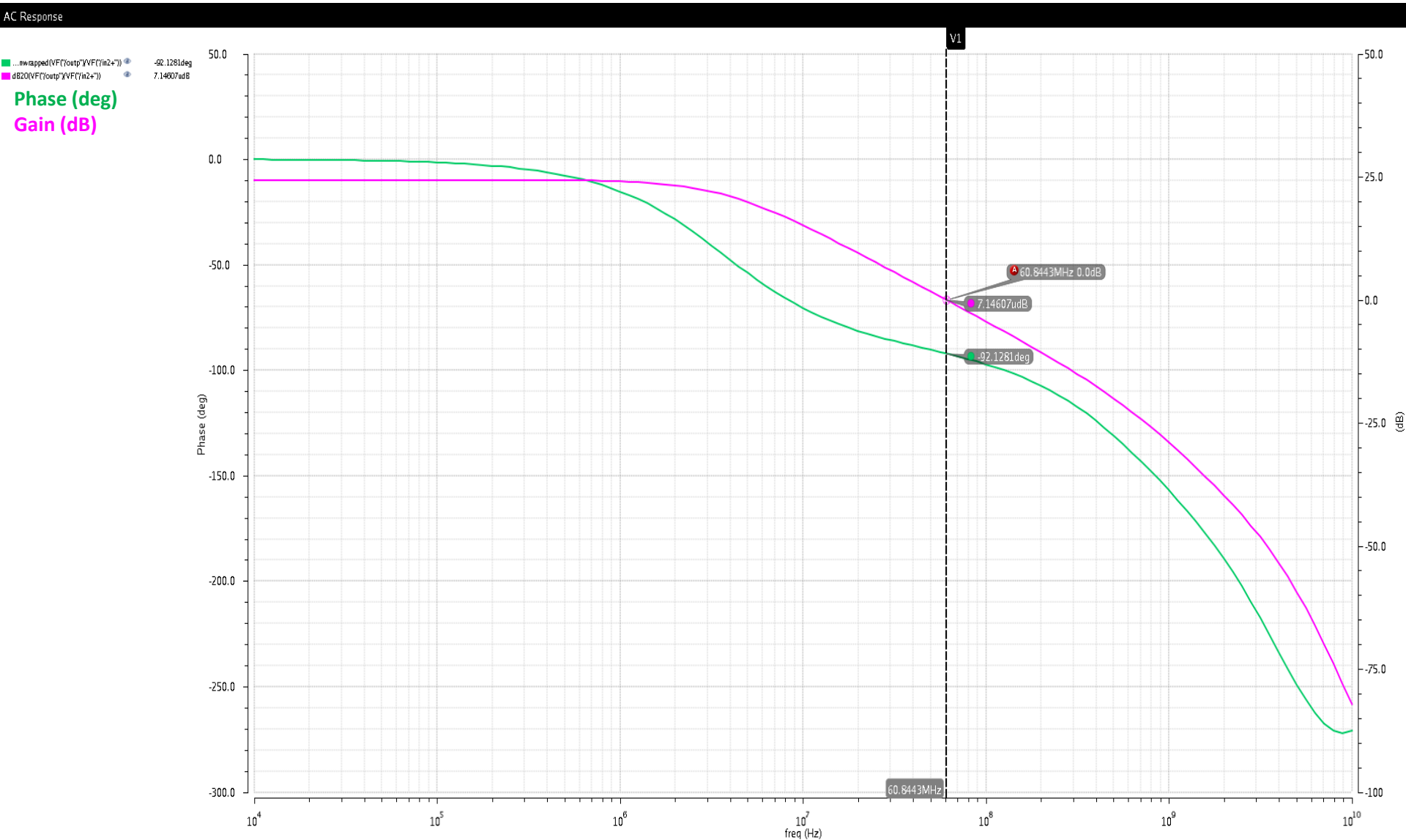
Testbench for the Single-Ended-Output OpAmps



AC Response (NMOS)



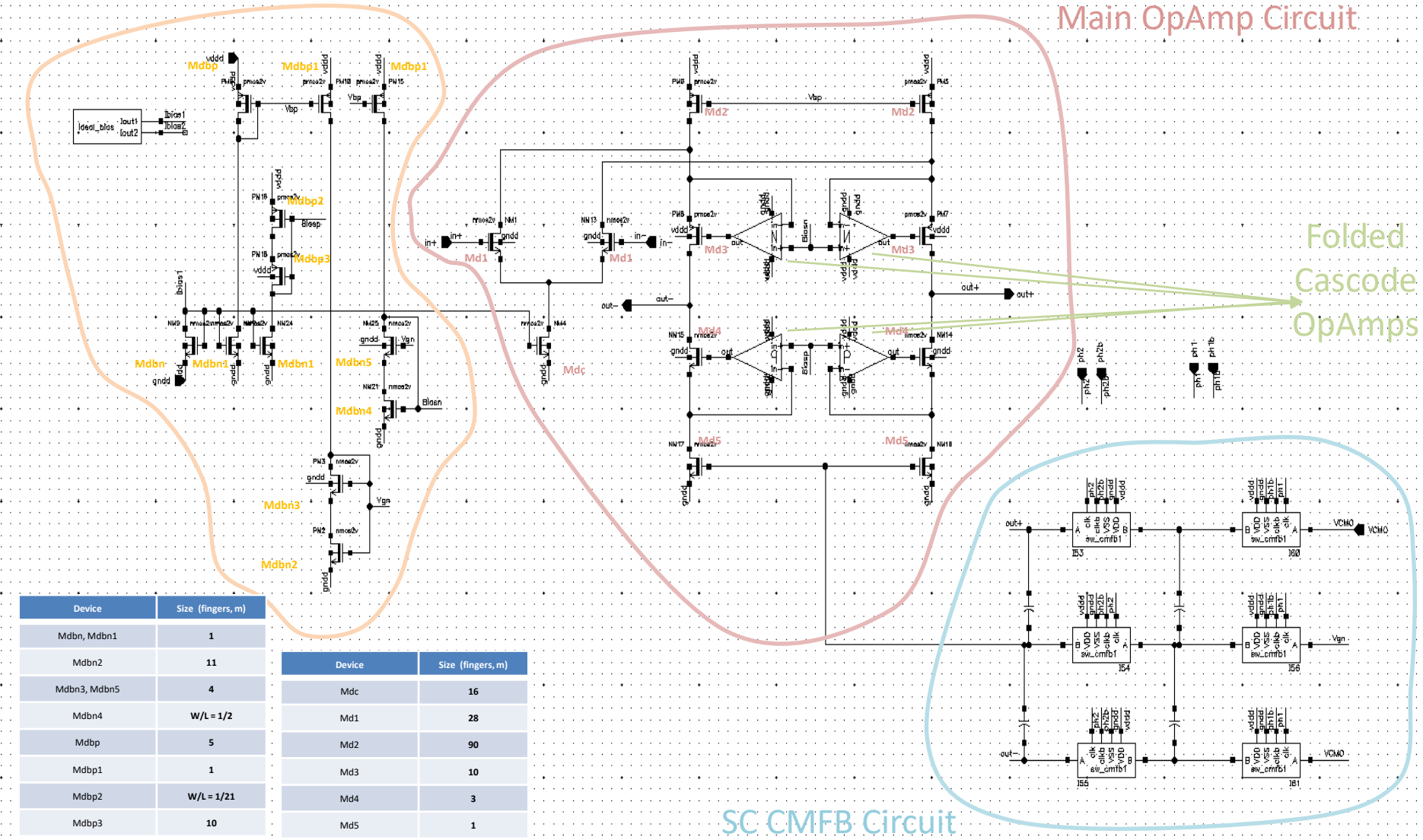
AC Response (PMOS)



Fully-Differential Gain-Boosted Telescopic OpAmp

Biasing Circuits

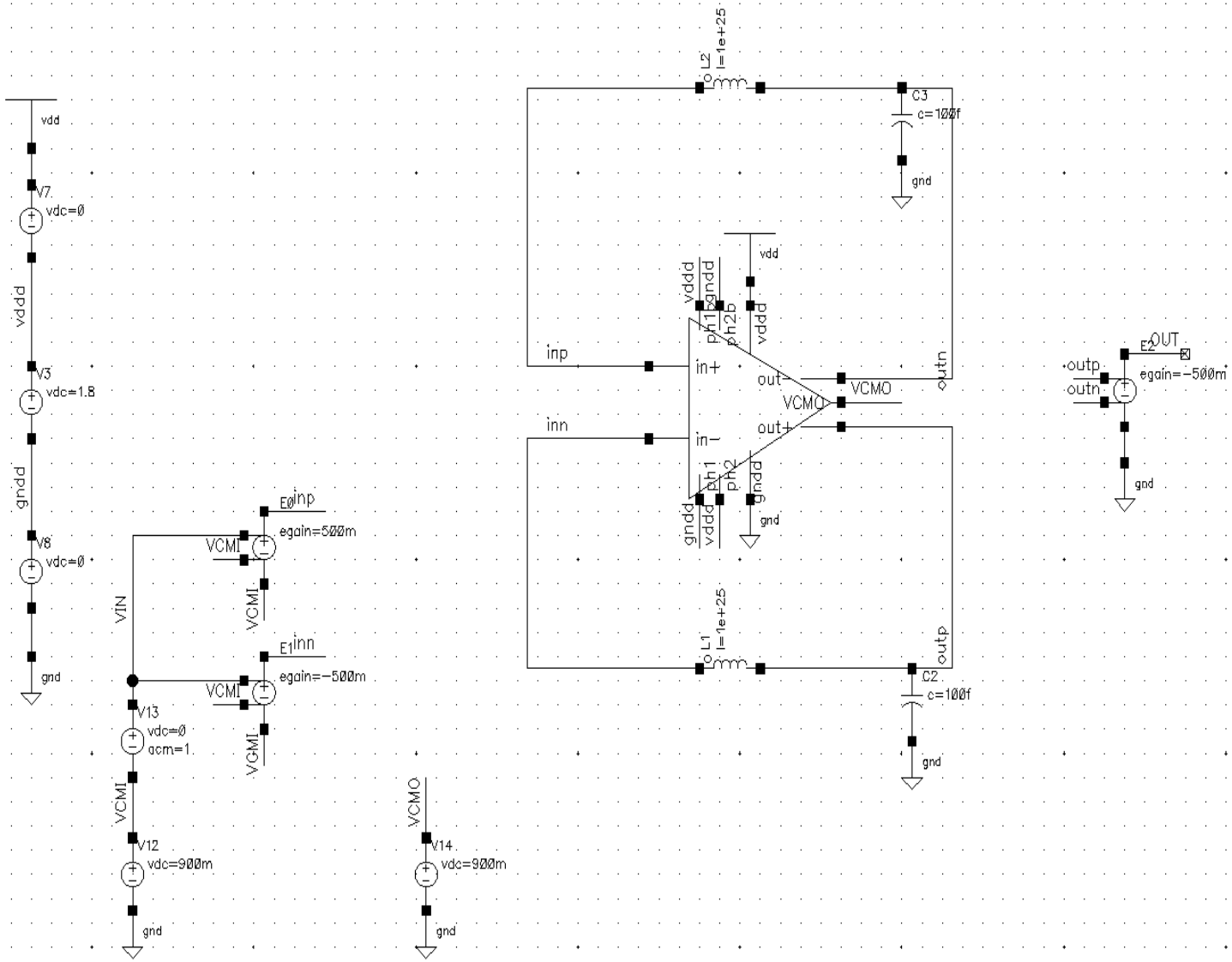
Main OpAmp Circuit



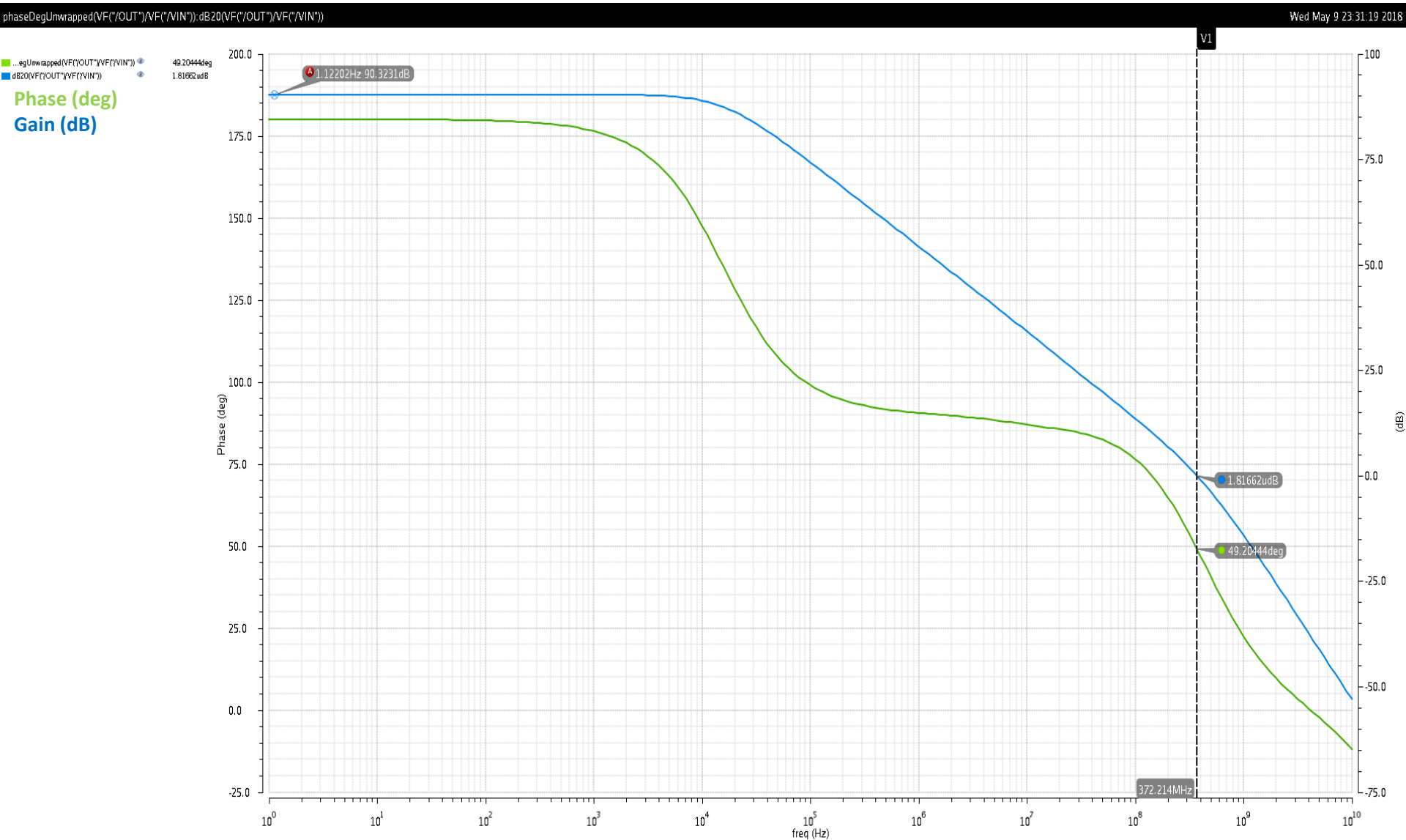
Device	Size (fingers, m)
Mdbn, Mdbn1	1
Mdbn2	11
Mdbn3, Mdbn5	4
Mdbn4	W/L = 1/2
Mdbp	5
Mdbp1	1
Mdbp2	W/L = 1/21
Mdbp3	10

Device	Size (fingers, m)
Mdc	16
Md1	28
Md2	90
Md3	10
Md4	3
Md5	1

Testbench for the Boosted-Gain OpAmp



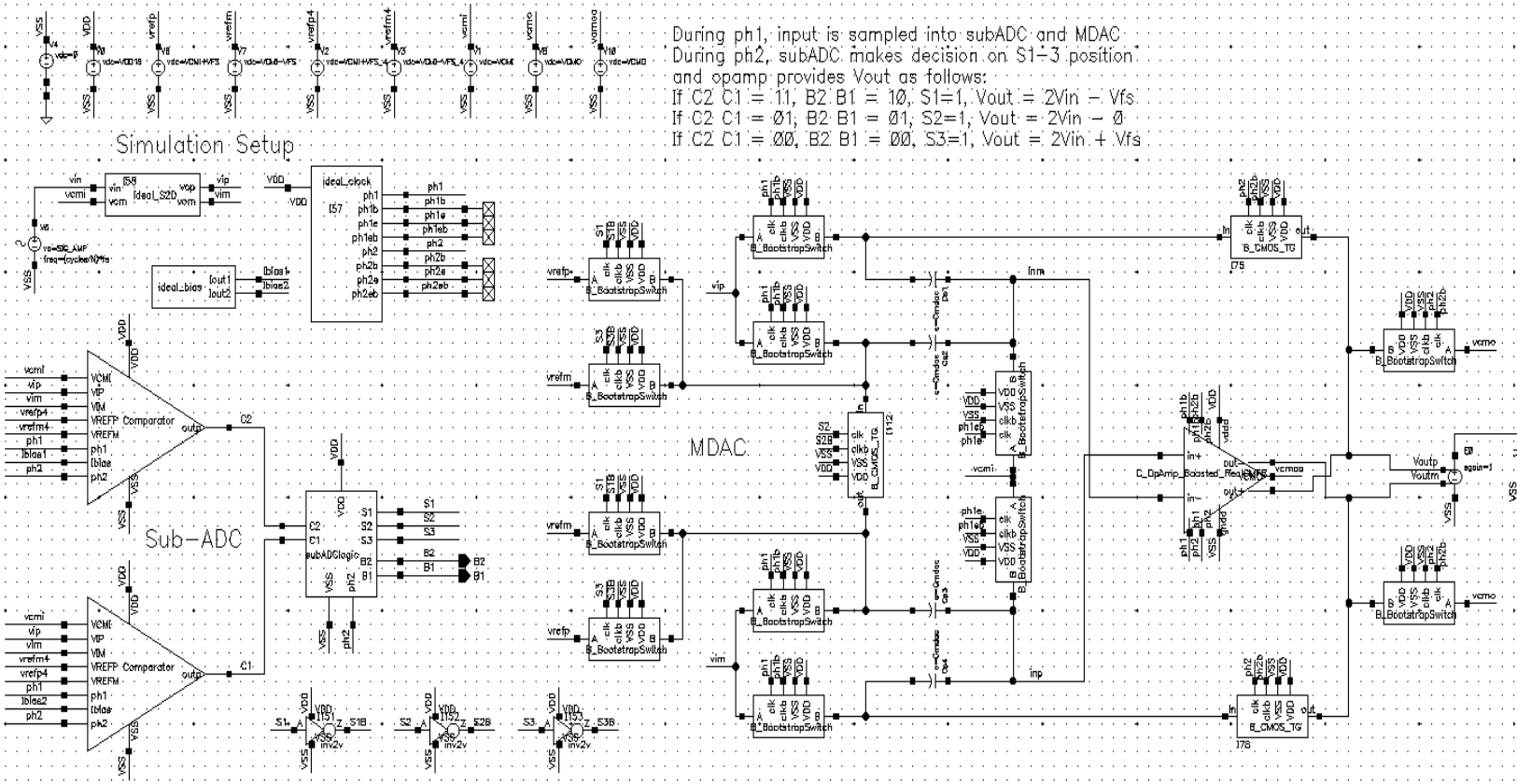
AC Response


phaseDegUnwrapped(VF("/OUT")/VF("/VIN")):dB20(VF("/OUT")/VF("/VIN"))
Wed May 9 23:31:19 2018
...egUnwrapped(VF("/OUT")/VF("/VIN")) 49.20444deg
dB20(VF("/OUT")/VF("/VIN")) 1.81662udB
Phase (deg)
Gain (dB)
V1
1.12202Hz 90.3231dB
1.81662udB
49.20444deg
372.214MHz
Phase (deg)
freq (Hz)
(dB)
200.0
175.0
150.0
125.0
100.0
75.0
50.0
25.0
0.0
-25.0
100
75.0
50.0
25.0
0.0
-25.0
-50.0
-75.0
10^0
10^1
10^2
10^3
10^4
10^5
10^6
10^7
10^8
10^9
10^{10}

(4)
System Simulations

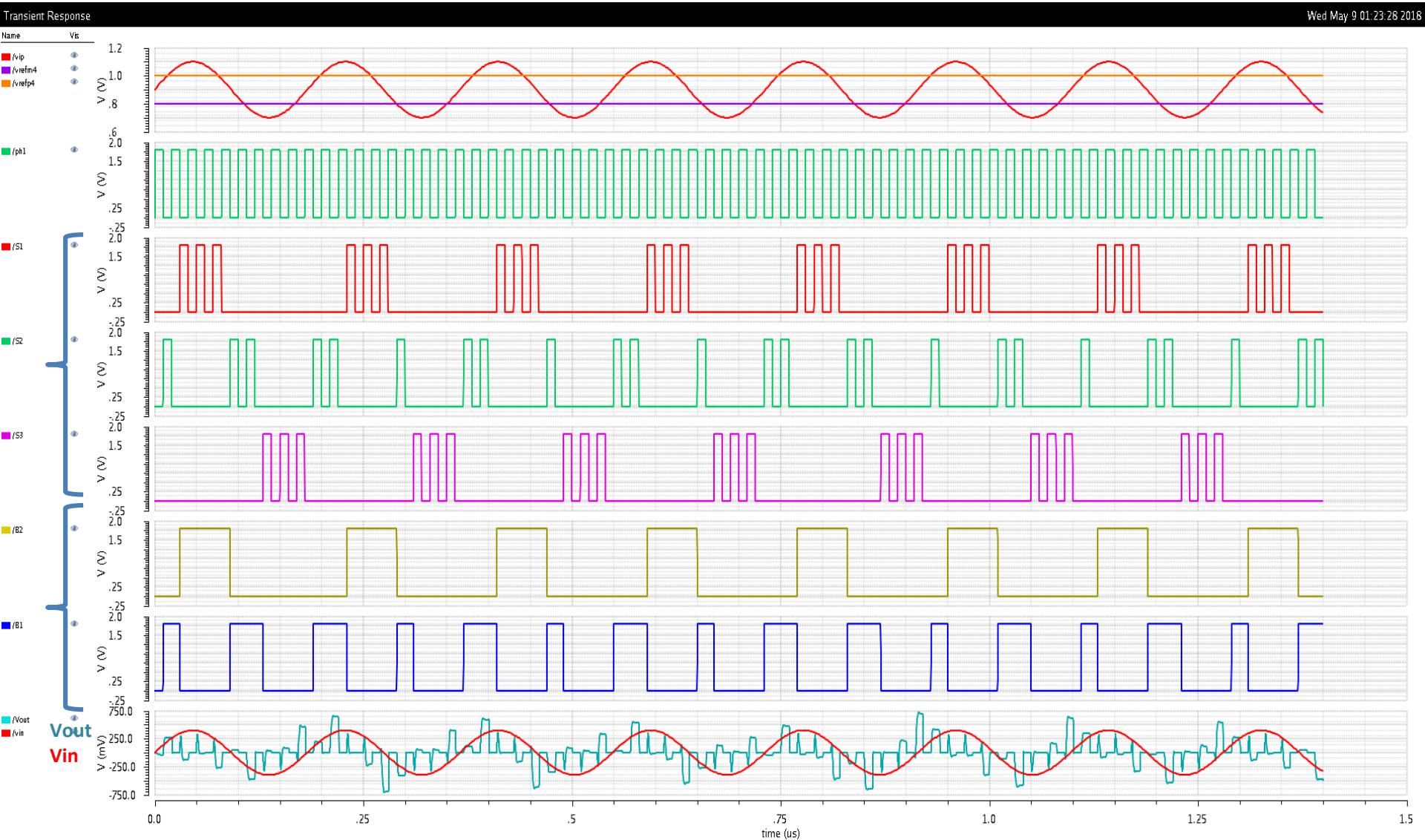
(Under Typical Conditions: TT, 27°, 1.8VDD, Signal Amp = 0.4v)

Testbench for the 1 stage 1.5bit/stage ADC



Simulation Results

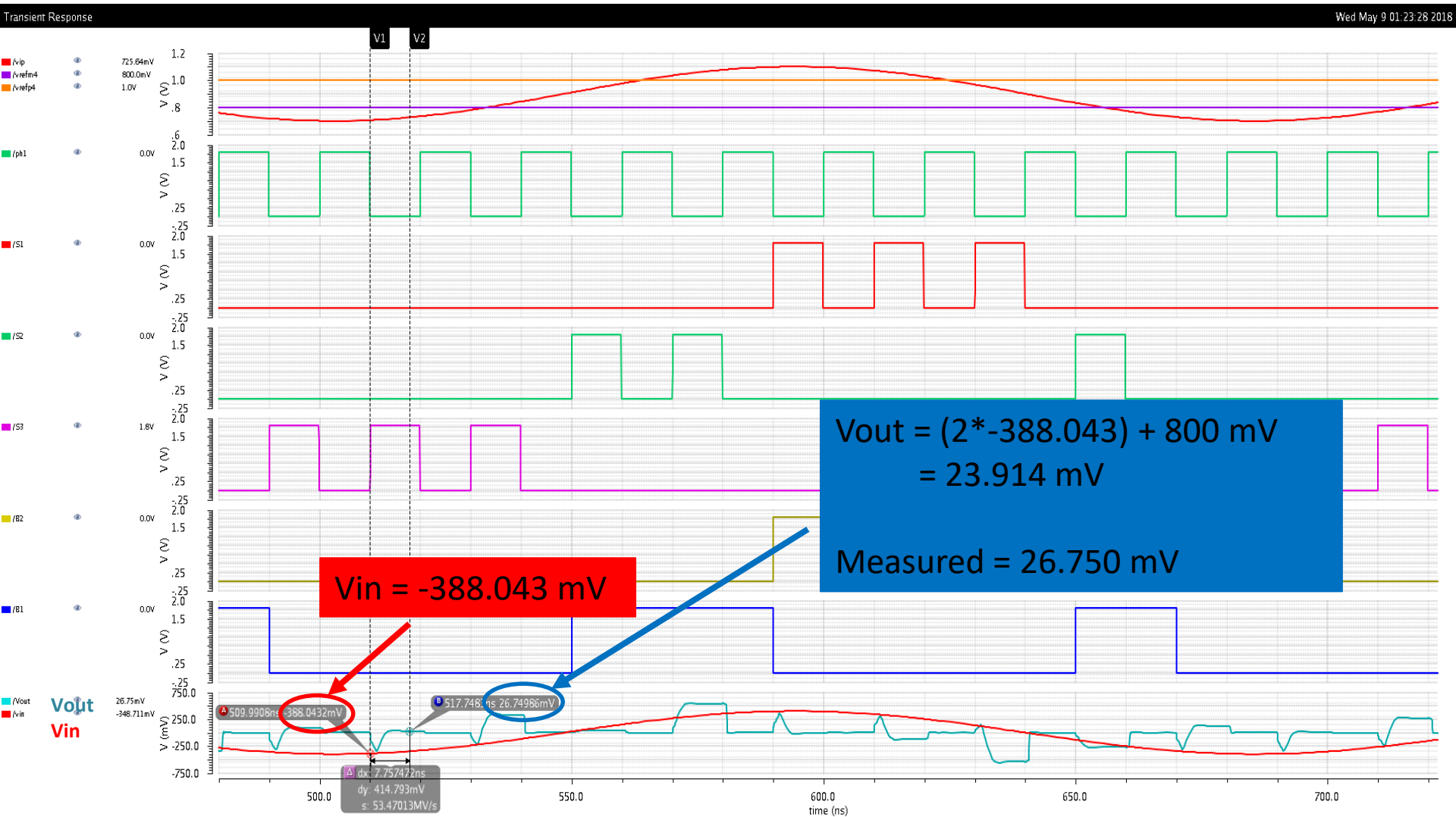
1- Normal Mode



Simulation Results

1- Normal Mode

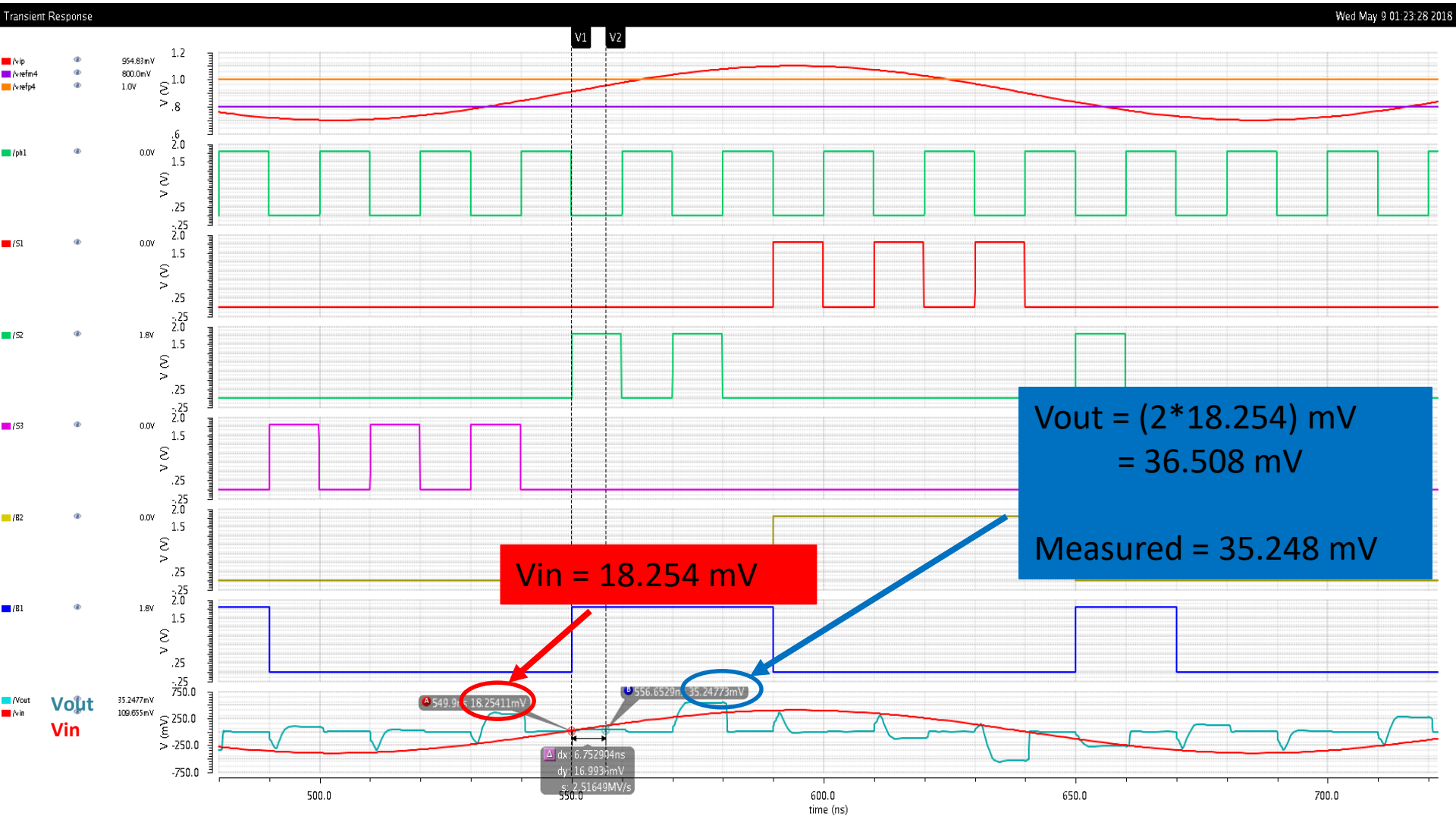
1) If Vin < Vrefm ➔ Vout = 2*Vin + Vfs



Simulation Results

1- Normal Mode

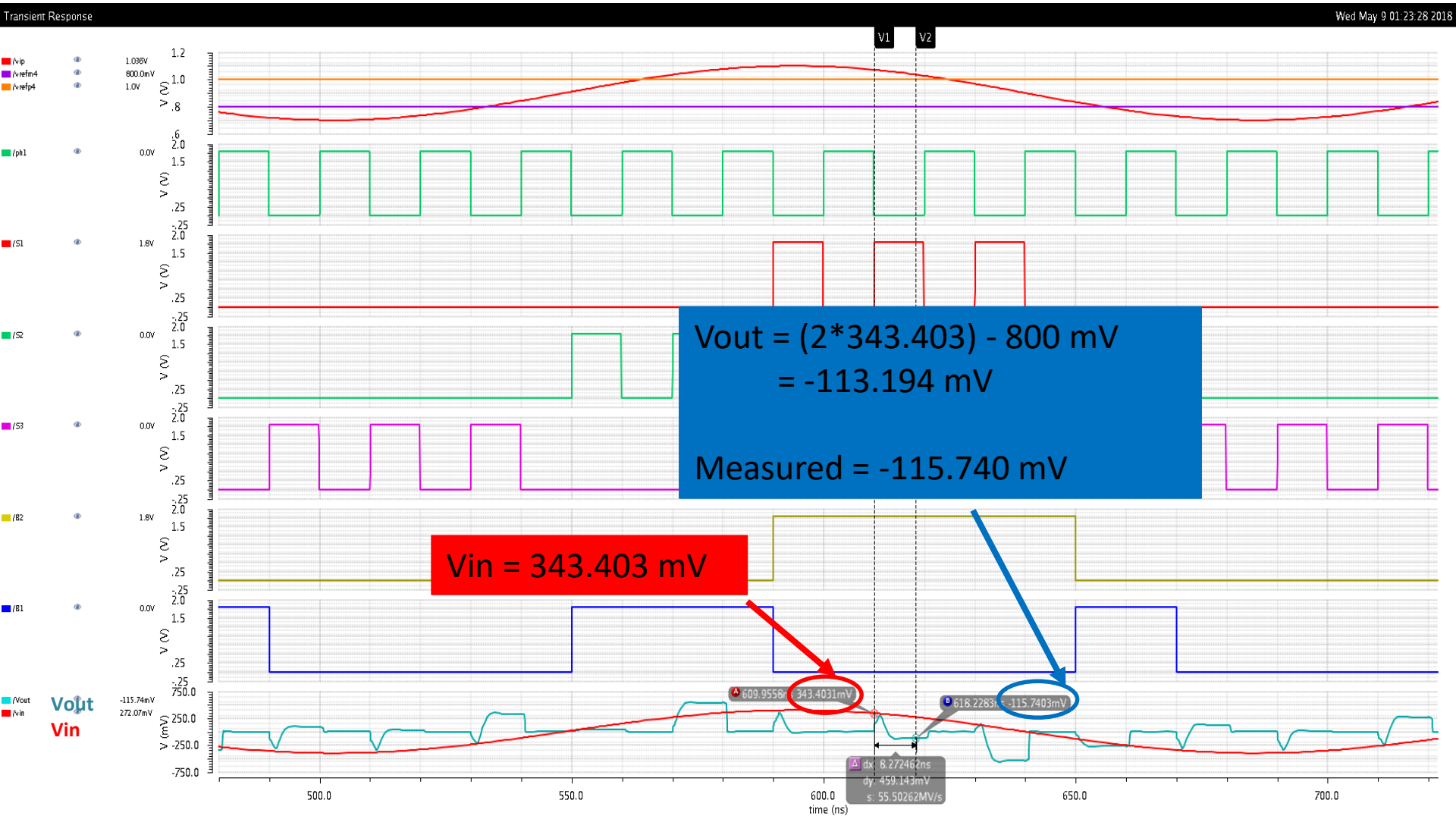
2) If Vrefm < Vin < Vrefp ➔ Vout = 2*Vin



Simulation Results

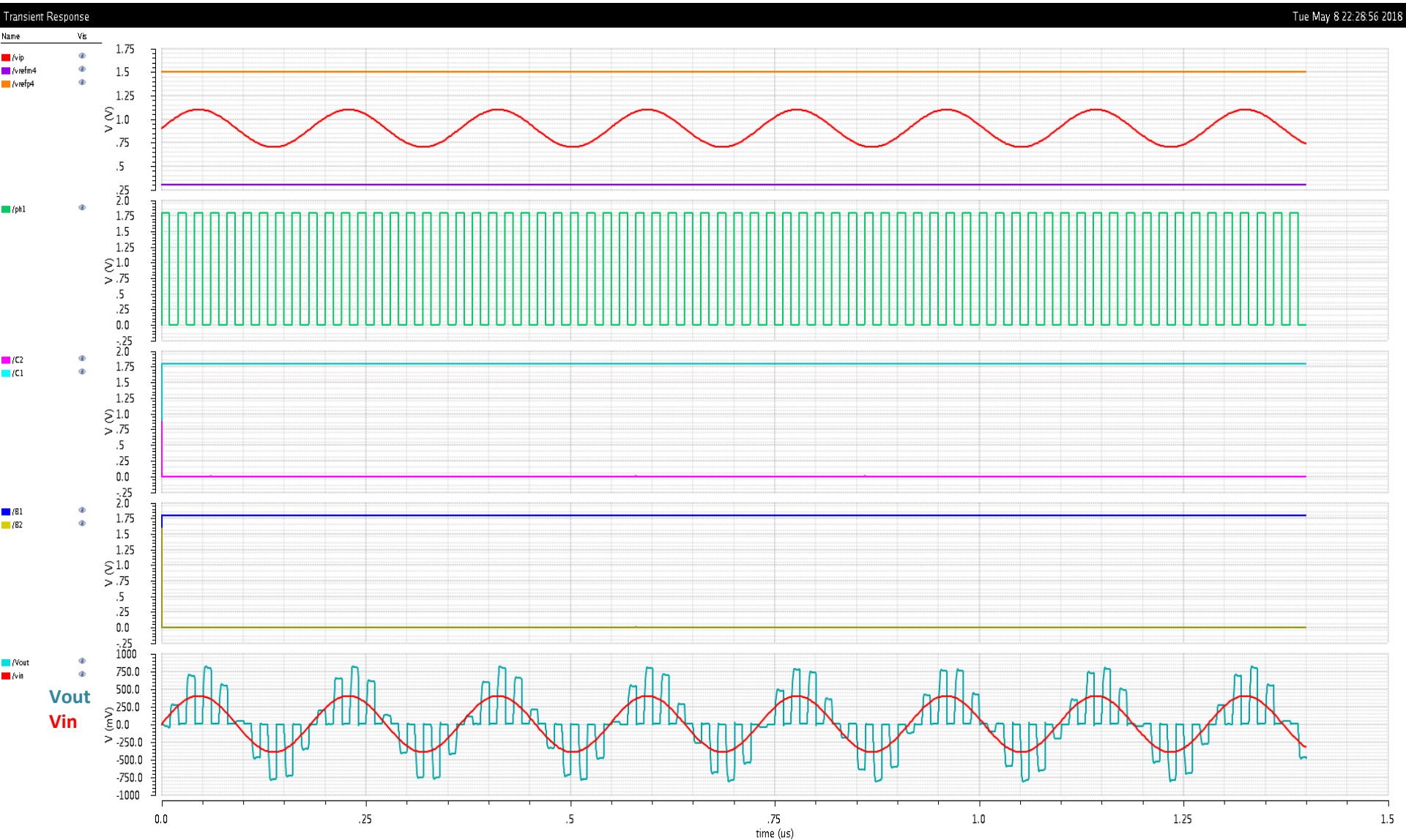
1- Normal Mode

3) If Vin > Vrefp ➔ Vout = 2*Vin - Vfs



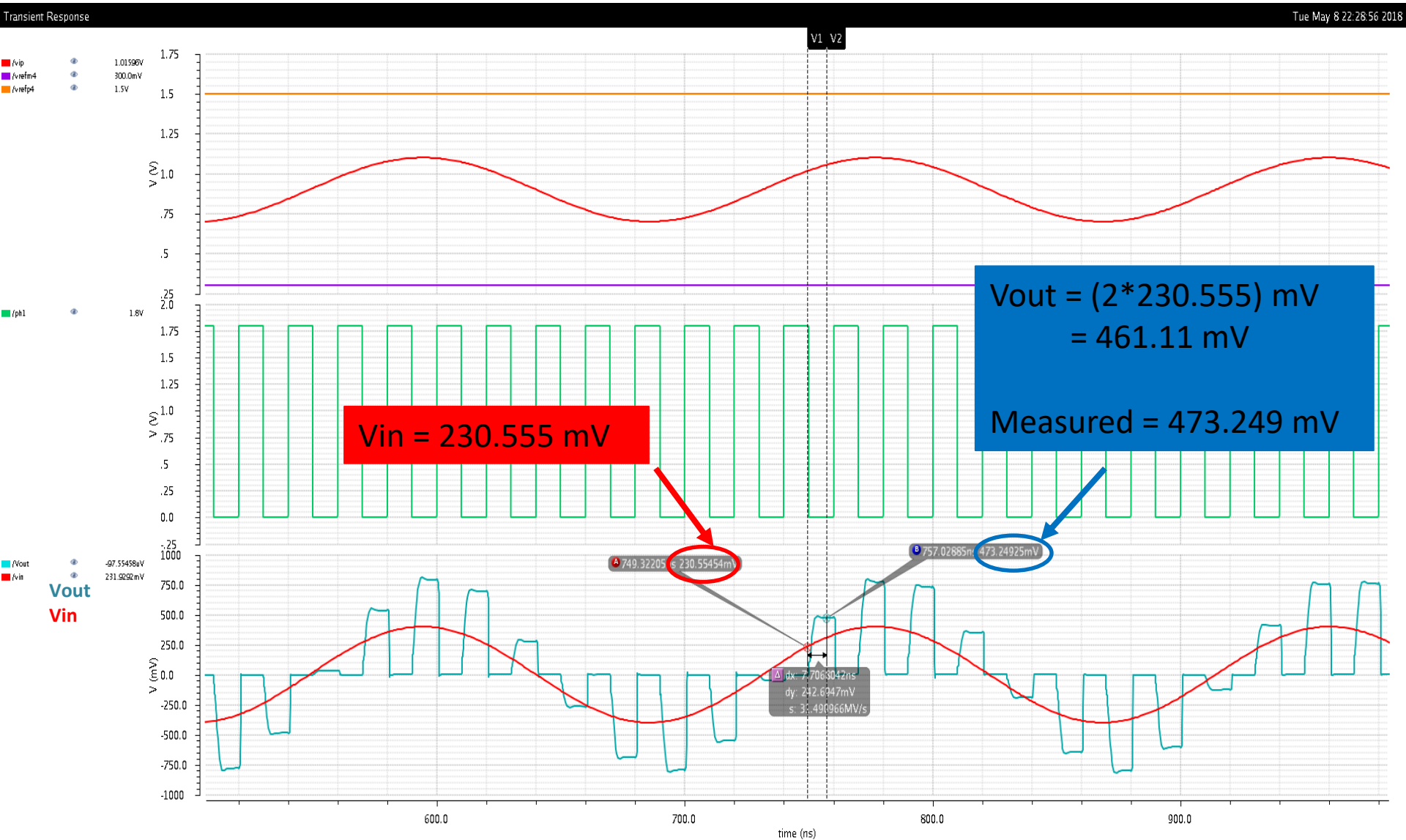
Simulation Results

2- 2x Gain Mode



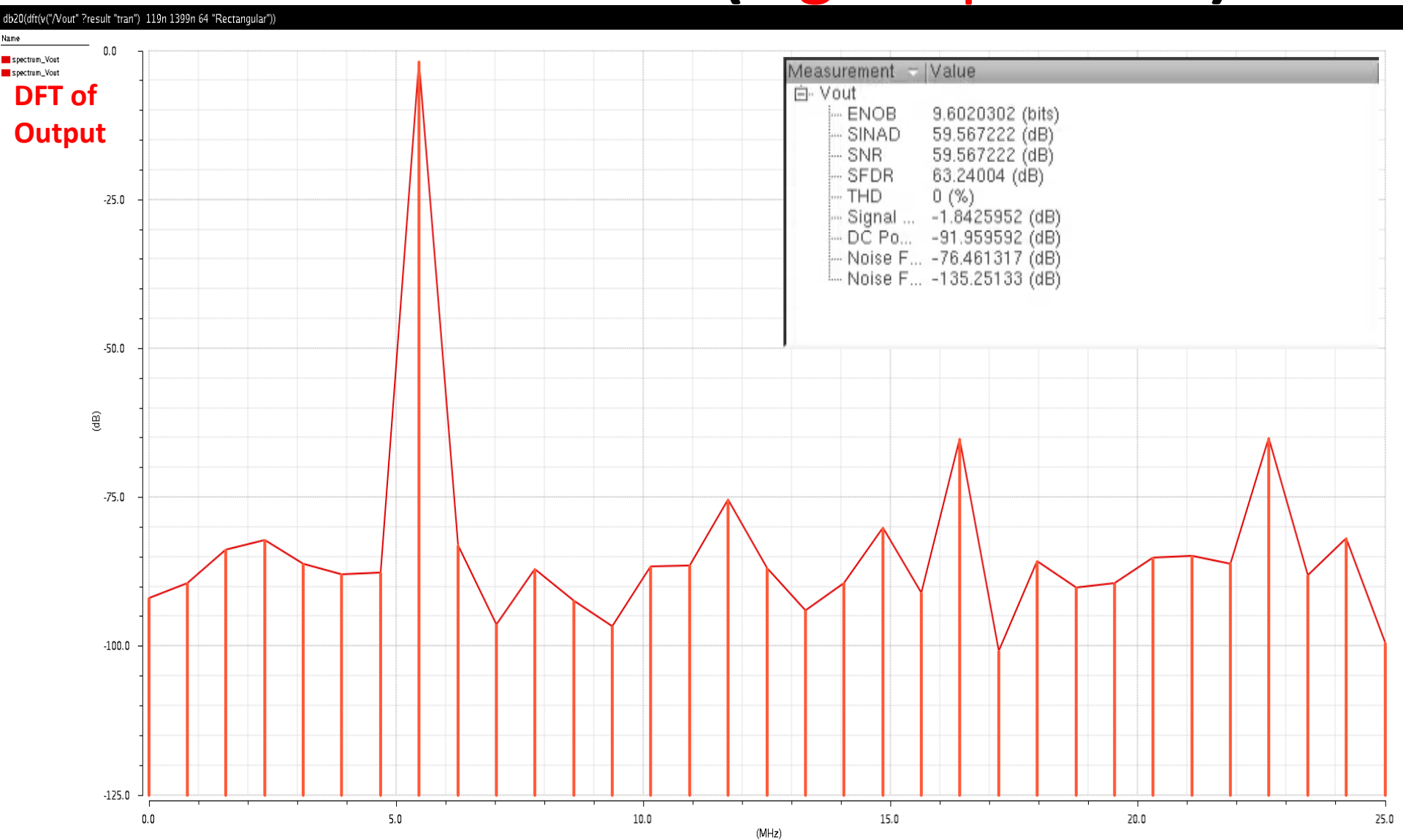
Simulation Results

2- 2x Gain Mode



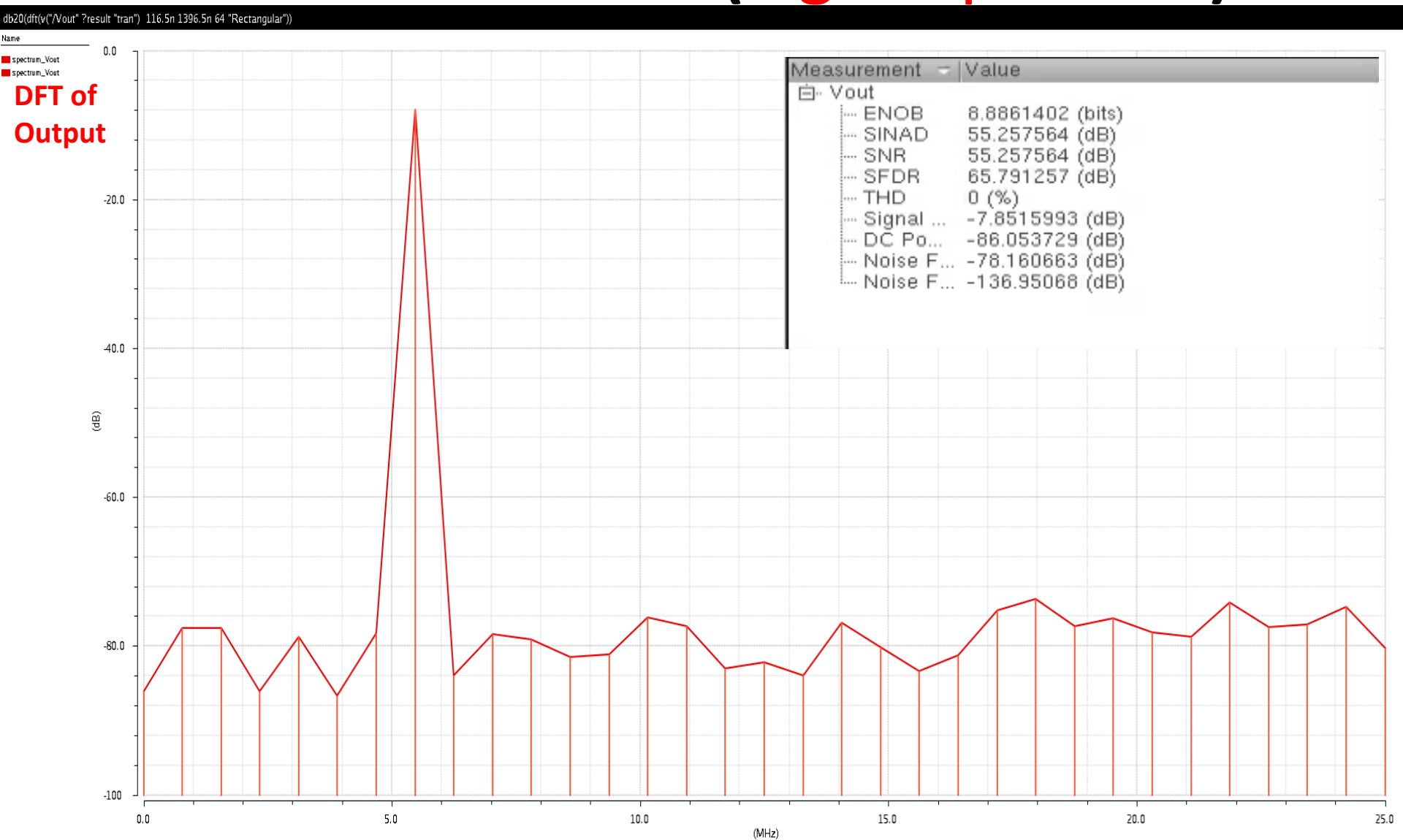
Simulation Results

2- 2x Gain Mode (Sig Amp = 0.4v)



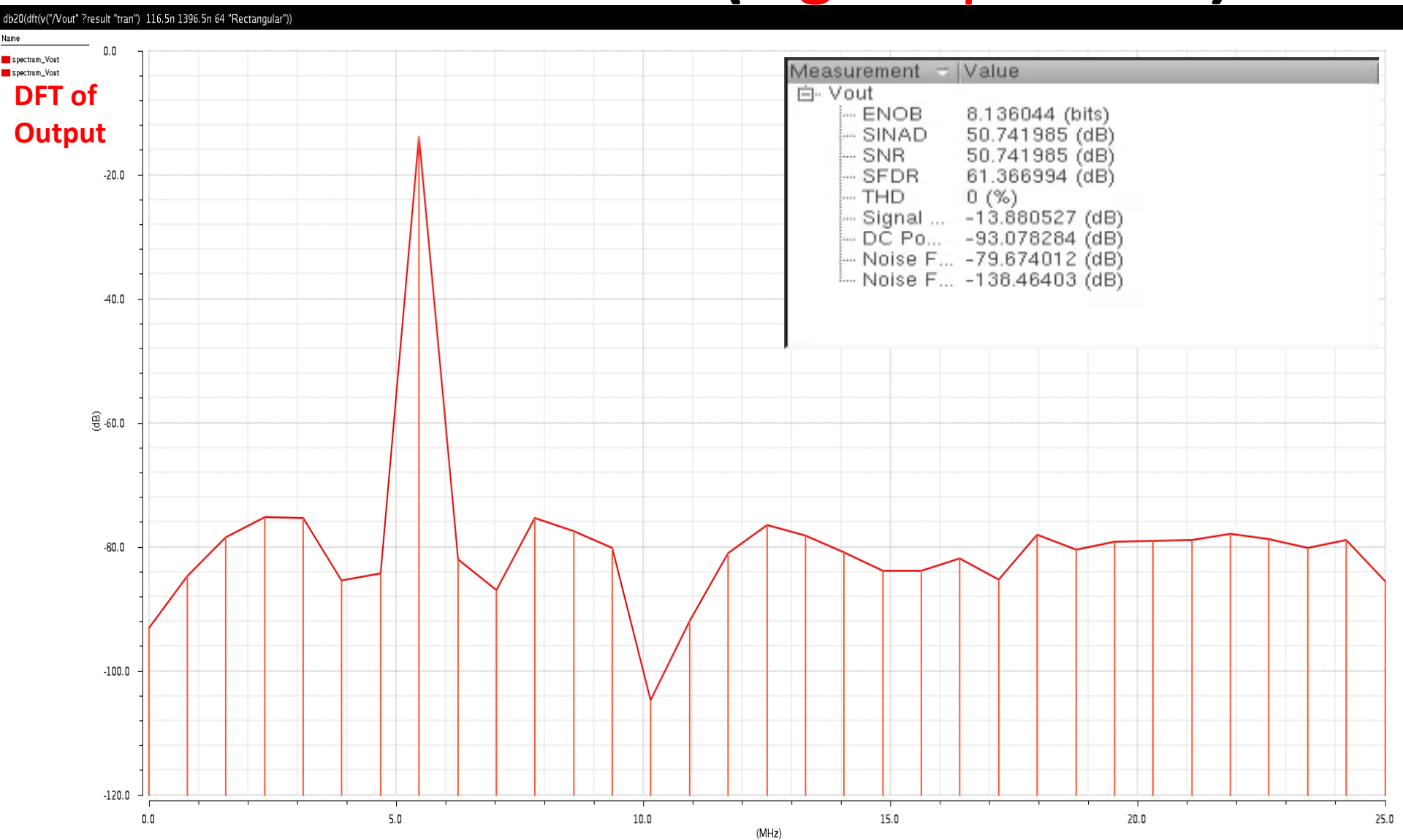
Simulation Results

2- 2x Gain Mode (Sig Amp = 0.2v)



Simulation Results

2- 2x Gain Mode (Sig Amp = 0.1v)

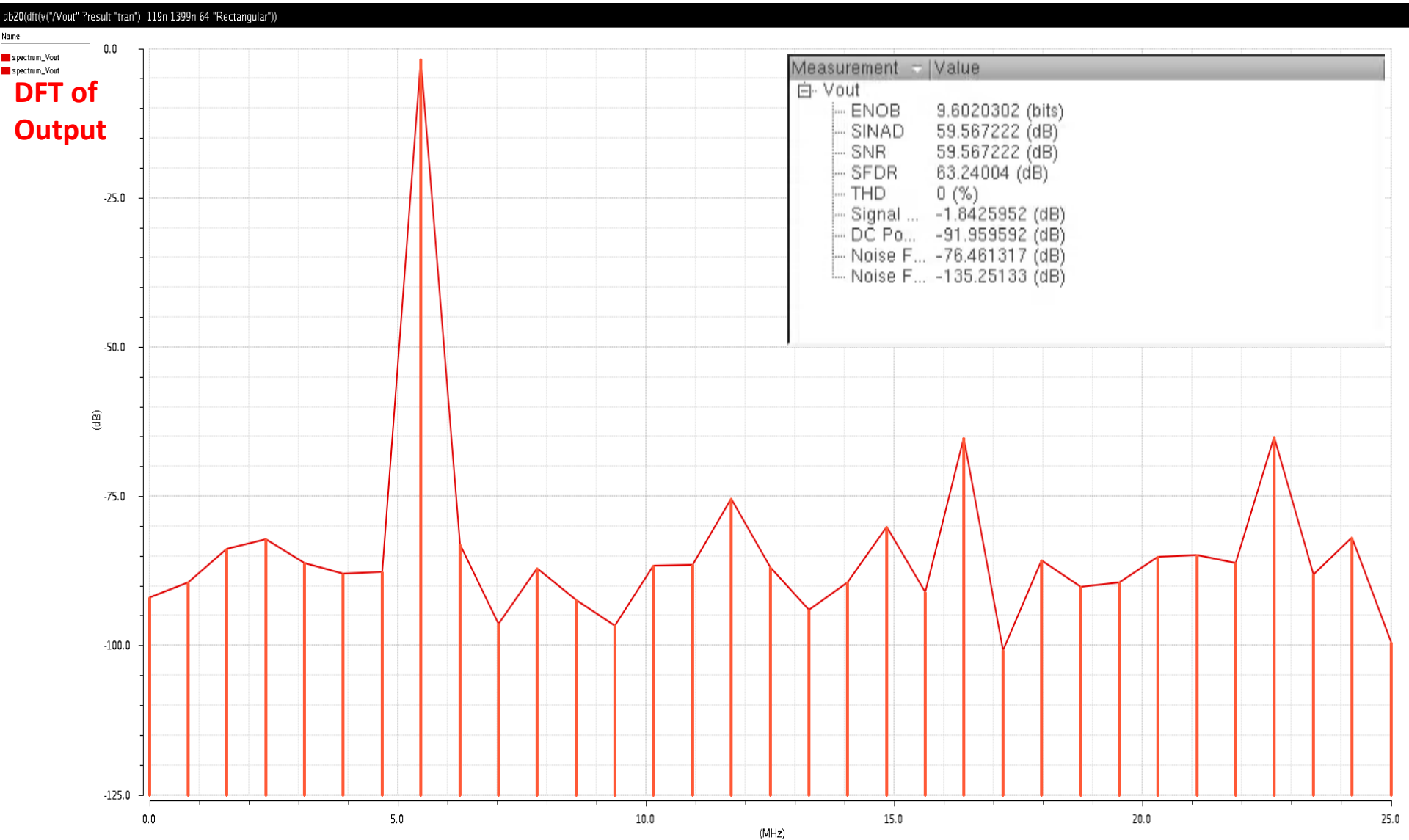


(5)
Corner Simulations

(For 2x Gain Mode)

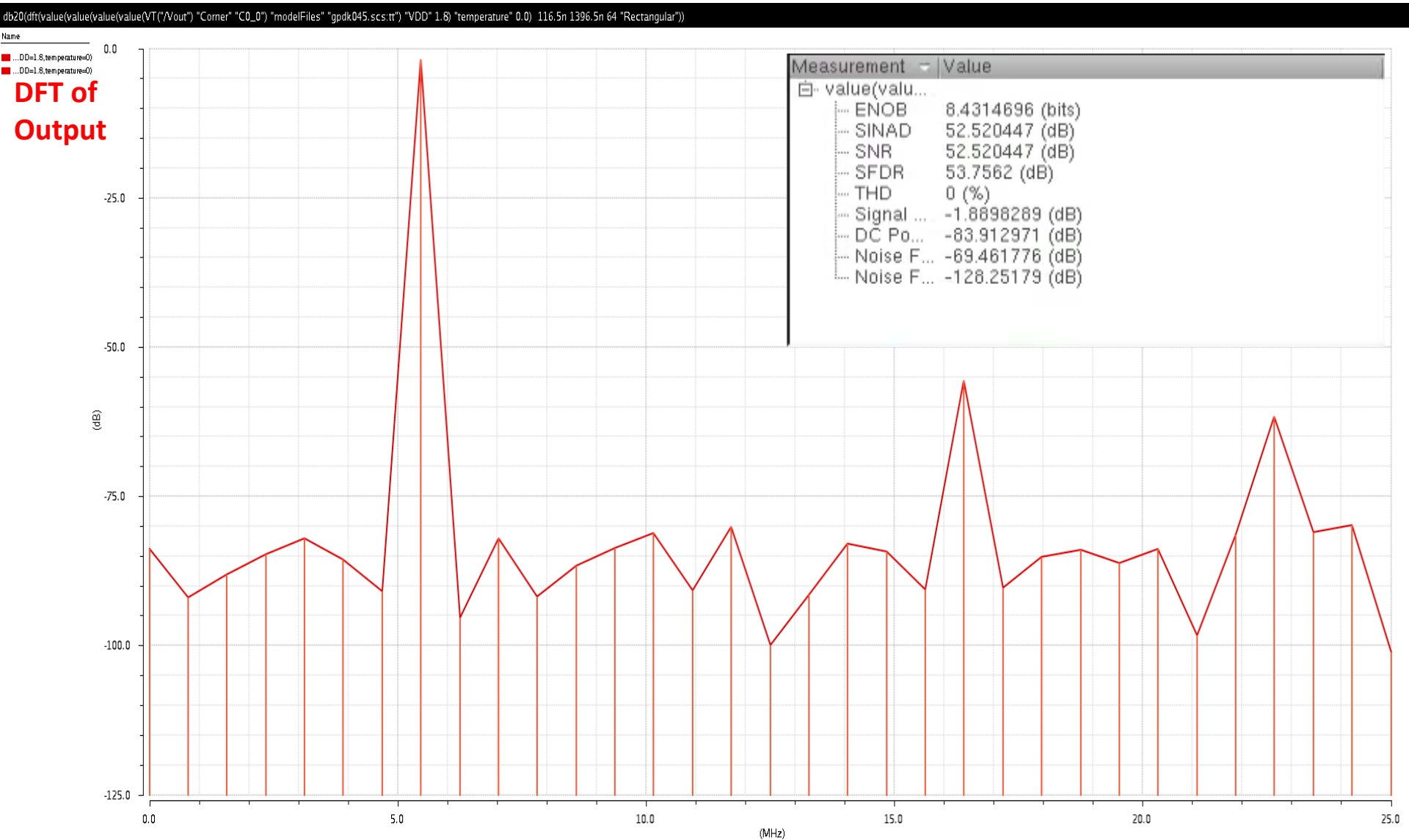
Simulation Results

TT, 1.8VDD, 27°



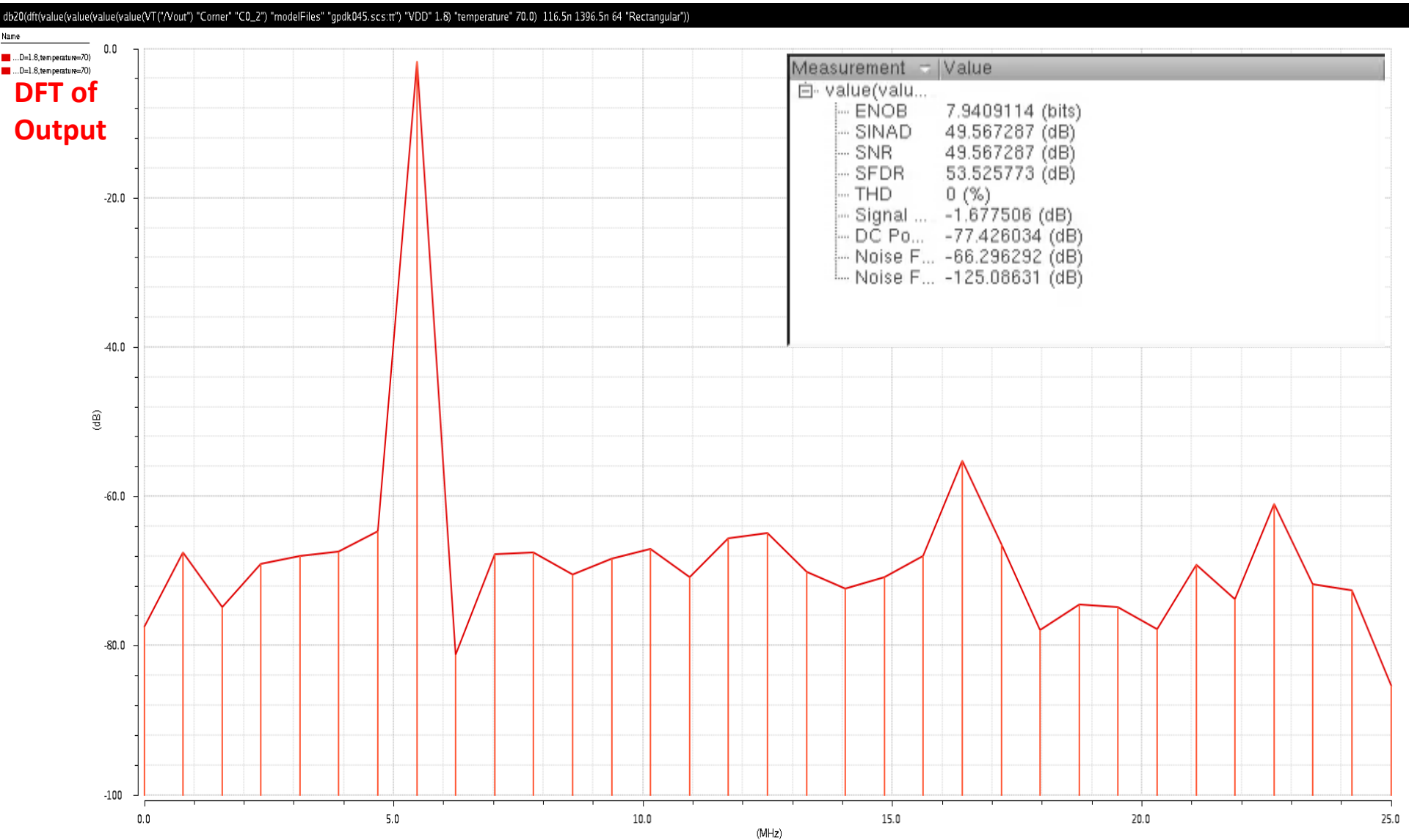
Simulation Results

TT, 1.8VDD, 0°



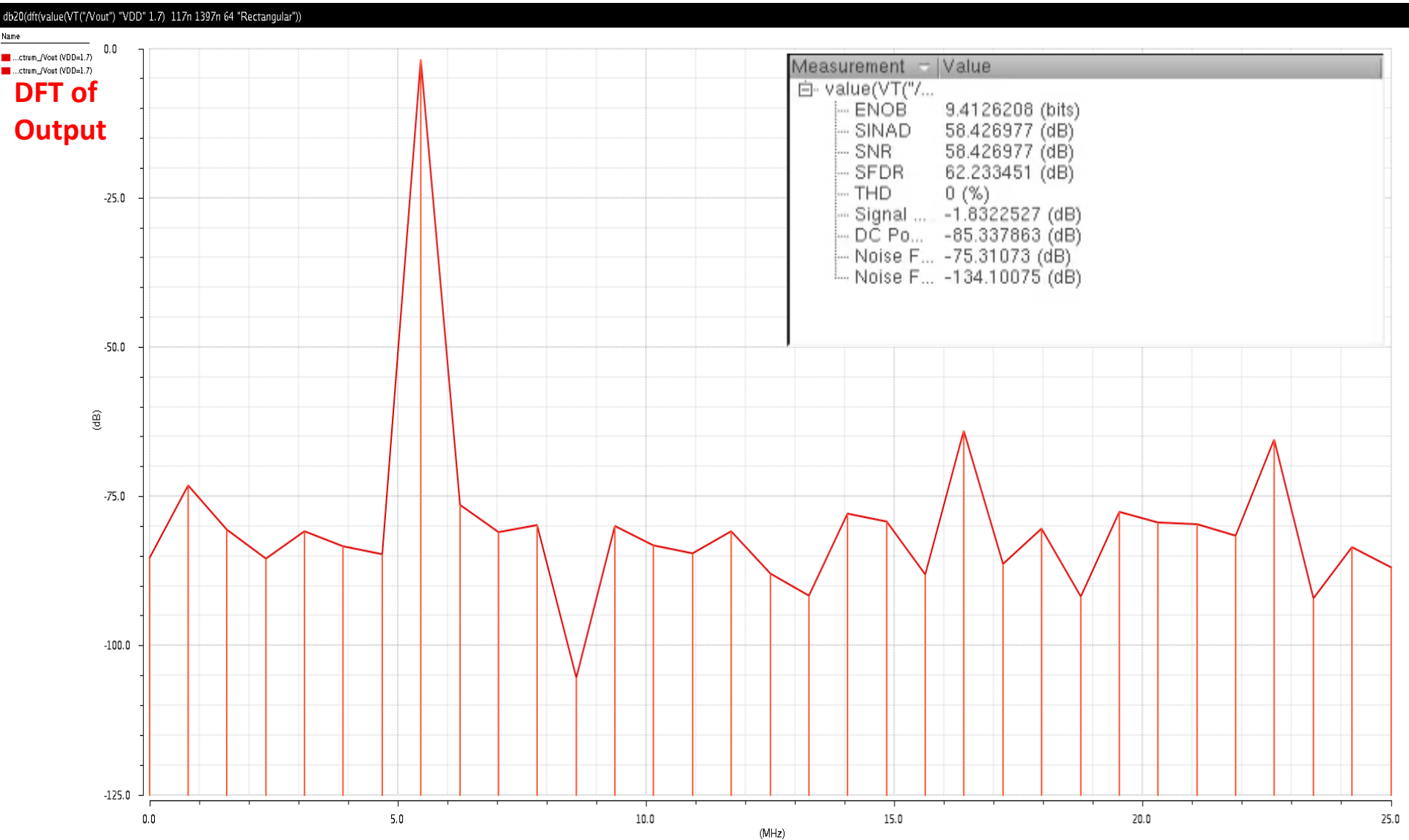
Simulation Results

TT, 1.8VDD, 70°



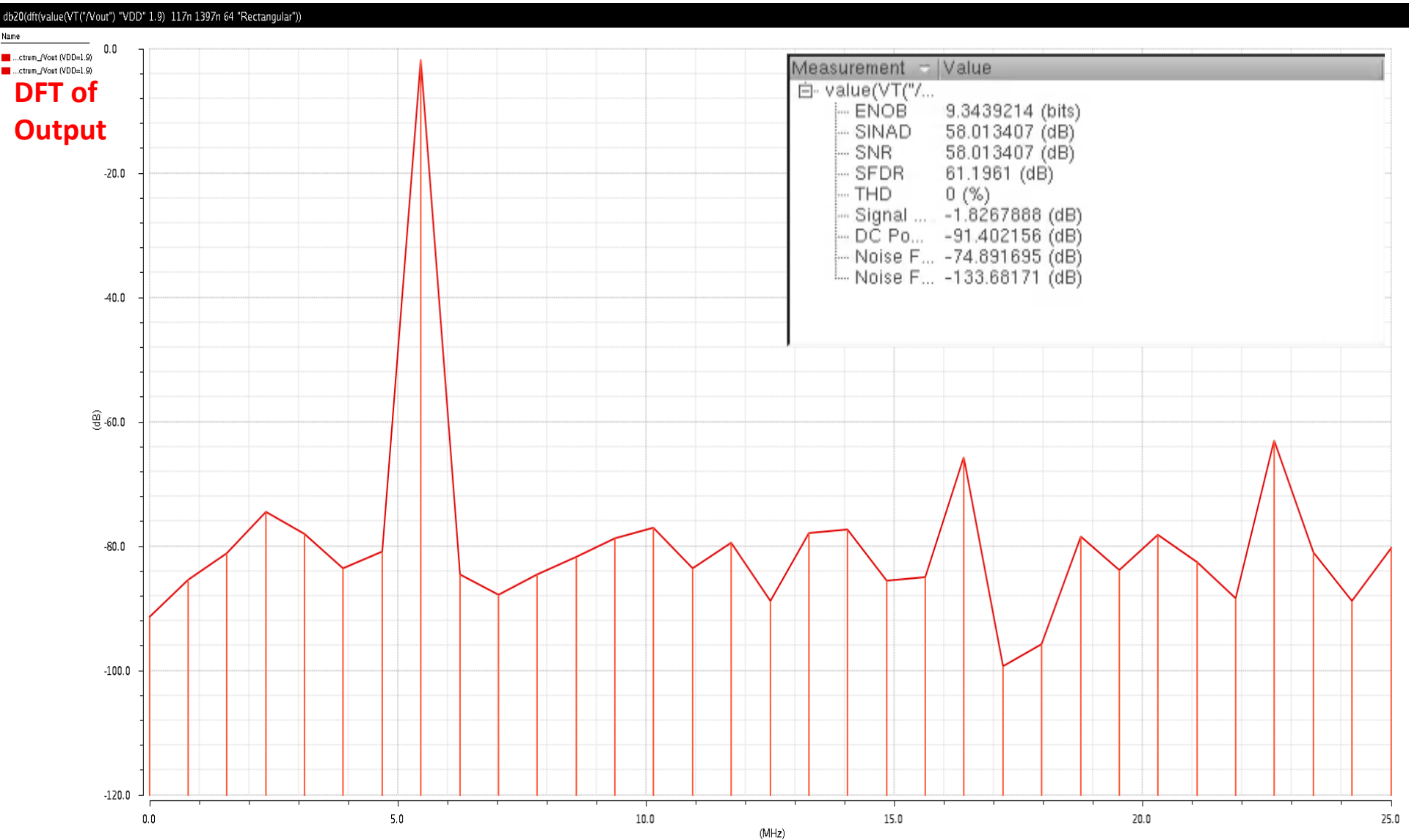
Simulation Results

TT, 1.7VDD, 27°



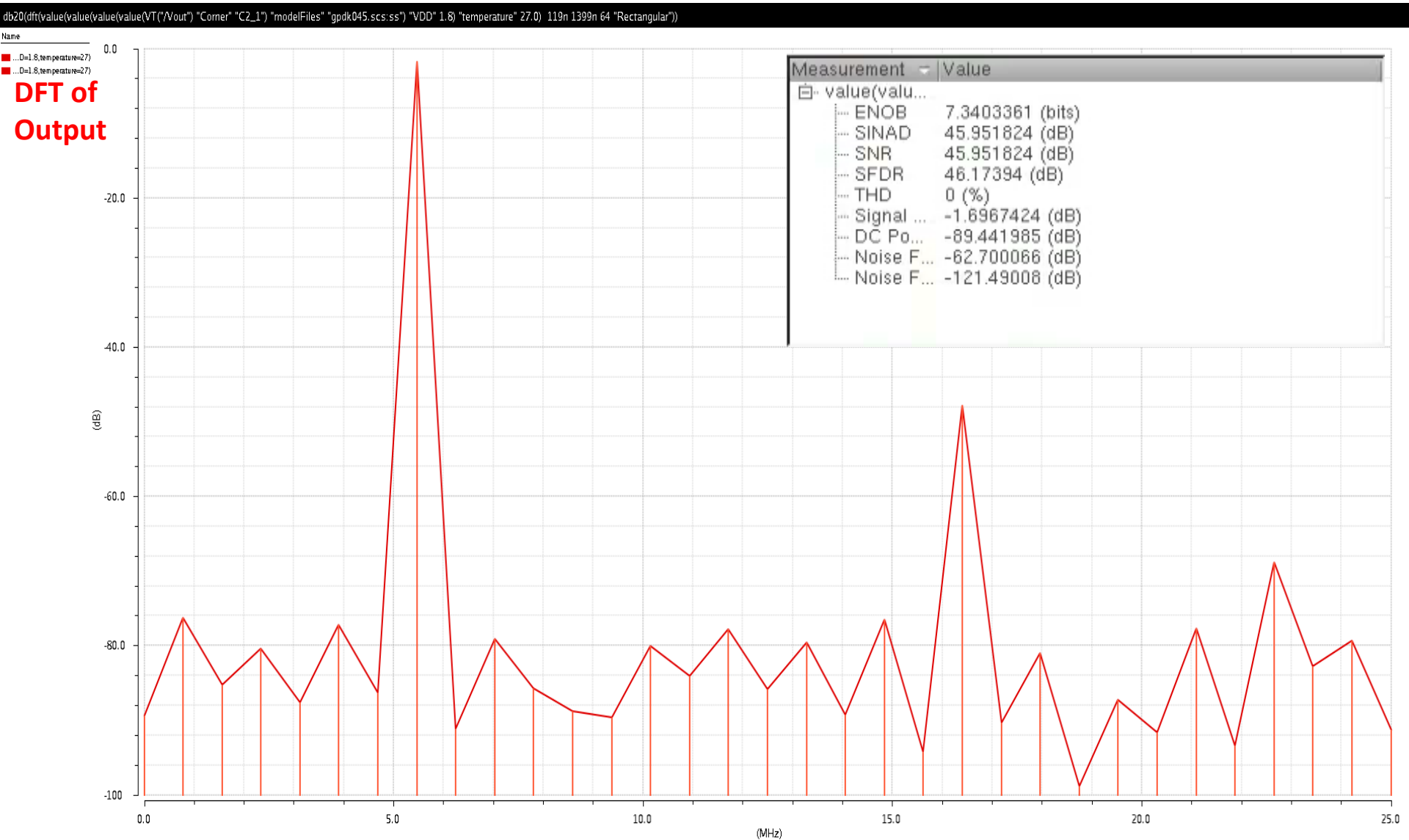
Simulation Results

TT, 1.9VDD, 27°



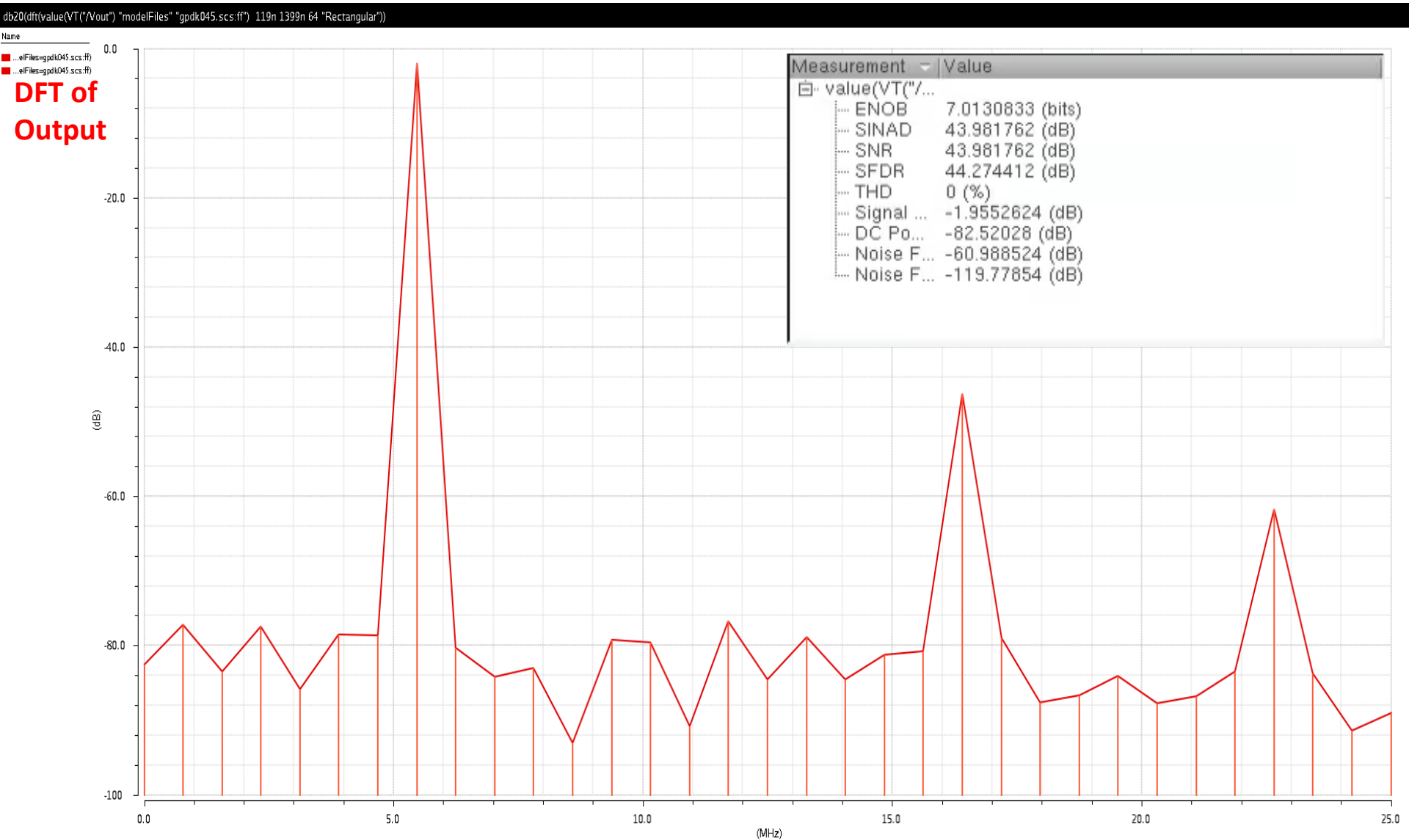
Simulation Results

SS, 1.8VDD, 27°



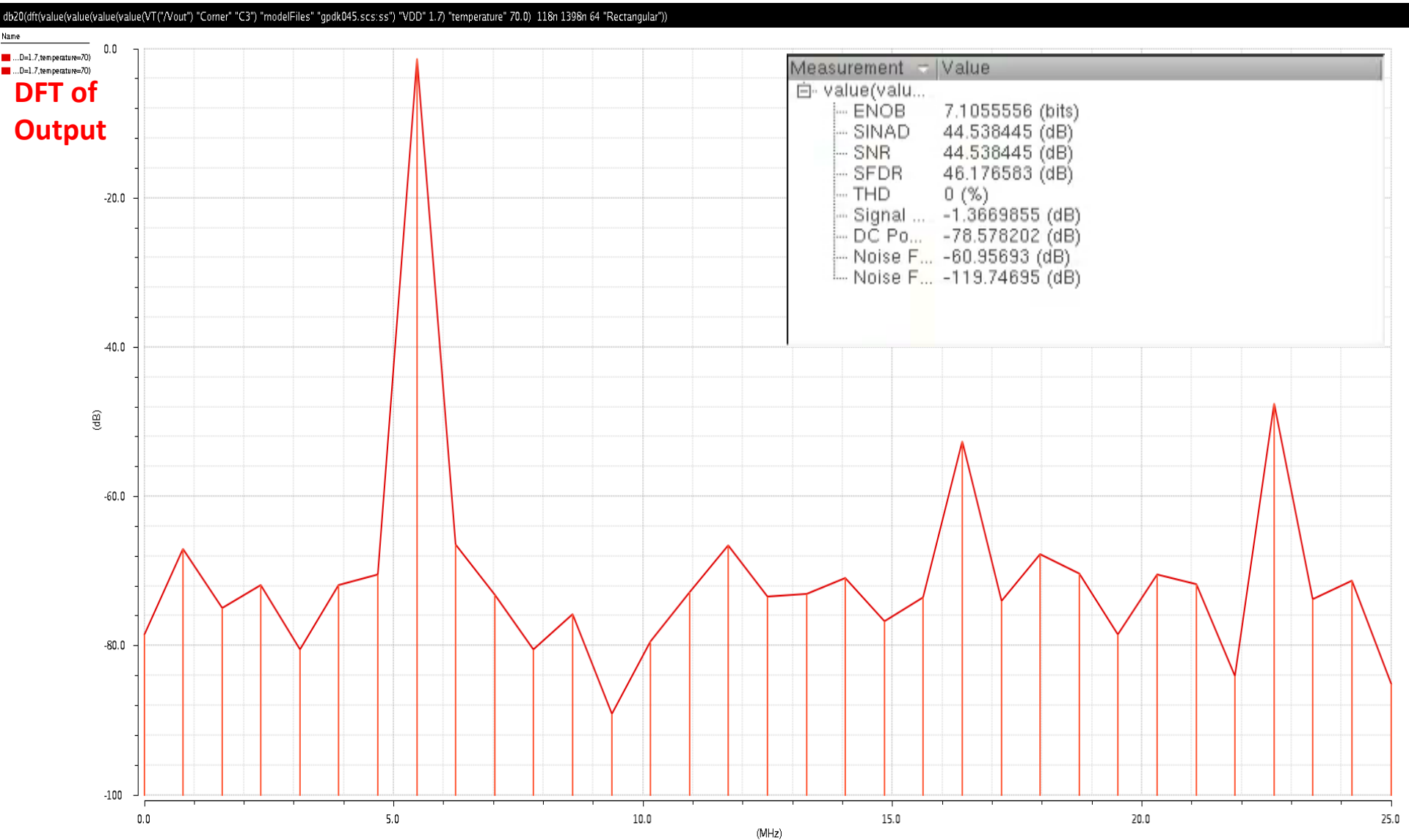
Simulation Results

FF, 1.8VDD, 27°



Simulation Results

SS, 1.7VDD, 70°



Summary

- OpAmps' AC responses:

AC Responses								
NMOS single-ended			PMOS single-ended			Boosted-Gain OpAmp		
Gain	PM	fu	Gain	PM	fu	Gain	PM	fu
22.2 dB	86.45°	73.9 MHz	**25.3 dB**	87.87°	60.8 MHz	**90.9 dB**	49.2°	372.2 MHz

- Power Consumption & FOM:

Circuit	Power consumtion
Comparators	2 * 439.74 uW
Sub-ADC Logic	3.569 uW
Bootstrap Switches	10 * 6.806 uW
OpAmp	1051.74 uW
Total	**2.0124 mW**
$FOM = \frac{10\ x\ P}{2^{ENOB} x\ f_S}$	**0.5179 pJ**

Summary

- Corners & Different input amplitudes:

Conditions		ENOB	SNR	
TT, 1.8VDD, 27°, SigAmp = 0.4v		**9.602 bits**	**59.567 dB**	
TT 1.8VDD 27°	Sig Amp = 0.2v	8.886 bits	55.258 dB	
	Sig Amp = 0.1v	8.136 bits	50.742 dB	
TT 1.8VDD SigAmp = 0.4v	Temp = 0°	8.431 bits	52.520 dB	
	Temp = 70°	7.941 bits	49.567 dB	
TT 27° SigAmp = 0.4v	VDD = 1.7v	9.413 bits	58.427 dB	
	VDD = 1.9v	9.344 bits	58.013 dB	
1.8VDD 27° SigAmp = 0.4v	SS	7.340 bits	45.951 dB	
	FF	**7.013 bits**	**43.982 dB**	Worst Case
SS, 1.7VDD, 70°, SigAmp = 0.4v		7.106 bits	44.538 dB	

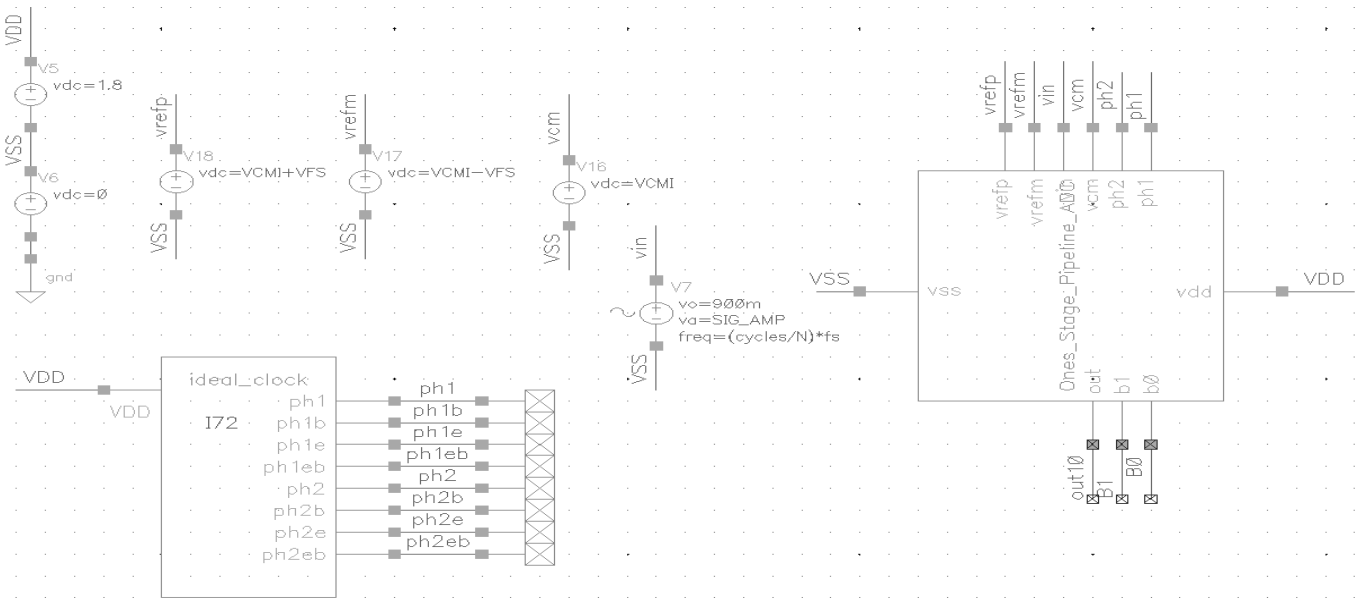
Corners

(6)
VerilogA Blocks
(Extra Points)

VerilogA:
1 stage 1.5 bit/stage ADC

```
// VerilogA for EE288, One_stage_pipleline_ADC, veriloga

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

module One_stage_pipleline_ADC(vdd,vss, vin,vrefm,vrefp,Vout,ph1,ph2,D0,D1,vcm);

input vrefm,vrefp,ph1,ph2,vcm,vin;
inout vss,vdd;
output vout,D0,D1;

electrical vin,out,vss,vdd,vrefm,vrefp,ph1,ph2,b0,b1,vcm;
parameter real clk_vth = 0.5, delay = 0, ttime = 1p;
real v,c1,c2;
analog begin
  @(cross(V(ph1) - clk_vth,+1)) begin
      if((V(vin) > V(vrefp)))
         begin
         c1 = 2;
         c2 = 2;
         v= 0;
         end
      else
         begin
         if((V(vin) > V(vrefm))) begin
            c1 = 2;
            c2 = 0;
           v= 0;
            end
         else begin
            c1 = 0;
            c2 = 0;
           v= 0;
            end
         end
      end

   @(cross(V(ph2) - clk_vth,+1)) begin
         if(c2 >=V(vdd)) begin
            v = (V(vin)-0.9)*2 -1;
           end
         else begin
               if(c1 >=V(vdd)) begin
                     v = (V(vin)- 0.9)*2;
                  end
               else begin
                     v = (V(vin)- 0.9)*2 + 1;
                  end
           end

         end

V(vout) <+ transition(v,delay,ttime);
V(D0) <+ transition(c1,delay,ttime);
V(D1) <+ transition(c2,delay,ttime);

end

endmodule
```

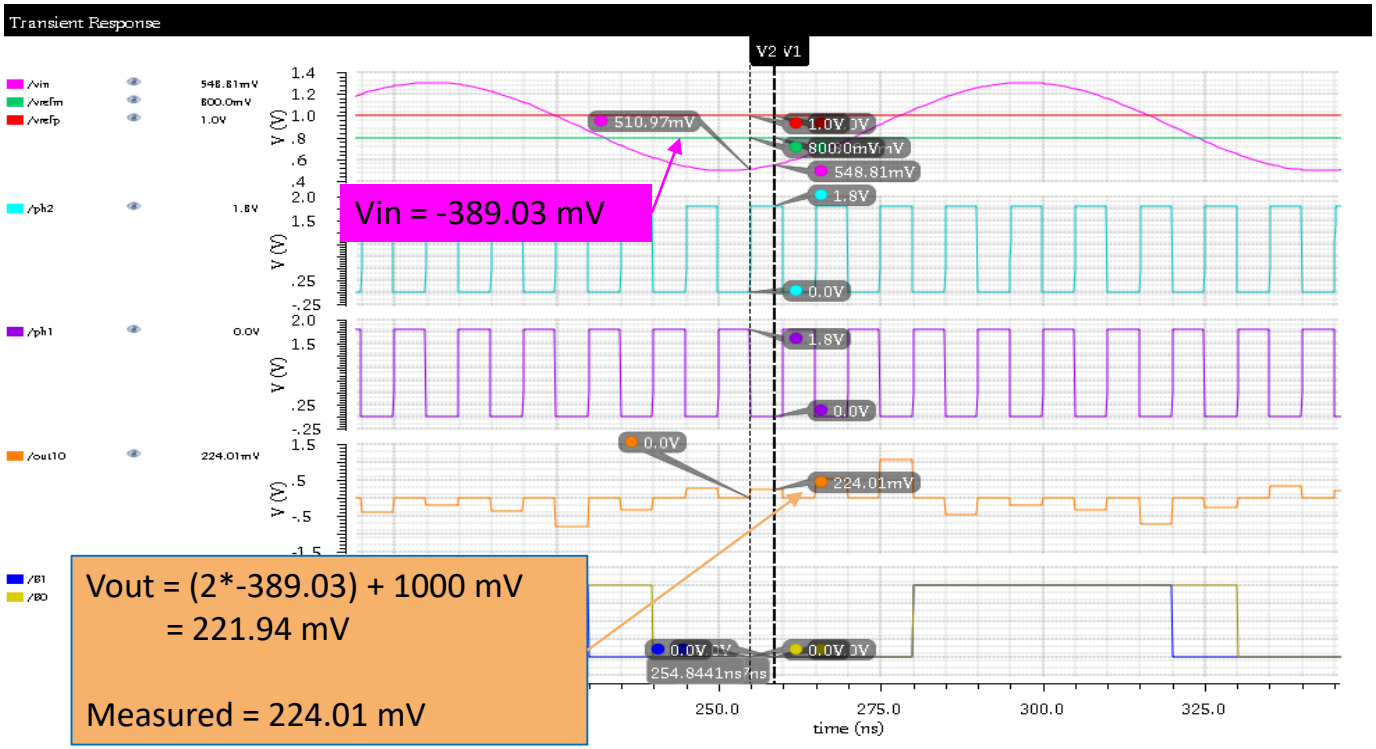
Testbench


VDD
V5
vdc=1.8
VSS
V6
vdc=0
gnd
vrefp
V18
vdc=VCMI+VFS
VSS
vrefm
V17
vdc=VCMI-VFS
VSS
vcm
V16
vdc=VCMI
VSS
vin
V7
vo=900m
va=SIG_AMP
freq=(cycles/N)*fs
VSS
VDD
ideal_clock
I72
ph1
ph1b
ph1e
ph1eb
ph2
ph2b
ph2e
ph2eb
VSS
vss
vdd
VDD
Ones_Stage_Pipeline_ADC
vrefp
vrefm
vin
vcm
ph2
ph1
out
b1
b0
out10
B1
B0

Simulation Results

1- Normal Mode

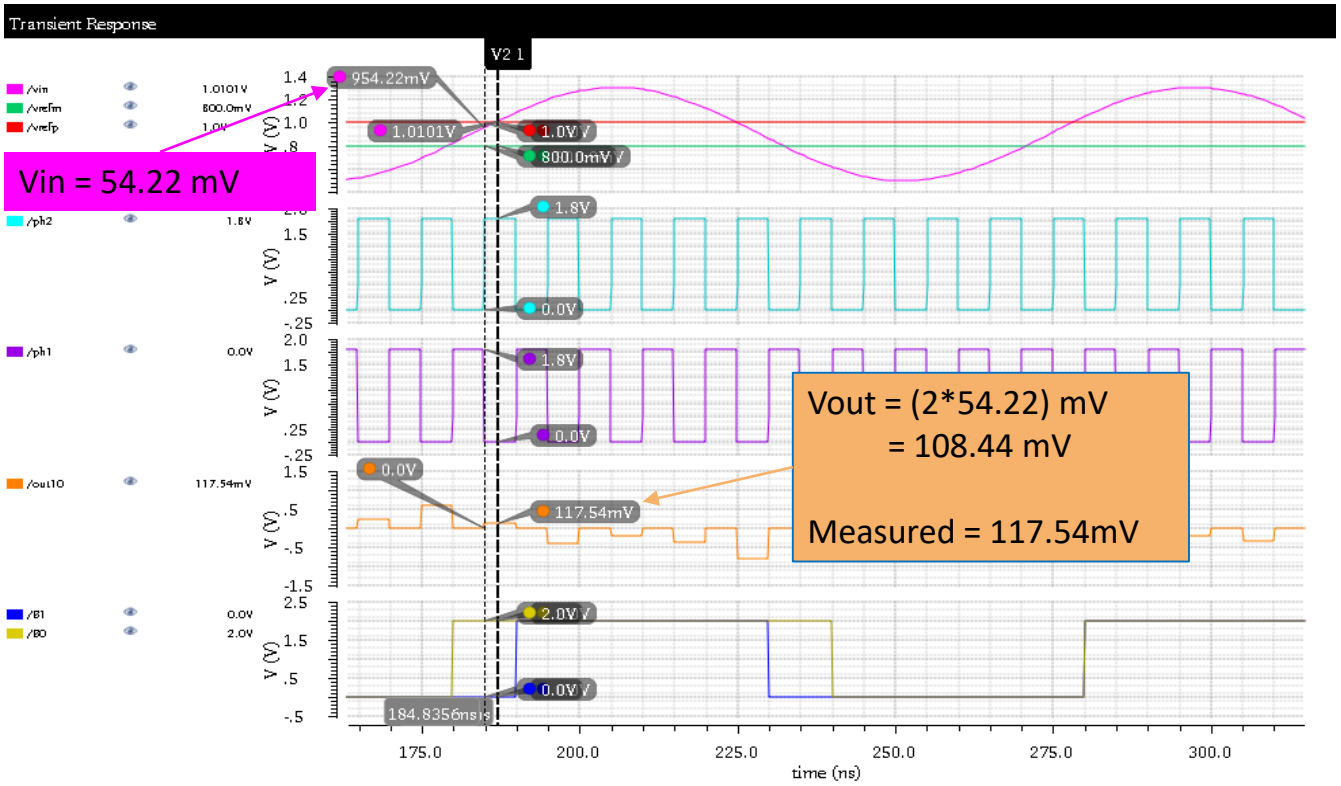
1) If Vin < Vrefm ➔ Vout = 2*Vin + Vfs



Simulation Results

1- Normal Mode

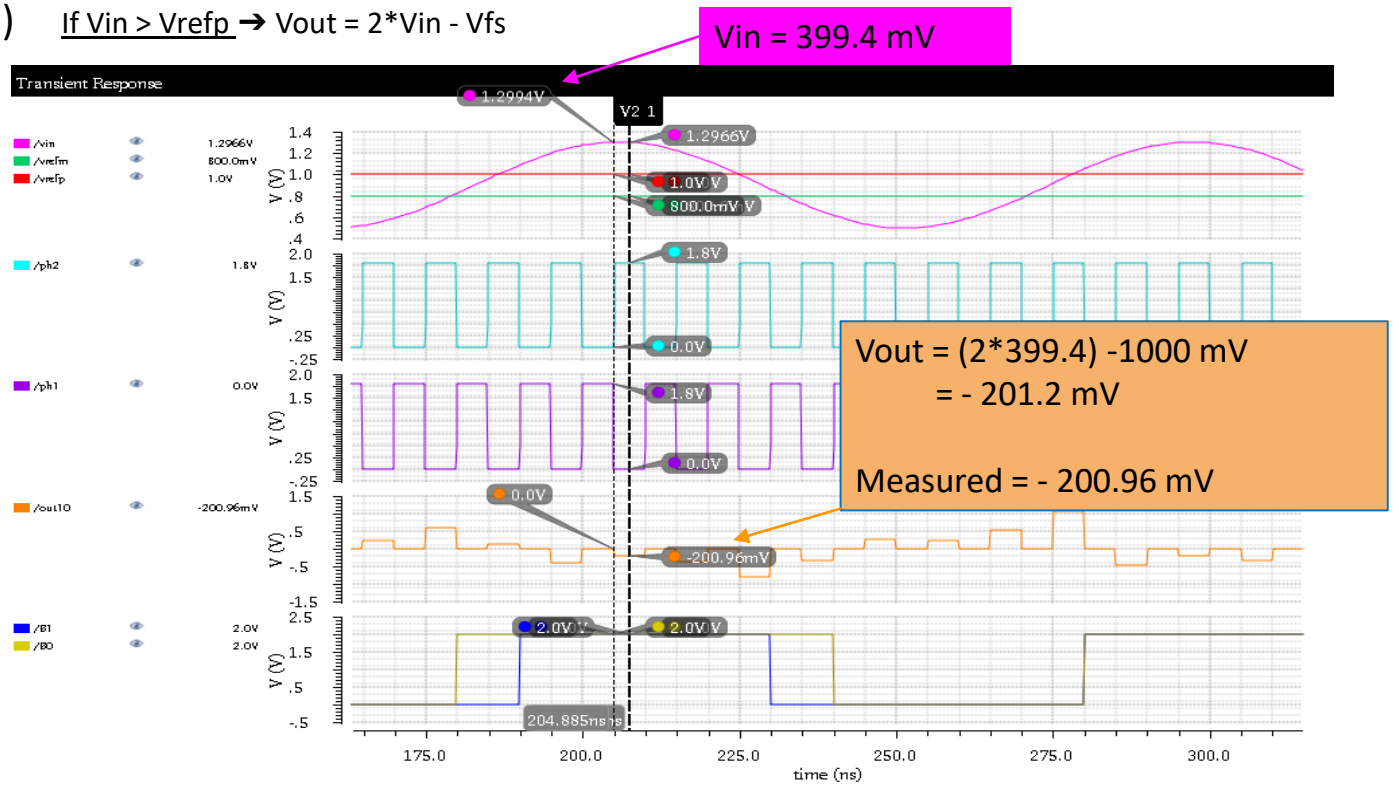
2) If Vrefm < Vin < Vrefp ➔ Vout = 2*Vin



Simulation Results

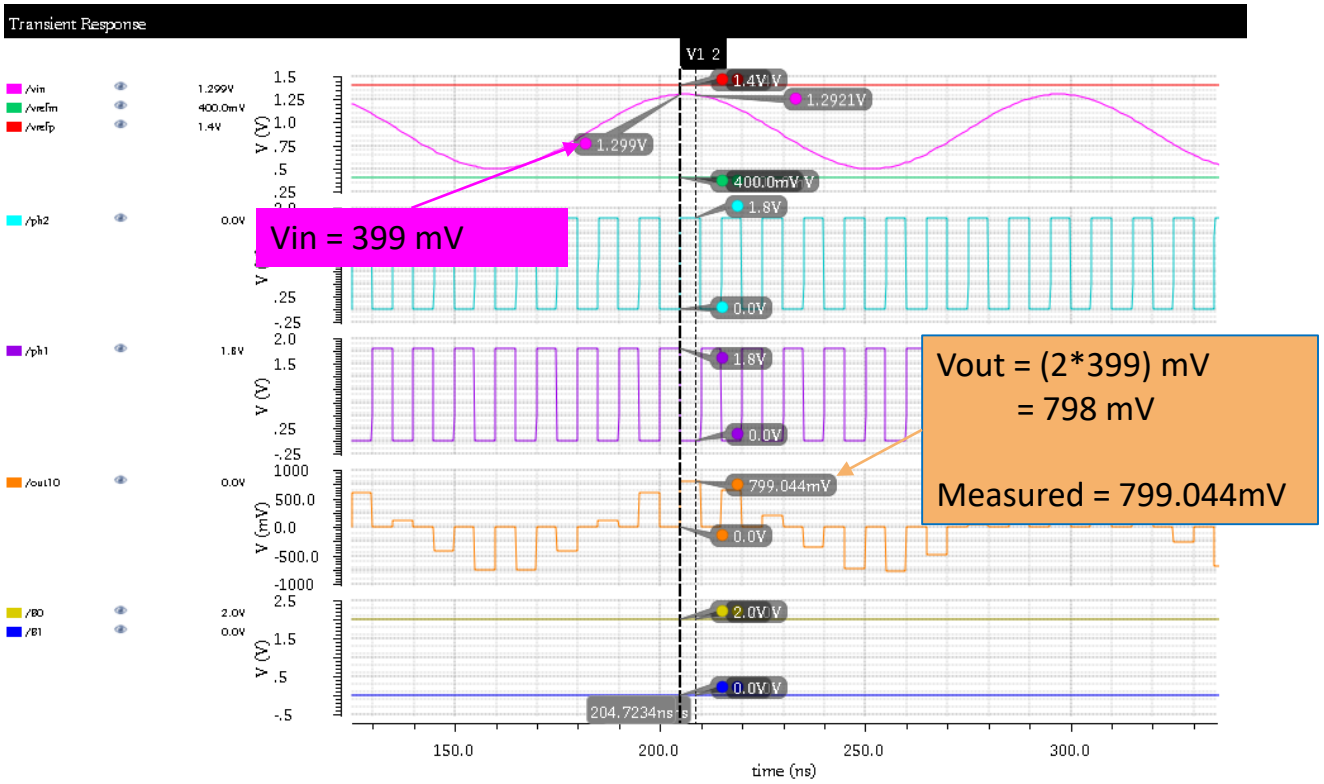
1- Normal Mode

3) <u>If Vin > Vrefp</u> ➔ Vout = 2*Vin - Vfs



Simulation Results

2- 2x Gain Mode

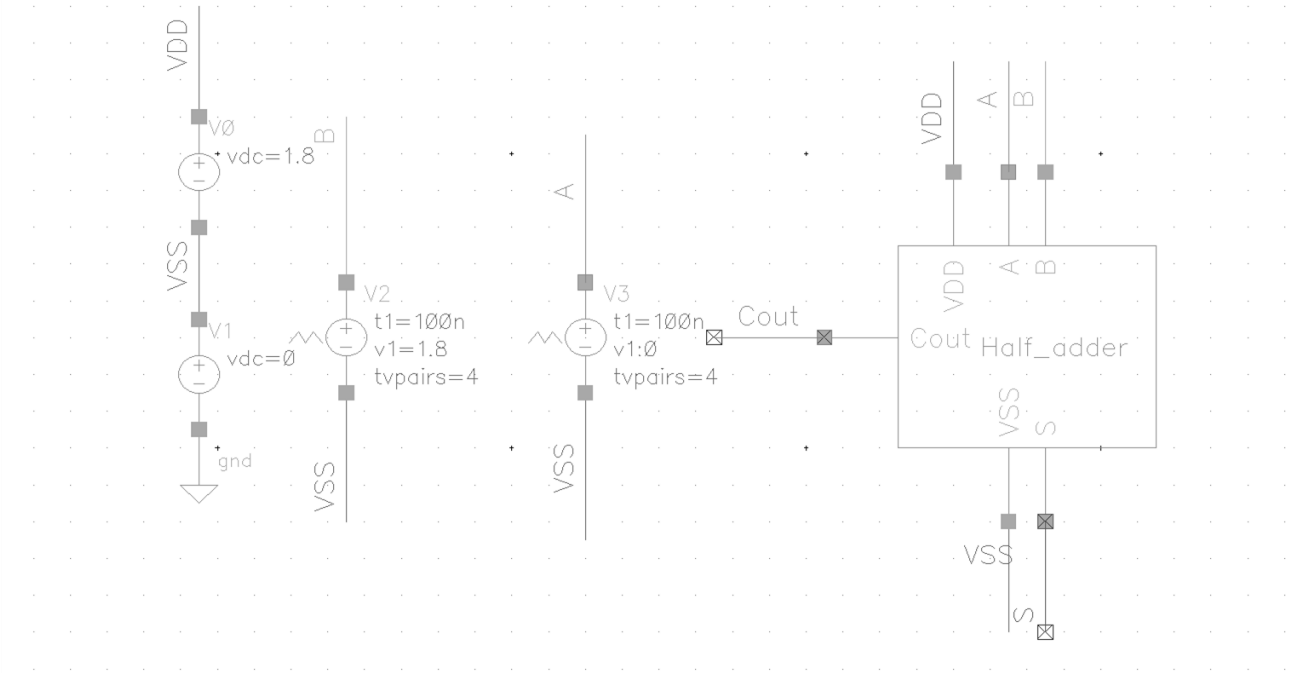


VerilogA:
Half Adder

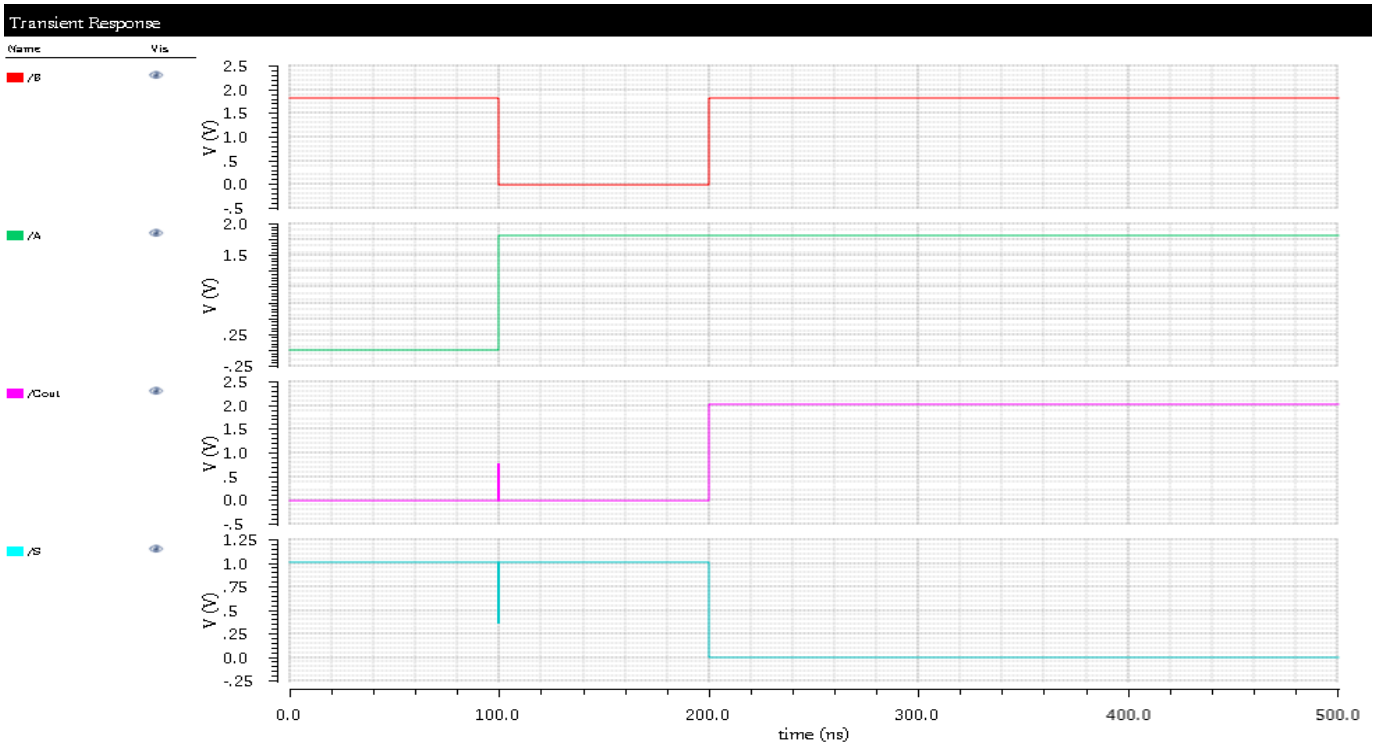
```
// VerilogA for EE288, half_adder, veriloga

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

module half_adder(A,B,Cout,VDD,VSS,S );

input A, B;

output Cout,S;

inout VDD, VSS;
parameter real  delay = 0, ttime = 1p;
real carri, sum;

electrical A, B, Cout, S, VSS, VDD;
```

```
analog begin
  if( (V(A)>=0.5) && (V(B)>=0.5) ) begin
     carri = 1;
     end
   else begin
     carri = 0;
     end
  if( (((V(A)>=0.5) && (V(B)>=0.5))|| ((V(A)<=0.5) && (V(B)<=0.5))))  begin
     sum =0 ;
     end
   else begin
     sum = 1;
     end
V(Cout) <+ transition(carri,delay,ttime);
V(S) <+ transition(sum,delay,ttime);
enD
endmodule
```

Test Bench:


VDD
VØ
vdc=1.8
VSS
V1
vdc=Ø
gnd
B
V2
t1=1ØØn
v1=1.8
tvpairs=4
VSS
A
V3
t1=1ØØn
v1:Ø
tvpairs=4
VSS
Cout
VDD
A
B
VDD
A
B
Cout
Half_adder
VSS
S
VSS
S

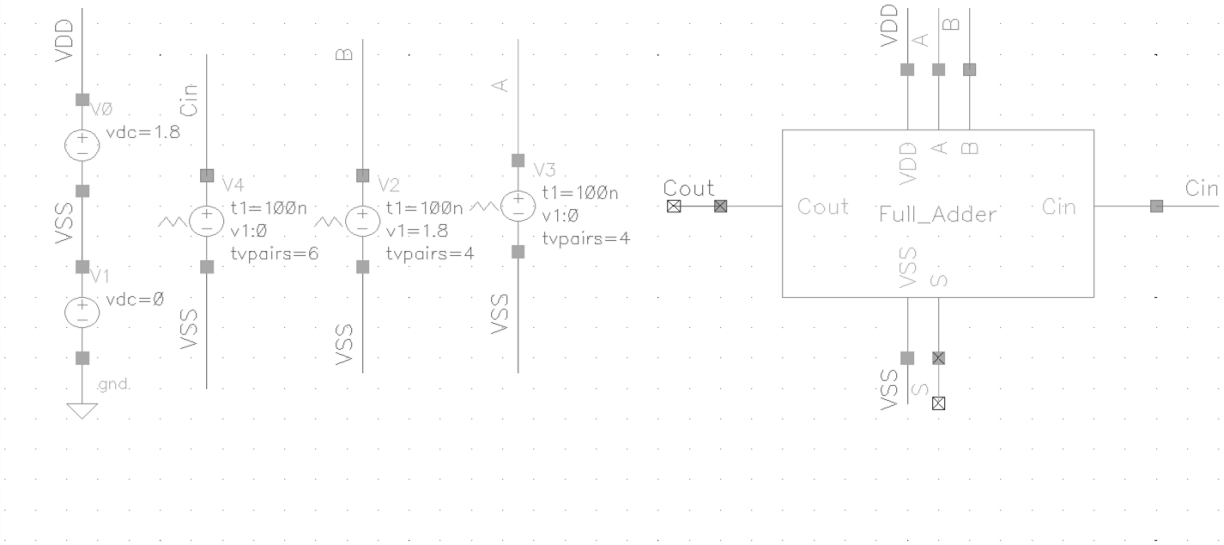
Simulation Results



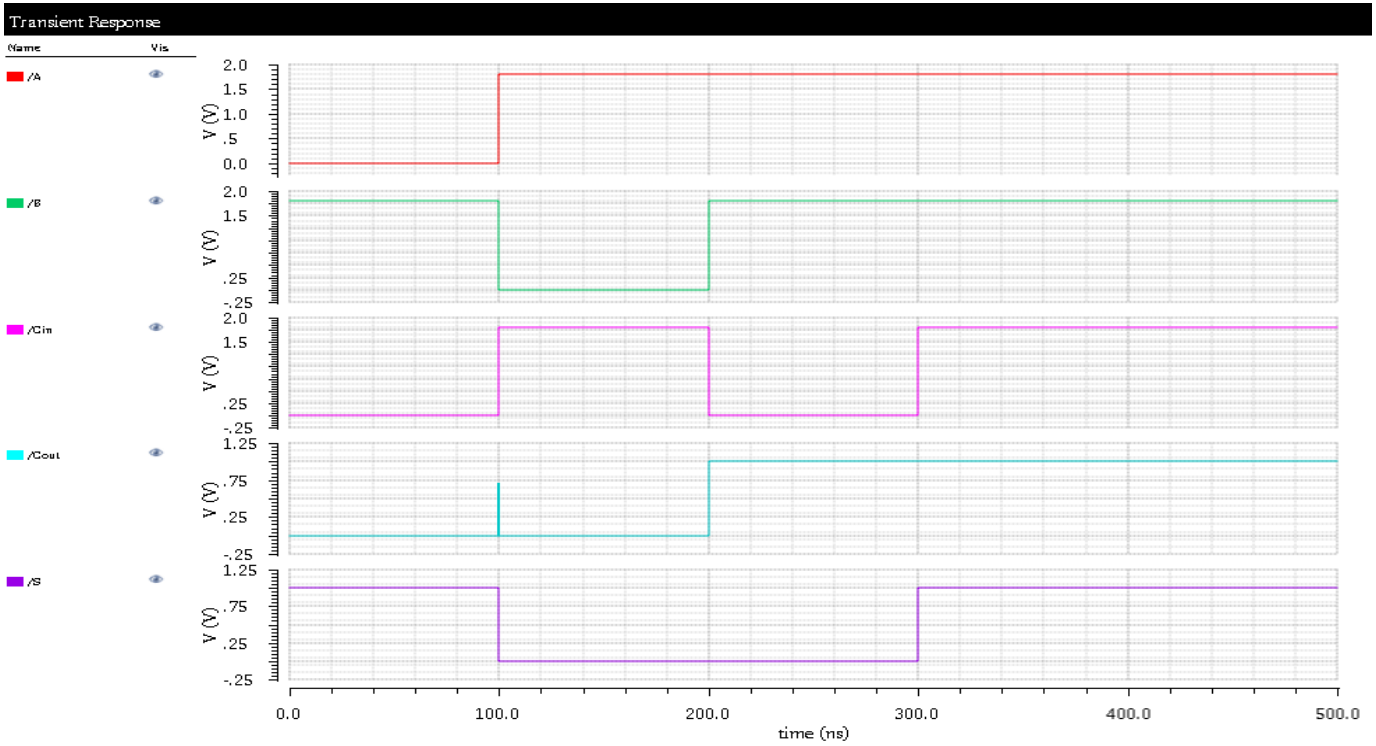
VerilogA:
Full Adder

```
// VerilogA for EE288, Full_Adder, veriloga
`include "constants.vams"
`include "disciplines.vams"
module Full_Adder(A,B,Cout,VDD,VSS,S,Cin );
input A, B,Cin;
output Cout,S;
inout VDD, VSS;
parameter real  delay = 0, ttime = 1p;
real carri, sum,xor1,AND1,AND2;
electrical A, B, Cout, S, VSS, VDD,Cin;
analog begin
    if( ((V(A)>=0.9 && (V(B)>=0.9))||(V(A)<=0.5 && (V(B)<=0.5))))  begin
      xor1=1 ;
      end
   else begin
      xor1 = 0;
      end
   if( ((V(Cin)>=0.9 && (xor1>=0.9))||(V(Cin)<=0.5 && (xor1<=0.5))))  begin
      sum=1 ;
      end
   else begin
      sum = 0;
      end
 if( (V(A)>=0.9) && (V(B)>=0.9) ) begin
      AND1 = 1;
      end
   else begin
      AND1 = 0;
      end
    if( (xor1>=0.9) && (V(Cin)>=0.9) ) begin
     AND2 = 1;
      end
   else begin
      AND2 = 0;
      end
   if((AND1 >= 0.9) ||(AND2 >= 0.9) ) begin
      carri= 1.0;
      end
   else begin
      carri = 0.0;
      end
V(Cout) <+ transition(carri,delay,ttime);
V(S) <+ transition(sum,delay,ttime);
end
endmodule
```

Test Bench


VDD
VØ
vdc=1.8
VSS
V1
vdc=Ø
gnd
Cin
V4
t1=100n
v1:Ø
tvpairs=6
VSS
B
V2
t1=100n
v1=1.8
tvpairs=4
VSS
A
V3
t1=100n
v1:Ø
tvpairs=4
VSS
Cout
Cout
Full_Adder
Cin
Cin
VDD
A
B
VDD
A
B
VSS
S
VSS
S

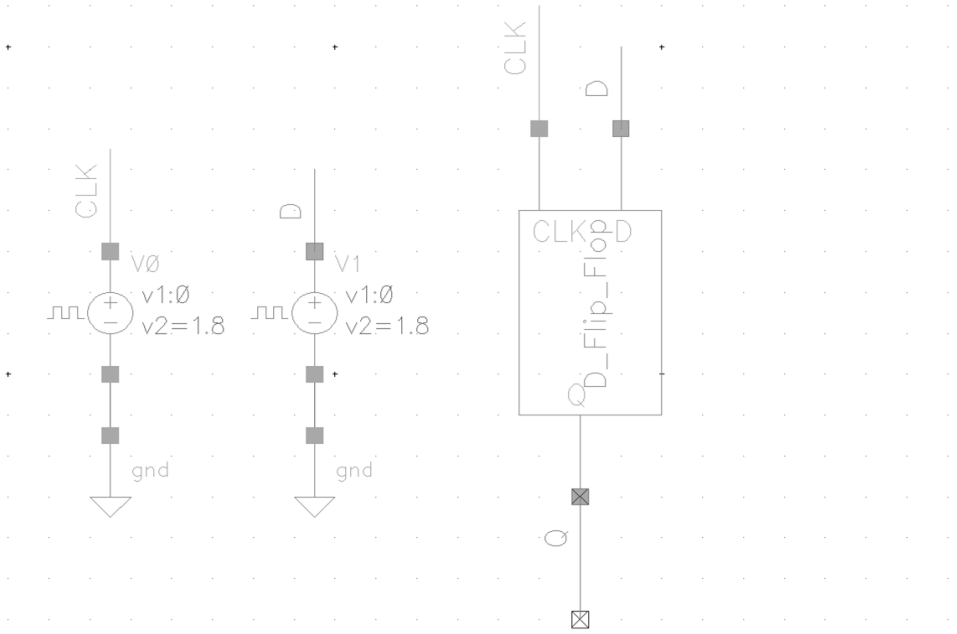
Simulation Results



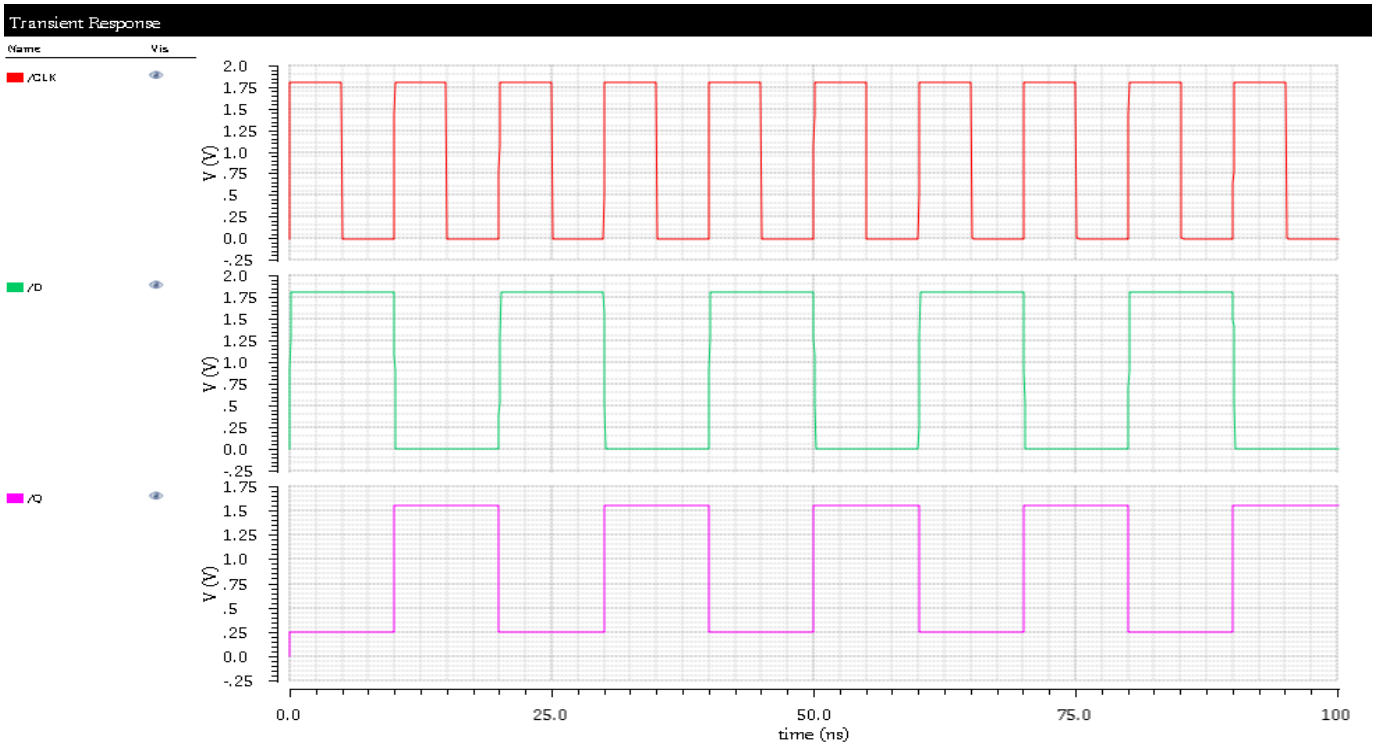
Verilog A:
D_Flip_Flop

```
// VerilogA for EE288, D_Flip_Flop, veriloga

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

module D_Flip_Flop(D,CLK,Q);

input D, CLK;

output Q;

parameter real clk_vth = 0.5, delay = 0, ttime = 1p;

real  out;

electrical D, CLK, Q;

analog begin

 @(cross(V(CLK) - clk_vth,+1)) begin

   out = V(D);

   end
V(Q) <+ transition(out,delay,ttime);

end

endmodule
```

Test Bench



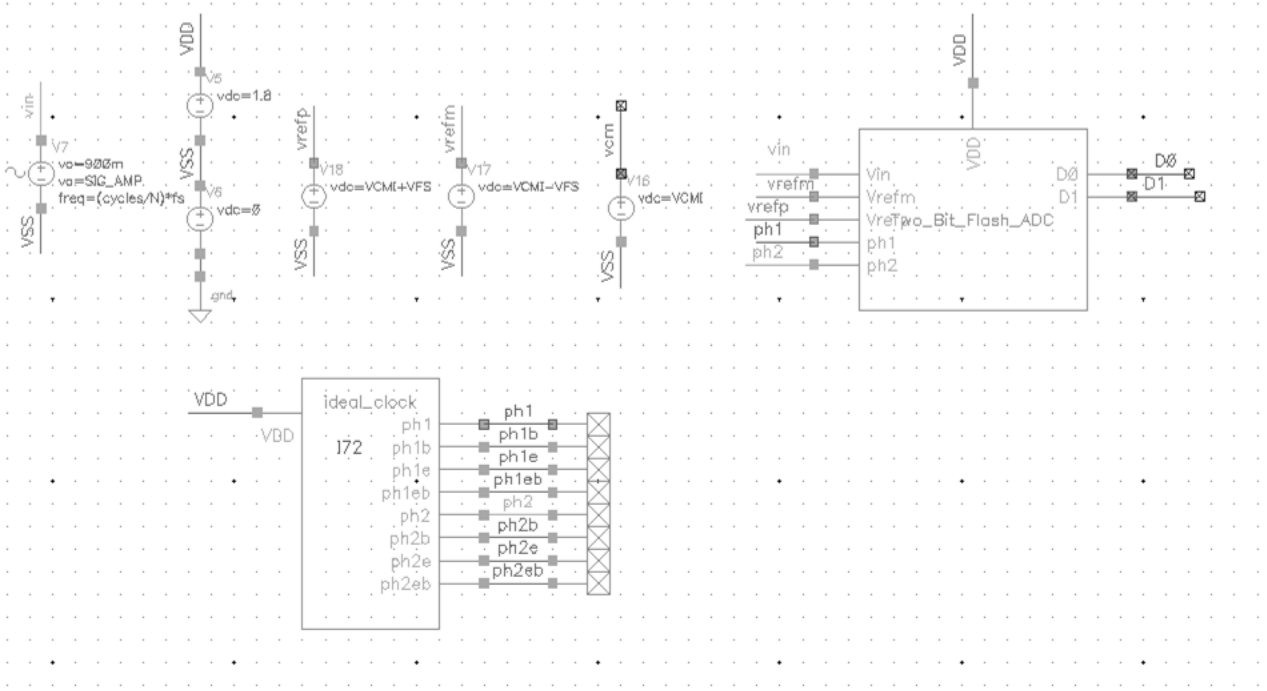
Simulation Results



Verilog A:
Two Bit Flash ADC

```
// VerilogA for ADC_Comparator, Two_Bit_Flash_ADC, veriloga
`include "constants.vams"
`include "disciplines.vams"
module Two_Bit_Flash_ADC(ph1, ph2, Vrefp, Vrefm,Vin, VDD, D0,D1 );

input ph1, ph2, Vrefp, Vrefm,Vin;

inout VDD;

output D0,D1;

electrical ph1, ph2, Vrefp, Vrefm,Vin, VDD, D0,D1 ;

parameter real clk_th = 0.5,delay=0, ttime = 1p;

real c1, c2, c3, d1, d0;

analog begin

   @(cross(V(ph1) - clk_th,+1)) begin

      if ((V(Vin)) > (V(Vrefp)-V(Vrefm))/2) begin
          c2 = V(VDD);
          c1 = V(VDD);
          c3 = V(VDD);
           d1 = 0;
         d0 = 0;
        end
      else begin
          if ((V(Vin)) > (V(Vrefp)-V(Vrefm))/4) begin
              c3 = 0;
              c2 = V(VDD);
              c1 = V(VDD);
             d1 = 0;
              d0 = 0;
              end
           if((V(Vin)) > (V(Vrefp)-V(Vrefm))/8) begin
              c3 = 0;
              c2 =0;
              c1 =V(VDD);
              d1 = 0;
              d0 = 0;
              end
    else begin
              c3 = 0;
              c2 = 0;
              c1 = 0;
             d1 = 0;
              d0 = 0;
           end
        end
     enD
  @(cross(V(ph2) - clk_th,+1)) begin
        if(c3 >= V(VDD)) begin
           d1 =V(VDD);
           d0 = V(VDD);
           end
        else begin
           if(c2 >= V(VDD)) begin
              d1 = V(VDD);
              d0 = 0;
              end
    else begin
              if (c1 >= V(VDD)) begin
                 d1 = 0;
                 d0 = V(VDD);
                 end
              else begin
                 d1 = 0;
                 d0 = 0;
                 end
           end
        end
     end
  V(D0) <+ transition(d0,delay,ttime);
  V(D1) <+ transition(d1,delay,ttime);
  end
  endmodule
```

Test Bench


VDD
V5
vdc=1.8
vin
V7
va=900m
va=SIG_AMP
freq={cycles/N}*fs
VSS
V6
vdc=0
gnd
vrefp
V18
vdc=VCMI+VFS
vrefm
V17
vdc=VCMI-VFS
vcm
V16
vdc=VCMI
Vin
Vrefm
vrefp
ph1
ph2
Two_Bit_Flash_ADC
DØ
D1
ideal_clock
I72
ph1b
ph1e
ph1eb
ph2b
ph2e
ph2eb

Simulation Results