EE288 – HW6 Report

1 stage of a 1.5 bit/stage Pipeline ADC

(using 45nm CMOS Technology)

Muhammad Aldacher

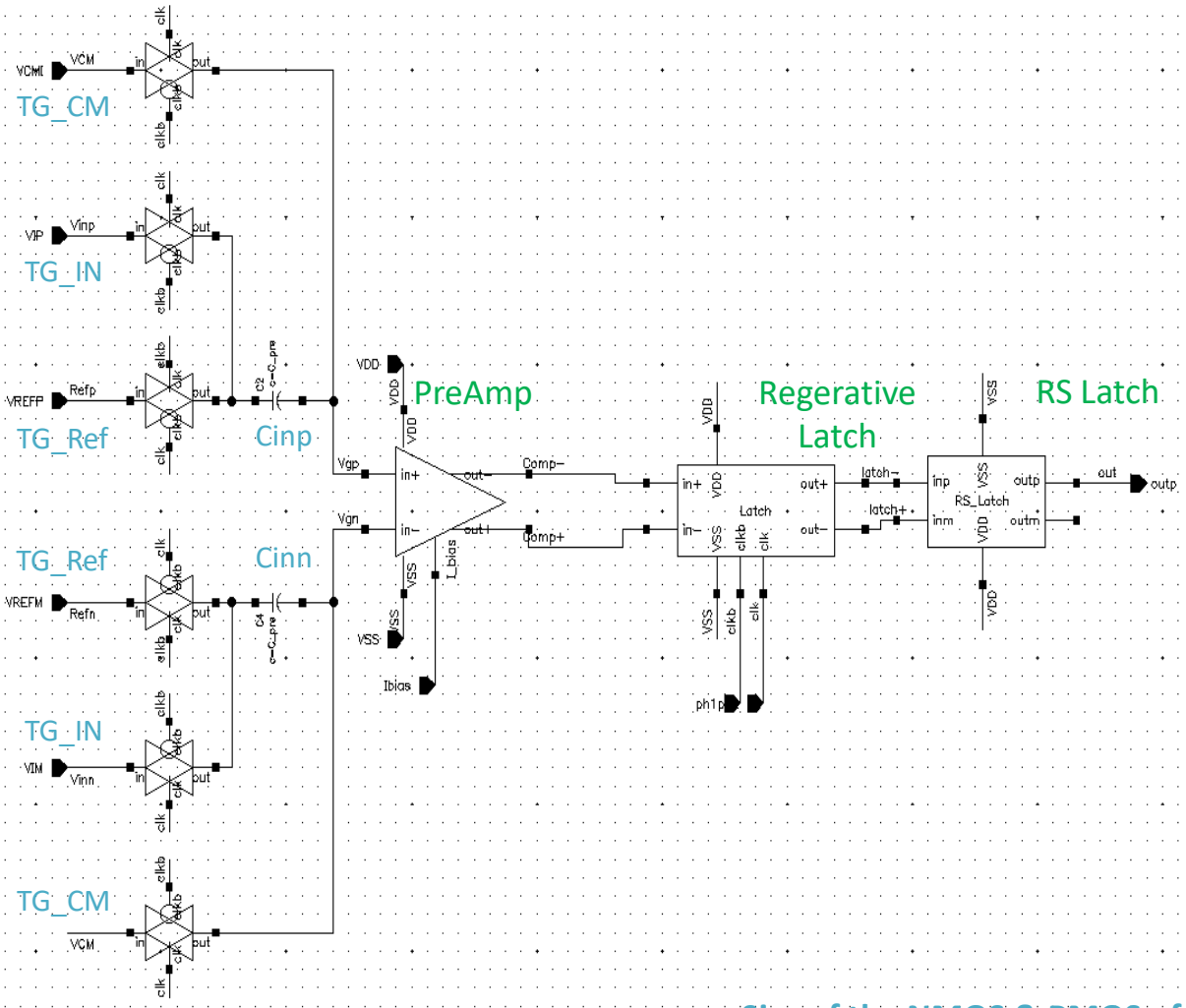
Student ID: 011510317

Overview

1. Comparator Circuit
2. Bootstrap & CMOS switches
3. Sub-ADC Logic Circuit
4. Whole System simulations
    a) Normal Mode
    b) 2x Gain Mode
5. VerilogA Blocks

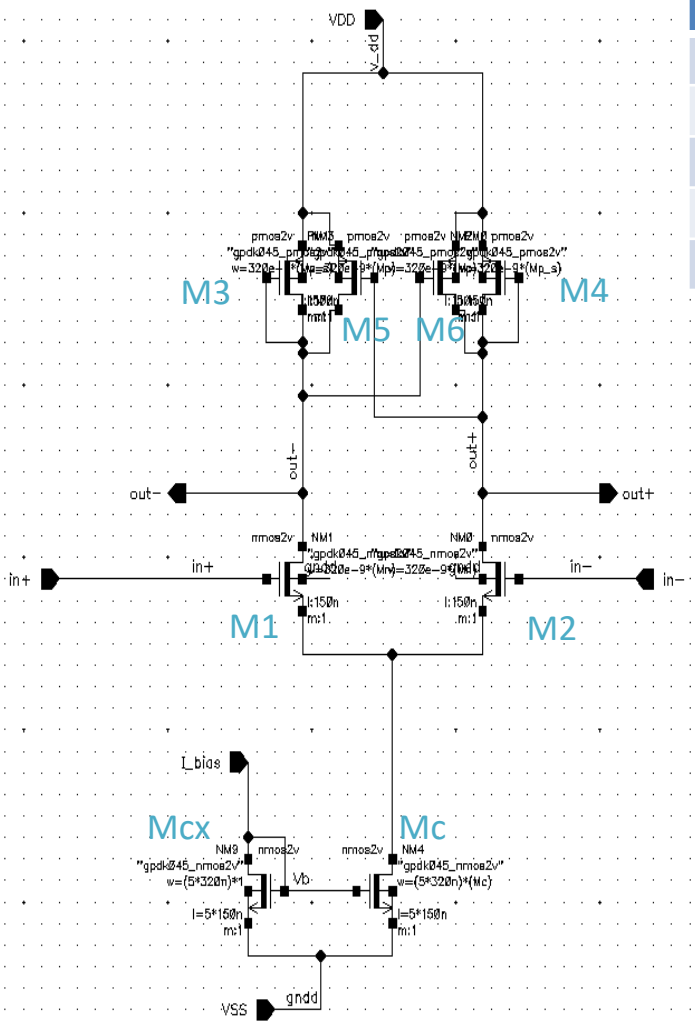
(1)
Comparator

Whole Comparator Circuit



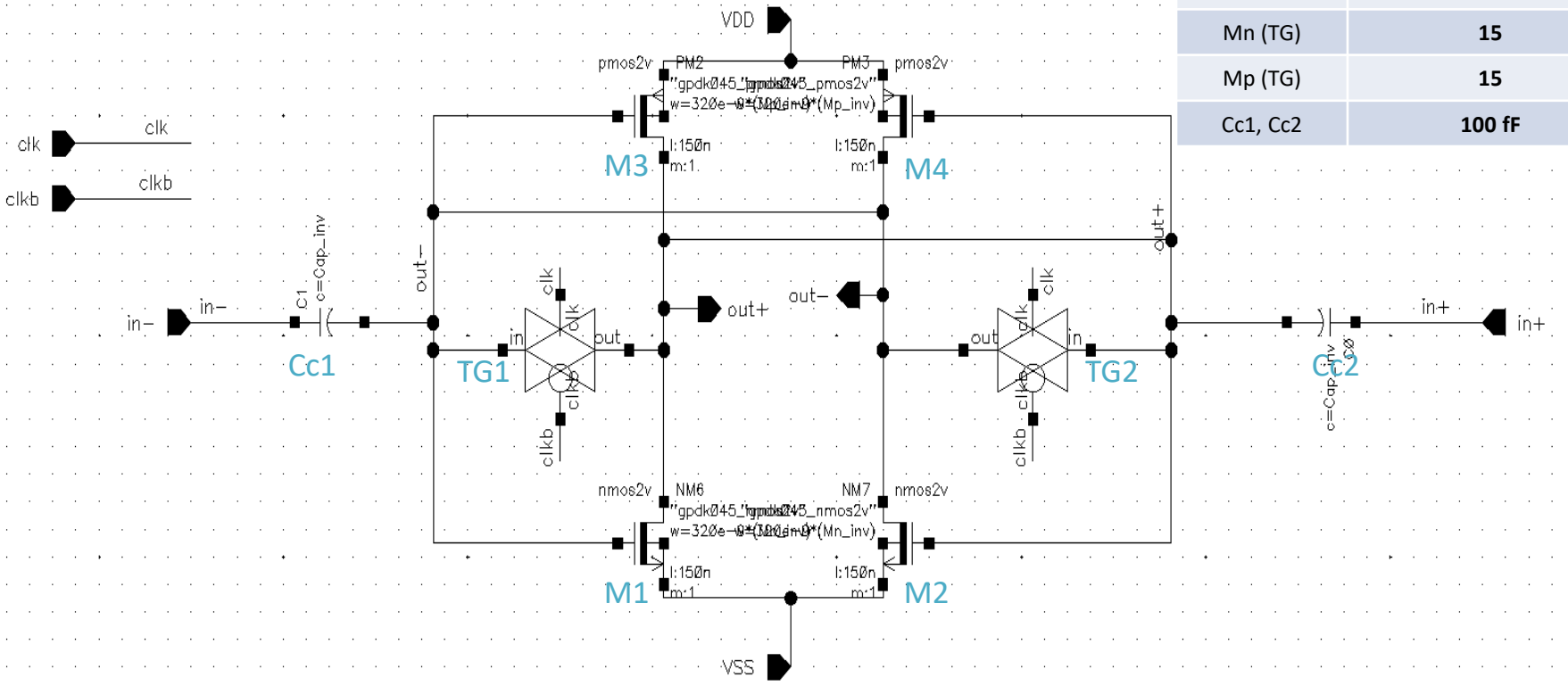
Size of the NMOS & PMOS of the TGs here (m) = 3

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**

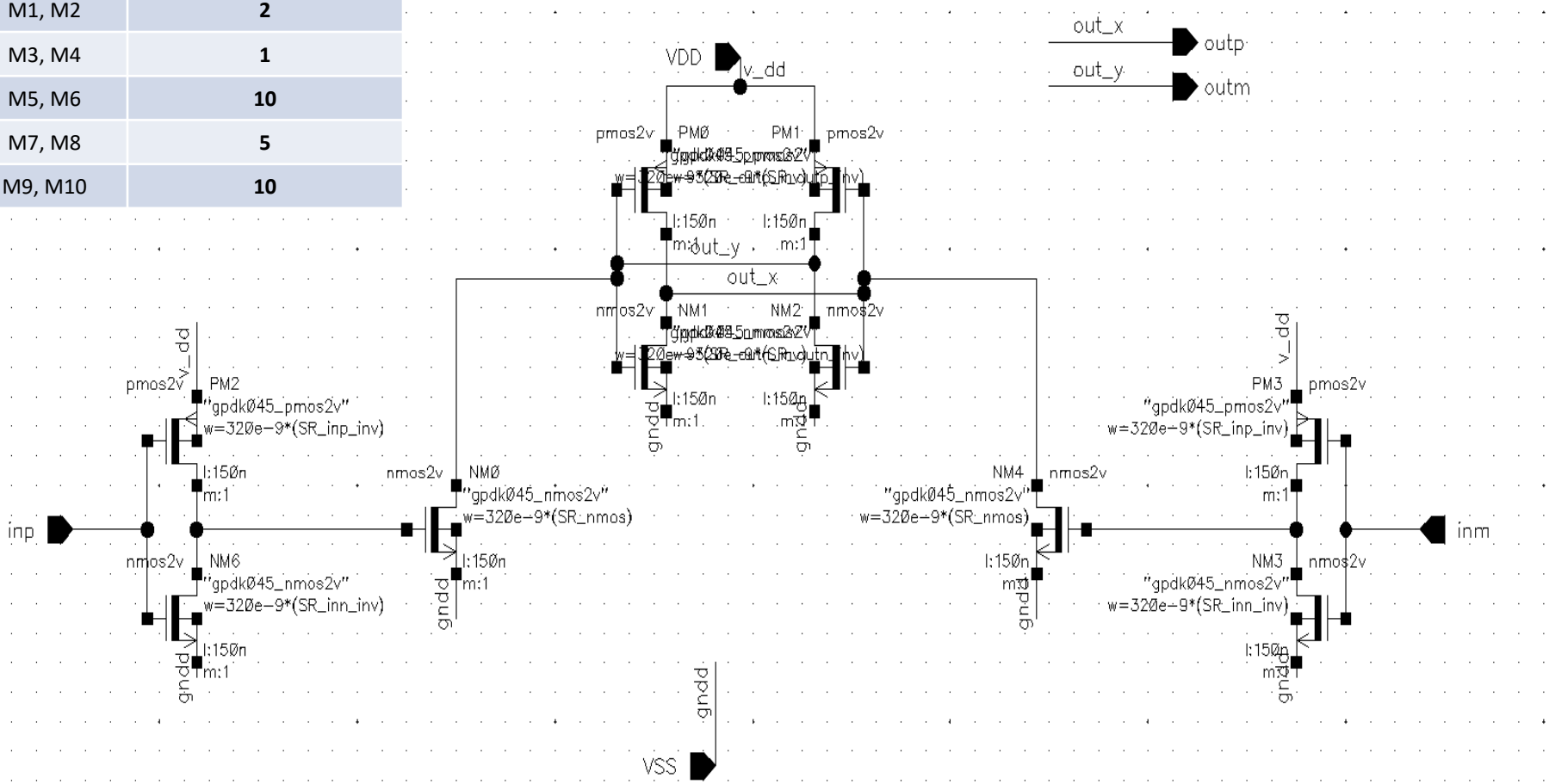
Regenerative Latch



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**

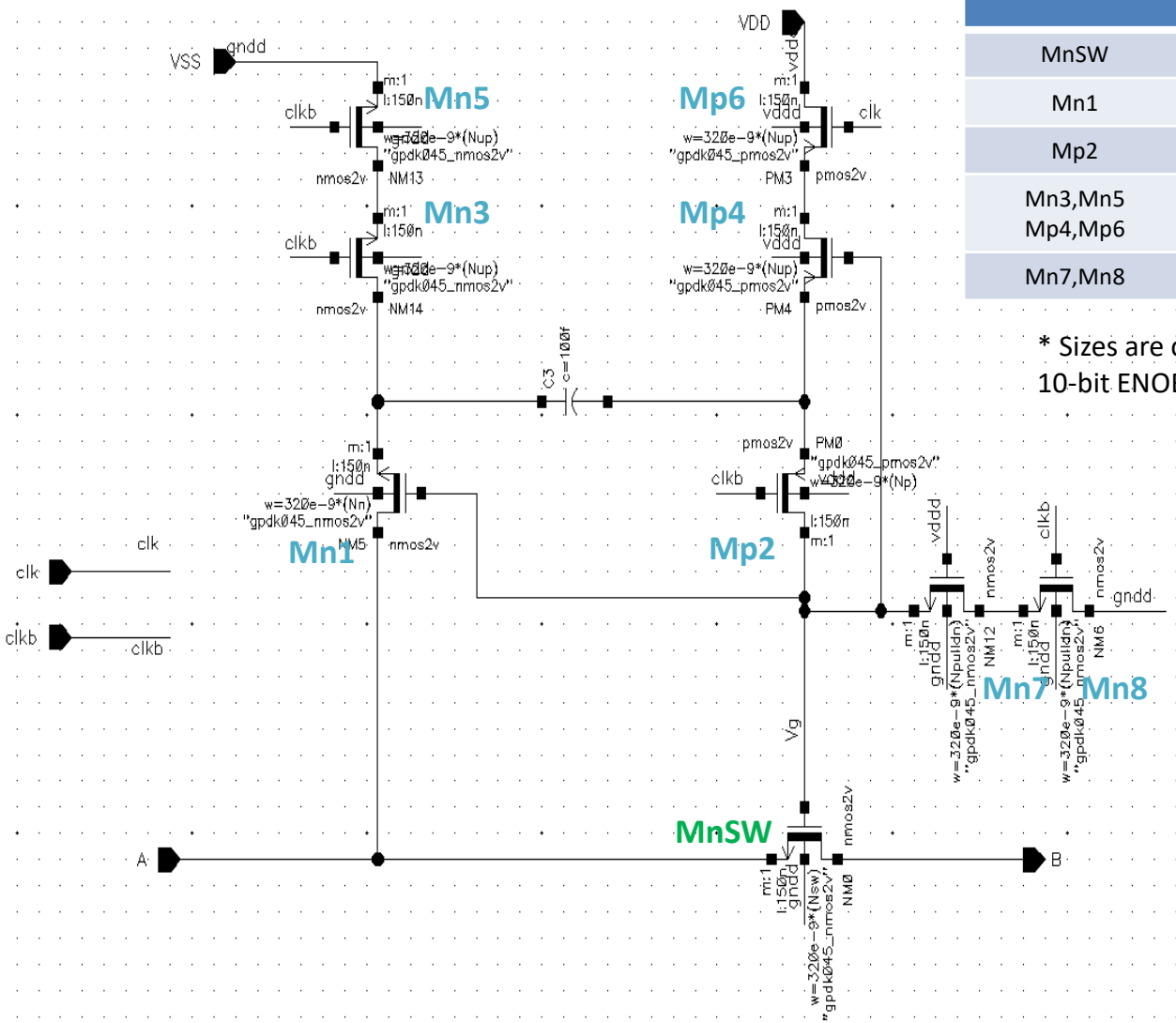


Overdrive test Waveforms

Diff Vin_min = 2 x 0.5 mV
(Smallest Resolution)

(2)
Switches Circuits

Bootstrap Switch

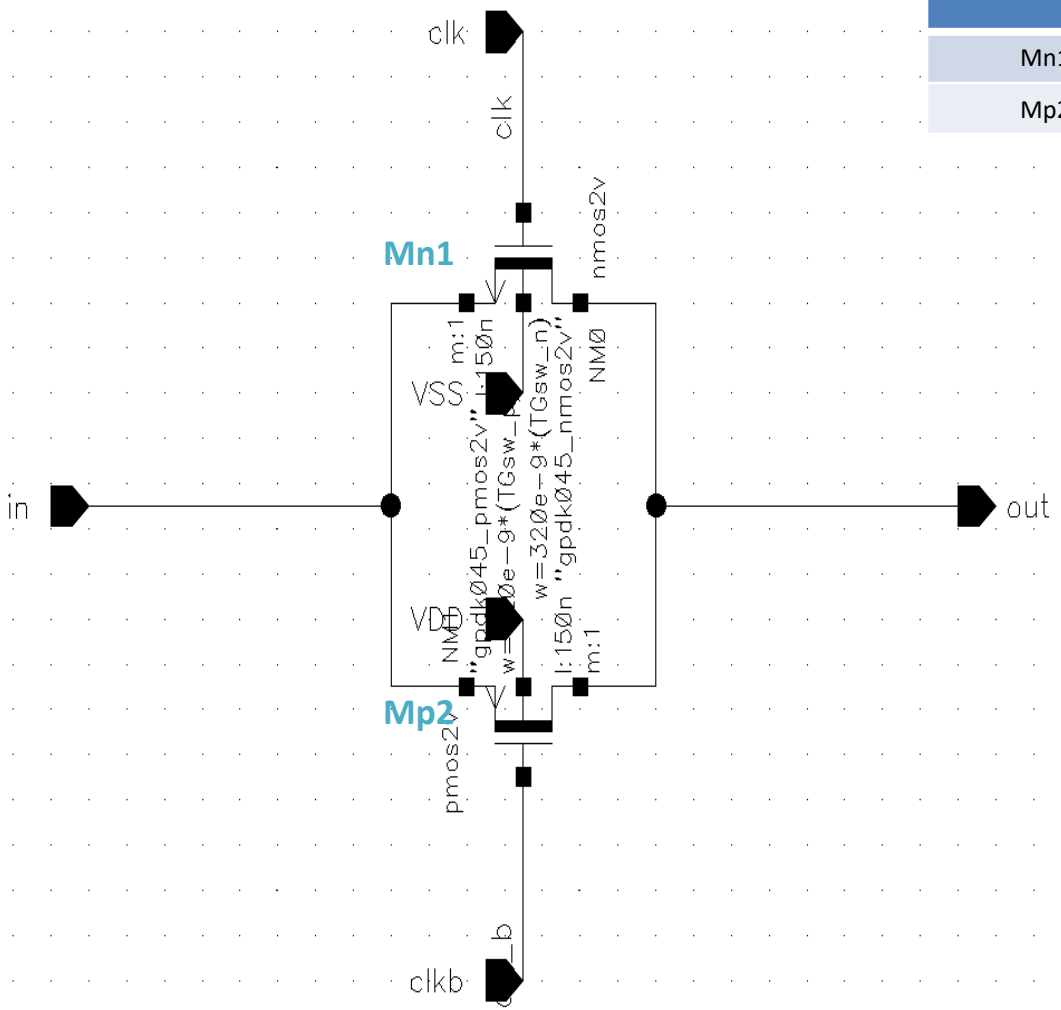


Device	Size (in terms of fingers, m)
MnSW	**30**
Mn1	**20**
Mp2	**1**
Mn3,Mn5 Mp4,Mp6	**1**
Mn7,Mn8	**28**

* Sizes are chosen to give a 10-bit ENOB for a 1pF load

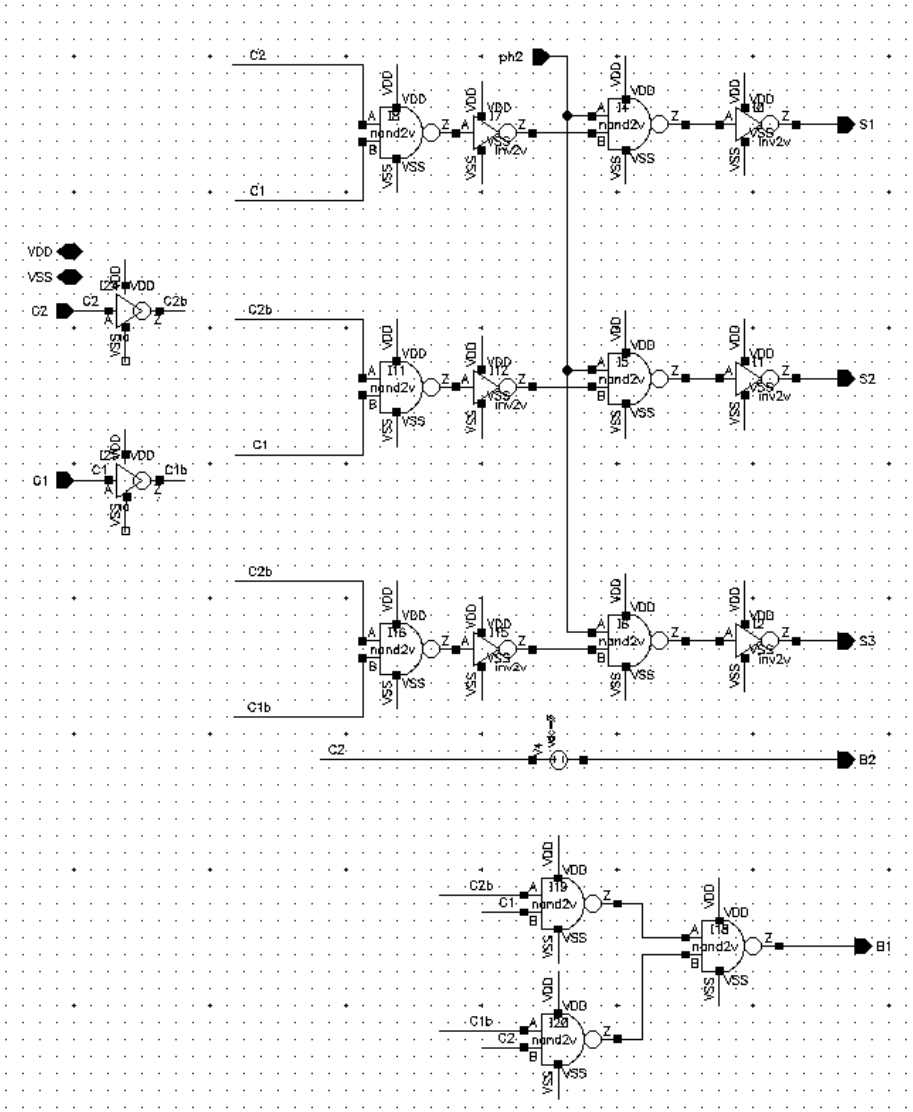
CMOS Transmission Gate Switch

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



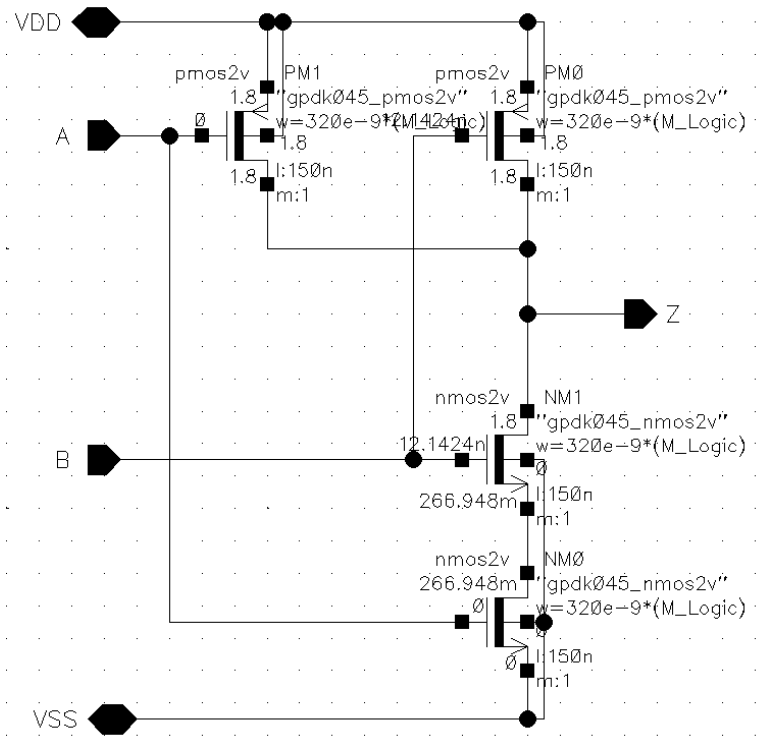
(3)
Sub-ADC Logic Circuit

Whole Logic Circuit

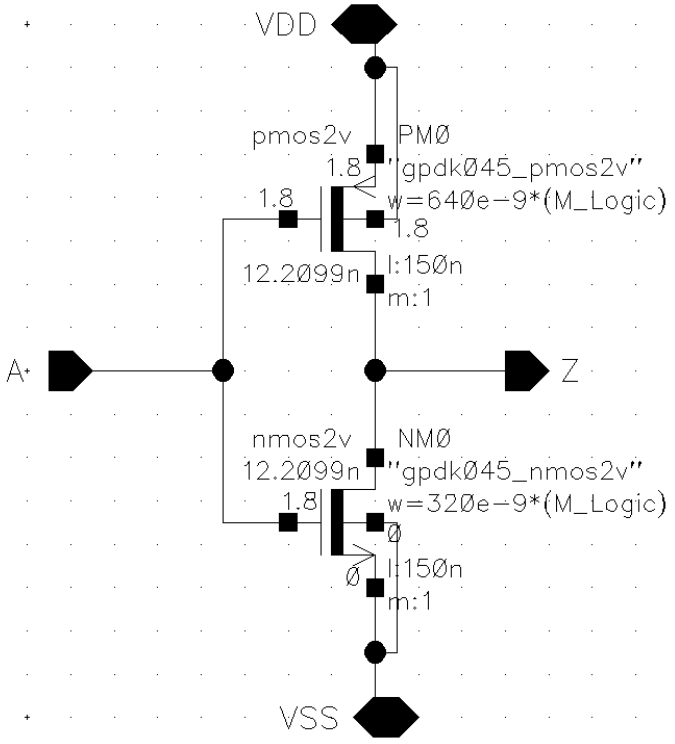


Logic Circuits used

NAND Gate Circuit



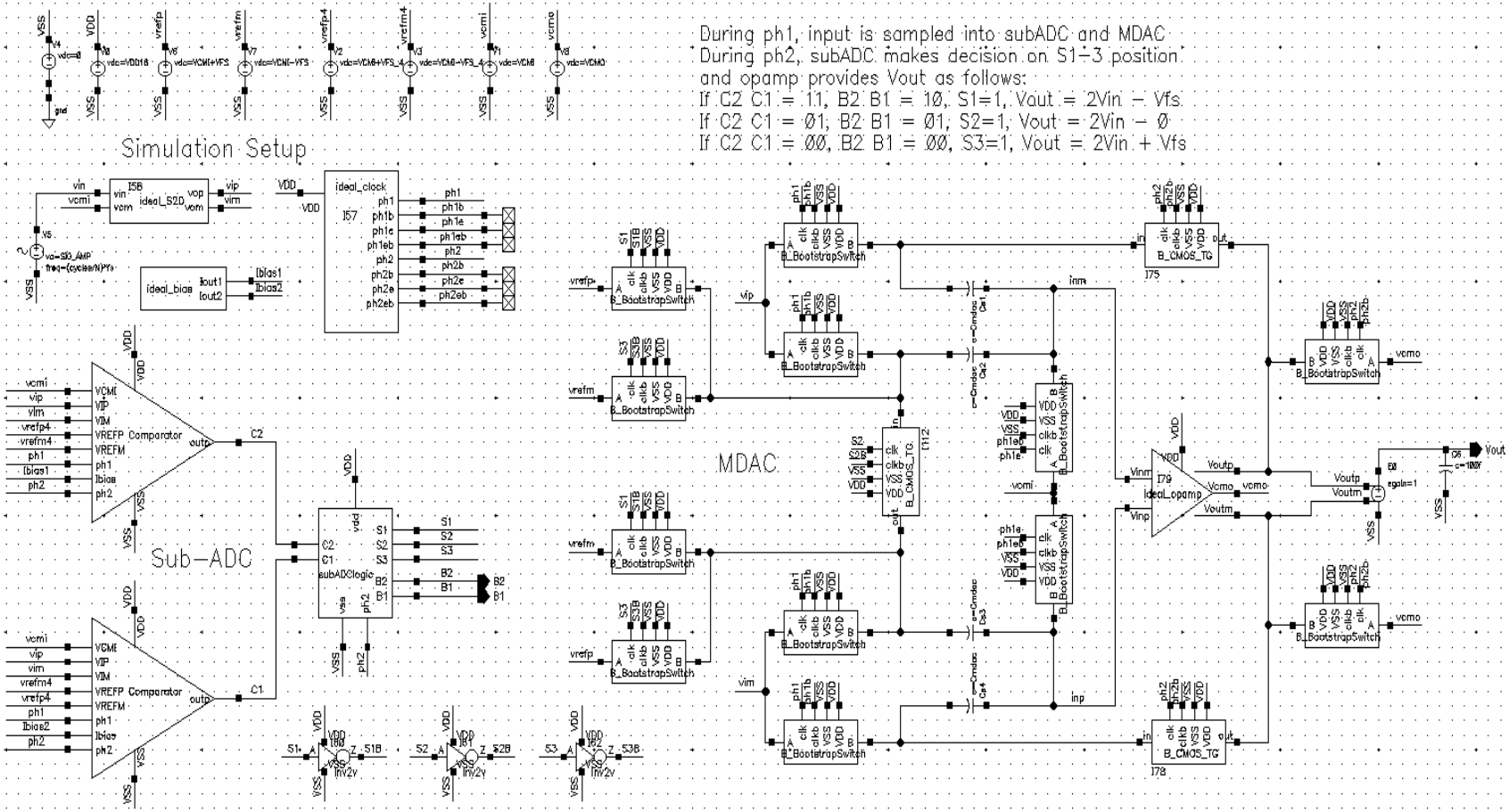
Inverter Circuit



M_Logic = 1

(4)
Whole System

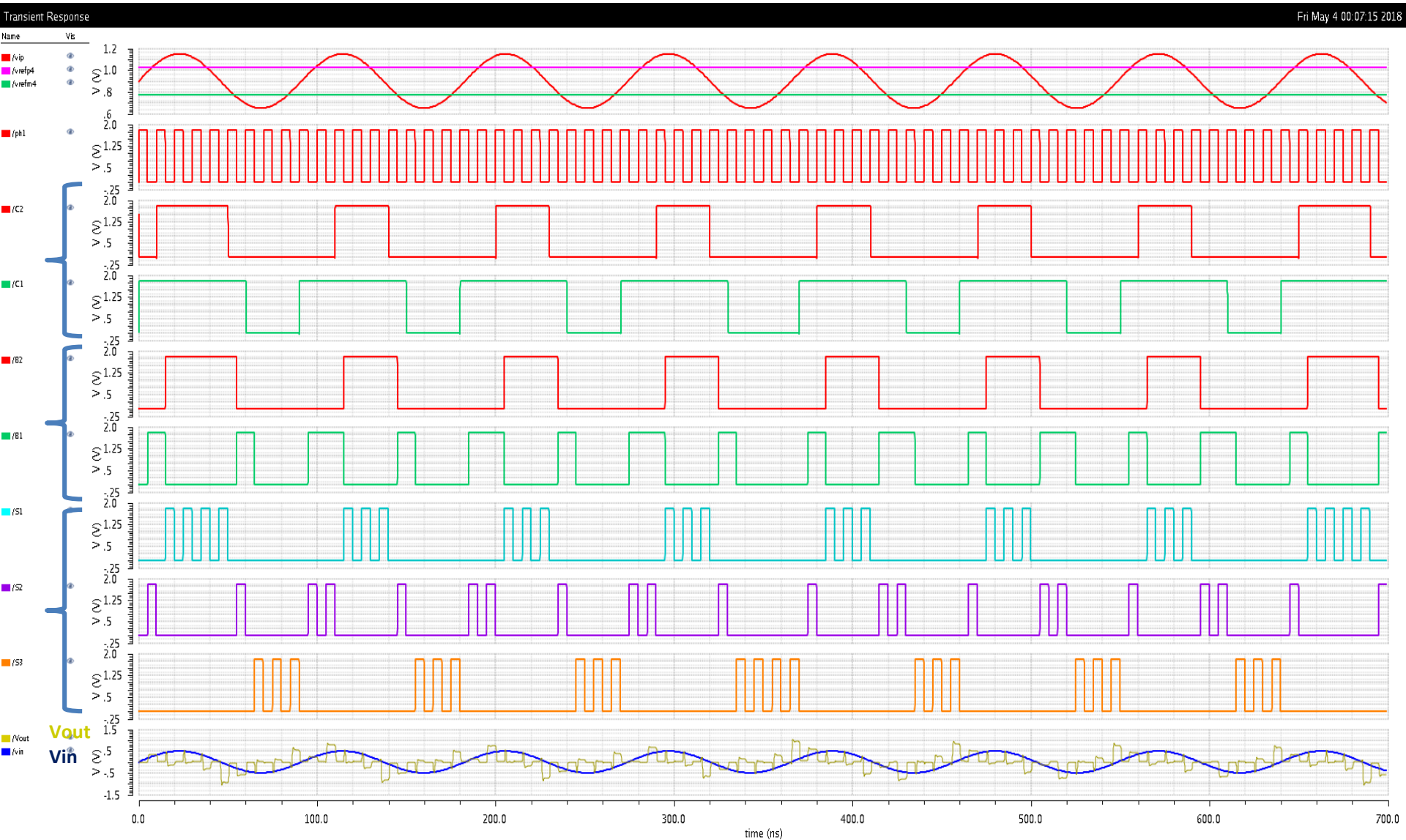
Testbench


During ph1, input is sampled into subADC and MDAC
During ph2, subADC makes decision on S1~3 position
and opamp provides Vout as follows:
If C2 C1 = 11, B2 B1 = 10, S1=1, Vout = 2Vin − Vfs
If C2 C1 = 01, B2 B1 = 01, S2=1, Vout = 2Vin − 0
If C2 C1 = 00, B2 B1 = 00, S3=1, Vout = 2Vin + Vfs
Simulation Setup
Sub-ADC
MDAC

Simulation Results
(a) Normal Mode

Simulation Results

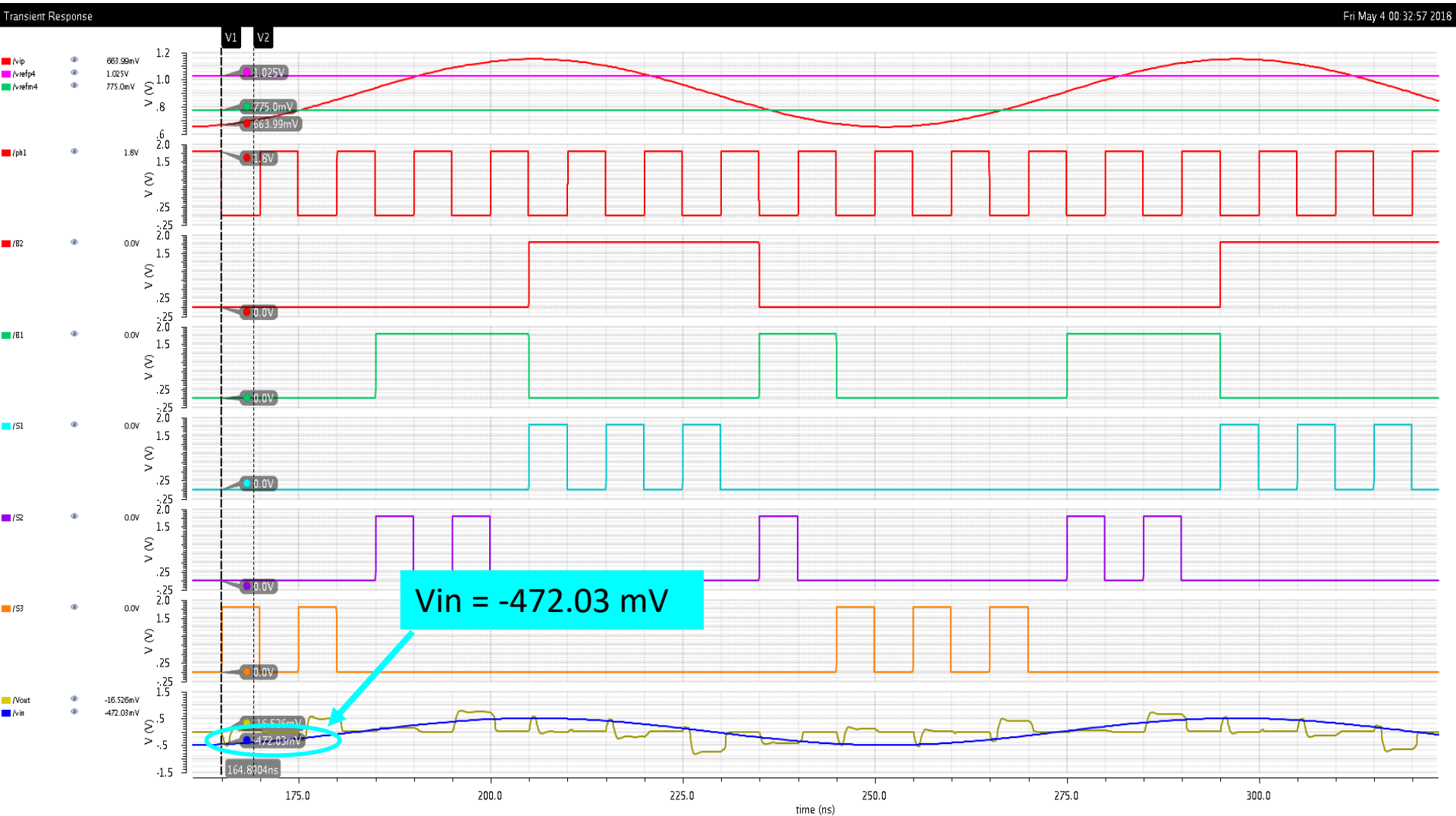
1- Normal Mode



Simulation Results

1- Normal Mode

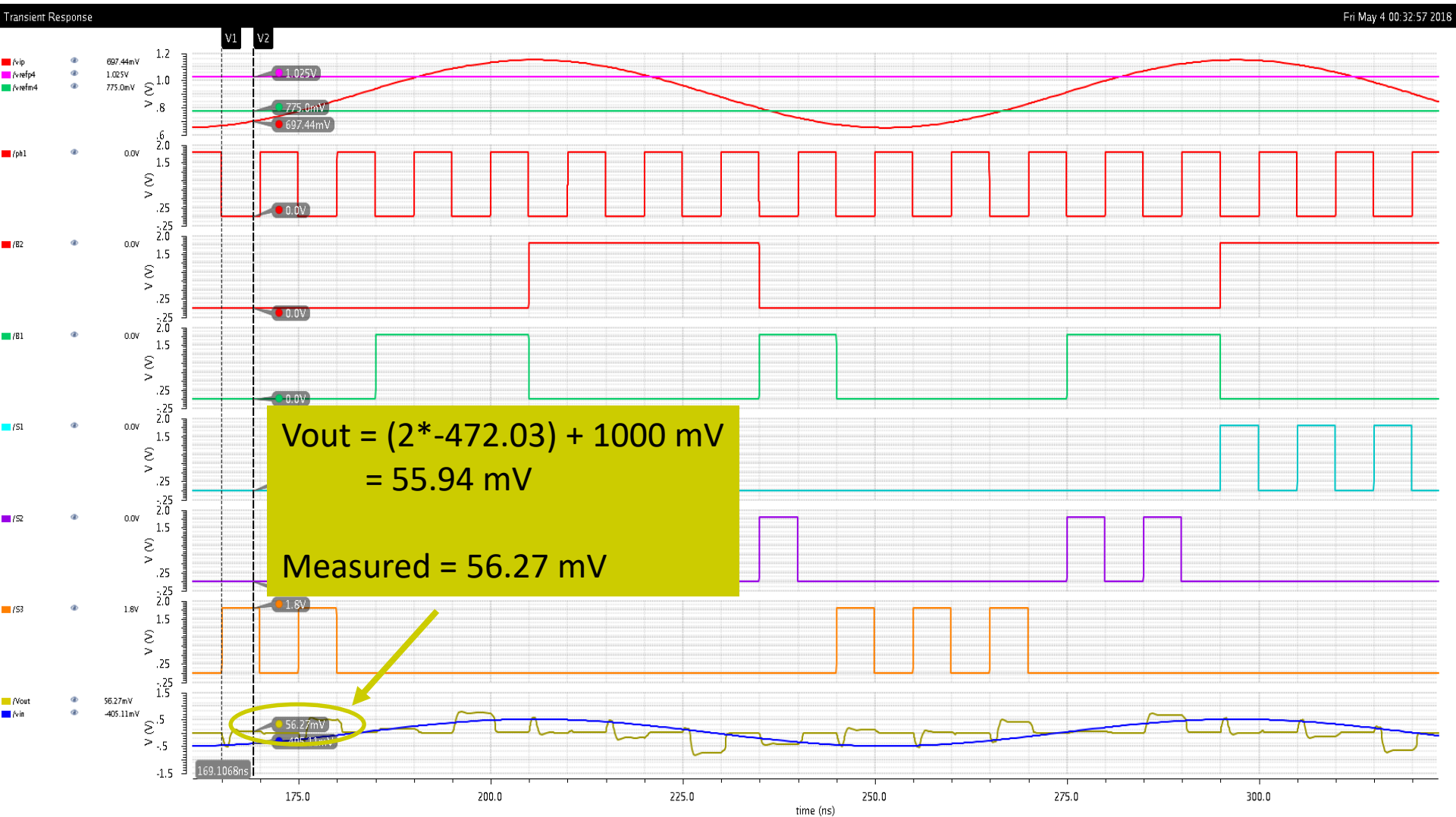
1) If Vin < Vrefm ➔ Vout = 2*Vin + Vfs



Simulation Results

1- Normal Mode

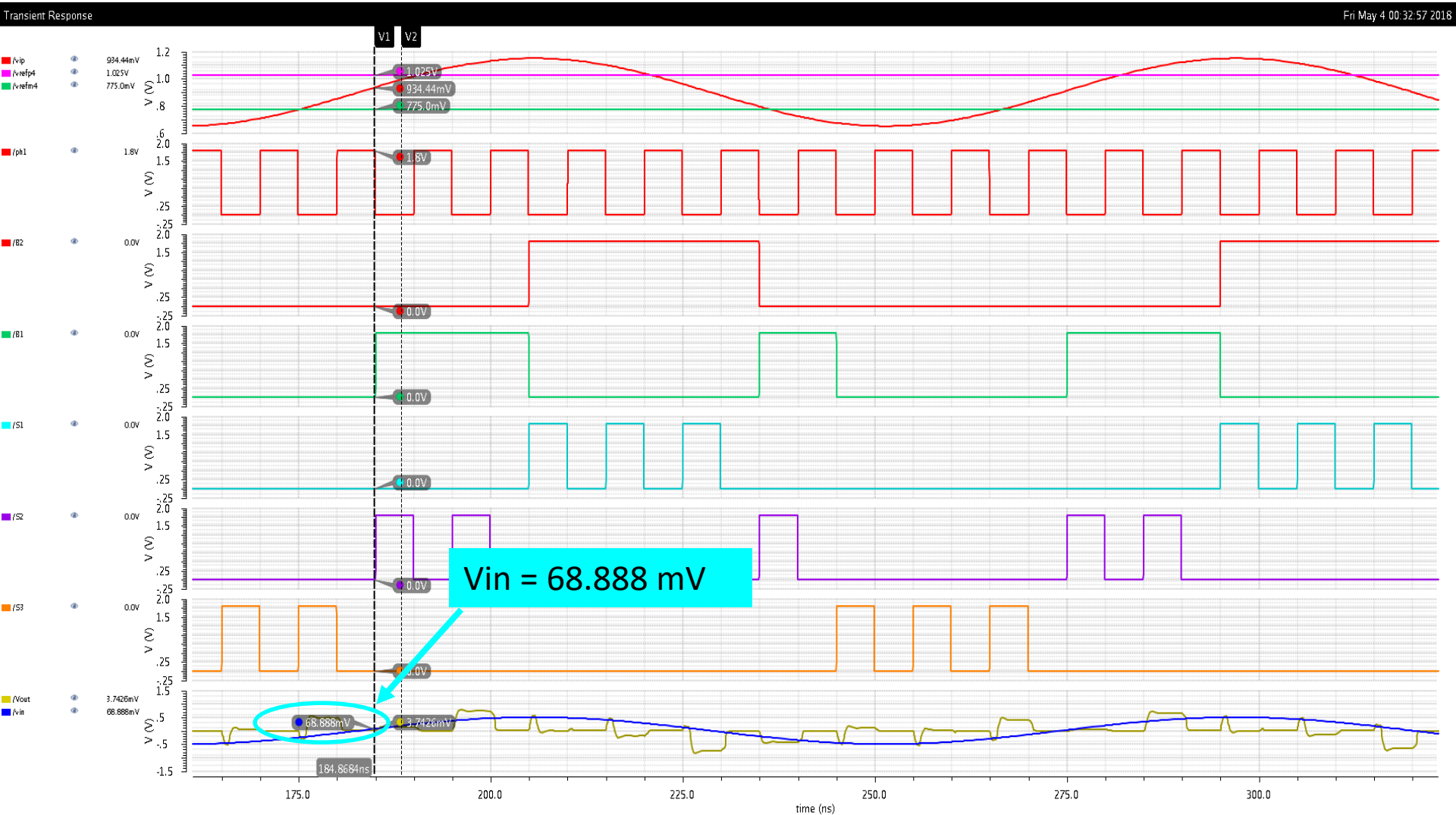
1) If Vin < Vrefm ➔ Vout = 2*Vin + Vfs



Simulation Results

1- Normal Mode

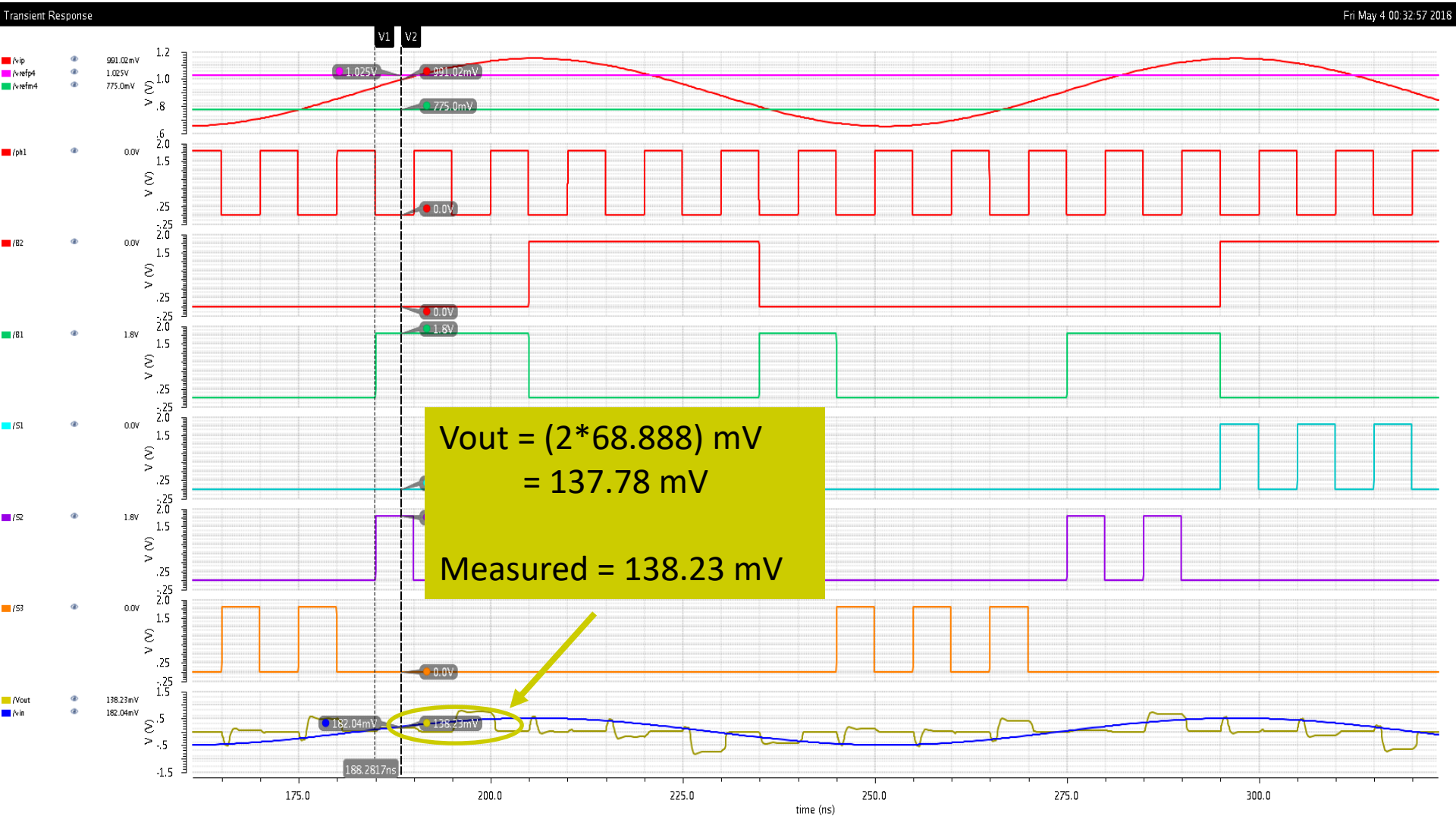
2) If Vrefm < Vin < Vrefp ➔ Vout = 2*Vin



Simulation Results

1- Normal Mode

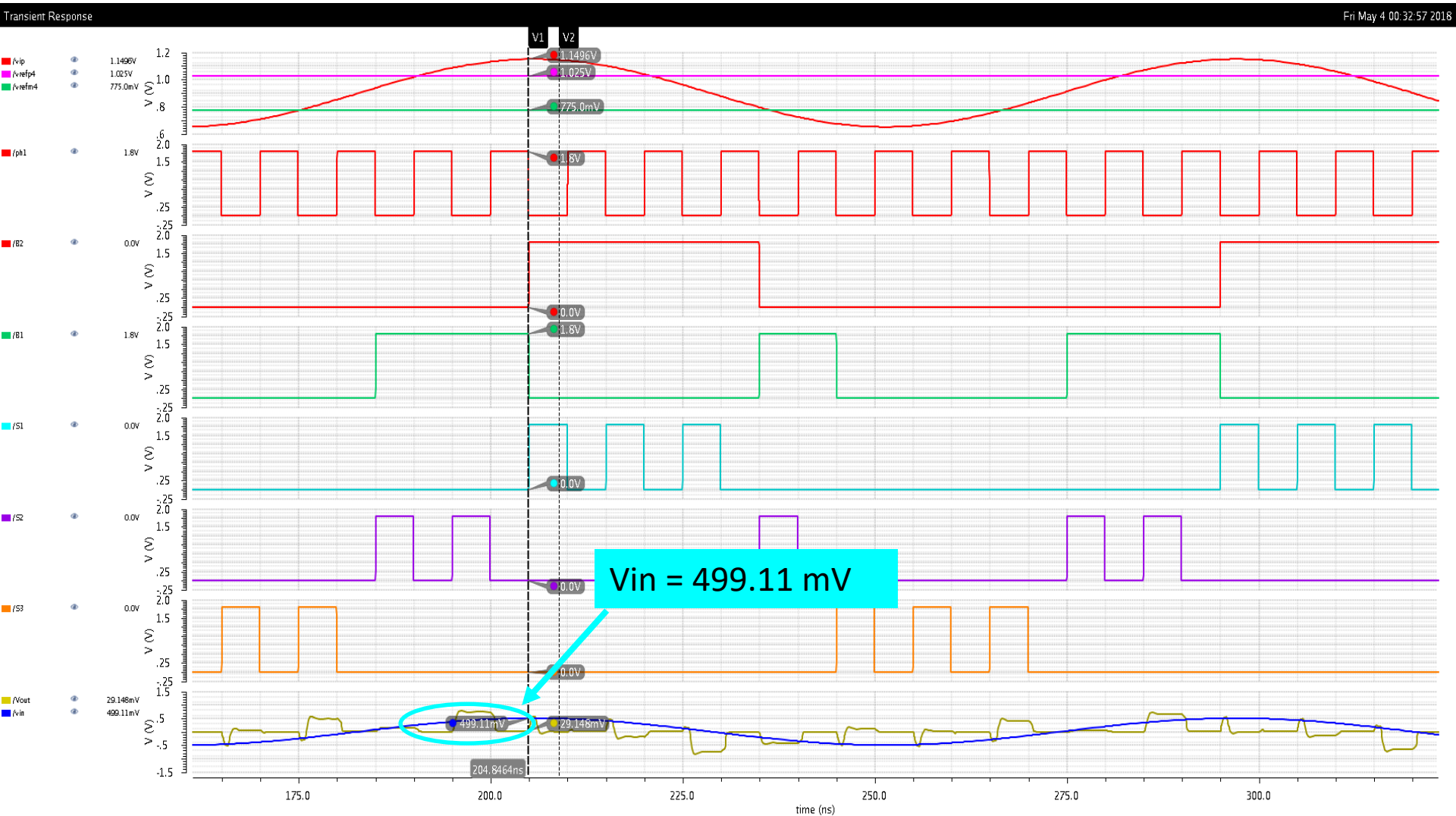
2) If Vrefm < Vin < Vrefp ➔ Vout = 2*Vin



Simulation Results

1- Normal Mode

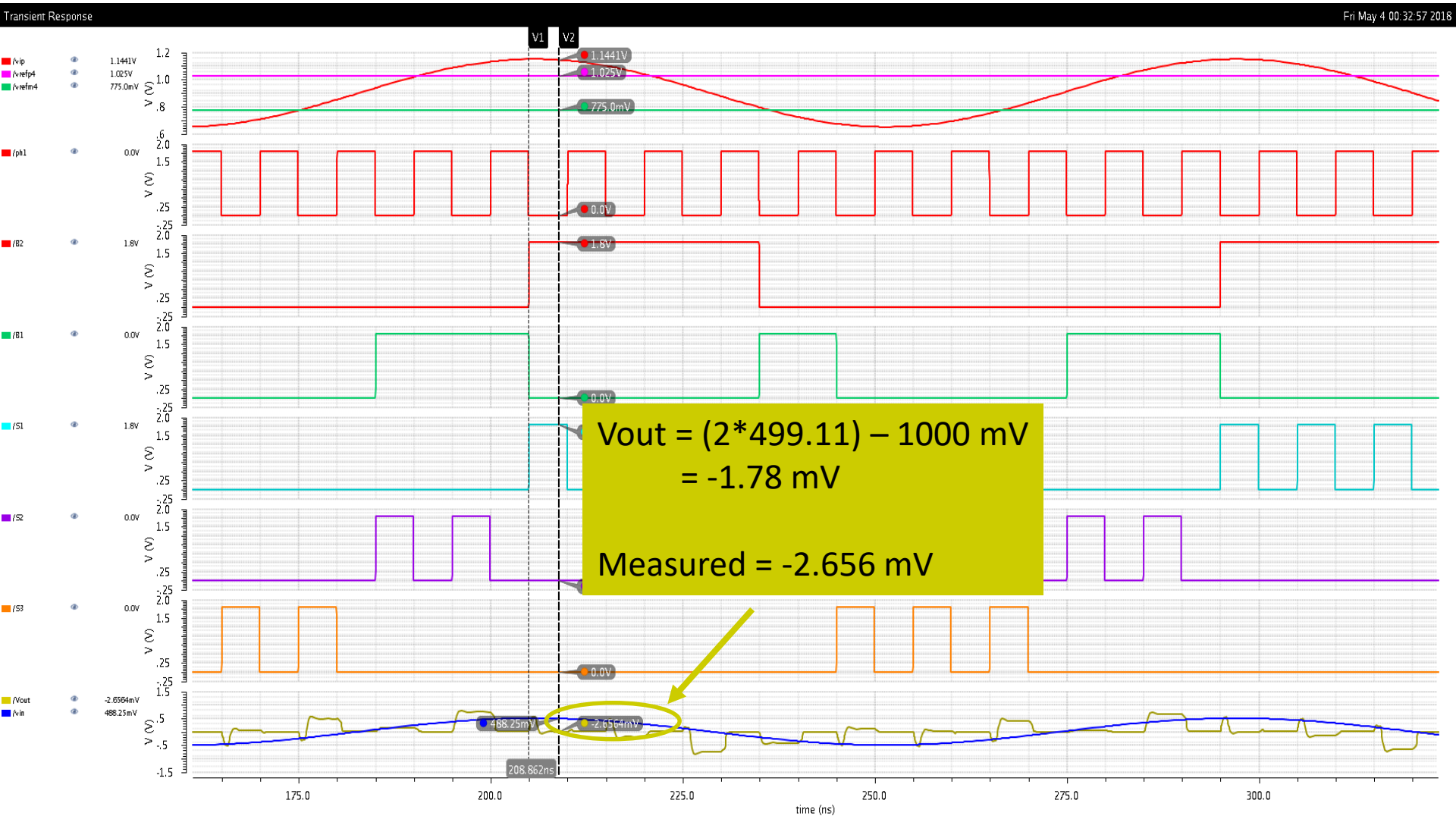
3) If Vin > Vrefp ➔ Vout = 2*Vin - Vfs



Simulation Results

1- Normal Mode

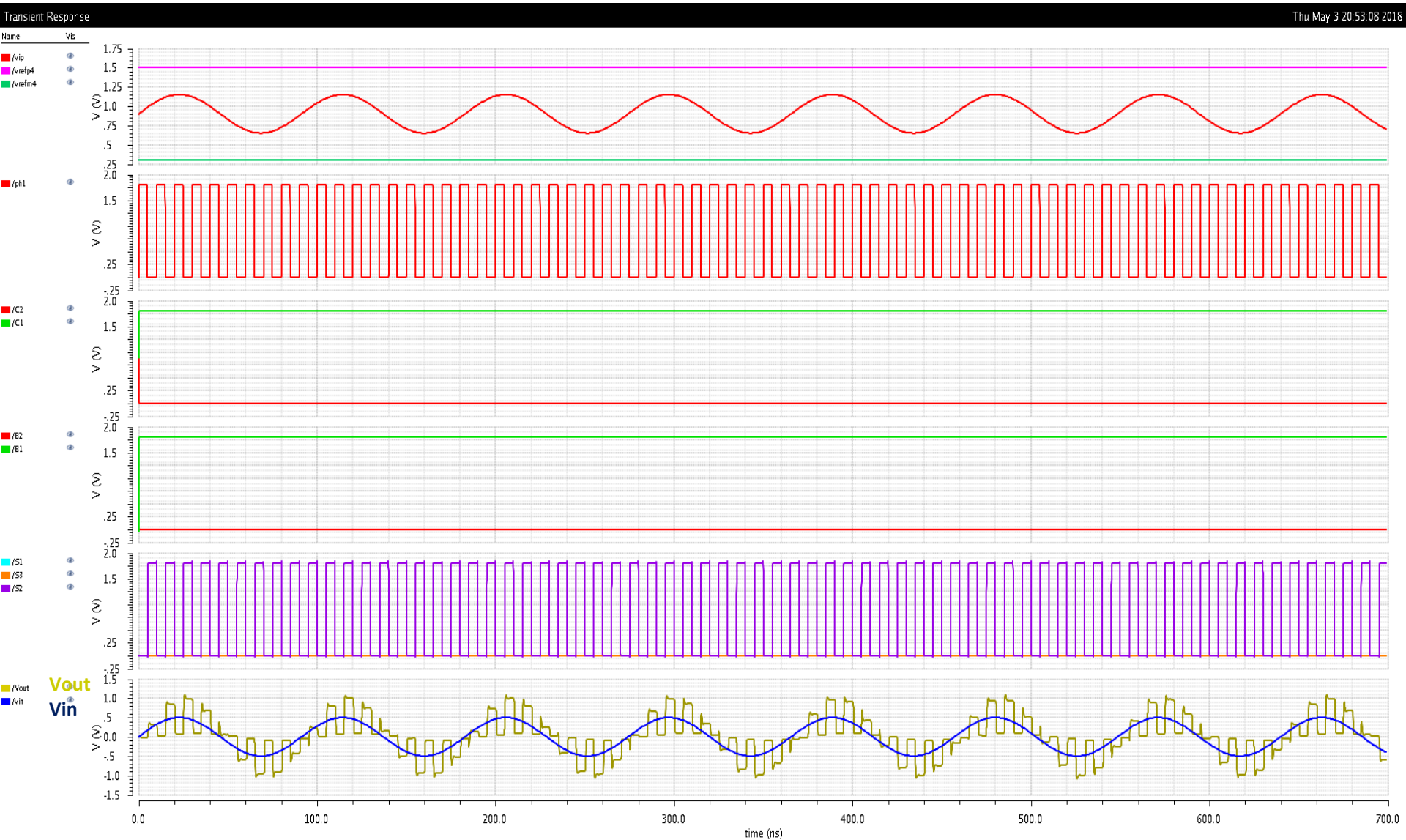
3) If Vin > Vrefp ➔ Vout = 2*Vin - Vfs



Simulation Results
(b) 2x Gain Mode

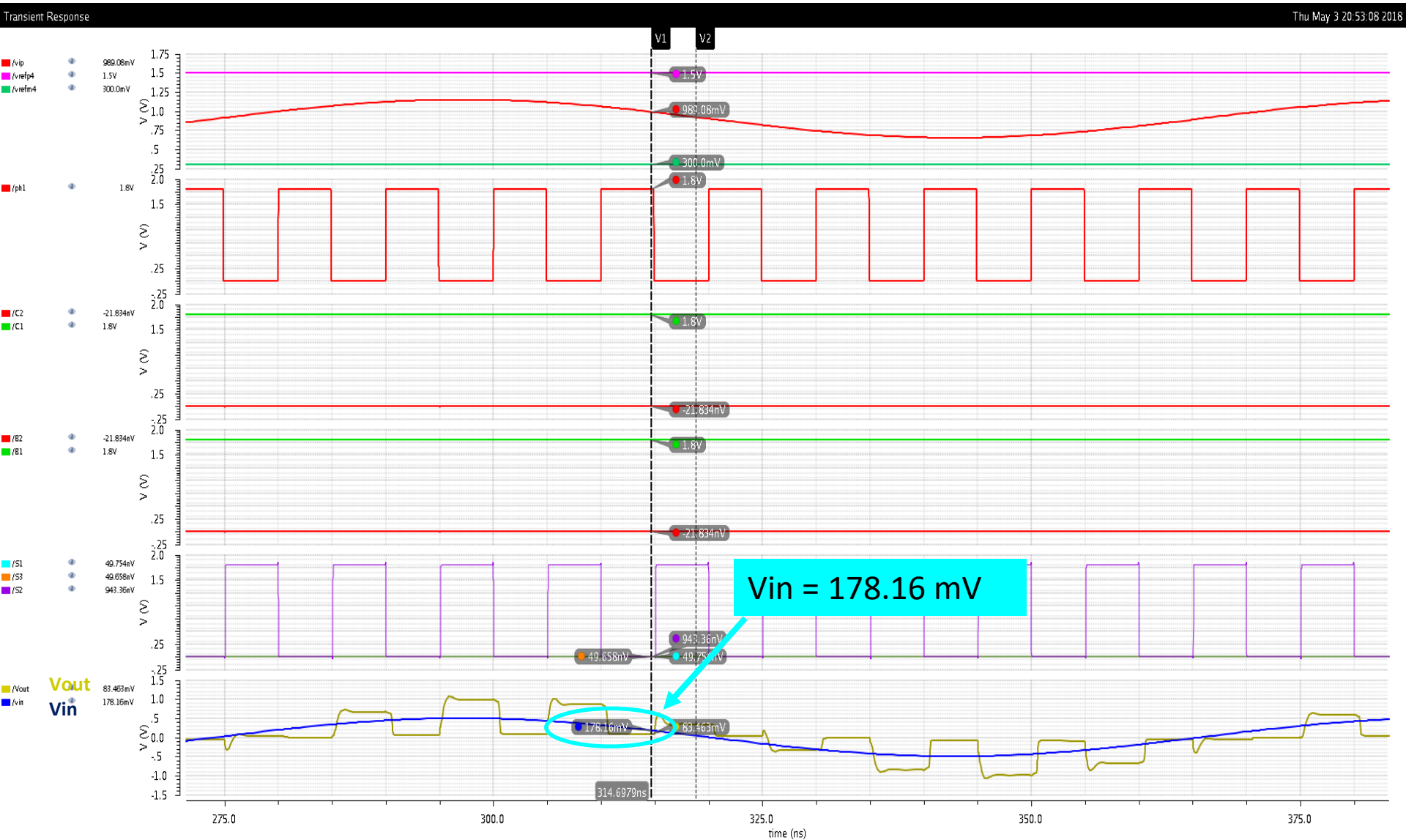
Simulation Results

2- 2x Gain Mode (Signal input Amp = 0.5v)



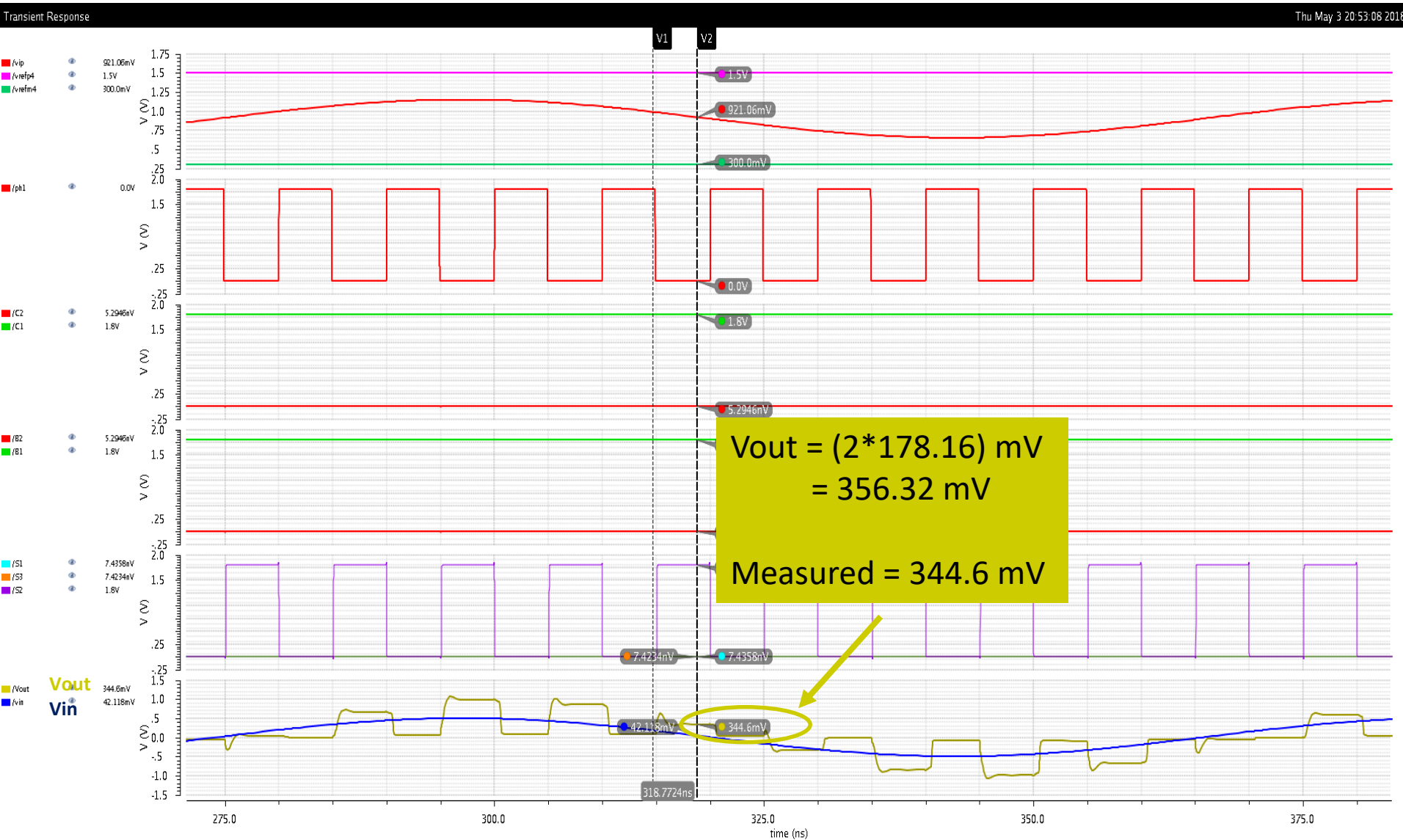
Simulation Results

2- 2x Gain Mode (Signal input Amp = 0.5v)



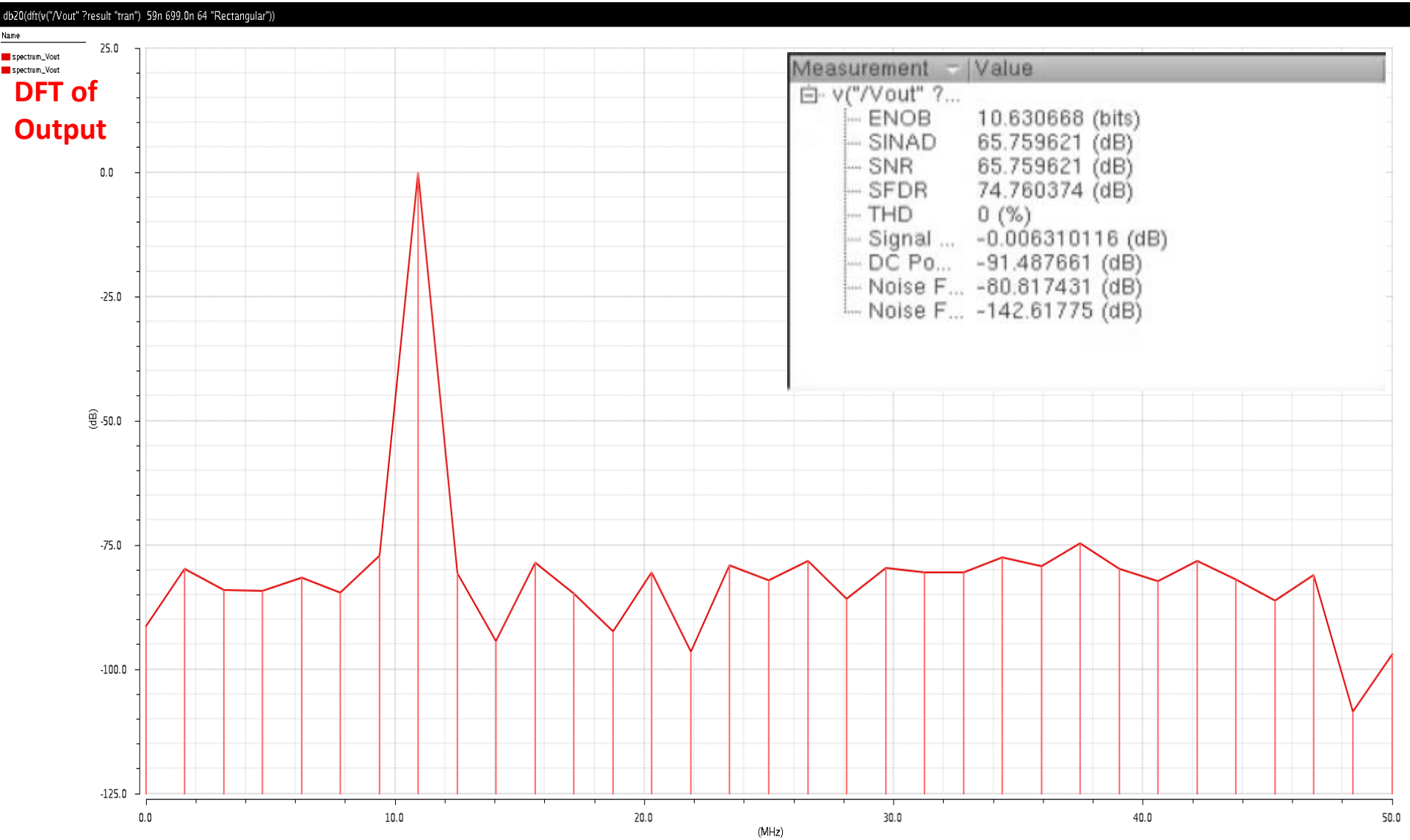
Simulation Results

2- 2x Gain Mode (Signal input Amp = 0.5v)



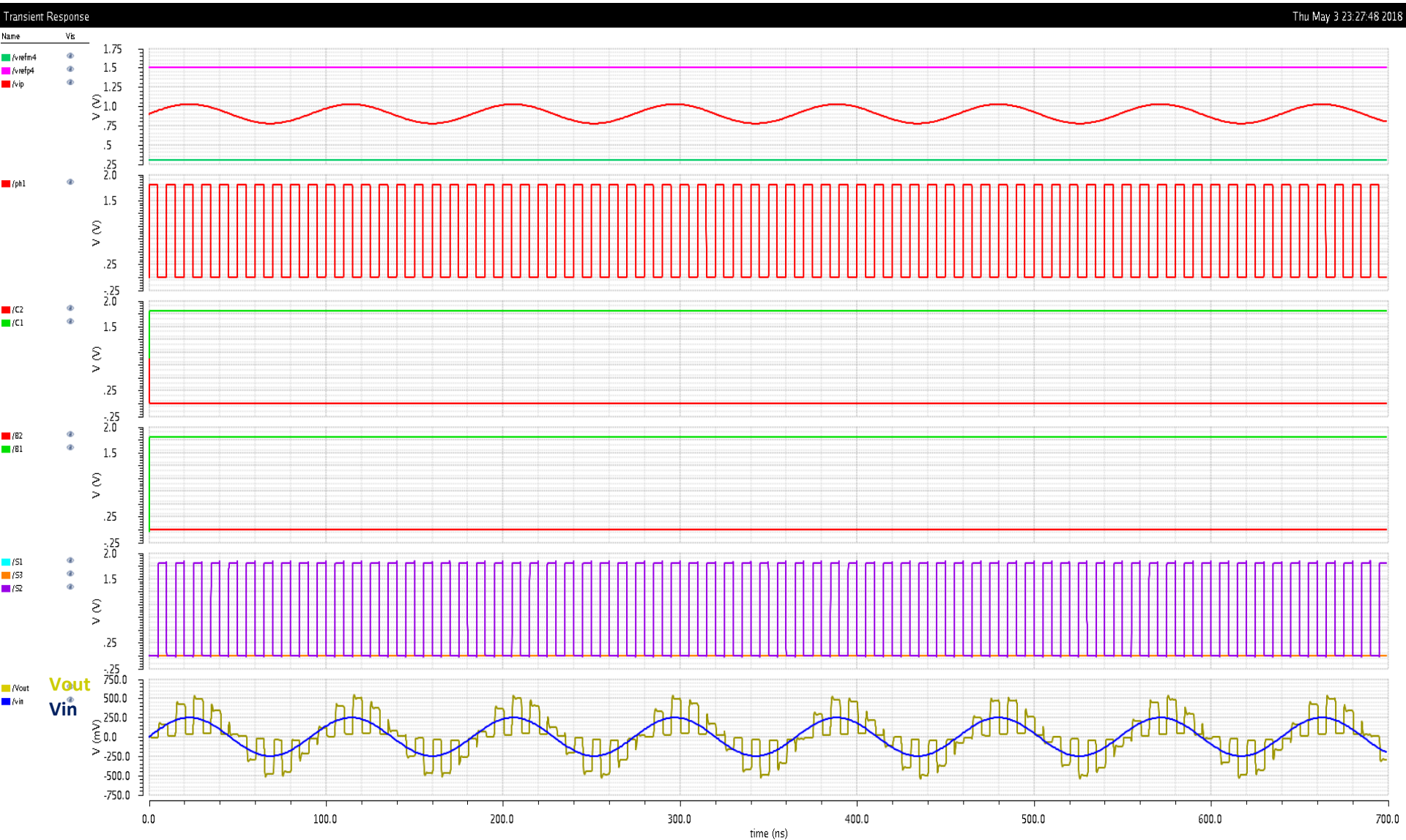
Simulation Results

2- 2x Gain Mode (Signal input Amp = 0.5v)



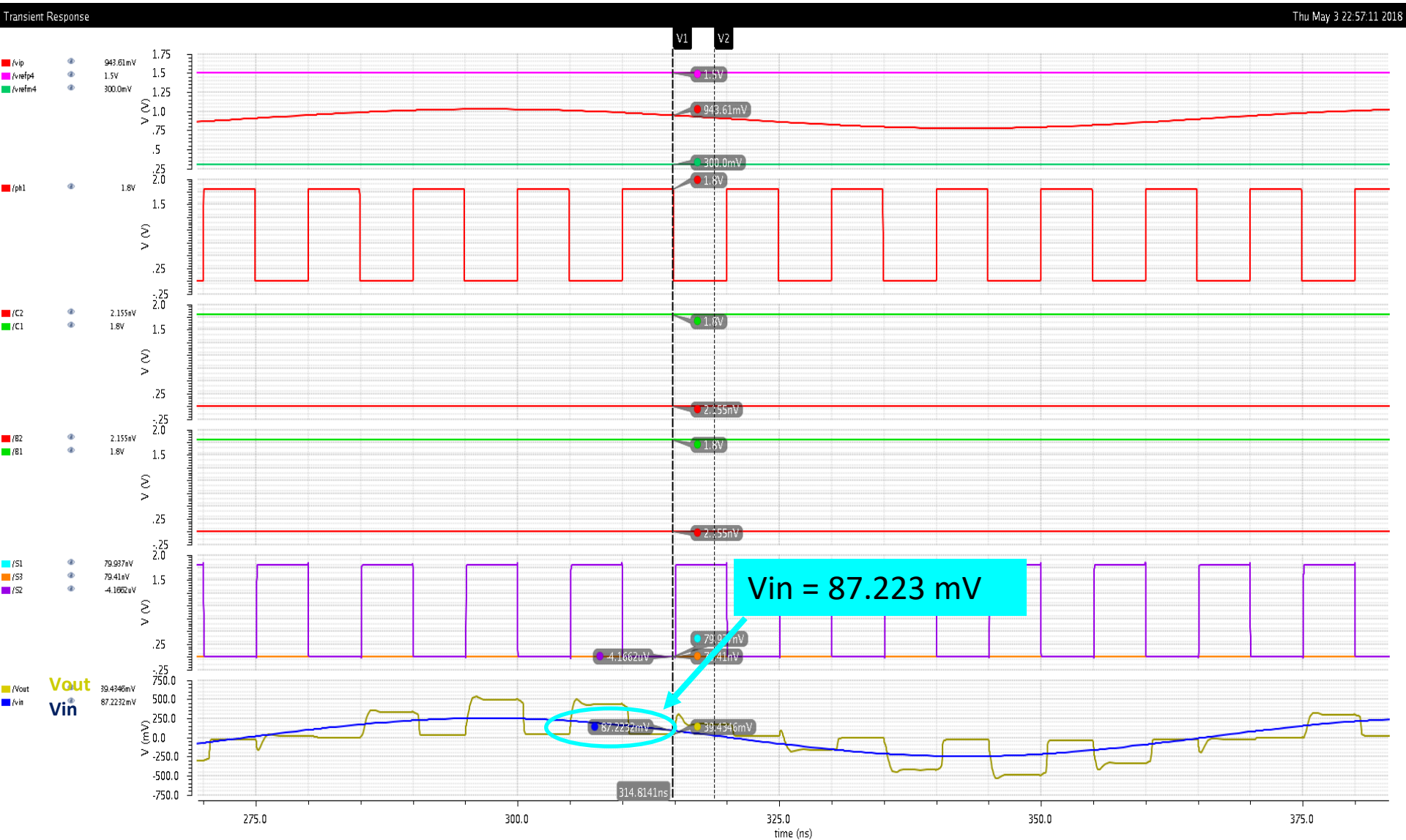
Simulation Results

2- 2x Gain Mode (Signal input Amp = 0.25v)



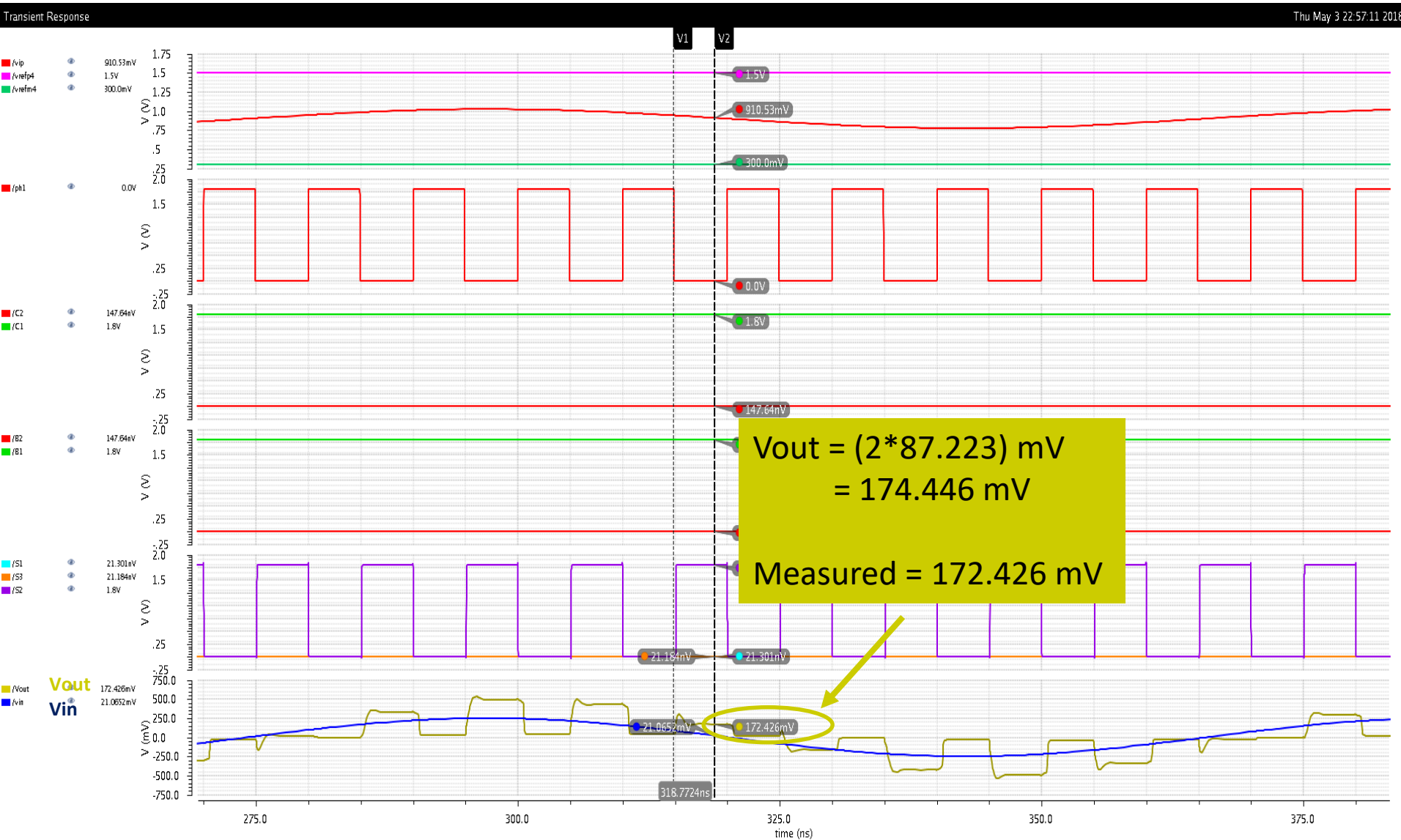
Simulation Results

2- 2x Gain Mode (Signal input Amp = 0.25v)



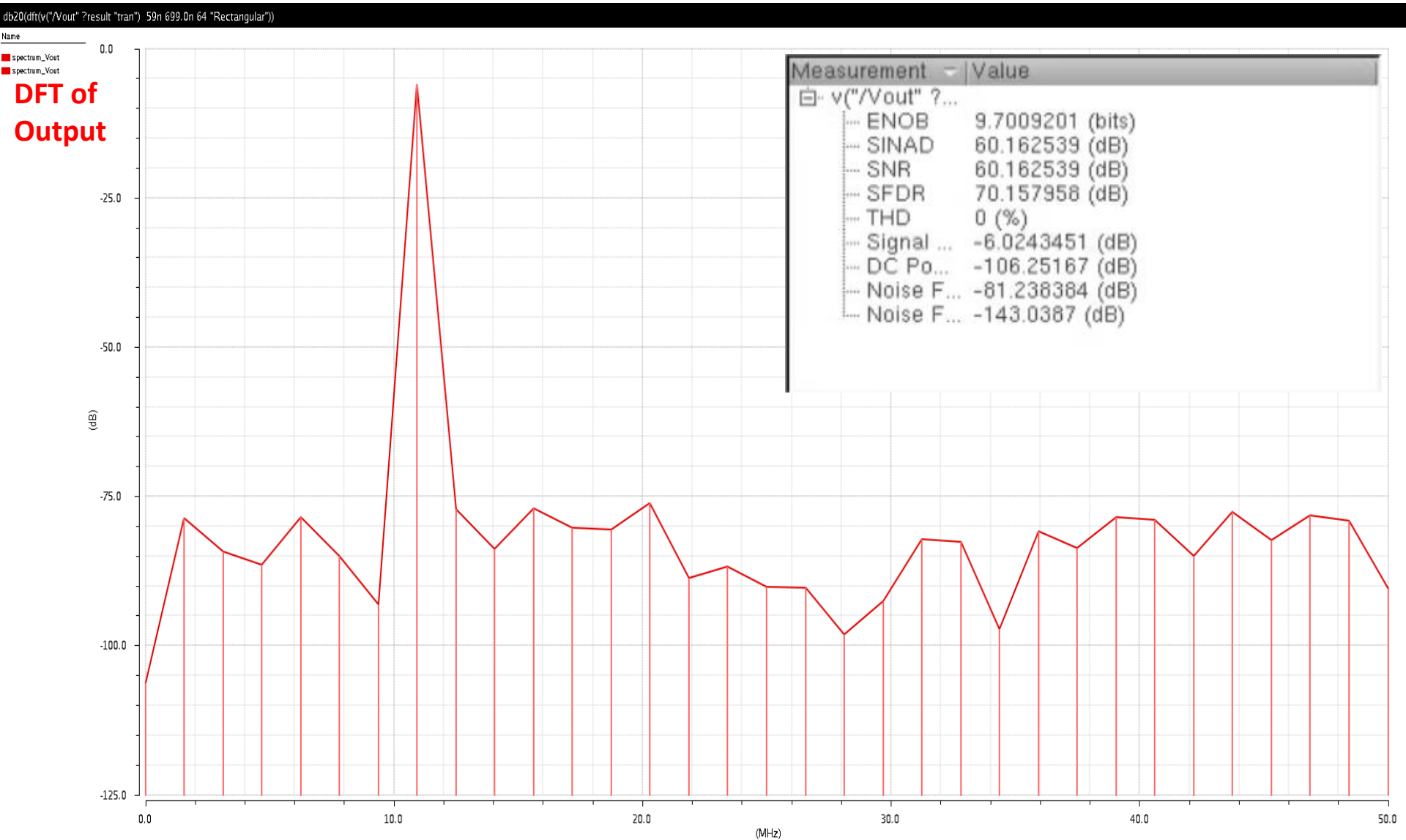
Simulation Results

2- 2x Gain Mode (Signal input Amp = 0.25v)



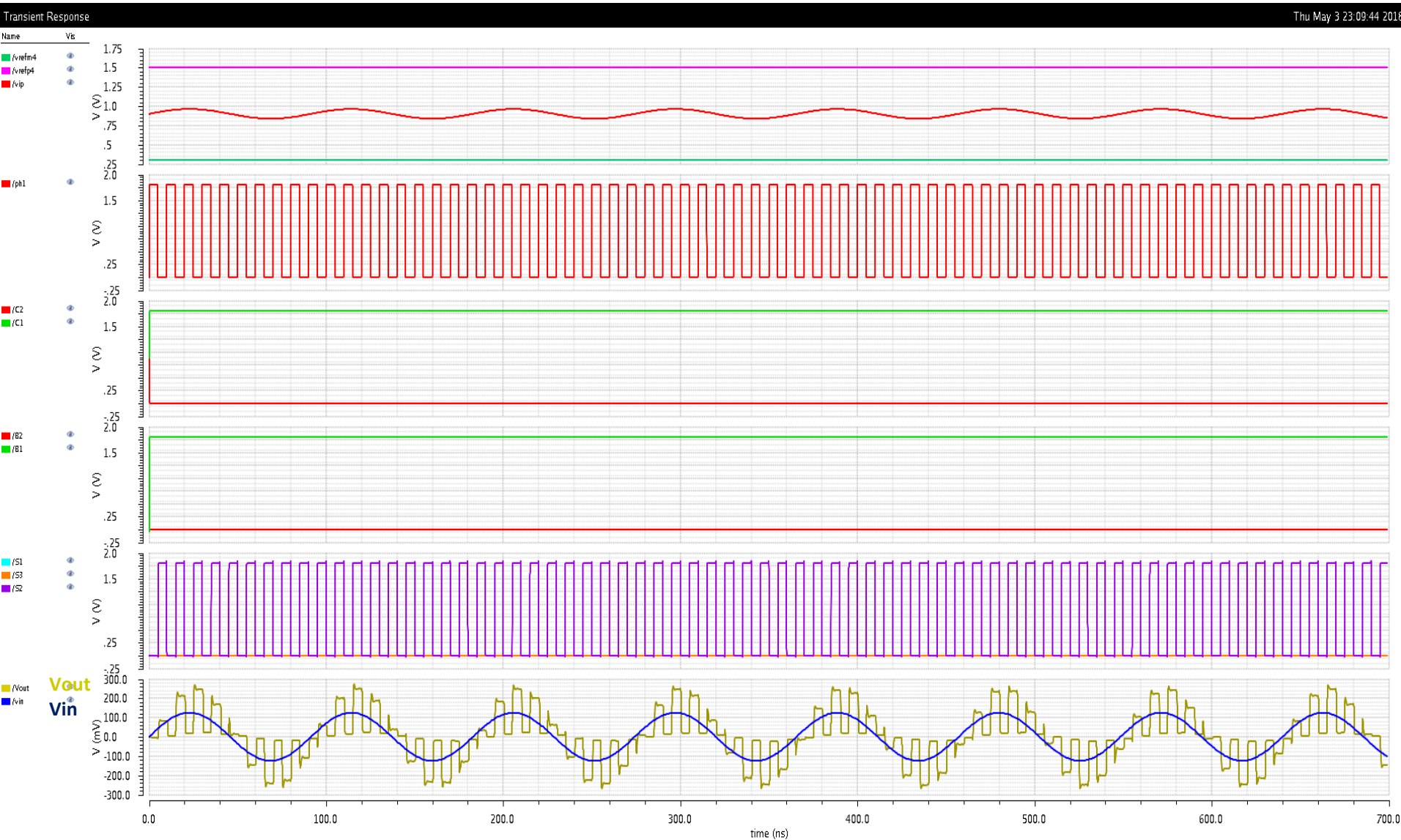
Simulation Results

2- 2x Gain Mode (Signal input Amp = 0.25v)



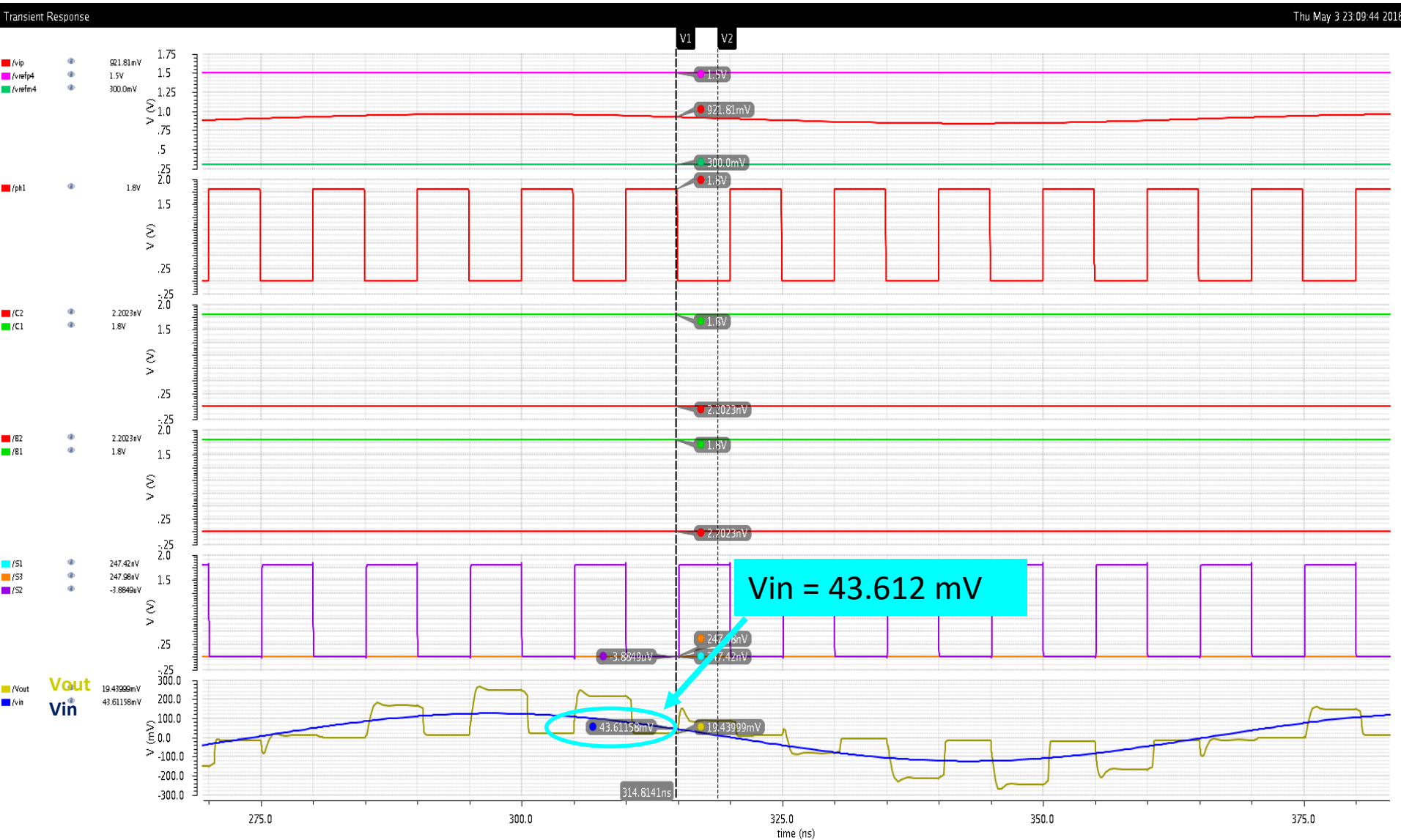
Simulation Results

2- 2x Gain Mode (Signal input Amp = 0.125v)



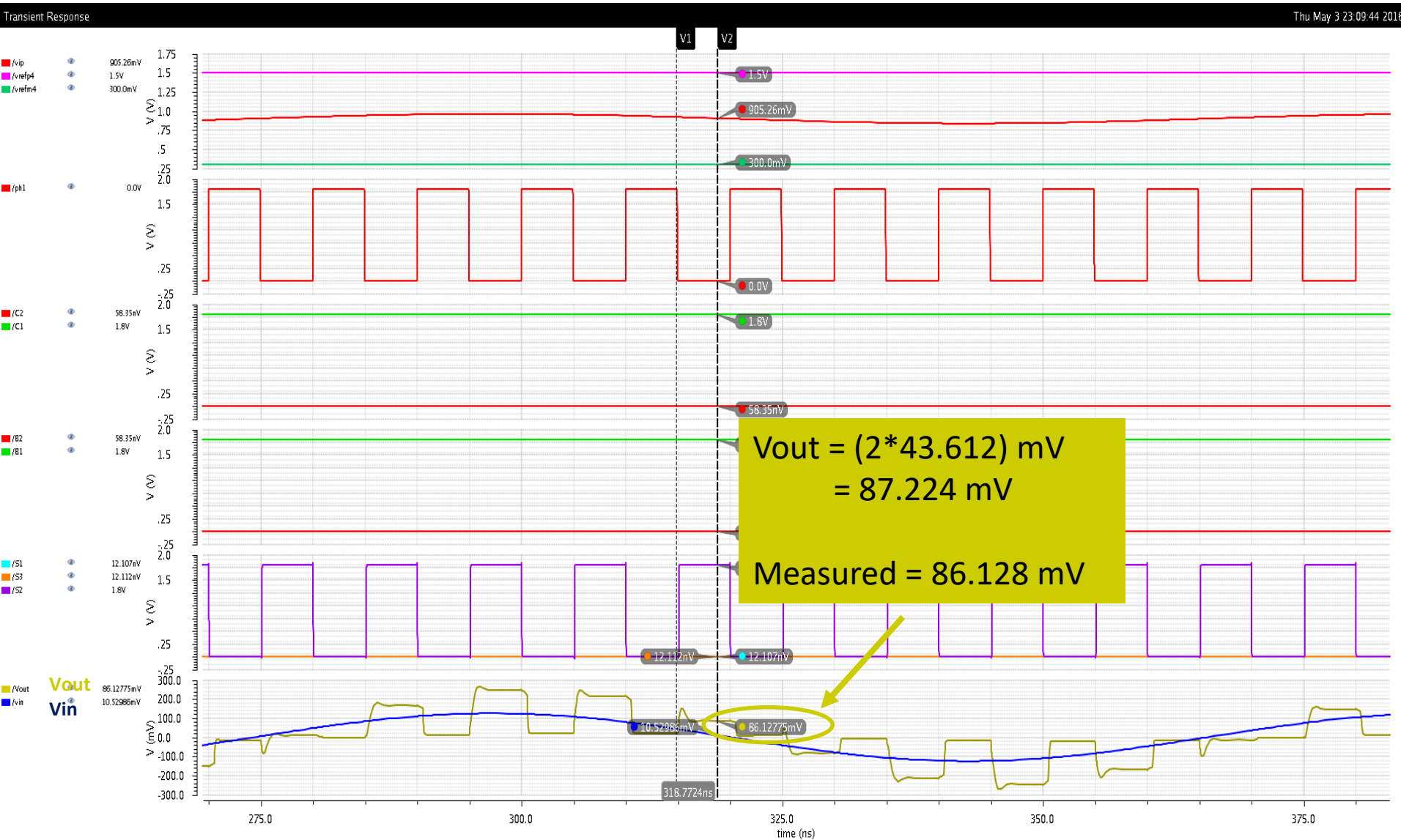
Simulation Results

2- 2x Gain Mode (Signal input Amp = 0.125v)



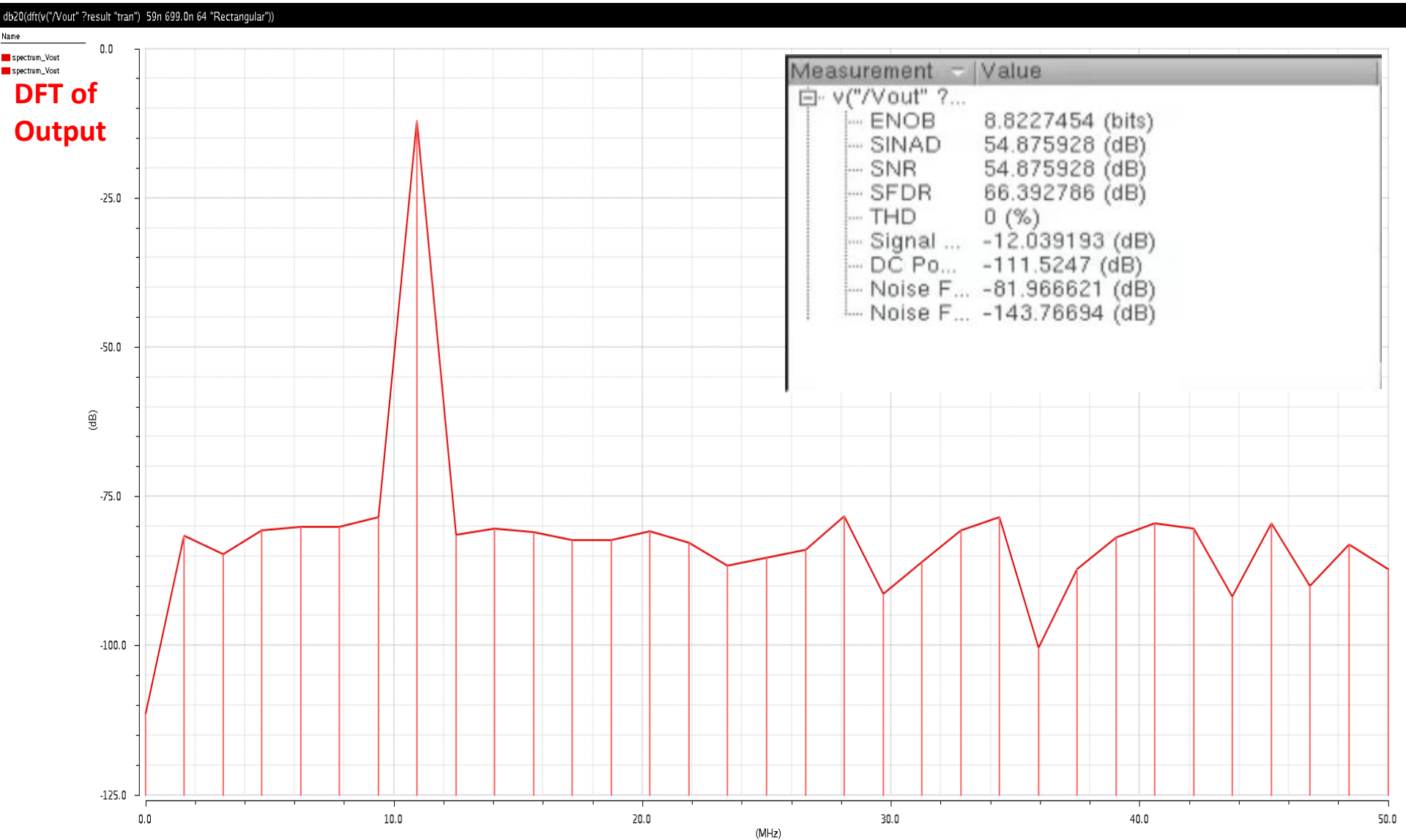
Simulation Results

2- 2x Gain Mode (Signal input Amp = 0.125v)



Simulation Results

2- 2x Gain Mode (Signal input Amp = 0.125v)



Simulation Results

2- 2x Gain Mode

Input Amplitude	Difference between Ideal output value & Measured output value	ENOB	SNR
0.5 V	~ 12 mV	**10.63 bits**	**65.76 dB**
0.25 V	~ 2 mV	**9.70 bits**	**8.82 dB**
0.125 V	~ 1 mV	**8.82 bits**	**54.88 dB**

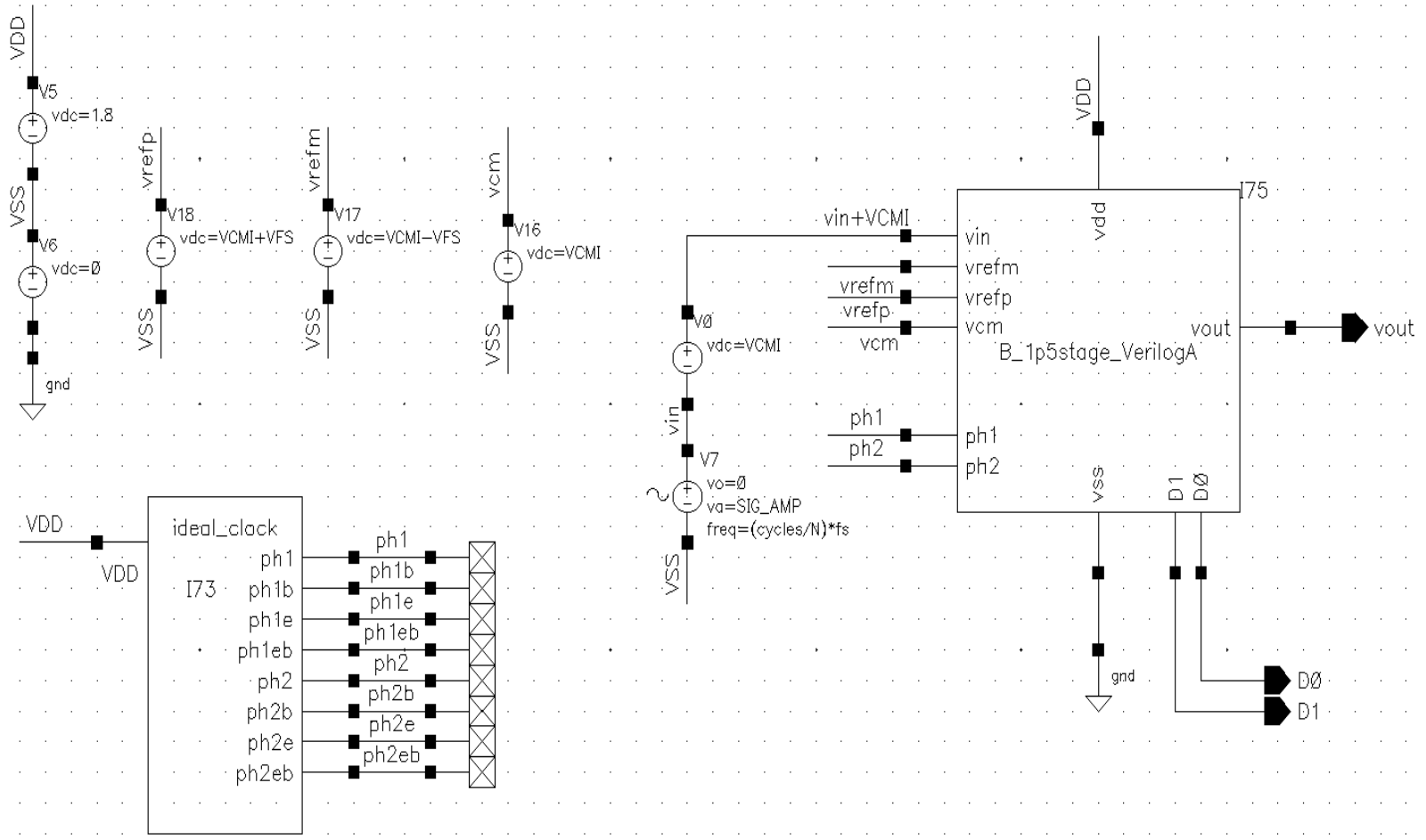
- It is observed that there is an offset between the measured 2x output with respect to the calculated value, & this offset decreases as the input signal amplitude decreases.
- As the amplitude of the input signal decreases, the signal power with respect to the noise decreases, causing a decrease in the ENOB & SNR values.

(5)
VerilogA Block

VerilogA:
1 stage 1.5 bit/stage ADC

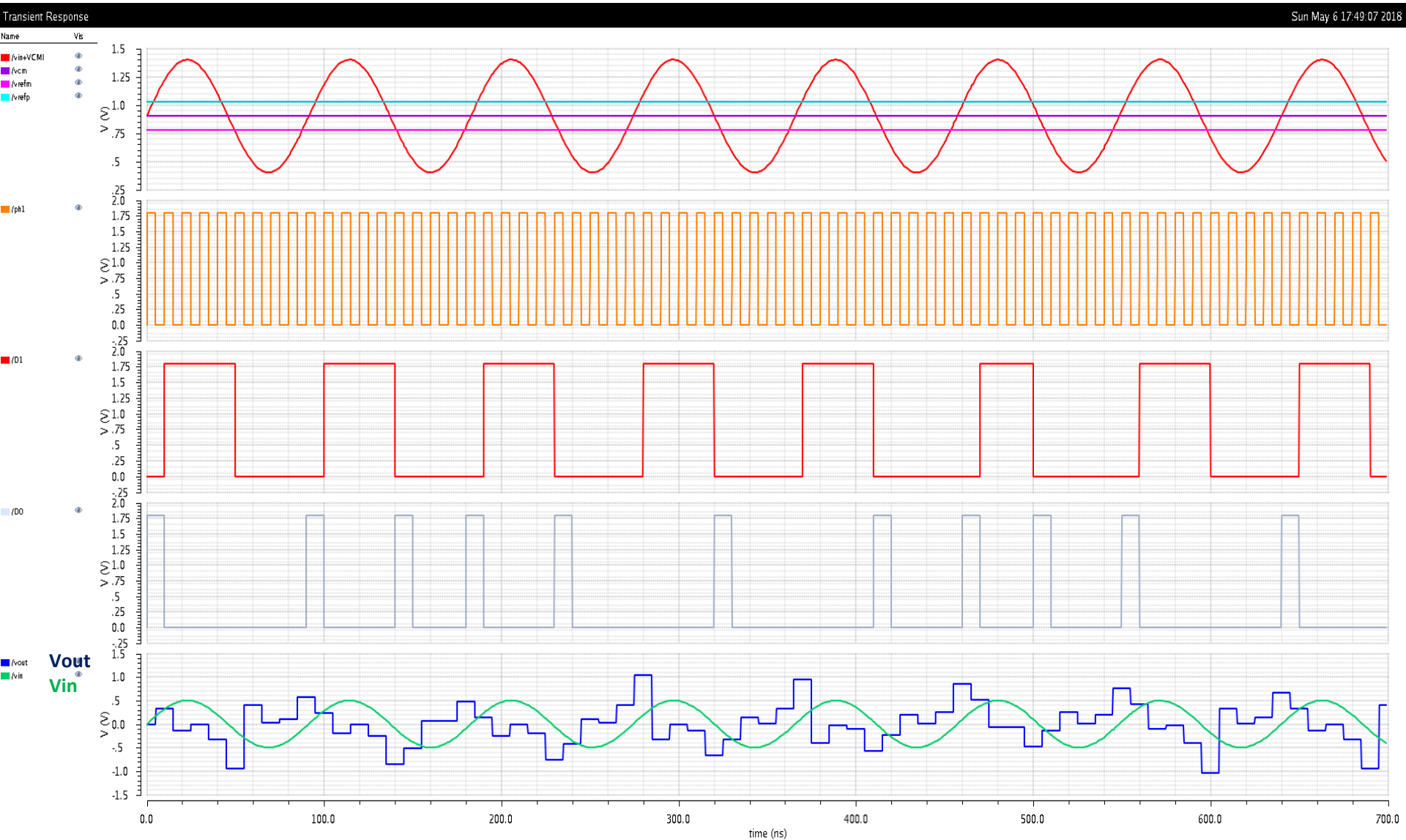
```
// VerilogA for ADC_FinalProject, B_1p5stage_VerilogA, veriloga

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

module B_1p5stage_VerilogA(vin,vrefp,vrefm,ph1,ph2,vcm,vout,D1,D0,vdd,vss);

parameter real clk_th=0.9;
parameter real delay = 0;
parameter real ttime = 1p;

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

electrical vdd,vss;
electrical vin,vrefp,vrefm,ph1,ph2,vcm;
electrical vout,D1,D0;

real dd1,dd0,vvout;

analog begin
        @(cross(V(ph1) - clk_th, +1)) begin
                if (V(vin)>V(vrefp)) begin
                        dd1 = V(vdd);        dd0 = V(vss);
                        end
                else if ((V(vin)<V(vrefp)) && (V(vin)>V(vrefm))) begin
                        dd1 = V(vss);        dd0 = V(vdd);
                        end
                else begin
                        dd1 = V(vss);        dd0 = V(vss);
                        end
        end

        @(cross(V(ph2) - clk_th, +1)) begin
                if (dd1 > 0.9) begin
                        vvout = ((V(vin) -0.9) * 2) - 1;
                        end
                else if (dd0 > 0.9) begin
                        vvout = (V(vin) -0.9) * 2;
                        end
                else begin
                        vvout = ((V(vin) -0.9) * 2) + 1;
                        end
        end

        V(D1) <+ transition(dd1,delay,ttime);
        V(D0) <+ transition(dd0,delay,ttime);
        V(vout) <+ transition(vvout,delay,ttime);

end
endmodule
```

Testbench


VDD
V5
vdc=1.8
VSS
V6
vdc=Ø
gnd
vrefp
V18
vdc=VCMI+VFS
VSS
vrefm
V17
vdc=VCMI−VFS
VSS
vcm
V16
vdc=VCMI
VSS
VDD
VDD
ideal_clock
I73
ph1
ph1b
ph1e
ph1eb
ph2
ph2b
ph2e
ph2eb
VDD
vin+VCMI
VØ
vdc=VCMI
vin
V7
vo=Ø
va=SIG_AMP
freq=(cycles/N)*fs
VSS
vrefm
vrefp
vcm
ph1
ph2
I75
vdd
vin
vrefm
vrefp
vcm
ph1
ph2
vss
D1
DØ
vout
B_1p5stage_VerilogA
vout
gnd
DØ
D1

Simulation Results

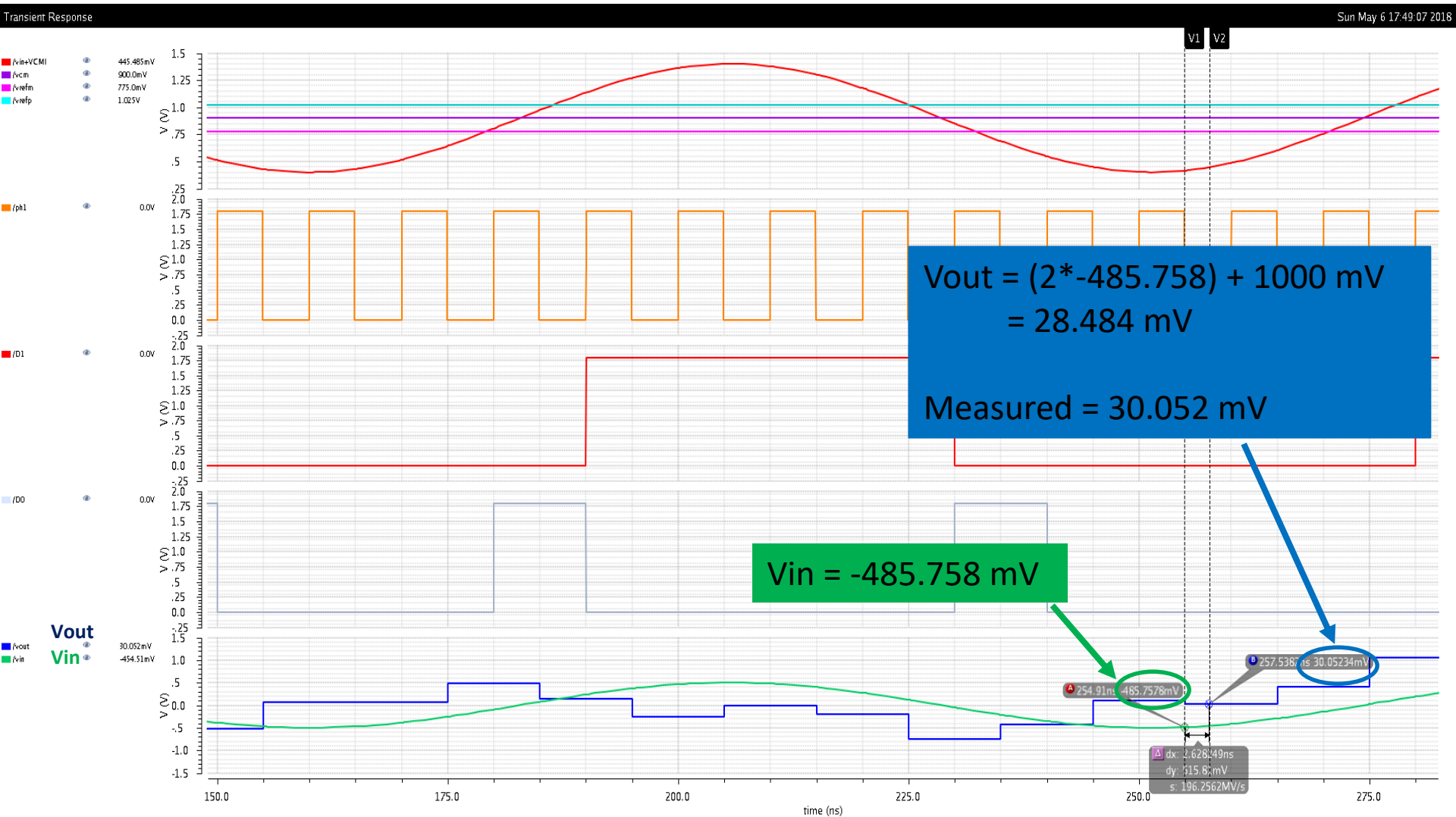
1- Normal Mode



Simulation Results

1- Normal Mode

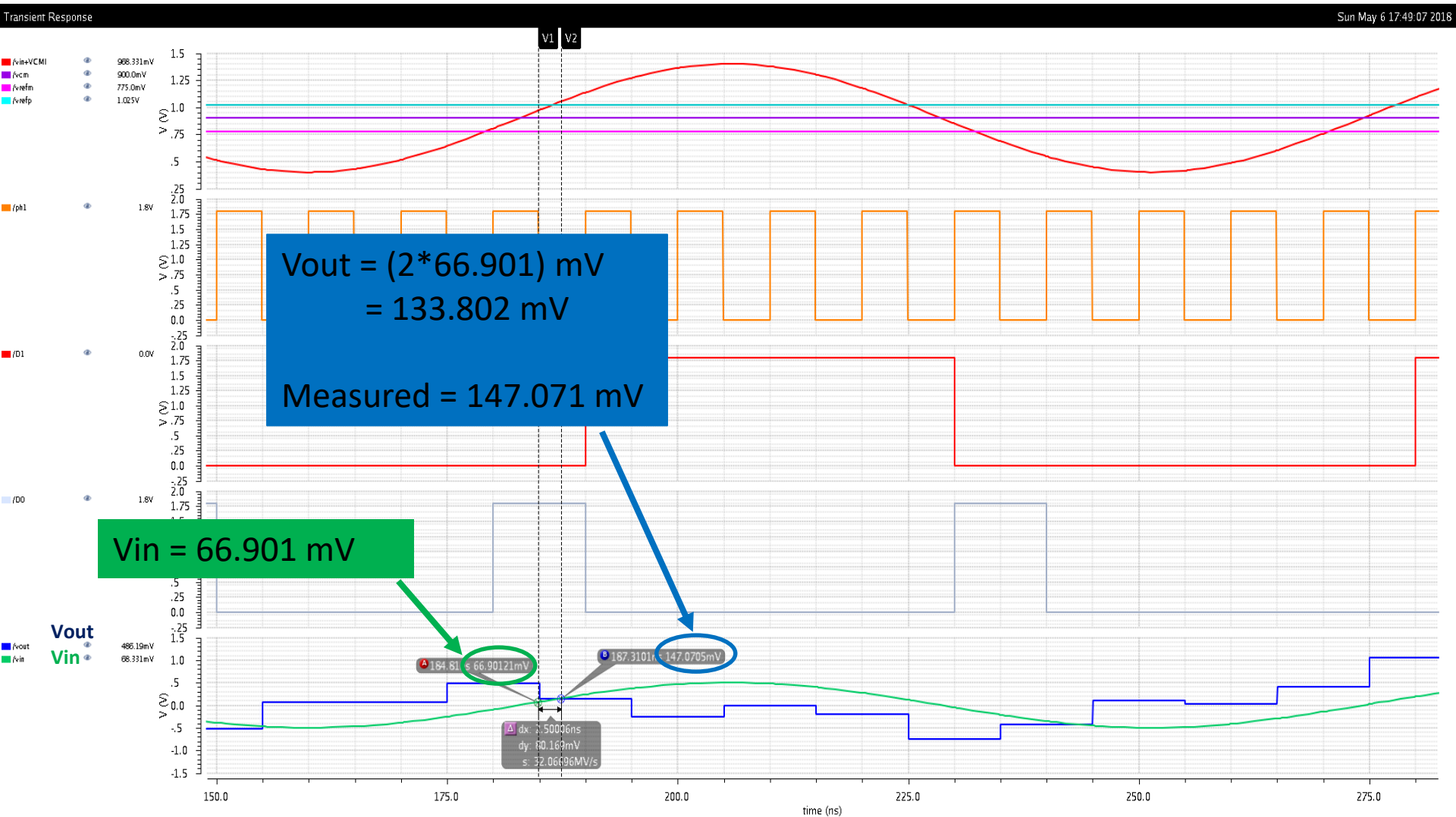
1) If Vin < Vrefm ➔ Vout = 2*Vin + Vfs



Simulation Results

1- Normal Mode

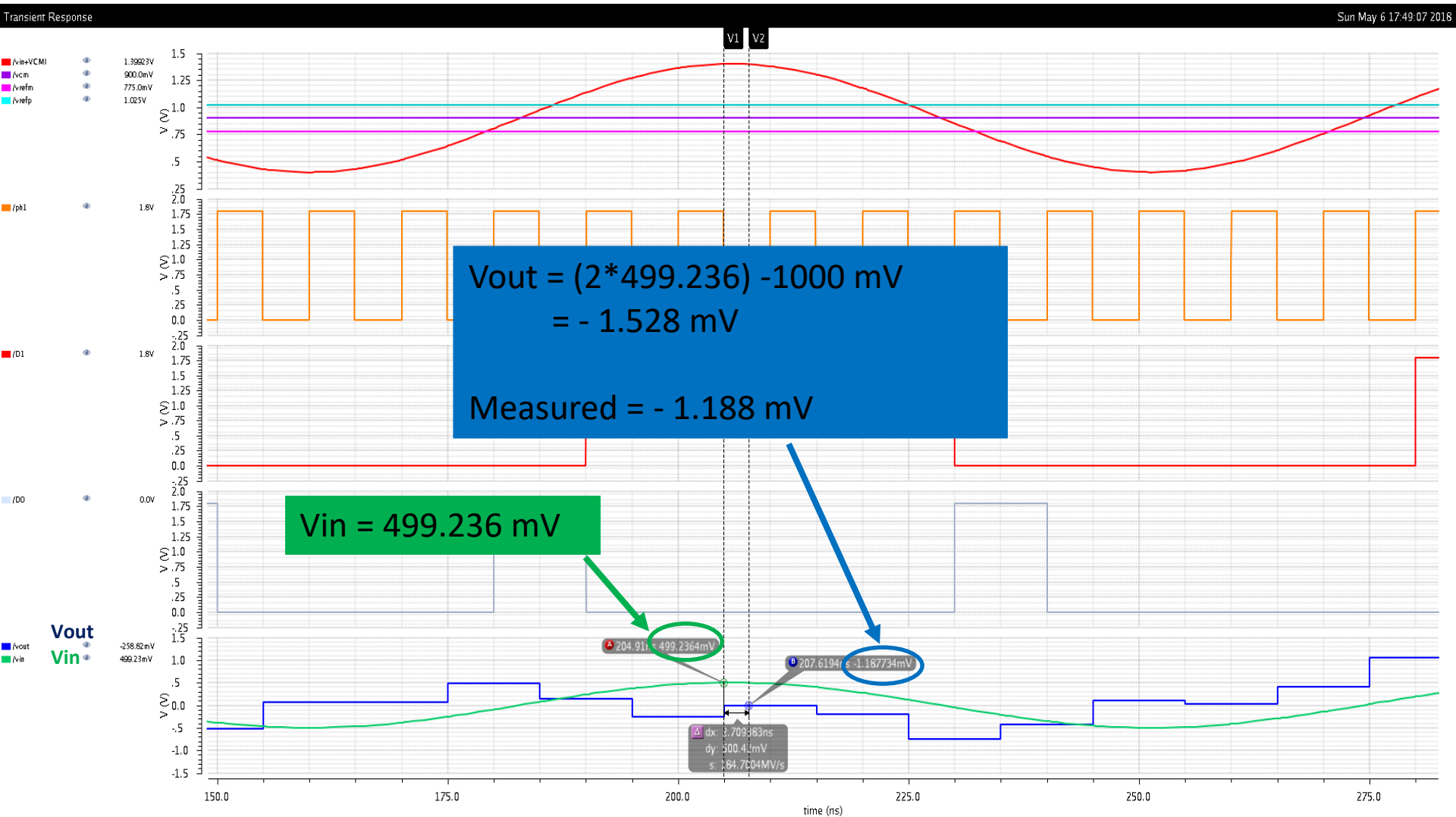
2) If Vrefm < Vin < Vrefp ➔ Vout = 2*Vin



Simulation Results

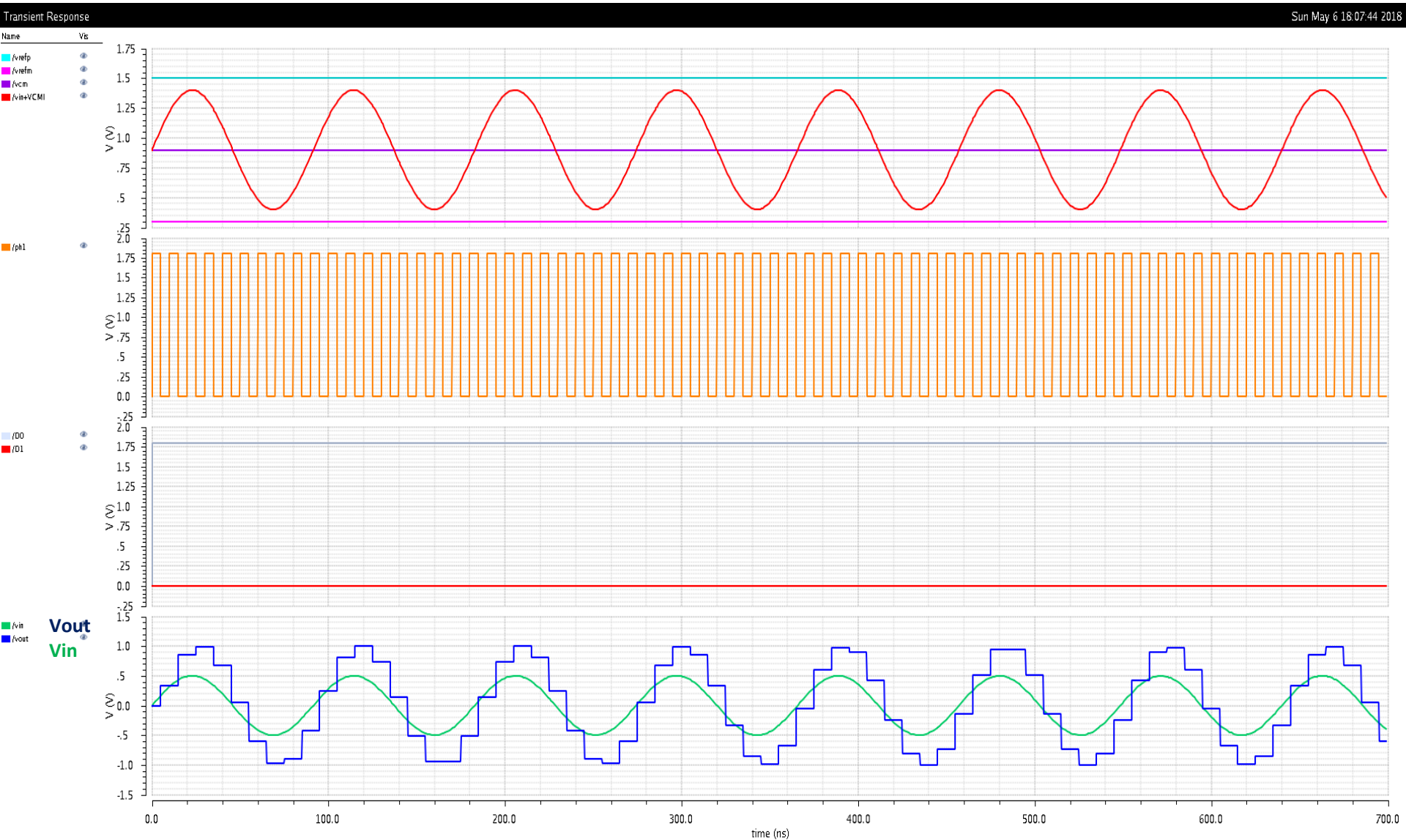
1- Normal Mode

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



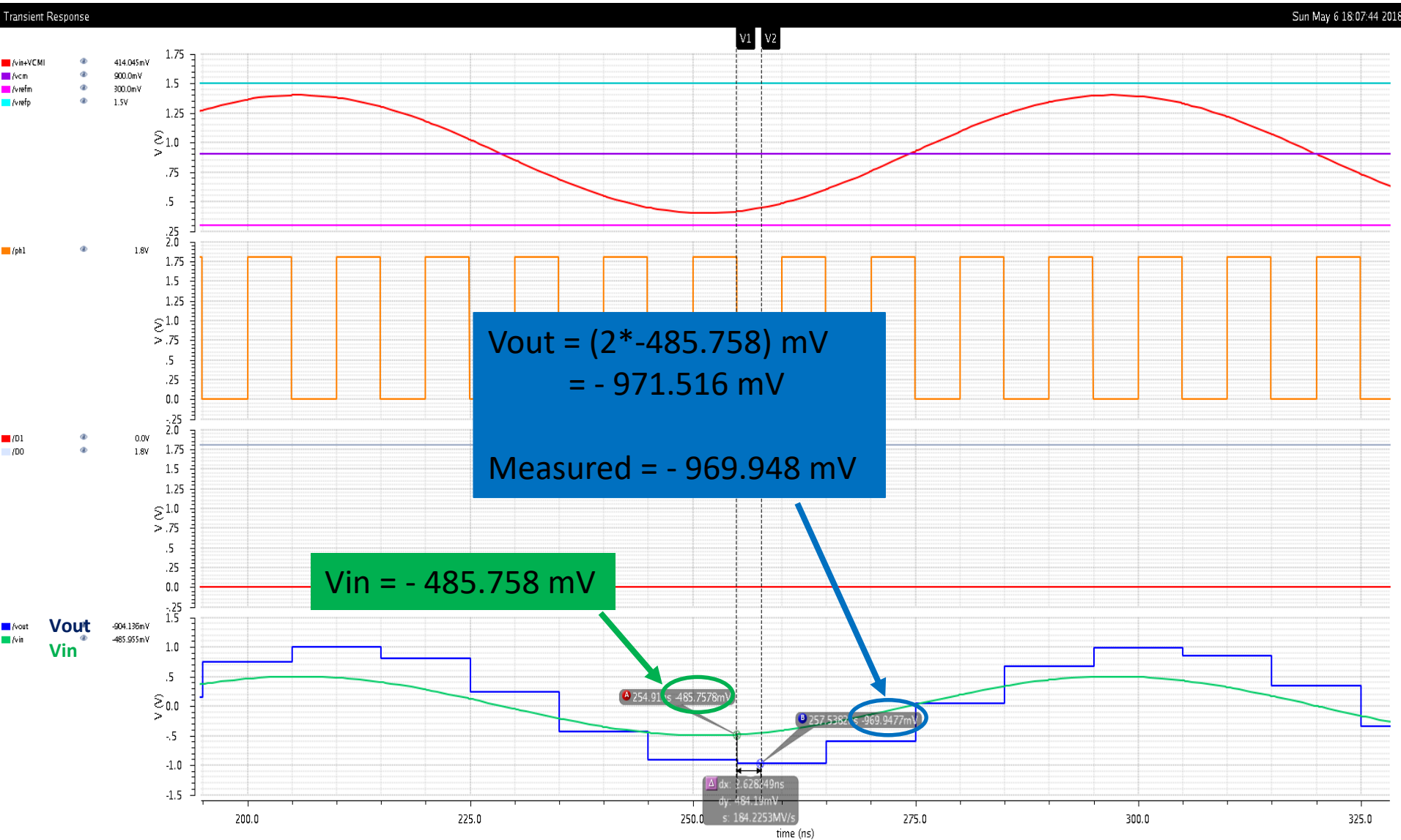
Simulation Results

2- 2x Gain Mode



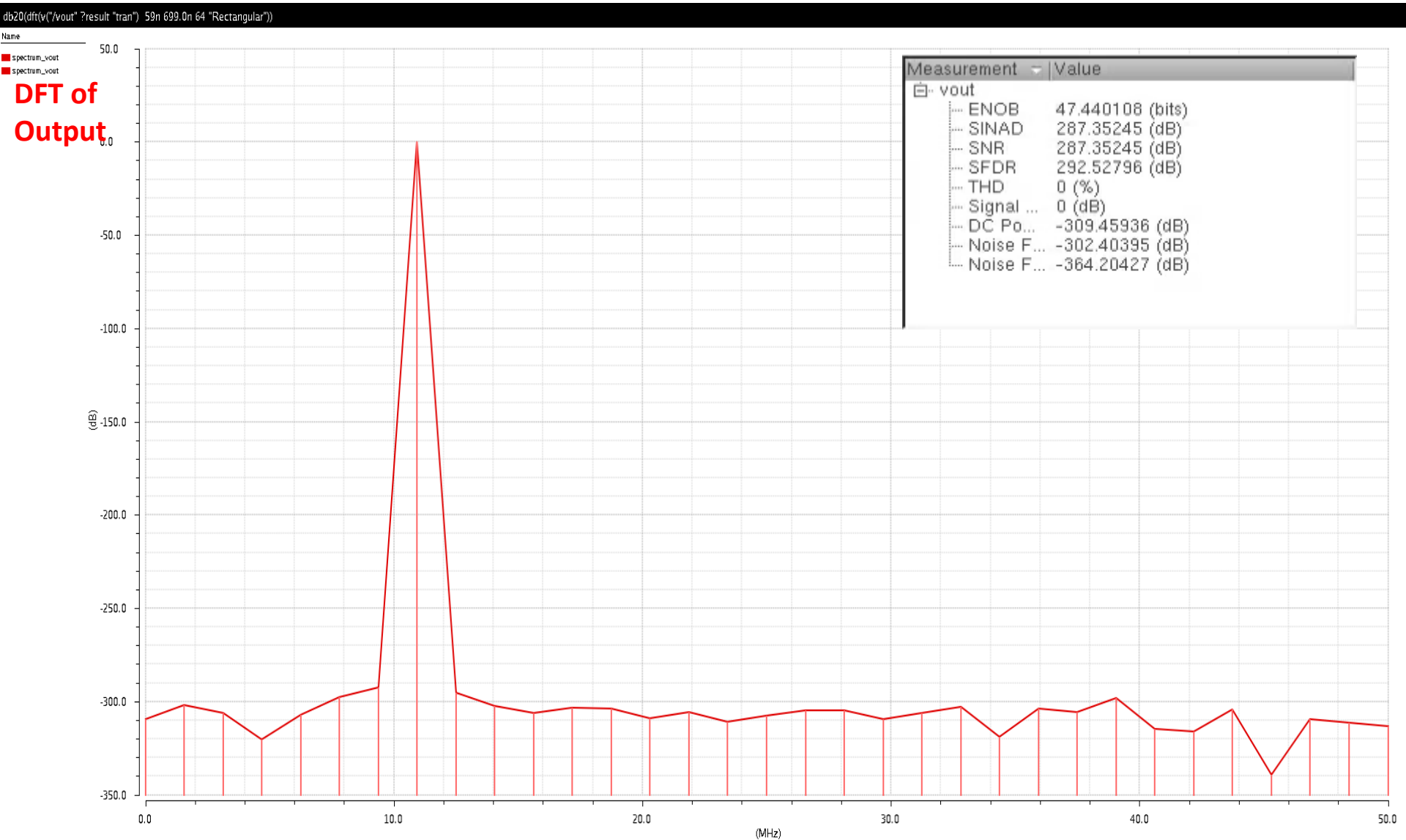
Simulation Results

2- 2x Gain Mode



Simulation Results

2- 2x Gain Mode



[Extra] VerilogA: Sub-ADC Logic

```
// VerilogA for ADC_FinalProject, B_SubADCLogic, veriloga

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

module B_SubADCLogic(C2,C1,ph2,S1,S2,S3,B2,B1,vdd,vss);

parameter real clk_th=0.9;
parameter real delay = 0;
parameter real ttime = 1p;

inout vdd,vss;
input C2,C1,ph2;
output S1,S2,S3,B2,B1;

electrical vdd,vss;
electrical C2,C1,ph2;
electrical S1,S2,S3,B2,B1;

real ss1,ss2,ss3,bb2,bb1;

analog begin
        @(cross(V(ph2) - clk_th, +1)) begin
                if ((V(C2)>0.9) && (V(C1)>0.9)) begin
                        ss1 = V(vdd);       ss2 = V(vss);       ss3 = V(vss);
                        bb2 = V(vdd);       bb1 = V(vss);
                        end
                else if ((V(C2)<0.9) && (V(C1)>0.9)) begin
                        ss1 = V(vss);       ss2 = V(vdd);       ss3 = V(vss);
                        bb2 = V(vss);       bb1 = V(vdd);
                        end
                else begin
                        ss1 = V(vss);       ss2 = V(vss);       ss3 = V(vdd);
                        bb2 = V(vss);       bb1 = V(vss);
                        end
        end

        @(cross(V(ph2) - clk_th, -1)) begin
                ss1 = V(vss);       ss2 = V(vss);       ss3 = V(vss);
        end

        V(S1) <+ transition(ss1,delay,ttime);
        V(S2) <+ transition(ss2,delay,ttime);
        V(S3) <+ transition(ss3,delay,ttime);
        V(B2) <+ transition(bb2,delay,ttime);
        V(B1) <+ transition(bb1,delay,ttime);


end
endmodule
```

SUMMARY

Summary Notes:

- When replacing the ideal switches with switches with real transistors, for better performance, the CMOS transmission gate switches are preferably used to connect 2 nodes, while the Bootstrapped switches are used to pass exact input values to a certain node.
- The comparators compare the input signal at the rising edge of ph1, then the outputs are generated at the rising edge of ph2.
- The value of the stage output depends on the value of the input signal right before the falling edge of ph1 (rising edge of ph2).
- Decreasing the amplitude of the input signal, makes it easier for the system to give a more closer 2x gain (less offset), but also decreases the SNR & the ENOB (since the signal power decreased with respect to the noise level).