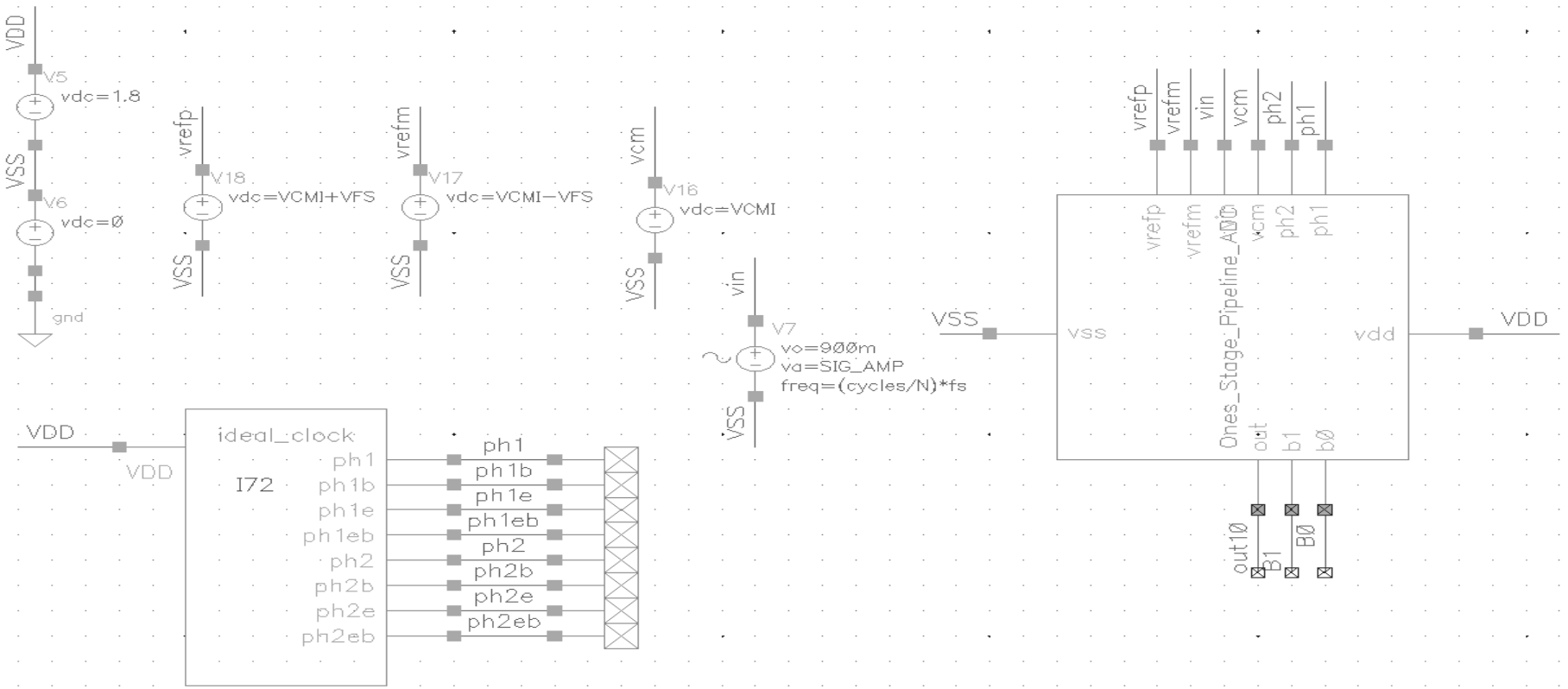
VerilogA Blocks
For a 10-bit Pipelined ADC

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,out,ph1,ph2,b0,b1,vcm);
input vrefm,vrefp,ph1,ph2,vcm,vin;
inout vss,vdd;
output out,b0,b1;
electrical vin,out,vss,vdd,vrefm,vrefp,ph1,ph2,b0,b1,vcm;
parameter real clk_vth = 0.5, delay = 0, ttime = 1p;
real v,B0,B1, vp, vm, vinput;
 analog begin
vp = V(vcm) + (V(vrefp)-V(vrefm))/8;
vm = V(vcm) - (V(vrefp)-V(vrefm))/8;
@(cross(V(ph1) - clk_vth,-1)) begin
      if((V(vin) >= vp))
          begin
          B0 = V(vss);
          B1 = V(vdd);
          end
         else if ((V(vin)<= vp ) && (V(vin)>= vm)) begin
             B0 = V(vdd);
             B1 = V(vss);
             end
          else begin
             B0 = V(vss);
             B1 = V(vss);
             end
       end
@(cross(V(ph2) - clk_vth,+1)) begin
                  if((B1 >= 0.9)) begin
              v = (V(vin)*2) - V(vrefp);
             end
          else if((B0 >= 0.9)) begin
              v = (V(vin)*2) - V(vcm);
             end
          else begin
              v = (V(vin)*2) - V(vrefm);
              end
          end
V(out) <+ transition(v,delay,ttime);
V(b0) <+ transition(B0,delay,ttime);
V(b1) <+ transition(B1,delay,ttime);
endendmodule
```

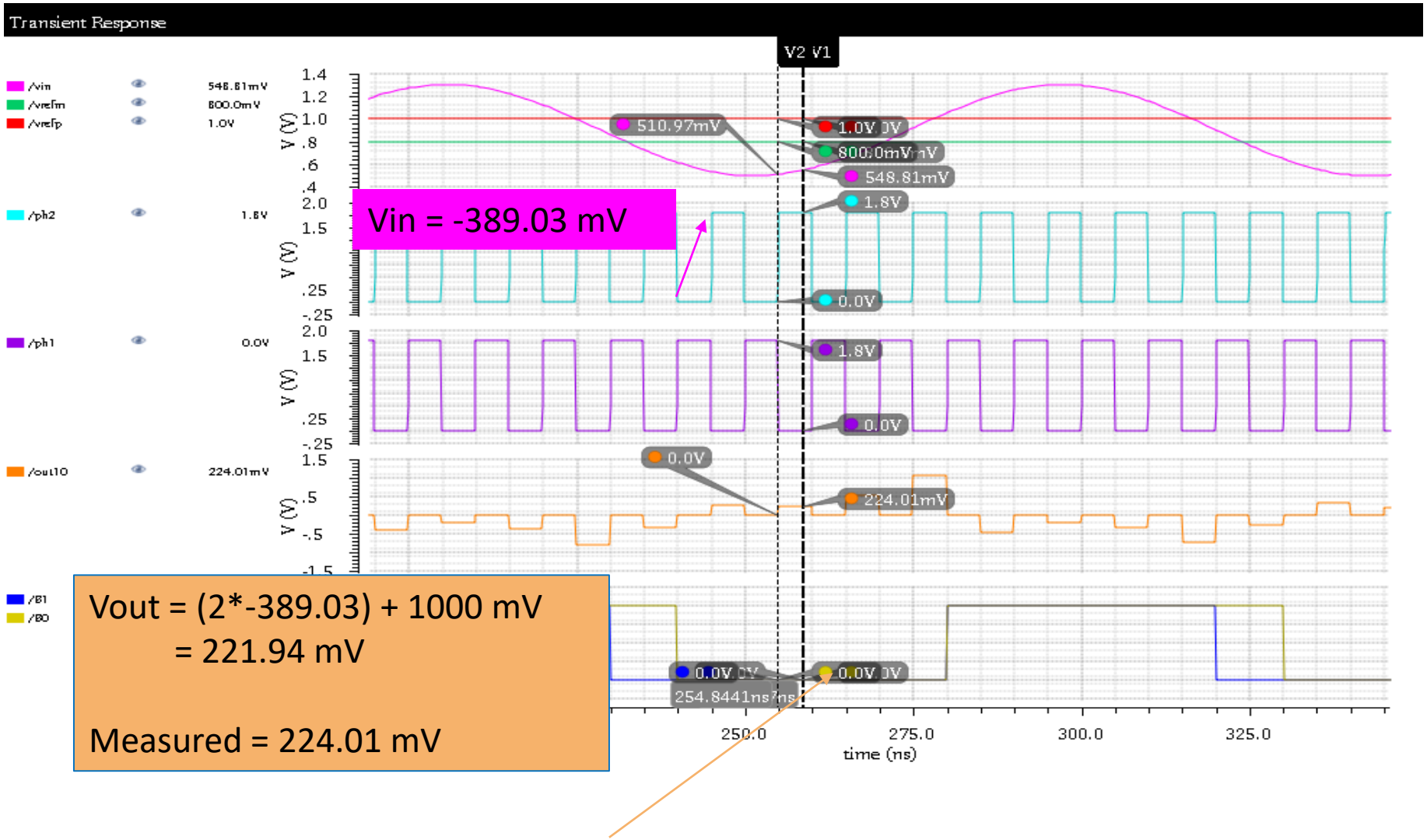
Testbench


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
VDD
ph1
ph1b
ph1e
ph1eb
ph2
ph2b
ph2e
ph2eb
VSS
vrefp
vrefm
vin
vcm
ph2
ph1
vss
vdd
Ones_Stage_Pipeline_ADC
out
b1
b0
VDD
out10
B1
B0

Simulation Results

1- Normal Mode

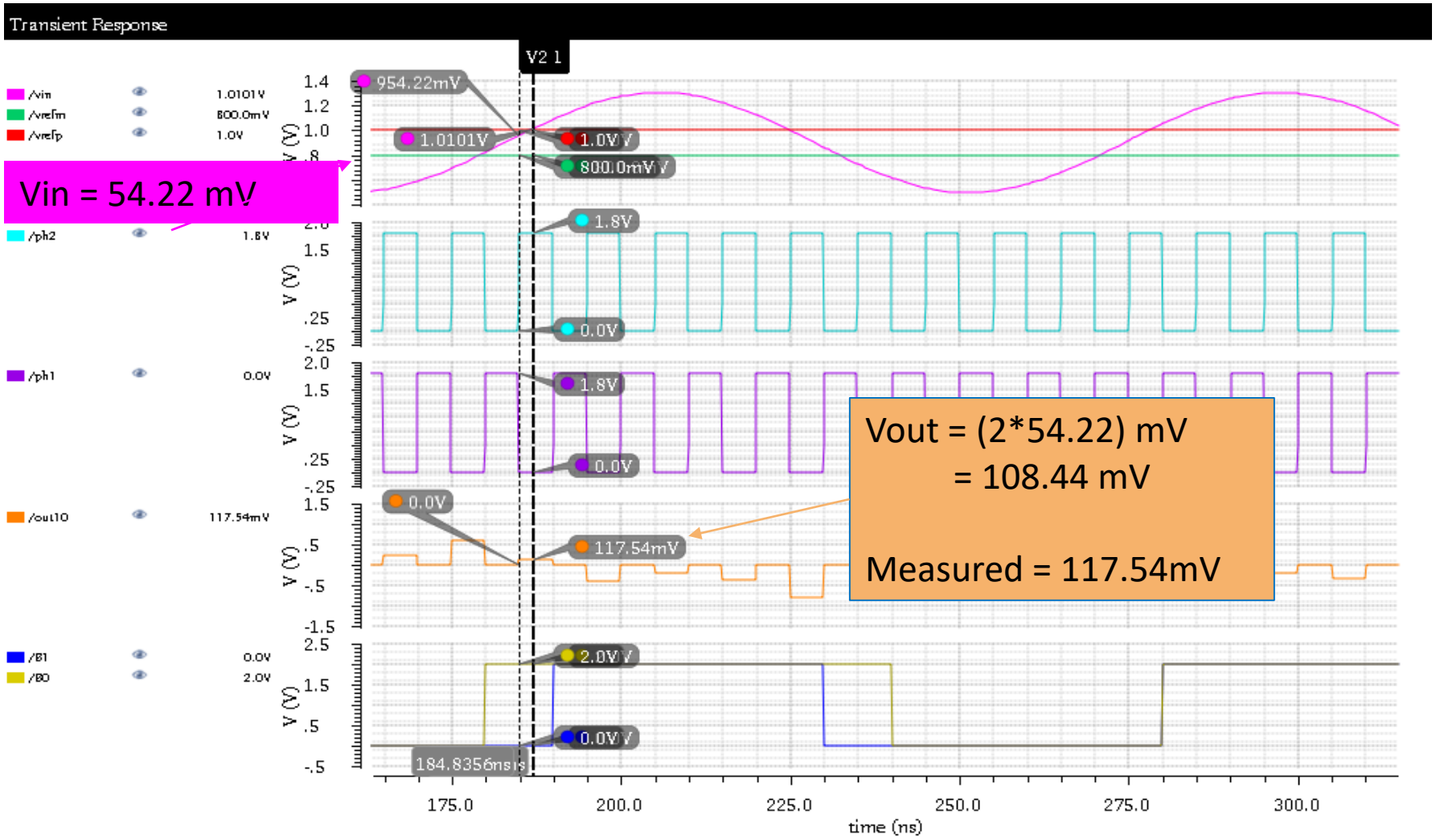
1) If Vin < Vrefm ➔ Vout = 2*Vin + Vfs



Simulation Results

1- Normal Mode

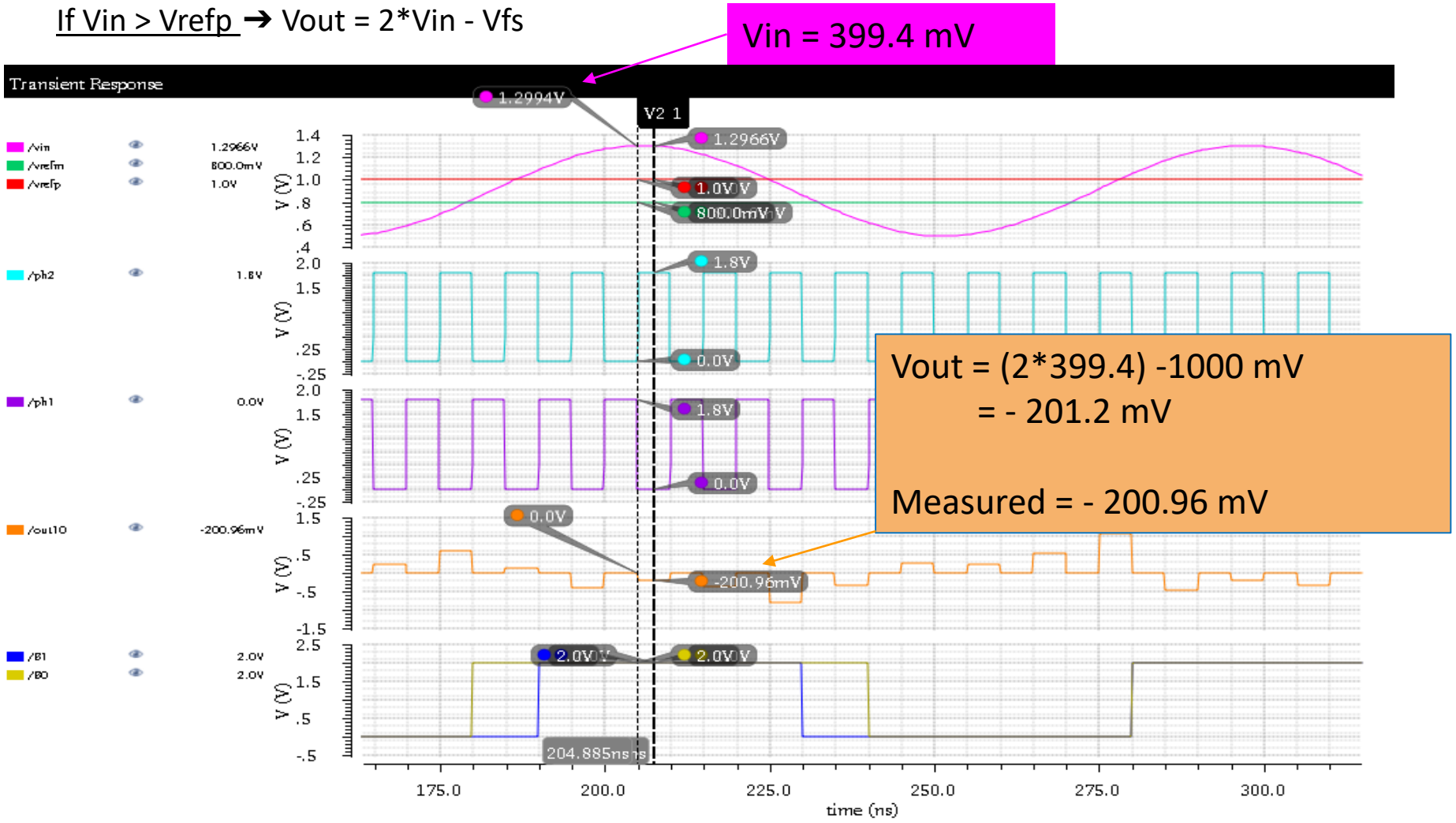
2) If Vrefm < Vin < Vrefp ➔ Vout = 2*Vin



Simulation Results

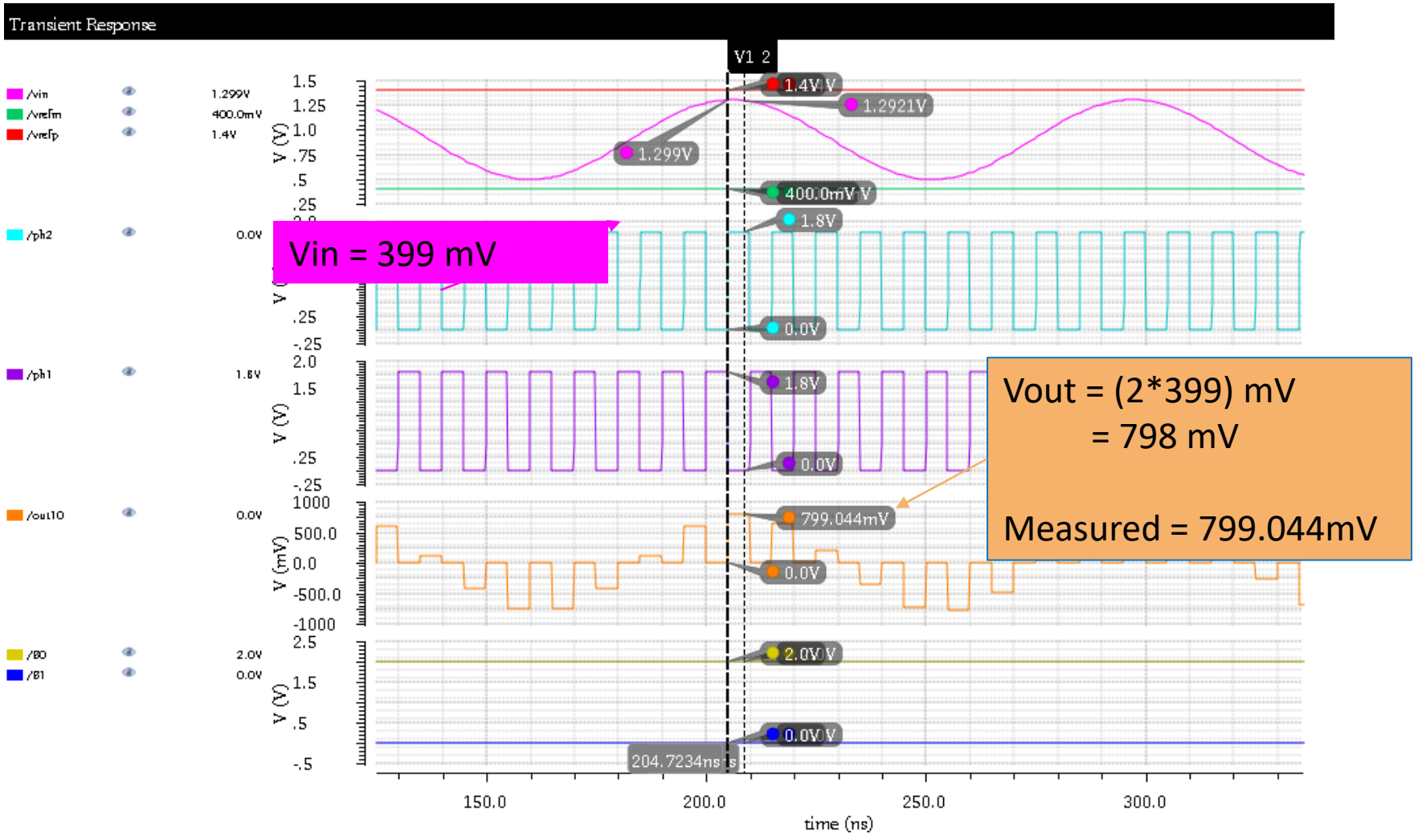
1- Normal Mode

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



Simulation Results

2- 2x Gain Mode

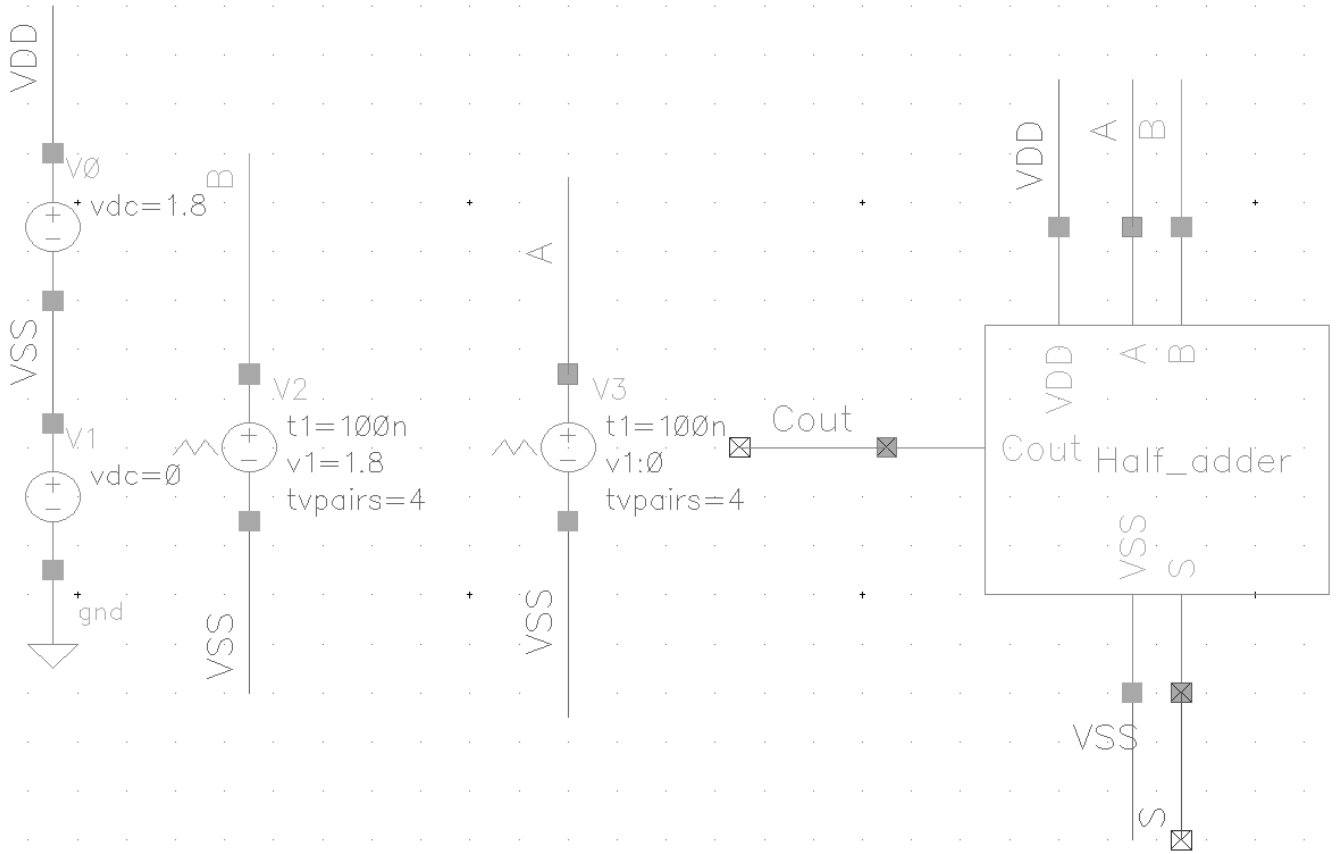


VerilogA:
Half Adder

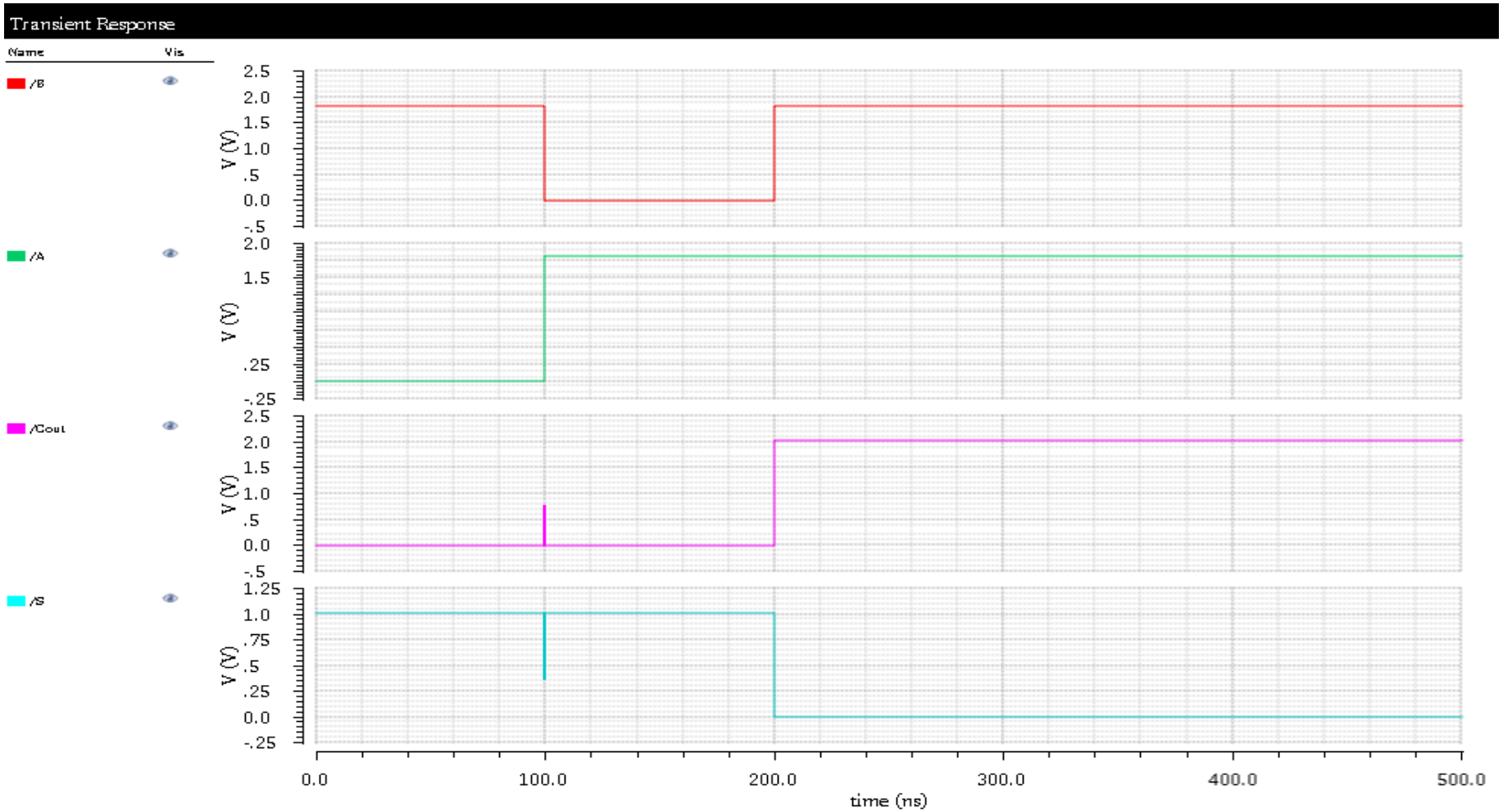
```
// VerilogA for EE288, half_adder, veriloga

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

Testbench


VDD
V0
vdc=1.8
VSS
V1
vdc=0
gnd
B
V2
t1=100n
v1=1.8
tvpairs=4
VSS
A
V3
t1=100n
v1:0
tvpairs=4
VSS
Cout
VDD
A
B
VDD
A
B
Cout
Half_adder
VSS
S
VSS
S

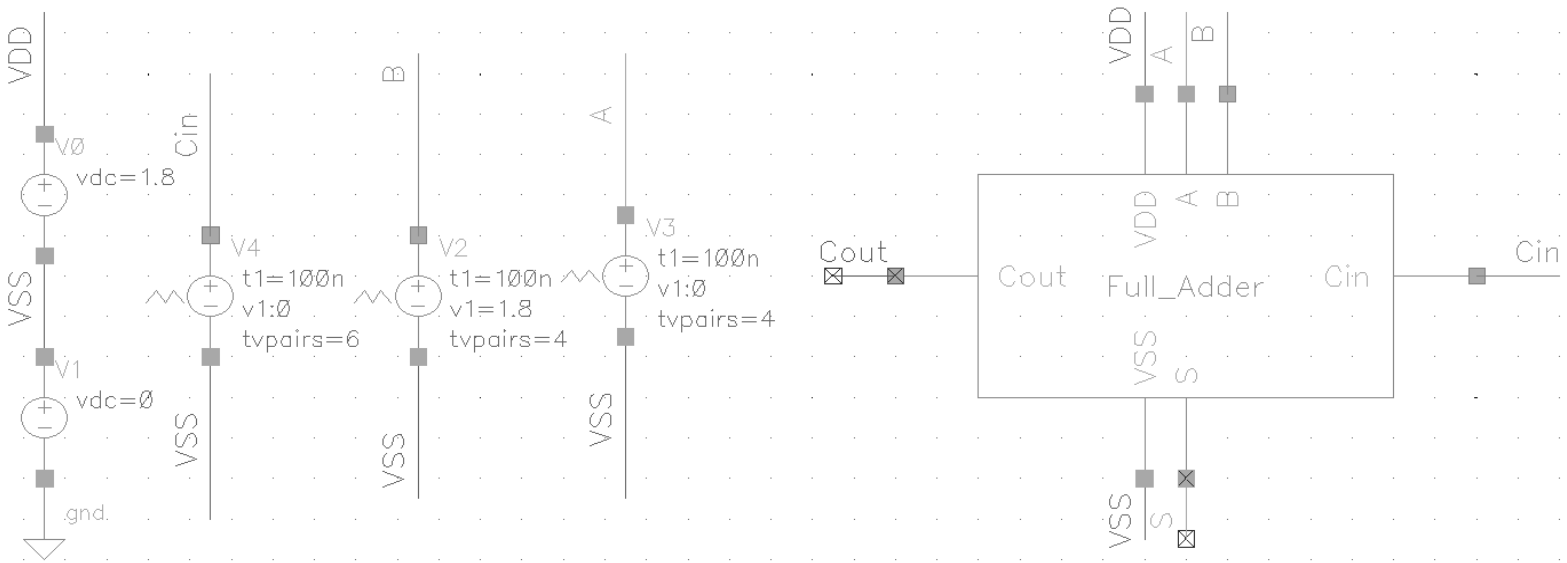
Simulation Results



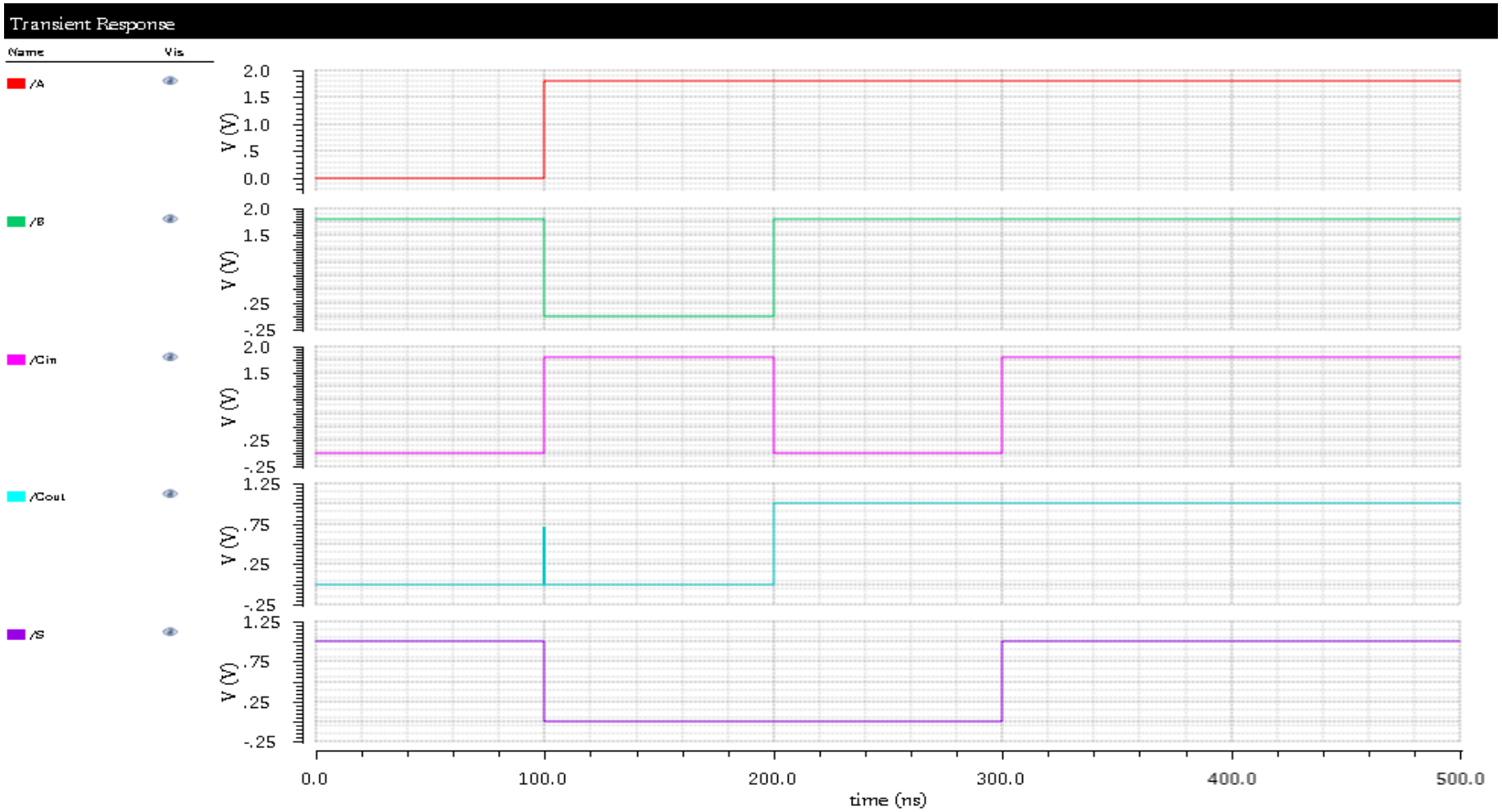
VerilogA:
Full Adder

```
// 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=V(VSS) ;
       end
   else begin
       xor1 = V(VDD);
       end
   if( ((V(Cin)>=0.9 && (xor1>=0.9))||(V(Cin)<=0.5 && (xor1<=0.5))))  begin
       sum= V(VSS);
       end
   else begin
       sum = V(VDD) ;
       end
 if( (V(A)>=0.9) && (V(B)>=0.9) ) begin
      AND1 = V(VDD);
       end
   else begin
       AND1 = V(VSS);
       end
    if( (xor1>=0.9) && (V(Cin)>=0.9) ) begin
      AND2 = V(VDD);
       end
   else begin
       AND2 = V(VSS);
       end
   if((AND1 >= 0.9) ||(AND2 >= 0.9) ) begin
       carri= V(VDD);
       end
   else begin
       carri = V(VSS);
       end
V(Cout) <+ transition(carri,delay,ttime);
V(S) <+ transition(sum,delay,ttime);
end
endmodule
```

Testbench


VDD
VØ
vdc=1.8
VSS
V1
vdc=Ø
gnd
Cin
V4
t1=1ØØn
v1:Ø
tvpairs=6
VSS
B
V2
t1=1ØØn
v1=1.8
tvpairs=4
VSS
A
V3
t1=1ØØn
v1:Ø
tvpairs=4
VSS
Cout
VDD
A
B
Cout
Full_Adder
Cin
VSS
S
VSS
S
Cin

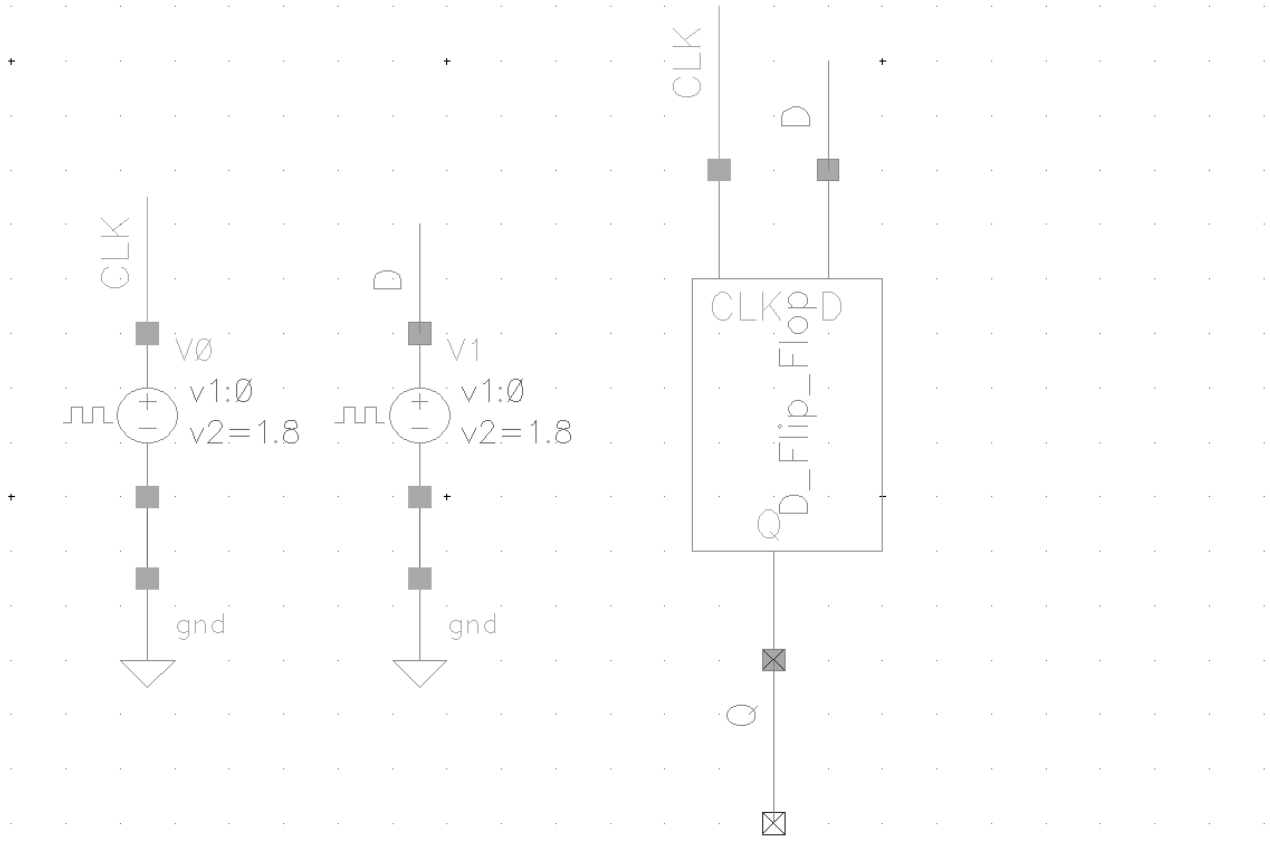
Simulation Results



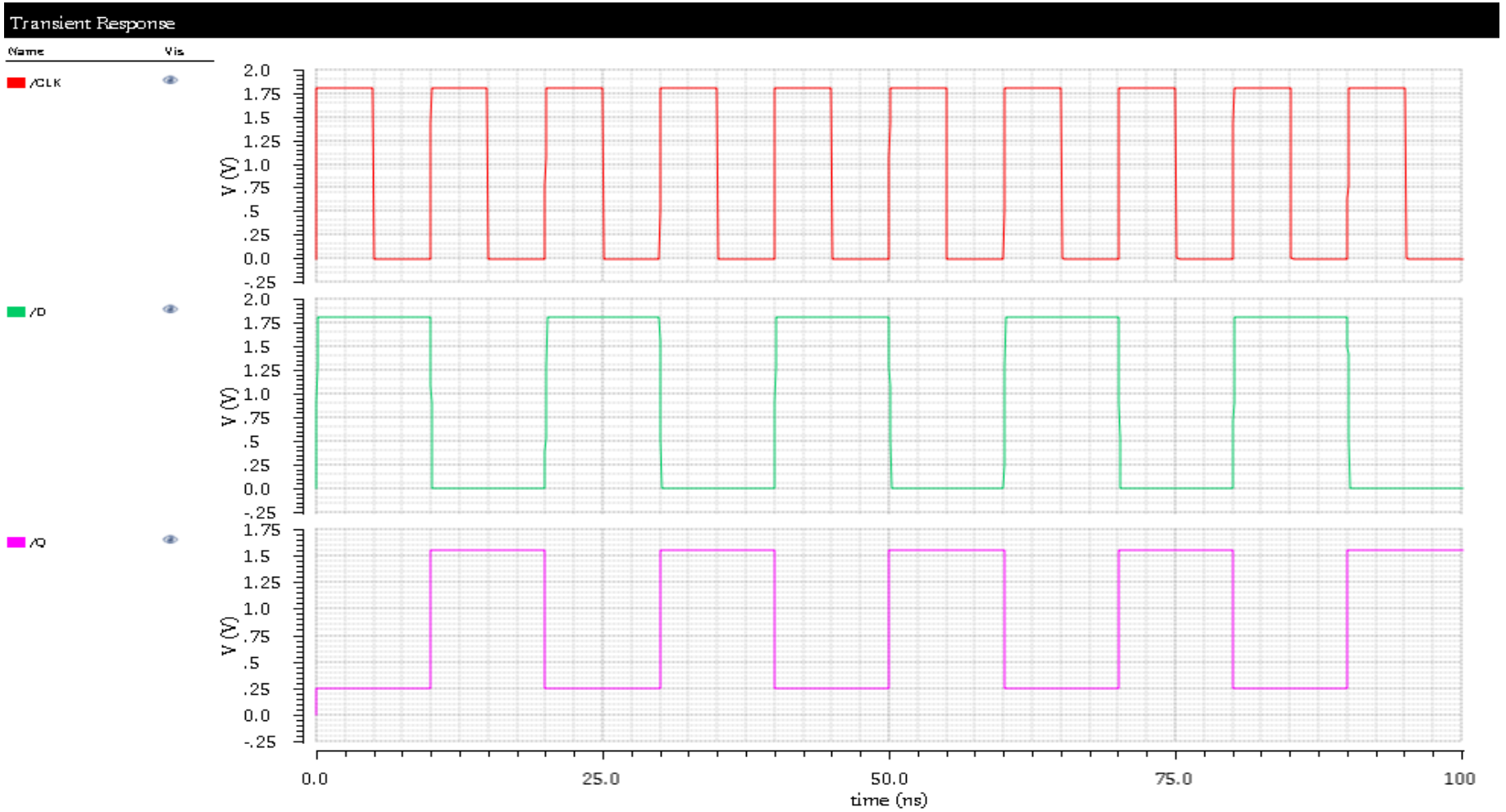
VerilogA:
D flipflop

```
// 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
```

Testbench



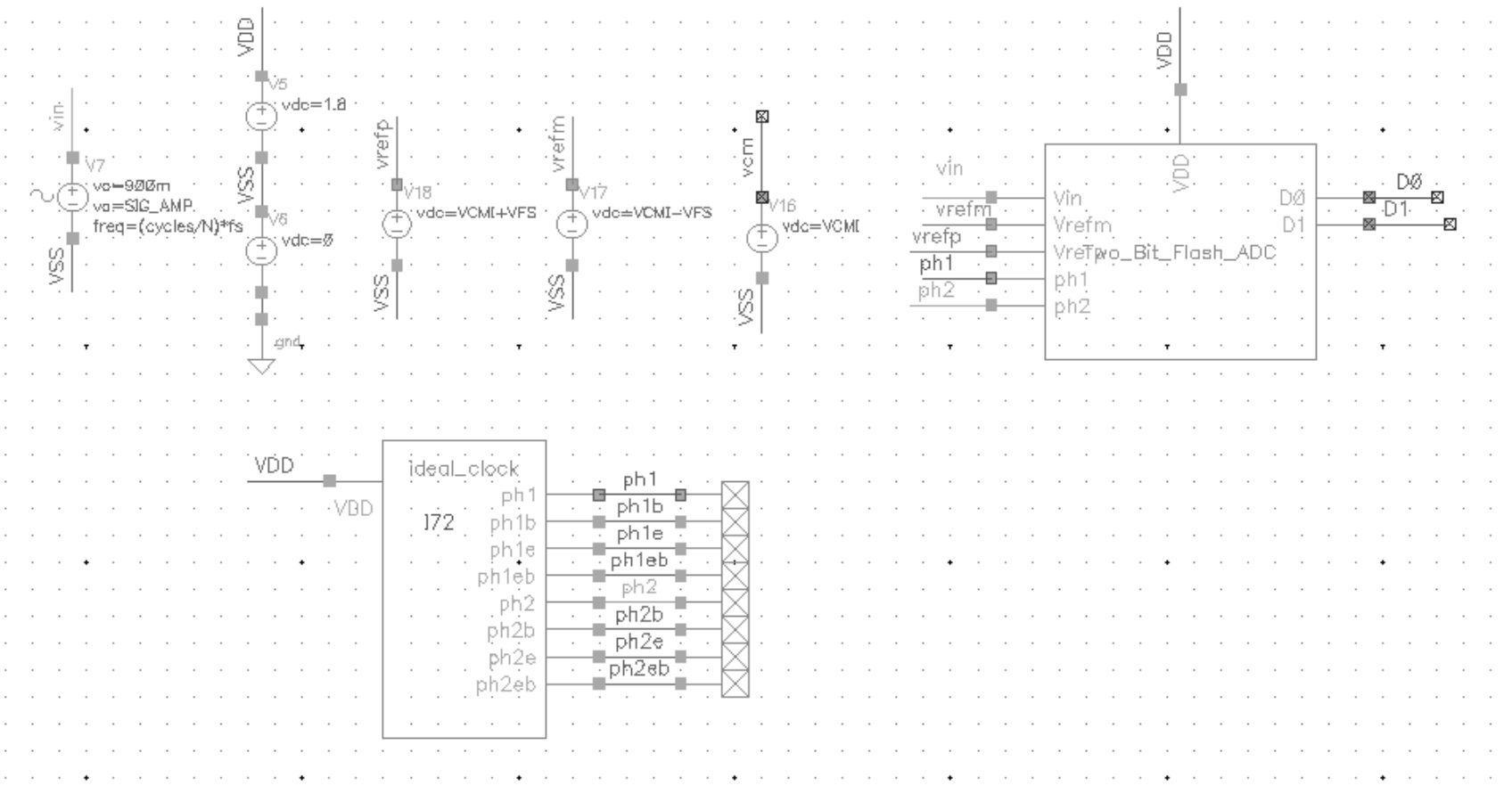
Simulation Results



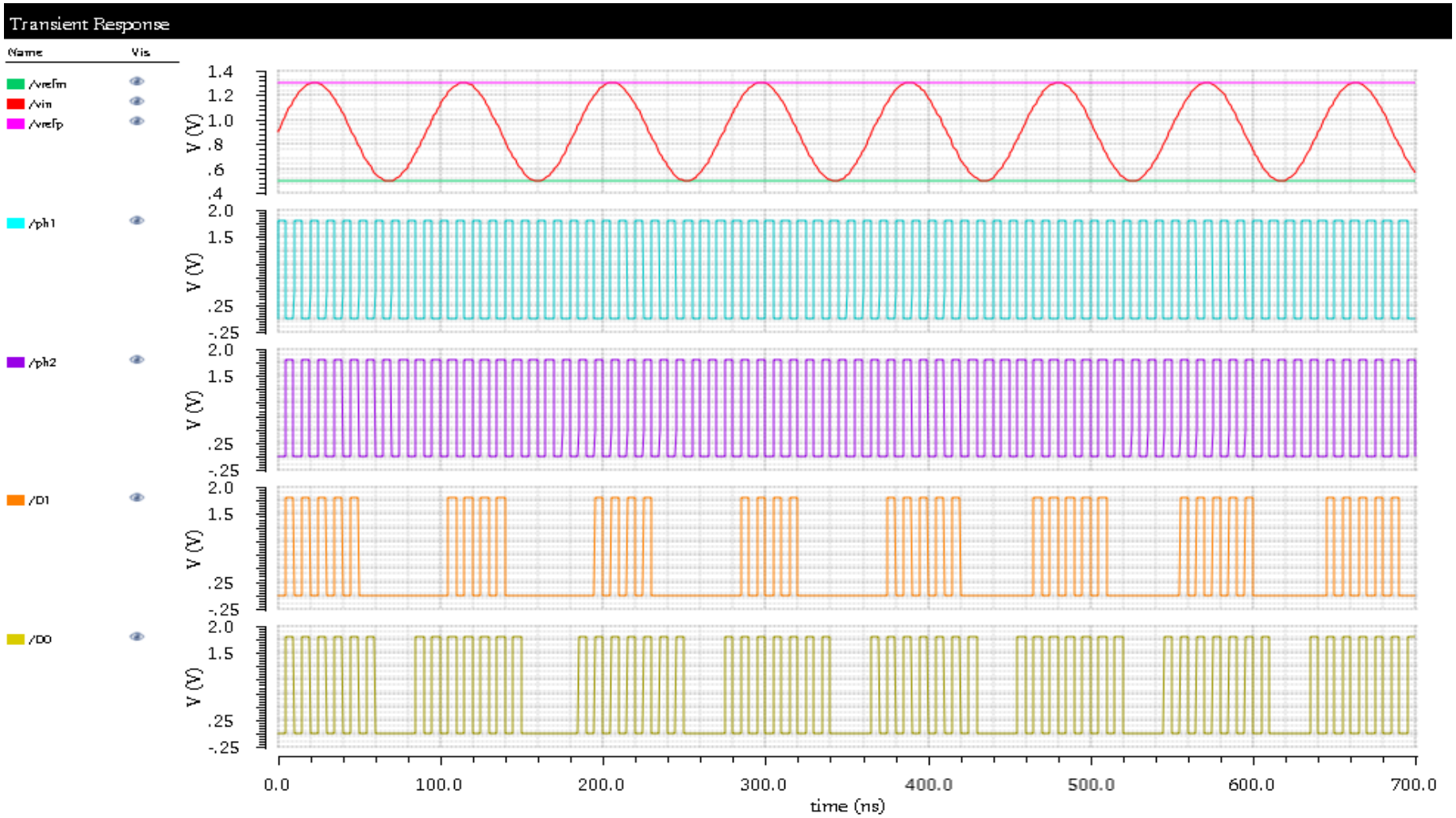
VerilogA:
Two Bit Flash ADC

```
// 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
```

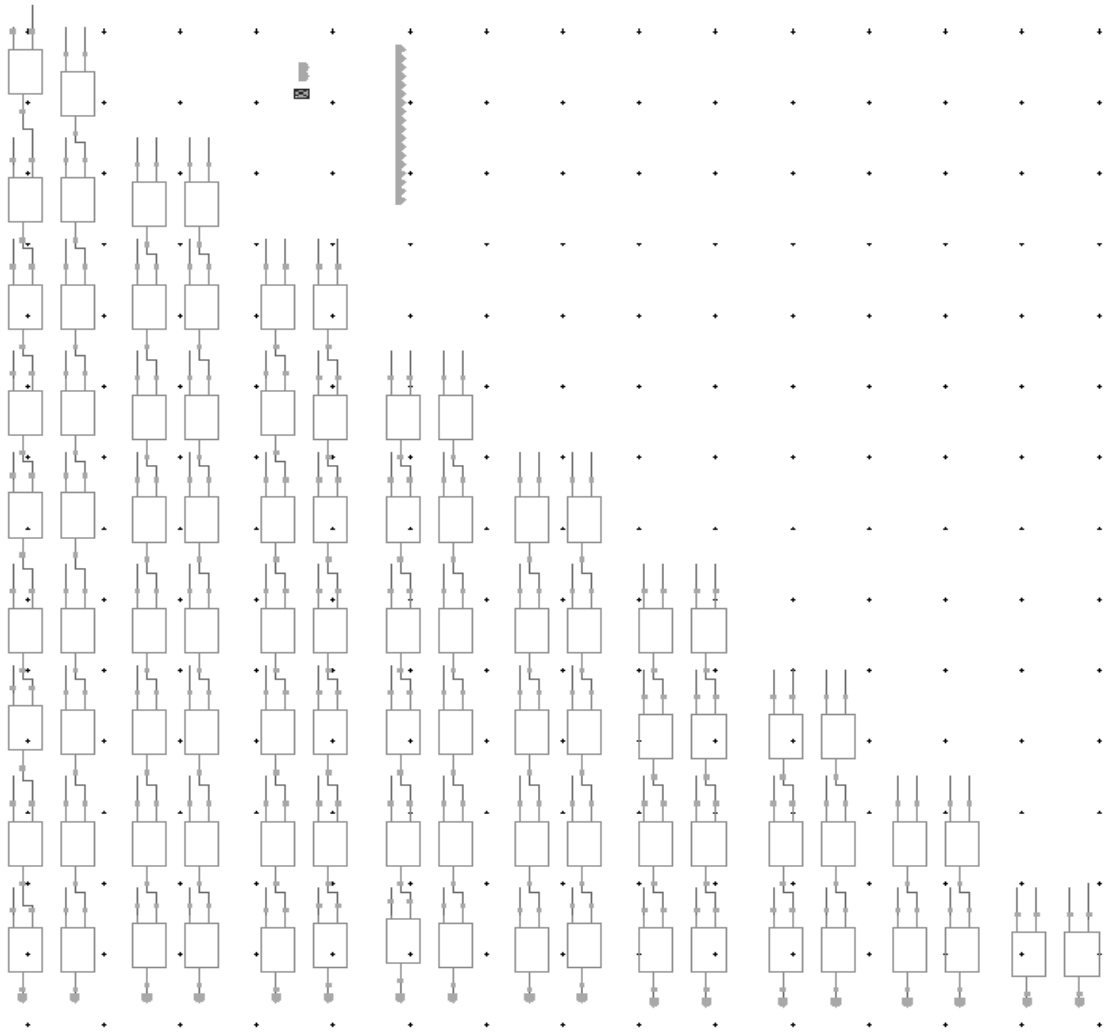
Testbench


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
Vin
Vrefm
Two_Bit_Flash_ADC
ph1
ph2
D0
D1
VDD
ideal_clock
I72
ph1
ph1b
ph1e
ph1eb
ph2
ph2b
ph2e
ph2eb

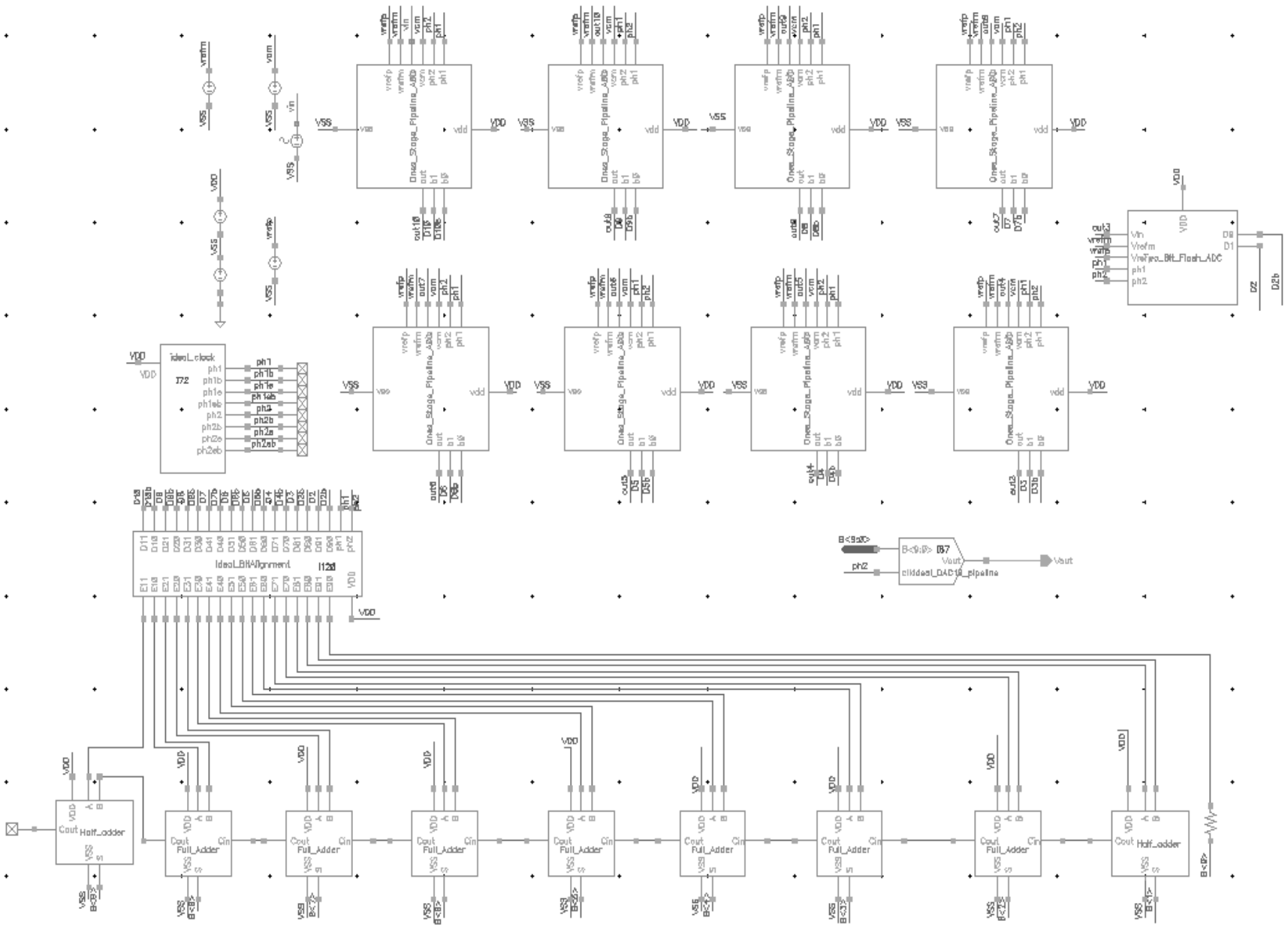
Simulation Results



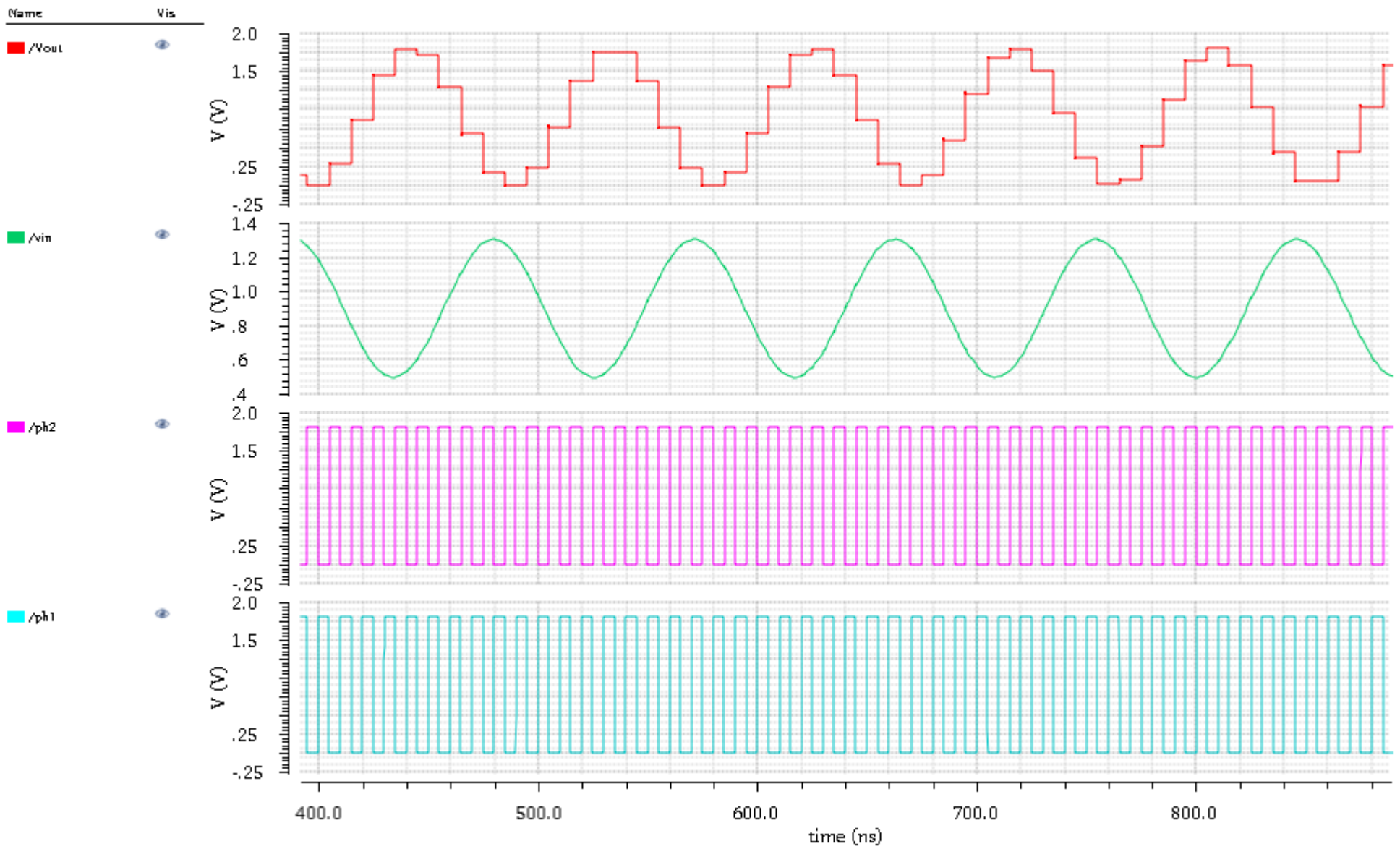
Bit Alignment Circuit



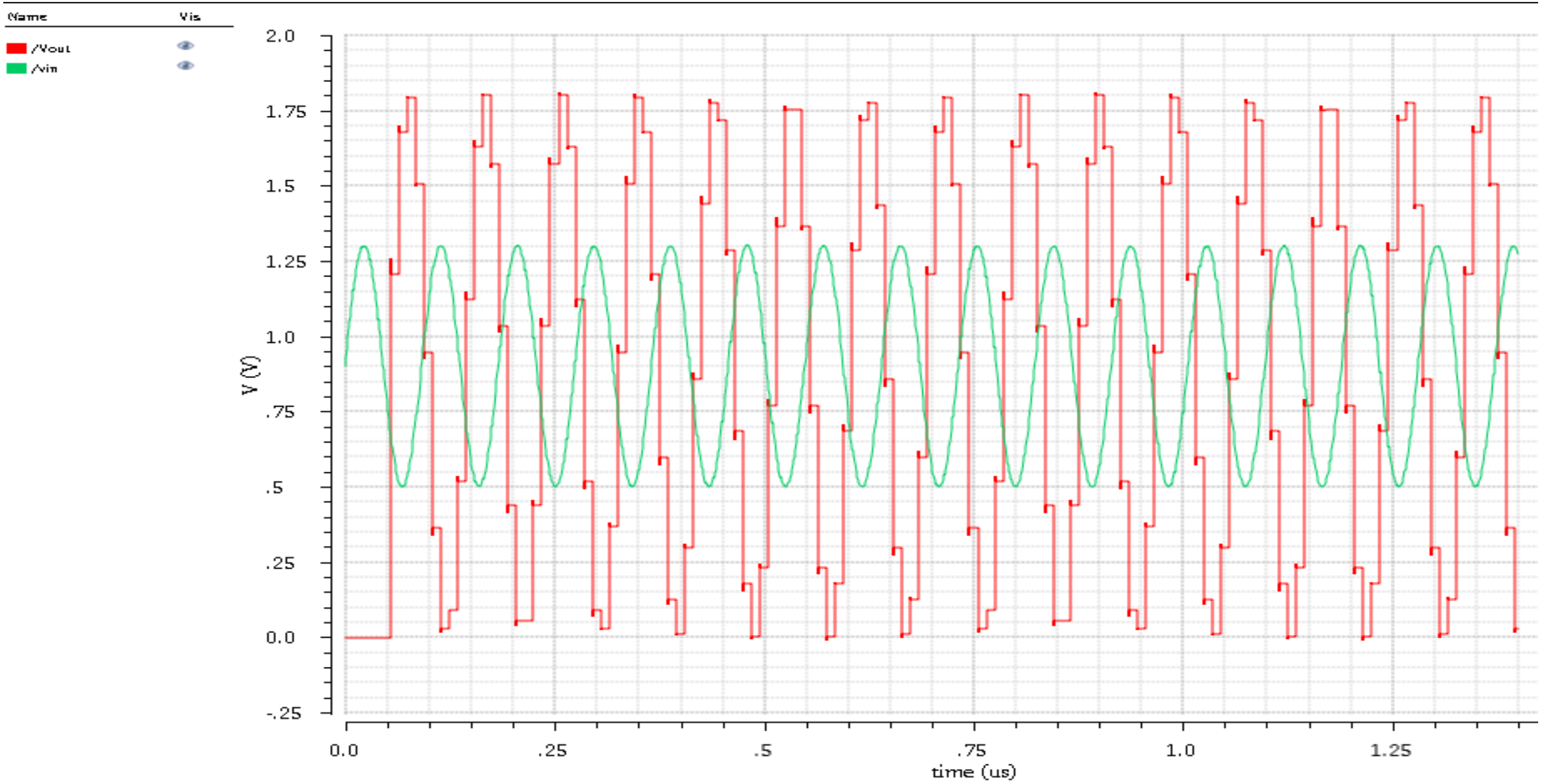
10 Bit Pipe-Line ADC Test Bench



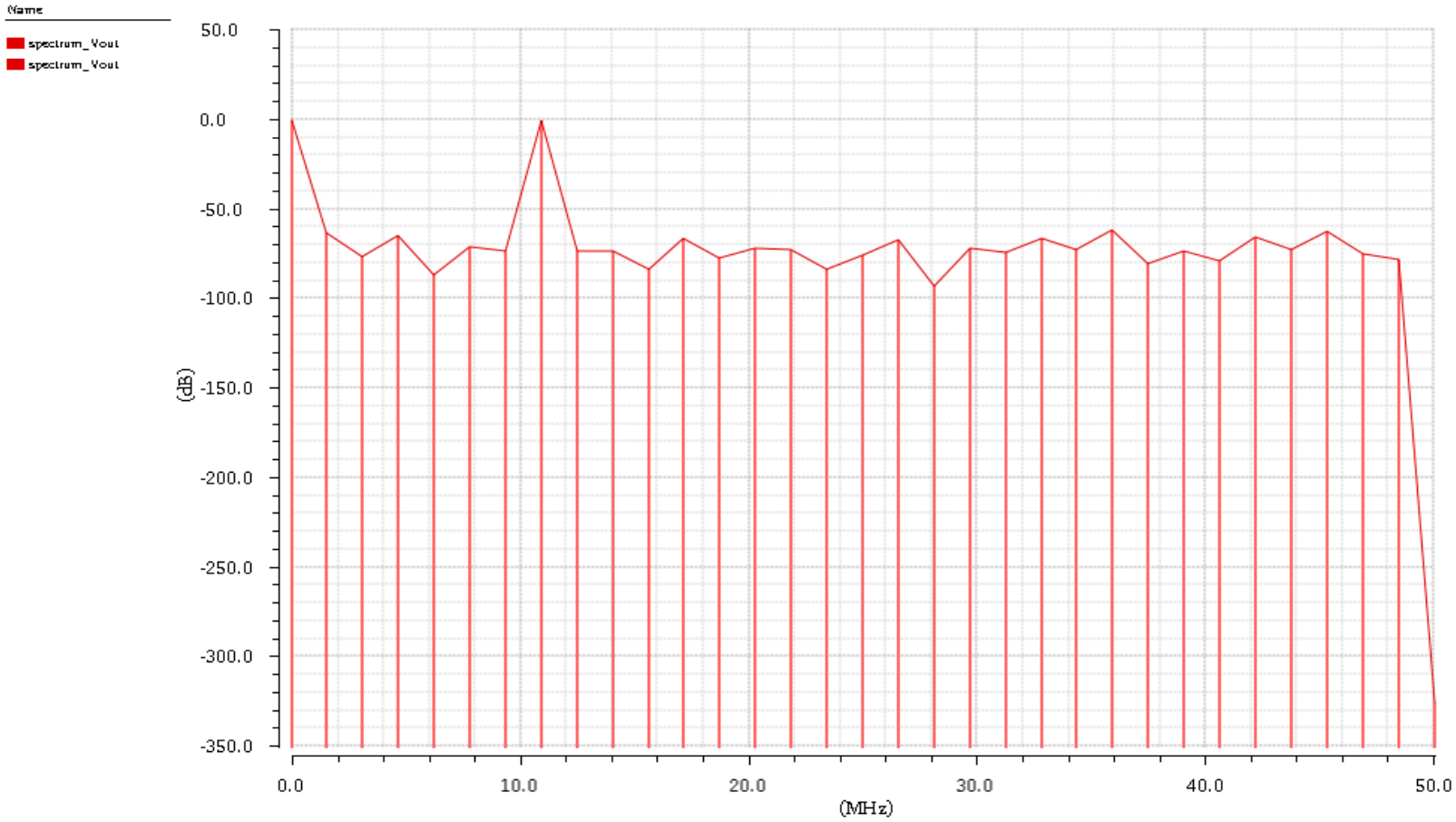
Transient Response



Transient Response



FFT



Spectrum

Signal/Expr Names
Vout

Input wave type	Time Domain Waveform	
FFT Input method	Calculate Sample Frequency	
Start/Stop Time	109n	749n
Sample Count/Freq	64	100.0M
Window Type	Rectangular	
Plot FFT (Units)	☑	dB
Start/End Freq 1.563M:50.00M	S 714.3k	365.7M
Signal bins	0	
Peak Sat. Level	0.0	
Harmonics	1	
Analysis Type	Signal Analysis	
Plot Mode	New Subwindow	

Plot

Outputs

Measurement	Value
Vout	
ENOB	8.6487803 (bits)
SINAD	53.828658 (dB)
SNR	53.828658 (dB)
SFDR	61.207841 (dB)
THD	0 (%)
Signal ...	-0.91614038 (dB)
DC Po...	-0.89926002 (dB)
Noise F...	-69.796298 (dB)
Noise F...	-131.59662 (dB)