Visvesvaraya National Institute
of
Technology, Nagpur

Device Modelling
(ENP302)

Lab Report

Submitted By :
Shruti Murarka (BT18ECE099)
Semester 6
Electronics and Communication Engineering Dept.

Submitted To :
Vivek Pawar (Course Instructor)
Dr. Ganesh Patil (Course Coordinator)

Contents

1 Assignment 1: Full Wave Rectifier 2
2 Assignment 2: Clipper, Clamper and Boolean Circuit 11
3 Assignment 3: CE Amplifier & Oscillators 25
4 Assignment 4: OPAMP 38
5 Assignment 5: FET 59
6 Assignment 6: GATES 72
7 Assignment 7: GATES 87
8 Assignment 8: Flip-Flops 96

Assignment 1: Full Wave Rectifier

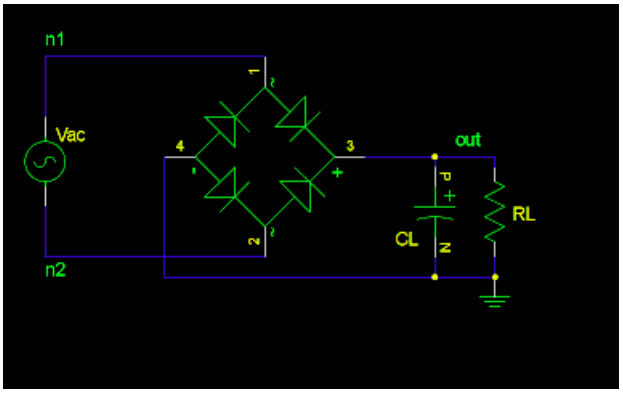


Figure 1: Full Wave Rectifier with RC Load.

Q1:

Aim

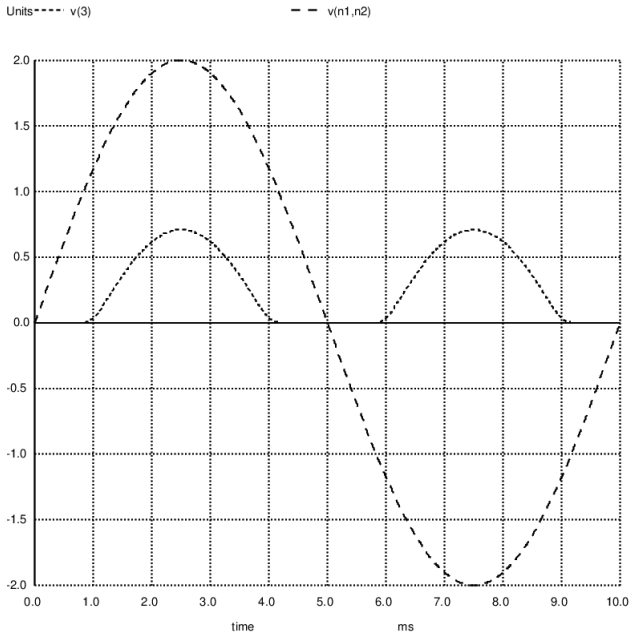
Simulate Full Wave Rectifier with Resistance as the Load using NGSPICE. Plot the Current and Voltage waveform at load.

Code

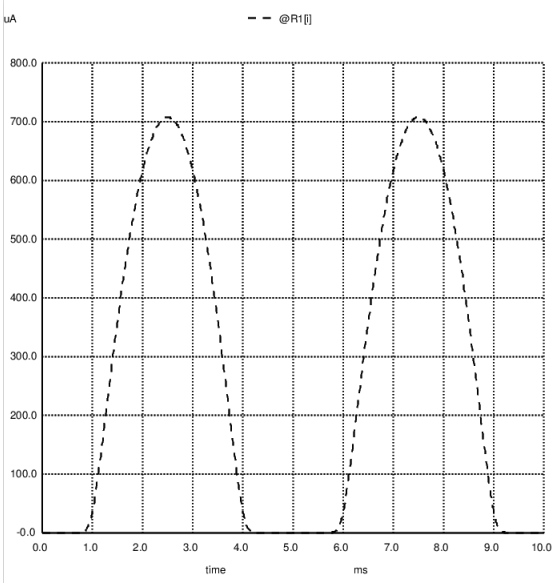
```
* fullwave rectiifier resistor
* Refer figure1 for nodes
* n1, n2 and out(3) are nodes
v1 n1 n2 sin(0 2 100)
D1 n1 3 D_shru
D2 n2 3 D_shru
D3 0 n1 D_shru
D4 0 n2 D_shru
R1 3 0 1K
.model D_shru D
.tran 10us 10ms *step of 10usec, total time of 10msec
.control
run
plot v(3)
plot v(n1,n2)
.options savecurrents
plot @R1[i]
.endc a
.end
```

Output

For developing & simulating the circuits, Ngspice is used.



(a) Input & Output Voltage Waveform.



(b) Output Current around R

Figure 2: FWR with load R.

Observations

It can be observed from Fig 2 that the the signal is fully rectified and there is no negative part left but there is a voltage drop because of diodes. Fig 2 shows the current at Load.

Conclusion

Hence we had successfully simulated Full Wave Rectifier with Resistance as the Load using NGSPICE.

Q2:

Aim

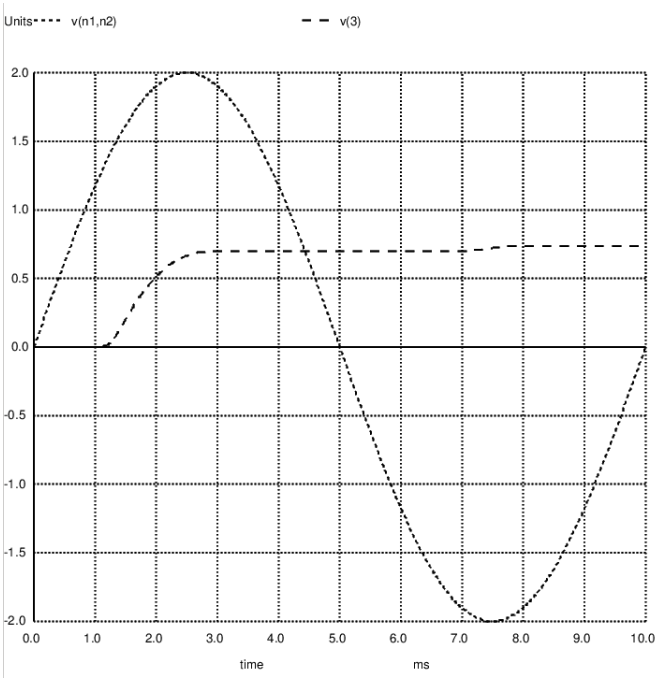
Simulate Full Wave Rectifier with Capacitor as the Load using NGSPICE. Plot the Current and Voltage waveform at load.

Code

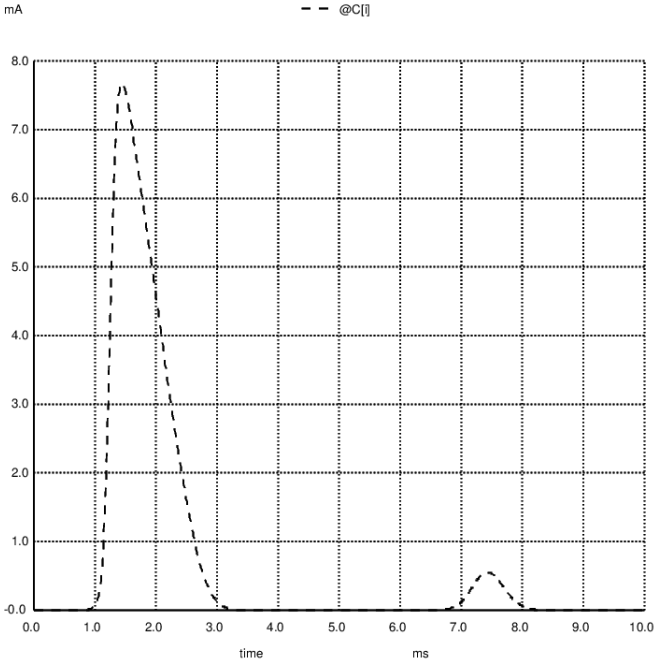
```
* fullwave rectiifier capacitor

*Refer figure1 for nodes
* n1, n2 and out(3) are nodes
v1 n1 n2 sin(0 2 100)

D1 n1 3 D_shru
D2 n2 3 D_shru
D3 0 n1 D_shru
D4 0 n2 D_shru
C 3 0 9.9uF

.model D_shru D
.tran 10us 10ms *step of 10usec, total time of 10msec
.control
run
plot v(3)
plot v(n1,n2)
.options savecurrents
plot @C[i]
.endc a
.end
```

Output

For developing & simulating the circuits, Ngspice is used.



(a) Input & Output Voltage Waveform.



(b) Output Current around C

Figure 3: FWR with load C.

Observations

It can be observed from Fig 3 that the the signal is fully rectified and there is no negative part left. But there is a voltage drop because of diodes. Fig. 3 shows the current at Load. **Here we have taken capacitor value as 9.9uF which is my roll number.**

Conclusion

Hence we had successfully simulated Full Wave Rectifier with Capacitance as the Load using NGSPICE.

Q3:

Aim

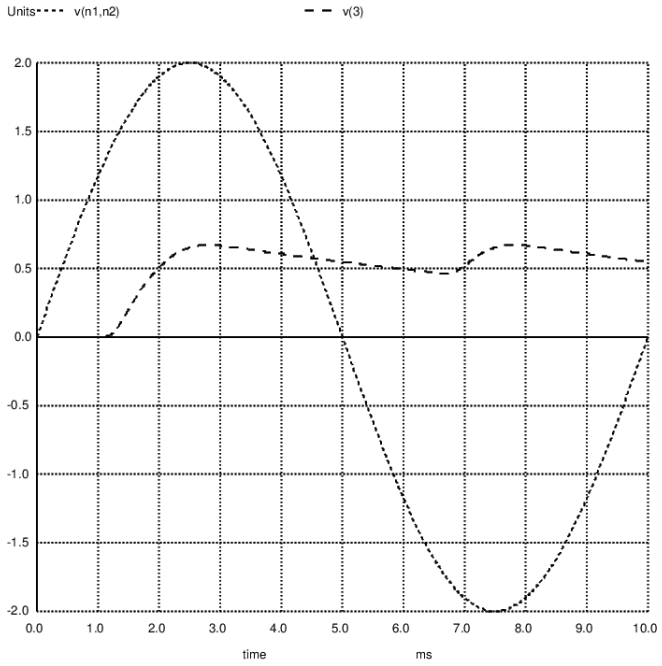
Simulate Full Wave Rectifier with Resistance and Capacitance Parallel combination as a Load using NGSPICE. Plot the Current and Voltage waveform at load.

Code

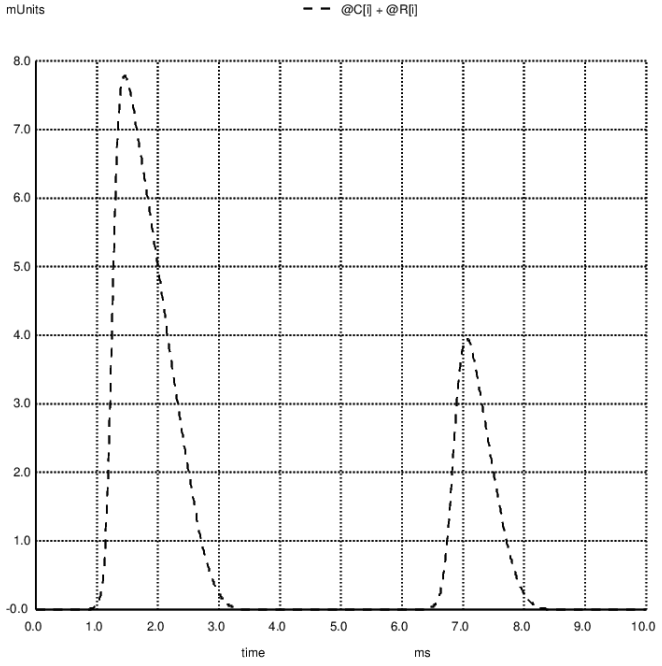
```
*fullwave rectifier RC

*Refer figure1 for nodes
* n1, n2 and out(3) are nodes
v1 n1 n2 sin(0 2 100)

D1 n1 3 D_shru
D2 n2 3 D_shru
D3 0 n1 D_shru
D4 0 n2 D_shru
R 3 0 1K
C 3 0 9.9uF

.model D_shru D
.tran 10us 10ms *step of 10usec, total time of 10msec
.control
run
plot v(3)
plot v(n1,n2)
.options savecurrents
plot @C[i] + @R[i]
.endc
.end
```

Output

For developing & simulating the circuits, Ngspice is used.



(a) Input & Output Voltage Waveform.



(b) Total Output Current

Figure 4: FWR with load RC.

Observations

It can be observed from Fig 4 that the the signal is fully rectified and there is no negative part left. But there is a voltage drop because of diodes. Fig 4 shows the current at Load. **Here we have taken capacitor value as 9.9uF which is my roll number.**

Conclusion

Hence we had successfully simulated Full Wave Rectifier with Parallel RC Combination as the Load using NGSPICE.

Assignment 2: Clipper, Clamper and Boolean Circuit

Q1:

Aim

Simulate the clipper circuit which clips the voltage above +3V and below -3V using NGSPICE.

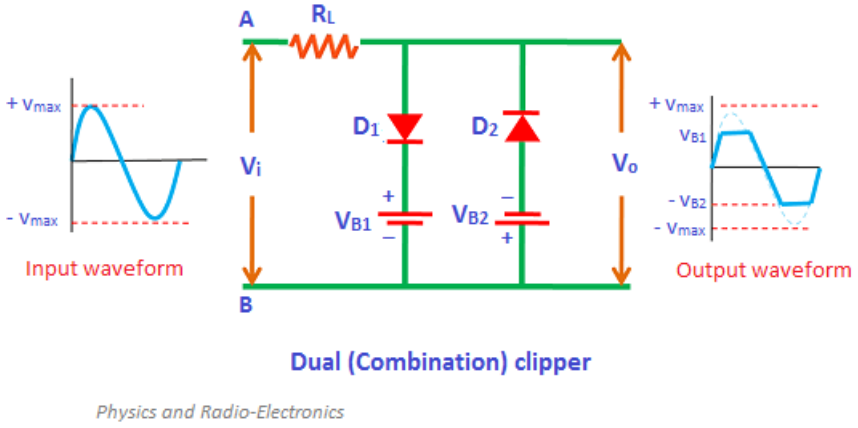


Figure 5: Combination Biased Diode Clipper.

Code

```
1  *dual_clipper.cir
2  *Refer figure5 for nodes
3  v1 1 0 sin(0 5 100 0 0 0)
4  r1 1 2 9.9k
5  d1 2 4 D_shru
6  v2 4 0 dc 2.4
7  d2 5 2 D_shru
8  v3 0 5 dc 2.4
9  .model D_shru D
10 .tran 1u 40m
11 .control
12 run
13 plot v(1) v(2)
14 .endc
15 .end
```

Output

For developing & simulating the circuits, Ngspice is used.

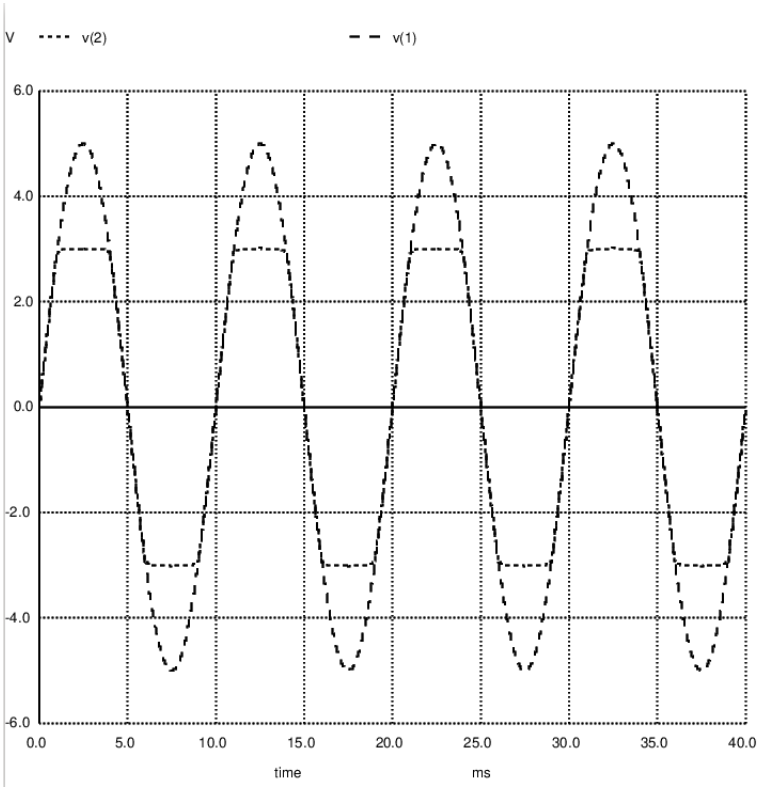


Figure 6: Input and output voltage waveforms.

Observations

A positive dual diode clipper has a range of conduction. Outside this range, the output does not follow the input. We have the conduction range from $[-3, 3]$ in this example. It can be observed in the Fig 6. **Here we have taken resistor value as 9.9K which is my roll number.**

$$|V_{dc}| = 3 - V_{fc} = 3 - 0.6 = 2.4V$$

Where V_{fc} is the forward cut-in voltage of the diode.

Conclusion

Hence we had successfully simulated clipper circuit which clips the voltage above +3V and below -3V.

Q2:

Aim

Simulate the Voltage Tripler Circuit using NGSPICE.

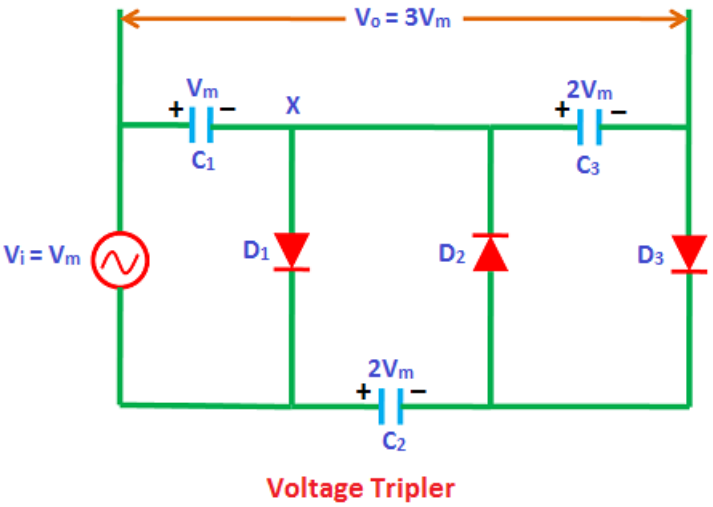


Figure 7: Voltage Tripler Circuit.

Code

```
*tripler.cir

*Refer figure7 for nodes
v1 1 0 sin(0 5 100 0 0 0)
c1 1 2 9.9uF
d1 2 0 D_shru
c2 0 5 9.9uF
d2 5 2 D_shru
c3 2 4 9.9uF
d3 4 5 D_shru
.model D_shru D
.tran 1u 300m
.control
run
plot v(1) v(1, 4)
.endc
.end
```

Output

For developing & simulating the circuits, Ngspice is used.

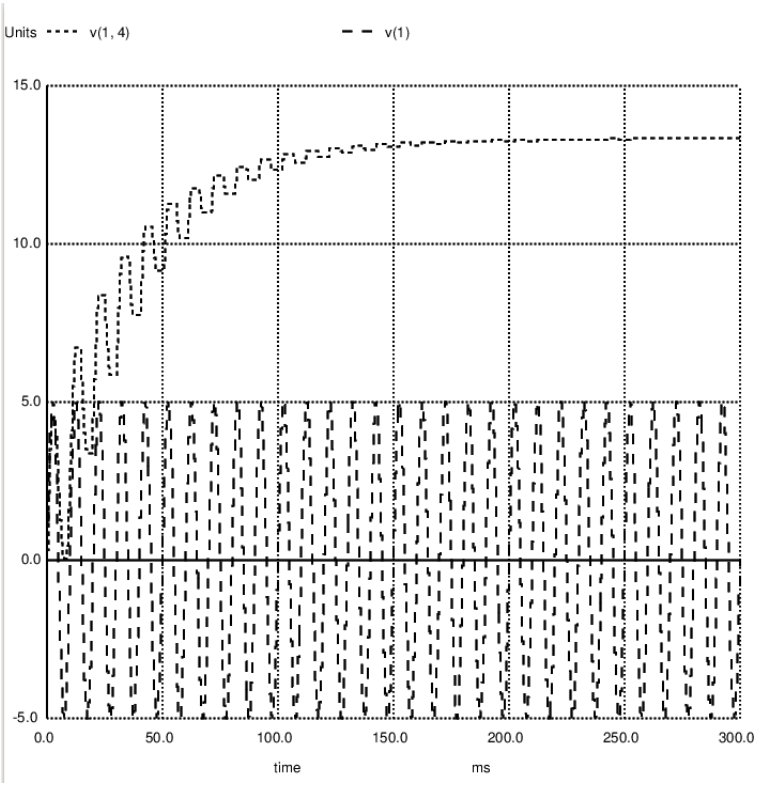


Figure 8: Input and output voltage waveforms.

Observations

In a tripler crcuit as the name suggests, we expect the voltage to triple. This happens as steady state is achieved. It can be observed in the Fig 8. **Here we have taken capacitor value as 9.9uF which is my roll number.**

$$|V_{dc}| = 3 * |V_{ac}| - 3 * V_{fc} = 3 * 5 - 3 * 0.6 = 13.2V$$

Where V_{fc} is the forward cut-in voltage of the diode.

Conclusion

Hence we successfully simulated the Voltage Tripler Circuit.

Q3:

Aim

Simulate the Clamper Circuit which Clamps the Voltage above the -9V with peak to peak voltage 20V using NGSPICE.

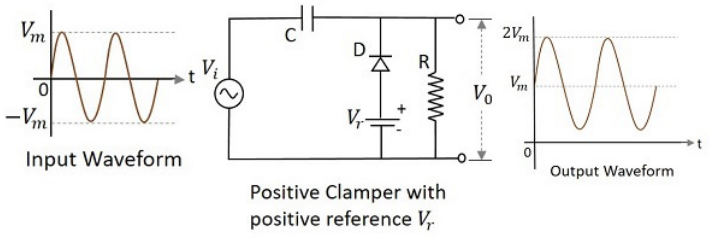


Figure 9: Positive clamper with positive vr Circuit.

Code

The DC reference voltage (V_{dc}) required in series with the diode is given by:

$$|V_{dc}| = |V_{low}| - V_{fc} = 1 - 0.6 = 0.4V$$

Where V_{fc} is the forward cut-in voltage of the diode and V_{low} is the lowest voltage peak.

```
1  *clamper.cir
2  * Refer figure 9 for nodes
3  v1 1 0 sin(0 10 100 0 0 0)
4  c1 1 2 1mF
5  d1 3 2 D_shru
6  vd 3 0 dc 0.3
7  r1 2 0 9.9k
8  .model D_shru D
9  .tran 1u 40m
10 .control
11 run
12 plot v(1) v(2)
13 .endc
14 .end
```

Output

For developing & simulating the circuits, Ngspice is used.

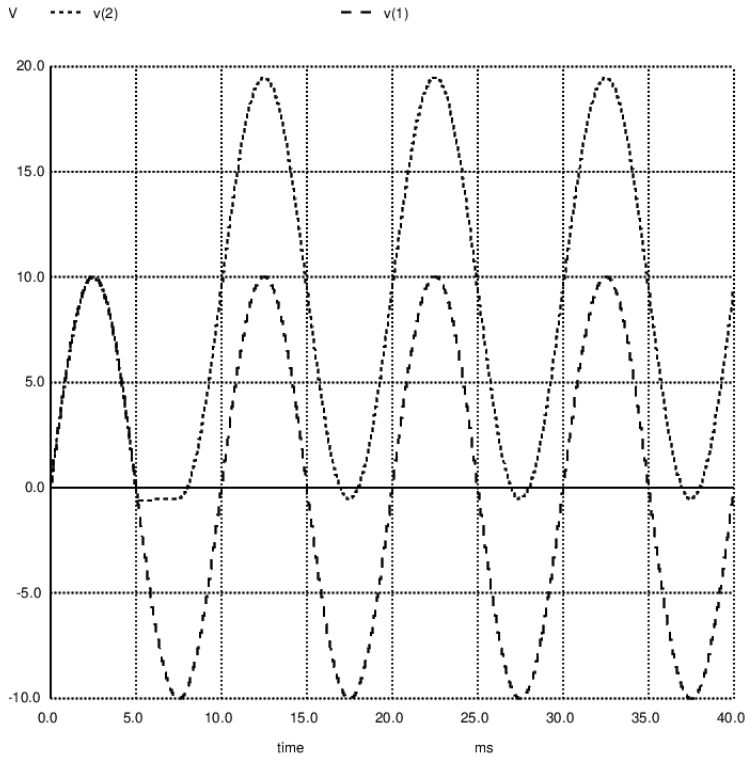


Figure 10: Input and output voltage waveforms.

Observations

A positive clamper shifts the entire input waveform to the 'Above-Zero' position. Further, if we add a DC bias, this zero can easily be changed. Here a -9V bias shifts the input waveform just 1 unit upwards. It can be observed in the Fig 10. **Here we have taken resistor value as 9.9K which is my roll number.**

Conclusion

Hence we had successfully simulated the Clamper Circuit which Clamps the Voltage above the -9V with peak to peak voltage 20V.

Q4:

Aim

Simulate the Boolean Functions given below using NGSPICE.
a) A+B.C b) A+B+C

Code for Diode-based Circuits

a) A+B.C

```
* A+BC
va 6 0 pulse(0 5v 0 0 0 10ms 20ms)
vb 1 0 pulse(0 5v 0 0 0 20ms 40ms)
vc 2 0 pulse(0 5v 0 0 0 30ms 60ms)

d1 3 1 D_shru
d2 3 2 D_shru
r2 3 4 9.9K
v2 4 0 10v
d3 3 5 D_shru
d4 6 5 D_shru
r3 5 0 9.9K
.model D_shru D
.control
tran 10us 60ms
plot V(6) v(1)+10 v(2)+20 V(5)+30
.endc
.end
```

b) A+B+C

```
* A+B+C
va 1 0 pulse(0 5v 0 0 0 20ms 40ms)
vb 2 0 pulse(0 5v 0 0 0 20ms 40ms)
vc 3 0 pulse(0 5v 0 0 0 40ms 80ms)
d1 1 4 D_shru
d2 2 4 D_shru
d3 3 4 D_shru
r1 4 0 9.9k
.model D_shru D
.tran 10u 80m
.control
run
plot v(1) v(2)+10 v(3)+20 v(4)+30
.endc
```

```
15 .end
```

Code for Transistor-based Circuits

a) A+B.C

```
1  * A + BC
2  V_bias  bias 0  DC 9
3  Vin_c   n1 0    PULSE(0 5 0 0 0 05m 10m)
4  Vin_b   n2 0    PULSE(0 5 0 0 0 10m 20m)
5  Vin_a   n3 0    PULSE(0 5 0 0 0 20m 40m)
6  Rin_c   n1 b1   10K
7  Rin_b   n2 b2   10K
8  Rin_a   n3 b3   10K
9  Q1  bias b1 intr my_BJT
10 Q2  intr b2 out1 my_BJT
11 R1  out1 0  100G
12 Q3  bias out1 out my_BJT
13 Q4  bias b3 out my_BJT
14 R2  out 0   1000K
15
16 .MODEL my_BJT npn is=1e-15
17 .tran 1u 80m
18 .control
19 run
20 plot v(n1) v(n2)+10 v(n3)+20 v(out)+30
21 .endc
22 .end
```

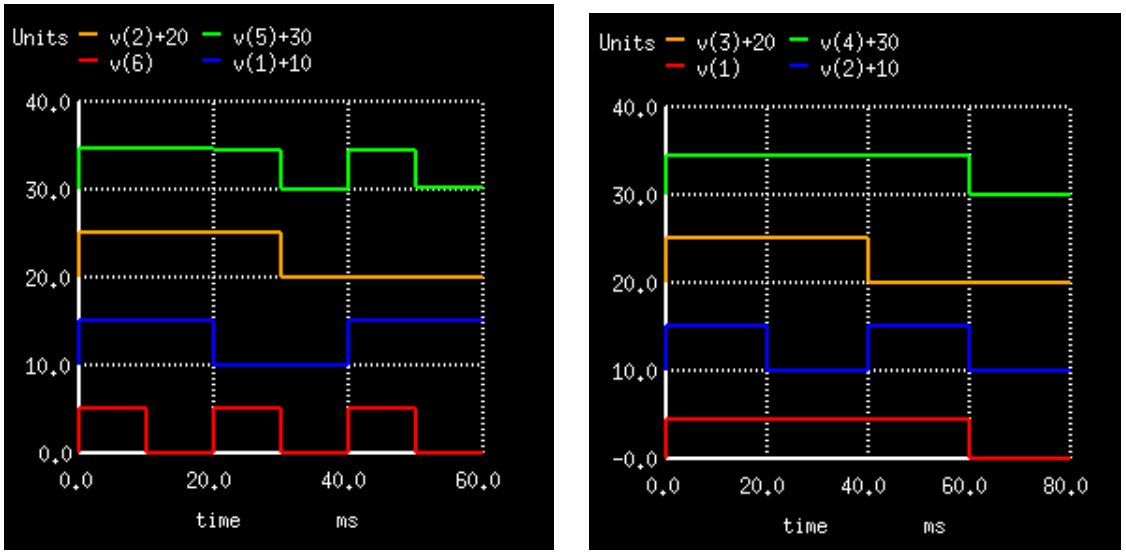
b) A+B+C

```
1  * A + B + C
2  V_bias  bias 0  DC 9
3
4  Vin_a   n1 0    PULSE(0 5 0 0 0 05m 10m)
5  Vin_b   n2 0    PULSE(0 5 0 0 0 10m 20m)
6  Vin_c   n3 0    PULSE(0 5 0 0 0 20m 40m)
7  Rin_a   n1 b1   10K
8  Rin_b   n2 b2   10K
9  Rin_c   n3 b3   10K
10 Q1  bias b1 out my_BJT
11 Q2  bias b2 out my_BJT
12 Q3  bias b3 out my_BJT
13 RL  out 0   100G
14
15 .MODEL my_BJT npn is=1e-15
```

```
16 .tran 1u 80m    * Transient response in steps of 1us and upto 80 ms
17 .control
18 run
19 plot v(n1) v(n2)+10 v(n3)+20 v(out)+30
20 .endc
21 .end
```

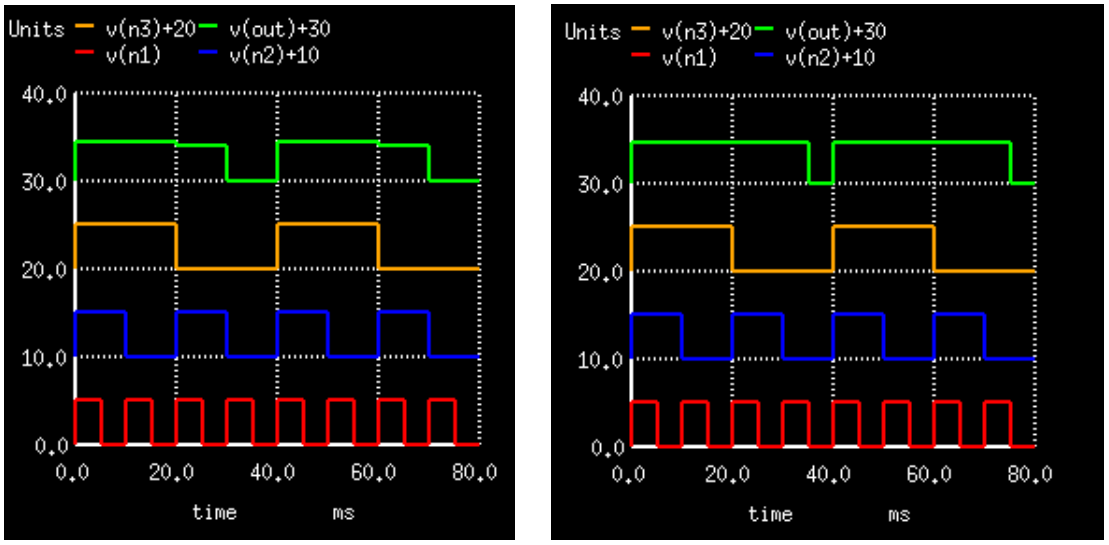
Output

For developing & simulating the circuits, Ngspice is used.
In Fig 11 & Fig 12, red=A, blue=B, yellow= C & green= output



(a) A+B.C. (b) A+B+C

Figure 11: Output for diode - based circuits.



(a) A+B.C. (b) A+B+C

Figure 12: Output for transistor - based circuits.

Observations

It can be observed from the fig 11.a and fig 12.a that V(out) = A+BC i.e. AND of the two signals: V(b) and V(c) with OR of the resultant signal with V(a). The signal V(out) has Boolean value 1 when either A has value 1 or when BC has value 1 or when both have value 1. Other than that, it always has Boolean value of 0.
It can be observed from the fig 11.b and fig 12.b that V(out) = A+B+C i.e. OR of the three signals: V(a), V(b) and V(c). The signal V(out) has Boolean value 0 only when all the three signals A, B, C have value 0. Other than that, it has Boolean value 1 when either or all of the inputs are 1.

Conclusion

We successfully simulated the given Boolean equations using diodes.

Q5:

Aim

Simulate the Transfer Synthesis Graph Given Below using NGSPICE.

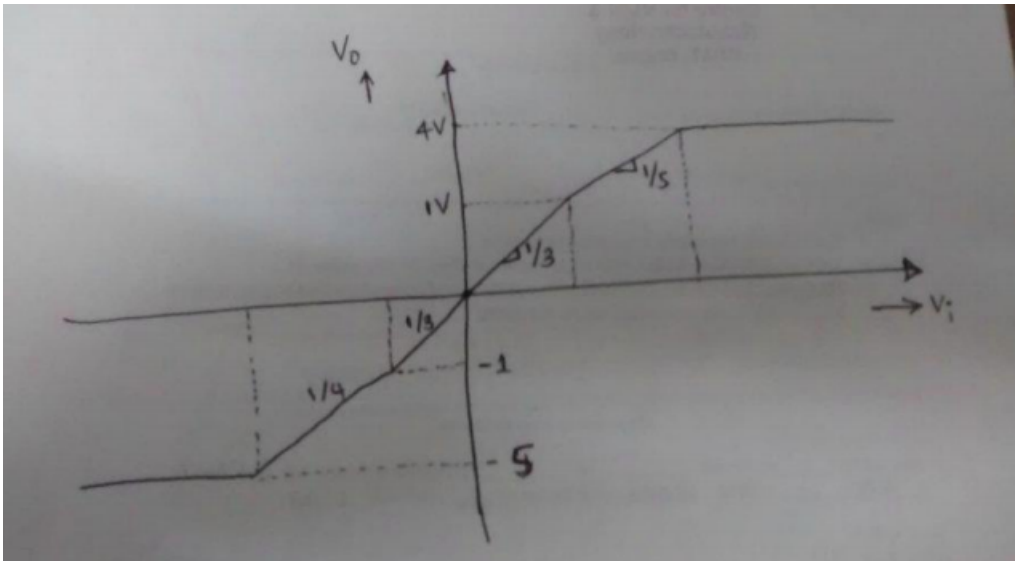


Figure 13: Required Transfer Graph.

Calculations

The given transfer function can be simulated using multiple diode branches as drain switches with suitable voltage cut-offs. The branches also have resistors that attach in parallel with the original 1k lower resistor. Also, the output would is to be clipped at +4 and -5 Volts and that too is achieved using diodes. The slope of $\frac{1}{3}$ is achieved by using a voltage divider circuit with 2kΩand 1kΩresistances and measuring the output across the 1kΩresistor. When a branch conducts, the resistance in that branch is in parallel with the original 1kΩresistor and is set such that the desired slope is obtained.

$$\begin{aligned} |V_{dc}| &= |V_{cutoff}| - V_{fc} \\ |V1| = |V2| &= 1 - 0.6 = 0.4V \\ |V3| &= 4 - 0.6 = 3.4V \\ |V4| &= 5 - 0.6 = 4.4V \end{aligned}$$

Where V_{fc} is the forward cut-in voltage of the diode $V1, V2, V3, V4$ are the DC voltages applied in series with the diodes. The direction of the diode and DC voltage

sources are determined by whether the conduction needs to happen on the positive side or negative side.

Since, R3 branch conducts from +1 to +4V output, and the slope is $\frac{1}{4}$. Similarly, the R4 branch conducts from -1 to -5V output, and the slope is $\frac{1}{5}$. Hence, the resistances R3 and R4 are given by

$$R3||R1 = R2 * \frac{1}{4}$$
$$\implies R3 = 1k\Omega$$

Similarly,

$$R4||R1||R3 = R4||0.5k\Omega = R2 * \frac{1}{5}$$
$$\implies R4 = 2k\Omega$$

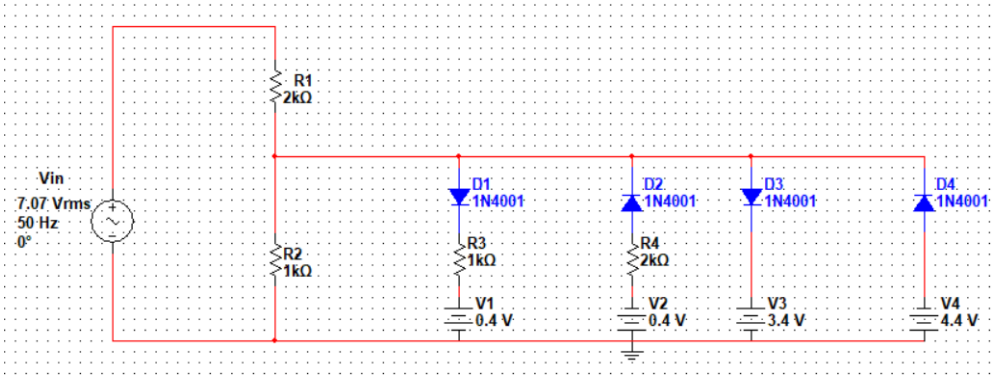


Figure 14: Voltage Divider and Clipper circuit

Code

```
1  * Voltage Divider and Clipper
2
3  Vin n0 0 SIN(0 10 50)
4  R1  n0 n1   2k
5  R2  n1 0    1k
6  D1  n1 n2 D_shru
7  R3  n2 n3 1K
8  V1  n3 0 dc 0.4
9  D2 n4 n1 D_shru
10 R4 n4 n5 2K
```

```
11 V2 0 n5 dc 0.4
12 D3 n1 n6 D_shru
13 V3 n6 0 dc 3.4
14 D4 n7 n1 D_shru
15 V4 0 n7 dc 4.4
16
17 .MODEL D_shru D
18 .control
19 DC Vin -25 25 0.5
20 * DC sweep on X axis from -25 to +25 Volts
21
22 plot V(n1) vs V(n0)
23 .endc
24 .end
```

Output

For developing & simulating the circuits, Ngspice is used.

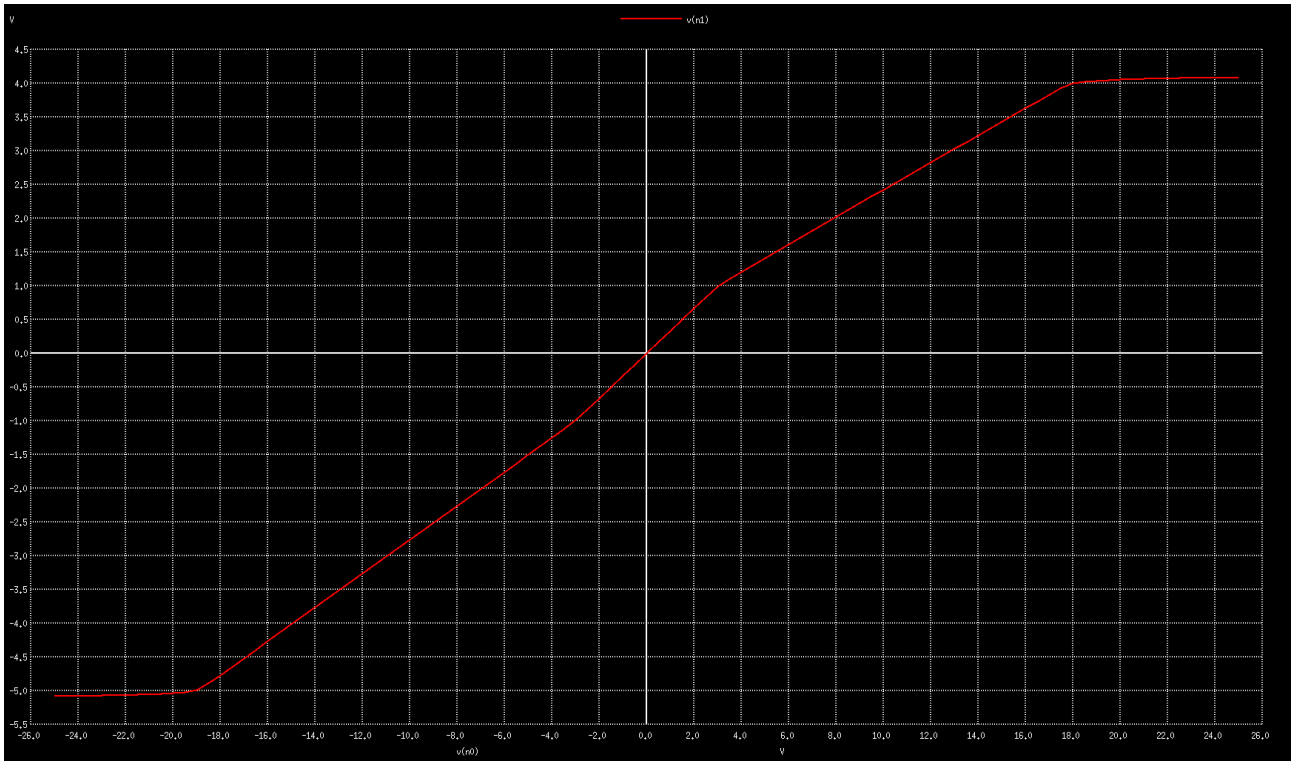


Figure 15: Transfer Synthesis Graph.

Observations

It is possible to obtain the required transfer characteristics graph using simple calculations and knowledge of voltage divider and clipper circuits. As we can see in the fig 15, the Transfer Synthesis Curve obtained is similar to the fig 13.

Conclusion

We had successfully simulated the Transfer Synthesis Graph given in fig 13.

Assignment 3: CE Amplifier & Oscillators

Q1:

Aim

Design and simulate Common Emitter Amplifier using NGspice.

- Find operational point by DC Analysis.
- Plot output and input voltage waveform.
- Plot input output characteristics of CE Amplifier.

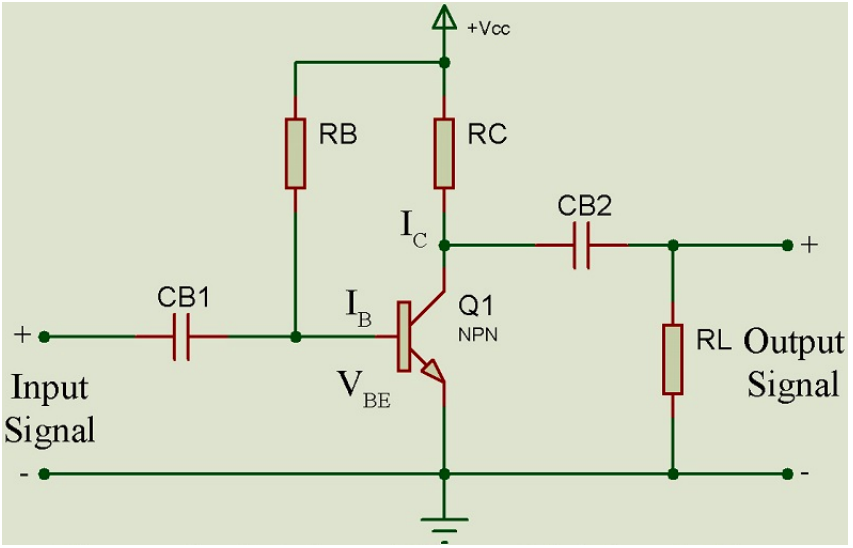


Figure 16: CE Amplifier.

Code

```
1  *Common Emmitter Amplifie: operating point
2
3  vbb 1 0 dc 3.7
4  R1 5 2 68k
5  R2 2 0 10K
6  Rc 4 5 10k
7  vcc 5 0 dc 12v
8  Q1 4 2 3 d_BJT
9  Re 3 0 1k
10
```

```
.model d_BJT npn bf = 60
.op
.control
run
print v(4,3)
print v(5,4)/5k
.endc
.end

*CE Amplifier
vin 1 0 dc 0 ac sine(0 5mV 100 0 0 0)
c1 1 2 10u
vcc 3 0 dc 5
r1 2 3 68k
r2 2 0 10k
r3 3 4 10k

q1 4 2 ne BJT_Shru
re ne 0 1K
ce ne 0 10u
c2 4 5 10u
rload 5 0 100k

.model BJT_Shru npn is=1e-15
.tran 0.1u 20m
.control
run
plot v(1) v(5) title 'CE Amplifier'
.endc
.end

*Common Emmitter Amplifier: input-output Char

vbb 1 0 dc 3.7
*R1 5 2 5k
R2 2 1 5K
Rc 4 5 1k
vcc 5 0 dc 12v
Q1 4 2 3 d_BJT
Re 3 0 1k

.model d_BJT npn bf = 60
.options savecurrents
.control
run
dc vbb 0 3 0.01
plot v(1,2)/5K  vs v(2,3)
.endc
.control
```

```
60 dc vcc 0 8 0.02
61 plot -@Rc[i] vs v(4,3)
62 .endc
63 .end
```

Output

For developing & simulating the circuits, Ngspice is used.

```
Circuit: *common emmitter amplifie: operating point

Doing analysis at TEMP = 27.000000 and TNOM = 27.000000


No. of Data Rows : 1
v(4,3) = 4.654697e+00
v(5,4)/5k = 1.333489e-03
```

Figure 17: CE Amplifier Operational Point.

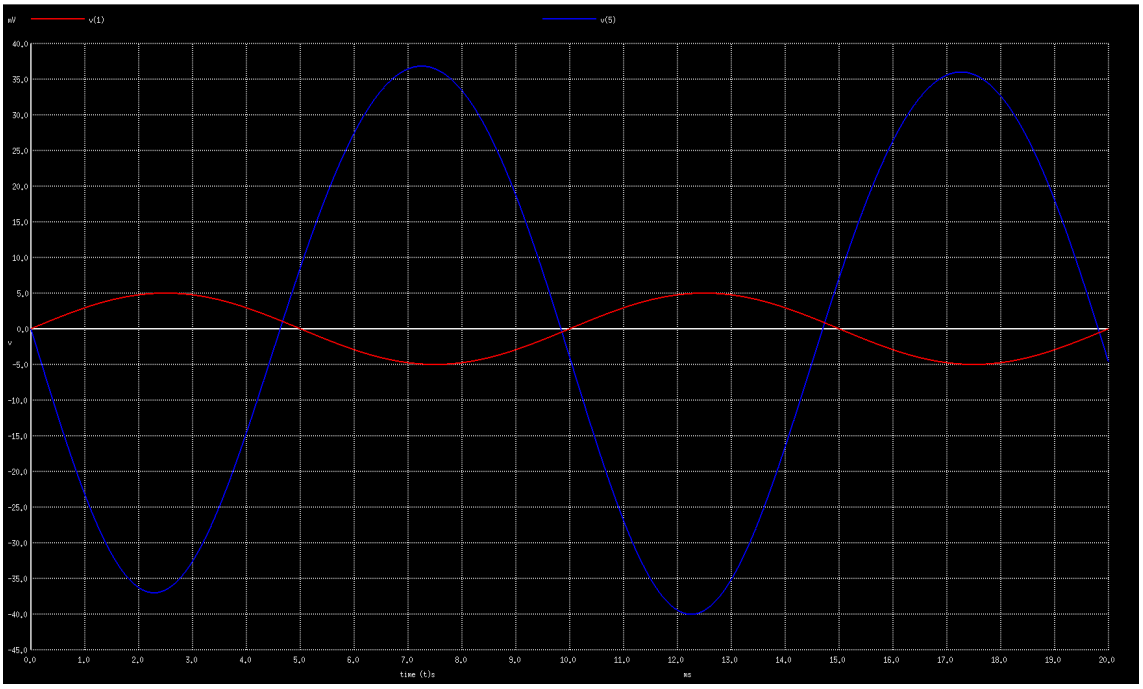
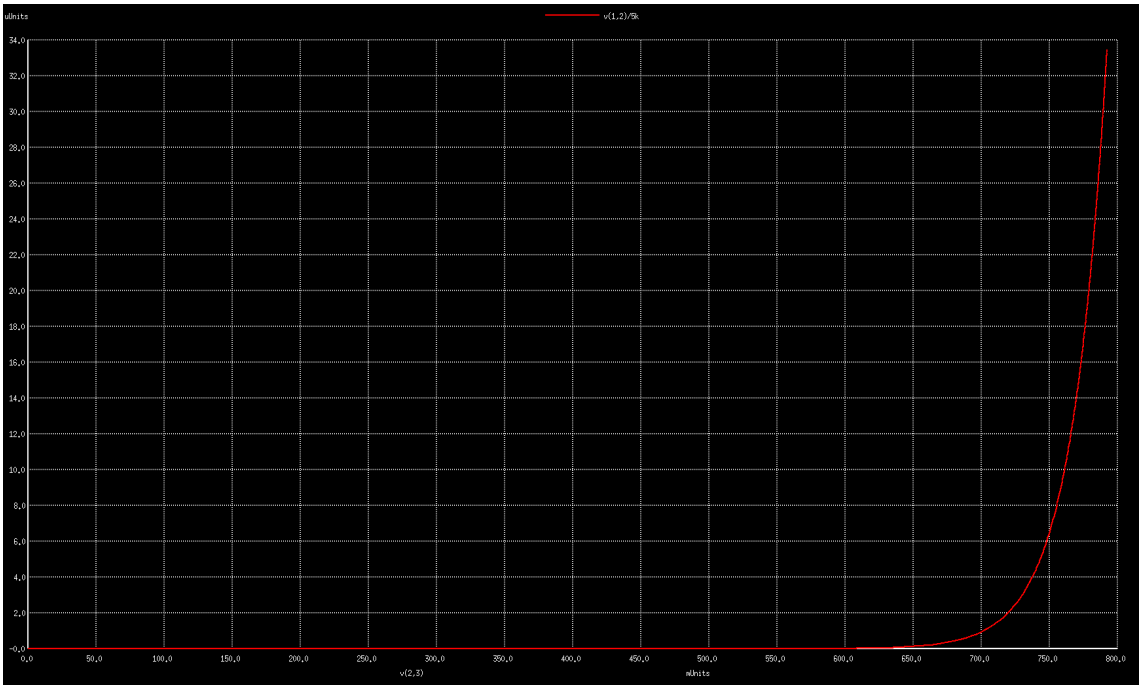
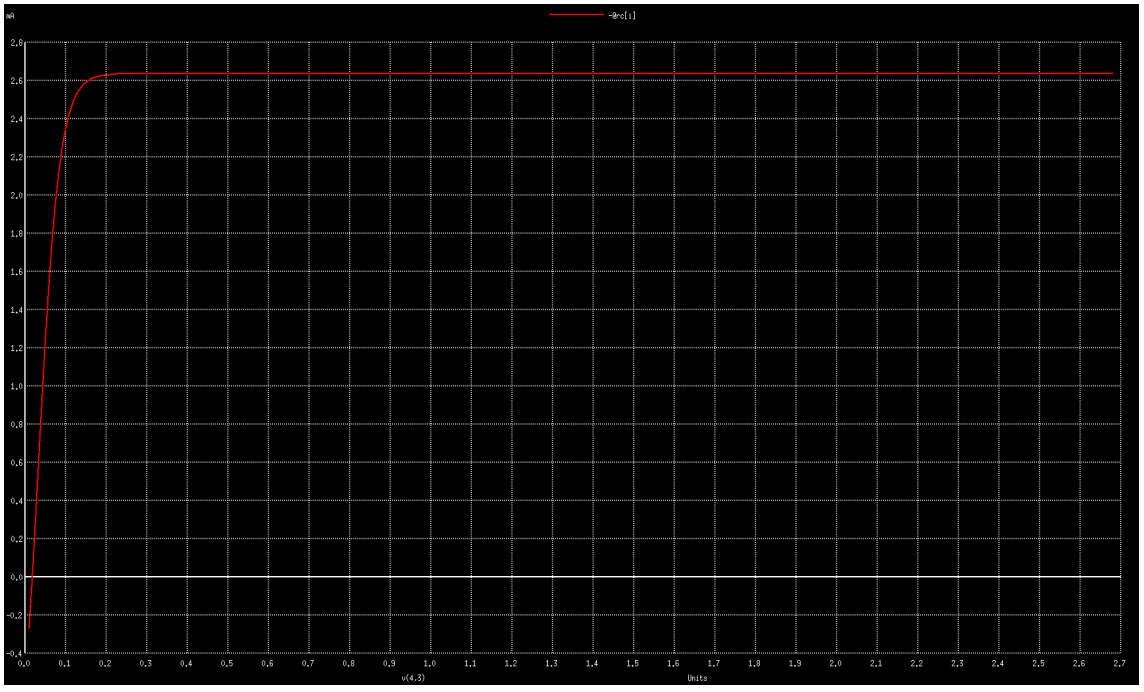


Figure 18: CE Amplifier Input and Output Voltage Waveforms.



(a) Input characteristics.



(b) Output characteristics.

Figure 19: Input and output characteristics of CE Amplifier.

Observations

Any rise in temperature T causes rise in Icbo and hence rise in Ic. This increased current through Re causes increased dc voltage drop across Re and results in reduced net emitter-to-base forward bias, reduced base current Ib and reduced collector current Ic. This reduction in Ic tends to cancel the rise in Ic as caused by temperature rise. Operating point stability is improved in self bias CE Amplifier. There is a phase relationship of 180 degrees in input and output voltage waveforms.

Conclusion

We successfully obtained operational point, input & output characteristics curve by DC analysis and output-input waveforms by AC analysis.

Q2:

Aim

Design and simulate RC Phase Shift Oscillator using NGspice.

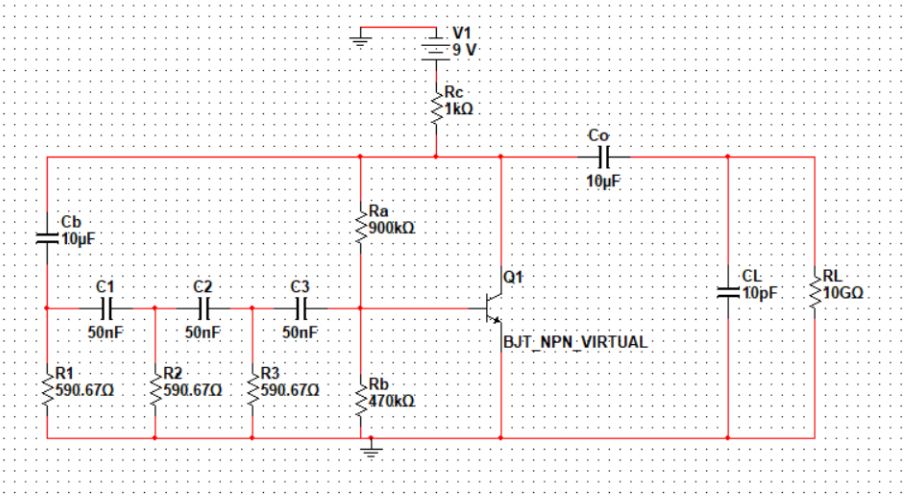


Figure 20: RC Phase Shift Oscillator.

Calculations

In RC phase shift oscillator, the output frequency is given by the formula:

$$f_{out} = \frac{1}{2\pi RC\sqrt{2N}}$$

Where N is the number of times the RC phase shift network is repeated. In this case it is 3. R and C are the values of the Resistor and Capacitor used in repeated RC network. Roll number is 99, hence the expected frequency is $\frac{99}{45} = 2.2kHz$. Fixing the value of capacitance to 50nF, the value of the resistance is given by:

$$R = \frac{1}{2\pi \times f_{out} \times 50 \times 10^{-9}\sqrt{6}} = \frac{1}{2\pi \times 2200 \times 50 \times 10^{-9}\sqrt{6}} = 590.6793949\Omega$$

Code

```
1 * RC Phase Shift oscillator at 2.2 kHz
2 .MODEL NPN_Shru npn is=1e-15
3
```

```
Vb  1 0 DC 9

Q1  nc nb 0 NPN_Shru

RC  1 nc    1K
Ra  nc nb   100K
Rb  nb 0    470K
Cb  nc n1   10u
Co  nc out  10u

R1  n1 0    590.6793949
C1  n1 n2   50n

R2  n2 0    590.6793949
C2  n2 n3   50n

R3  n3 0    590.6793949
C3  n3 nb   50n

RL  out 0   10G
CL  out 0   10p


.tran 1u 25m 8m  *Transient response in steps of 1uS, from 8mS ...
    to 25mS
.control
run
plot v(out)
.endc
.end
```

Output

For developing & simulating the circuits, Ngspice is used.

Observations

The output waveform of the oscillator requires certain time to reach proper amplitude. This is due to the limited charge storing capacity of the capacitors. Also, the output frequency was very close to 2.2kHz and the same can be verified from the graph displayed. The three stages of repeated RC circuit suffice for the total loop phase change of 180°and the Common Emitter BJT amplifier adds another 180°of phase change, ensuring constructive interference and sustained oscillations with unity gain. Hence, satisfying the Barkhausen criteria. Finally, the load resistor and capacitor are chosen such that the load impedance is sufficiently large and minimal leakage occurs on measurement.

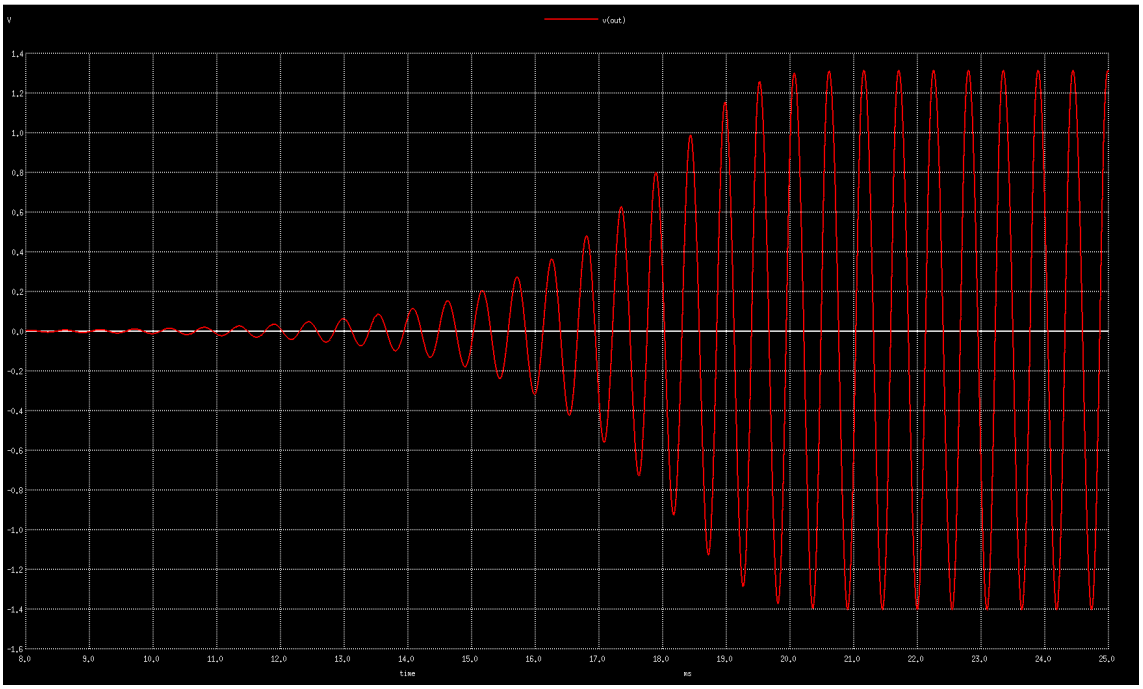


Figure 21: Output waveform of RC Phase Shift Oscillator.

Conclusion

Hence we had successfully simulated RC Phase Shift Oscillator for 2.2KHz ($\frac{99}{45} = 2.2kHz$).

Assignment Q1:

Aim

Design and simulate Colpitt's Oscillator using NGspice.

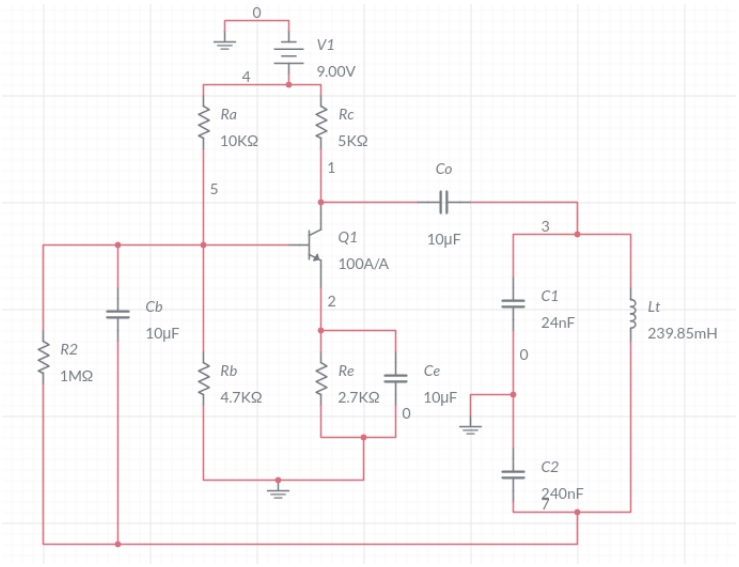


Figure 22: Colpitt Oscillator.

Calculation

In Colpitt oscillator, the output frequency is given by the formula:

$$f_{out} = \frac{1}{2\pi\sqrt{LC_T}}$$

Where $C_T = \frac{C_1 \times C_2}{C_1 + C_2}$. And C is the capacitor used in parallel with the inductors in the Hartley tank. Roll number is 99, hence the required frequency is $\frac{99}{45} = 2.2kHz$. Fixing the value of C_T to 21.82nF, the value of the capacitance is given by:

$$L = \frac{1}{4\pi^2 f_{out}^2 C_T} = \frac{1}{4\pi^2 \times 2200^2 \times 21.82 \times 10^{-9}} = 239.85mH$$

Code

```
1 * Colpritt oscillator at 2.2 kHz
```

```
.MODEL NPN_Shru npn is=1e-15

Vb  4 0 DC 9

Q1  1 5 2 NPN_Shru

Rc  1 4    5K
Re  2 0    2.7K
Ce  2 0    10u
Ra  4 5    10K
Rb  5 0    4.7K
Cb  5 7    10u
R2  5 7    1M

Co  1 3  10u
C1  3 0  24n
Lt  3 7  240m
C2  7 0  240n

.tran 1u 5000u 20u  *Transient response in steps of 1uS, from ...
   20uS to 5000uS
.control
run
plot v(3)
.endc
.end
```

Output

For developing & simulating the circuits, Ngspice is used.

Observations

The output waveform of the oscillator requires certain time to reach proper amplitude. This is due to the limited charge storing capacity of the capacitors and inductors used, mainly in the Colpitt tank part. Also, the output frequency was very close to 2.2kHz and the same can be verified from the graph displayed. The Colpitt tank introduces the total loop phase change of 180°and the Common Emitter BJT amplifier adds another 180°of phase change, ensuring constructive interference and sustained oscillations with unity gain. Hence, satisfying the Barkhausen criteria.

Conclusion

Hence we had successfully simulated Colpitt Oscillator for 2.2KHz ($\frac{99}{45} = 2.2kHz$).

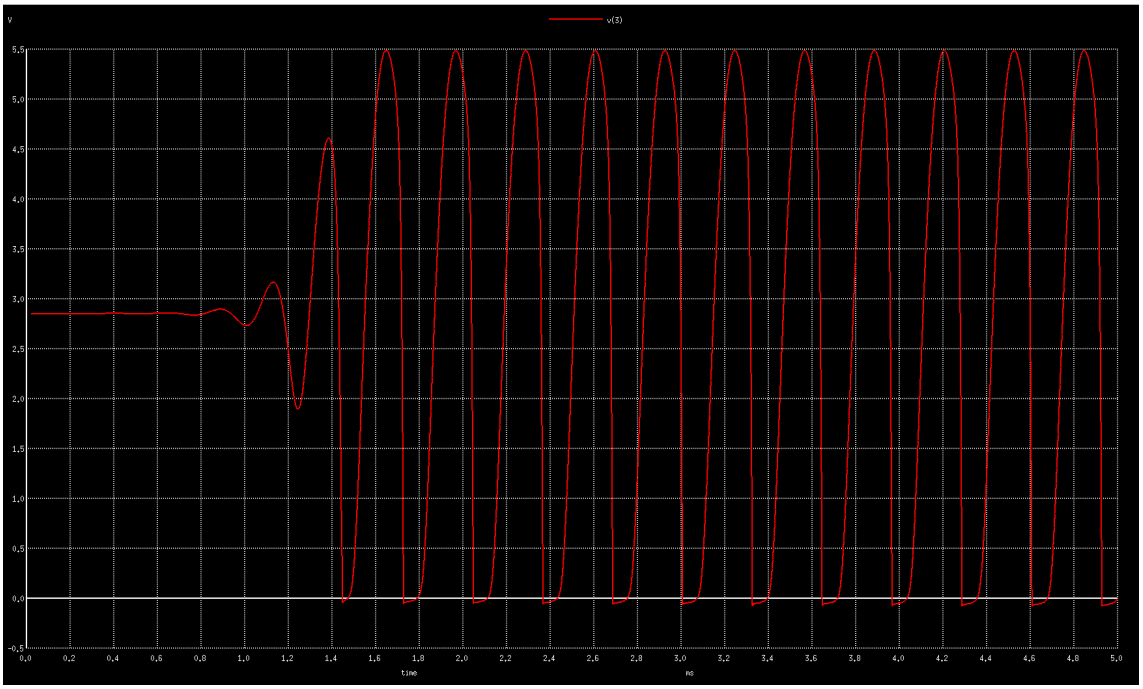


Figure 23: Output Waveform of Colpitt Oscillator.

Assignment Q2:

Aim

Design and simulate Hartley Oscillator using NGspice.

Calculation

In Hartley oscillator, the output frequency is given by the formula:

$$f_{out} = \frac{1}{2\pi\sqrt{L_T C}}$$

Where $L_T = L_1 + L_2$. And C is the capacitor used in parallel with the inductors in the Hartley tank. Roll number is 99, hence the required frequency is $\frac{99}{45} = 2.2kHz$. Fixing the value of L_T to 60mH, the value of the capacitance is given by:

$$C = \frac{1}{4\pi^2 f_{out}^2 L_T} = \frac{1}{4\pi^2 \times 2200^2 \times 60 \times 10^{-3}} = 87.22nF$$

Code

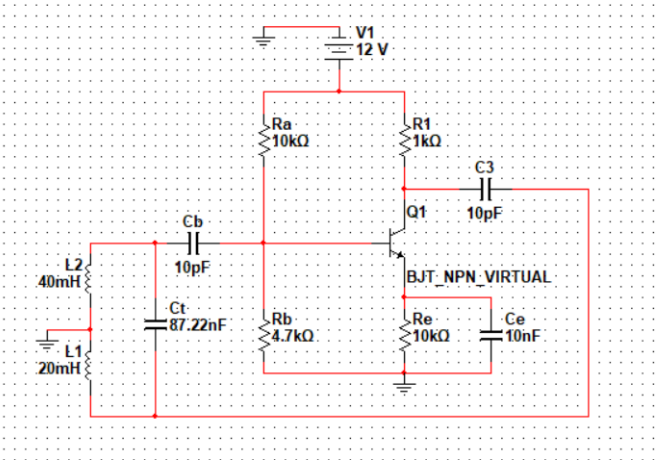


Figure 24: Hartley Oscillator.

```
* Hartley oscillator at 2.2 kHz
.MODEL NPN_Shru npn is=1e-15

Vb   1 0 DC 12

Q1   nc nb ne NPN_Shru

Rc   1 nc     1K
Re   ne 0     10K
Ce   ne 0     10n
Ra   1 nb     10K
Rb   nb 0     4.7K
Cb   2 nb     10p

Co   nc out   10p
L1   out 0    20m
Ct   out 2    87.22n
L2   2 0      40m

.tran 1u 100u 20u *Transient response in steps of 1uS, from 20uS ...
    to 100uS
.control
run
plot v(out)
.endc
.end
```

Output

For developing & simulating the circuits, Ngspice is used.

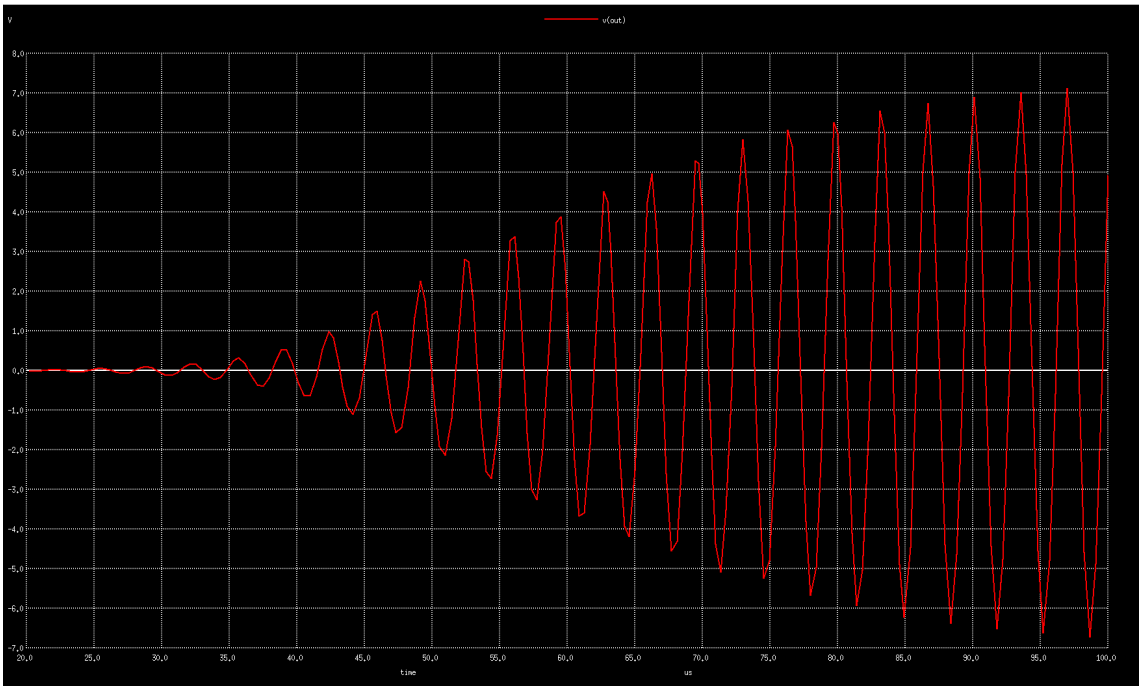


Figure 25: Output Waveform of Hartley Oscillator.

Observations

The output waveform of the oscillator requires certain time to reach proper amplitude. This is due to the limited charge storing capacity of the capacitors and inductors used, mainly in the Hartley tank part. Also, the output frequency was very close to 2.2kHz and the same can be verified from the graph displayed. The Hartley tank introduces the total loop phase change of 180°and the Common Emitter BJT amplifier adds another 180°of phase change, ensuring constructive interference and sustained oscillations with unity gain. Hence, satisfying the Barkhausen criteria.

Conclusion

Hence we had successfully simulated Hartley Oscillator for 2.2KHz ($\frac{99}{45} = 2.2kHz$).

Assignment 4: OPAMP

Differentiator:

Aim

Simulate the Differentiator using the NGSPICE
(a) Sine wave as input
(b) Square wave as input

Calculation

In a differentiator, the lower and higher cutoff frequencies are given using the below formulae:

$$f_{lower} = \frac{1}{2\pi R_f C_{in}}$$
$$f_{higher} = \frac{1}{2\pi R_i C_f}$$

Where R_i, C_{in} are the input resistance and capacitance connected in series. And, R_f, Cf are the feedback resistance and capacitance connected in parallel. Hence, for roll number 99, the lower cutoff frequency is determined as $\frac{99}{45} = 2.2KHz$. Since a bandwidth of 5KHz is required, the higher cutoff frequency is $2.2 + 5 = 7.2kHz$. Fixing both $C_{in} = C_f = 10nF$,

$$R_f = \frac{1}{2\pi f_{lower} C_{in}} = \frac{1}{2\pi 2200 \times 10^{-8}} = 7.23431K\Omega$$
$$R_i = \frac{1}{2\pi f_{higher} C_f} = \frac{1}{2\pi 7200 \times 10^{-8}} = 2.2105K\Omega$$

The output voltage is given by

$$V_{out} = -R_f C_1 \frac{dv_{in}}{dt}$$

Circuit Diagram

The circuit diagram is shown below.

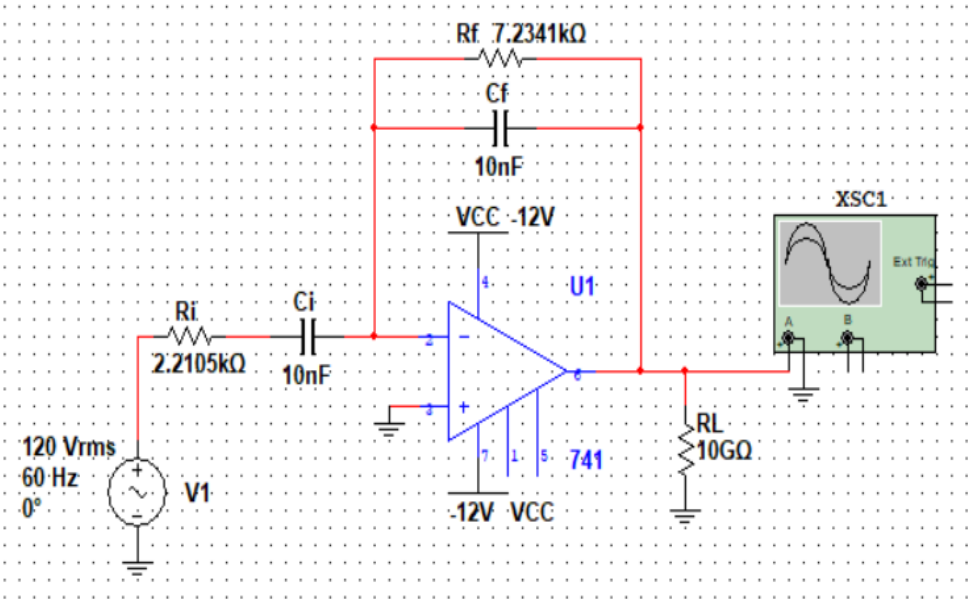


Figure 26: Differentiator using Op-Amp

Code

```
1  *differentiator using op-amp
2
3  vin 1 0 sin(0 5 2.2K)
4  *Vin 1 0 PULSE(0 5 0 0 0 05m 10m)
5  *Uncomment any one of the above two lines to use sine or square ...
       input
6
7  vdc1 4 0 12
8  vdc2 5 0 -12
9
10 r1 1 2 5K
11 c1 2 3 10n
12
13 r2 3 6 1K
14 c2 3 6 10p
15
16 .include LF356.MOD
17 XU1 0 3 4 5 6 LF356/NS
18
19 *.tran 1us 10ms 8ms *for sine
20 .trans 1us 40ms *for pulse
21 .options savecurrents
22 .control
```

```
23 run
24 plot v(1) v(6)
25 .endc
26 .end
```

Output

The input output waveforms are plotted below.

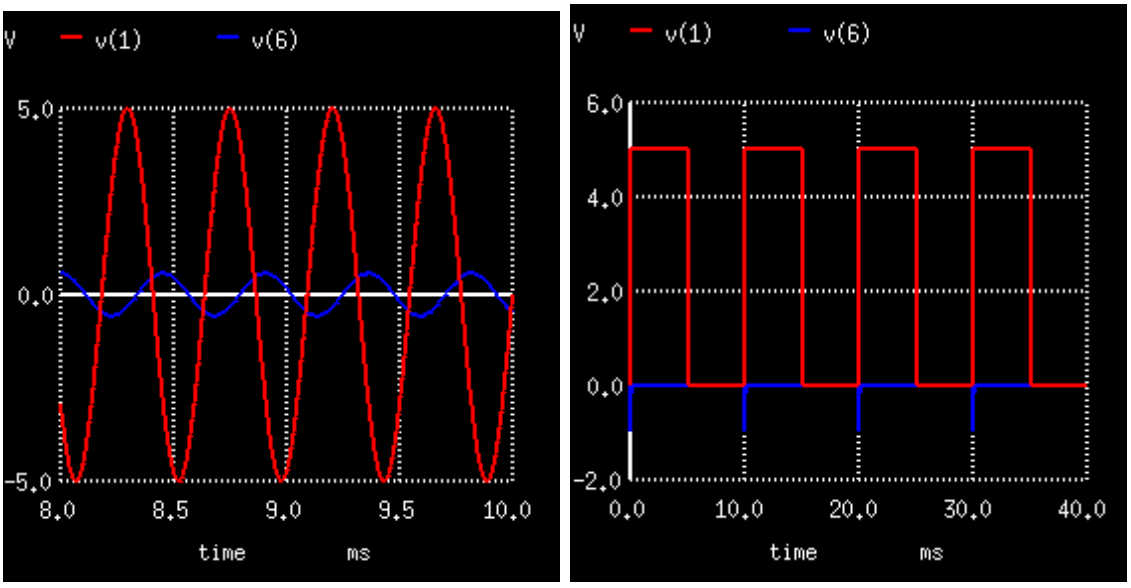


Figure 27: Input and Output waveforms of Differentiator

Observations and Conclusion

The output waveform of the differentiator is 180° out of phase with the differentiation of the input waveform due to the negative sign in the expression. So, as the input is of the form sine and the output is of the form of -cosine. Hence, the differentiator circuit with the required cutoff frequencies was successfully simulated.

Integrator:

Aim

Simulate the Integrator using NGSPICE
(a) Sine wave as input
(b) Square wave as input

Calculation

In an integrator, the cutoff frequency is given by

$$f_{cutoff} = \frac{1}{2\pi R_i C_f}$$

Where R_i, C_f are the input resistance and feedback capacitance respectively. Hence, for roll number 99, the lower cutoff frequency is determined as $\frac{99}{45} = 2.2KHz$. Fixing $C_f = 0.1\mu F$,

$$R_i = \frac{1}{2\pi f_{cutoff} C_f} = \frac{1}{2\pi 2200 \times 10^{-7}} = 723.431\Omega$$

The output waveform is given by:

$$V_{out} = -\frac{1}{R_i C_f} \int_0^t V_{in} dt$$

Circuit Diagram

The circuit diagram is shown below.

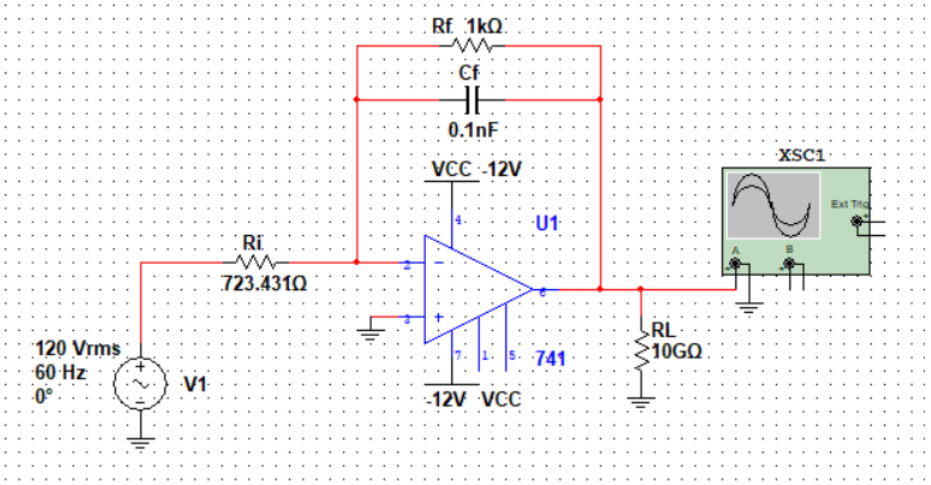


Figure 28: Integrator using Op-Amp

Code

```
1  *integrator opamp
2
3  .include LF356.MOD
4
5  vin 1 0 PULSE(-5 5 0 0 0 0.2272m 0.4545m)
6  *vin 1 0 sine(0 5 2.2K 0 0 0)
7  *Uncomment any one of the above two lines to use sine or square ...
       input
8
9  ri 1 3 723.431
10 rf 3 4 1K
11 cf 3 4 0.1u
12
13 vcc 5 0 12
14 vdd 6 0 -12
15
16 Rl 4 0 10G
17
18 XU1 0 3 5 6 4 LF356/NS
19
20 .tran 1u 10m 8m
21 .control
22 run
23 plot v(1) v(4)
24 .endc
25 .end
```

Output

The input output waveforms are plotted below.

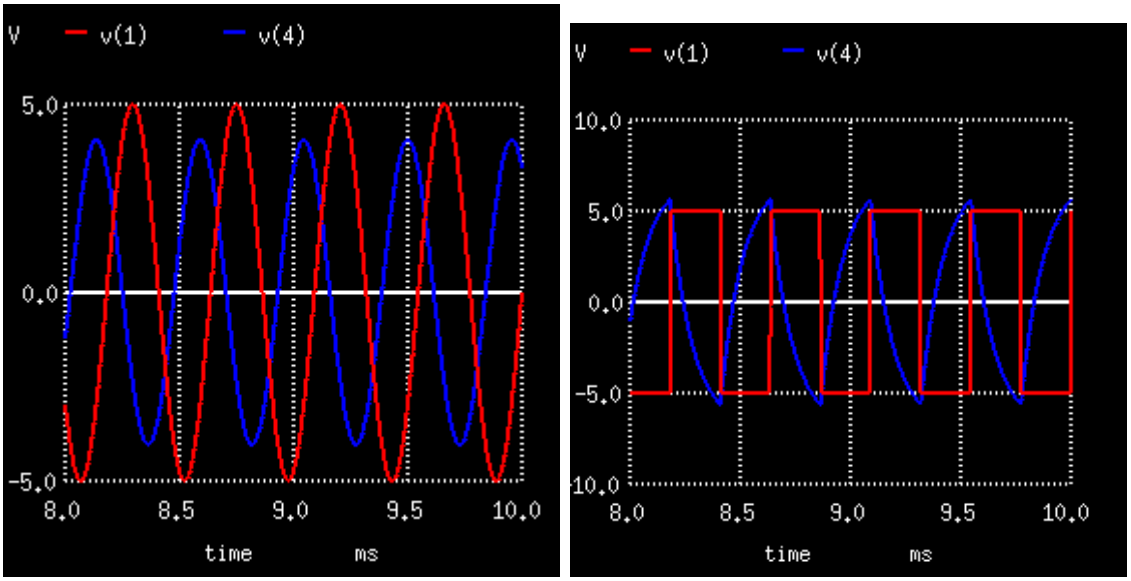


Figure 29: Input and Output waveforms of Integrator

Observations and Conclusion

The output waveform of the integrators 180° out of phase with the integration of the input waveform due to the negative sign in the expression. So, as the input is of the form sine and the output is of the form of cosine. Hence, the integrator circuit with the required cutoff frequency was successfully simulated.

Assignment Q1: Simulate the following using NGSPICE.
(a)High Pass Filter
(b)BandPass Filter
(c)BandStop Filter(Notch Filter)
Find out the cut-off frequencies for all the filters by tool and by hand and compare the values.

(a) High Pass Filter:

Calculation

In a high pass filter, the cut-in frequency is given by

$$f_{cut-in} = \frac{1}{2\pi R_1 C_1}$$

Where R_1, C_1 are the input resistance and capacitance respectively. Fixing $C_f = 10nF$ & $R_1 = 7.2K\Omega$,

$$f_{cut-in} = \frac{1}{2\pi R_1 C_1} = \frac{1}{2\pi 7.2 \times 10^3 \times 10 \times 10^{-9}} = 2.2KHz$$

Hence, lower cut-in frequency is $\frac{99}{45} = 2.2KHz$.

The circuit diagram is shown below:

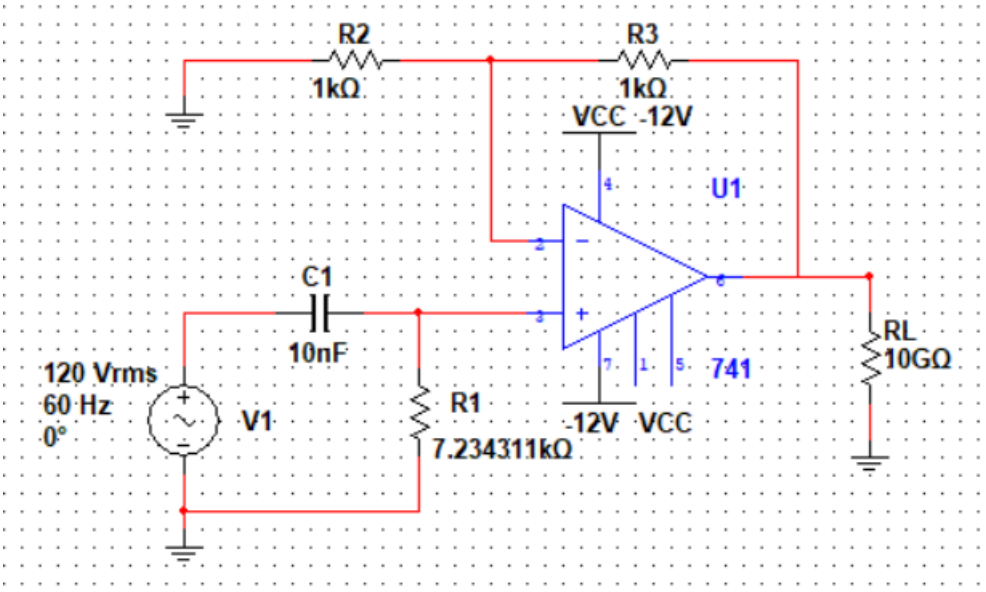


Figure 30: High Pass Filter using Op-Amp

Code

```
1  *high pass filter opamp
2  .include LF356.MOD
3
4  V1   1 0      DC 0 AC 1
5  C1   1 pi     10n
6  R1   pi 0     7.23431K
7
8  R2   ni 0     1K
9  R3   ni out   1K
10
11 RL   out 0    10G
12
13 Vcc vp 0     DC 12
14 Vee vn 0     DC -12
15
16 X1   pi ni vp vn out LF356/NS
17
18 .ac dec 100 1 1Meg * AC sweep from 100Hz to 1MHz in steps of 1Hz
19
20 .control
21 run
22 plot db(v(out))
23 .endc
24 .end
```

Output

The gain bandwidth plot is shown below.

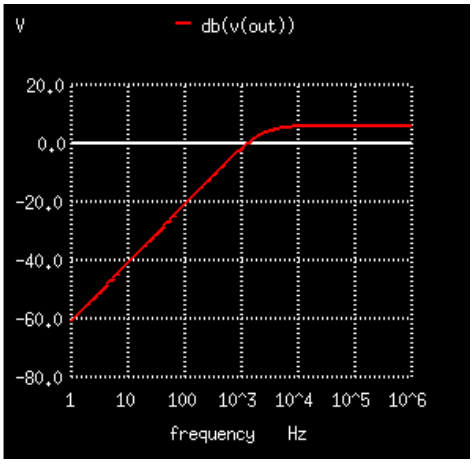


Figure 31: Gain Bandwidth Plot of High Pass Filter.

Observations and Conclusion

The high pass filter filters out all inputs with frequencies less than the cut-in frequency. In this case, it is evident from the graph and calculations that the -3dB cut-in frequency lies at 2.2KHz. Hence, the high pass filter circuit with the required cut-in frequency was successfully simulated.

(b) Band Pass Filter:

Calculation

In a band pass filter, the lower and higher cutoff frequencies are given using the below formulae:

$$f_{lower} = \frac{1}{2\pi R_1 C_1}$$
$$f_{higher} = \frac{1}{2\pi R_4 C_2}$$

Where R_1, C_1 are the first stage resistance and capacitance respectively. And, $R_4, C2$ are the second stage resistance and capacitance respectively. Fixing both $C_1 = C_2 = 10nF$ & $R_1 = 7.2K\Omega, R_2 = 2.2K\Omega$,

$$f_{lower} = \frac{1}{2\pi R_1 C_1} = \frac{1}{2\pi 7.2 \times 10^3 \times 10^{-8}} = 2.2KHz$$
$$f_{higher} = \frac{1}{2\pi R_4 C_2} = \frac{1}{2\pi 2.2 \times 10^3 \times 10^{-8}} = 7.2KHz$$

Hence, the lower cutoff frequency is $\frac{99}{45} = 2.2KHz$ with a bandwidth of 5KHz. The higher cutoff frequency is $2.2 + 5 = 7.2kHz$.

Circuit Diagram

The circuit diagram is shown below.

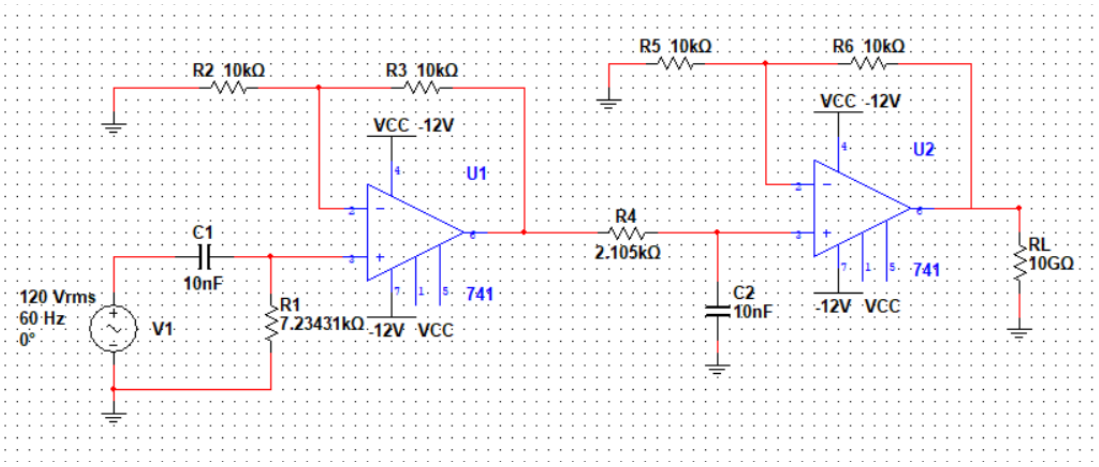


Figure 32: Band Pass Filter using Op-Amp

Code

```
1  *bandpassfilter opamp
2  .include LF356.MOD
3  V1   1 0      DC 0 AC 1
4  C1   1 p1     10n
5  R1   p1 0     7.23431K
6
7  R2   n1 0     10K
8  R3   n1 4     10K
9
10 R4   4 p2     2.2105K
11 C2   p2 0     10n
12
13 R5   n2 0     10K
14 R6   n2 out   10K
15 RL   out 0    10G
16
17 Vcc vp 0      DC 12
18 Vee vn 0      DC -12
19
20 X1   p1 n1 vp vn 4   LF356/NS
21 X2   p2 n2 vp vn out LF356/NS
22
23 .ac dec 100 1 1Meg    * AC sweep from 100Hz to 1MHz in steps of 1Hz
24 .control
25 run
26 plot db(v(out))
27 .endc
28 .end
```

Output

The gain bandwidth plot is shown below.

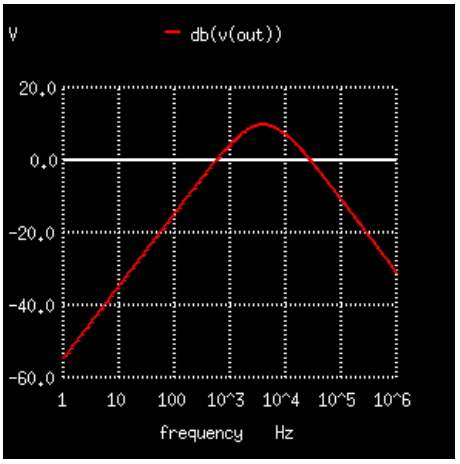


Figure 33: Gain Bandwidth Plot of Band Pass Filter.

Observations and Conclusion

The band pass filter filters out all inputs with frequencies less than the lower cutoff frequency and higher than the higher cutoff frequency. In this case, it is evident from the graph and calculations that the -3dB bandwidth is 5kHz lies from 2.2KHz to 7.2KHz. Also, due to the narrow bandwidth, the flat gain region is almost negligible in size. Hence, the band pass filter circuit with the required lower and higher cutoff frequencies.

(c) Band Reject Filter (Notch Filter):

Calculation

In a notch filter, the notch frequency is given using the given formula:

$$f_{notch} = \frac{1}{2\pi RC}$$

Where R, C are the resistance and capacitance values respectively. Assuming all resistances are equal in magnitude and similarly the capacitances are equal in magnitude. Fixing $C = 10nF$ & $R = 7.2K\Omega$,

$$f_{notch} = \frac{1}{2\pi RC} = \frac{1}{2\pi 7.2 \times 10^3 \times 10^{-8}} = 2.2KHz$$

Hence, the notch frequency is $\frac{99}{45} = 2.2KHz$.

Circuit Diagram

The circuit diagram is shown below.

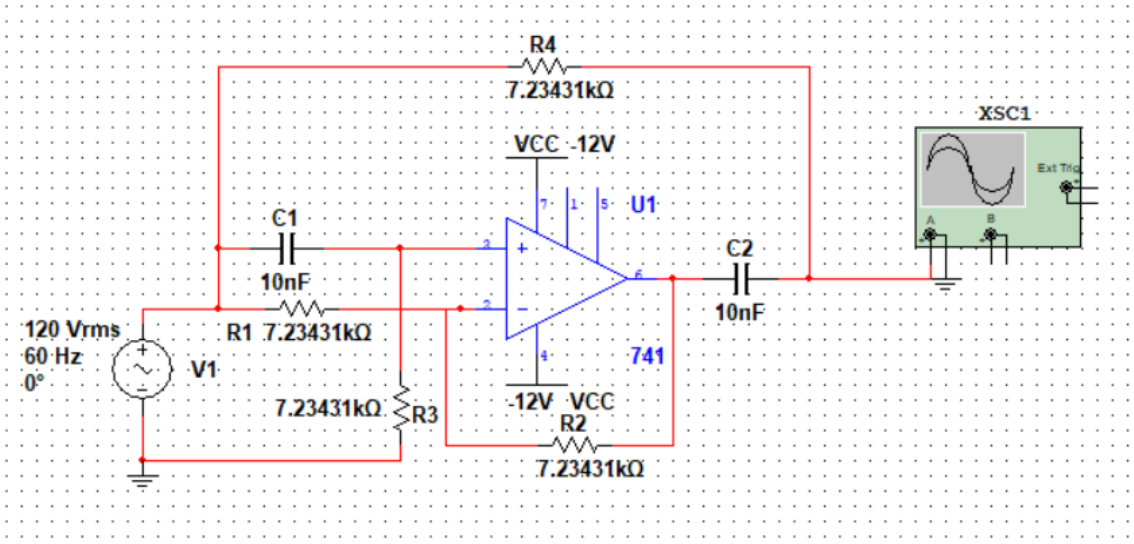


Figure 34: Band Pass Filter using Op-Amp

Code

```
*band reject filter opamp
.include LF356.MOD

V1   1 0      DC 0 AC 1
C1   1 pi     10n
R1   1 ni     7.23431K
R3   pi 0     7.23431K
R2   ni 2     7.23431K
C2   2 out    10n
R4   1 out    7.23431K
Vcc vp 0      DC 12
Vee vn 0      DC -12

X1   pi ni vp vn 2   LF356/NS
.ac dec 100 1 1Meg    *AC sweep from 100Hz to 1MHz in steps of 1Hz
.control
run
plot db(v(out))
.endc
.end
```

Output

The gain bandwidth plot is shown below.

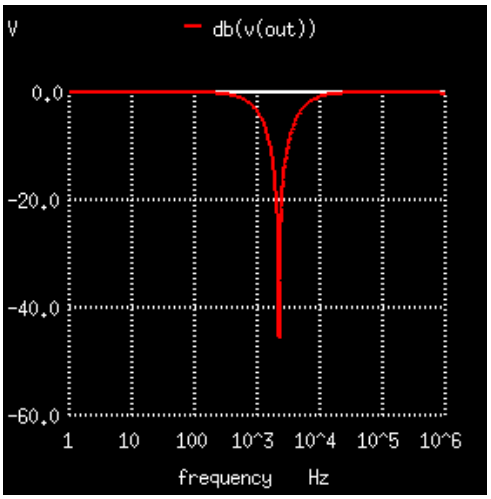


Figure 35: Gain Bandwidth Plot of Band Reject Filter.

Observations and Conclusion

The band reject filter filters inputs with frequency equal to the the notch frequency. In this case, it is evident from the graph and calculations that the filter rejects only the 2.2KHz frequency inputs. Hence, the band reject filter circuit with the required notch frequency was successfully simulated.

Assignment Q2: Simulate the following using NGSPICE
(a) Inverting Summer.
(b) Non-Inverting Summer.

(a) Inverting Summer:

Calculation

In an inverting summer, the output voltage is given by:

$$V_{out} = -\sum_i V_i$$

Where V_i are the input voltages. When all resistances are equal in magnitude.

Circuit Diagram

The circuit diagram is shown below.

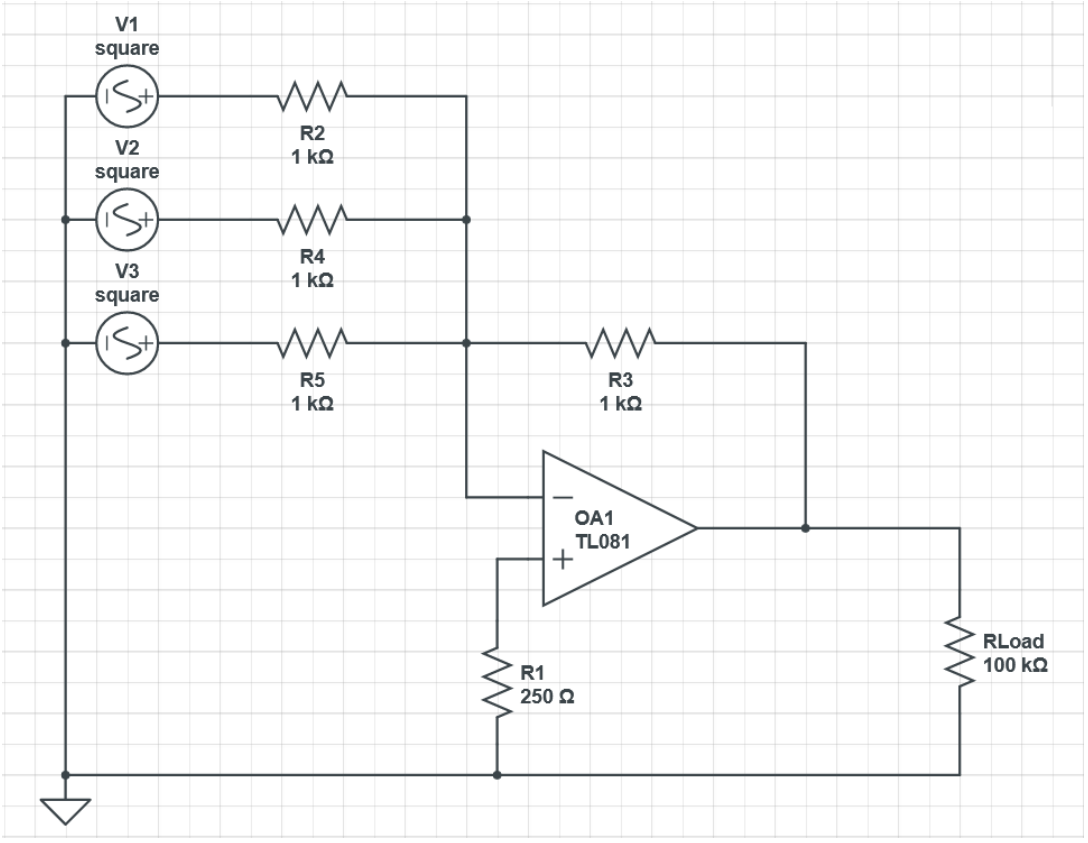


Figure 36: Inverting Summer using Op-Amp

Code

```
1  *inverting summer opamp
2  .include LF356.MOD
3
4  Va 1 0 PULSE(0 5 0 0 0 10m 20m)
5  Vb 2 0 PULSE(0 5 0 0 0 20m 40m)
6  Vc 3 0 PULSE(0 5 0 0 0 40m 80m)
7
8  r1 1 4 1K
9  r2 2 4 1K
10 r3 3 4 1K
11 r4 7 0 250
12 r5 4 8 1K
13
14 vcc 5 0 12
15 vdd 6 0 -12
16
17 XU1 7 4 5 6 8 LF356/NS
18
19 .tran 10u 80m
20 .control
21 run
22 plot v(1) v(2)+10 v(3)+20 v(8)+60
23 .endc
24 .end
```

Output

The input output waveforms are plotted below.

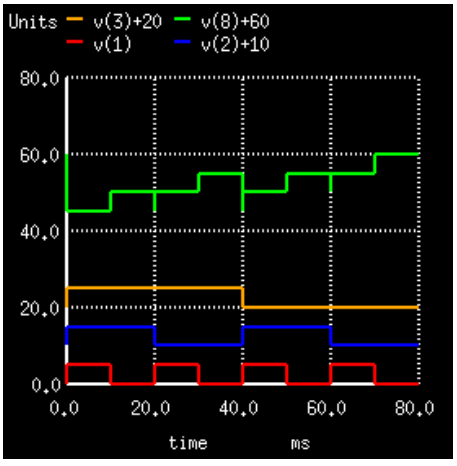


Figure 37: Input and Output waveforms of Inverting Summer

Observations and Conclusion

The output waveform is observed to be the inversion of the sum of the input waveforms. Hence the required inverting summer was successfully simulated.

(b) Non Inverting Summer:

Calculation

In a non inverting summer, the output voltage is given by:

$$V_{out} = \sum_i V_i$$

Where V_i are the input voltages. When all resistances are equal in magnitude.

Circuit Diagram

The circuit diagram is shown below.

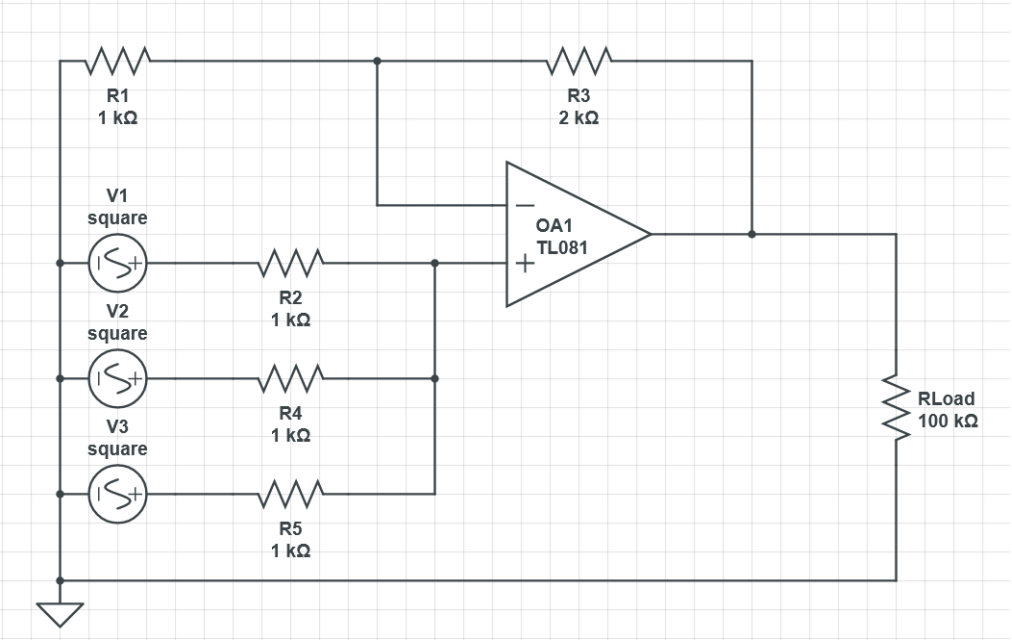


Figure 38: Non Inverting Summer using Op-Amp

Code

```
*non inverting summer opamp
.include LF356.MOD
Va 1 0 PULSE(0 5 0 0 0 10m 20m)
Vb 2 0 PULSE(0 5 0 0 0 20m 40m)
Vc 3 0 PULSE(0 5 0 0 0 40m 80m)
r1 1 4 1K
r2 2 4 1K
r3 3 4 1K
r4 7 0 1K
r5 7 8 2K
vcc 5 0 12
vdd 6 0 -12

XU1 4 7 5 6 8 LF356/NS
.tran 10u 80m
.control
run
plot v(1) v(2)+10 v(3)+20 v(8)+40
.endc
.end
```

Output

The input output waveforms are plotted below.

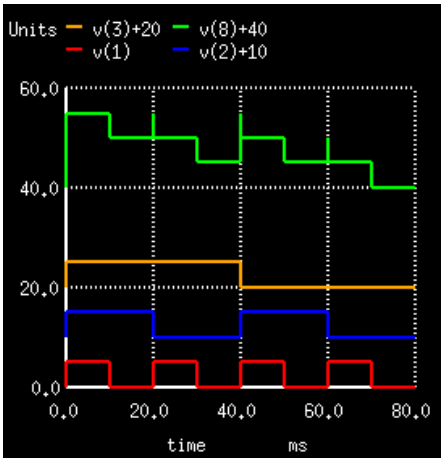


Figure 39: Input and Output waveforms of Non Inverting Summer

Observations and Conclusion

The output waveform is observed to be the sum of the input waveforms. Hence the required non inverting summer was successfully simulated.

Assignment Q3: Simulate positive and negative clipper circuit for sinusoidal input using NGSPICE.

(a) Positive Clipper:

Calculation

In a positive Clipper,

$$V_{out} = V_i \qquad , V_i < V_{ref}$$
$$V_{out} = V_{ref} \qquad , V_i > V_{ref}$$

Circuit Diagram

The circuit diagram is shown below.

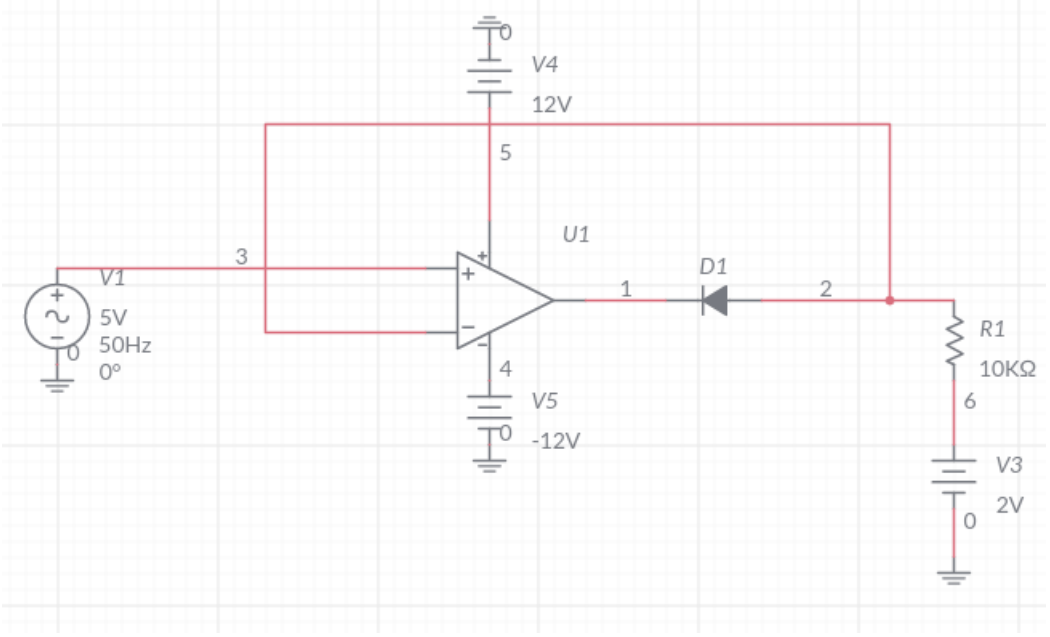


Figure 40: Positive Clipper using Op-Amp

Code

```
1 *Positive clipper OpAmp
2 .include LF356.MOD
3 Vin 1 0 sin(0 5 50)
```

```
4
5  R 2 6 10k
6  D1 2 5 D_shru
7  Vr 6 0 2v
8
9  Vp 3 0 12v
10 Vn 4 0 -12v
11 XU1 1 2 3 4 5 LF356/NS
12
13 .model D_shru D
14 .tran 0.1m 100m
15 .control
16 run
17 plot v(1) v(2)
18 .endc
19 .end
```

Output

The input output waveforms are plotted below.

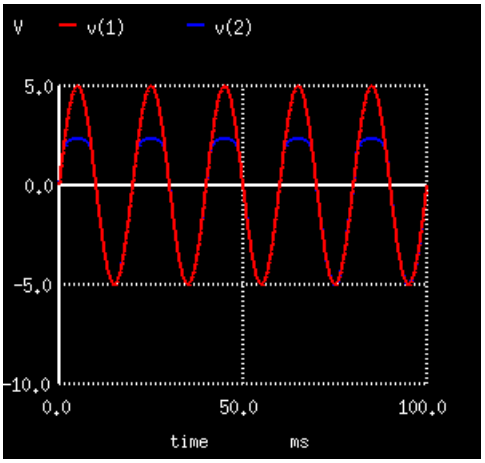


Figure 41: Input and Output waveforms of Positive Clipper

Conclusion

Positive clipper circuit using Op Amp and diode was simulated and the clipping operation was successfully implemented.

(b) Negative Clipper:

Calculation

In a negative Clipper,

$$V_{out} = V_i \qquad , V_i > V_{ref}$$
$$V_{out} = V_{ref} \qquad , V_i < V_{ref}$$

Circuit Diagram

The circuit diagram is shown below.

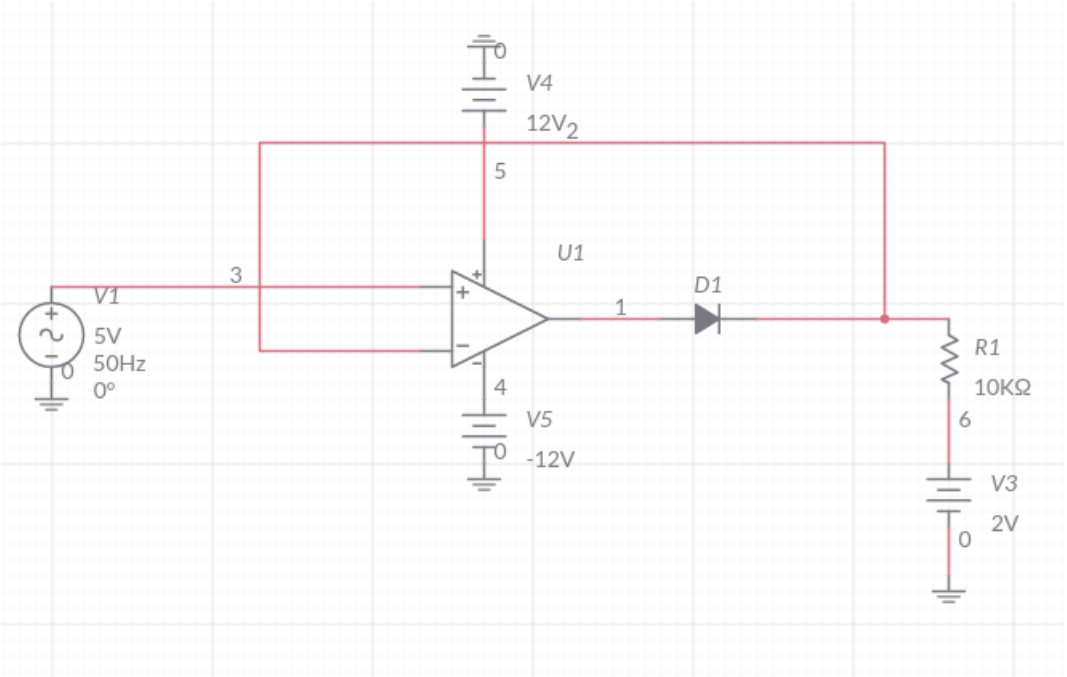


Figure 42: Positive Clipper using Op-Amp

Code

```
1 *Negative clipper OpAmp
2 .include LF356.MOD
3 Vin 1 0 sin(0 5 50)
4
5 R 2 6 10k
6 D1 5 5 D_shru
```

```
Vr 6 0 -2v

Vp 3 0 12v
Vn 4 0 -12v
XU1 1 2 3 4 5 LF356/NS

.model D_shru D
.tran 0.1m 100m
.control
run
plot v(1) v(2)
.endc
.end
```

Output

The input output waveforms are plotted below.

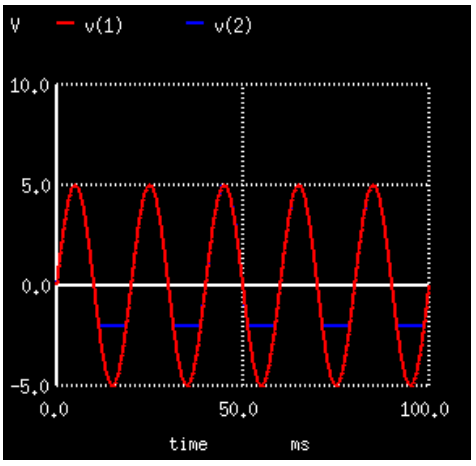


Figure 43: Input and Output waveforms of Negative Clipper

Conclusion

Negative clipper circuit using Op Amp and diode was simulated and the clipping operation was successfully implemented.

Assignment 5: FET

Inverter Circuit:

Aim

To simulate inverter circuit using MOSFETs.

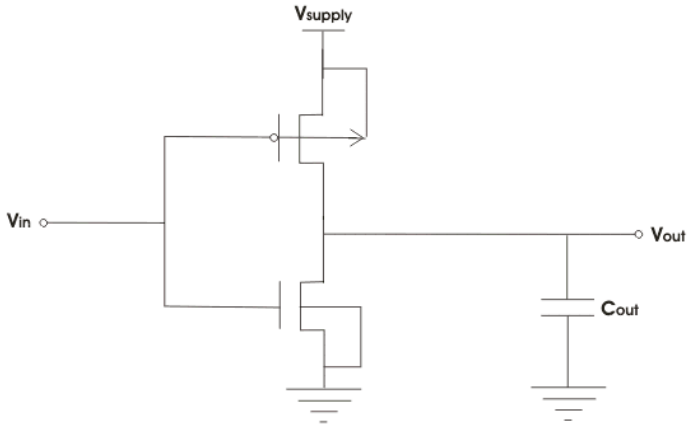


Figure 44: Inverter using MOSFETs.

Code

```
1  *CMOS inverter
2  vin 1 0 pulse(0v 5v 0 1ns 1ns 40ns 80ns)
3  vdd 3 0 dc 5v
4
5  *MX nd ng ns nb mname
6  m1 3 1 2 3 mos_shru l=1u w=100u
7  .model mos_shru pmos vto=-1v kp=80u
8  m2 2 1 0 0 mos_Shru l=1u w=40u
9  .model mos_Shru nmos vto=1v kp=200u
10
11 .tran 10ns 200ns
12 .control
13 run
14 plot v(1) v(2)+10 title 'CMOS inverter' xlabel 'time'
15 .endc
16 .end
```

Output

The input output waveforms are plotted below.

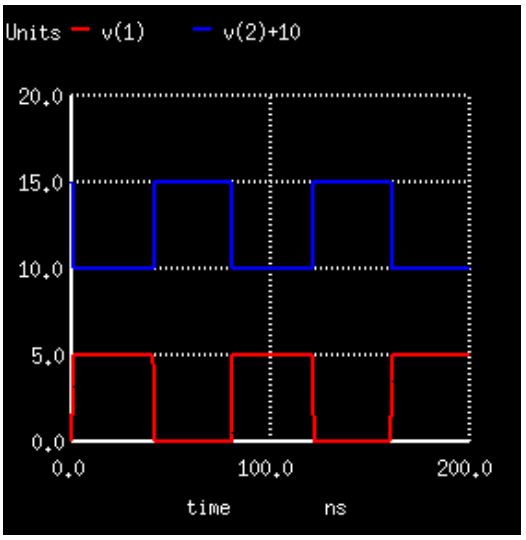


Figure 45: Input and Output waveforms of Inverter.

Observations & Conclusion

CMOS Inverter diagram consists of both nmos-pmos pair sharing common gate. It inverts the input signal as shown in Fig 45.

Assignment Q1: Simulate Common-Drain Amplifier Using JFET.
(a) Find out the Gain.
(b) Find out the MidBand Gain Using AC Analysis.

Circuit Diagram

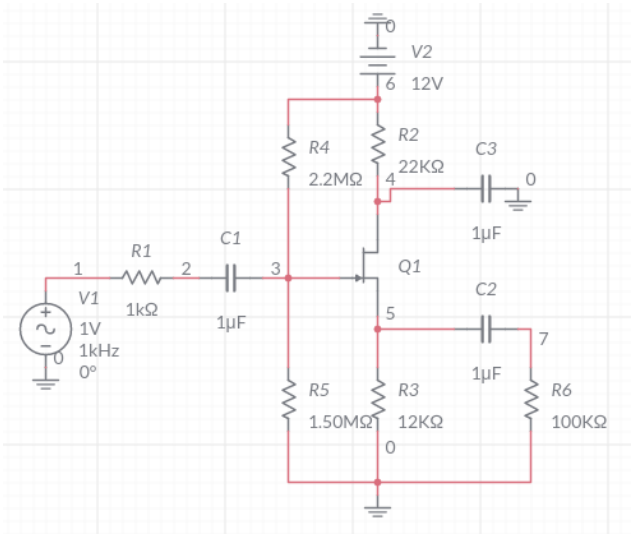


Figure 46: Common-Drain Amplifier using N-JFET.

Code

```
*common drain amplifier

*v1 1 0 sine(0 4 500)
v1 1 0 ac=1 dc=1
r1 1 2 1k
c1 2 3 1u

*JX nd ng ns mname
J1 4 3 5 fet_shru
.model fet_shru NJF

rd 4 6 22k
vcc 6 0 12v
r2 6 3 2.2mega
r3 3 0 1.5mega
r4 5 0 12k
c3 5 7 1u
r5 7 0 100k
```

```
19 c4 4 0 1u
20
21 *.tran 1u 10ms
22 * unmcomment above one for transient analysis
23 * uncomment below one for ac analysis
24 .ac dec 100 1 100k
25
26 .control
27 run
28
29 * plot v(1) v(7)
30 * unmcomment above one for ip op plot
31 * uncomment below one for gain plot
32 plot db(v(7))
33
34 .endc
35 .end
```

Output

The input output waveforms are plotted below.

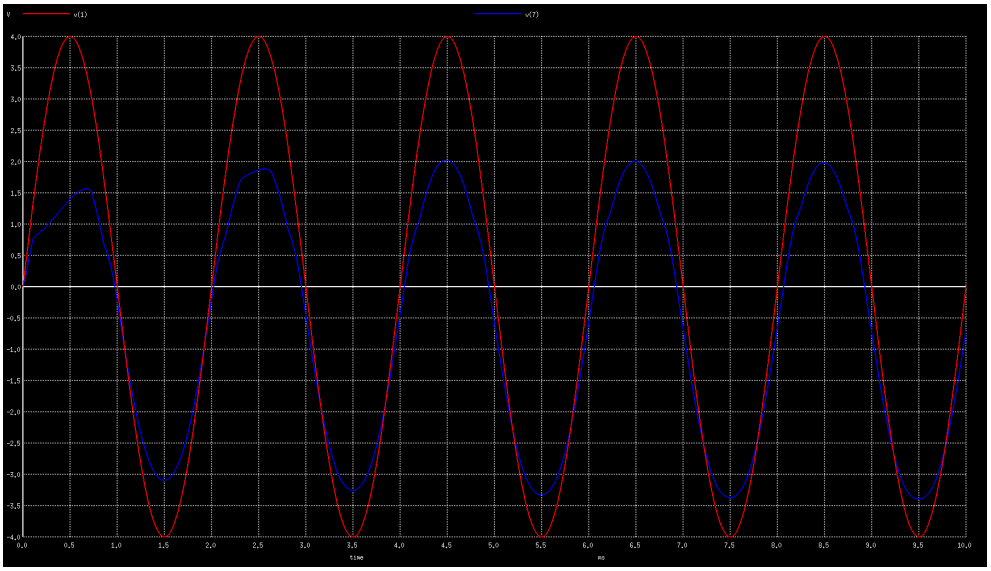
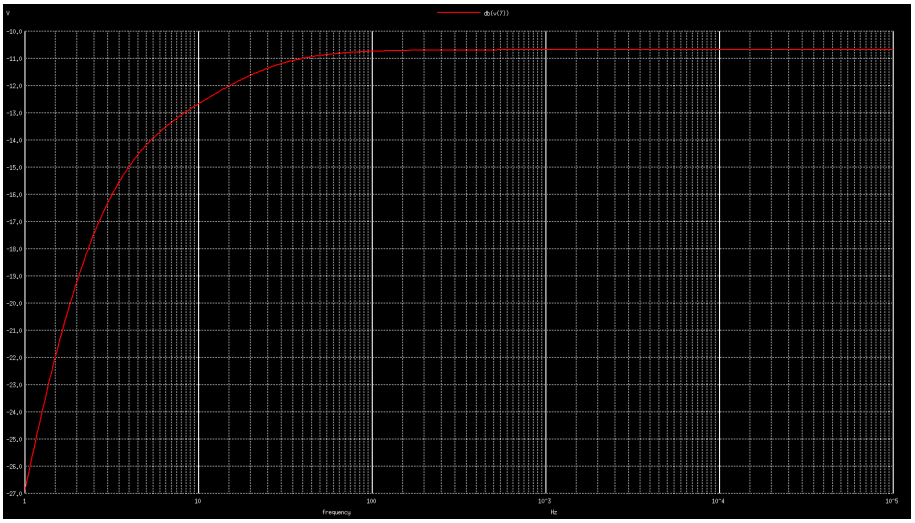


Figure 47: Ip & Op waveforms of Common-Drain Amplifier using N-JFET.

The gain plot is plotted below.



(a) Gain plot.

x0 = 1378.91, y0 = -10.4255

(b) MidBand Gain.

Figure 48: Common-Drain Amplifier using N-JFET.

Observations & Conclusion

- Voltage Gain = $V_o/V_{in} = 2/4 = 0.5 < 1$
- MidBand Gain - The Midband Gain of a transistor is the transistor's gain at its mid frequencies. The midband gain is where the transistor's gain is at the highest and most constant level in its bandwidth. From Fig 48.(b)
  MidBand Gain = -10.42 dB

Assignment Q2:

Aim

To simulate transfer characteristics of n-type MOSFET.

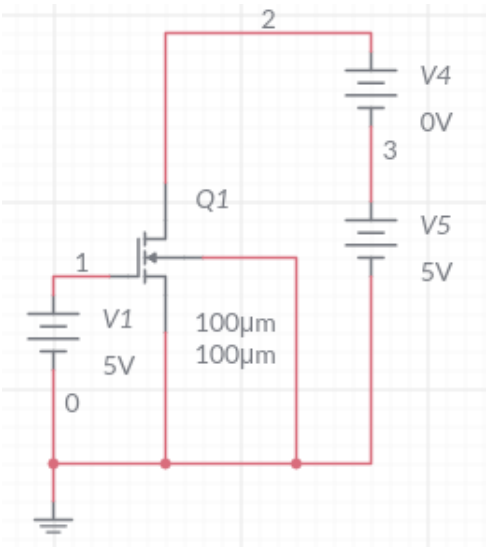


Figure 49: N-type MOSFET.

Code

```
1  *nmos tf char
2
3  vdd 3 0 dc 5v
4  vd 3 2 dc 0v
5  vgs 1 0 dc 5v
6
7  *MXX nd ng ns nb mname
8  MN 2 1 0 0 Mos_Shru
9  .model Mos_Shru NMOS
10
11 .dc vgs 0 10 1m
12 .control
13 run
14 plot i(vd) vs v(1) ylabel 'Ids' xlabel 'Vgs' title 'Transfer ...
       Characteristics'
15 .endc
16 .end
```

Output

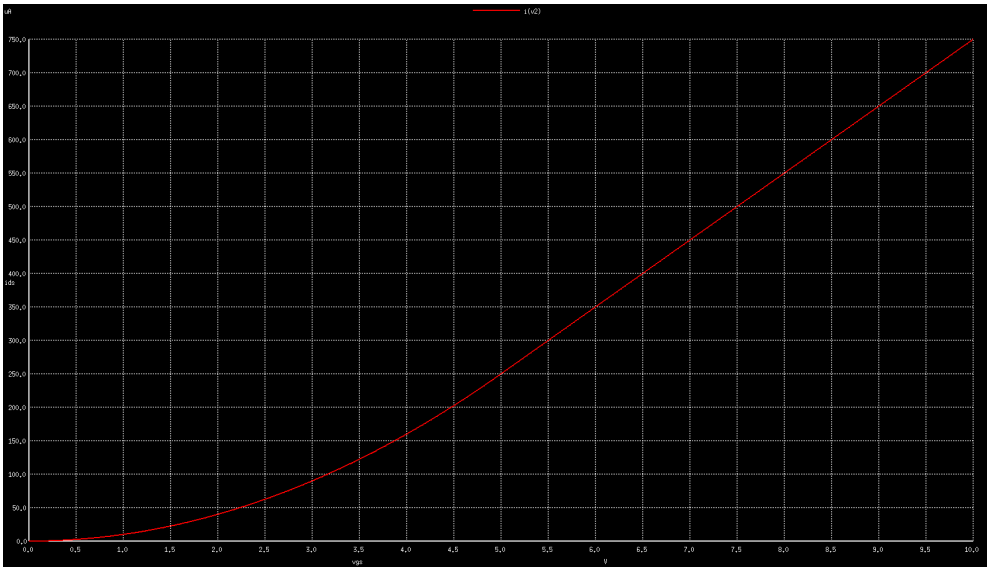


Figure 50: Transfer Characteristics of NMOS.

Observations & Conclusion

The transfer characteristic relates drain current I_D response to the input gate-source driving voltage V_{GS}. As V_{GS} increases for the nMOS transistor in Fig 50, the threshold voltage is reached where drain current elevates.

Assignment Q3:

Aim

To simulate drain and transfer characteristics of p-type MOSFET.

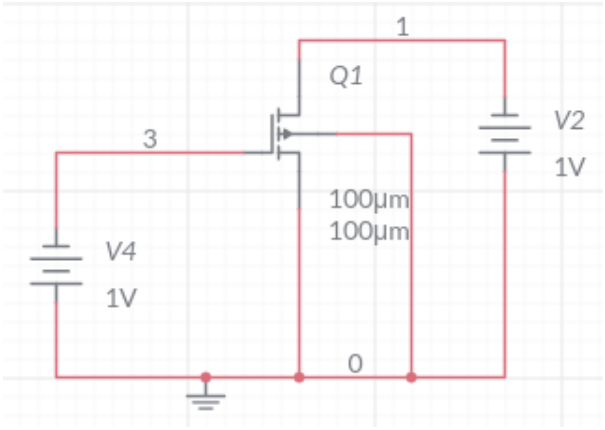


Figure 51: P-type MOSFET.

Code

```
*pmos drain and tf char
vgs 0 3 dc 1
M1 1 3 0 0 Mos_shru
vdd 4 1 dc 0
* uncomment above for drain char
* uncomment below  for transfer char
*vdd 1 4 dc 0
vds 0 4 dc 1
.model Mos_shru pmos

.dc vds 0 15 1m vgs 2 8 1
* uncomment above for drain char
* uncomment below  for transfer char
*.dc vgs 0 5 1

.control
run
plot i(vdd) ylabel 'Ids' xlabel 'Vds' title 'Drain Characteristics'
*plot i(vdd) vs v(3) ylabel 'Ids' xlabel 'Vgs' title 'Transfer ...
    Characteristics'
.endc
.end
```

Output

Drain Characteristics of PMOS is plotted below

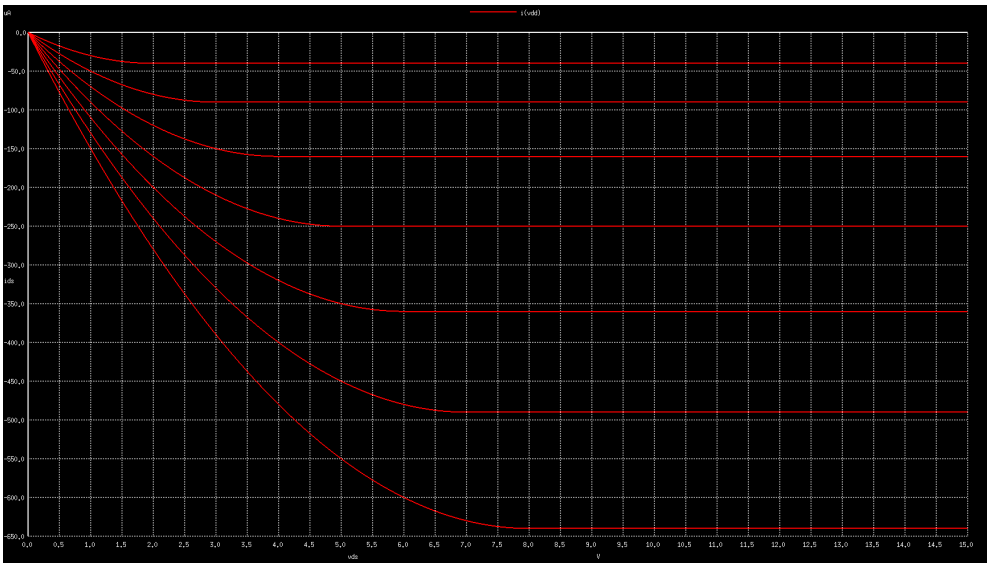


Figure 52: Drain Characteristics of pmos.

Transfer Characteristics of PMOS is plotted below

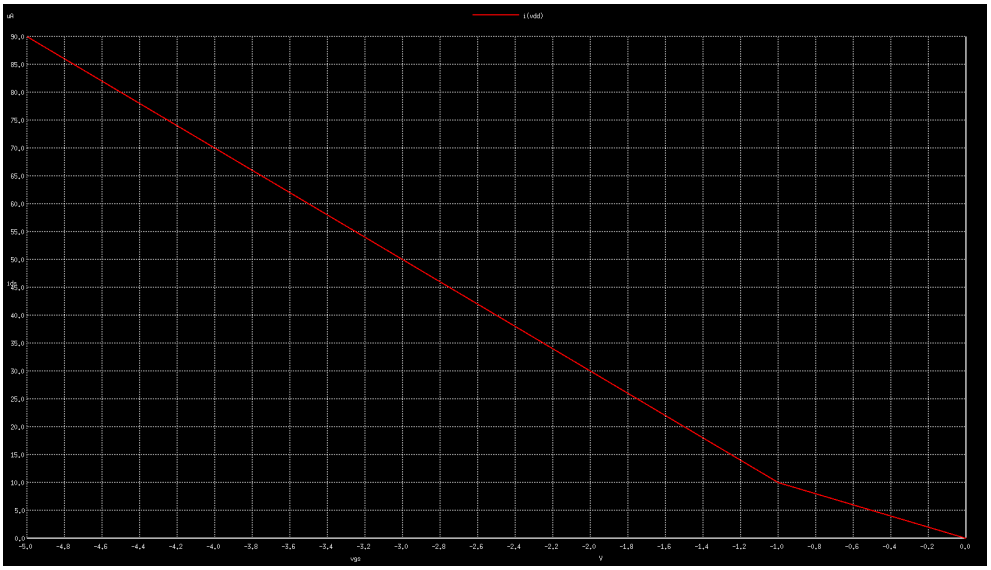


Figure 53: Transfer Characteristics of pmos.

Observations & Conclusion

It is observed that pMOS transistor input characteristic is analogous to the nMOS transistor except the ID and VGS polarities are reversed.

Assignment Q4: Simulate Common-Gate Amplifier using MOSFET.
(a) Find out the Gain.
(b) Find out the MidBand Gain Using AC Analysis.

Circuit Diagram

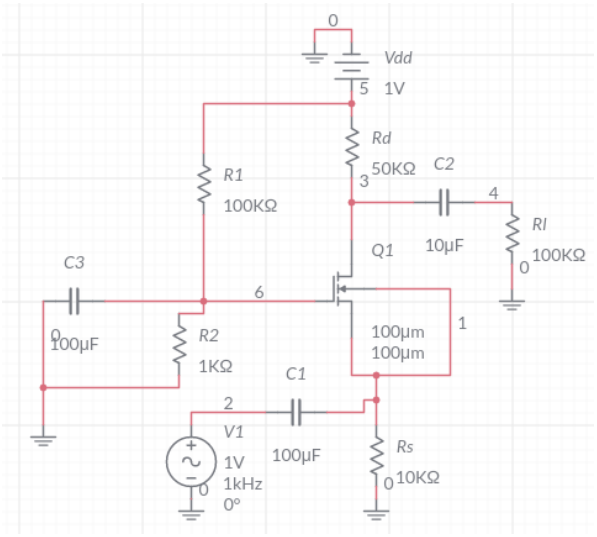


Figure 54: Common-Gate Amplifier using NMOS.

Code

```
1  * common gate amplifier MOSFET
2
3  Vin 1 0 sine(0 4 350)
4  *Vin 1 0 dc=0 ac=5
5  C1 1 s1 100u
6  Rs s1 0 10K
7
8  Vdd 2 0 DC 1V
9  Rd 2 d1 50k
10 C2 d1 op 100u
11 Rl op 0 100k
12
13 R1 2 g 100K
14 R2 g 0 1K
15 Cg g 0 100u
16
17 *MX nd ng ns nb mname
18 M1 d1 g s1 s1 Mos_Shru
```

```
19 .model Mos_Shru NMOS
20
21 .tran 1u 10m
22 * unmcomment above one for transient analysis
23 * uncomment below one for ac analysis
24 *.ac dec 100 1 1Mega
25
26 .control
27 run
28
29 plot v(1) v(op)
30 * unmcomment above one for ip op plot
31 * uncomment below one for gain plot
32 *plot db(v(op))
33
34 .endc
35 .end
```

Output

The input output waveforms are plotted below.

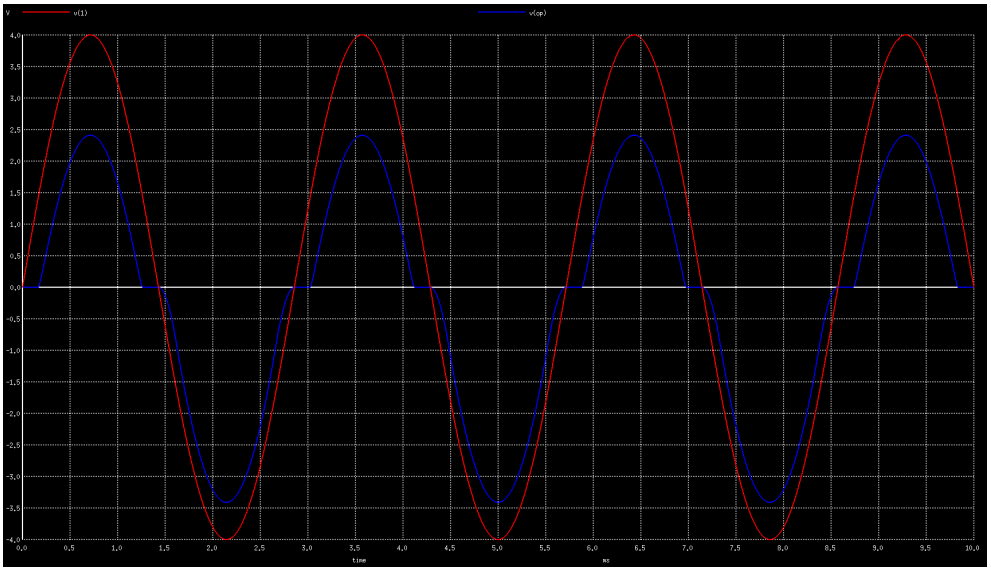
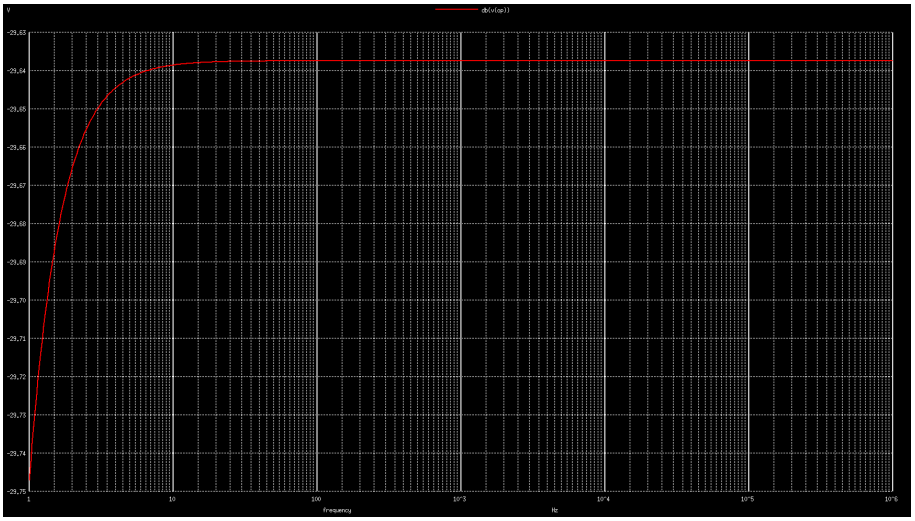


Figure 55: Ip & Op waveforms of Common-Gate Amplifier using NMOS.

The gain plot is plotted below.



(a) Gain plot.

x0 = 13219.4, y0 = -29.6347

(b) MidBand Gain.

Figure 56: Common-Gate Amplifier using NMOS.

Observations & Conclusion

- Voltage Gain = $V_o/V_{in} = 2.95/4 = 0.73 < 1$
- MidBand Gain - The Midband Gain of a transistor is the transistor's gain at its mid frequencies. The midband gain is where the transistor's gain is at the highest and most constant level in its bandwidth. From Fig 56.(b)
  MidBand Gain = -29.63 dB

Assignment 6: GATES

Assignment Q1: Simulate 2-i/p XOR Gate with minimum number of 2-i/p NAND Gates.

Circuit Diagram

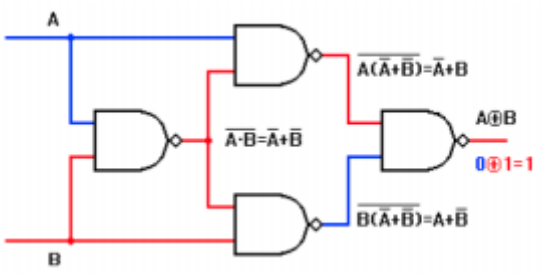


Figure 57: Circuit diagram

Code

```
*XOR using nand

va a 0 PULSE(0 9.9 0 1u 1u 10m 15m)
vb b 0 PULSE(9.9 0 0 1u 1u 20m 35m)

.subckt cmos_2nand in1 in2 res
* name port1 port2  ...
    .model MOSN NMOS level=8 version=3.3.0
    .model MOSP PMOS level=8 version=3.3.0
    * transistor models

    vdd d 0 dc 10
    *biasing voltage

    r1 in1 1 1k
    r2 in2 2 1k


    r4 10 res 1k
    * resistors

    m1 10 1 d d MOSP
    m2 10 2 d d MOSP


```

```
26
27      m4 10 2 11 0 MOSN
28      m5 11 1 0 0 MOSN
29
30      * mn drain gate source bulk model
31      * nmos bulk to ground, pmos bulk to vdd
32  .ends cmos_2nand
33
34  x1 a b c1 cmos_2nand
35  x2 a c1 c2 cmos_2nand
36  x3 b c1 c3 cmos_2nand
37  x4 c2 c3 out cmos_2nand
38
39
40
41  .tran 10u 50m
42  .control
43  run
44
45  set color0 = white
46  set color1 = black
47  set color2 = blue
48  set color3 = green
49  set color4 = red
50  set color5 = violet
51
52  plot v(a) v(b)+20 v(out)+40
53
54
55  .endc
56  .end
```

Output

The input output waveforms are plotted below.

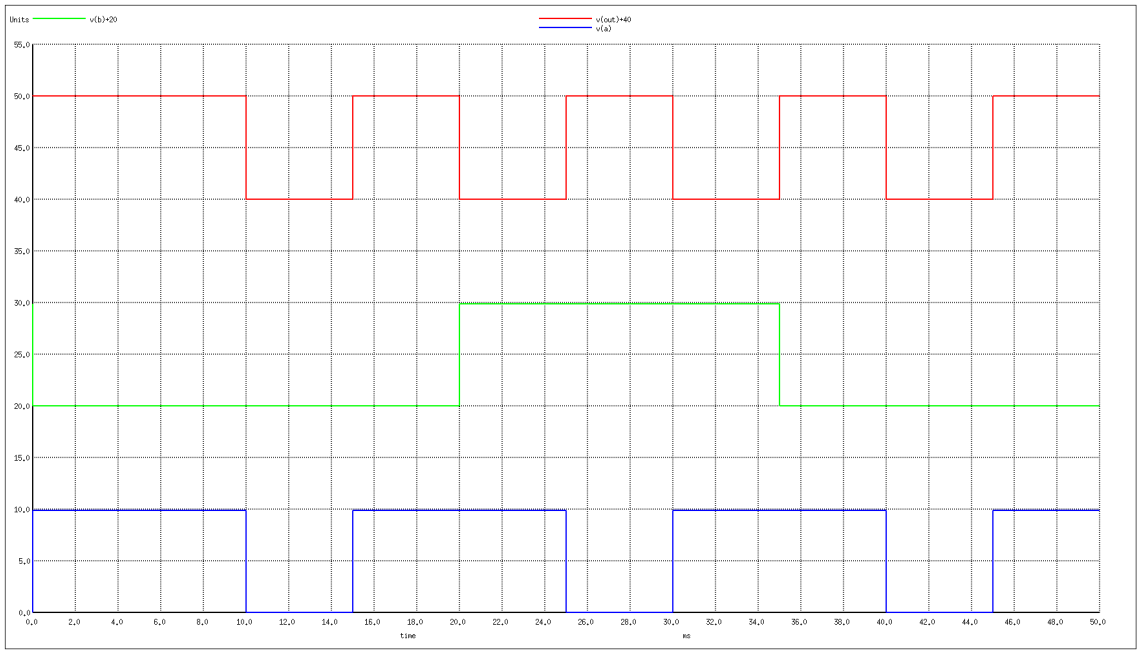


Figure 58: Ip & Op waveforms of XOR.

Observations & Conclusion

XOR gate is a digital logic gate that gives a true output when the number of true inputs is odd. An XOR gate implements an exclusive or; that is, a true output results if only one of the inputs to the gate is true. If both inputs are false or both are true, a false output results. Number of 2-input NAND gates required to implement a 2-input XOR gate is 4.

Assignment Q2: Simulate 3-i/p AND Gate with minimum number of 2-i/p NAND Gates

Circuit Diagram

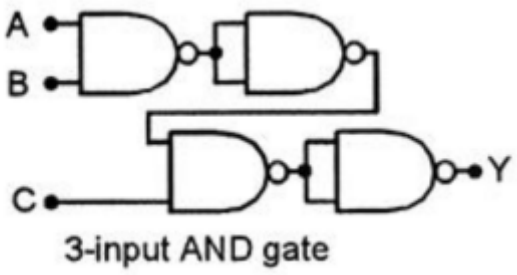


Figure 59: Circuit diagram

Code

```
1  *3ip and using 2ip nand
2
3  va a 0 PULSE(0 9.9 0 1u 1u 10m 15.1m)
4  vb b 0 PULSE(9.9 0 0 1u 1u 20m 35.2m)
5  vc c 0 PULSE(0 9.9 0 1u 1u 40m 75.3m)
6
7  .subckt cmos_2nand in1 in2 res
8  * name port1 port2  ...
9      .model MOSN NMOS level=8 version=3.3.0
10     .model MOSP PMOS level=8 version=3.3.0
11     * transistor models
12
13     vdd d 0 dc 10
14     *biasing voltage
15
16
17     r1 in1 1 1k
18     r2 in2 2 1k
19     r4 10 res 1k
20     * resistors
21
22     m1 10 1 d d MOSP
23     m2 10 2 d d MOSP
24     m4 10 2 11 0 MOSN
25     m5 11 1 0 0 MOSN
26
```

```
27      * mn drain gate source bulk model
28      * nmos bulk to ground, pmos bulk to vdd
29  .ends cmos_2nand
30
31  x1 a b c1 cmos_2nand
32  x2 c1 c1 c2 cmos_2nand
33  x3 c c2 c3 cmos_2nand
34  x4 c3 c3 out cmos_2nand
35
36  .tran 10u 160m
37  .control
38  run
39  set color0 = white
40  set color1 = black
41  set color2 = blue
42  set color3 = green
43  set color4 = red
44  set color5 = violet
45  plot v(a) v(b)+20  v(c)+40 v(out)+60
46
47  .endc
48  .end
```

Output

The input output waveforms are plotted below.

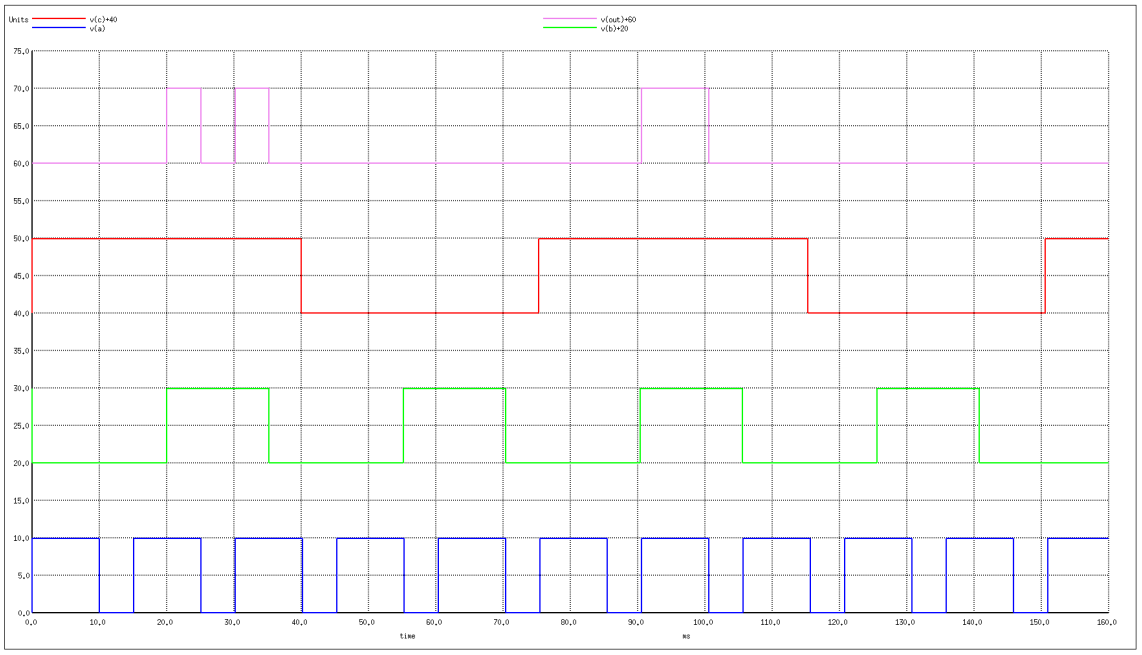


Figure 60: Ip & Op waveforms of AND.

Observations & Conclusion

The AND gate is a basic digital logic gate that implements logical conjunction. A HIGH output (1) results only if all the inputs to the AND gate are HIGH (1). If even one input is LOW, LOW output results. The function can be extended to any number of inputs. Number of 2-input NAND gates required to implement a 3-input AND gate is 4.

Assignment Q3: Simulate 3-i/p NAND Gate with minimum number of 2-i/p NAND Gates.

Circuit Diagram

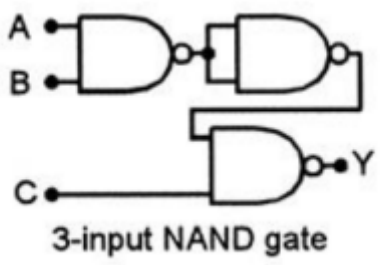


Figure 61: Circuit diagram

Code

```
*3ip and using 2ip nand

va a 0 PULSE(0 9.9 0 1u 1u 10m 15.1m)
vb b 0 PULSE(9.9 0 0 1u 1u 20m 35.2m)
vc c 0 PULSE(0 9.9 0 1u 1u 40m 75.3m)

.subckt cmos_2nand in1 in2 res
* name port1 port2  ...
    .model MOSN NMOS level=8 version=3.3.0
    .model MOSP PMOS level=8 version=3.3.0
    * transistor models

    vdd d 0 dc 10
    *biasing voltage


    r1 in1 1 1k
    r2 in2 2 1k
    r4 10 res 1k
    * resistors


    m1 10 1 d d MOSP
    m2 10 2 d d MOSP

```

```
27      m4 10 2 11 0 MOSN
28      m5 11 1 0 0 MOSN
29
30      * mn drain gate source bulk model
31      * nmos bulk to ground, pmos bulk to vdd
32  .ends cmos_2nand
33
34  x1 a b c1 cmos_2nand
35  x2 c1 c1 c2 cmos_2nand
36  x3 c c2 out cmos_2nand
37
38
39  .tran 10u 160m
40  .control
41  run
42  set color0 = white
43  set color1 = black
44  set color2 = blue
45  set color3 = green
46  set color4 = red
47  set color5 = violet
48  plot v(a) v(b)+20  v(c)+40 v(out)+60
49
50  .endc
51  .end
```

Output

The input output waveforms are plotted below.

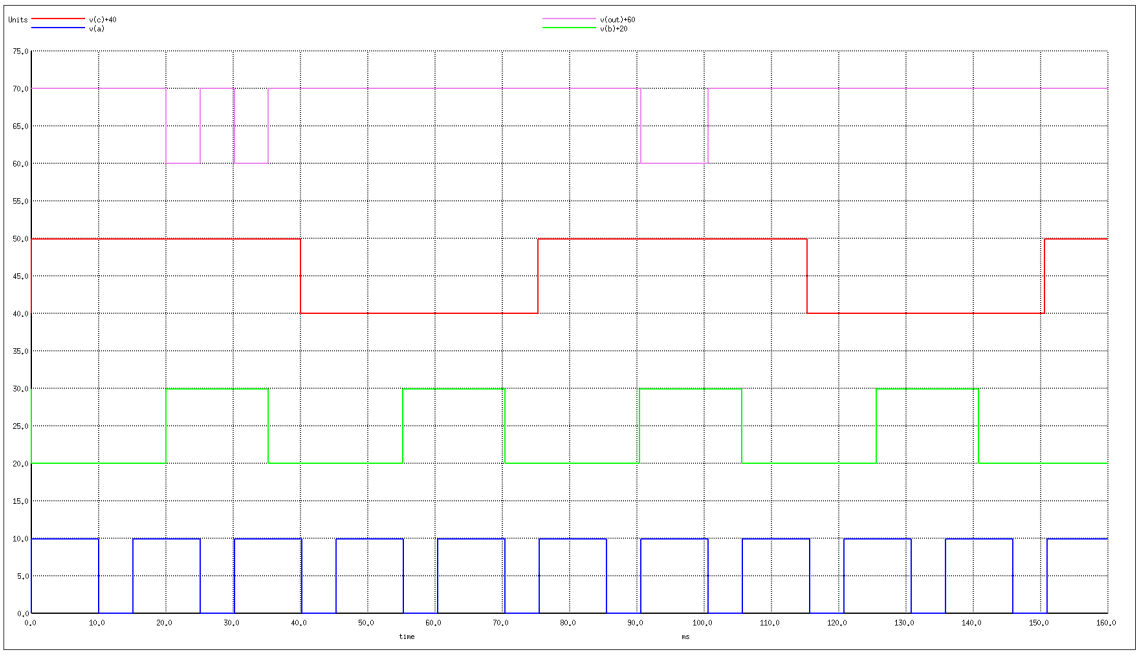


Figure 62: Ip & Op waveforms of 3-ip NAND.

Observations & Conclusion

We have successfully simulated a program of a 3-i/p NAND Gate with the minimum number of 2-i/p NAND Gates. Number of 2-input NAND gates required to implement a 3-input NAND gate is 3.

Assignment Q4: Simulate AB+C Boolean Expression using CMOS Logic.

Circuit Diagram

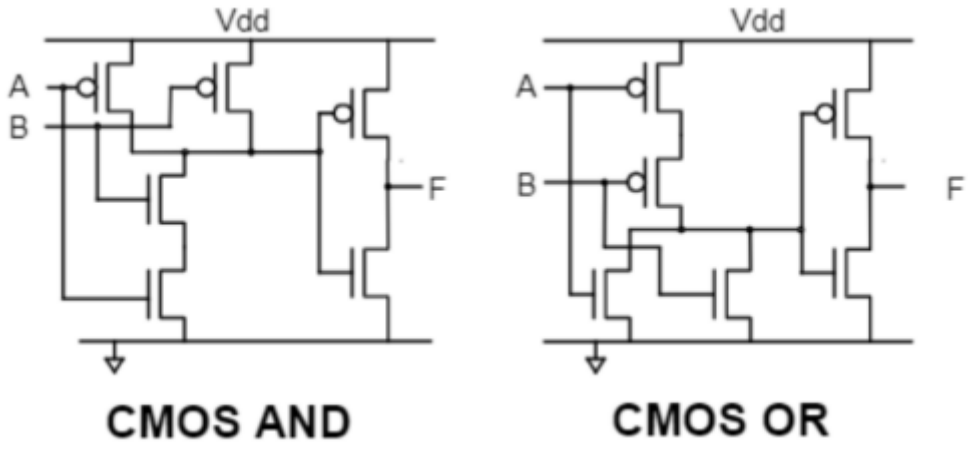


Figure 63: Circuit diagram

Code

```
1  * Logic AB+C using CMOS
2
3  va a 0 PULSE(0 9.9 0 1u 1u 10m 20m)
4  vb b 0 PULSE(9.9 0 0 1u 1u 20m 40m)
5  vc c 0 PULSE(0 9.9 0 1u 1u 40m 80m)
6
7
8  .subckt cmos_or i1 i2 res
9  * name port1 port2 port3 ...
10     .model MOSN NMOS level=8 version=3.3.0
11     .model MOSP PMOS level=8 version=3.3.0
12
13
14     vdd d 0 dc 10
15
16     r1 i1 1 1k
17     r2 i2 2 1k
18     r3 6 res 1k
19     r4 4 5 1k
20
21     m1 3 1 d d MOSP
22     m2 4 2 3 d MOSP
23     m3 6 5 d d MOSP
```

```
24
25      m4 4 1 0 0 MOSN
26      m5 6 5 0 0 MOSN
27      m6 4 2 0 0 MOSN
28  .ends cmos_or
29
30  .subckt cmos_and i1 i2 res
31  * name port1 port2 port3 ...
32      .model MOSN NMOS level=8 version=3.3.0
33      .model MOSP PMOS level=8 version=3.3.0
34
35      vdd d 0 dc 10
36
37      r1 5 res 1k
38      r2 i1 1 1k
39      r3 i2 2 1k
40      r4 3 4 1k
41
42      m1 3 1 d d MOSP
43      m2 3 2 d d MOSP
44      m3 5 4 d d MOSP
45
46      m4 3 2 6 0 MOSN
47      m5 6 1 0 0 MOSN
48      m6 5 4 0 0 MOSN
49  .ends cmos_and
50
51
52  x1 a b r1 cmos_and
53  x2 c r1 out cmos_or
54
55  .tran 10u 160m
56  .control
57  run
58  set color0 = white
59  set color1 = black
60  set color2 = blue
61  set color3 = green
62  set color4 = red
63  set color5 = yellow
64  set color6 = violet
65  plot v(a) v(b)+20 v(c)+40 v(out)+60
66
67  .endc
68  .end
```

Output

The input output waveforms are plotted below.

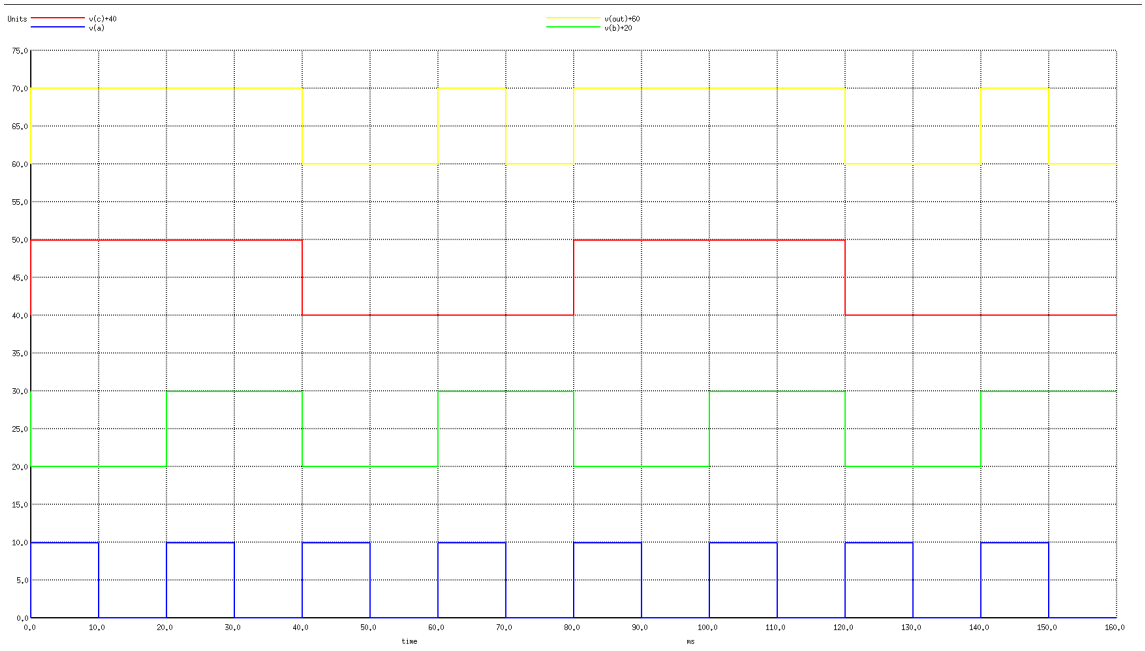


Figure 64: Ip & Op waveforms of AB+C circuit.

Observations & Conclusion

We have successfully simulated a program of AB+C Boolean Expression using CMOS Logic.

Assignment Q5: Simulate Majority 3 Gate Using Nand Gates Only.

(You are allowed to use only 2-i/p and 3-i/p NAND Gates Only)

Circuit Diagram

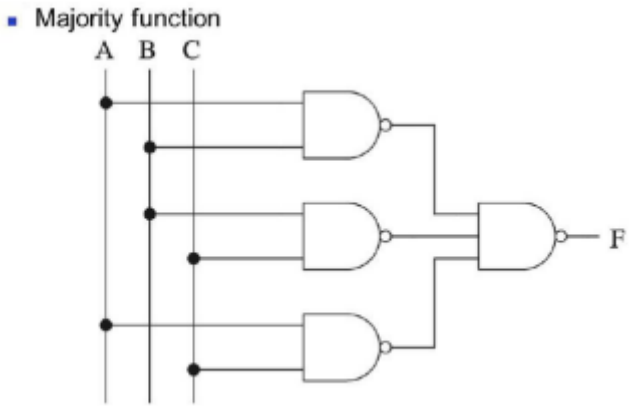


Figure 65: Circuit diagram

Code

```
1  *3 Input majority gate from NAND gate
2
3  va a 0 PULSE(0 9.9 0 1u 1u 10m 15.1m)
4  vb b 0 PULSE(9.9 0 0 1u 1u 20m 35.4m)
5  vc c 0 PULSE(0 9.9 0 1u 1u 40m 75.3m)
6
7  .subckt cmos_2nand in1 in2 res
8  * name port1 port2  ...
9      .model MOSN NMOS level=8 version=3.3.0
10     .model MOSP PMOS level=8 version=3.3.0
11     * transistor models
12
13     vdd d 0 dc 10
14     *biasing voltage
15
16     r1 in1 1 1k
17     r2 in2 2 1k
18     r4 10 res 1k
19     * resistors
20
21     m1 10 1 d d MOSP
```

```
22      m2 10 2 d d MOSP
23      m4 10 2 11 0 MOSN
24      m5 11 1 0 0 MOSN
25
26      * mn drain gate source bulk model
27      * nmos bulk to ground, pmos bulk to vdd
28  .ends cmos_2nand
29
30  * 3-INPUT NAND GATE
31  .SUBCKT NAND3 A B C Y
32  X1 A B D cmos_2nand
33  X2 D D E cmos_2nand
34  X3 C E Y cmos_2nand
35  .ENDS
36
37  X1 a b Y1 cmos_2nand
38  X2 b c Y2 cmos_2nand
39  X3 c a Y3 cmos_2nand
40  X4 Y1 Y2 Y3 out NAND3
41
42  .control
43  tran 0.01m 80ms
44  run
45  set color0 = white
46  set color1 = black
47  set color2 = blue
48  set color3 = green
49  set color4 = red
50  set color5 = violet
51  plot v(a) v(b)+20  v(c)+40 v(out)+60
52  .endc
53  .end
```

Output

The input output waveforms are plotted below.

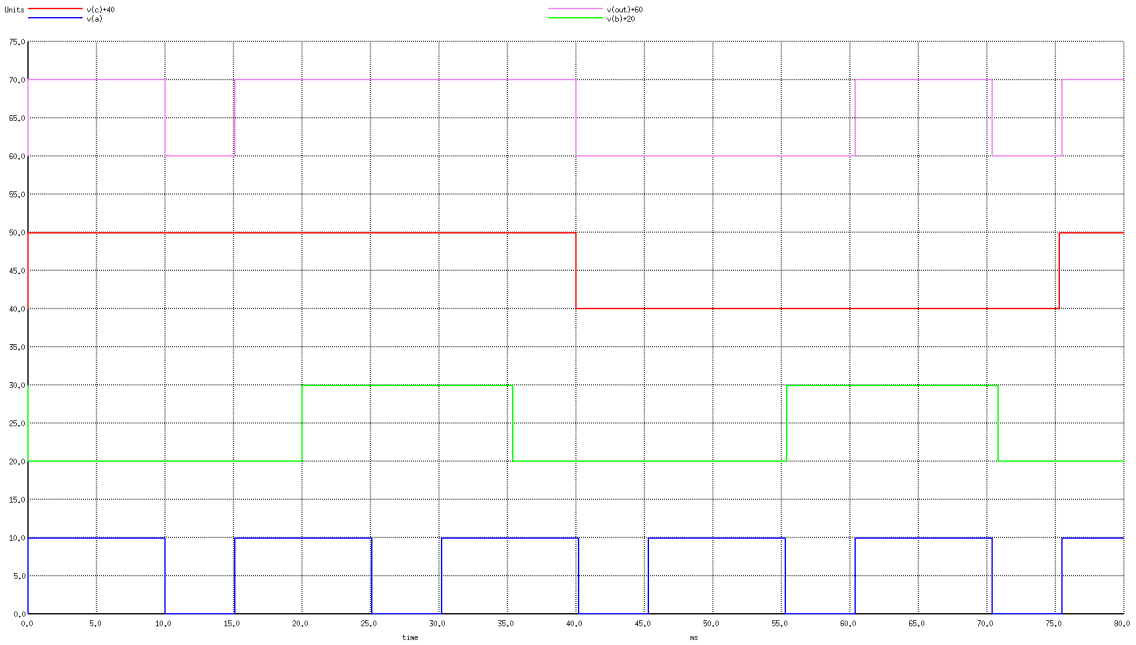


Figure 66: Ip & Op waveforms of Majority circuit.

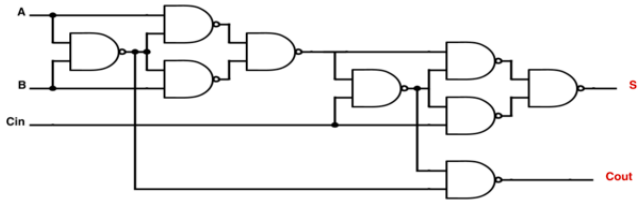
Observations & Conclusion

The 3-input Majority gate was implemented using three 2-input NAND gates and one 3-input NAND gate.

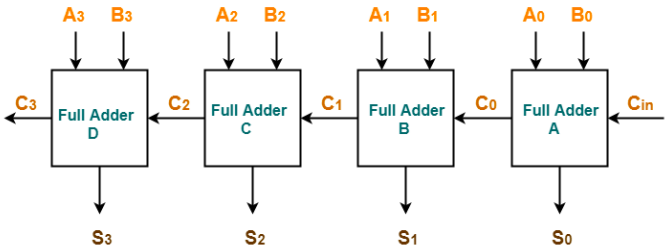
Assignment 7: GATES

Assignment Q1: Simulate 4-bit Adder using 1-bit Adder symbol.

Circuit Diagram



(a) 1-bit Full Adder using NAND only



(b) 4-bit Full Adder using 1-bit full Adder.

Figure 67: 4-bit Full Adder circuit

Code

```
*4 bit full adder
vdd vcc 0 dc 15v
vA1 a1 0 pulse (0v 9.9v 0 0 0 10ms 50ms)
vB1 b1 0 pulse (0v 9.9v 5ms 0 0 10ms 50ms)
vA2 a2 0 pulse (0v 9.9v 0 0 0 10ms 50ms)
vB2 b2 0 pulse (0v 9.9v 5ms 0 0 10ms 50ms)
vA3 a3 0 pulse (0v 9.9v 0 0 0 10ms 50ms)
vB3 b3 0 pulse (0v 9.9v 5ms 0 0 10ms 50ms)
vA4 a4 0 pulse (0v 9.9v 0 0 0 10ms 50ms)
vB4 b4 0 pulse (0v 9.9v 5ms 0 0 10ms 50ms)
vcin cin 0 pulse (0v 9.9v 0 0 0 5ms 20ms)

*NAND Gate
.subckt nandgate 2 3 4
    vdd 1 0 dc 5v
    m1 1 2 4 1 mos1 l=10u w = 100u
```

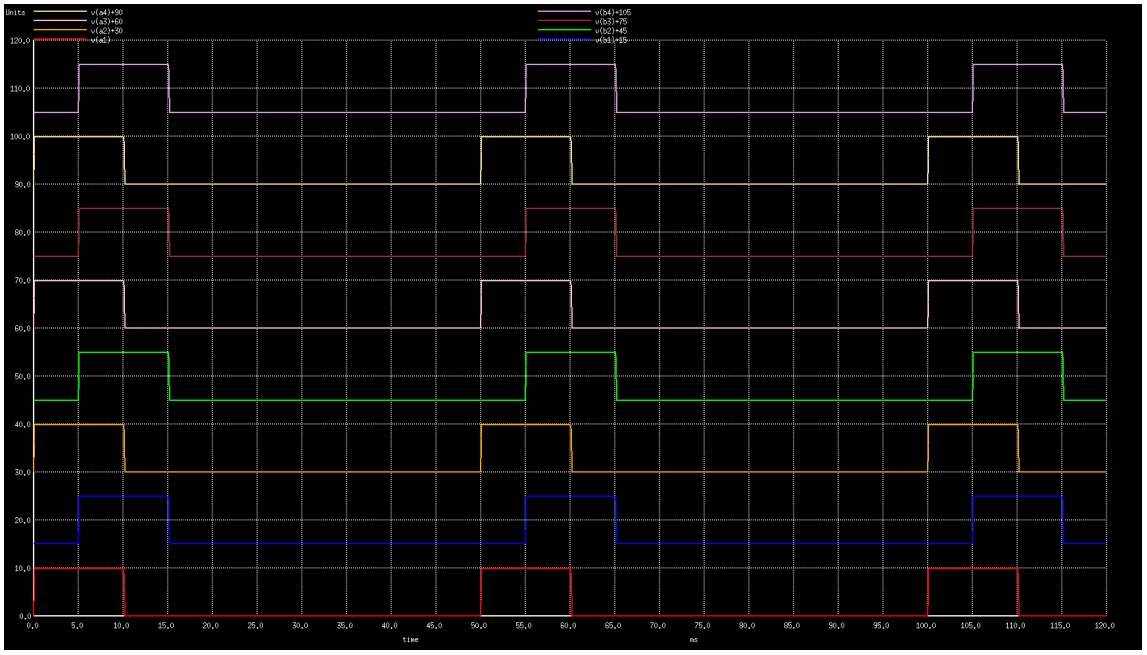
```
17      .model mos1 pmos vto=1.5v kp=200u
18      m2 1 3 4 1 mos2 l=10u w=100u
19      .model mos2 pmos vto=1.5v kp=200u
20      m3 4 2 5 5 mos3 l=10u w=100u
21      .model mos3 nmos vto=1.5v kp=200u
22      m4 5 3 0 0 mos4 l=10u w=100u
23      .model mos4 nmos vto=1.5v kp=200u
24  .ends nandgate
25
26  *1bit adder
27      .subckt onebitadder A B C S Co
28          * NODES: INPUT(2), CARRY-IN, OUTPUT, CARRY-OUT
29          X1 A B X1 nandgate
30          X2 A X1 X2 nandgate
31          X3 B X1 X3 nandgate
32          X4 X2 X3 X4 nandgate
33          X5 C X4 X5 nandgate
34          X6 C X5 X6 nandgate
35          X7 X4 X5 X7 nandgate
36          X8 X6 X7 S nandgate
37          X9 X5 X1 Co nandgate
38      .ends onebitadder
39
40  xadder1 a1 b1 cin s1 c1 onebitadder
41  xadder2 a2 b2 c1 s2 c2 onebitadder
42  xadder3 a3 b3 c2 s3 c3 onebitadder
43  xadder4 a4 b4 c3 s4 cout onebitadder
44
45  .tran 0.1ms 120ms
46  .control
47  run
48  plot v(a1) v(b1)+15  v(a2)+30 v(b2)+45 v(a3)+60 v(b3)+75 ...
        v(a4)+90 v(b4)+105
49  plot v(cin) v(s1)+13 v(c1)+20 v(s2)+30 v(c2)+40 v(s3)+50 ...
        v(c3)+60 v(s4)+70 v(cout)+80
50
51  .endc
52  .end
```

Output

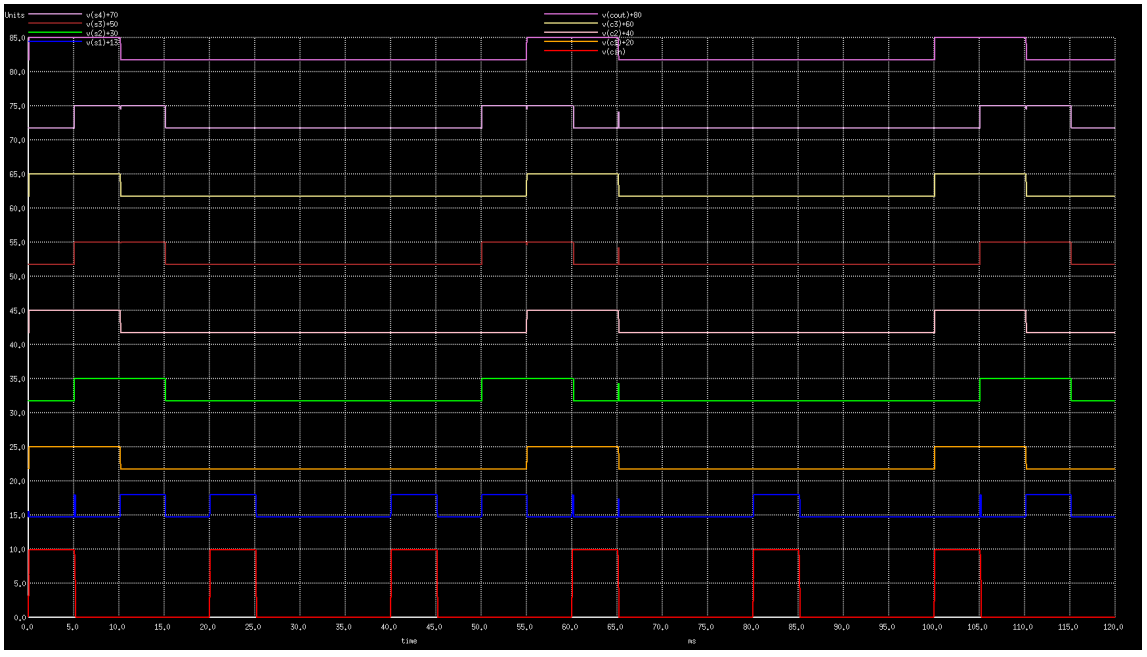
The input output waveforms are plotted below.

Observations & Conclusion

4-bit Adder was successfully implemented using 1-bit adder. 1-bit adder is implemented using only NAND gates.



(a) Input Waveforms (a1, b1, a2, b2, a3, b3, a4, b4)



(b) Output Waveforms (cin, s1, c1, s2, c2, s3, c3, s4, cout).

Figure 68: 4-bit Full Adder

Assignment Q2: Simulate Half Subtractor using Ngspice.

Circuit Diagram

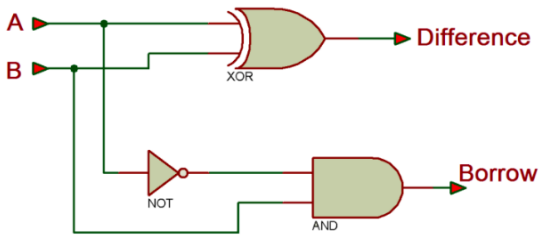


Figure 69: Half Subtractor

Code

```
*Half Subtractor
va a 0 PULSE(0 9.9 0 1u 1u 10m 15m)
vb b 0 PULSE(9.9 0 0 1u 1u 20m 35m)

.subckt cmos_2nand in1 in2 res
* name port1 port2  ...
    .model MOSN NMOS level=8 version=3.3.0
    .model MOSP PMOS level=8 version=3.3.0
    * transistor models
    vdd d 0 dc 10
    *biasing voltage

    r1 in1 1 1k
    r2 in2 2 1k

    r4 10 res 1k
    * resistors

    m1 10 1 d d MOSP
    m2 10 2 d d MOSP

    m4 10 2 11 0 MOSN
    m5 11 1 0 0 MOSN

    * mn drain gate source bulk model
    * nmos bulk to ground, pmos bulk to vdd
.ends cmos_2nand

.subckt cmos_and i1 i2 res
```

```
30 * name port1 port2 port3 ...
31     .model MOSN NMOS level=8 version=3.3.0
32     .model MOSP PMOS level=8 version=3.3.0
33
34     vdd d 0 dc 10
35
36     r1 5 res 1k
37     r2 i1 1 1k
38     r3 i2 2 1k
39     r4 3 4 1k
40
41     m1 3 1 d d MOSP
42     m2 3 2 d d MOSP
43     m3 5 4 d d MOSP
44
45     m4 3 2 6 0 MOSN
46     m5 6 1 0 0 MOSN
47     m6 5 4 0 0 MOSN
48 .ends cmos_and
49
50 x1 a b c1 cmos_2nand
51 x2 a c1 c2 cmos_2nand
52 x3 b c1 c3 cmos_2nand
53 x4 c2 c3 difference cmos_2nand
54
55 x5 a a not cmos_2nand
56 x6 not b borrow cmos_and
57
58 .tran 10u 50m
59 .control
60 run
61 set color0 = white
62 set color1 = black
63 set color2 = blue
64 set color3 = green
65 set color4 = red
66 set color5 = yellow
67 set color6 = violet
68 plot v(a) v(b)+20 v(difference)+40 v(borrow)+60
69 .endc
70 .end
```

Output

The input output waveforms are plotted below.

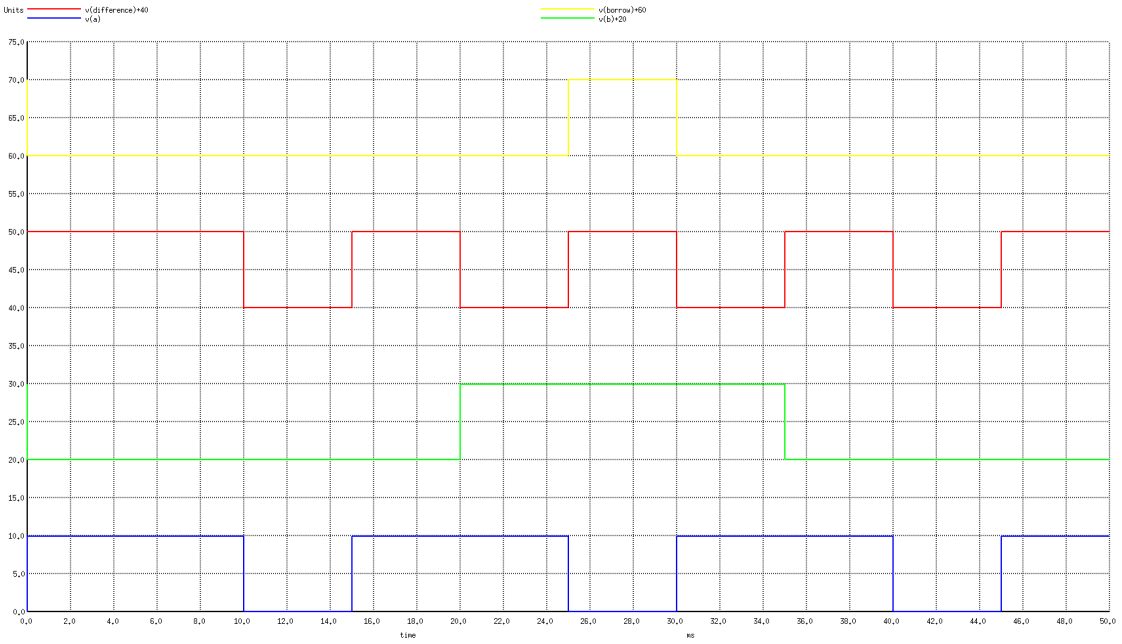


Figure 70: Ip & Op waveforms of Half Subtractor (a, b, diff, borrow).

Observations & Conclusion

Half Subtractor was implemented successfully using MOSFETS.

Assignment Q3: Simulate AOI22 circuit using Ngspice.

Circuit Diagram

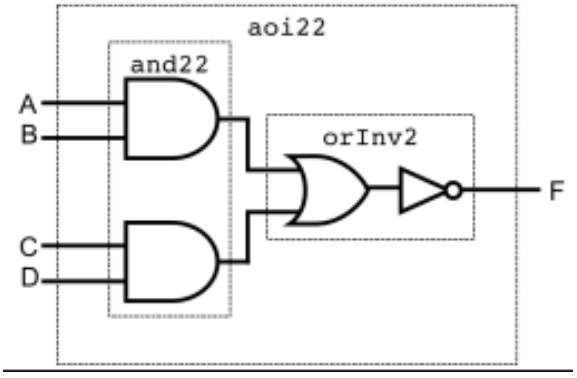


Figure 71: AOI22 circuit diagram.

Code

```
*aoi22

va a 0 PULSE(0 9.9 0 1u 1u 10m 15m)
vb b 0 PULSE(9.9 0 0 1u 1u 20m 35m)
vc c 0 PULSE(0 9.9 0 1u 1u 40m 80m)
vd d 0 PULSE(9.9 0 0 1u 1u 80m 100m)

.subckt cmos_2nand in1 in2 res
* name port1 port2  ...
    .model MOSN NMOS level=8 version=3.3.0
    .model MOSP PMOS level=8 version=3.3.0

    vdd d 0 dc 10

    r1 in1 1 1k
    r2 in2 2 1k

    r4 10 res 1k

    m1 10 1 d d MOSP
    m2 10 2 d d MOSP

    m4 10 2 11 0 MOSN
    m5 11 1 0 0 MOSN

```

```
26 .ends cmos_2nand
27
28
29 .subckt cmos_or i1 i2 res
30 * name port1 port2 port3 ...
31     .model MOSN NMOS level=8 version=3.3.0
32     .model MOSP PMOS level=8 version=3.3.0
33
34     vdd d 0 dc 10
35
36     r1 i1 1 1k
37     r2 i2 2 1k
38     r3 6 res 1k
39     r4 4 5 1k
40
41     m1 3 1 d d MOSP
42     m2 4 2 3 d MOSP
43     m3 6 5 d d MOSP
44
45     m4 4 1 0 0 MOSN
46     m5 6 5 0 0 MOSN
47     m6 4 2 0 0 MOSN
48 .ends cmos_or
49
50 .subckt cmos_and i1 i2 res
51 * name port1 port2 port3 ...
52     .model MOSN NMOS level=8 version=3.3.0
53     .model MOSP PMOS level=8 version=3.3.0
54
55     vdd d 0 dc 10
56
57     r1 5 res 1k
58     r2 i1 1 1k
59     r3 i2 2 1k
60     r4 3 4 1k
61
62     m1 3 1 d d MOSP
63     m2 3 2 d d MOSP
64     m3 5 4 d d MOSP
65
66     m4 3 2 6 0 MOSN
67     m5 6 1 0 0 MOSN
68     m6 5 4 0 0 MOSN
69 .ends cmos_and
70
71 x1 a b r1 cmos_and
72 x2 c d r2 cmos_and
73 x3 r1 r2 or cmos_or
74 x4 or or out cmos_2nand
```

```
75
76 .tran 10u 160m
77 .control
78 run
79 set color0 = white
80 set color1 = black
81 set color2 = blue
82 set color3 = green
83 set color4 = red
84 set color5 = yellow
85 set color6 = violet
86 plot v(a) v(b)+20 v(c)+40 v(d)+60 v(out)+80
87
88 .endc
89 .end
```

Output

The input output waveforms are plotted below.

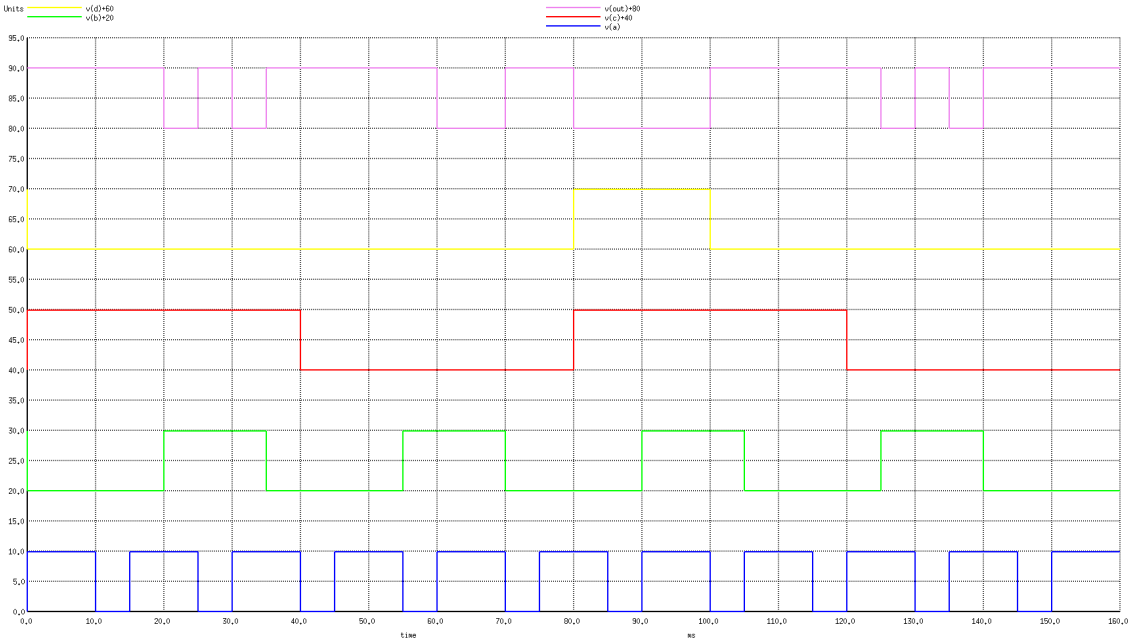


Figure 72: Ip & Op waveforms of AOI22.

Observations & Conclusion

AOI22 circuit was implemented successfully using MOSFETS.

Assignment 8: Flip-Flops

Assignment Q1: Simulate the edge triggering D-Flip Flop.

Circuit Diagram

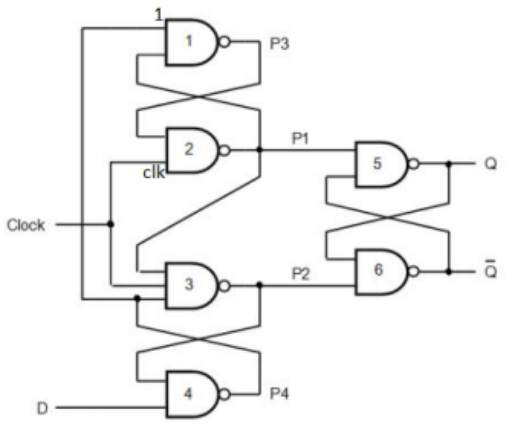


Figure 73: Circuit diagram

Characteristics Table

Clk	D	Q		Description
↓ » 0	X	Q	$\overline{Q}$	Memory no change
↑ » 1	0	0	1	Reset Q » 0
↑ » 1	1	1	0	Set Q » 1

Figure 74: Characteristics Table for D FlipFlop

Code

```
1  *Edge_trigerred D-Flip flop
2
3  *NAND
4  .subckt NAND n1 n2 d1
5      Vdc dc 0 10
6      m1 d1 n1 s1 0 NFET
```

```
7       m2 s1 n2 0 0 NFET
8       m3 dc n1 d1 dc PFET
9       m4 dc n2 d1 dc PFET
10      .model NFET NMOS
11      .model PFET PMOS
12  .ends
13
14  *NAND3IP USING 2IPNAND
15  .subckt NAND3 in1 in2 in3 out3
16      X1 in1 in2 out1 NAND
17      X2 out1 out1 out2 NAND
18      X3 in3 out2 out3 NAND
19  .ends
20
21  Va A 0 PULSE(0 9.9 1us 1us 1us 10ms 20ms)
22  Vcin C 0 PULSE(9.9 0 1us 1us 1us 15ms 45ms)
23
24  X1 p4 p1 p3 NAND
25  X2 C p3 p1 NAND
26  X3 p1 C p4 p2 NAND3
27  X4 p2 A p4 NAND
28  X5 p1 Qbar Q NAND
29  X6 p2 Q Qbar NAND
30
31  .control
32  tran 0.01ms 90ms
33  Set Xbrushwidth=2
34  plot V(C) V(A)+20 V(Q)+40 V(Qbar)+60
35  .endc
36  .end
```

Output

The input output waveforms are plotted below.

Observations & Conclusion

We have successfully implemented positive edge D flip flop.

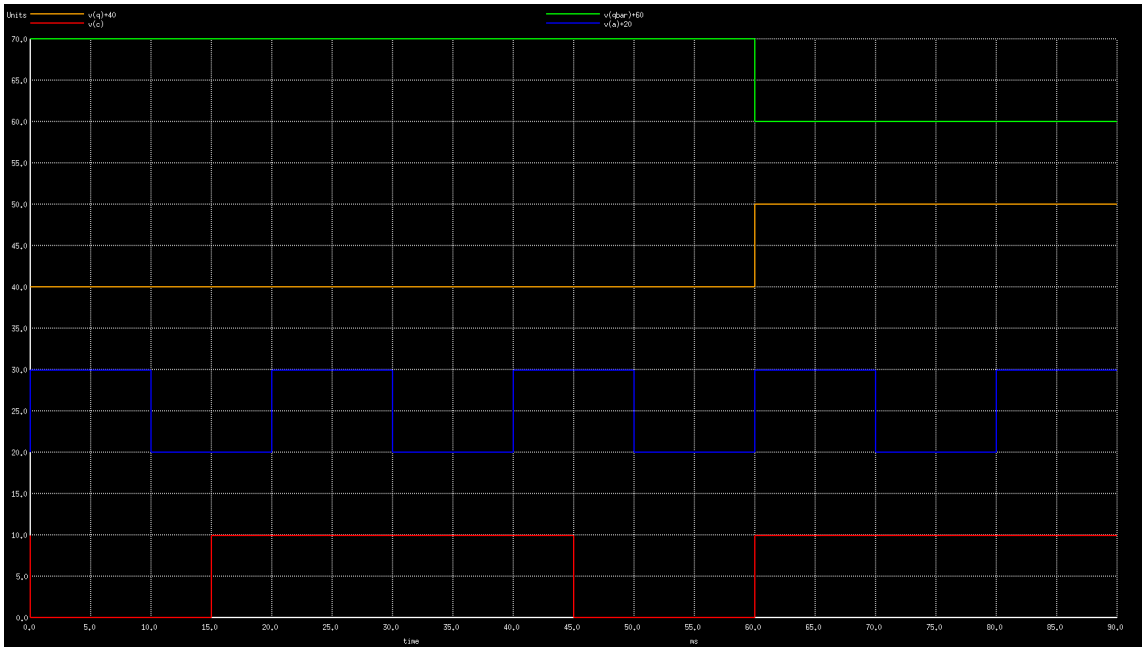


Figure 75: Ip & Op waveforms of D-edge FF (clk, d, q, qbar).

Assignment Q2: Simulate the negative level triggering D-Flip Flop.

Circuit Diagram

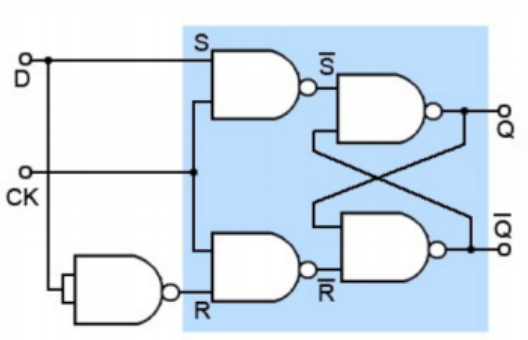


Figure 76: Circuit diagram

Code

```
1 *D flipflop negative level
```

```
2
3  *NAND
4  .subckt nand A B OP
5      Vdd 1 0 dc 10v
6      m1 1 A OP 1 mosp
7      m2 1 B OP 1 mosp
8      m3 OP A 3 0 mosn
9      m4 3 B 0 0 mosn
10     .model mosn nmos
11     .model mosp pmos
12 .ends
13
14 Vd d 0 pulse(0 9.9 0 0 0 100ns 300ns)
15 Vclk clk 0 pulse(0 9.9 0 0 0 125ns 250ns)
16 X1 d d notd nand
17 X2 d clk p1 nand
18 X3 notd clk p2 nand
19 X4 p1 notq q nand
20 X5 p2 q notq nand
21
22 .tran 1ns 1000ns
23 .control
24 run
25 plot v(d) v(clk)+20 v(q)+40 v(notq)+60
26 .endc
27 .end
```

Output

The input output waveforms are plotted below.

Observations & Conclusion

We have successfully implemented negative level D flip flop.

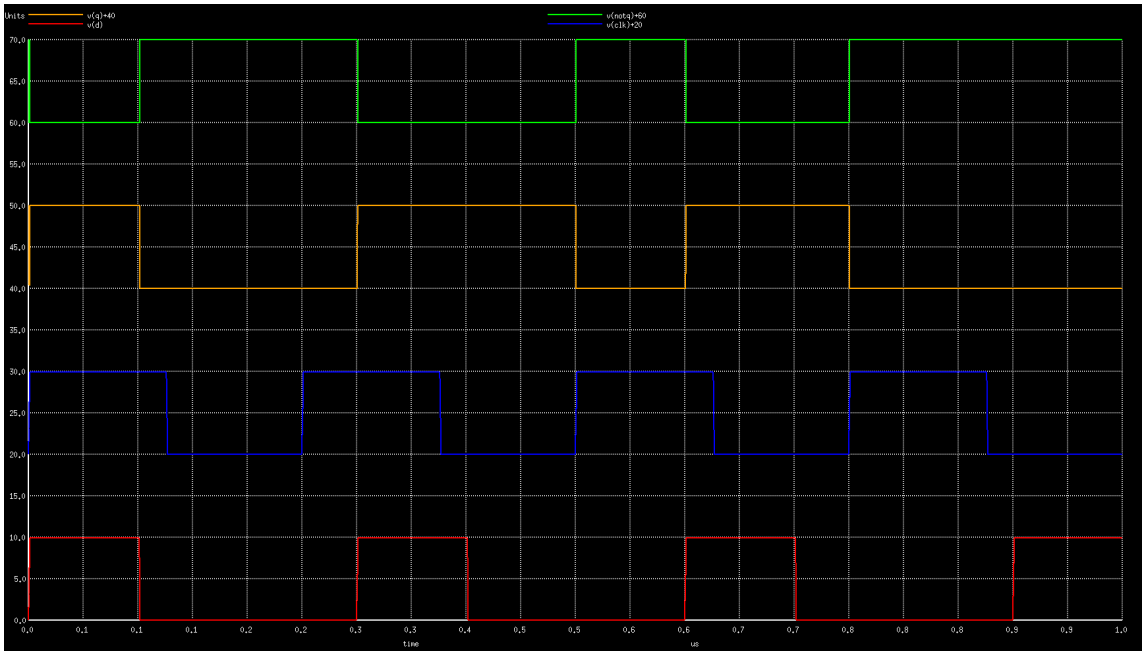


Figure 77: Ip & Op waveforms of D-edge FF (d, clk, q, qbar).

Assignment Q3: Simulate T-Flip Flop using Ngspice.

Circuit Diagram

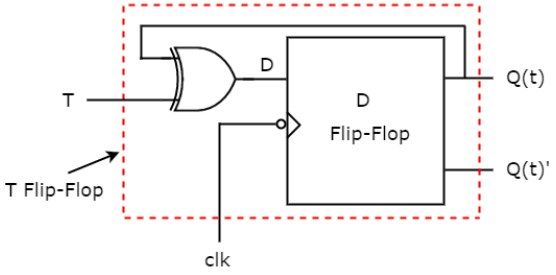


Figure 78: Circuit diagram

Characteristics Table

Code

```
1 *tflipflop
2 * xor cmos
```

CLK	T	Q	Q+1
_↑-	0	0	0
_↑-	0	1	1
_↑-	1	0	1
_↑-	1	1	0

Figure 79: Characteristics Table for T FlipFlop

```
3  .subckt XOR n1 n2 b3
4      .model NFET NMOS
5      .model PFET PMOS
6      m1 d1 n1 b1 d1 PFET
7      m2 b1 n1 0 0 NFET
8      m3 n1 n2 b3 d1 PFET
9      m4 b3 n2 0 0 NFET
10 .ends
11
12 *Edge-trigerred D-Flip flop
13     *NAND
14     .subckt NAND n1 n2 d1
15         Vdc dc 0 10
16         m1 d1 n1 s1 0 NFET
17         m2 s1 n2 0 0 NFET
18         m3 dc n1 d1 dc PFET
19         m4 dc n2 d1 dc PFET
20         .model NFET NMOS
21         .model PFET PMOS
22     .ends
23
24     *NAND3IP USING 2IPNAND
25     .subckt NAND3 in1 in2 in3 out3
26         X1 in1 in2 out1 NAND
27         X2 out1 out1 out2 NAND
28         X3 in3 out2 out3 NAND
29     .ends
30     .subckt Dflipflop A C Q Qbar
31         X1 p4 p1 p3 NAND
32         X2 C p3 p1 NAND
33         X3 p1 C p4 p2 NAND3
34         X4 p2 A p4 NAND
35         X5 p1 Qbar Q NAND
36         X6 p2 Q Qbar NAND
37     .ends
```

```
38
39 Va A 0 PULSE(9.9 0 1us 1us 1us 20ms 45ms)
40 Vcin C 0 PULSE(0 9.9 1us 1us 1us 10ms 20ms)
41 X1 A Q D XOR
42 X2 D C Q Qbar Dflipflop
43 .control
44 tran 0.01ms 90ms
45 Set Xbrushwidth=2
46 plot V(C) V(A)+20 V(Q)+40 V(Qbar)+60
47 .endc
48 .end
```

Output

The input output waveforms are plotted below.

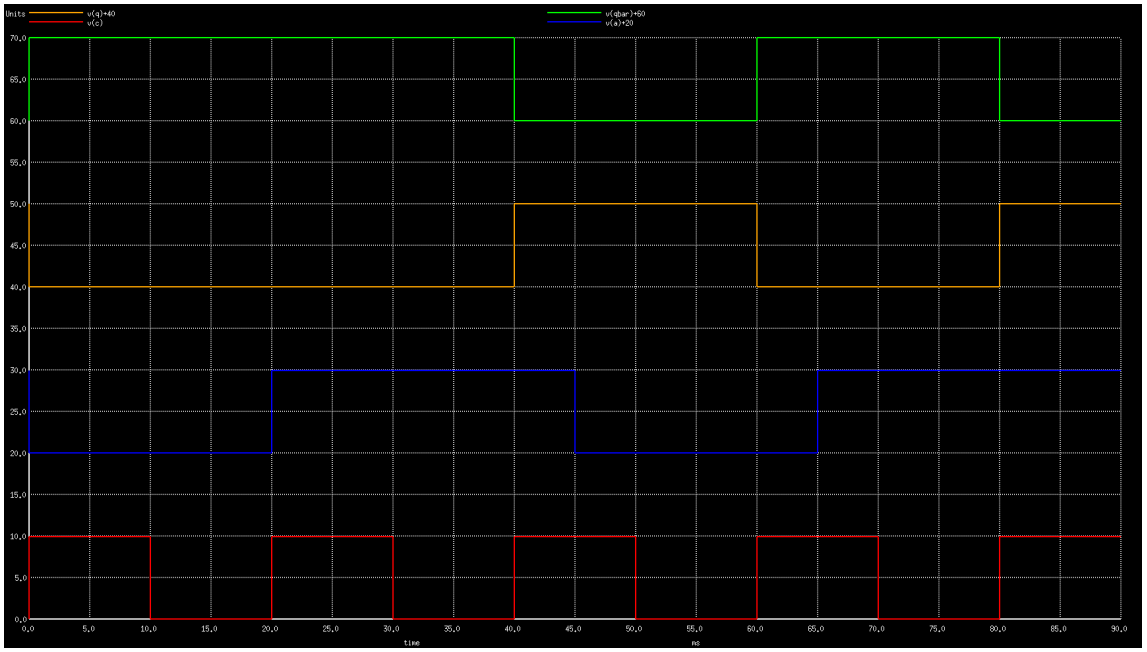


Figure 80: Ip & Op waveforms of T-edge FF (clk, t, q, qbar).

Observations & Conclusion

T flip Flop is implemented successfully using D-edge flip flop. It is observed that for positive edge, if T is low output(T=0) remains same while for high state T(T=1) output toggles.