EE671: Assignment 2

Anubhav Bhatla, 200070008

September 5, 2022

1 Q-1

1.1 Part (a)

Complementary Pass gate Logic generates both true and complement outputs, therefore we can generate the XOR-XNOR pair using the following circuit:

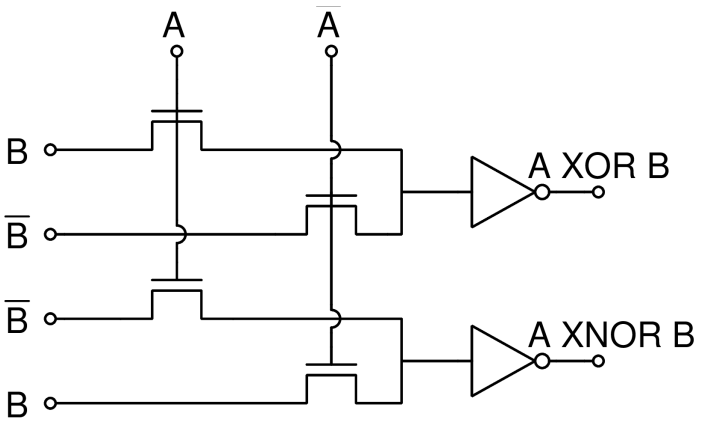


Fig. CPL Implementation of XOR and XNOR

We use CMOS inverters to restore the logic level after each stage. Because of the inverter, the multiplexer for the XOR output first calculates XNOR function given by $A.B + \bar{A}\bar{B}$. Similarly, for the XNOR output, we generate the XOR expression $= A.\bar{B} + \bar{A}.B$ which will be inverted by the inverter.

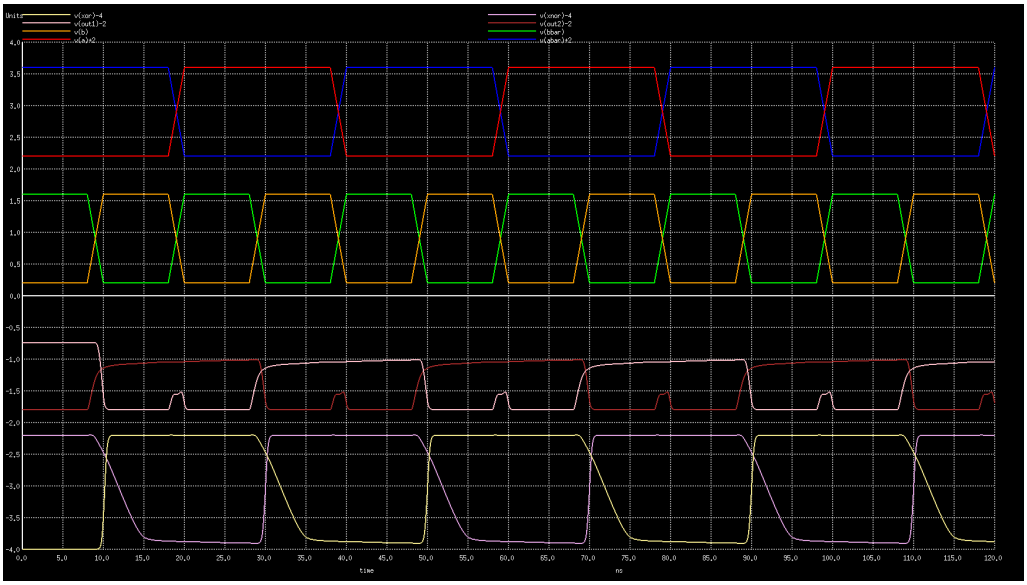
1.1.1 Code Snippet

```
CPL design for XOR gate

* Importing required files
.include Switch.txt
.include Inverter.txt
.include models-180nm
* Defining necessary parameters
.param Trep1= 40n
.param Trep2 = {Trep1/2.0}
.param Trf = {Trep1/20.0}
.param Tw1 = {Trep1/2.0 - Trf}
.param Tw2 = {Trep2/2.0 - Trf}
.param hival=1.6
.param loval=0.2
* Generating pulses for A, Abar, B, Bbar
V1 A 0 DC 0 PULSE({loval} {hival} {Tw1} {Trf} {Trf} {Tw1} {Trep1})
V2 Abar 0 DC 0 PULSE({hival} {loval} {Tw1} {Trf} {Trf} {Tw1} {Trep1})
V3 B 0 DC 0 PULSE({loval} {hival} {Tw2} {Trf} {Trf} {Tw2} {Trep2})
V4 Bbar 0 DC 0 PULSE({hival} {loval} {Tw2} {Trf} {Trf} {Tw2} {Trep2})
* Supply Voltage
vdd dummy 0 dc 1.8
vdummy dummy supply dc 0 ac 0
* Switch Matrix
x1 B Bbar Bbar B A Abar Out1 Out2 swmat
* Inverter for XOR
x2 supply Out1 XOR inv
*Inverter for XNOR
x3 supply Out2 XNOR inv
*Load capacitors
Cxor XOR 0 108f
Cxnor XNOR 0 108f
* Performing transient analysis
.tran 1pS {3*Trep1} 0nS
.control
run
* Plotting input voltage waveforms
```

```
plot V(A)+2 V(Abar)+2 V(B) V(Bbar) V(XOR)-2 V(XNOR)-2
plot i(vdummy)
meas tran i_avg AVG i(vdummy)
.endc
.end
```

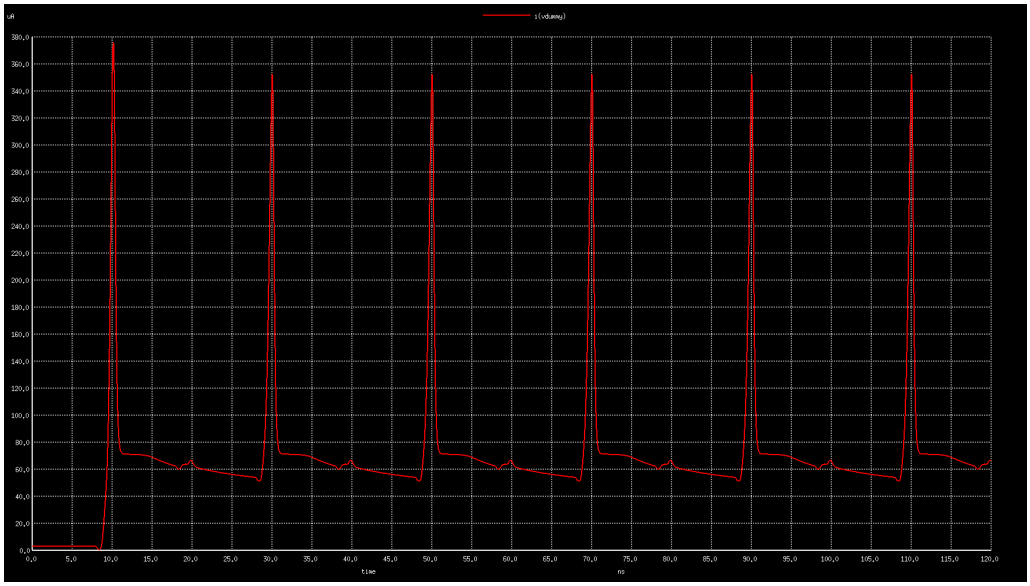
1.1.2 Simulation Results

Given below is the plot for voltages at the output of the switch matrix and the final XOR and XNOR outputs using transient analysis to show functionality of the gate:



We can observe the difference between the voltage waveforms at Out1 and XOR, Out2 and XNOR. This voltage difference is due to the presence of the inverters, which help in restoring voltage levels at each stage.

Given below is the plot for current drawn from V_{DD}:



The average current value was measured to be $69.26\mu A$, which is quite high due to the static power consumption in the pMOS transistor, also known as Buffer Leakage Current.

1.2 Part (b)

Buffer Leakage Current can be avoided by adding a pull-up pMOS as shown in the following figure:

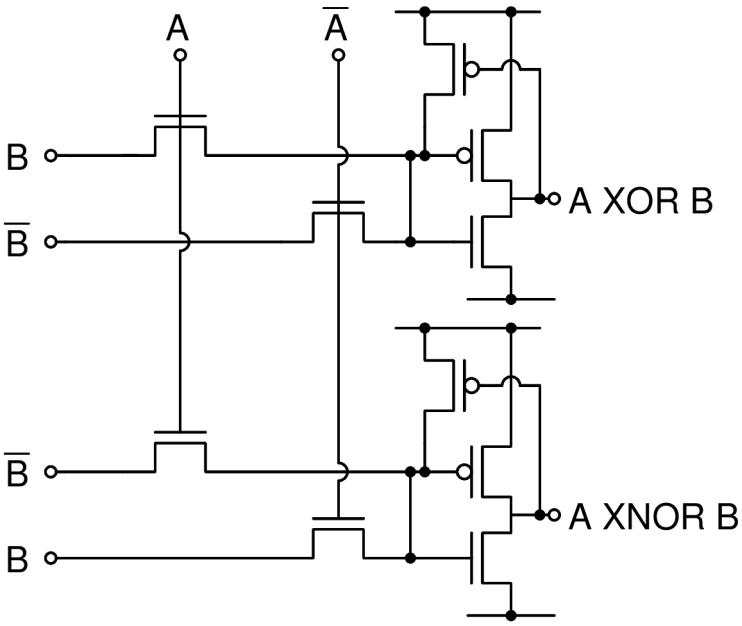


Fig. Pull-up pMOS to avoid leakage current

When the multiplexer output is 'low', the inverter output is 'high'. The pMOS is therefore 'off' and has no effect. When the multiplexer output goes 'high', the inverter output goes 'low' and the pMOS turns 'on'. Now, as the multiplexer output approaches $V_{DD} - V_{T_n}$, the nMOS switch in the multiplexer turns off. However, the pMOS pull-up remains 'on' and takes the inverter input all the way to V_{DD}. This avoids leakage in the inverter.

However, the multiplexer nMOS and the pull-up pMOS constitute a pseudo-nMOS inverter. Therefore, the multiplexer output cannot be pulled 'low' unless the transistor geometries are appropriately ratioed.
In my simulation, I used the smallest possible dimensions for the pull-up pMOS i.e. $L = 0.18\mu m$, $W = 0.24\mu m$.

1.2.1 Code Snippet

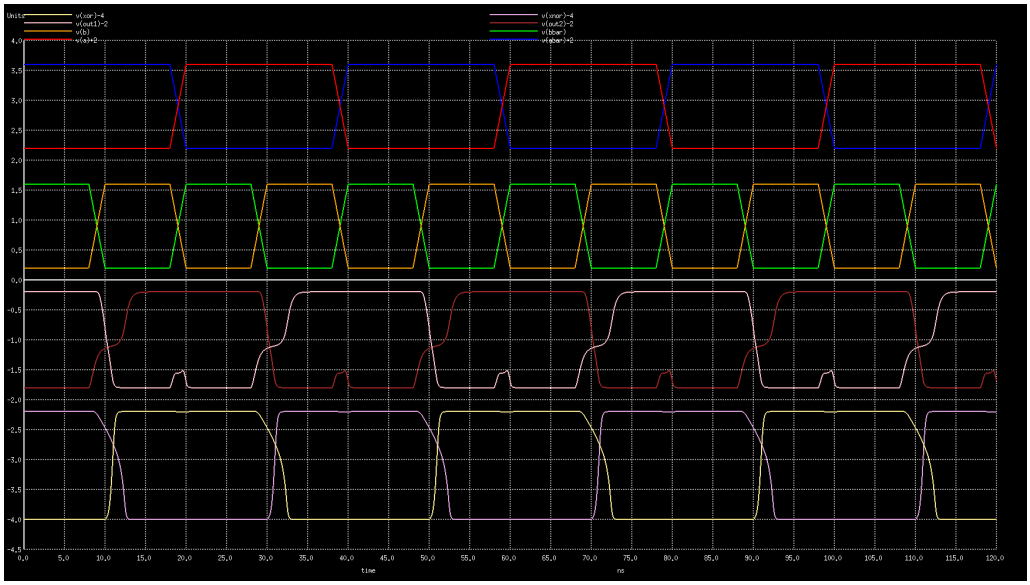
```
CPL design for XOR gate

* Importing required files
.include Switch.txt
.include Inverter.txt
```

```
.include models-180nm
* Defining necessary parameters
.param Trep1= 40n
.param Trep2 = {Trep1/2.0}
.param Trf = {Trep1/20.0}
.param Tw1 = {Trep1/2.0 - Trf}
.param Tw2 = {Trep2/2.0 - Trf}
.param hival=1.6
.param loval=0.2
* Generating pulses for A, Abar, B, Bbar
V1 A 0 DC 0 PULSE({loval} {hival} {Tw1} {Trf} {Trf} {Tw1} {Trep1})
V2 Abar 0 DC 0 PULSE({hival} {loval} {Tw1} {Trf} {Trf} {Tw1} {Trep1})
V3 B 0 DC 0 PULSE({loval} {hival} {Tw2} {Trf} {Trf} {Tw2} {Trep2})
V4 Bbar 0 DC 0 PULSE({hival} {loval} {Tw2} {Trf} {Trf} {Tw2} {Trep2})
* Supply Voltage
vdd dummy 0 dc 1.8
vdummy dummy supply dc 0 ac 0
* Switch Matrix
x1 B Bbar Bbar B A Abar Out1 Out2 swmat
* Inverter for XOR
x2 supply Out1 XOR inv
* Inverter for XNOR
x3 supply Out2 XNOR inv
* Pull-up PMOS
.param Wp = 0.24U
.param L = 0.18U
MP1 Out1 XOR Supply Supply cmosp
+ L=L W=Wp AD = (2*L*Wp) AS = (2*L*Wp) PD = (2*(2*L+Wp)) PS = (2*(2*L+Wp))
MP2 Out2 XNOR Supply Supply cmosp
+ L=L W=Wp AD = (2*L*Wp) AS = (2*L*Wp) PD = (2*(2*L+Wp)) PS = (2*(2*L+Wp))
*Load capacitors
Cxor XOR 0 108f
Cxnor XNOR 0 108f
* Performing transient analysis
.tran 1pS {3*Trep1} 0nS
.control
run
* Plotting input voltage waveforms
```

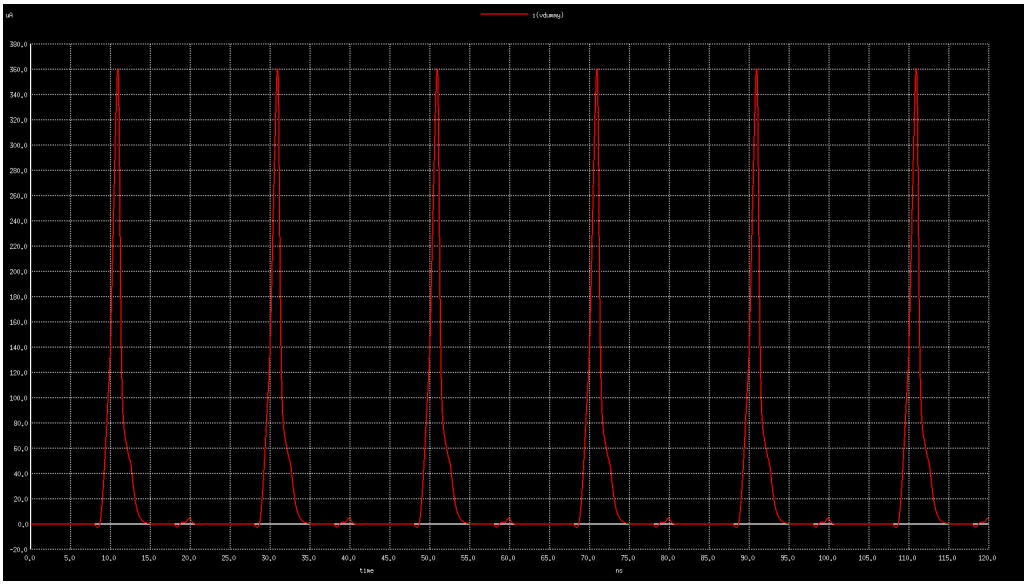
```
plot V(A)+2 V(Abar)+2 V(B) V(Bbar) V(Out1)-2 V(Out2)-2 V(XOR)-4 V(XNOR)-4
plot i(vdummy)
meas tran i_avg AVG i(vdummy)
.endc
.end
```

1.2.2 Simulation Results

Given below is the plot for voltages at the output of the switch matrix and the final XOR and XNOR outputs using transient analysis of the circuit with pull-up pMOS transistors:



Given below is the plot for current drawn from V_{DD}:



The average current value was measured to be $27.51\mu A$, which is much lower than the previous due to the pull-up pMOS transistors which help in avoiding leakage current.

1.3 Part (c)

In real-life circuits, each gate has its delay for computation and this may severly affect the working of our ciruits. In order to observe the effects of such delays on our outputs, we consider the inverter contribute a delay of 2ns, i.e. the $\bar{A}$ and $\bar{B}$ are 2ns delayed to A and B inputs:

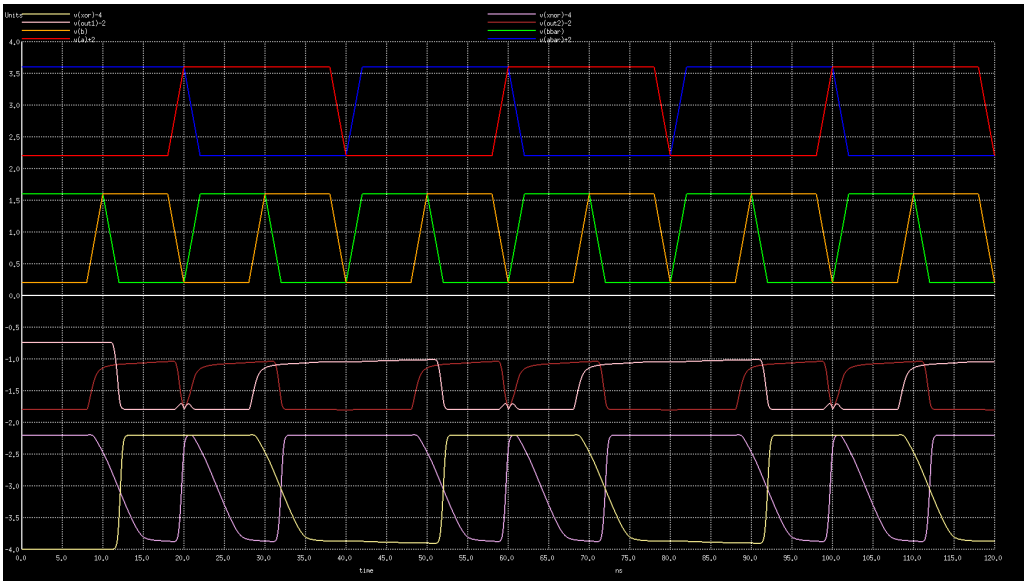
1.3.1 Code Snippet

```
CPL design for XOR gate

* Importing required files
.include Switch.txt
.include Inverter.txt
.include models-180nm
* Defining necessary parameters
.param Trep1= 40n
.param Trep2 = {Trep1/2.0}
```

```
.param Trf = {Trep1/20.0}
.param Tw1 = {Trep1/2.0 - Trf}
.param Tw2 = {Trep2/2.0 - Trf}
.param hival=1.6
.param loval=0.2
* Generating pulses for A, Abar, B, Bbar
V1 A 0 DC 0 PULSE({loval} {hival} {Tw1} {Trf} {Trf} {Tw1} {Trep1})
V2 Abar 0 DC 0 PULSE({hival} {loval} {Tw1+2n} {Trf} {Trf} {Tw1} {Trep1})
V3 B 0 DC 0 PULSE({loval} {hival} {Tw2} {Trf} {Trf} {Tw2} {Trep2})
V4 Bbar 0 DC 0 PULSE({hival} {loval} {Tw2+2n} {Trf} {Trf} {Tw2} {Trep2})
* Supply Voltage
vdd dummy 0 dc 1.8
vdummy dummy supply dc 0 ac 0
* Switch Matrix
x1 B Bbar Bbar B A Abar Out1 Out2 swmat
* Inverter for XOR
x2 supply Out1 XOR inv
*Inverter for XNOR
x3 supply Out2 XNOR inv
*Load capacitors
Cxor XOR 0 108f
Cxnor XNOR 0 108f
* Performing transient analysis
.tran 1pS {3*Trep1} 0nS
.control
run
* Plotting input voltage waveforms
plot V(A)+2 V(Abar)+2 V(B) V(Bbar) V(Out1)-2 V(Out2)-2 V(XOR)-4 V(XNOR)-4
plot i(vdummy)
meas tran i_avg AVG i(vdummy)
.endc
.end
```

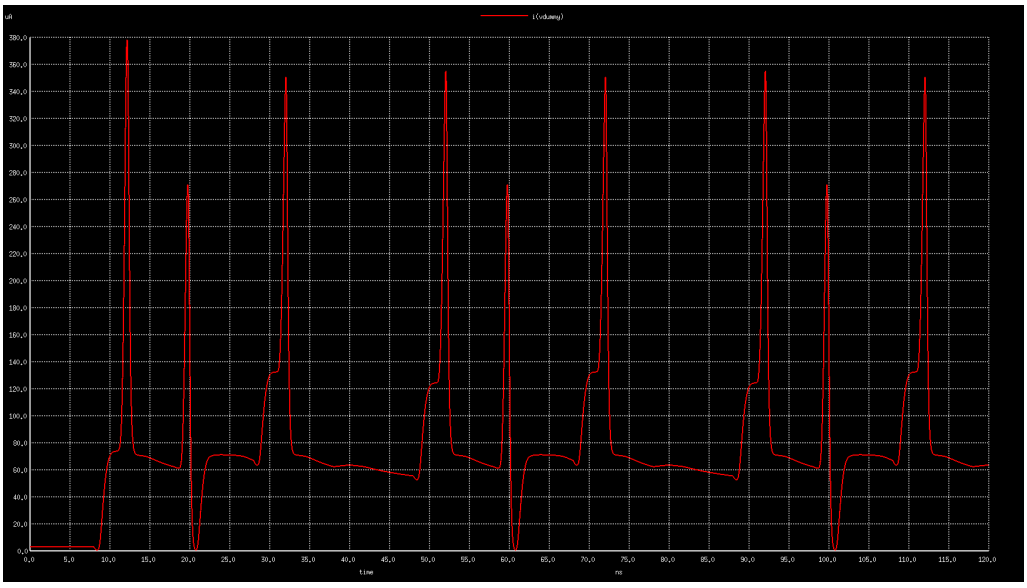
1.3.2 Simulation Results

Given below is the plot for voltages at the output of the switch matrix and the final XOR and XNOR outputs using transient analysis with additional 2ns delays for $\bar{A}$ and $\bar{B}$:



We can observe that the output waveform has unwanted peaks due to the delayed inputs, which correspond to intermediate transition state of the input.

Given below is the plot for current drawn from V_{DD}:



Similar to the voltage waveform, we can observe unwanted peaks in the current which correspond to the same transition states.

The average current value was measured to be $78.34\mu A$, which is significantly higher than the previous case due to the unwanted current surges.

1.4 Part (d)

Upon adding the pull-up pMOS, we observe that the multiplexer nMOS and the pull-up pMOS constitute a pseudo nMOS inverter which resists the output being pulled to 'low'. To fix this problem we need to appropriately ratio the pMOS, in our case we used a minimal-sized pMOS transistor. But if we take the width of the pMOS to be 4 times the minimum possible width, i.e. $W = 4 \times 0.24\mu m = 0.96\mu m$, our output waveforms will change drastically:

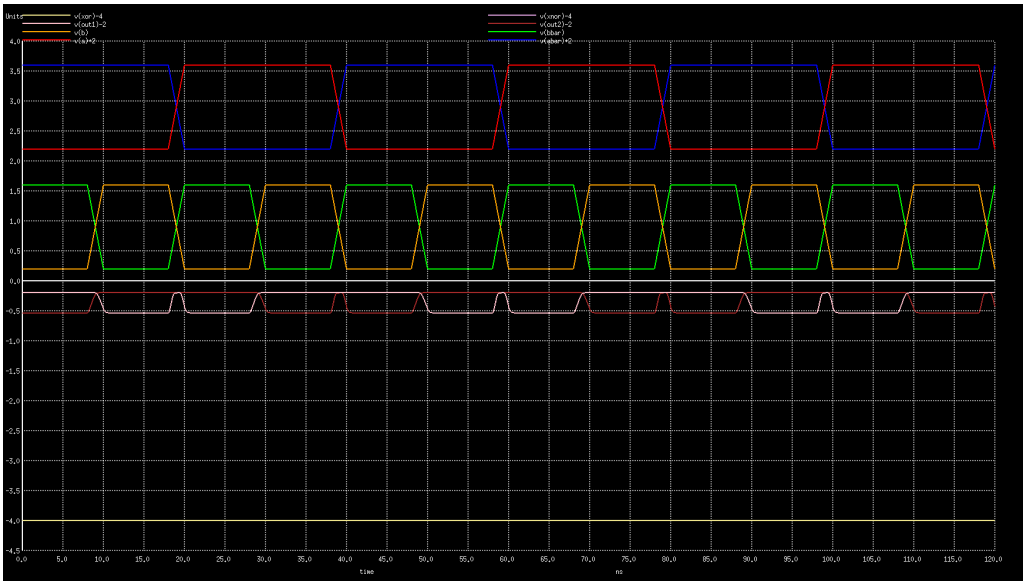
1.4.1 Code Snippet

```
CPL design for XOR gate

* Importing required files
.include Switch.txt
.include Inverter.txt
.include models-180nm
* Defining necessary parameters
.param Trep1= 40n
.param Trep2 = {Trep1/2.0}
.param Trf = {Trep1/20.0}
.param Tw1 = {Trep1/2.0 - Trf}
.param Tw2 = {Trep2/2.0 - Trf}
.param hival=1.6
.param loval=0.2
* Generating pulses for A, Abar, B, Bbar
V1 A 0 DC 0 PULSE({loval} {hival} {Tw1} {Trf} {Trf} {Tw1} {Trep1})
V2 Abar 0 DC 0 PULSE({hival} {loval} {Tw1} {Trf} {Trf} {Tw1} {Trep1})
V3 B 0 DC 0 PULSE({loval} {hival} {Tw2} {Trf} {Trf} {Tw2} {Trep2})
V4 Bbar 0 DC 0 PULSE({hival} {loval} {Tw2} {Trf} {Trf} {Tw2} {Trep2})
* Supply Voltage
vdd dummy 0 dc 1.8
vdummy dummy supply dc 0 ac 0
* Switch Matrix
x1 B Bbar Bbar B A Abar Out1 Out2 swmat
```

```
* Inverter for XOR
x2 supply Out1 XOR inv
* Inverter for XNOR
x3 supply Out2 XNOR inv
* Pull-up PMOS
.param Wp = 4*0.24U
.param L = 0.18U
MP1 Out1 XOR Supply Supply cmosp
+ L=L W=Wp AD = (2*L*Wp) AS = (2*L*Wp) PD = (2*(2*L+Wp)) PS = (2*(2*L+Wp))
MP2 Out2 XNOR Supply Supply cmosp
+ L=L W=Wp AD = (2*L*Wp) AS = (2*L*Wp) PD = (2*(2*L+Wp)) PS = (2*(2*L+Wp))
*Load capacitors
Cxor XOR 0 108f
Cxnor XNOR 0 108f
* Performing transient analysis
.tran 1pS {3*Trep1} 0nS
.control
run
* Plotting input voltage waveforms
plot V(A)+2 V(Abar)+2 V(B) V(Bbar) V(Out1)-2 V(Out2)-2 V(XOR)-4 V(XNOR)-4
plot i(vdummy)
meas tran i_avg AVG i(vdummy)
.endc
.end
```

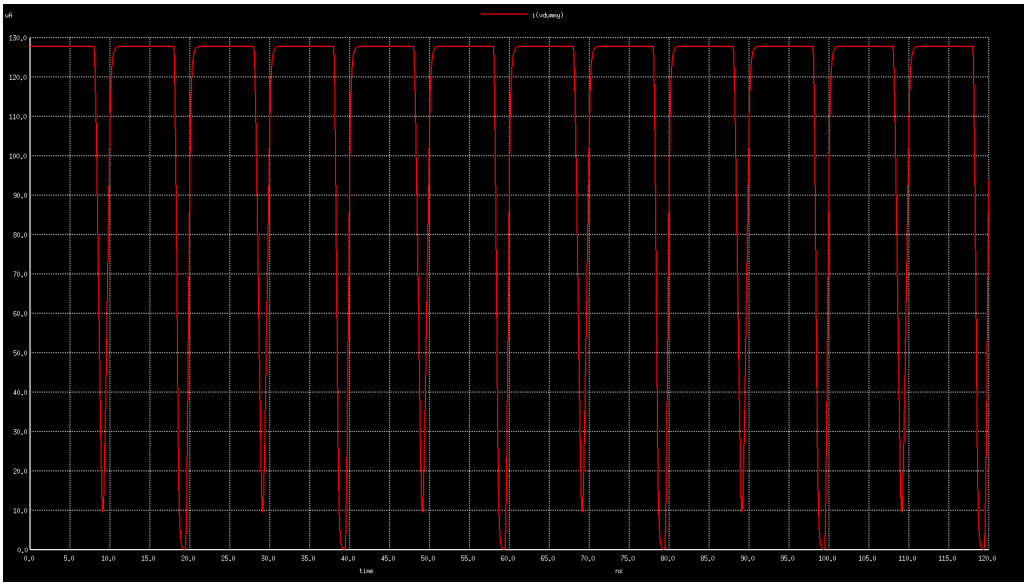
1.4.2 Simulation Results

Given below is the plot for voltages at the output of the switch matrix and the final XOR and XNOR outputs using transient analysis with the width for the pull-up pMOS $W = 4 \times 0.24\mu m = 0.96\mu m$:



We can observe that Out1 and Out2 voltage waveforms are almost 'high' for the entire duration due to the large width pMOS pulling up the voltages to 'high'. The multiplexer is unable to pull down the voltages which leads to incorrect outputs. The inverters following the switch circuit change this logic 'high' to 'low' and thus our final output waveforms for XOR and XNOR are both 'low'.

Given below is the plot for current drawn from V_{DD}:



The average current value was measured to be $111.57\mu A$, which is the highest among all previous cases. This is because the gate inputs to the pMOS is always 'low', keeping the pMOS turned 'on' at all times, which constantly draws a lot of current from supply.

2 Q-2

Like CPL, Cascade Voltage Switch Logic (CVSL) requires both true and complement signals and also provides both true and complement otuputs (dual rail logic). But the circuit in a CVSL implementation is self-latching, which reduces ratoing requirements.
In order to create a CVSL implementation for XOR-XNOR, I came up with the following circuit:

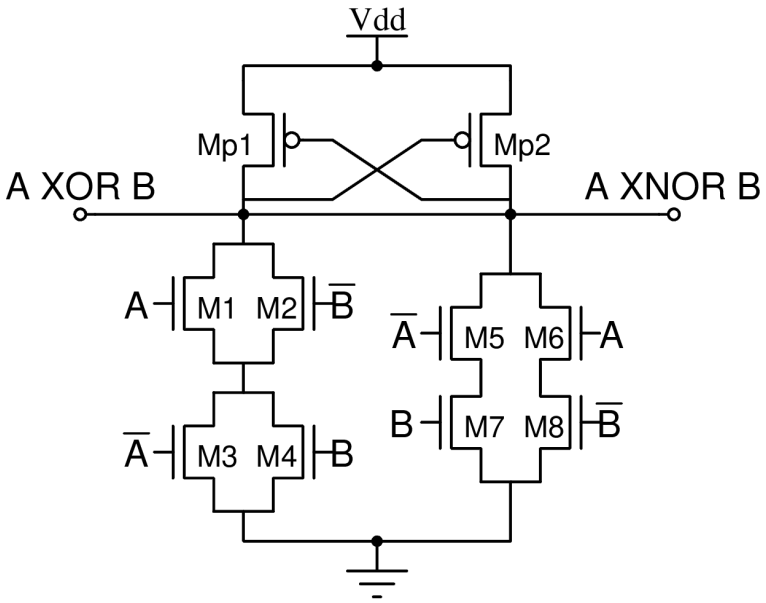


Fig. CVSL implementation for XOR and XNOR

Since at all times 2 nMOS transistors can be in series, I scaled the width of each nMOS by a factor of 2, i.e. $W = 2 \times 0.665\mu m = 1.33\mu m$.

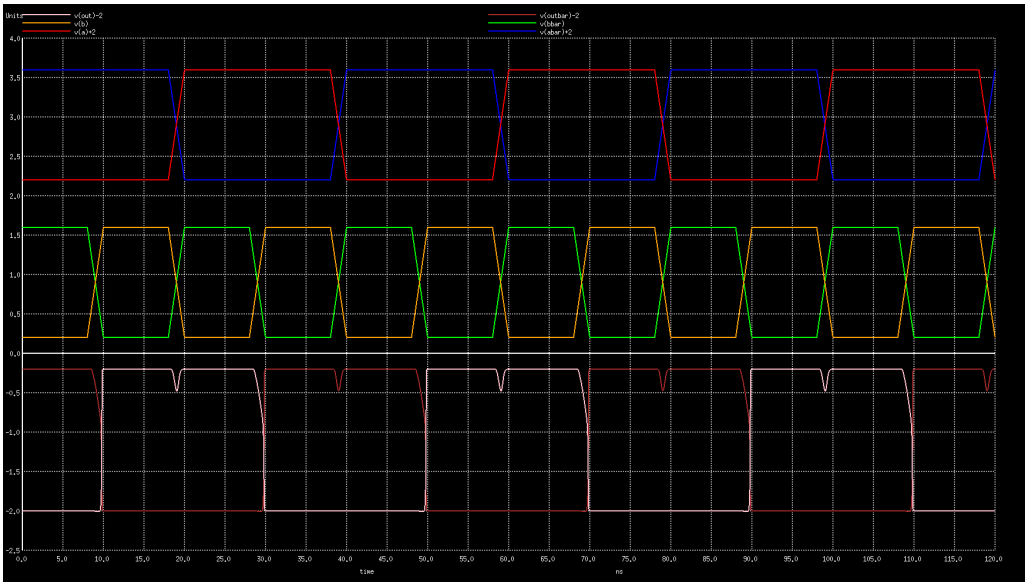
2.1 Code Snippet

```
CVSL design for XOR gate

* Importing required files
.include models-180nm
* Defining necessary parameters
.param Trep1= 40n
.param Trep2 = {Trep1/2.0}
.param Trf = {Trep1/20.0}
.param Tw1 = {Trep1/2.0 - Trf}
.param Tw2 = {Trep2/2.0 - Trf}
.param hival=1.6
.param loval=0.2
.param Wp = 1.91U
.param Wn = 2*0.665U
.param L = 0.18U
```

```
* Generating pulses for A, Abar, B, Bbar
V1 A 0 DC 0 PULSE({loval} {hival} {Tw1} {Trf} {Trf} {Tw1} {Trep1})
V2 Abar 0 DC 0 PULSE({hival} {loval} {Tw1} {Trf} {Trf} {Tw1} {Trep1})
V3 B 0 DC 0 PULSE({loval} {hival} {Tw2} {Trf} {Trf} {Tw2} {Trep2})
V4 Bbar 0 DC 0 PULSE({hival} {loval} {Tw2} {Trf} {Trf} {Tw2} {Trep2})
* Supply Voltage
vdd supply 0 dc 1.8
* PMOS transistors
MP1 Out Outbar Supply Supply cmosp
+ L=L W=Wp AD = (2*L*Wp) AS = (2*L*Wp) PD = (2*(2*L+Wp)) PS = (2*(2*L+Wp))
MP2 Outbar Out Supply Supply cmosp
+ L=L W=Wp AD = (2*L*Wp) AS = (2*L*Wp) PD = (2*(2*L+Wp)) PS = (2*(2*L+Wp))
* Circuit for XOR logic
MN1 Out A 1 1 cmosn
+ L=L W=Wn AD = (2*L*Wn) AS = (2*L*Wn) PD = (2*(2*L+Wn)) PS = (2*(2*L+Wn))
MN2 Out Bbar 1 1 cmosn
+ L=L W=Wn AD = (2*L*Wn) AS = (2*L*Wn) PD = (2*(2*L+Wn)) PS = (2*(2*L+Wn))
MN3 1 Abar 0 0 cmosn
+ L=L W=Wn AD = (2*L*Wn) AS = (2*L*Wn) PD = (2*(2*L+Wn)) PS = (2*(2*L+Wn))
MN4 1 B 0 0 cmosn
+ L=L W=Wn AD = (2*L*Wn) AS = (2*L*Wn) PD = (2*(2*L+Wn)) PS = (2*(2*L+Wn))
* Circuit for XNOR logic
MN5 Outbar Abar 2 2 cmosn
+ L=L W=Wn AD = (2*L*Wn) AS = (2*L*Wn) PD = (2*(2*L+Wn)) PS = (2*(2*L+Wn))
MN6 Outbar A 3 3 cmosn
+ L=L W=Wn AD = (2*L*Wn) AS = (2*L*Wn) PD = (2*(2*L+Wn)) PS = (2*(2*L+Wn))
MN7 2 B 0 0 cmosn
+ L=L W=Wn AD = (2*L*Wn) AS = (2*L*Wn) PD = (2*(2*L+Wn)) PS = (2*(2*L+Wn))
MN8 3 Bbar 0 0 cmosn
+ L=L W=Wn AD = (2*L*Wn) AS = (2*L*Wn) PD = (2*(2*L+Wn)) PS = (2*(2*L+Wn))
* Performing transient analysis
.tran 1pS {3*Trep1} 0nS
.control
run
* Plotting input voltage waveforms
plot V(A)+2 V(Abar)+2 V(B) V(Bbar) V(Out)-2 V(Outbar)-2
.endc
.end
```

2.2 Simulation Results

Given below is the plot for voltages at the XOR and XNOR outputs using transient analysis:



Using the plot above we can conclude that we successfully created XOR and XNOR outputs using our CVSL-based circuit design.