EE671: Assignment 2

Name: Anoushka Dey

Roll No: 210010010

September 2023

1 Question 1

For this assignment, I used the inverter that I had designed in assignment 1 with rise and fall times as 220 ps. I included the model as a sub-circuit in the main .cir file. This was followed by writing out the code for the netlist as shown in the figure given in the assignment.
The design involved creating 4 stages in the design: Input shaper, Device under test, Test load and Load on Load. If a voltage source is used as the input signal, it would not reflect realistic values as it would have zero impedance. Therefore, as per the instructions in the assignment, I buffered the voltage source using loaded inverters and used the output of the buffer as the input signal for the design.
The delay of the logic gate under test was measured from 50% of input transition to 50% of output transition. The average of the delays measured using both rising and falling inputs was taken to be the gate delay.
The entire set-up was run multiple times, changing the number of inverters as load on the device under test from 1 to 8. The delays were then saved in a .csv file and plotted using matplotlib and pandas. Using the plot obtained, I found the values of $\tau = 12.6$ ps and the y intercept = 28.8 ps and hence $p_{inv} = 2.285$. The value of $\gamma = \frac{Width\ of\ PMOS}{Width\ of\ NMOS} = 2.884 \approx 3$. This is the ngspice code that I have written:

```
Assignment 2
.include models-180nm
*Name: Anoushka Dey
*Roll 210010010 :  nn = 10

*Rise and Fall times to achieve: 200+2*10=220 ps

*Parameters
*geometry parameters
.param pw = 1.865U
.param pl = 0.18U
```

```
.param pad = {2*pw*pl}
.param pas = {2*pw*pl}
.param ppd = {2*(pw + 2*pl)}
.param pps = {2*(pw + 2*pl)}

.param nw = 0.6465U
.param nl = 0.18U
.param nad = {2*nw*nl}
.param nas = {2*nw*nl}
.param npd = {2*(nw + 2*nl)}
.param nps = {2*(nw + 2*nl)}

* Minimum Inverter
.subckt inv supply Inp Output
*  This subcircuit defines a CMOS inverter with equal n and p widths
MP1 Output Inp Supply Supply cmosp
+ L={pl} W={pw} AD={pad} AS={pas} PD={ppd} PS={pps}
MN1 Output Inp 0      0      cmosn
+ L={nl} W={nw} AD={nad} AS={nas} PD={npd} PS={nps}
.ends

*Input Shaper
* pulse with time period of Trep, rise and fall times = Trep/20
.param Trep= 5n
.param Trf = {Trep/20.0}
.param Tw = {Trep/2.0 - Trf}
.param hival=1.8
.param loval=0.0
Vpulse pgen 0 DC 0 PULSE({loval} {hival} {Tw} {Trf} {Trf} {Tw} {Trep})


vdd supply 0 dc 1.8
xinvstart supply pgen node1 inv
xinv1 supply node1 node2 inv
xinv2 supply node2 dutin inv
xinv3 supply node2 node3 inv
xinv4 supply node2 node3 inv
xinv5 supply node2 node3 inv
C1 node3 0 0.1p

*Device Under Test
xdut supply dutin dutout inv

*Test Load
xinv6 supply dutout node5 inv
xinv7 supply dutout node5 inv
```

```
xinv8 supply dutout node5 inv
xinv9 supply dutout node5 inv
xinv10 supply dutout node5 inv
xinv11 supply dutout node5 inv
xinv12 supply dutout node5 inv
xinv13 supply dutout node5 inv

*Load on load
xinv14 supply node5 node6 inv
xinv15 supply node5 node6 inv
xinv16 supply node5 node6 inv
xinv17 supply node5 node6 inv
C2 node6 0 0.1p

.tran 1pS {3*Trep} 0nS
.control
run
meas tran rise_delay TRIG v(dutin) VAL=0.9 RISE=2 TARG v(dutout) VAL=0.9 FALL=2
meas tran fall_delay TRIG v(dutin) VAL=0.9 FALL=2 TARG v(dutout) VAL=0.9 RISE=2
let delay={(rise_delay+fall_delay)/2}
print delay
.endc
.end
```

This is the python code:

```
import pandas as pd
import matplotlib.pyplot as plt
data=pd.read_csv('ee671_assgn2.csv')
plt.plot(data['n'],data['delay'],'ro')
plt.plot(data['n'],data['delay'])
plt.xlabel('No. of inverters (Fan out)')
plt.ylabel('Delay')
plt.title('Graph of Delay vs No. of Inverters')
#-----------------------------------------------
tau=(data['delay'][1]-data['delay'][0])/(data['n'][1]-data['n'][0])
tau
#-----------------------------------------------
print((4.14e-11-tau)/tau)
```

The .csv file containing the delay values has been included in the zip folder.
This is the image of the delay vs no. of inverters plot:
All delays are in seconds

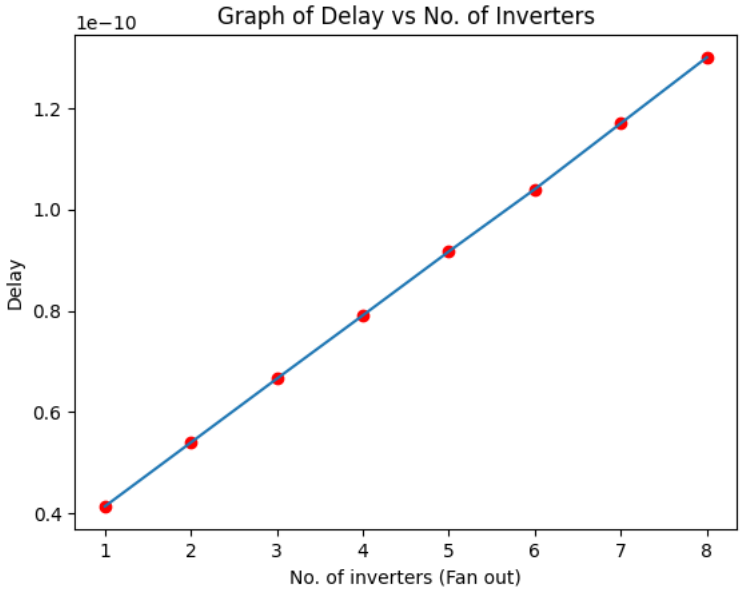

Table 1: Delay Values

No. of Inverters	Rising Delay	Falling Delay	Avg. Delay
1	4.151690e-11	4.119349e-11	4.14e-11
2	5.464846e-11	5.343972e-11	5.40e-11
3	6.739444e-11	6.573789e-11	6.66e-11
4	8.004572e-11	7.817442e-11	7.91e-11
5	9.266515e-11	9.078948e-11	9.17e-11
6	1.052931e-10	1.035602e-10	1.04e-10
7	1.179382e-10	1.164465e-10	1.17e-10
8	1.306005e-10	1.294232e-10	1.30e-10