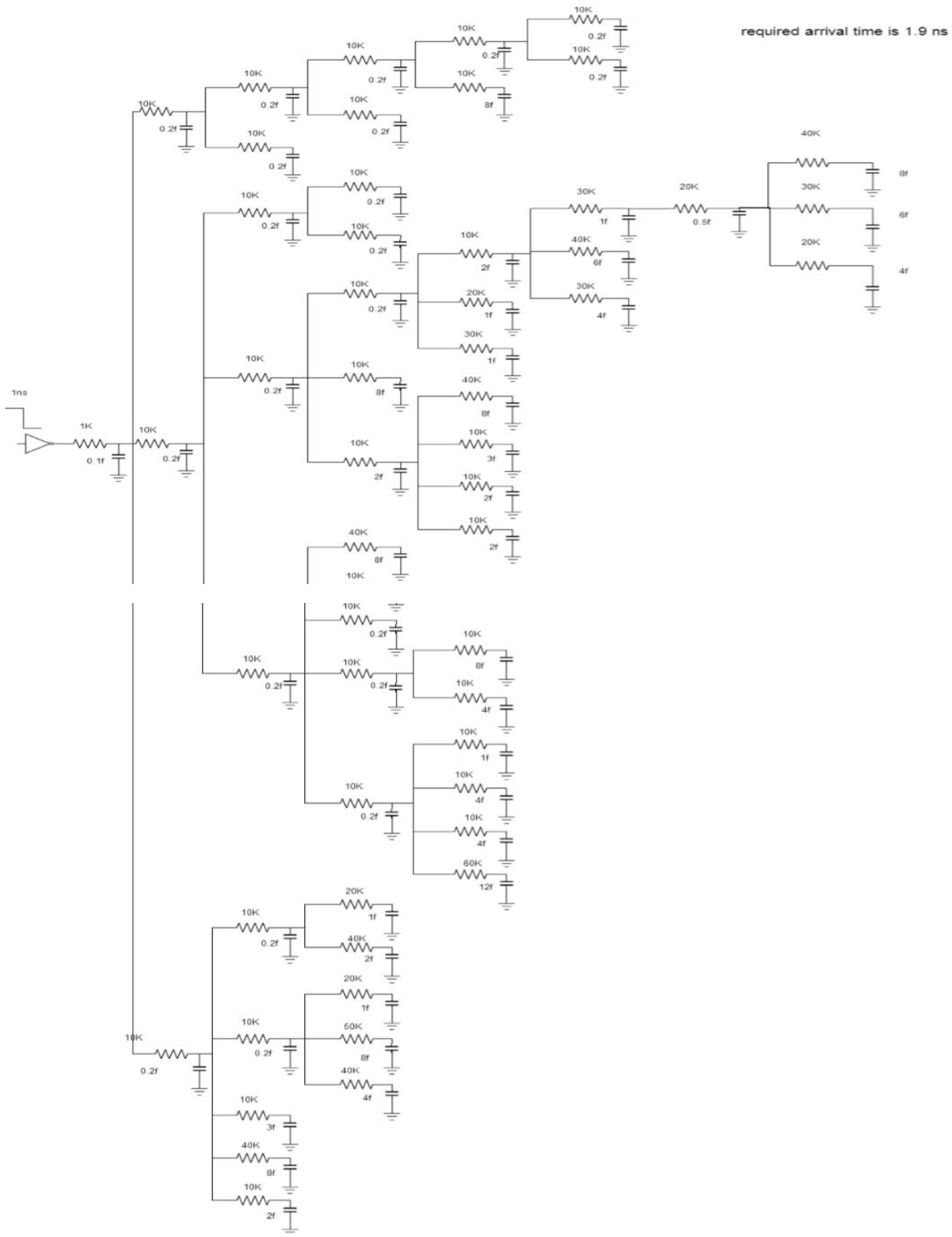


For the above RC network, we must insert buffers at points such that the arrival time is 1.9ns.

To achieve this, we will first number all the nodes and perform the calculations. The calculations will follow the below steps,

a. Determine (Li, Ti) at all the end points.
b. Then we traverse to the point before on the same line. We can determine the load and time as shown,

$$Ti' = Ti - Ri * Ci$$
$$Li' = Li + Ci$$

c. Complete this traverse back along the line for all the end points.
d. At Fan-out points, we need to choose the optimum design option from the available design option at various points. This can be done using the below method

$$In\ case\ we\ have\ design\ options\ like\ (L1, T1), (L2, T2), (L3, T3)$$
$$T = min\ (T1, T2, T3)$$
$$L = L1 + L2 + L3$$

e. Once we do that, we must do a redundancy check. This can be done using the following method,

$$Suppose\ we\ have\ two\ design\ options\ (L1, T1)\ and\ (L2, T2)$$
$$If\ L1 >= L2\ and\ T1 <= T2$$
$$Then, (L1, T1)\ is\ redundant.$$

f. When buffer is inserted, we use different set of formulas to calculate the time.

When we insert buffer,

L = Cbuff

Tbuff = Tmin – Rbuff * L

Using these steps mentioned above, the calculations can be done to determine the various design options and buffer insertion points.

First, I will show the different points and the name of the points that I have used below

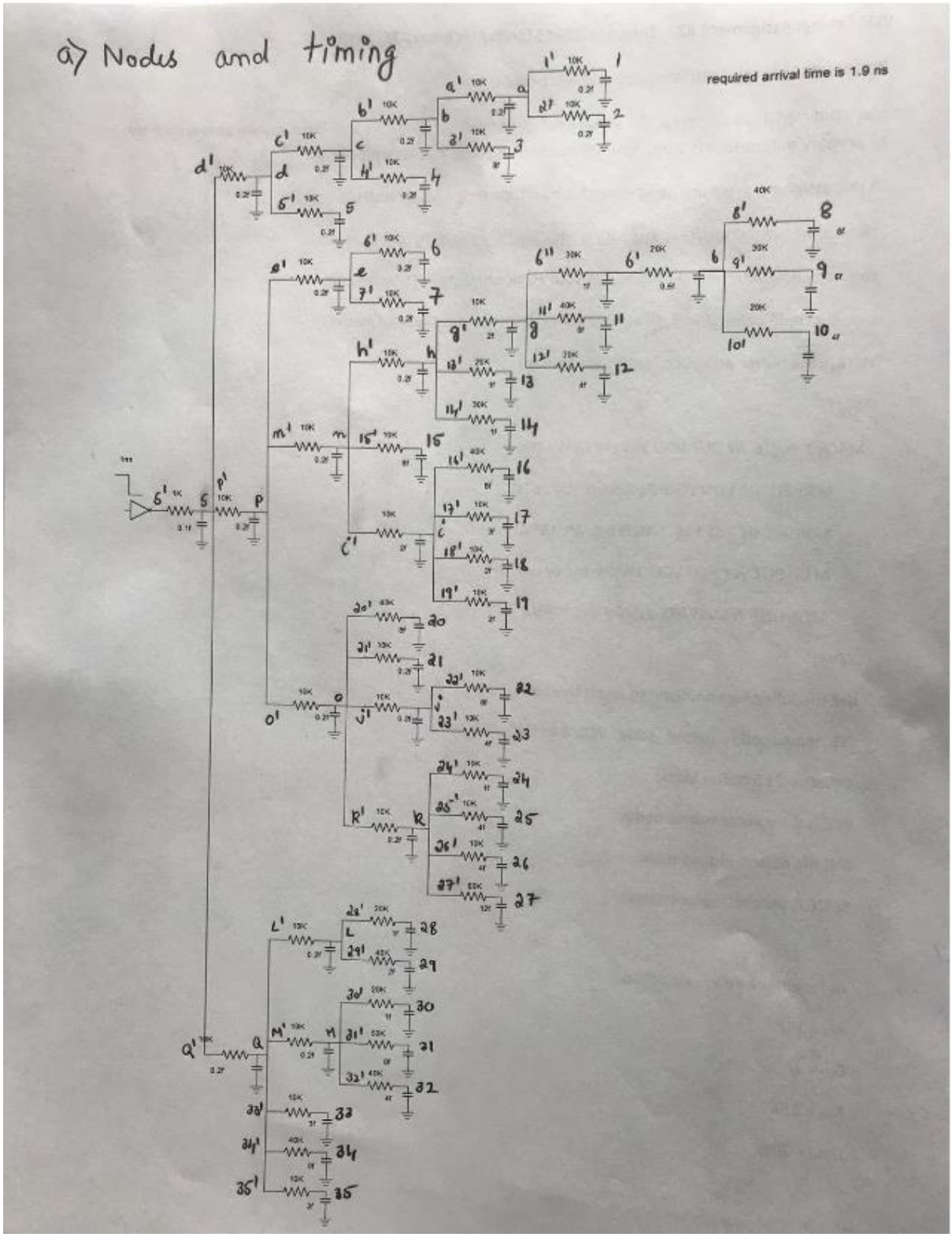


Figure 1

The Values are Tabulated as shown. The first set shown here shows the final edges and their values when traversed. These values are calculated at nodes with

Node	Load (fF)	Time (ns)
1	0.2	1.9
2	0.2	1.9
3	8	1.9
4	0.2	1.9
5	0.2	1.9
6	0.2	1.9
7	0.2	1.9
8	8	1.9
9	6	1.9
10	4	1.9
11	6	1.9
12	4	1.9
13	1	1.9
14	1	1.9
15	8	1.9
16	8	1.9
17	3	1.9
18	2	1.9
19	2	1.9
20	8	1.9
21	0.2	1.9
22	8	1.9
23	4	1.9
24	1	1.9
25	4	1.9
26	4	1.9
27	12	1.9
28	1	1.9
29	2	1.9
30	1	1.9
31	8	1.9
32	4	1.9
33	3	1.9
34	8	1.9
35	2	1.9

Node	Load (fF)	Time (ns)
1’	0.2	1.898
2’	0.2	1.898
3’	8	1.82
4’	0.2	1.898
5’	0.2	1.898
6’	0.2	1.898
7’	0.2	1.898
8’	8	1.58
9’	6	1.72
10’	4	1.82
11’	6	1.66
12’	4	1.78
13’	1	1.88
14’	1	1.87
15’	8	1.82
16’	8	1.58
17’	3	1.87
18’	2	1.88
19’	2	1.88
20’	8	1.58
21’	0.2	1.898
22’	8	1.82
23’	4	1.86
24’	1	1.89
25’	4	1.86
26’	4	1.86
27’	12	1.18
28’	1	1.88
29’	2	1.82
30’	1	1.88
31’	8	1.5
32’	4	1.74
33’	3	1.87
34’	8	1.58
35’	2	1.88

Vinay Manjunath Prabhu
825088894

The values below shown is of the fan outs and the redundant options

Node	Load (fF)	Time (ns)	Redundancy
a	0.4	1.898	
	2	1.8778	Redundant
a'	0.6	1.892	
b	8.6	1.82	
	2	1.795	
b'	8.8	1.734	Redundant
	2.2	1.773	
c	2.4	1.773	
	2	1.752	
c'	2.6	1.747	
	2.2	1.73	
d	2.4	1.73	
	2.8	1.747	
	2	1.726	
d'	2.6	1.704	Redundant
	3	1.717	
	2.2	1.704	
e	0.4	1.898	
	2	1.877	Redundant
e'	0.6	1.892	
f	18	1.58	
	2	1.551	
f'	18.5	1.21	Redundant
	2.5	1.501	
f''	3.5	1.396	
g	13.5	1.396	
	2	1.369	
g'	15.5	1.241	Redundant
	4	1.329	
h	6	1.329	
	2	1.306	
h'	6.2	1.267	Redundant
	2.2	1.284	
i	15	1.58	
	2	1.553	
I'	17	1.41	Redundant
	2.2	1.531	
n	12.4	1.284	
	2	1.258	
n'	12.6	1.158	Redundant
	2.2	1.236	

j	12	1.82	
	2	1.794	
j'	12.2	1.698	Redundant
	2.2	1.772	
k	21	1.18	
	2	1.15	
k'	21.2	0.968	
	2.2	1.128	
o	12.6	1.128	
	2	1.102	
o'	12.8	1	Redundant
	2.2	1.08	
p	5	1.08	
	2	1.058	
p'	5.2	1.028	Redundant
	2.2	1.006	
l	3	1.82	
	2	1.799	
l'	3.2	1.788	
	2.2	1.777	
m	13	1.5	
	2	1.473	
m'	13.2	1.368	Redundant
	2.2	1.451	
q	17.4	1.451	
	2	1.422	
q'	17.6	1.275	Redundant
	2.2	1.4	
s	6.6	1.006	
	2	0.983	
s'	6.7	0.993	
	2.1	0.98	

Note: The design in Blue is for the buffer. Redundant options are removed at each stage.

Based on the calculation done, the buffer was inserted at points D, C, B, H, G, F, N, I, O, J, K, L, M, Q, P, S. Total of 16 Buffers was used for this problem. To compare the performance before and after buffer insertion, we need to calculate the Elmore delay with and without buffer. This will be shown below.

Elmore Delay Calculation without buffer: For the Elmore delay calculation, we have the following nodes as shown,

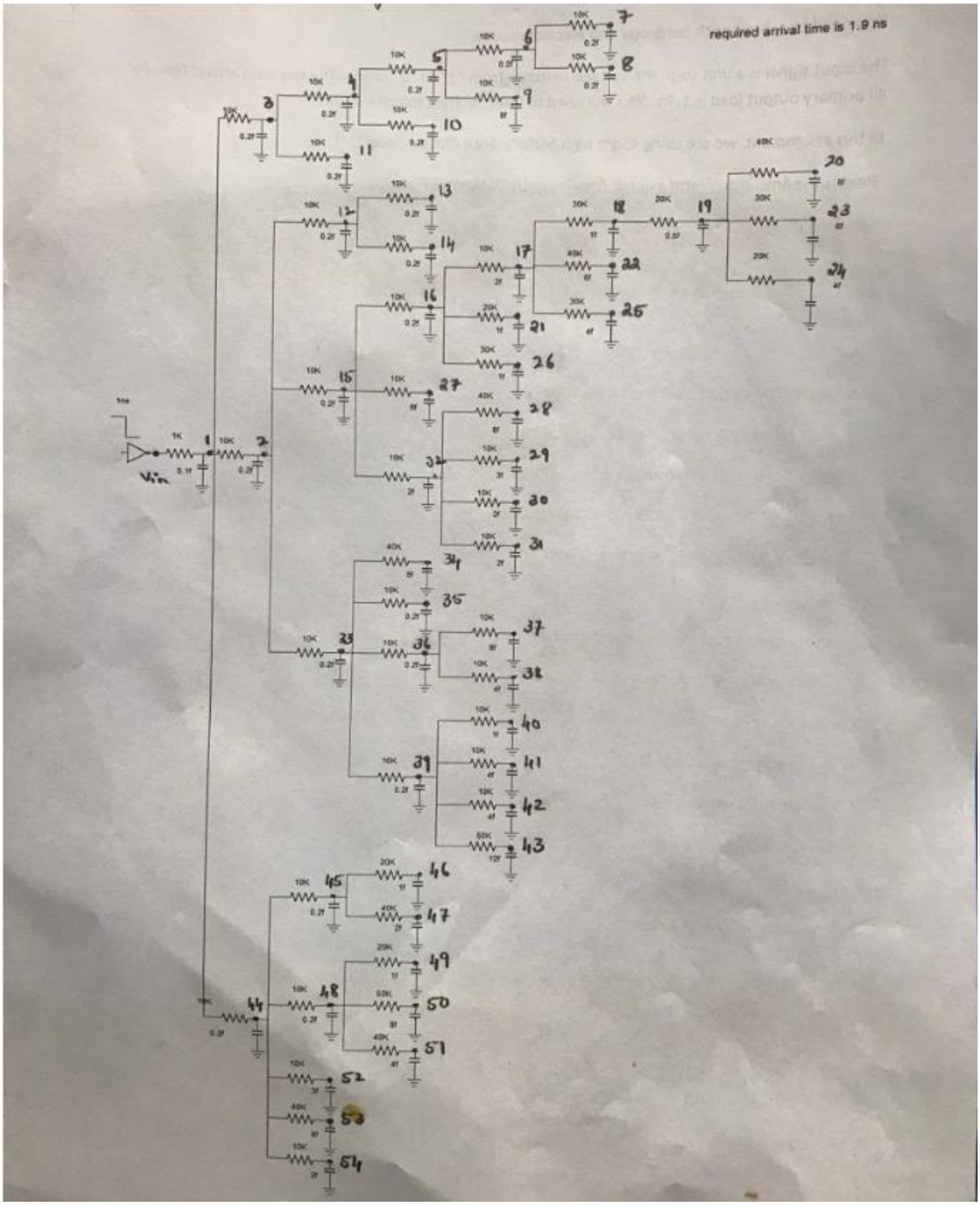


Figure 2

T_7 = (2663.8 + 96 + 92 + 88 + 6 + 2) * 10^{-12} = 2.95ns
T_8 = (2663.8 + 96 + 92 + 88 + 6 + 2) * 10^{-12} = 2.95ns
T_9 = (2663.8 + 96 + 92 + 88 + 80) * 10^{-12} = 3.019ns
T_{10} = (2663.8 + 96 + 92 + 2) * 10^{-12} = 2.853ns
T_{11} = (2663.8 + 96 + 2) * 10^{-12} = 2.761ns
T_{13} = (2663.8 + 1009 + 6 + 2) * 10^{-12} = 3.68ns
T_{14} = (2663.8 + 1009 + 6 + 2) * 10^{-12} = 3.68ns
T_{20} = (2663.8 + 1009 + 589 + 337 + 315 + 585 + 370 + 320) * 10^{-12} = 6.18ns
T_{23} = (2663.8 + 1009 + 589 + 337 + 315 + 585 + 370 + 180) * 10^{-12} = 6.05ns
T_{24} = (2663.8 + 1009 + 589 + 337 + 315 + 585 + 370 + 80) * 10^{-12} = 5.95ns
T_{22} = (2663.8 + 1009 + 589 + 337 + 315 + 240) * 10^{-12} = 5.15ns
T_{25} = (2663.8 + 1009 + 589 + 337 + 315 + 120) * 10^{-12} = 5.03ns
T_{21} = (2663.8 + 1009 + 589 + 337 + 20) * 10^{-12} = 4.62ns
T_{26} = (2663.8 + 1009 + 589 + 337 + 30) * 10^{-12} = 4.62ns
T_{27} = (2663.8 + 1009 + 589 + 80) * 10^{-12} = 4.34ns
T_{28} = (2663.8 + 1009 + 589 + 170 + 320) * 10^{-12} = 4.75ns
T_{29} = (2663.8 + 1009 + 589 + 170 + 30) * 10^{-12} = 4.46ns
T_{30} = (2663.8 + 1009 + 589 + 170 + 20) * 10^{-12} = 4.45ns
T_{31} = (2663.8 + 1009 + 589 + 170 + 20) * 10^{-12} = 4.45ns
T_{34} = (2663.8 + 1009 + 418 + 320) * 10^{-12} = 4.41ns
T_{35} = (2663.8 + 1009 + 418 + 2) * 10^{-12} = 4.09ns
T_{37} = (2663.8 + 1009 + 418 + 122 + 80) * 10^{-12} = 4.29ns
T_{38} = (2663.8 + 1009 + 418 + 122 + 40) * 10^{-12} = 4.25ns
T_{40} = (2663.8 + 1009 + 418 + 212 + 10) * 10^{-12} = 4.31ns
T_{41} = (2663.8 + 1009 + 418 + 212 + 40) * 10^{-12} = 4.34ns
T_{42} = (2663.8 + 1009 + 418 + 212 + 40) * 10^{-12} = 4.34ns
T_{43} = (2663.8 + 1009 + 418 + 212 + 720) * 10^{-12} = 5.02ns
T_{46} = (2663.8 + 296 + 320 + 20) * 10^{-12} = 3.29ns
T_{47} = (2663.8 + 296 + 320 + 80) * 10^{-12} = 3.35ns
T_{49} = (2663.8 + 296 + 132 + 20) * 10^{-12} = 3.11ns
T_{50}= (2663.8 + 296 + 132 + 400) * 10^{-12} = 3.49ns
T_{51} = (2663.8 + 296 + 132 + 160) * 10^{-12} = 3.25ns
T_{52} = (2663.8 + 296 + 30) * 10^{-12} = 2.98ns
T_{53} = (2663.8 + 296 + 30) * 10^{-12} = 3.27ns
T_{54} = (2663.8 + 296 + 30) * 10^{-12} = 2.97ns

As we can see from the above values, the Elmore delay is more than 1.9ns. This means that the arrival time will also be more than the expected arrival time. To reduce this, we insert buffers. This is shown below.

Elmore Delay Calculation without buffer: For the Elmore delay calculation, we have the following nodes as shown,

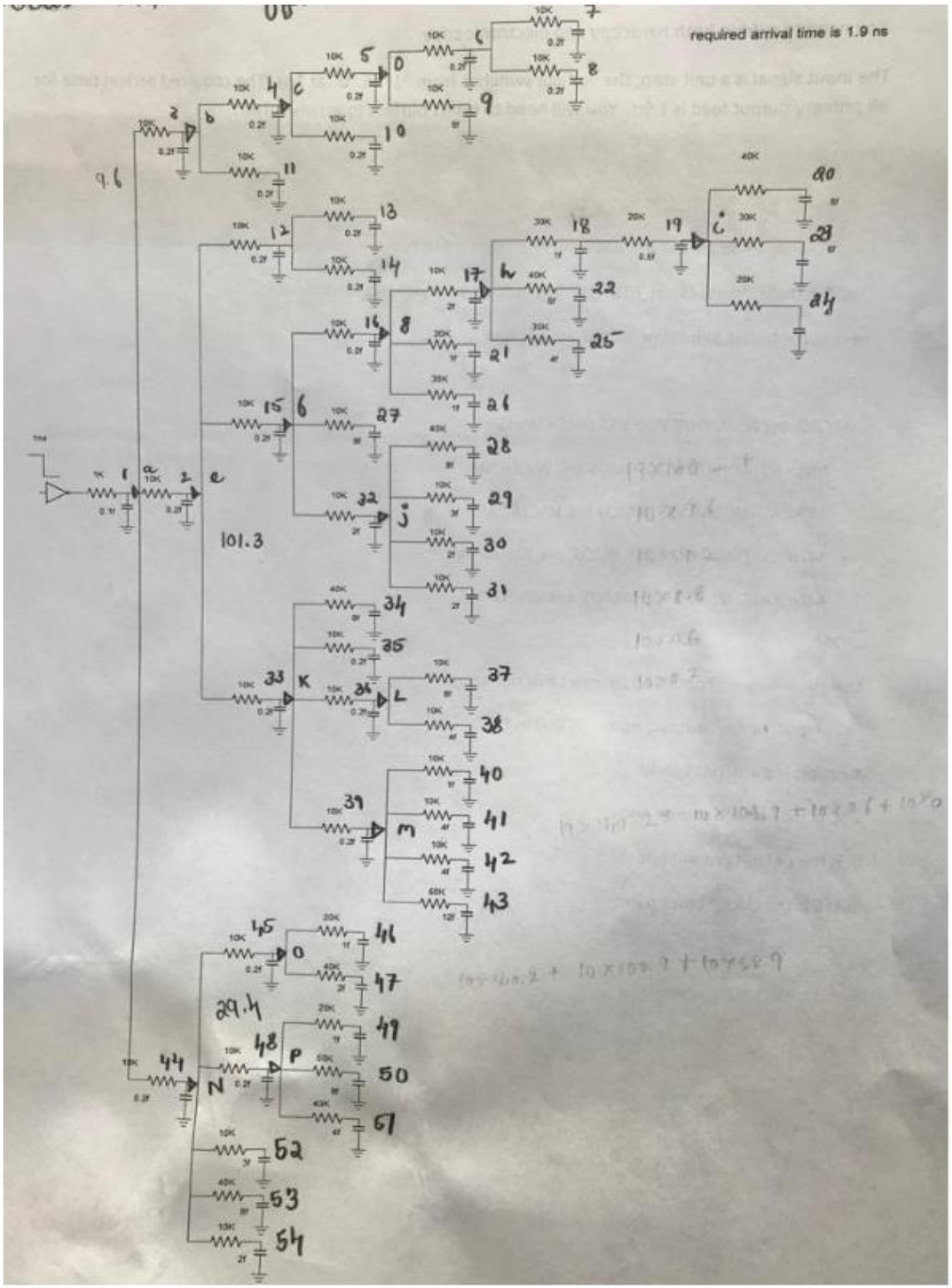


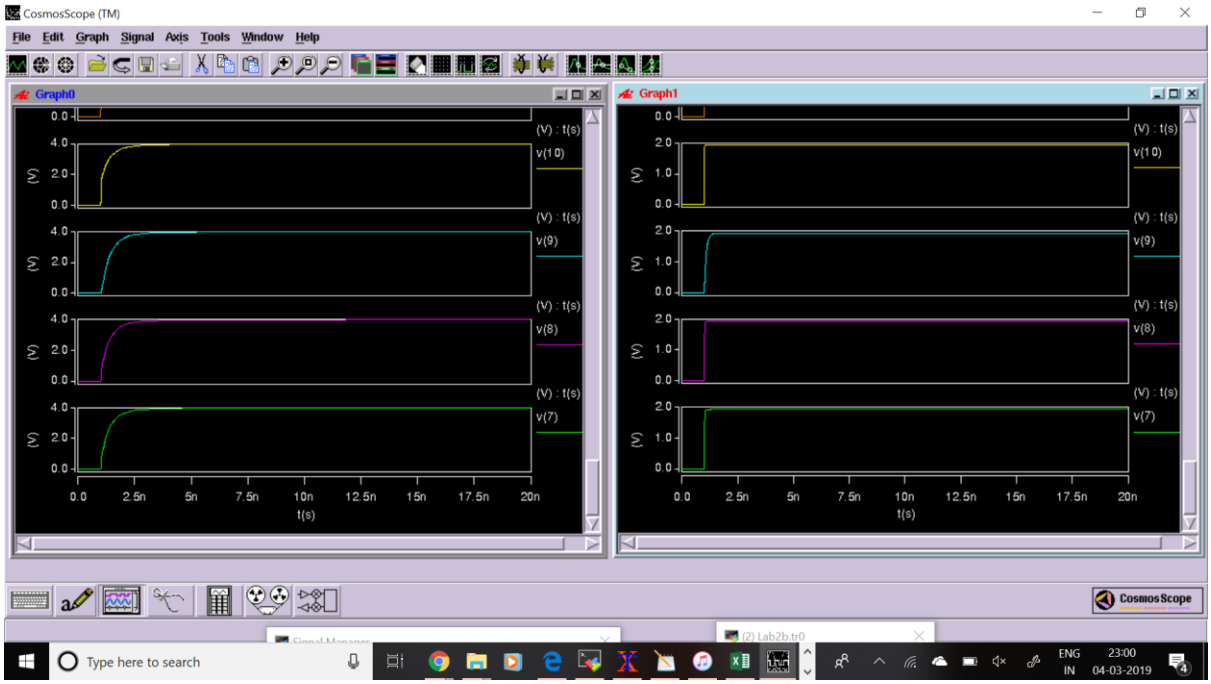
Figure 3

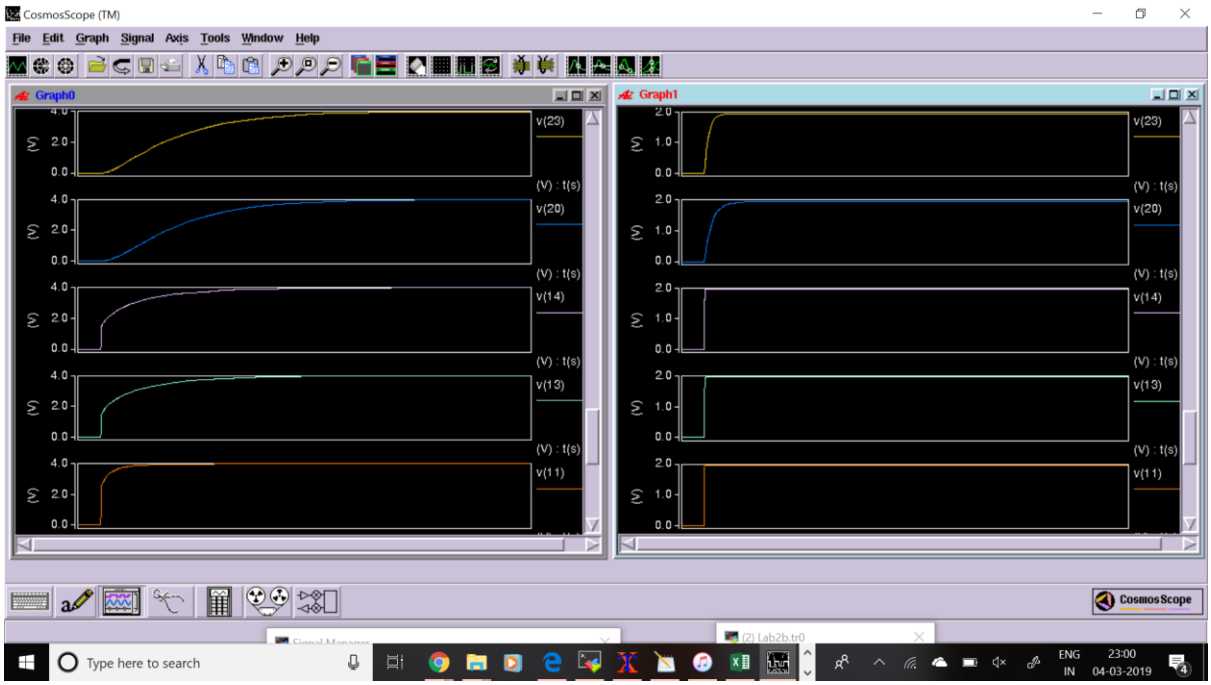
$T_7 = (39.9 + 22 + 22 + 22 + 6 + 2 + 80 + 3.3 + 1.2 + 1.2 + 4.2) * 10^{-12} = 0.203ns$
$T_8 = (39.9 + 22 + 22 + 22 + 6 + 2 + 80 + 3.3 + 1.2 + 1.2 + 4.2) * 10^{-12} = 0.203ns$
$T_9 = (39.9 + 22 + 22 + 22 + 8 + 80 + 3.3 + 1.2 + 1.2 + 4.2) * 10^{-12} = 0.203ns$
$T_{10} = (39.9 + 22 + 22 + 2 + 60 + 3.3 + 1.2 + 1.2) * 10^{-12} = 0.151ns$
$T_{11} = (39.9 + 22 + 2 + 40 + 3.3 + 1.2) * 10^{-12} = 0.108ns$
$T_{13} = (39.9 + 22 + 6 + 2 + 40 + 3.3 + 2.5) * 10^{-12} = 0.115ns$
$T_{14} = (39.9 + 22 + 6 + 2 + 40 + 3.3 + 2.5) * 10^{-12} = 0.115ns$
$T_{20} = (39.9 + 22 + 22 + 22+ 40 + 105 + 50 + 320 + 3.3 + 2.5 + 7.1 + 2 + 6.75 + 9) * 10^{-12} = 0.771ns$
$T_{23} = (39.9 + 22 + 22 + 22+ 40 + 105 + 50 + 180 + 3.3 + 2.5 + 7.1 + 2 + 6.75 + 9) * 10^{-12} = 0.631ns$
$T_{24} = (39.9 + 22 + 22 + 22+ 40 + 105 + 50 + 80 + 3.3 + 2.5 + 7.1 + 2 + 6.75 + 9) * 10^{-12} = 0.531ns$
$T_{22} = (39.9 + 22 + 22 + 22+ 40 + 240 + 3.3 + 2.5 + 7.1 + 2 + 6.75) * 10^{-12} = 0.507ns$
$T_{25} = (39.9 + 22 + 22 + 22+ 40 + 120 + 3.3 + 2.5 + 7.1 + 2 + 6.75) * 10^{-12} = 0.387ns$
$T_{21} = (39.9 + 22 + 22 + 22+ 20 + 3.3 + 2.5 + 7.1 + 2 + 120) * 10^{-12} = 0.220ns$
$T_{26} = (39.9 + 22 + 22 + 22+ 30 + 3.3 + 2.5 + 7.1 + 2 + 100) * 10^{-12} = 0.230ns$
$T_{27} = (39.9 + 22 + 22 + 80 + 3.3 + 2.5 + 7.1 + 100) * 10^{-12} = 0.236ns$
$T_{28} = (39.9 + 22 + 22 + 40 + 320 + 3.3 + 2.5 + 7.1 + 80) * 10^{-12} = 0.536ns$
$T_{29} = (39.9 + 22 + 22 + 40 + 30 + 3.3 + 2.5 + 7.1 + 80) * 10^{-12} = 0.246ns$
$T_{30} = (39.9 + 22 + 22 + 40 + 20 + 3.3 + 2.5 + 7.1+ 80) * 10^{-12} = 0.236ns$
$T_{31} = (39.9 + 22 + 22 + 40 + 20 + 3.3 + 2.5 + 7.1 + 80) * 10^{-12} = 0.236ns$
$T_{34} = (39.9 + 22 + 22 + 320 + 3.3 + 2.5 + 6.3 + 60) * 10^{-12} = 0.476ns$
$T_{35} = (39.9 + 22 + 22 + 2 + 3.3 + 2.5 + 6.3 + 80) * 10^{-12} = 0.158ns$
$T_{37} = (39.9 + 22 + 22 + 22 + 80 + 3.3 + 2.5 + 6.3 + 6 + 80) * 10^{-12} = 0.284ns$
$T_{38} = (39.9 + 22 + 22 + 22 + 40 + 3.3 + 2.5 + 6.3 + 6 + 80) * 10^{-12} = 0.244ns$
$T_{40} = (39.9 + 22 + 22 + 22 + 10 + 3.3 + 2.5 + 6.3 + 10.5+ 80) * 10^{-12} = 0.218ns$
$T_{41} = (39.9 + 22 + 22 + 22 + 40 + 3.3 + 2.5 + 6.3 + 10.5 + 80) * 10^{-12} = 0.248ns$
$T_{42} = (39.9 + 22 + 22 + 22 + 40 + 3.3 + 2.5 + 6.3 + 10.5 + 80) * 10^{-12} = 0.248ns$
$T_{43} = (39.9 + 22 + 22 + 22 + 720 + 3.3 + 2.5 + 6.3 + 10.5 + 80) * 10^{-12} = 0.928ns$
$T_{46} = (39.9 + 22 + 22 + 80 + 3.3 + 8.7 + 1.5 + 60) * 10^{-12} = 0.177ns$
$T_{47} = (39.9 + 22 + 22 + 20 + 3.3 + 8.7 + 6.5 + 60) * 10^{-12} = 0.237ns$
$T_{49} = (39.9 + 22 + 22 + 80 + 3.3 + 8.7 + 6.5 + 60) * 10^{-12} = 0.182ns$
$T_{50} = (39.9 + 22 + 22 + 400 + 3.3 + 8.7 + 6.5 + 60) * 10^{-12} = 0.562ns$
$T_{51} = (39.9 + 22 + 22 + 160 + 3.3 + 8.7 + 60) * 10^{-12} = 0.322ns$
$T_{52} = (39.9 + 22 + 30 + 3.3 + 8.7 + 40) * 10^{-12} = 0.143ns$
$T_{53} = (39.9 + 22 + 320 + 3.3 + 8.7 + 40) * 10^{-12} = 0.433ns$
$T_{54} = (39.9 + 22 + 20 + 3.3 + 8.7 + 40) * 10^{-12} = 0.133ns$

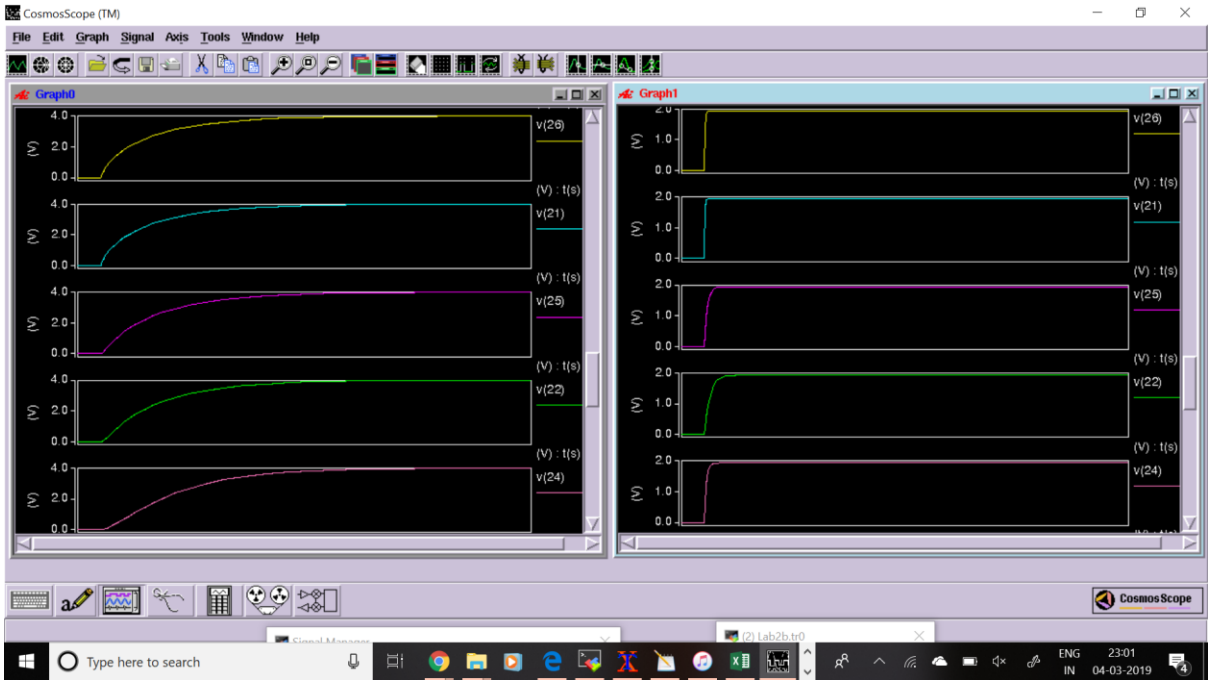
As we can see from the above values, when buffer gets inserted at the determined points, the Elmore delay drops down by a large value. The expected arrival can be achieved using the buffers at these positions determined.

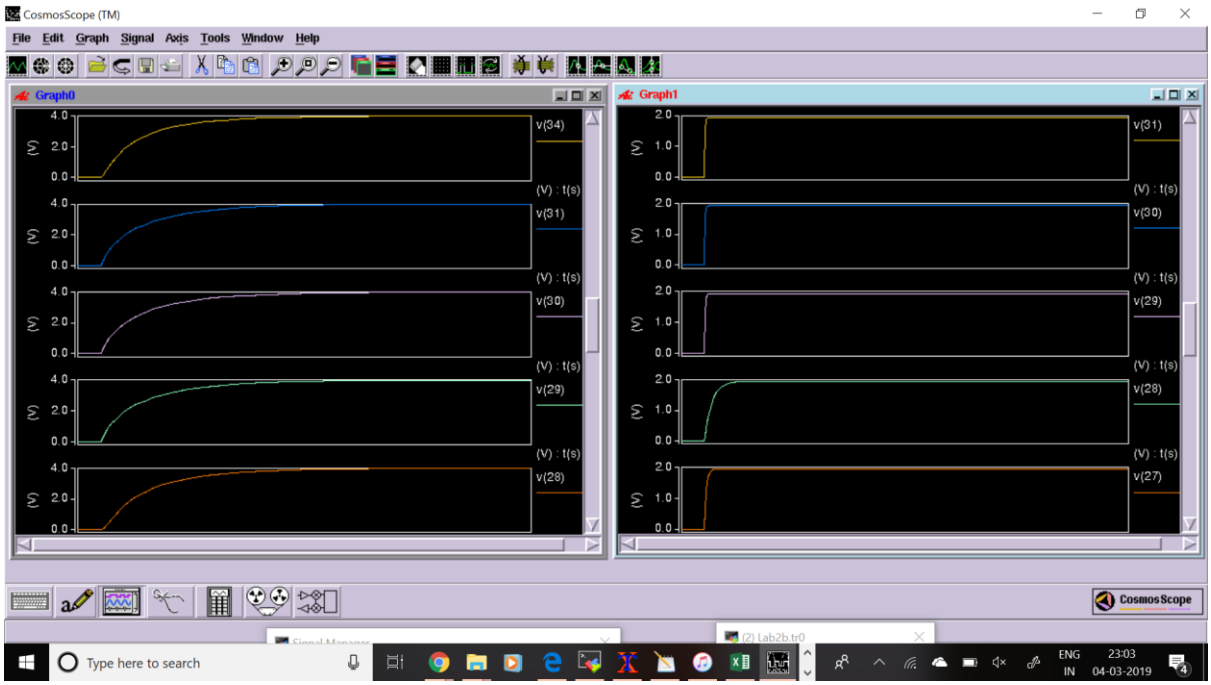
Waveform: On the left is the one without buffer. On the right is the one with buffer.

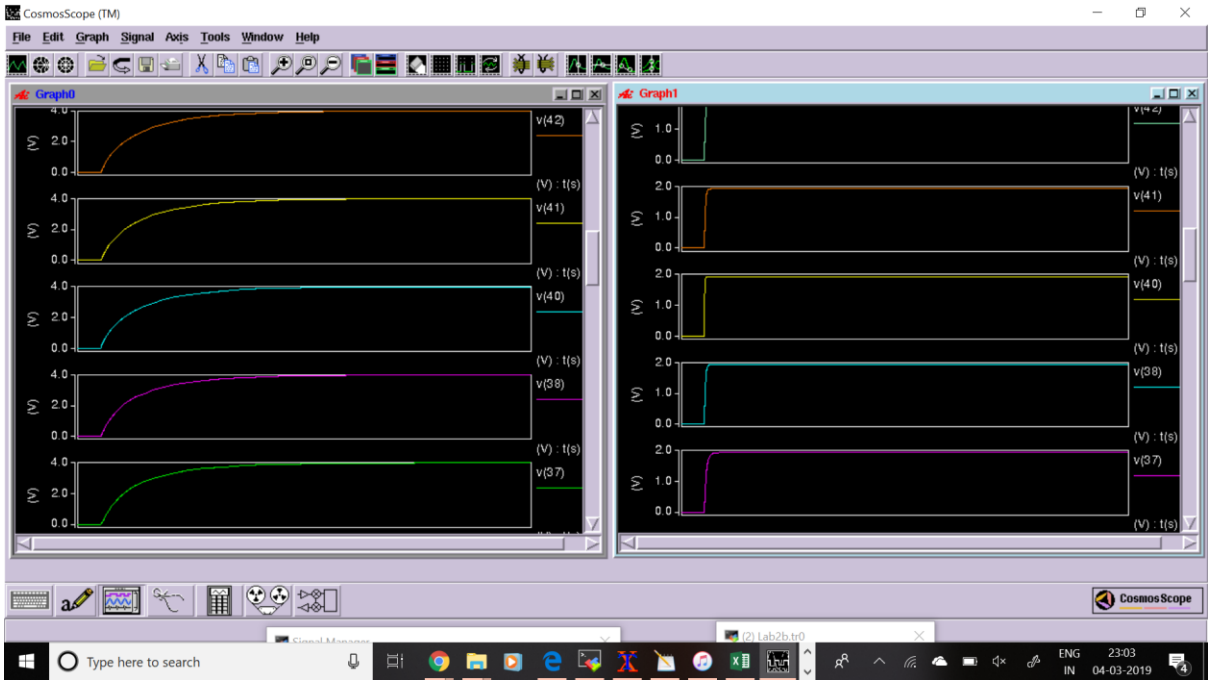
Vinay Manjunath Prabhu
825088894

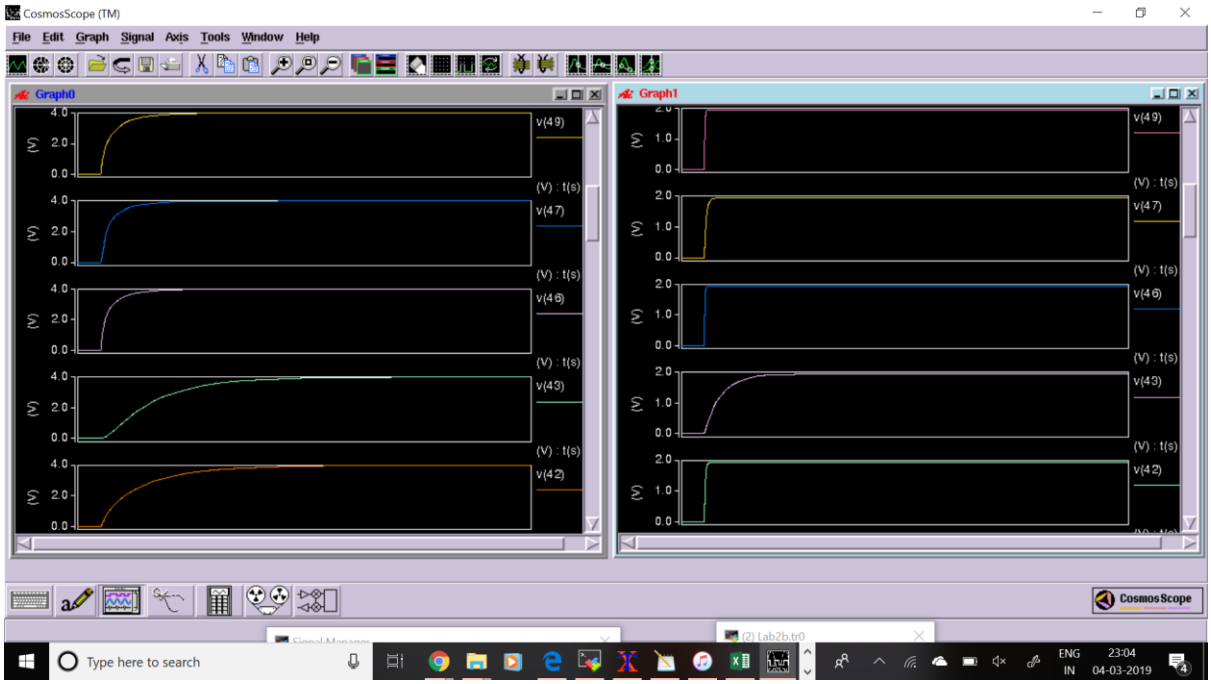


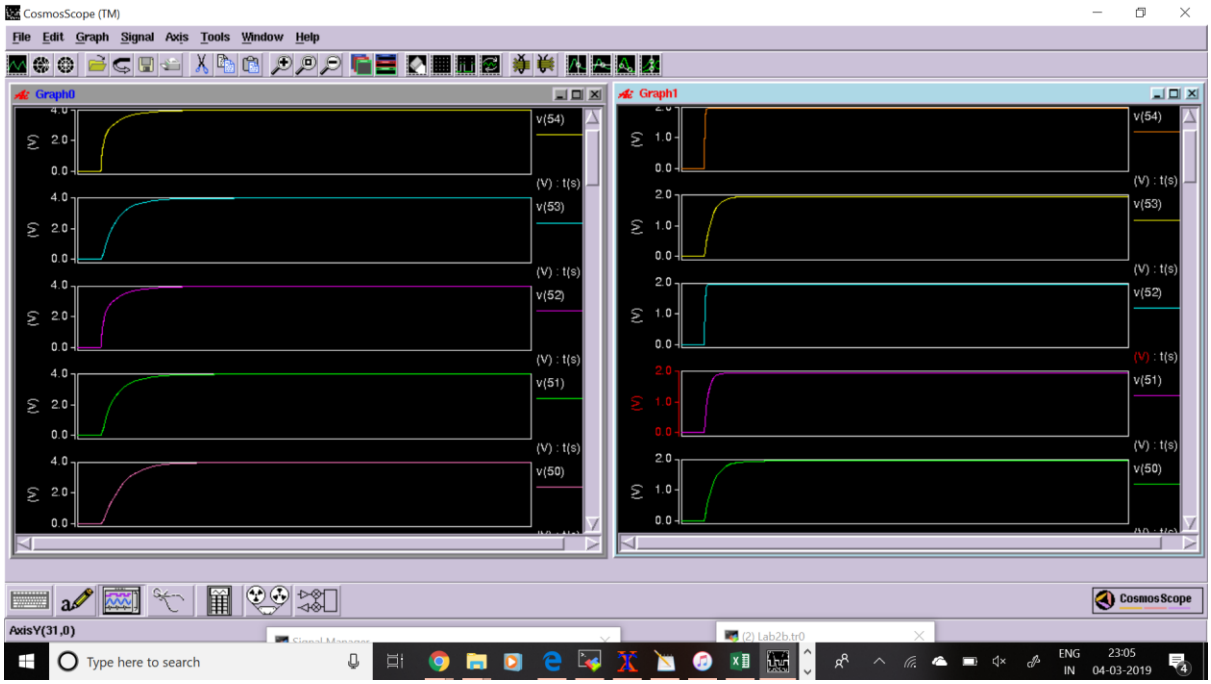



CosmosScope (TM)
File Edit Graph Signal Axis Tools Window Help
Graph0
Graph1
v(26)
v(21)
v(25)
v(22)
v(24)
(V) : t(s)
CosmosScope
Type here to search
ENG
IN
23:01
04-03-2019


CosmosScope (TM)
File Edit Graph Signal Axis Tools Window Help
Graph0
Graph1
v(34)
v(31)
v(30)
v(29)
v(28)
v(27)
(V) : t(s)
CosmosScope
Type here to search
ENG
IN
23:03
04-03-2019


CosmosScope (TM)
File Edit Graph Signal Axis Tools Window Help
Graph0
Graph1
v(42)
v(41)
v(40)
v(38)
v(37)
(V) : t(s)
CosmosScope
Type here to search
ENG
IN
23:03
04-03-2019


CosmosScope (TM)
File Edit Graph Signal Axis Tools Window Help
Graph0
Graph1
v(49)
v(47)
v(46)
v(43)
v(42)
(V) : t(s)
CosmosScope
Type here to search
ENG
IN
23:04
04-03-2019



The comparison clearly shows the reduction of delay. The buffer adds a small value to the delay, but in an overall view, brings down the delay by a large value. Hence adding buffers at the determined points in the design will help make the design more efficient.

APPENDIX

1. **HSPICE Code for design without Buffer** (The Nodes are taken with reference to figure 2 in the above report)

```
*This file will describe the HSPICE of Lab 2
*HSPICE code for without buffer

*Define the voltage source that is unit step function
vs Vin gnd PWL(0 0V 1n 0V 1.01n 4V 10.01n 4V 20n 4v 50n 4v)

*Define the design
.option post

*Defining R in the design

R1 Vin 1 1K
R2 1 2 10k
R3 1 3 10k
R4 3 4 10k
R5 4 5 10k
R6 5 6 10k
R7 6 7 10k
R8 6 8 10k
R9 5 9 10k
R10 4 10 10k
R11 3 11 10k
R12 2 12 10k
R13 12 13 10k
R14 12 14 10k
R15 2 15 10k
R16 15 16 10k
R17 16 17 10k
R18 17 18 30k
R19 18 19 20k
R20 19 20 40k
R21 16 21 20k
R22 17 22 40k
R23 19 23 30k
R24 19 24 20k
R25 17 25 30k
R26 16 26 30k
R27 15 27 10k
R28 32 28 40k
R29 32 29 10k
R30 32 30 10k
R31 32 31 10k
R32 15 32 10k
```

R33 2 33 10k
R34 33 34 40k
R35 33 35 10k
R36 33 36 10k
R37 36 37 10k
R38 36 38 10k
R39 33 39 10k
R40 39 40 10k
R41 39 41 10k
R42 39 42 10k
R43 39 43 60k
R44 1 44 10k
R45 44 45 10k
R46 45 46 20k
R47 45 47 40k
R48 44 48 10k
R49 48 49 20k
R50 48 50 50k
R51 48 51 40k
R52 44 52 10k
R53 44 53 40k
R54 44 54 10k

**Defining C in the design*

C1 1 gnd 0.1fF
C2 2 gnd 0.2fF
C3 3 gnd 0.2fF
C4 4 gnd 0.2fF
C5 5 gnd 0.2fF
C6 6 gnd 0.2fF
C7 7 gnd 0.2fF
C8 8 gnd 0.2fF
C9 9 gnd 8fF
C10 10 gnd 0.2fF
C11 11 gnd 0.2fF
C12 12 gnd 0.2fF
C13 13 gnd 0.2fF
C14 14 gnd 0.2fF
C15 15 gnd 0.2fF
C16 16 gnd 0.2fF
C17 17 gnd 2fF
C18 18 gnd 1fF
C19 19 gnd 0.5fF
C20 20 gnd 8fF
C21 21 gnd 1fF
C22 22 gnd 6fF
C23 23 gnd 6fF

```
C24 24 gnd 4fF
C25 25 gnd 4fF
C26 26 gnd 1fF
C27 27 gnd 8fF
C28 28 gnd 8fF
C29 29 gnd 3fF
C30 30 gnd 2fF
C31 31 gnd 2fF
C32 32 gnd 2fF
C33 33 gnd 0.2fF
C34 34 gnd 8fF
C35 35 gnd 0.2fF
C36 36 gnd 0.2fF
C37 37 gnd 8fF
C38 38 gnd 4fF
C39 39 gnd 0.2fF
C40 40 gnd 1fF
C41 41 gnd 4fF
C42 42 gnd 4fF
C43 43 gnd 12fF
C44 44 gnd 0.2fF
C45 45 gnd 0.2fF
C46 46 gnd 1fF
C47 47 gnd 2fF
C48 48 gnd 0.2fF
C49 49 gnd 1fF
C50 50 gnd 8fF
C51 51 gnd 4fF
C52 52 gnd 3fF
C53 53 gnd 8fF
C54 54 gnd 2fF


.tran 0.01n 20n
.print tran V(vs) V(7)
.print tran V(vs) V(8)
.print tran V(vs) V(9)
.print tran V(vs) V(10)
.print tran V(vs) V(11)
.print tran V(vs) V(13)
.print tran V(vs) V(14)
.print tran V(vs) V(20)
.print tran V(vs) V(23)
.print tran V(vs) V(24)
.print tran V(vs) V(22)
.print tran V(vs) V(25)
.print tran V(vs) V(21)
.print tran V(vs) V(26)
```

```
.print tran V(vs) V(27)
.print tran V(vs) V(28)
.print tran V(vs) V(29)
.print tran V(vs) V(30)
.print tran V(vs) V(31)
.print tran V(vs) V(34)
.print tran V(vs) V(35)
.print tran V(vs) V(37)
.print tran V(vs) V(38)
.print tran V(vs) V(40)
.print tran V(vs) V(41)
.print tran V(vs) V(42)
.print tran V(vs) V(43)
.print tran V(vs) V(46)
.print tran V(vs) V(47)
.print tran V(vs) V(49)
.print tran V(vs) V(50)
.print tran V(vs) V(51)
.print tran V(vs) V(52)
.print tran V(vs) V(53)
.print tran V(vs) V(54)
.option post

.end
```

2. **HSPICE Code for design with Buffer** (The Nodes are taken with reference to figure 2 in the above report)

```
*This file will describe the HSPICE of Lab 2
*HSPICE code for without buffer
.include ./cmos45nm_PTM_HP
*Define the voltage source that is unit step function
vs Vin gnd PWL(0 0V 1n 0V 1.01n 4V 10.01n 4V 20n 4v 50n 4v)

*BUFFER Macro
.MACRO buf16 IN OUT VDD VSS L=45n W=45n
M0p N1 IN VDD VDD PMOS L=L W='16*W'
M0n N1 IN VSS VSS NMOS L=L W='16*W'
M1p OUT N1 VDD VDD PMOS L=L W='32*W'
M1n OUT N1 VSS VSS NMOS L=L W='32*W'

.EOM

*Define the design
.option post
```

**Defining R in the design*

R1 Vin 1 1K
R2 a 2 10k
R3 a 3 10k
R4 b 4 10k
R5 c 5 10k
R6 d 6 10k
R7 6 7 10k
R8 6 8 10k
R9 d 9 10k
R10 c 10 10k
R11 b 11 10k
R12 e 12 10k
R13 12 13 10k
R14 12 14 10k
R15 e 15 10k
R16 f 16 10k
R17 g 17 10k
R18 h 18 30k
R19 18 19 20k
R20 i 20 40k
R21 g 21 20k
R22 h 22 40k
R23 i 23 30k
R24 i 24 20k
R25 h 25 30k
R26 g 26 30k
R27 f 27 10k
R28 j 28 40k
R29 j 29 10k
R30 j 30 10k
R31 j 31 10k
R32 f 32 10k
R33 e 33 10k
R34 k 34 40k
R35 k 35 10k
R36 k 36 10k
R37 l 37 10k
R38 l 38 10k
R39 k 39 10k
R40 m 40 10k

R41 m 41 10k
R42 m 42 10k
R43 m 43 60k
R44 a 44 10k
R45 n 45 10k
R46 o 46 20k
R47 o 47 40k
R48 n 48 10k
R49 p 49 20k
R50 p 50 50k
R51 p 51 40k
R52 n 52 10k
R53 n 53 40k
R54 n 54 10k

**Defining C in the design*

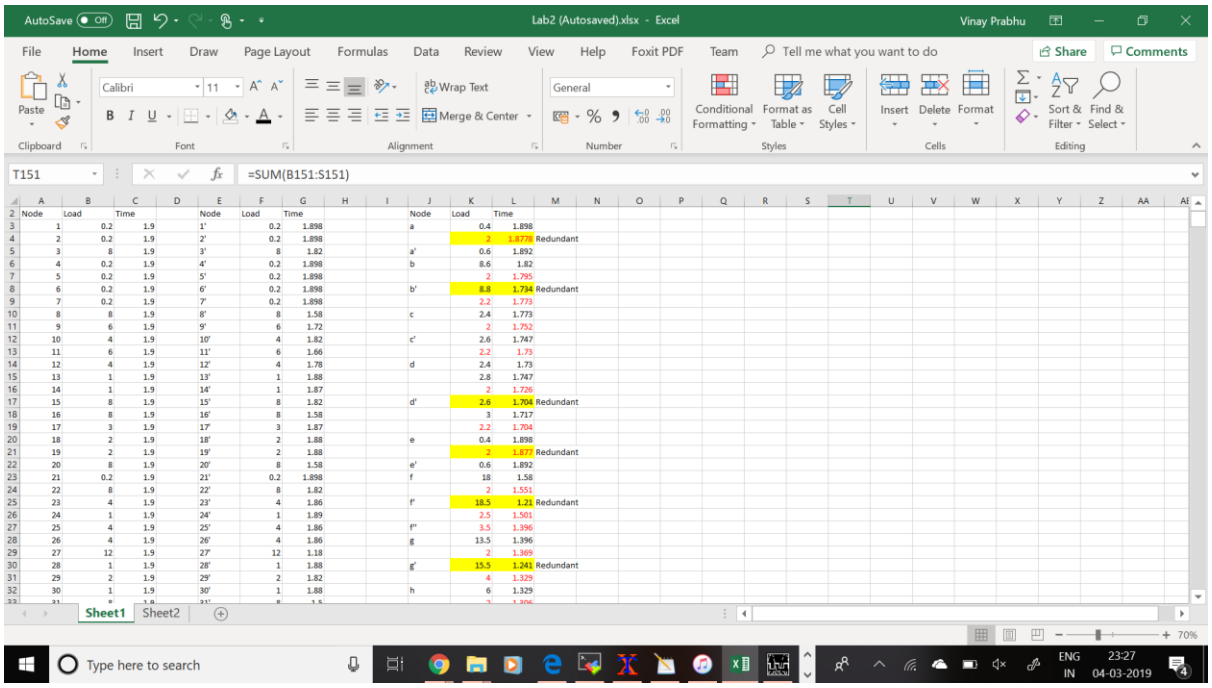
C1 1 gnd 0.1fF
C2 2 gnd 0.2fF
C3 3 gnd 0.2fF
C4 4 gnd 0.2fF
C5 5 gnd 0.2fF
C6 6 gnd 0.2fF
C7 7 gnd 0.2fF
C8 8 gnd 0.2fF
C9 9 gnd 8fF
C10 10 gnd 0.2fF
C11 11 gnd 0.2fF
C12 12 gnd 0.2fF
C13 13 gnd 0.2fF
C14 14 gnd 0.2fF
C15 15 gnd 0.2fF
C16 16 gnd 0.2fF
C17 17 gnd 2fF
C18 18 gnd 1fF
C19 19 gnd 0.5fF
C20 20 gnd 8fF
C21 21 gnd 1fF
C22 22 gnd 6fF
C23 23 gnd 6fF
C24 24 gnd 4fF
C25 25 gnd 4fF
C26 26 gnd 1fF
C27 27 gnd 8fF

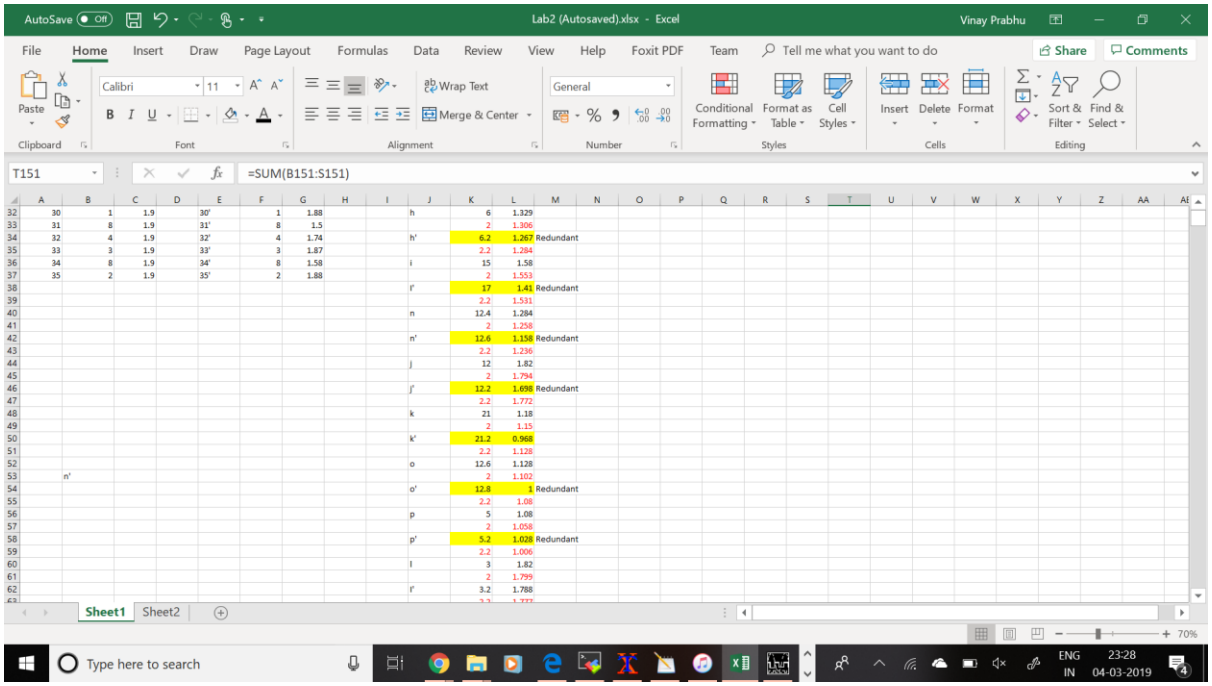
C28 28 gnd 8fF
C29 29 gnd 3fF
C30 30 gnd 2fF
C31 31 gnd 2fF
C32 32 gnd 2fF
C33 33 gnd 0.2fF
C34 34 gnd 8fF
C35 35 gnd 0.2fF
C36 36 gnd 0.2fF
C37 37 gnd 8fF
C38 38 gnd 4fF
C39 39 gnd 0.2fF
C40 40 gnd 1fF
C41 41 gnd 4fF
C42 42 gnd 4fF
C43 43 gnd 12fF
C44 44 gnd 0.2fF
C45 45 gnd 0.2fF
C46 46 gnd 1fF
C47 47 gnd 2fF
C48 48 gnd 0.2fF
C49 49 gnd 1fF
C50 50 gnd 8fF
C51 51 gnd 4fF
C52 52 gnd 3fF
C53 53 gnd 8fF
C54 54 gnd 2fF

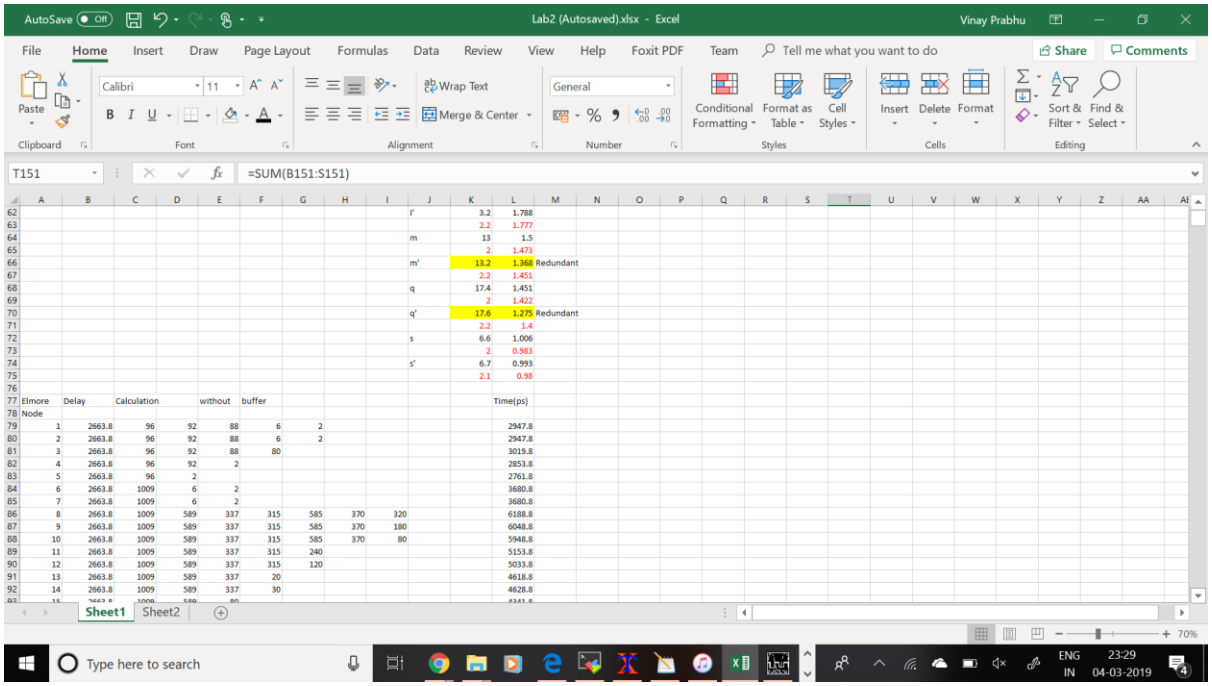
**Buffer insertion*
X1 1 a Vin gnd buf16
X2 3 b Vin gnd buf16
X3 4 c Vin gnd buf16
X4 5 d Vin gnd buf16
X5 2 e Vin gnd buf16
X6 15 f Vin gnd buf16
X7 16 g Vin gnd buf16
X8 17 h Vin gnd buf16
X9 19 i Vin gnd buf16
X10 32 j Vin gnd buf16
X11 33 k Vin gnd buf16
X12 36 l Vin gnd buf16
X13 39 m Vin gnd buf16
X14 44 n Vin gnd buf16

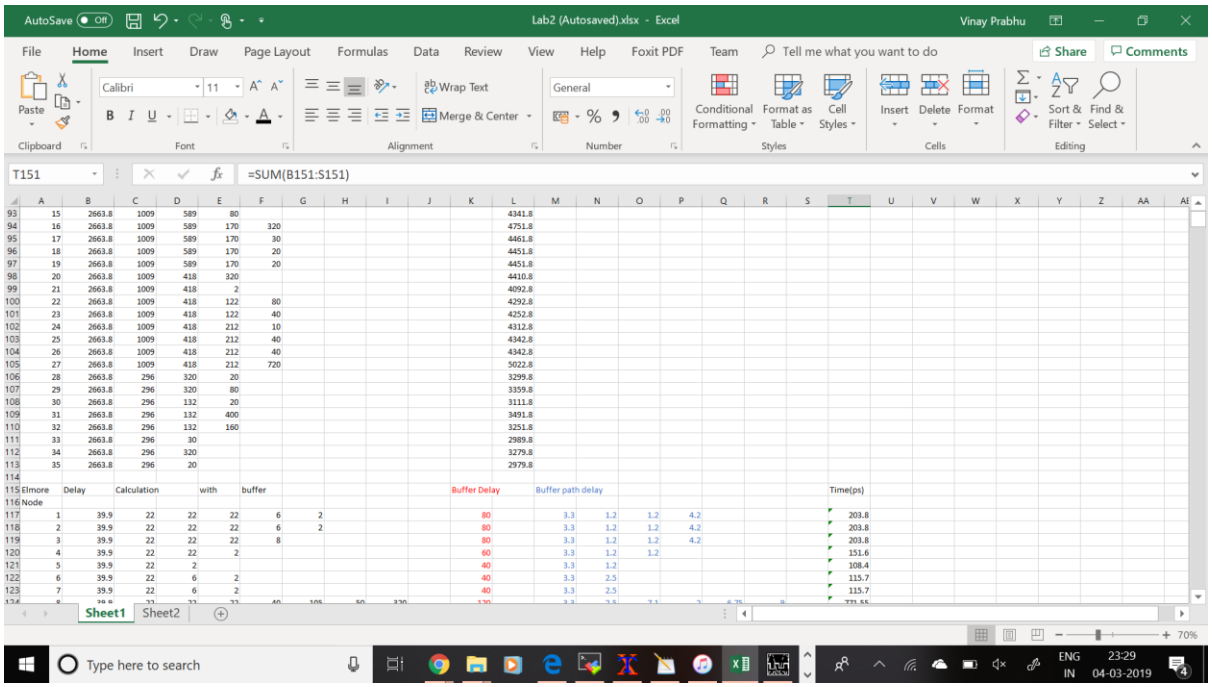
```
X15 45 o Vin gnd buf16
X16 48 p Vin gnd buf16

*Perform the analysis

.tran 0.01n 20n
.print tran V(vs) V(7)
.print tran V(vs) V(8)
.print tran V(vs) V(9)
.print tran V(vs) V(10)
.print tran V(vs) V(11)
.print tran V(vs) V(13)
.print tran V(vs) V(14)
.print tran V(vs) V(20)
.print tran V(vs) V(23)
.print tran V(vs) V(24)
.print tran V(vs) V(22)
.print tran V(vs) V(25)
.print tran V(vs) V(21)
.print tran V(vs) V(26)
.print tran V(vs) V(27)
.print tran V(vs) V(28)
.print tran V(vs) V(29)
.print tran V(vs) V(30)
.print tran V(vs) V(31)
.print tran V(vs) V(34)
.print tran V(vs) V(35)
.print tran V(vs) V(37)
.print tran V(vs) V(38)
.print tran V(vs) V(40)
.print tran V(vs) V(41)
.print tran V(vs) V(42)
.print tran V(vs) V(43)
.print tran V(vs) V(46)
.print tran V(vs) V(47)
.print tran V(vs) V(49)
.print tran V(vs) V(50)
.print tran V(vs) V(51)
.print tran V(vs) V(52)
.print tran V(vs) V(53)
.print tran V(vs) V(54)
.option post

.end
```

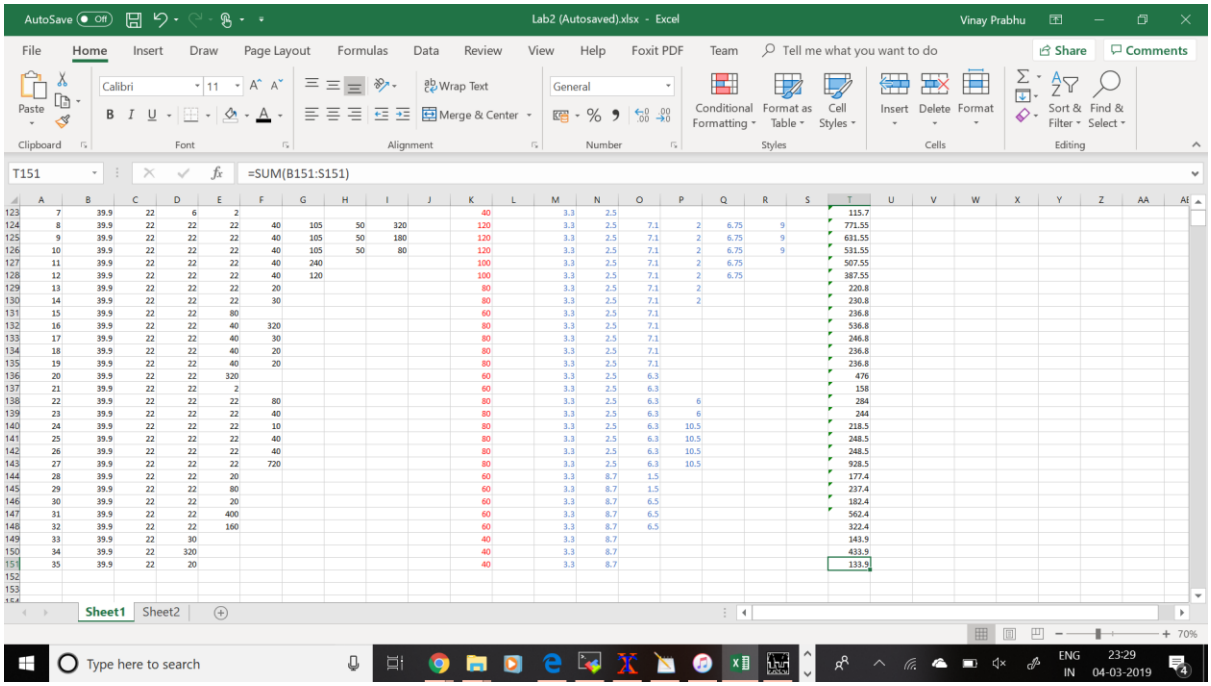
3. XL Sheet of the Calculations






Lab2 (Autosaved).xlsx - Excel
Vinay Prabhu
=SUM(B151:S151)
Sheet1
Sheet2
Type here to search


Lab2 (Autosaved).xlsx - Excel
Vinay Prabhu
=SUM(B151:S151)
Sheet1
Sheet2
Type here to search


Lab2 (Autosaved).xlsx - Excel
Vinay Prabhu
=SUM(B151:S151)
Sheet1
Sheet2
Type here to search
ENG
IN
23:29
04-03-2019