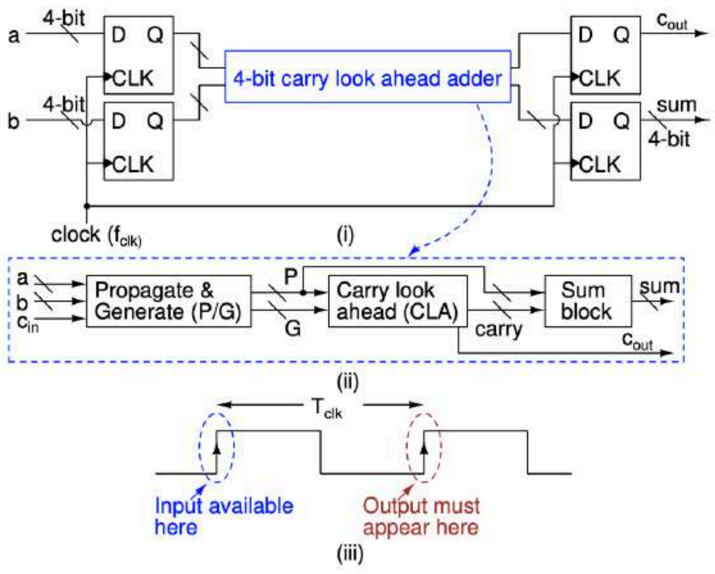
VLSI Course Project - Tanmay Pathak

2018102023

Question 1) - Proposed Structure of Adder

The circuit diagram given in a question is attached below



The adder to be implemented is the Carry Look Ahead Adder (CLA Adder). In this type of adder the current stage of the adder is determined by the previous stage ie. C_i is dependent upon C_{i-1} and thus the relation can be expressed as "recursive" which can help in arriving to the initial condition or C_0. C_0 helps in pre-calculate all the carry bits before hand and avoids the "undefined" state for output.

Again the question gave the following equations for the (i+1)th carry bit.

$$p_i = a_i \oplus b_i$$

$$g_i = a_i.b_i$$

And the carry out (c_{i+1}) of the ith bit position can be written as (assuming $c_0 = 0$) follows

$$c_{i+1} = (p_i c_i) + g_i,\ i=1,2,3,4$$

The adder can be divided into 3 blocks

1) Propagate and generate
2) Carry look ahead
3) Sum block

Propagate and generate

This blocks performs two primary tasks; firstly it decides whether the previous carry will be propagated ahead or not, second whether the given bits will generate a carry or not.

This block contains a XOR gate and an AND gate. We will enter the inputs as a_i and bi. From the equations for p_i and g_i we can find both those values since a_i and b_i is now given. We can then use all the equations to construct a table possibility with all the possibilities.

a_i	b_i	c_i	c_{i+1}
0	0	0	Neither Propagation nor generation of carry bit
0	0	1	
0	1	0	Previously generated carry bit is propagated ahead
1	0	0	
0	1	1	
1	0	1	
1	1	0	A new carry bit generated
1	1	1	

Carry Look Ahead

This block generates all the required carry bits beforehand by taking in all the p_i and g_i required. Since we are using a 4-bit adder, we will have the following equations generated.
We know the general equation as

$$c_n = p_{n-1}\, c_{n-1} + g_{n-1}$$

Thus for a bit adder we have to find c_4

$$c_4 = p_3\, c_3 + g_3$$

By performing recursive calculations we get

$$c_4 = p_3p_2p_1p_0c_0 + p_3p_2p_1g_0 + p_3p_2g_1 + p_3g_2 + g_3$$

Sum Block

The sum block represents the final equation which gives the the i^{th} sum bit is given by

$$sum_i = p_i \oplus c_i$$

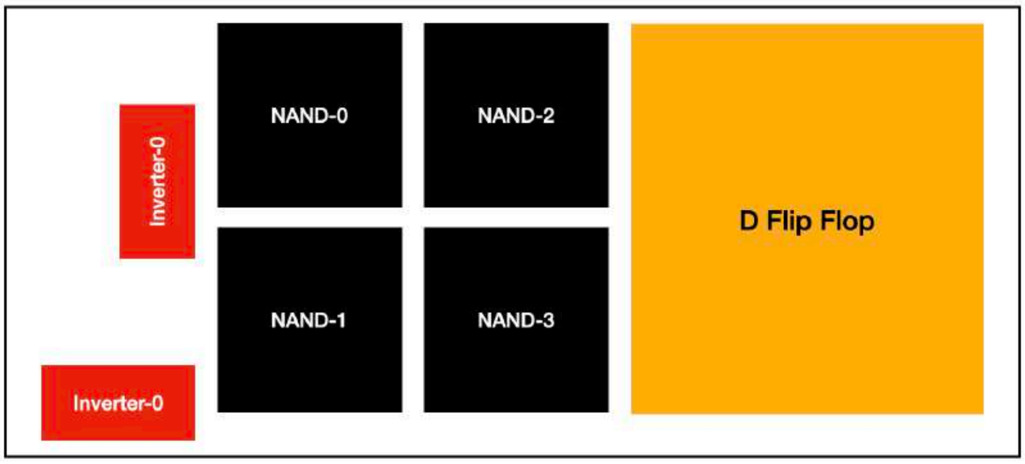
This sum block is simply an XOR gate.

Moreover, the entire adder also uses Flip-Flops to take input at the positive edge and give output on the next positive edge of the clock.

Question 2) - Design Details - Topology and sizing

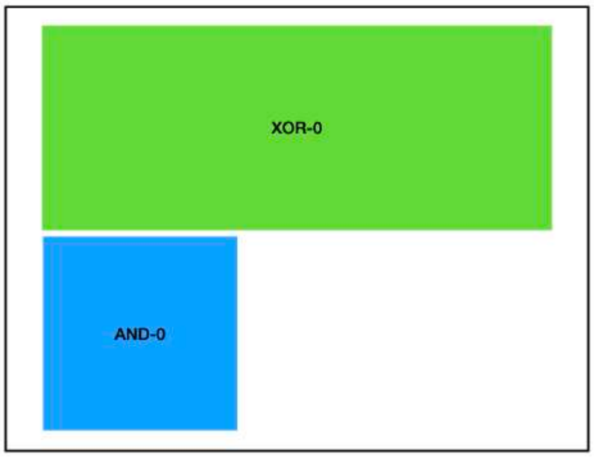
Flip-Flop

This will be the first block of the CLA adder and uses flip-flops to take the inputs. The flip-flops contain two latches which can be implemented using NAND gates and inverters. The shape chosen in rectangular for space optimisation. The flip-flops have the area 767λ by 268λ.



Propagate and Generate block

As mentioned above the propagate and generate block uses a XOR gate and an AND gate to give p_i and g_i. The following is the area occupied by the block: 326λ by 257λ

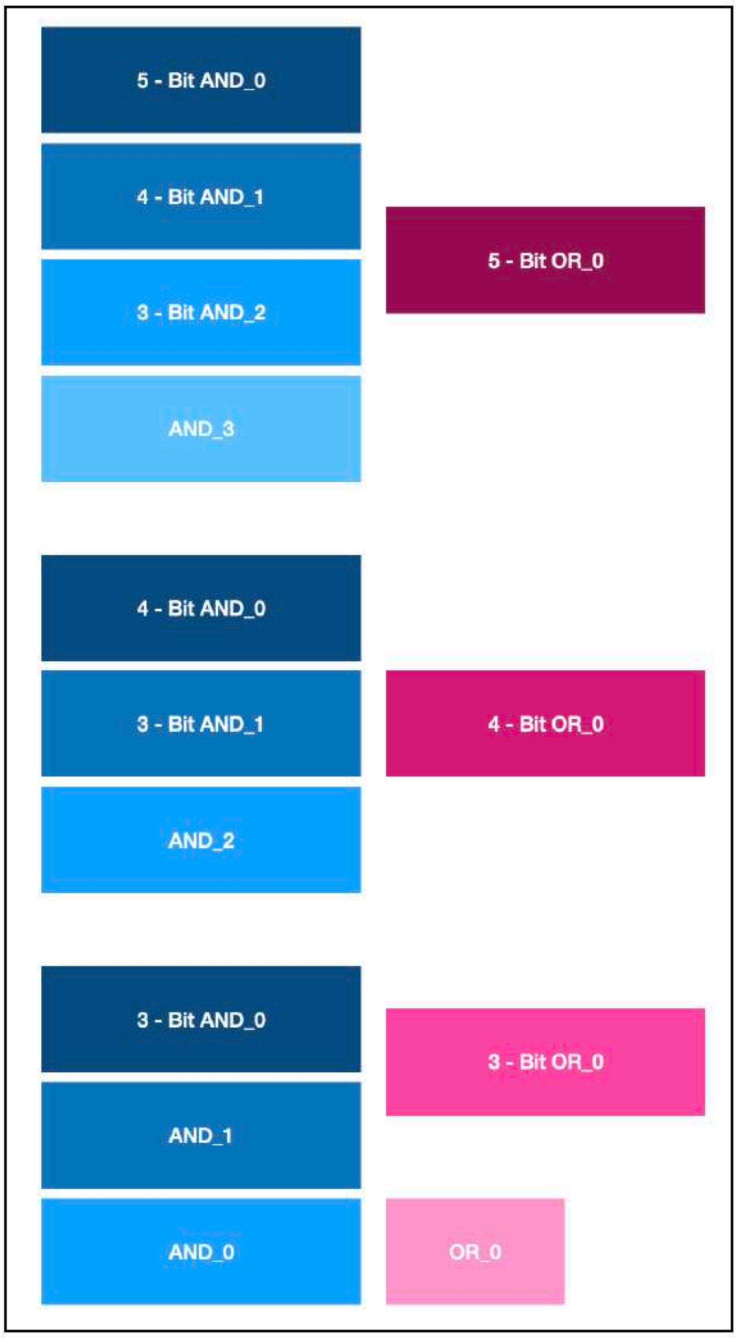


Carry Look Ahead Block

Uses multiple input AND and OR gates to generate carry bits. The exact numbers are

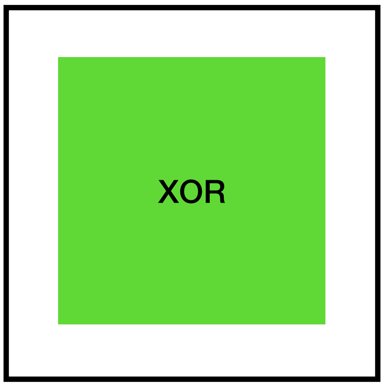
No. of Inputs	AND gates	OR gates
2	4	1
3	3	1
4	2	1
5	1	1

The implementation uses the following size 872λ by 2304λ



<u>Sum Block</u>

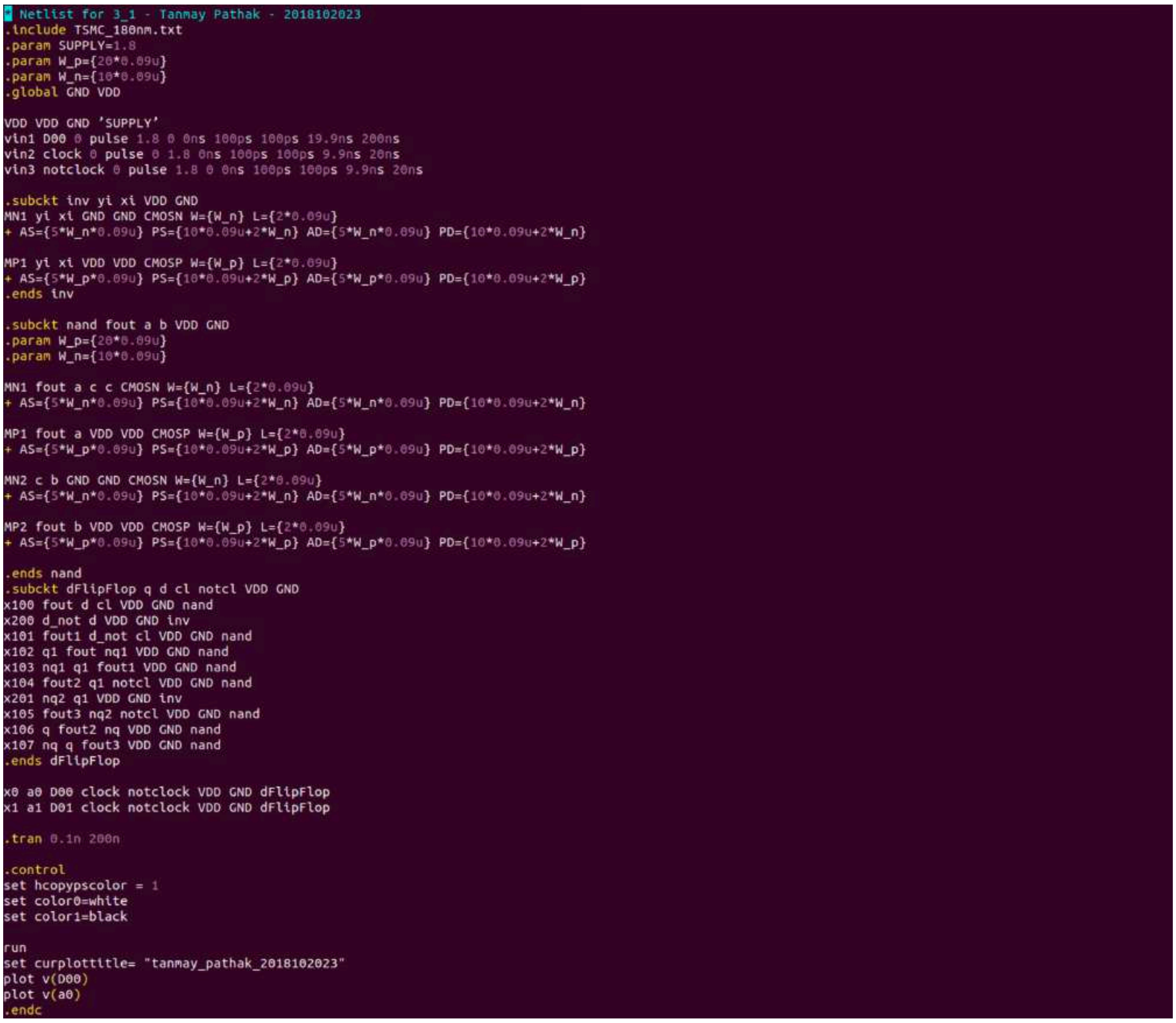
The sum uses a single XOR gate to generate the sum bit. The area occupied by it should be equal to that of a standard XOR gate which is 307λ by 119λ



Question 3) - Blockwise NGSPICE simulation for verification

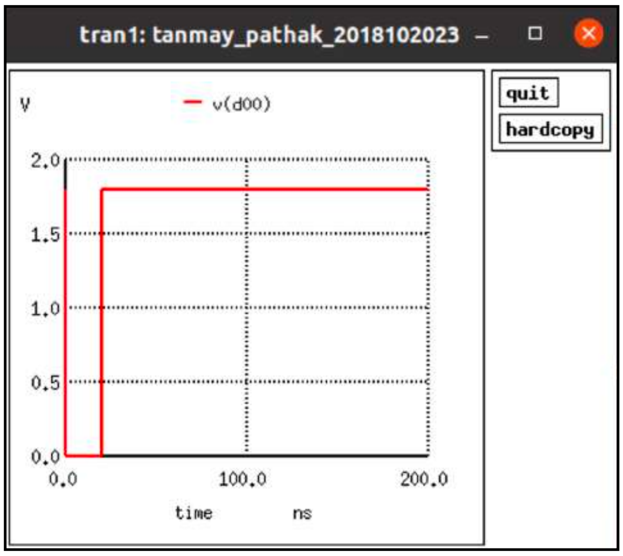
1) <u>D - FlipFlop</u>

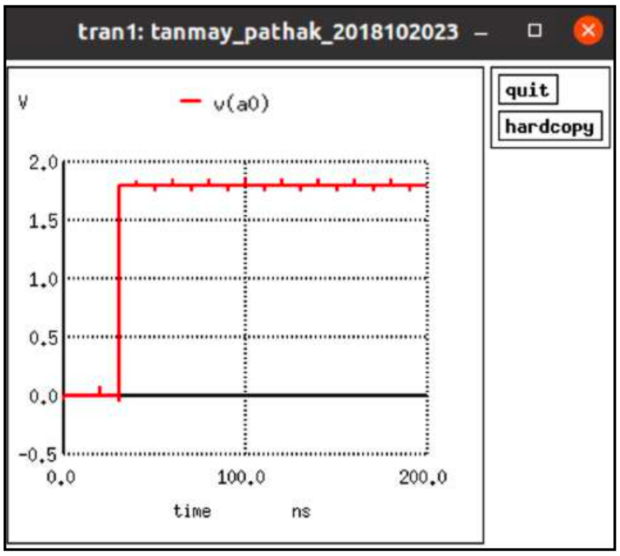
The netlist used in given below



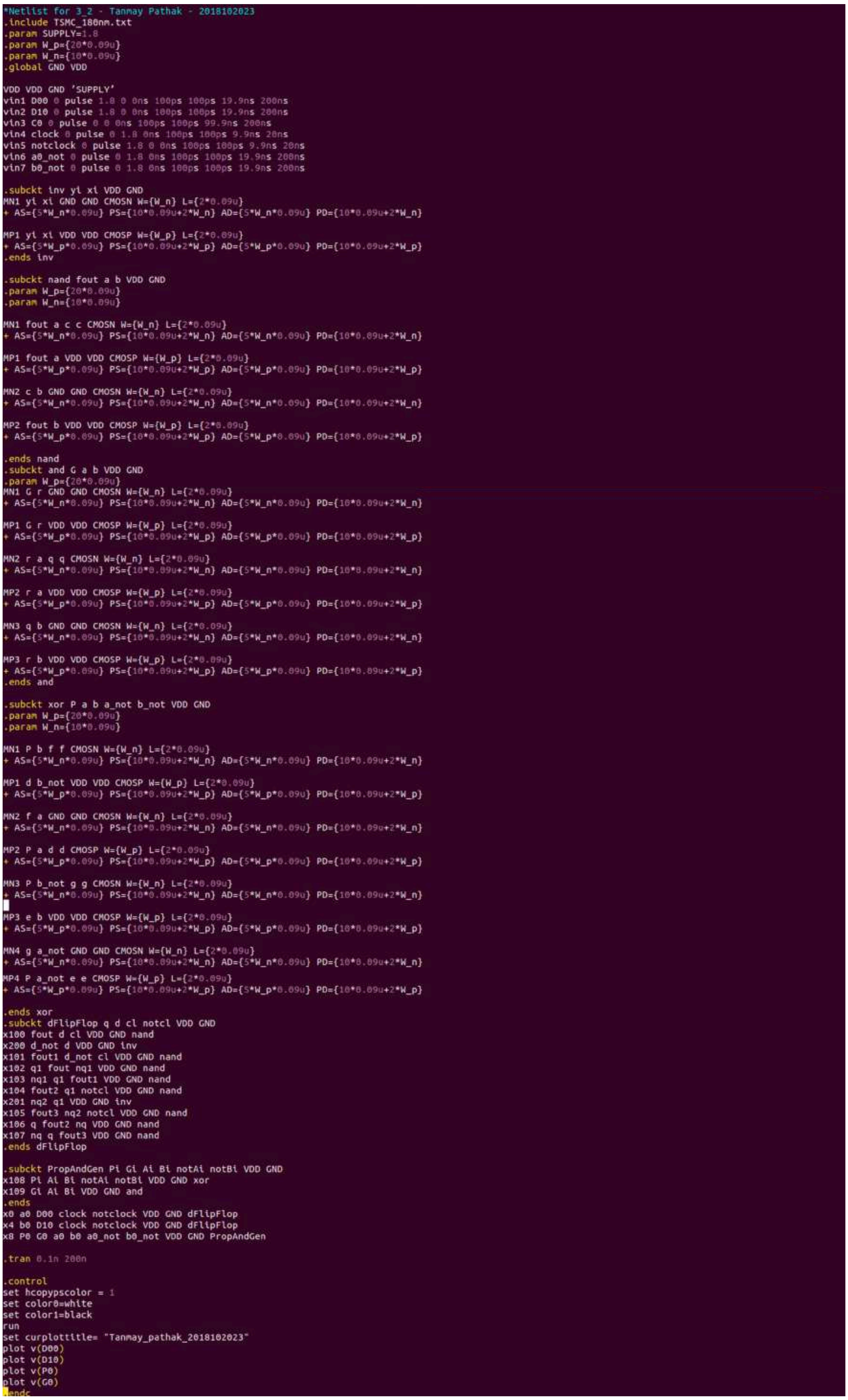
```
* Netlist for 3_1 - Tanmay Pathak - 2018102023
.include TSMC_180nm.txt
.param SUPPLY=1.8
.param W_p={20*0.09u}
.param W_n={10*0.09u}
.global GND VDD

VDD VDD GND 'SUPPLY'
vin1 D00 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin2 clock 0 pulse 0 1.8 0ns 100ps 100ps 9.9ns 20ns
vin3 notclock 0 pulse 1.8 0 0ns 100ps 100ps 9.9ns 20ns

.subckt inv yi xi VDD GND
MN1 yi xi GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 yi xi VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends inv

.subckt nand fout a b VDD GND
.param W_p={20*0.09u}
.param W_n={10*0.09u}

MN1 fout a c c CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 fout a VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 c b GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 fout b VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

.ends nand
.subckt dFlipFlop q d cl notcl VDD GND
x100 fout d cl VDD GND nand
x200 d_not d VDD GND inv
x101 fout1 d_not cl VDD GND nand
x102 q1 fout nq1 VDD GND nand
x103 nq1 q1 fout1 VDD GND nand
x104 fout2 q1 notcl VDD GND nand
x201 nq2 q1 VDD GND inv
x105 fout3 nq2 notcl VDD GND nand
x106 q fout2 nq VDD GND nand
x107 nq q fout3 VDD GND nand
.ends dFlipFlop

x0 a0 D00 clock notclock VDD GND dFlipFlop
x1 a1 D01 clock notclock VDD GND dFlipFlop

.tran 0.1n 200n

.control
set hcopypscolor = 1
set color0=white
set color1=black

run
set curplottitle= "tanmay_pathak_2018102023"
plot v(D00)
plot v(a0)
.endc
```

The following plots show the delay in a0 and D00



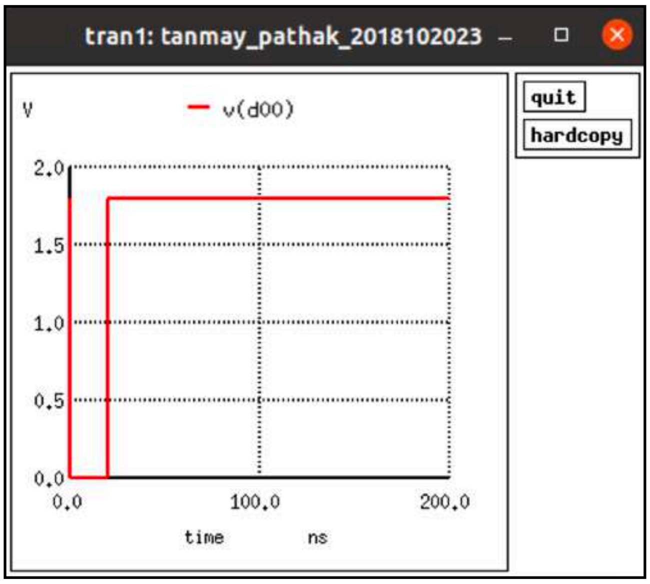


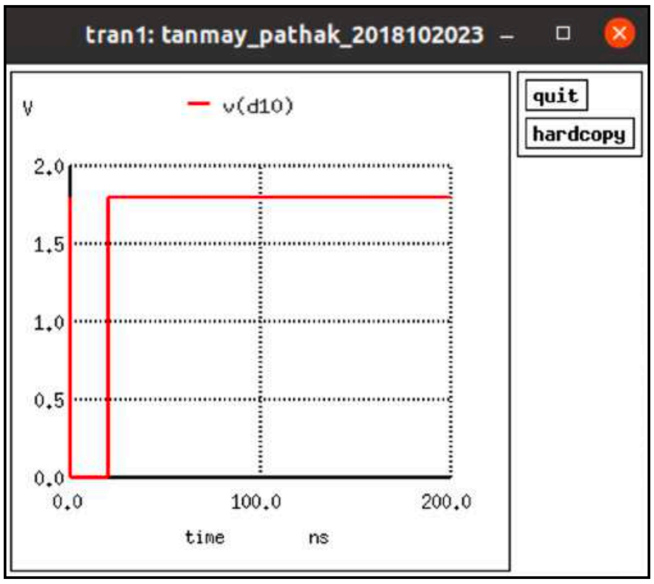
2) Propagate and Generate

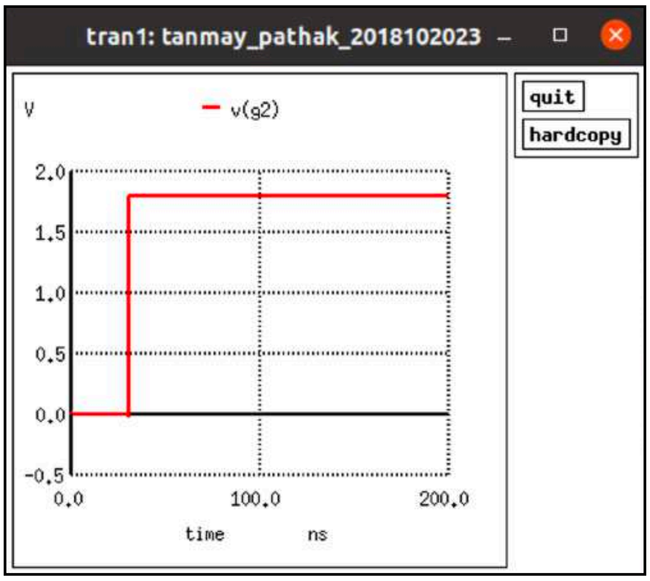


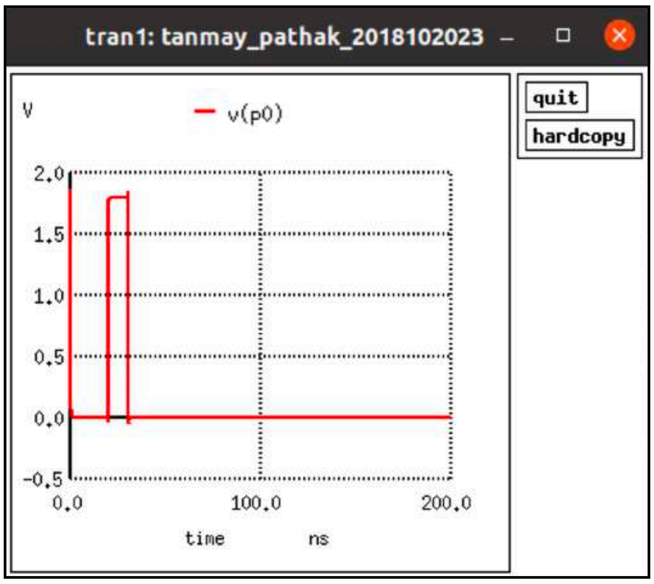
```
*Netlist for 3_2 - Tanmay Pathak - 2018102023
.include TSMC_180nm.txt
.param SUPPLY=1.8
.param W_p={20*0.09u}
.param W_n={10*0.09u}
.global GND VDD

VDD VDD GND 'SUPPLY'
vin1 D00 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin2 D10 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin3 C0 0 pulse 0 0 0ns 100ps 100ps 99.9ns 200ns
vin4 clock 0 pulse 0 1.8 0ns 100ps 100ps 9.9ns 20ns
vin5 notclock 0 pulse 1.8 0 0ns 100ps 100ps 9.9ns 20ns
vin6 a0_not 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin7 b0_not 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns

.subckt inv yi xi VDD GND
MN1 yi xi GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 yi xi VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends inv

.subckt nand fout a b VDD GND
.param W_p={20*0.09u}
.param W_n={10*0.09u}

MN1 fout a c c CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 fout a VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 c b GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 fout b VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

.ends nand
.subckt and G a b VDD GND
.param W_p={20*0.09u}
MN1 G r GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 G r VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 r a q q CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 r a VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN3 q b GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP3 r b VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends and

.subckt xor P a b a_not b_not VDD GND
.param W_p={20*0.09u}
.param W_n={10*0.09u}

MN1 P b f f CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 d b_not VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 f a GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 P a d d CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN3 P b_not g g CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP3 e b VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN4 g a_not GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}
MP4 P a_not e e CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

.ends xor
.subckt dFlipFlop q d cl notcl VDD GND
x100 fout d cl VDD GND nand
x200 d_not d VDD GND inv
x101 fout1 d_not cl VDD GND nand
x102 q1 fout nq1 VDD GND nand
x103 nq1 q1 fout1 VDD GND nand
x104 fout2 q1 notcl VDD GND nand
x201 nq2 q1 VDD GND inv
x105 fout3 nq2 notcl VDD GND nand
x106 q fout2 nq VDD GND nand
x107 nq q fout3 VDD GND nand
.ends dFlipFlop

.subckt PropAndGen Pi Gi Ai Bi notAi notBi VDD GND
x108 Pi Ai Bi notAi notBi VDD GND xor
x109 Gi Ai Bi VDD GND and
.ends
x0 a0 D00 clock notclock VDD GND dFlipFlop
x4 b0 D10 clock notclock VDD GND dFlipFlop
x8 P0 G0 a0 b0 a0_not b0_not VDD GND PropAndGen

.tran 0.1n 200n

.control
set hcopypscolor = 1
set color0=white
set color1=black
run
set curplottitle= "Tanmay_pathak_2018102023"
plot v(D00)
plot v(D10)
plot v(P0)
plot v(G0)
.endc
```

The output below show the generated p_i and g_i for the $D_{00} = 1$ and $D_{10} = 1$.



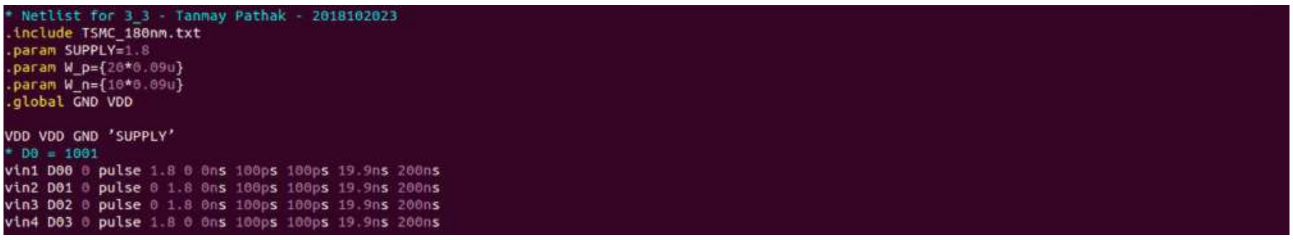




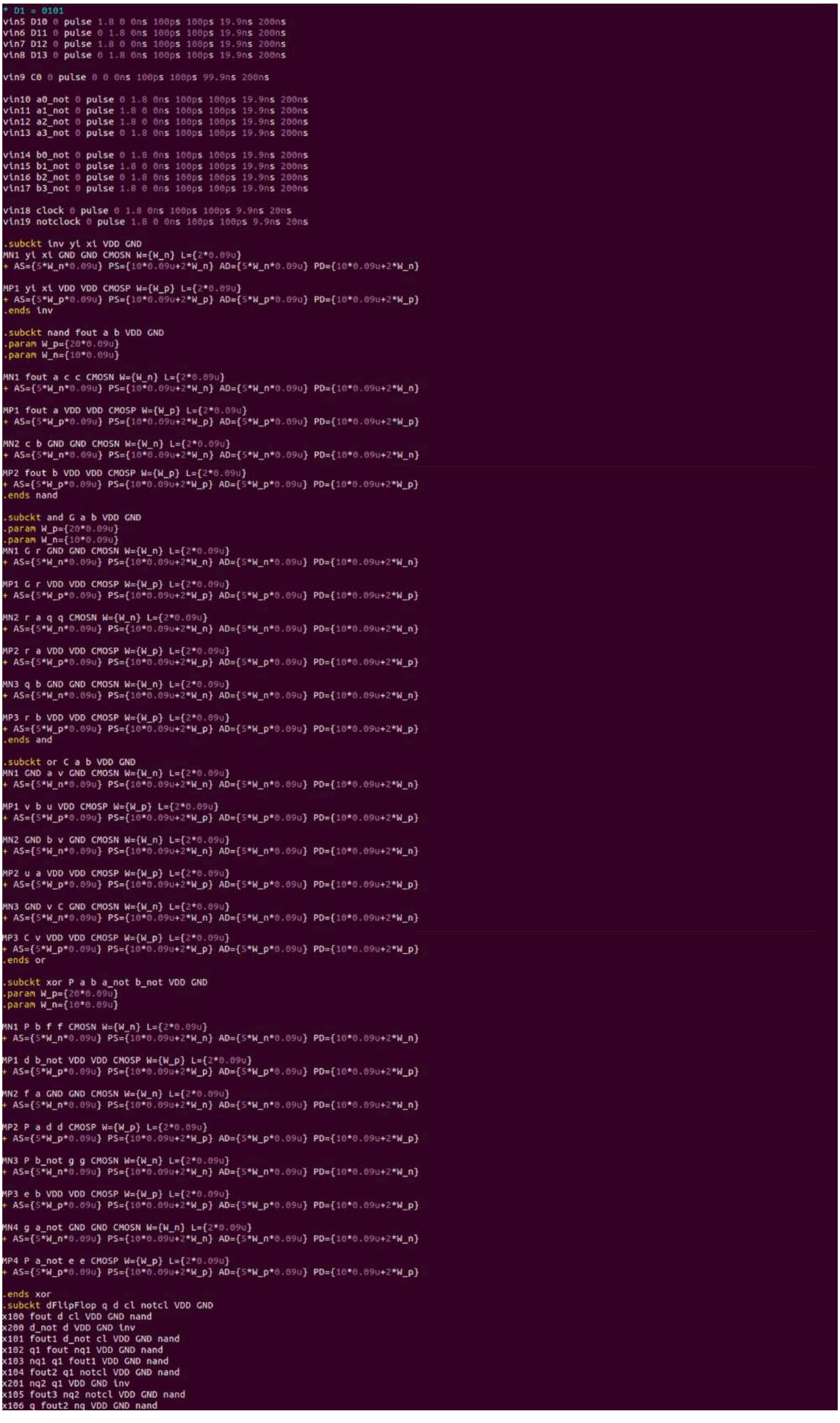


3) Carry Look Ahead Block

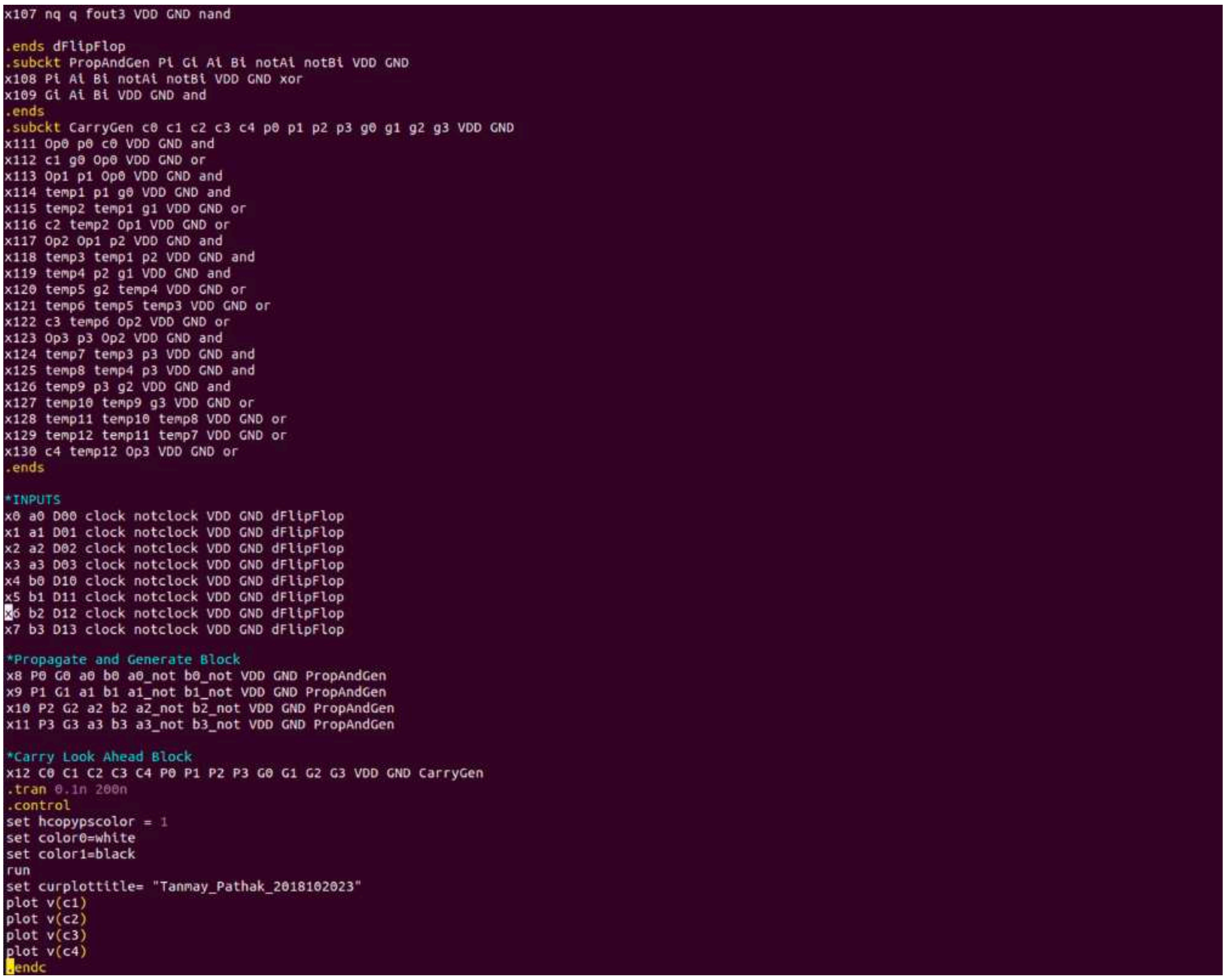
The netlist is given below



```
* Netlist for 3_3 - Tanmay Pathak - 2018102023
.include TSMC_180nm.txt
.param SUPPLY=1.8
.param W_p={20*0.09u}
.param W_n={10*0.09u}
.global GND VDD

VDD VDD GND 'SUPPLY'
* D0 = 1001
vin1 D00 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin2 D01 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin3 D02 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin4 D03 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
```



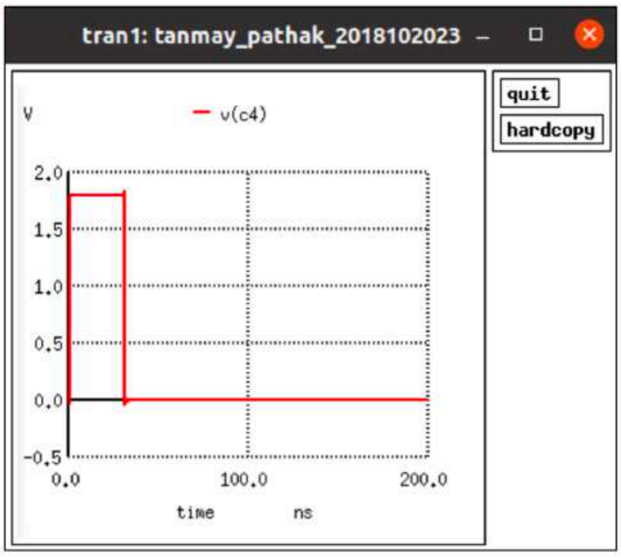
```
* D1 = 0101
vin5 D10 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin6 D11 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin7 D12 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin8 D13 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns

vin9 C0 0 pulse 0 0 0ns 100ps 100ps 99.9ns 200ns

vin10 a0_not 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin11 a1_not 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin12 a2_not 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin13 a3_not 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns

vin14 b0_not 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin15 b1_not 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin16 b2_not 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin17 b3_not 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns

vin18 clock 0 pulse 0 1.8 0ns 100ps 100ps 9.9ns 20ns
vin19 notclock 0 pulse 1.8 0 0ns 100ps 100ps 9.9ns 20ns

.subckt inv yi xi VDD GND
MN1 yi xi GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 yi xi VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends inv

.subckt nand fout a b VDD GND
.param W_p={20*0.09u}
.param W_n={10*0.09u}

MN1 fout a c c CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 fout a VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 c b GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 fout b VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends nand

.subckt and G a b VDD GND
.param W_p={20*0.09u}
.param W_n={10*0.09u}
MN1 G r GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 G r VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 r a q q CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 r a VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN3 q b GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP3 r b VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends and

.subckt or C a b VDD GND
MN1 GND a v GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 v b u VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 GND b v GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 u a VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN3 GND v C GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP3 C v VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends or

.subckt xor P a b a_not b_not VDD GND
.param W_p={20*0.09u}
.param W_n={10*0.09u}

MN1 P b f f CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 d b_not VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 f a GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 P a d d CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN3 P b_not g g CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP3 e b VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN4 g a_not GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP4 P a_not e e CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

.ends xor
.subckt dFlipFlop q d cl notcl VDD GND
x100 fout d cl VDD GND nand
x200 d_not d VDD GND inv
x101 fout1 d_not cl VDD GND nand
x102 q1 fout nq1 VDD GND nand
x103 nq1 q1 fout1 VDD GND nand
x104 fout2 q1 notcl VDD GND nand
x201 nq2 q1 VDD GND inv
x105 fout3 nq2 notcl VDD GND nand
x106 q fout2 nq VDD GND nand
```

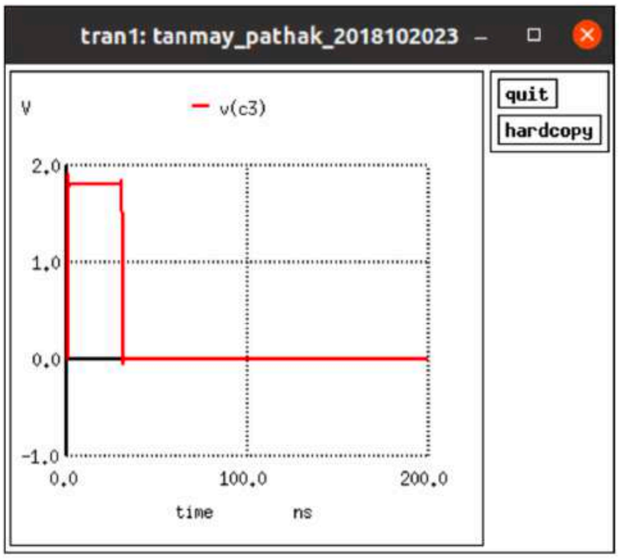


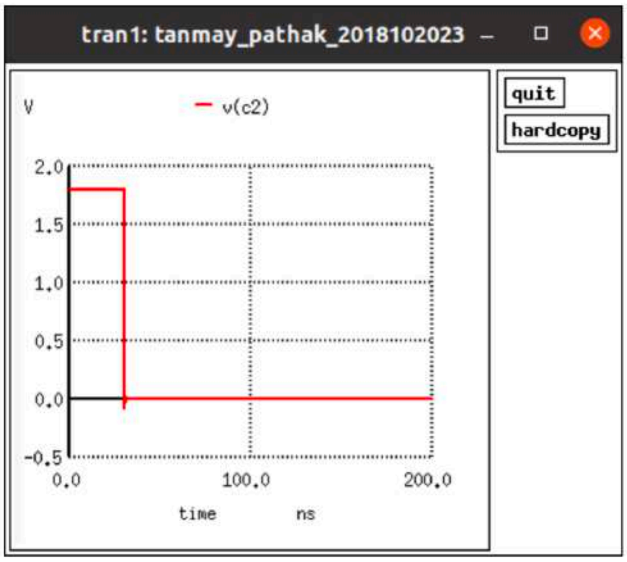
```
x107 nq q fout3 VDD GND nand

.ends dFlipFlop
.subckt PropAndGen Pi Gi Ai Bi notAi notBi VDD GND
x108 Pi Ai Bi notAi notBi VDD GND xor
x109 Gi Ai Bi VDD GND and
.ends
.subckt CarryGen c0 c1 c2 c3 c4 p0 p1 p2 p3 g0 g1 g2 g3 VDD GND
x111 Op0 p0 c0 VDD GND and
x112 c1 g0 Op0 VDD GND or
x113 Op1 p1 Op0 VDD GND and
x114 temp1 p1 g0 VDD GND and
x115 temp2 temp1 g1 VDD GND or
x116 c2 temp2 Op1 VDD GND or
x117 Op2 Op1 p2 VDD GND and
x118 temp3 temp1 p2 VDD GND and
x119 temp4 p2 g1 VDD GND and
x120 temp5 g2 temp4 VDD GND or
x121 temp6 temp5 temp3 VDD GND or
x122 c3 temp6 Op2 VDD GND or
x123 Op3 p3 Op2 VDD GND and
x124 temp7 temp3 p3 VDD GND and
x125 temp8 temp4 p3 VDD GND and
x126 temp9 p3 g2 VDD GND and
x127 temp10 temp9 g3 VDD GND or
x128 temp11 temp10 temp8 VDD GND or
x129 temp12 temp11 temp7 VDD GND or
x130 c4 temp12 Op3 VDD GND or
.ends

*INPUTS
x0 a0 D00 clock notclock VDD GND dFlipFlop
x1 a1 D01 clock notclock VDD GND dFlipFlop
x2 a2 D02 clock notclock VDD GND dFlipFlop
x3 a3 D03 clock notclock VDD GND dFlipFlop
x4 b0 D10 clock notclock VDD GND dFlipFlop
x5 b1 D11 clock notclock VDD GND dFlipFlop
x6 b2 D12 clock notclock VDD GND dFlipFlop
x7 b3 D13 clock notclock VDD GND dFlipFlop

*Propagate and Generate Block
x8 P0 G0 a0 b0 a0_not b0_not VDD GND PropAndGen
x9 P1 G1 a1 b1 a1_not b1_not VDD GND PropAndGen
x10 P2 G2 a2 b2 a2_not b2_not VDD GND PropAndGen
x11 P3 G3 a3 b3 a3_not b3_not VDD GND PropAndGen

*Carry Look Ahead Block
x12 C0 C1 C2 C3 C4 P0 P1 P2 P3 G0 G1 G2 G3 VDD GND CarryGen
.tran 0.1n 200n
.control
set hcopypscolor = 1
set color0=white
set color1=black
run
set curplottitle= "Tanmay_Pathak_2018102023"
plot v(c1)
plot v(c2)
plot v(c3)
plot v(c4)
.endc
```

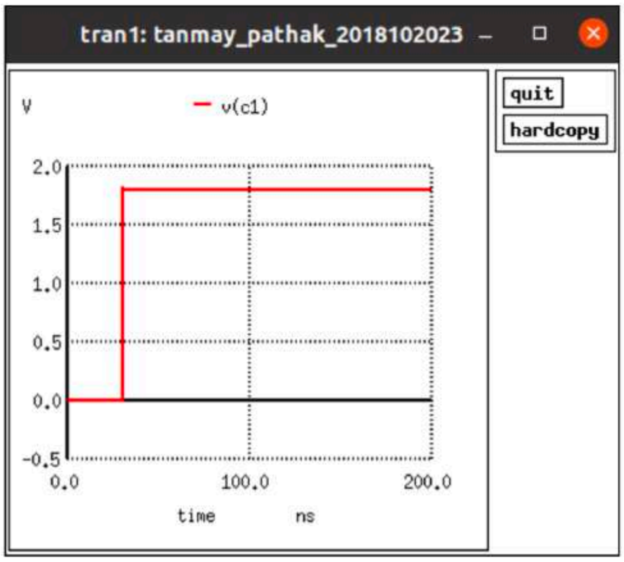
Input number were D_0 = 1001 and D_1=0101. The carry bits (C_4 C_3 C_2 C_1) are 0001

OUTPUTS



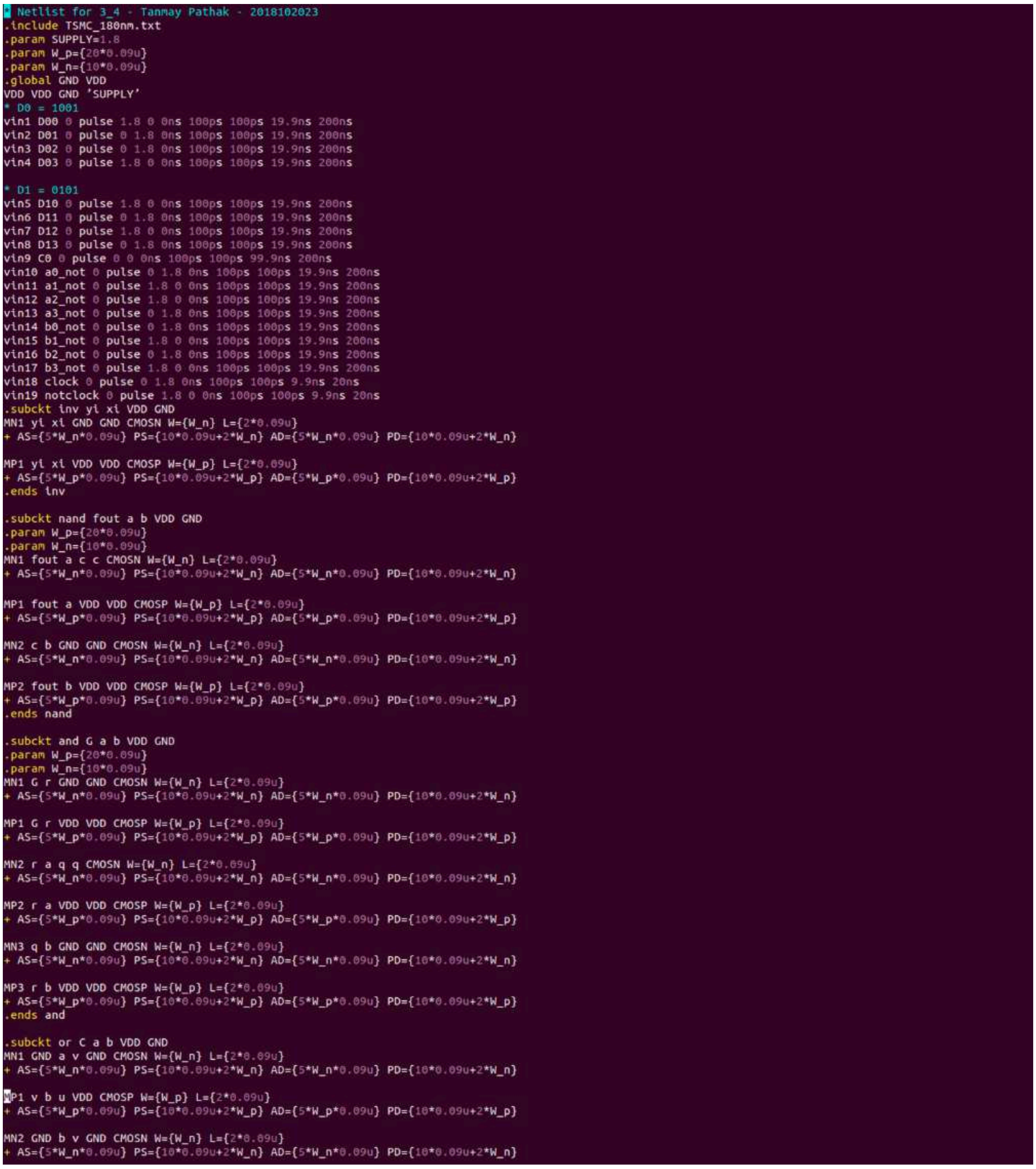




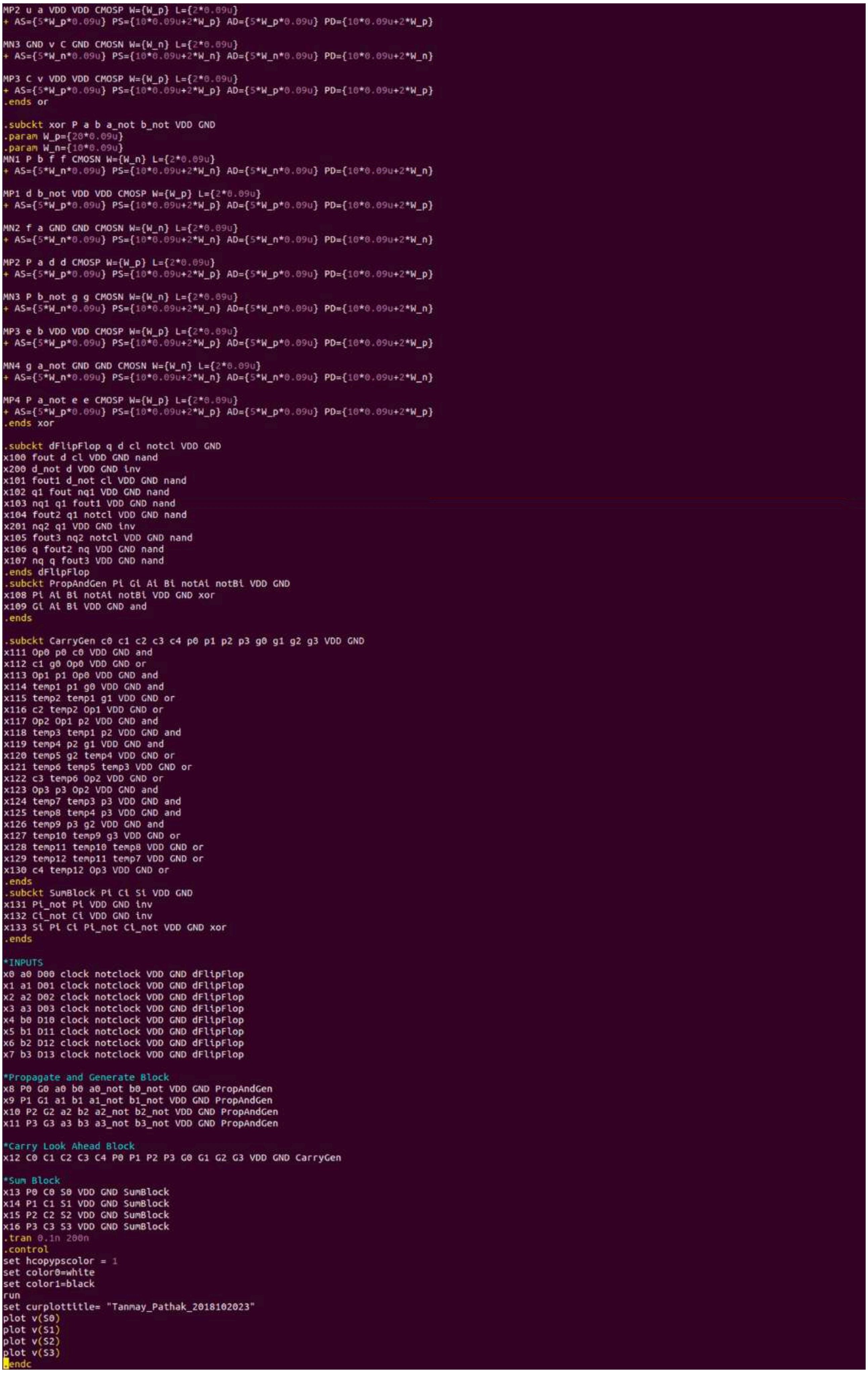


4) SUM Block

The netlist used for the question is given below



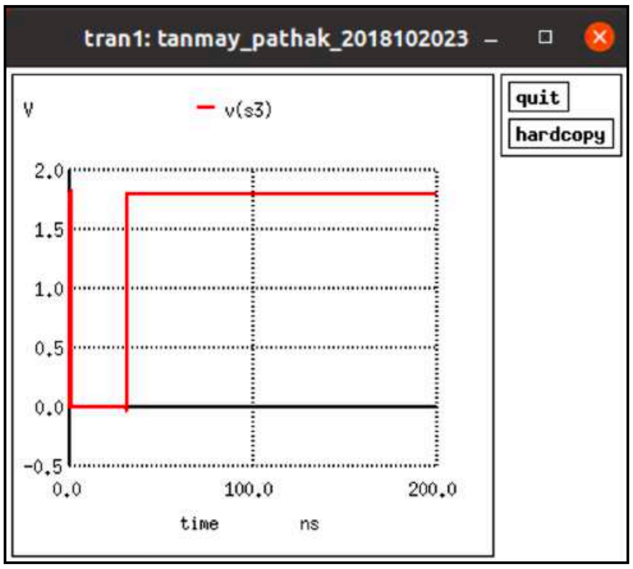
```
* Netlist for 3_4 - Tanmay Pathak - 2018102023
.include TSMC_180nm.txt
.param SUPPLY=1.8
.param W_p={20*0.09u}
.param W_n={10*0.09u}
.global GND VDD
VDD VDD GND 'SUPPLY'
* D0 = 1001
vin1 D00 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin2 D01 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin3 D02 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin4 D03 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns

* D1 = 0101
vin5 D10 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin6 D11 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin7 D12 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin8 D13 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin9 C0 0 pulse 0 0 0ns 100ps 100ps 99.9ns 200ns
vin10 a0_not 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin11 a1_not 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin12 a2_not 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin13 a3_not 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin14 b0_not 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin15 b1_not 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin16 b2_not 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin17 b3_not 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin18 clock 0 pulse 0 1.8 0ns 100ps 100ps 9.9ns 20ns
vin19 notclock 0 pulse 1.8 0 0ns 100ps 100ps 9.9ns 20ns
.subckt inv yi xi VDD GND
MN1 yi xi GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 yi xi VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends inv

.subckt nand fout a b VDD GND
.param W_p={20*0.09u}
.param W_n={10*0.09u}
MN1 fout a c c CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 fout a VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 c b GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 fout b VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends nand

.subckt and G a b VDD GND
.param W_p={20*0.09u}
.param W_n={10*0.09u}
MN1 G r GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 G r VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 r a q q CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 r a VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN3 q b GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP3 r b VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends and

.subckt or C a b VDD GND
MN1 GND a v GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 v b u VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 GND b v GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}
```

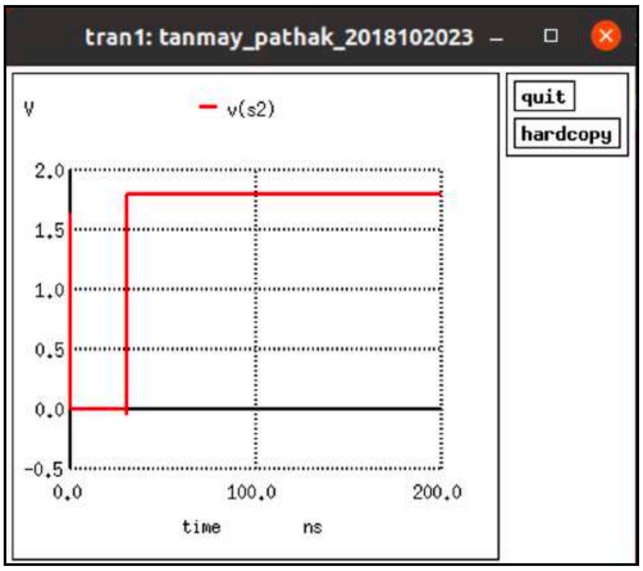


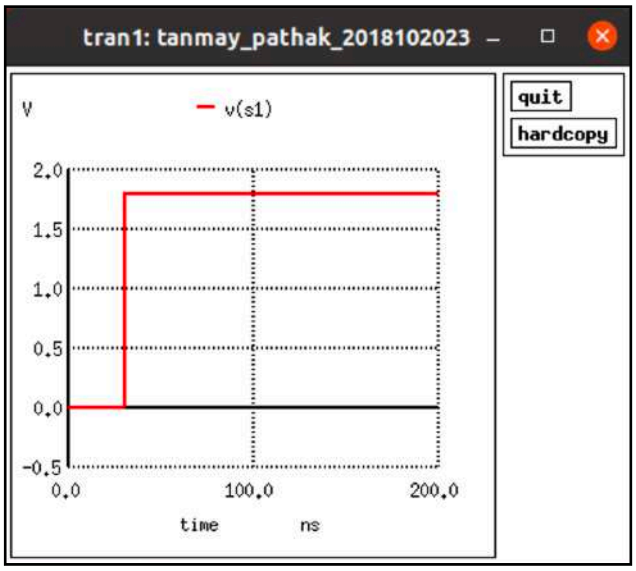
```
MP2 u a VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN3 GND v C GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP3 C v VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends or

.subckt xor P a b a_not b_not VDD GND
.param W_p={20*0.09u}
.param W_n={10*0.09u}
MN1 P b f f CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 d b_not VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 f a GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 P a d d CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN3 P b_not g g CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP3 e b VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN4 g a_not GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP4 P a_not e e CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends xor

.subckt dFlipFlop q d cl notcl VDD GND
x100 fout d cl VDD GND nand
x200 d_not d VDD GND inv
x101 fout1 d_not cl VDD GND nand
x102 q1 fout nq1 VDD GND nand
x103 nq1 q1 fout1 VDD GND nand
x104 fout2 q1 notcl VDD GND nand
x201 nq2 q1 VDD GND inv
x105 fout3 nq2 notcl VDD GND nand
x106 q fout2 nq VDD GND nand
x107 nq q fout3 VDD GND nand
.ends dFlipFlop
.subckt PropAndGen Pi Gi Ai Bi notAi notBi VDD GND
x108 Pi Ai Bi notAi notBi VDD GND xor
x109 Gi Ai Bi VDD GND and
.ends

.subckt CarryGen c0 c1 c2 c3 c4 p0 p1 p2 p3 g0 g1 g2 g3 VDD GND
x111 Op0 p0 c0 VDD GND and
x112 c1 g0 Op0 VDD GND or
x113 Op1 p1 Op0 VDD GND and
x114 temp1 p1 g0 VDD GND and
x115 temp2 temp1 g1 VDD GND or
x116 c2 temp2 Op1 VDD GND or
x117 Op2 Op1 p2 VDD GND and
x118 temp3 temp1 p2 VDD GND and
x119 temp4 p2 g1 VDD GND and
x120 temp5 g2 temp4 VDD GND or
x121 temp6 temp5 temp3 VDD GND or
x122 c3 temp6 Op2 VDD GND or
x123 Op3 p3 Op2 VDD GND and
x124 temp7 temp3 p3 VDD GND and
x125 temp8 temp4 p3 VDD GND and
x126 temp9 p3 g2 VDD GND and
x127 temp10 temp9 g3 VDD GND or
x128 temp11 temp10 temp8 VDD GND or
x129 temp12 temp11 temp7 VDD GND or
x130 c4 temp12 Op3 VDD GND or
.ends
.subckt SumBlock Pi Ci Si VDD GND
x131 Pi_not Pi VDD GND inv
x132 Ci_not Ci VDD GND inv
x133 Si Pi Ci Pi_not Ci_not VDD GND xor
.ends

*INPUTS
x0 a0 D00 clock notclock VDD GND dFlipFlop
x1 a1 D01 clock notclock VDD GND dFlipFlop
x2 a2 D02 clock notclock VDD GND dFlipFlop
x3 a3 D03 clock notclock VDD GND dFlipFlop
x4 b0 D10 clock notclock VDD GND dFlipFlop
x5 b1 D11 clock notclock VDD GND dFlipFlop
x6 b2 D12 clock notclock VDD GND dFlipFlop
x7 b3 D13 clock notclock VDD GND dFlipFlop

*Propagate and Generate Block
x8 P0 G0 a0 b0 a0_not b0_not VDD GND PropAndGen
x9 P1 G1 a1 b1 a1_not b1_not VDD GND PropAndGen
x10 P2 G2 a2 b2 a2_not b2_not VDD GND PropAndGen
x11 P3 G3 a3 b3 a3_not b3_not VDD GND PropAndGen

*Carry Look Ahead Block
x12 C0 C1 C2 C3 C4 P0 P1 P2 P3 G0 G1 G2 G3 VDD GND CarryGen

*Sum Block
x13 P0 C0 S0 VDD GND SumBlock
x14 P1 C1 S1 VDD GND SumBlock
x15 P2 C2 S2 VDD GND SumBlock
x16 P3 C3 S3 VDD GND SumBlock
.tran 0.1n 200n
.control
set hcopypscolor = 1
set color0=white
set color1=black
run
set curplottitle= "Tanmay_Pathak_2018102023"
plot v(S0)
plot v(S1)
plot v(S2)
plot v(S3)
.endc
```

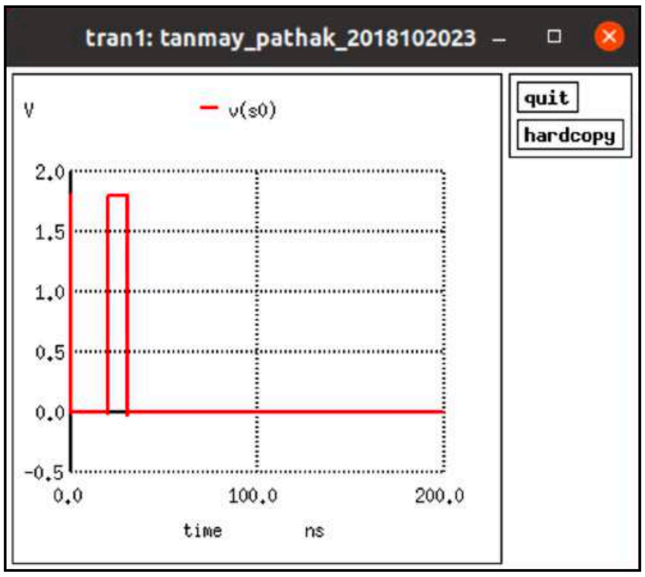
OUTPUTS

Input number were D_0 = 1001 and D_1=0101. The carry bits (S_4 S_3 S_2 S_1) are 1110









Question 4) - Setup time, hold time and clock to Q delay from NGSPICE sim

Result of transient analysis

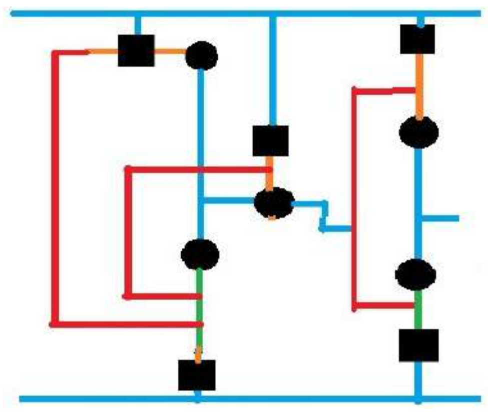
	Measurements for FlipFlop
tpd	1.014270E-08
targ	3.01927E-08
trig	2.005000E-08

Question 5) - Stick diagrams of all unique gates in your design

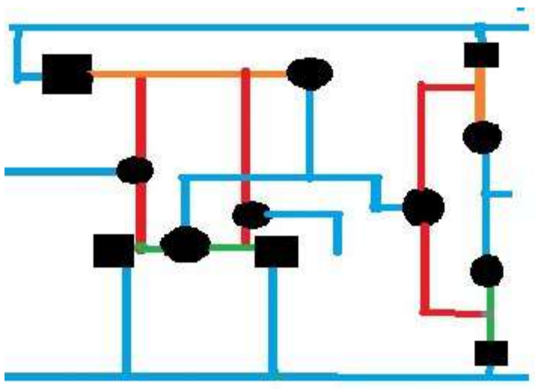
The design uses three unique gates

1) AND gate
2) OR gate
3) XOR gate

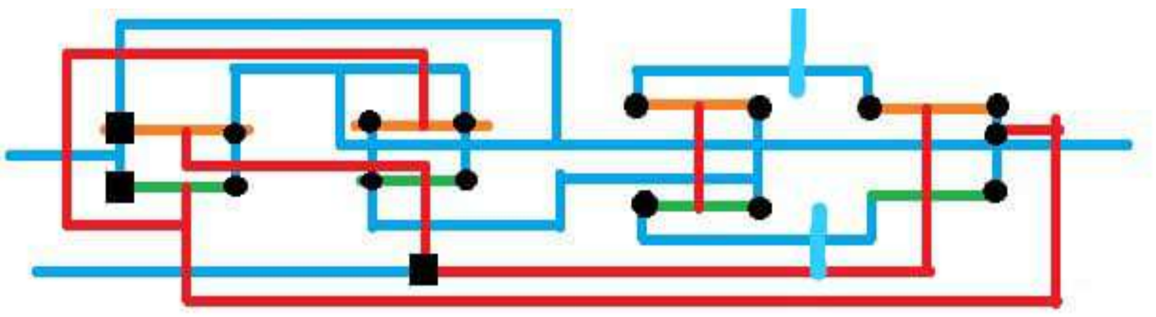
AND Gate stick diagram



OR Gate stick diagram

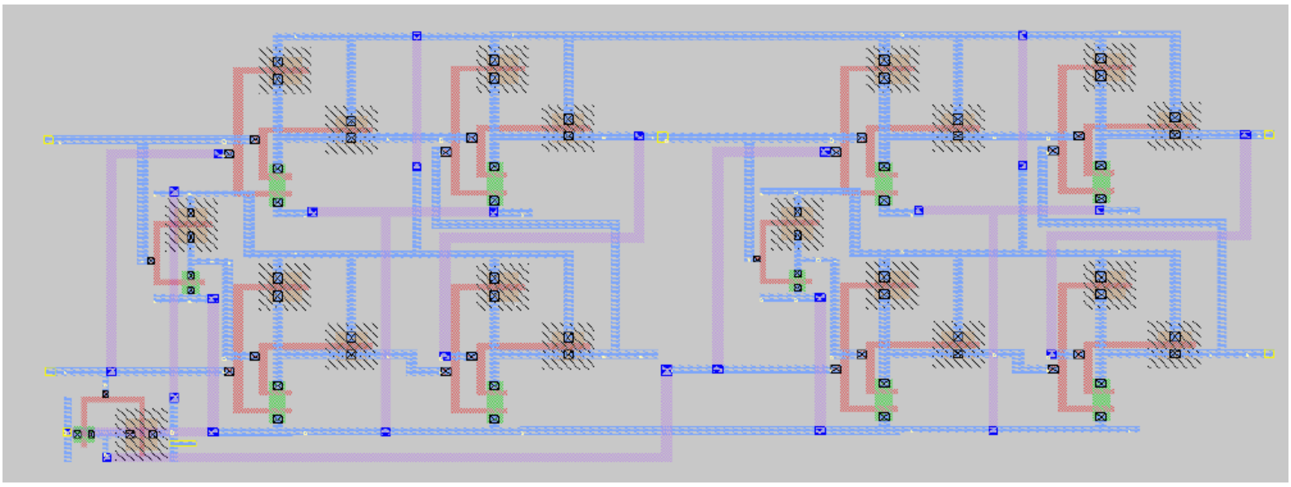


XOR Gate stick diagram

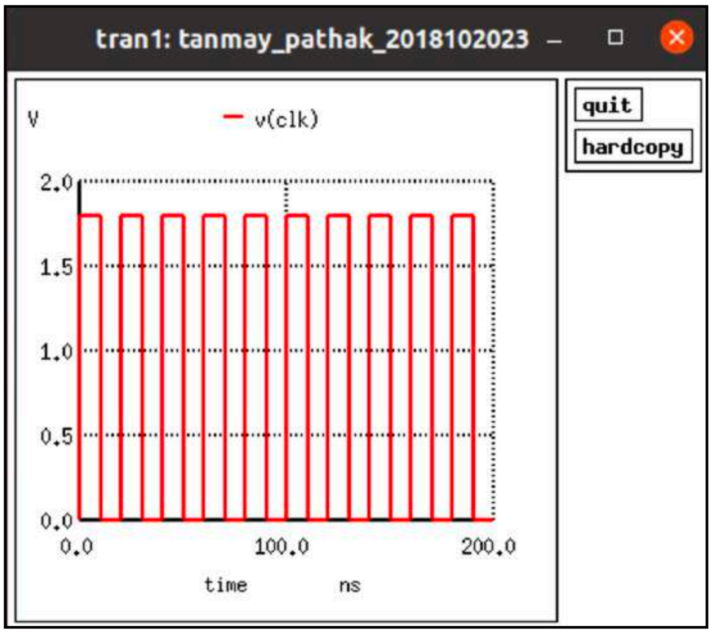


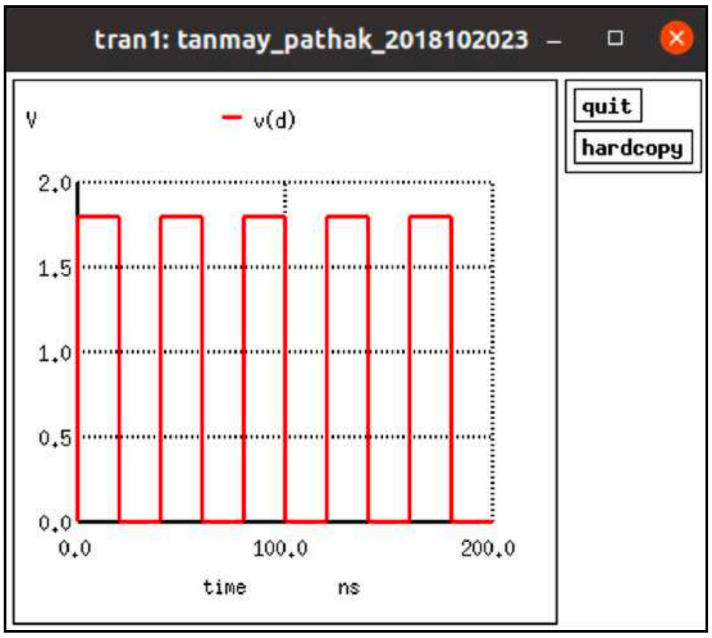
Question 6) - Layout each block using MAGIC

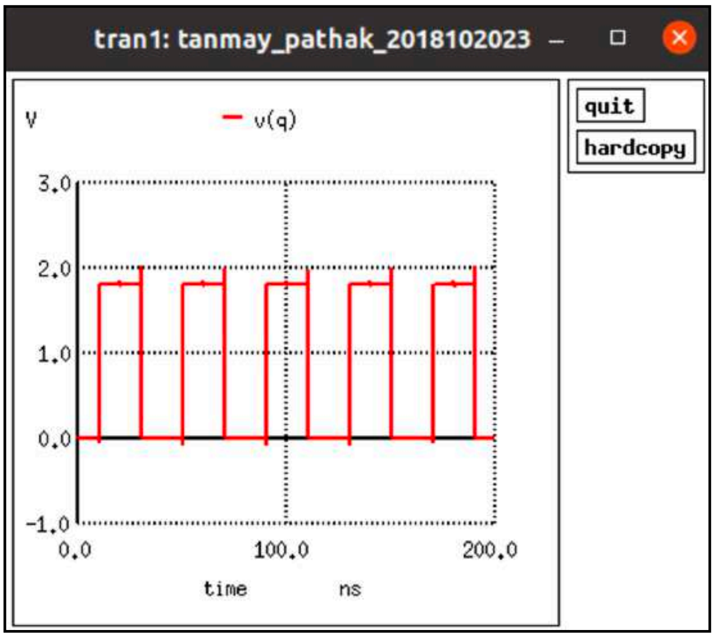
1) Flip-Flop

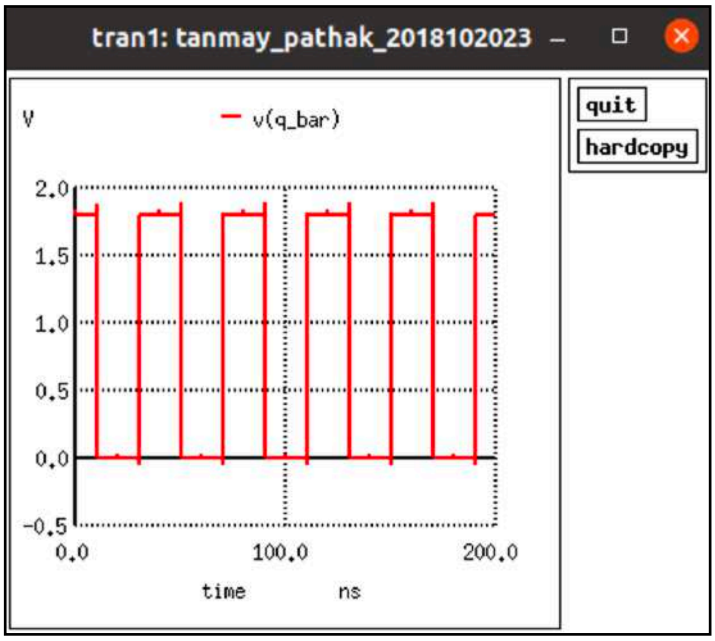


OUTPUT - Magic layout simulation

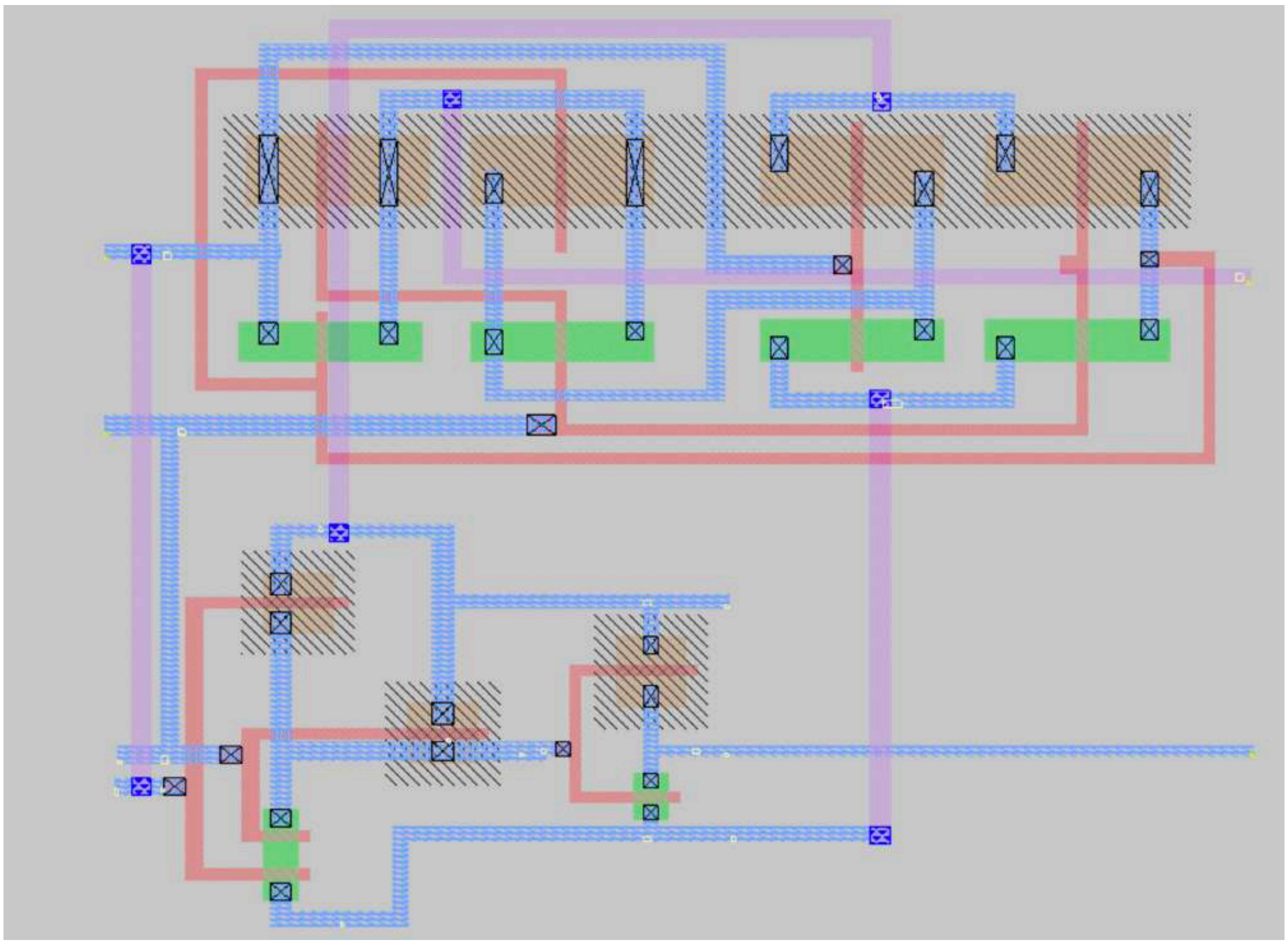




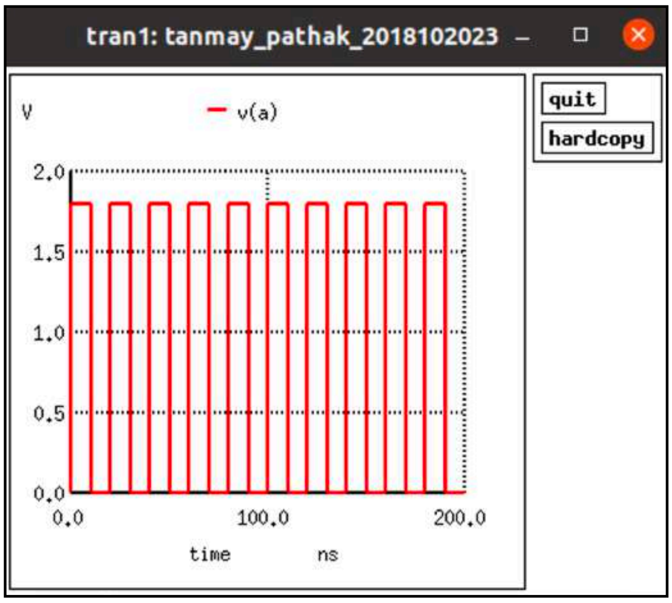


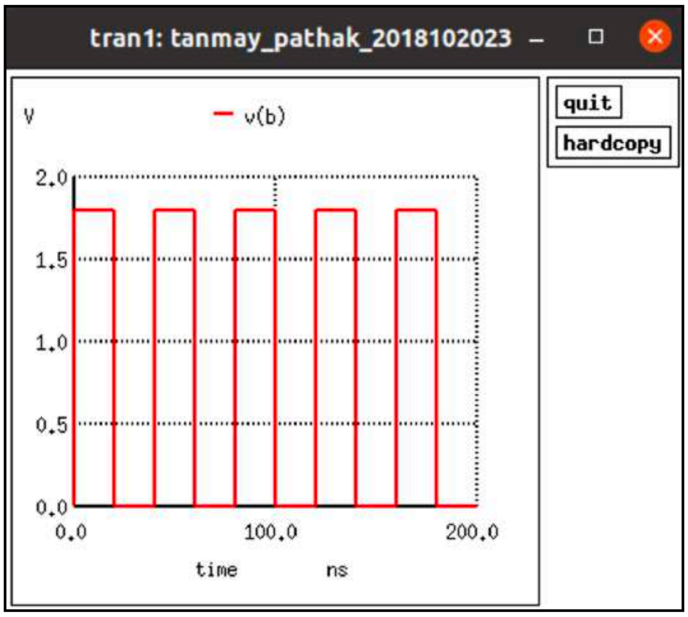


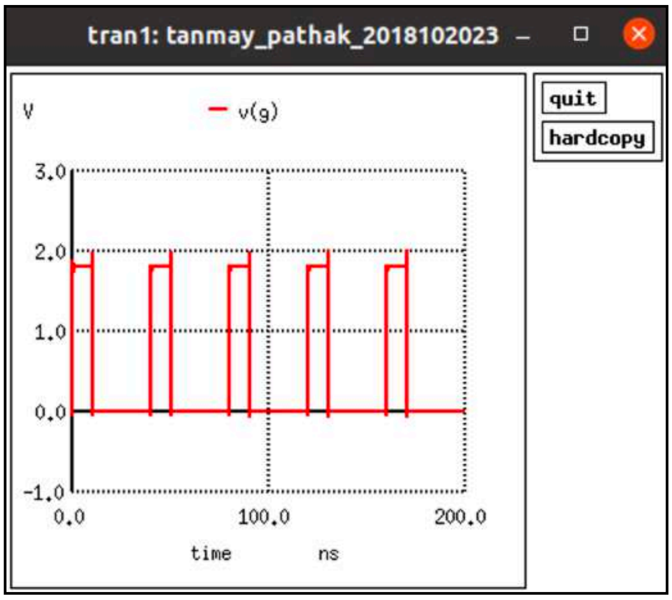
2) Propagate and generate block

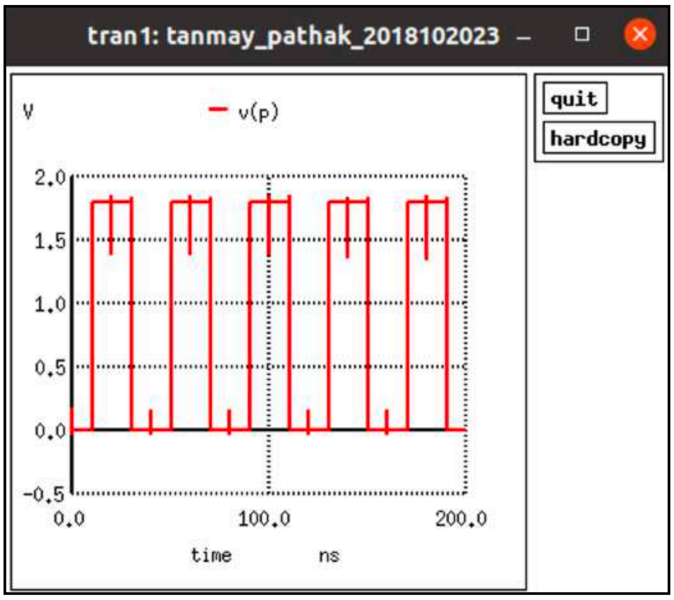


OUTPUT - Magic layout simulation

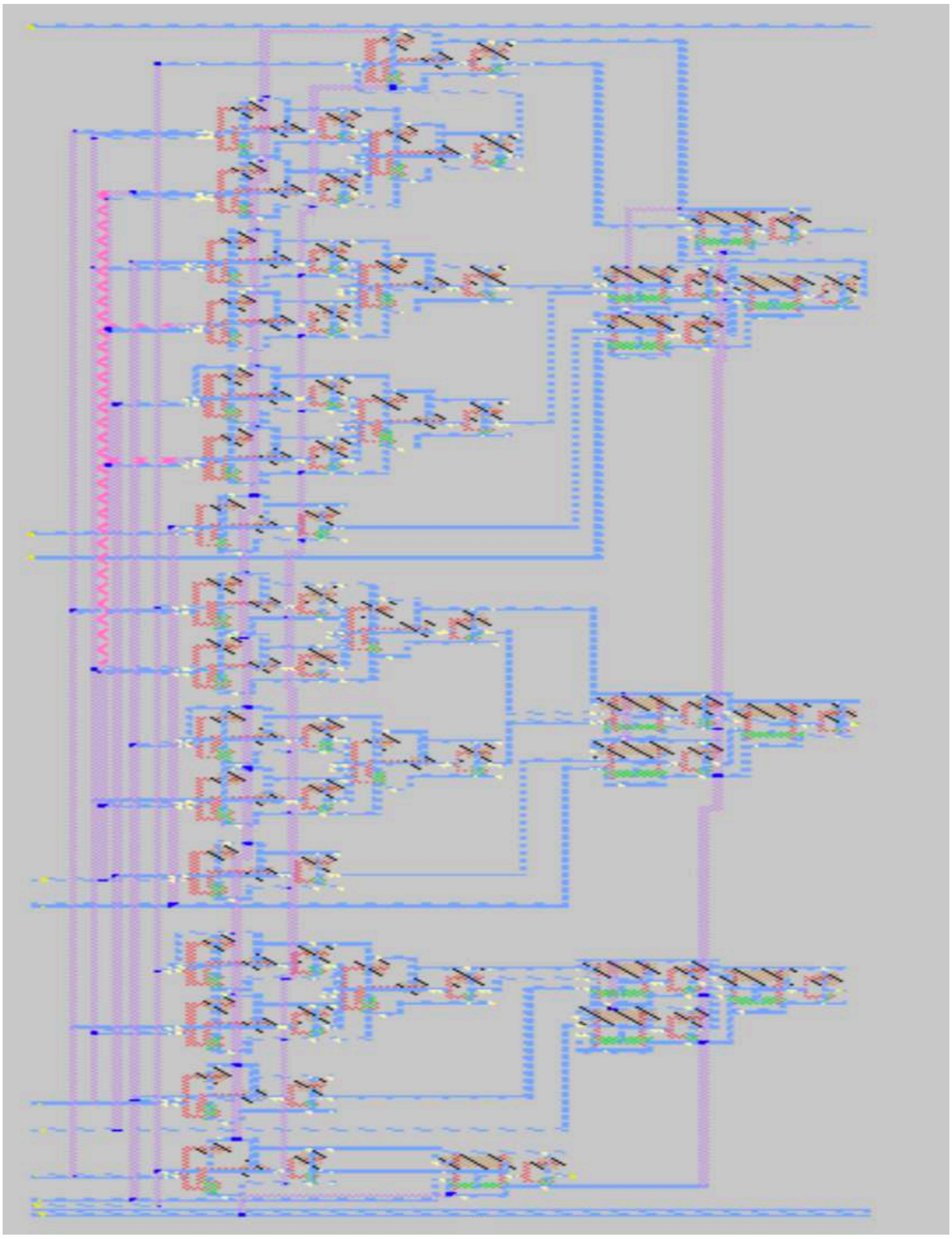




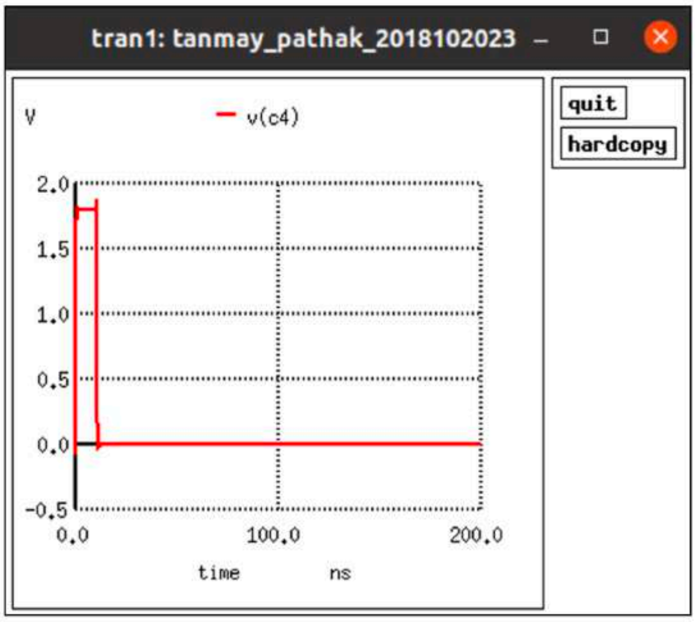


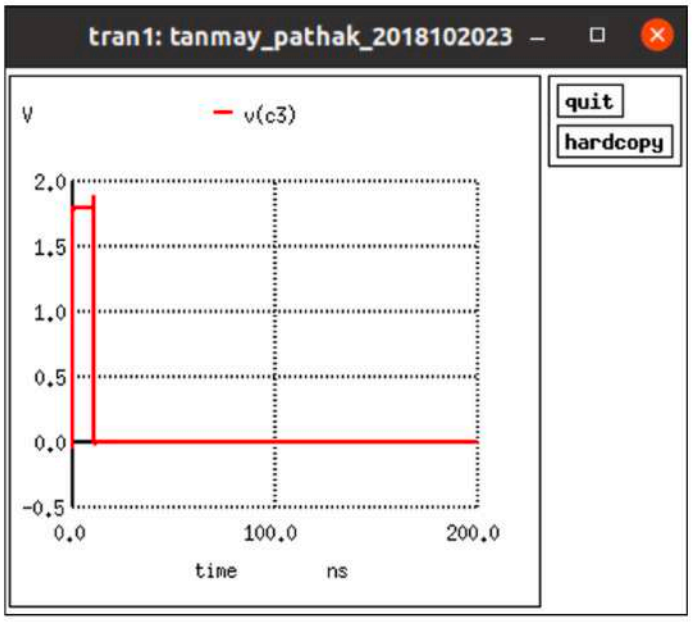


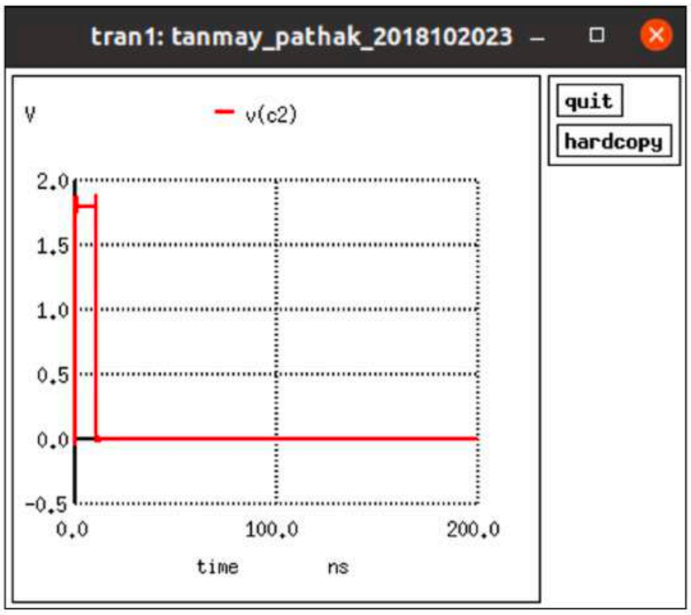
3) Carry Look Ahead Block

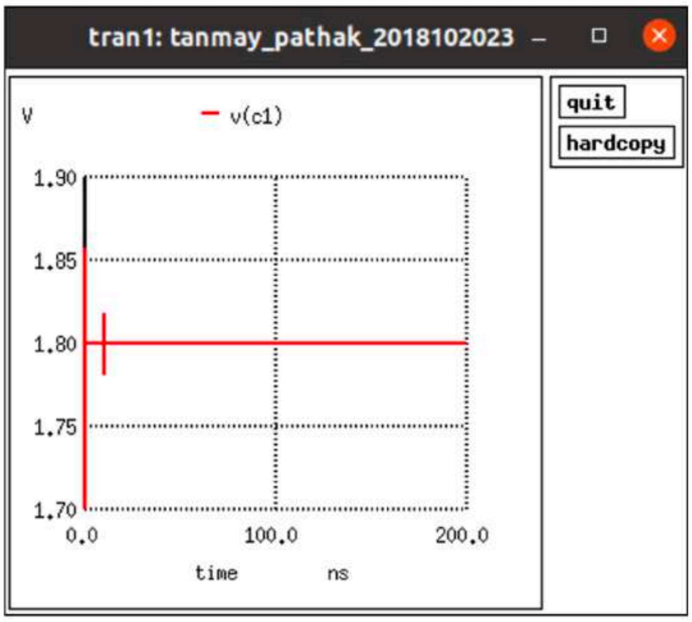


OUTPUT - Magic layout simulation

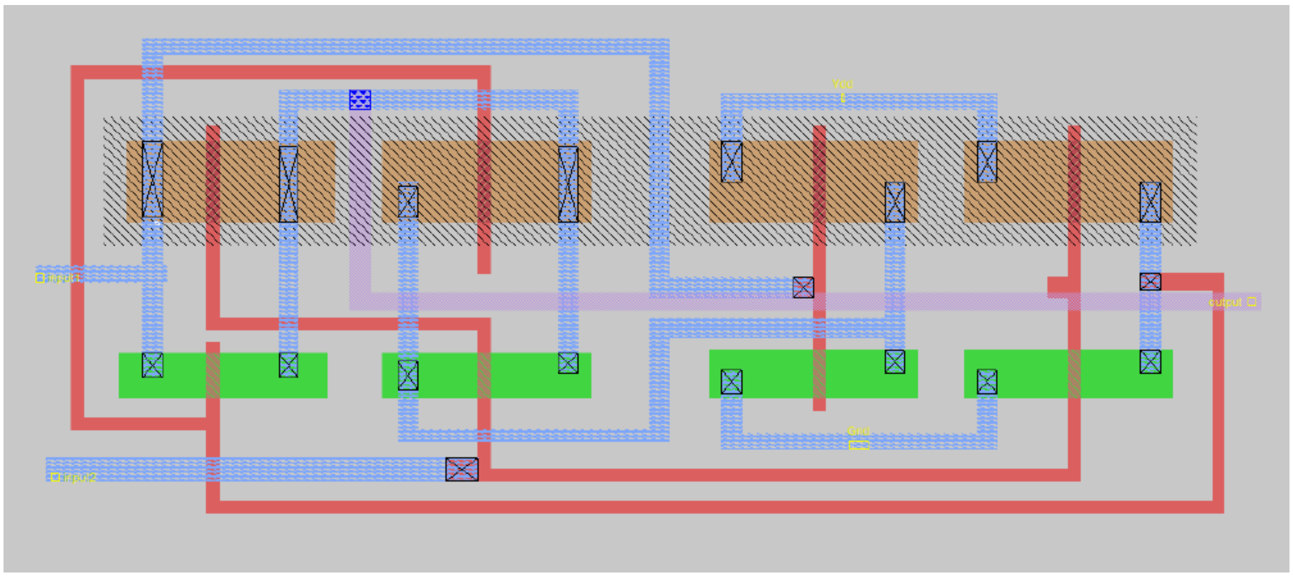




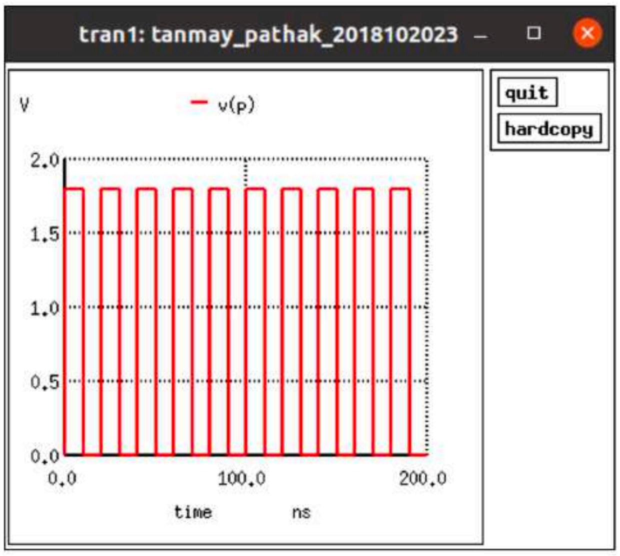


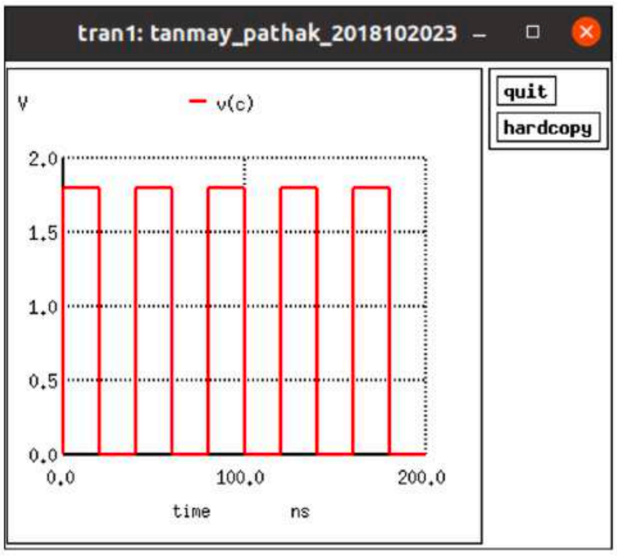


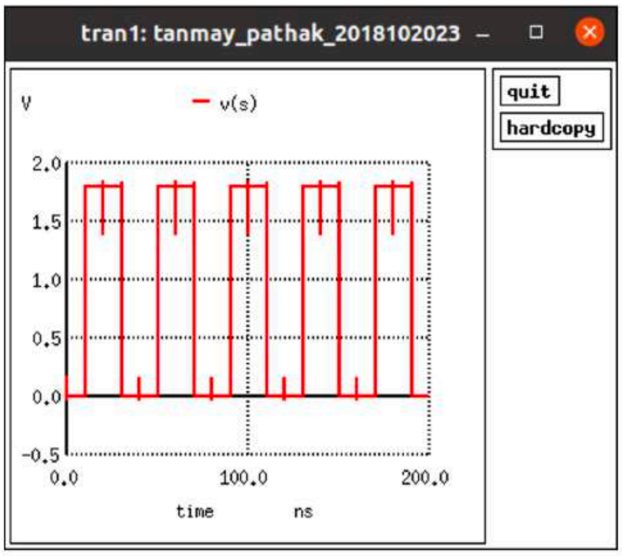
4) SUM Block



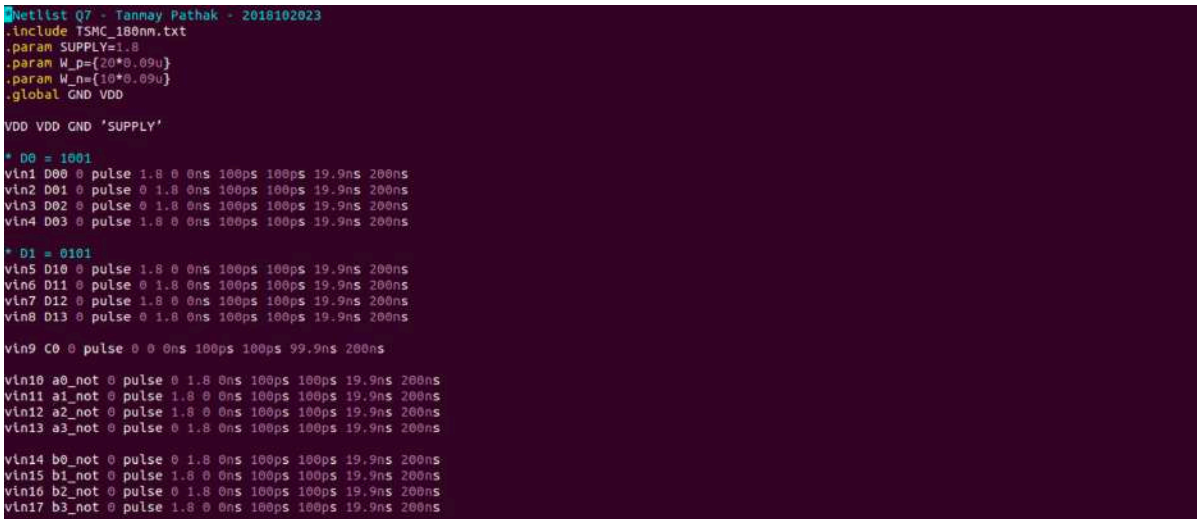
OUTPUT - Magic layout simulation



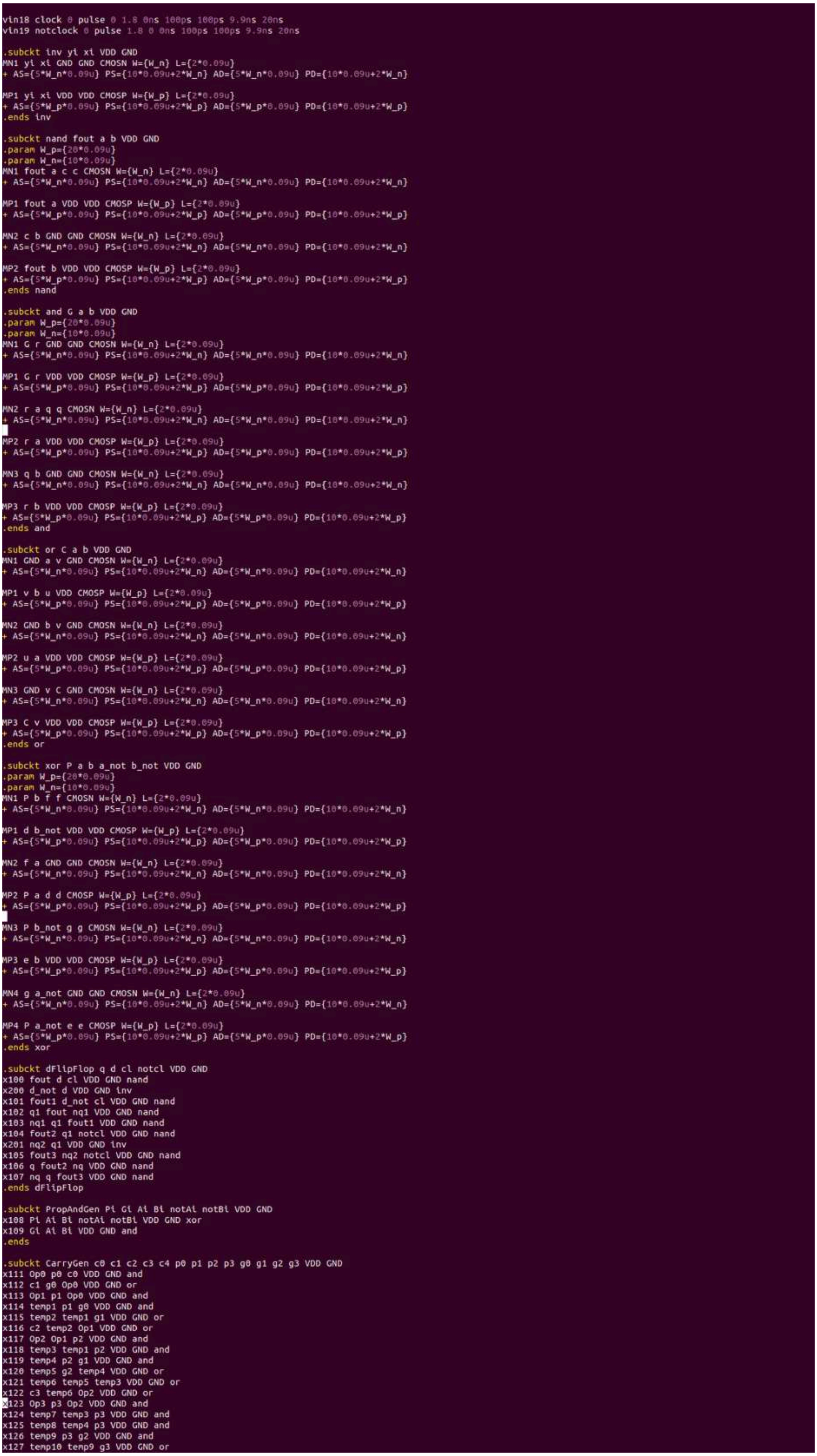




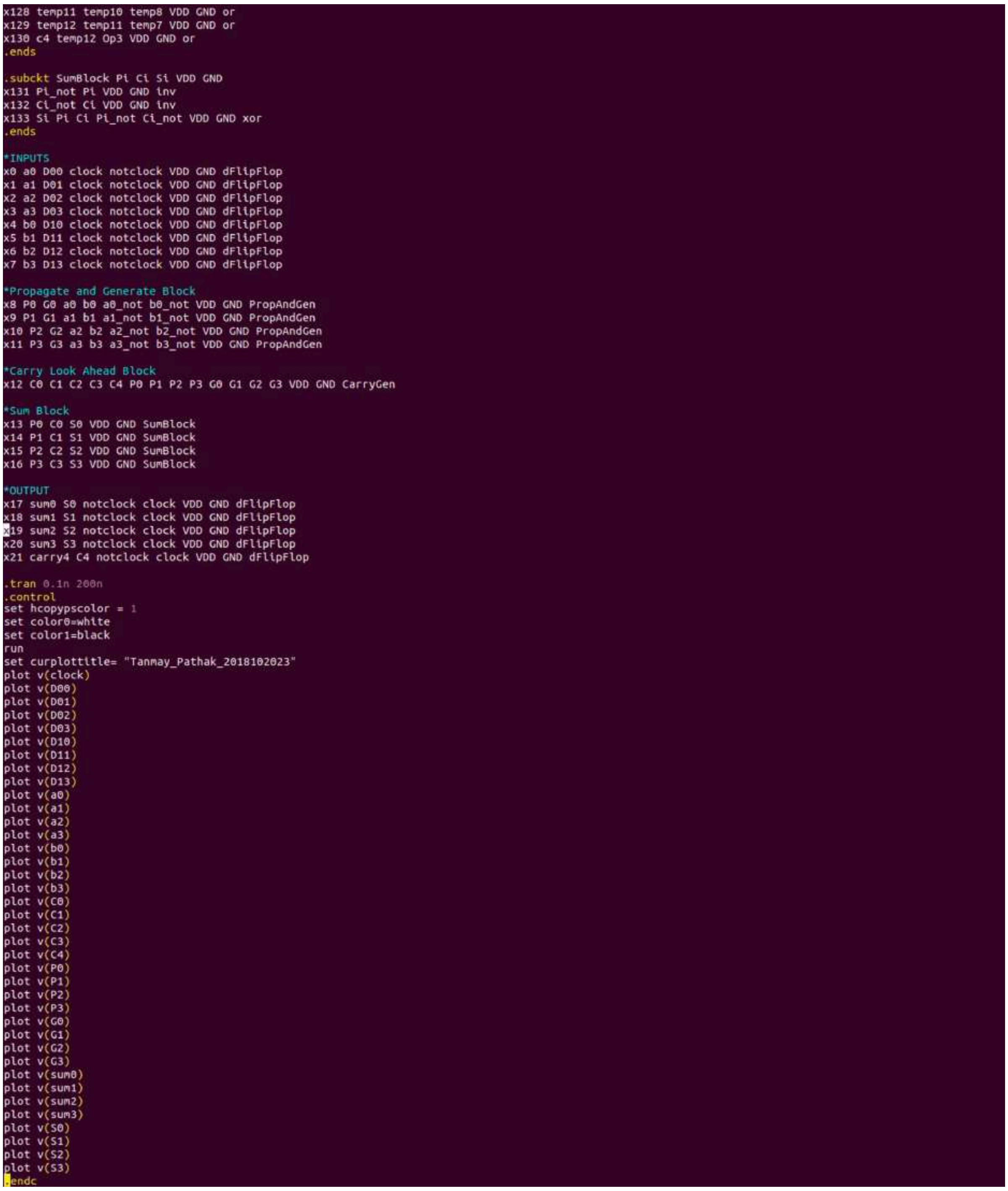
Question 7) - Verification using NGSPICE simulations



```
*Netlist Q7 - Tanmay Pathak - 2018102023
.include TSMC_180nm.txt
.param SUPPLY=1.8
.param W_p={20*0.09u}
.param W_n={10*0.09u}
.global GND VDD

VDD VDD GND 'SUPPLY'

* D0 = 1001
vin1 D00 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin2 D01 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin3 D02 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin4 D03 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns

* D1 = 0101
vin5 D10 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin6 D11 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin7 D12 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin8 D13 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns

vin9 C0 0 pulse 0 0 0ns 100ps 100ps 99.9ns 200ns

vin10 a0_not 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin11 a1_not 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin12 a2_not 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin13 a3_not 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns

vin14 b0_not 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin15 b1_not 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
vin16 b2_not 0 pulse 0 1.8 0ns 100ps 100ps 19.9ns 200ns
vin17 b3_not 0 pulse 1.8 0 0ns 100ps 100ps 19.9ns 200ns
```

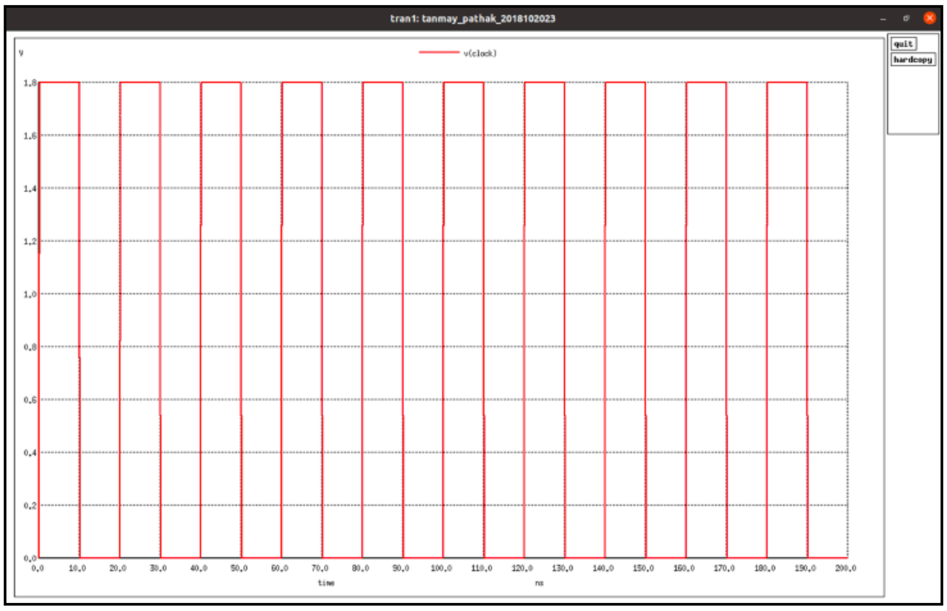


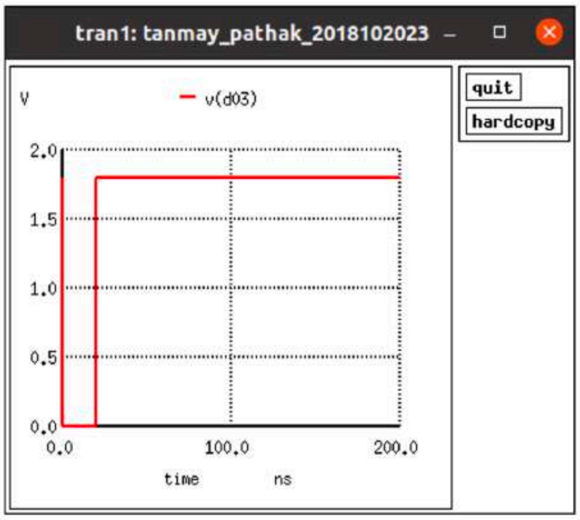
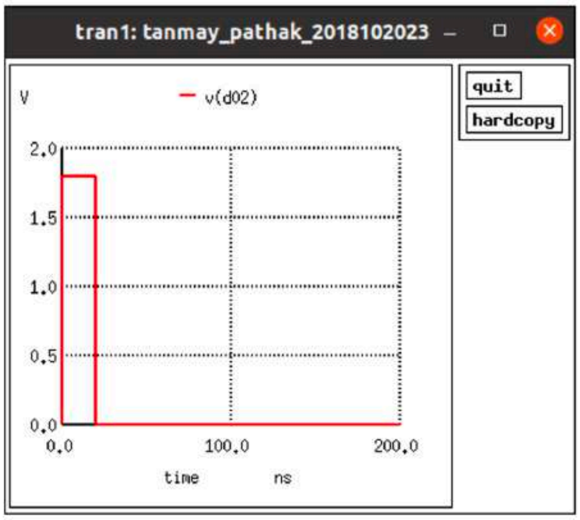
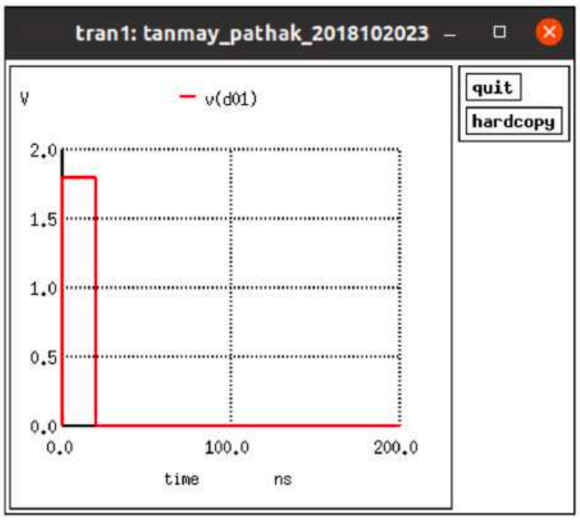
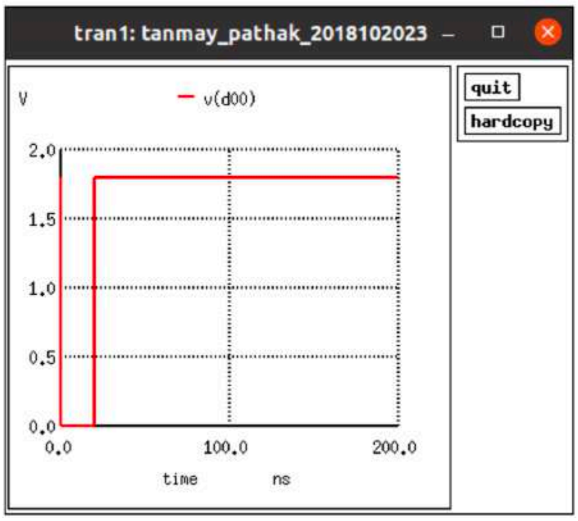
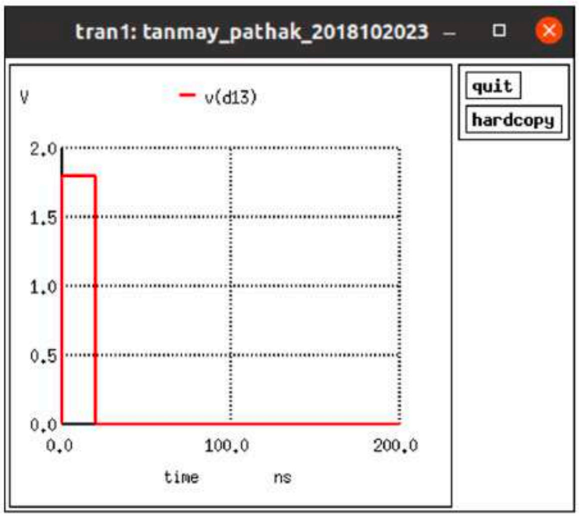
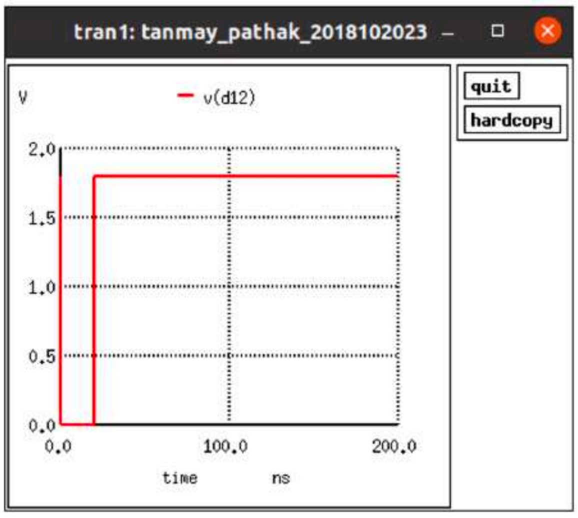
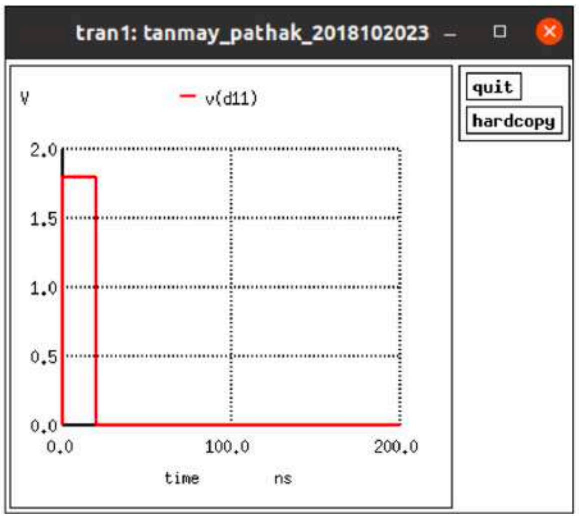
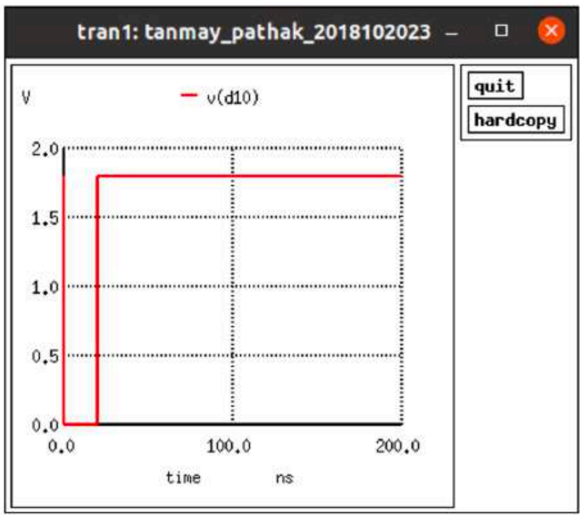
```
vin18 clock 0 pulse 0 1.8 0ns 100ps 100ps 9.9ns 20ns
vin19 notclock 0 pulse 1.8 0 0ns 100ps 100ps 9.9ns 20ns

.subckt inv yi xi VDD GND
MN1 yi xi GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 yi xi VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends inv

.subckt nand fout a b VDD GND
.param W_p={20*0.09u}
.param W_n={10*0.09u}
MN1 fout a c c CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 fout a VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 c b GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 fout b VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends nand

.subckt and G a b VDD GND
.param W_p={20*0.09u}
.param W_n={10*0.09u}
MN1 G r GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 G r VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 r a q q CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 r a VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN3 q b GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP3 r b VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends and

.subckt or C a b VDD GND
MN1 GND a v GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 v b u VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 GND b v GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 u a VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN3 GND v C GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP3 C v VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends or

.subckt xor P a b a_not b_not VDD GND
.param W_p={20*0.09u}
.param W_n={10*0.09u}
MN1 P b f f CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP1 d b_not VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN2 f a GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP2 P a d d CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN3 P b_not g g CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP3 e b VDD VDD CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}

MN4 g a_not GND GND CMOSN W={W_n} L={2*0.09u}
+ AS={5*W_n*0.09u} PS={10*0.09u+2*W_n} AD={5*W_n*0.09u} PD={10*0.09u+2*W_n}

MP4 P a_not e e CMOSP W={W_p} L={2*0.09u}
+ AS={5*W_p*0.09u} PS={10*0.09u+2*W_p} AD={5*W_p*0.09u} PD={10*0.09u+2*W_p}
.ends xor

.subckt dFlipFlop q d cl notcl VDD GND
x100 fout d cl VDD GND nand
x200 d_not d VDD GND inv
x101 fout1 d_not cl VDD GND nand
x102 q1 fout nq1 VDD GND nand
x103 nq1 q1 fout1 VDD GND nand
x104 fout2 q1 notcl VDD GND nand
x201 nq2 q1 VDD GND inv
x105 fout3 nq2 notcl VDD GND nand
x106 q fout2 nq VDD GND nand
x107 nq q fout3 VDD GND nand
.ends dFlipFlop

.subckt PropAndGen Pi Gi Ai Bi notAi notBi VDD GND
x108 Pi Ai Bi notAi notBi VDD GND xor
x109 Gi Ai Bi VDD GND and
.ends

.subckt CarryGen c0 c1 c2 c3 c4 p0 p1 p2 p3 g0 g1 g2 g3 VDD GND
x111 Op0 p0 c0 VDD GND and
x112 c1 g0 Op0 VDD GND or
x113 Op1 p1 Op0 VDD GND and
x114 temp1 p1 g0 VDD GND and
x115 temp2 temp1 g1 VDD GND or
x116 c2 temp2 Op1 VDD GND or
x117 Op2 Op1 p2 VDD GND and
x118 temp3 temp1 p2 VDD GND and
x119 temp4 p2 g1 VDD GND and
x120 temp5 g2 temp4 VDD GND or
x121 temp6 temp5 temp3 VDD GND or
x122 c3 temp6 Op2 VDD GND or
x123 Op3 p3 Op2 VDD GND and
x124 temp7 temp3 p3 VDD GND and
x125 temp8 temp4 p3 VDD GND and
x126 temp9 p3 g2 VDD GND and
x127 temp10 temp9 g3 VDD GND or
```



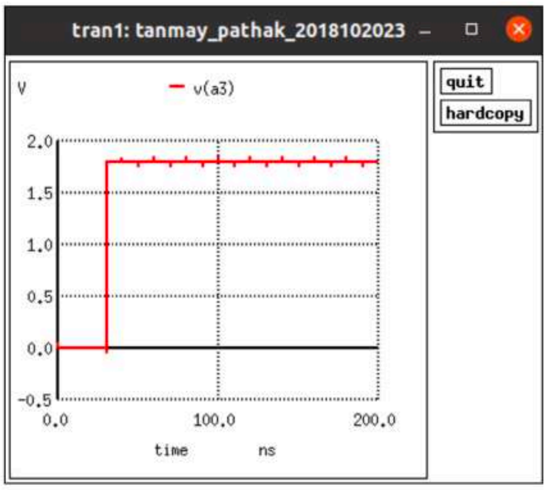
```
x128 temp11 temp10 temp8 VDD GND or
x129 temp12 temp11 temp7 VDD GND or
x130 c4 temp12 Op3 VDD GND or
.ends

.subckt SumBlock Pi Ci Si VDD GND
x131 Pi_not Pi VDD GND inv
x132 Ci_not Ci VDD GND inv
x133 Si Pi Ci Pi_not Ci_not VDD GND xor
.ends

*INPUTS
x0 a0 D00 clock notclock VDD GND dFlipFlop
x1 a1 D01 clock notclock VDD GND dFlipFlop
x2 a2 D02 clock notclock VDD GND dFlipFlop
x3 a3 D03 clock notclock VDD GND dFlipFlop
x4 b0 D10 clock notclock VDD GND dFlipFlop
x5 b1 D11 clock notclock VDD GND dFlipFlop
x6 b2 D12 clock notclock VDD GND dFlipFlop
x7 b3 D13 clock notclock VDD GND dFlipFlop

*Propagate and Generate Block
x8 P0 G0 a0 b0 a0_not b0_not VDD GND PropAndGen
x9 P1 G1 a1 b1 a1_not b1_not VDD GND PropAndGen
x10 P2 G2 a2 b2 a2_not b2_not VDD GND PropAndGen
x11 P3 G3 a3 b3 a3_not b3_not VDD GND PropAndGen

*Carry Look Ahead Block
x12 C0 C1 C2 C3 C4 P0 P1 P2 P3 G0 G1 G2 G3 VDD GND CarryGen

*Sum Block
x13 P0 C0 S0 VDD GND SumBlock
x14 P1 C1 S1 VDD GND SumBlock
x15 P2 C2 S2 VDD GND SumBlock
x16 P3 C3 S3 VDD GND SumBlock

*OUTPUT
x17 sum0 S0 notclock clock VDD GND dFlipFlop
x18 sum1 S1 notclock clock VDD GND dFlipFlop
x19 sum2 S2 notclock clock VDD GND dFlipFlop
x20 sum3 S3 notclock clock VDD GND dFlipFlop
x21 carry4 C4 notclock clock VDD GND dFlipFlop

.tran 0.1n 200n
.control
set hcopypscolor = 1
set color0=white
set color1=black
run
set curplottitle= "Tanmay_Pathak_2018102023"
plot v(clock)
plot v(D00)
plot v(D01)
plot v(D02)
plot v(D03)
plot v(D10)
plot v(D11)
plot v(D12)
plot v(D13)
plot v(a0)
plot v(a1)
plot v(a2)
plot v(a3)
plot v(b0)
plot v(b1)
plot v(b2)
plot v(b3)
plot v(C0)
plot v(C1)
plot v(C2)
plot v(C3)
plot v(C4)
plot v(P0)
plot v(P1)
plot v(P2)
plot v(P3)
plot v(G0)
plot v(G1)
plot v(G2)
plot v(G3)
plot v(sum0)
plot v(sum1)
plot v(sum2)
plot v(sum3)
plot v(S0)
plot v(S1)
plot v(S2)
plot v(S3)
.endc
```

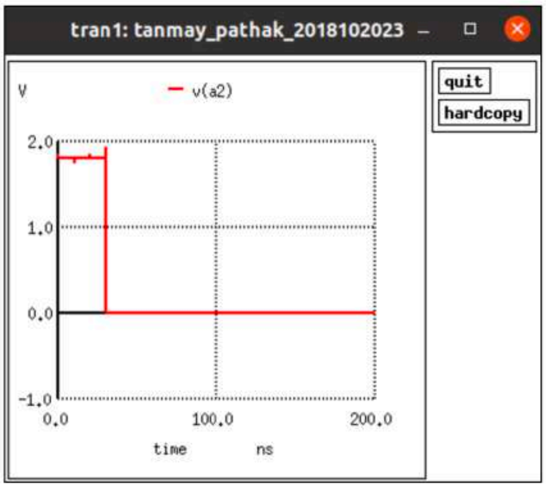
INPUT Plots

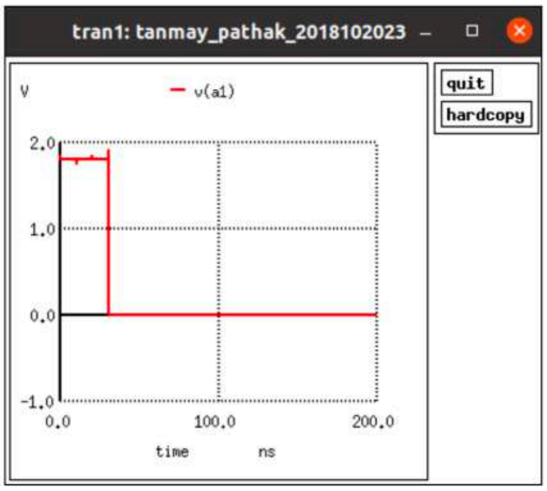


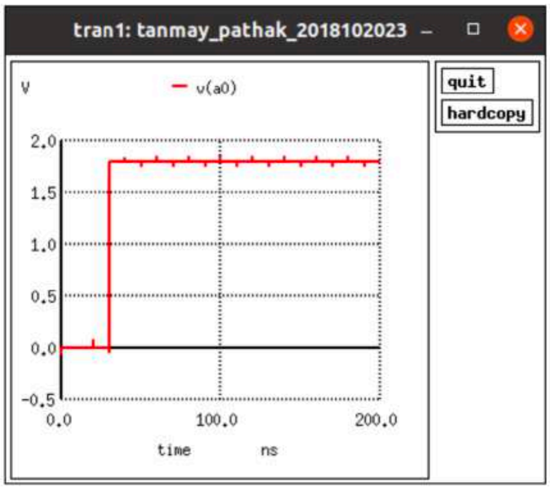

tran1: tanmay_pathak_2018102023
quit
hardcopy
V
v(d03)
2.0
1.5
1.0
0.5
0.0
0.0
100.0
200.0
time
ns

tran1: tanmay_pathak_2018102023
quit
hardcopy
V
v(d02)
2.0
1.5
1.0
0.5
0.0
0.0
100.0
200.0
time
ns

tran1: tanmay_pathak_2018102023
quit
hardcopy
V
v(d01)
2.0
1.5
1.0
0.5
0.0
0.0
100.0
200.0
time
ns

tran1: tanmay_pathak_2018102023
quit
hardcopy
V
v(d00)
2.0
1.5
1.0
0.5
0.0
0.0
100.0
200.0
time
ns

tran1: tanmay_pathak_2018102023
quit
hardcopy
V
v(d13)
2.0
1.5
1.0
0.5
0.0
0.0
100.0
200.0
time
ns

tran1: tanmay_pathak_2018102023
quit
hardcopy
V
v(d12)
2.0
1.5
1.0
0.5
0.0
0.0
100.0
200.0
time
ns

tran1: tanmay_pathak_2018102023
quit
hardcopy
V
v(d11)
2.0
1.5
1.0
0.5
0.0
0.0
100.0
200.0
time
ns

tran1: tanmay_pathak_2018102023
quit
hardcopy
V
v(d10)
2.0
1.5
1.0
0.5
0.0
0.0
100.0
200.0
time
ns

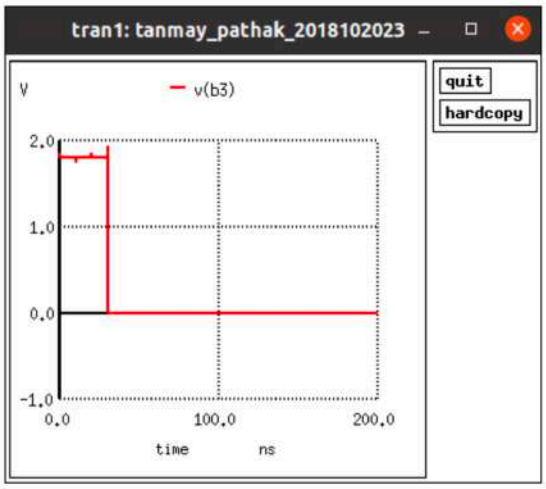
OUTPUT Plots - FlipFlops

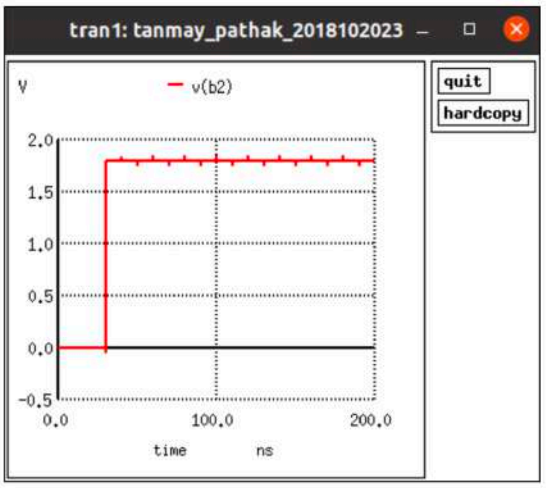


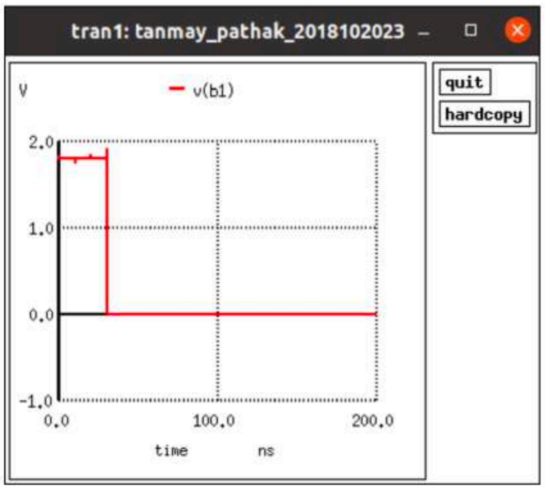


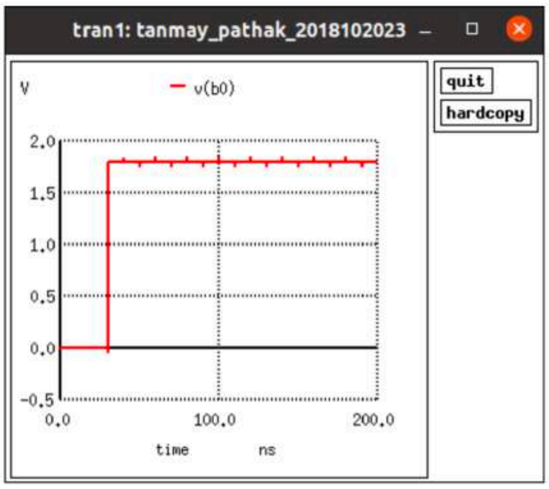




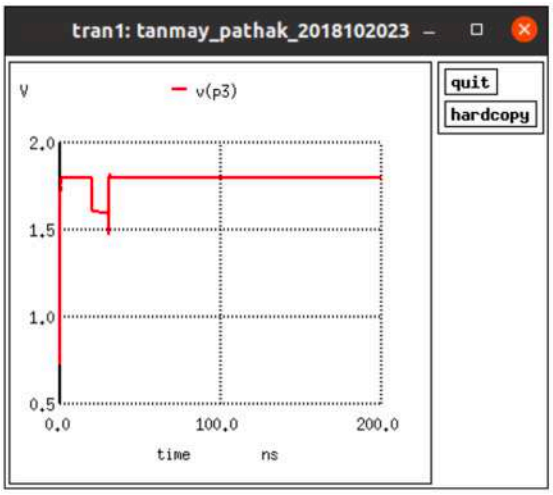


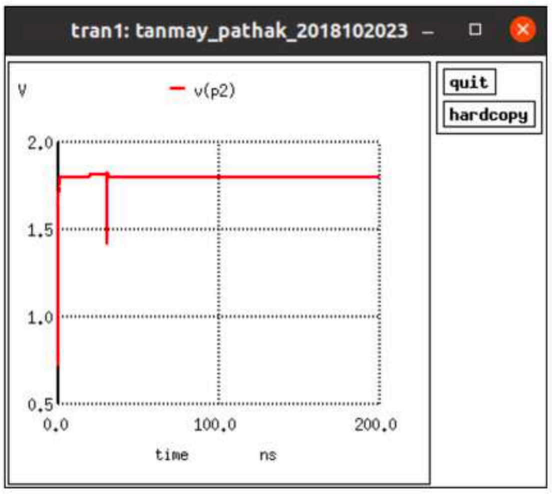


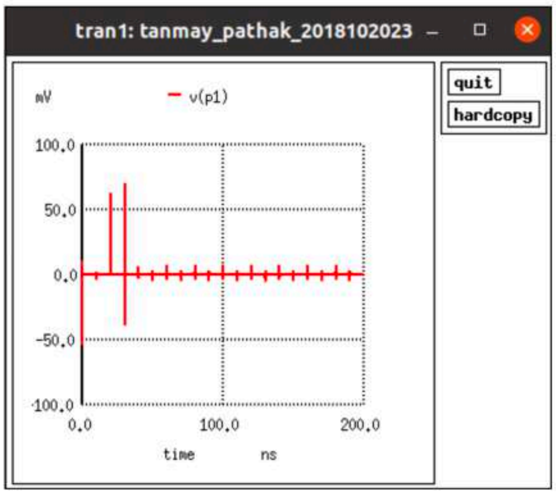


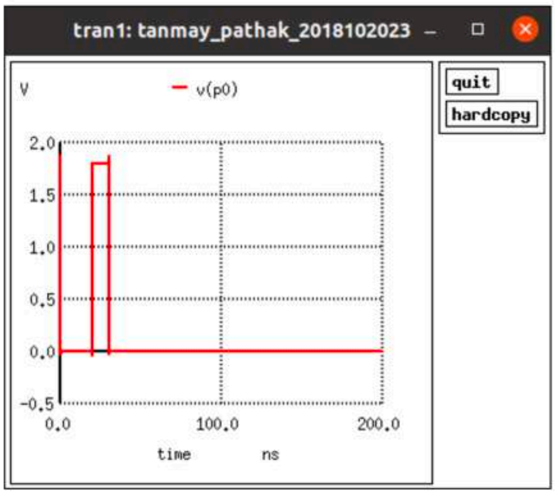


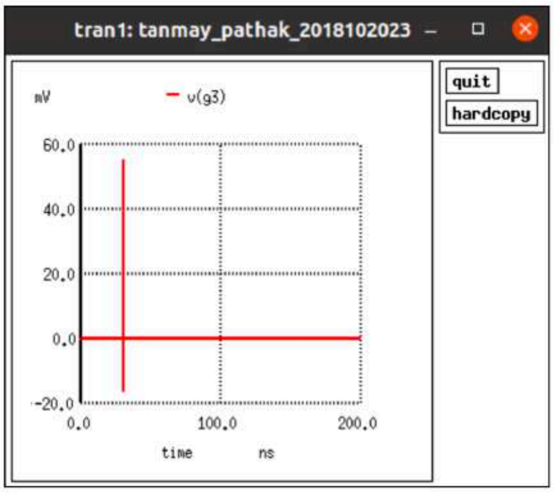
OUTPUT Plots - Propagate and generate block

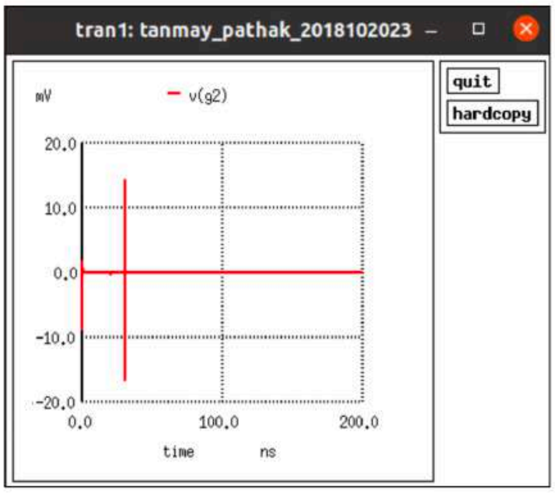


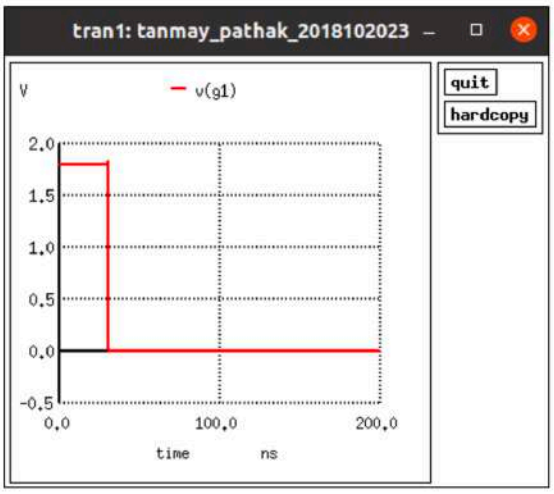


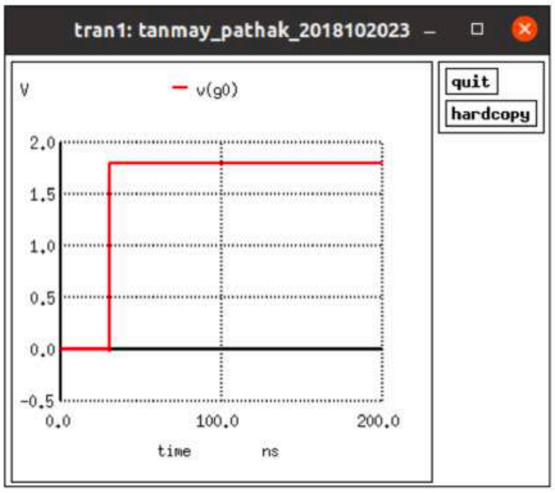




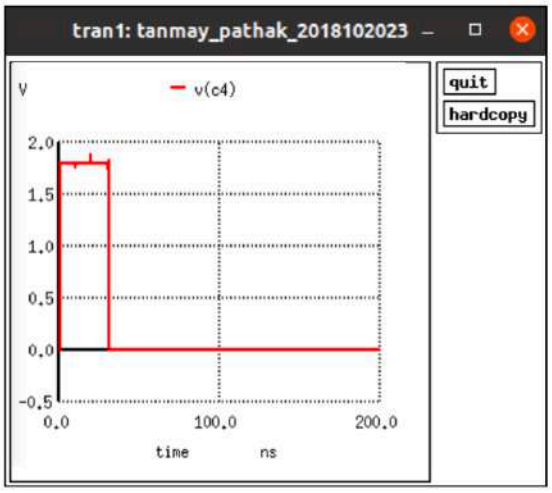


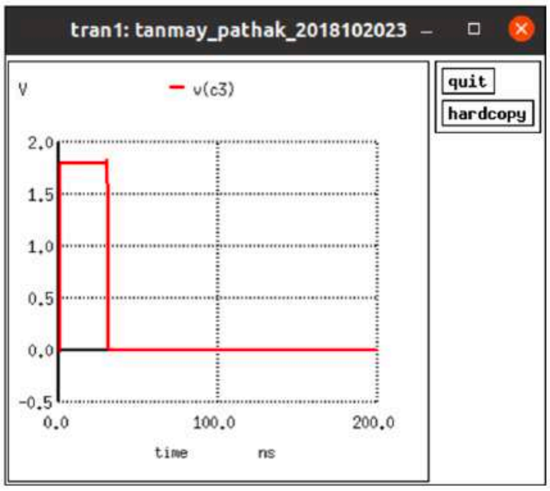


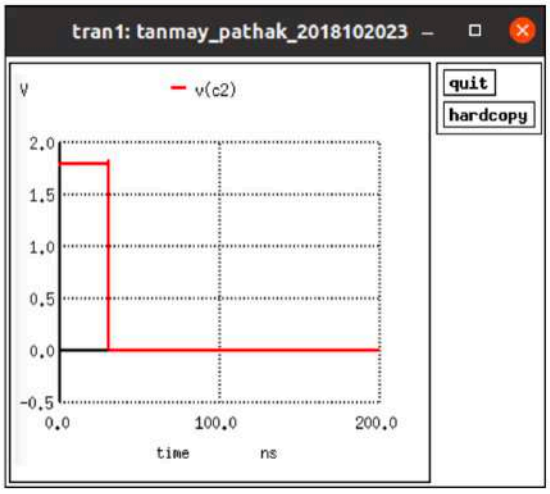


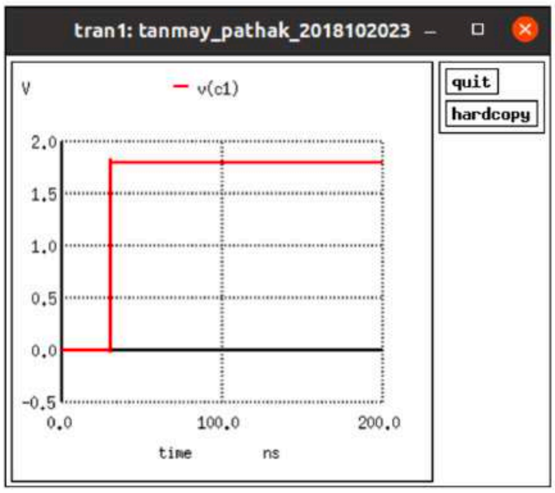


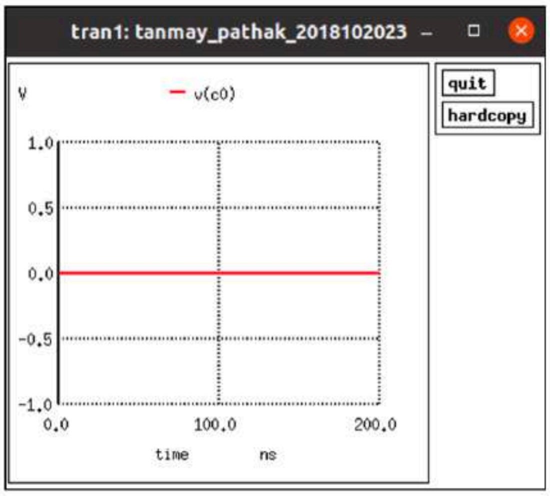
OUTPUT Plots - Carry Look Ahead Block



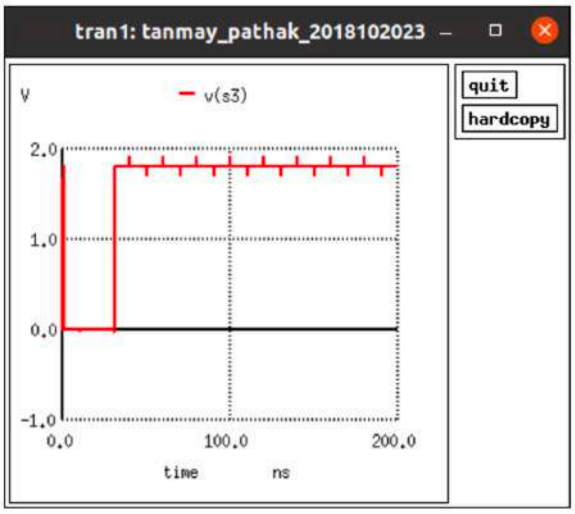


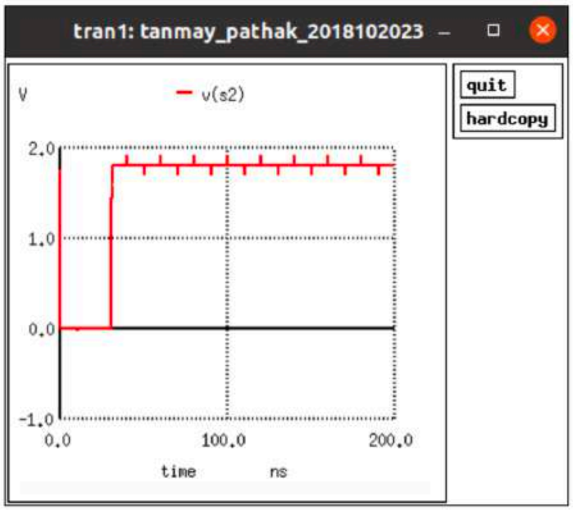


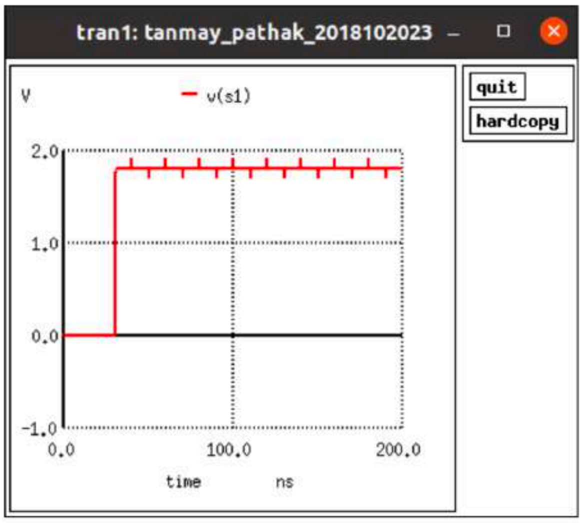


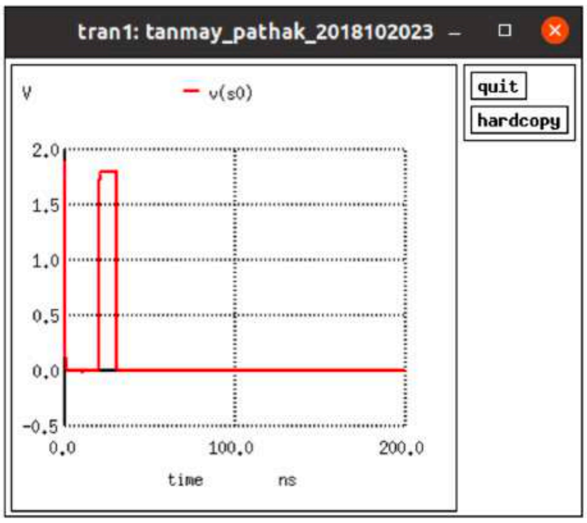


OUTPUT Plots - SUM Block

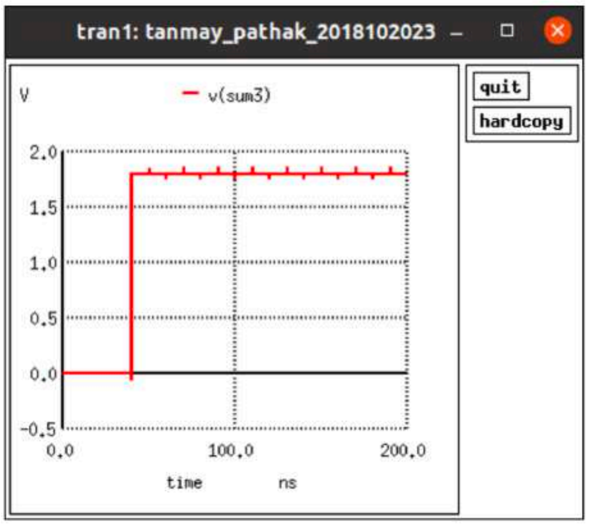


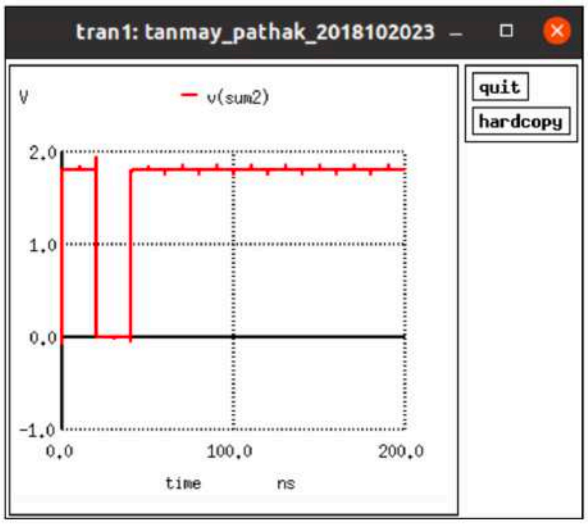


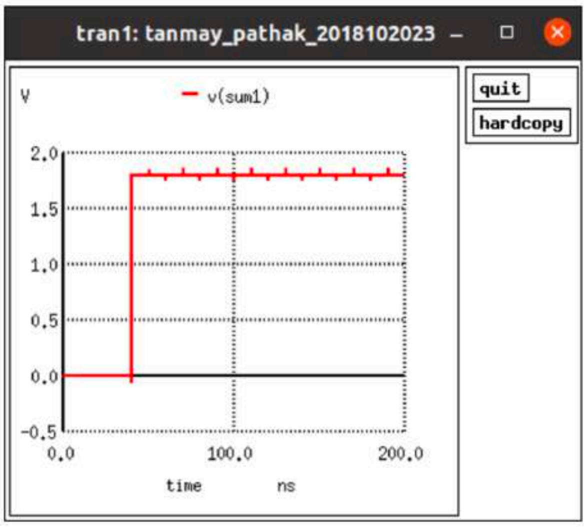


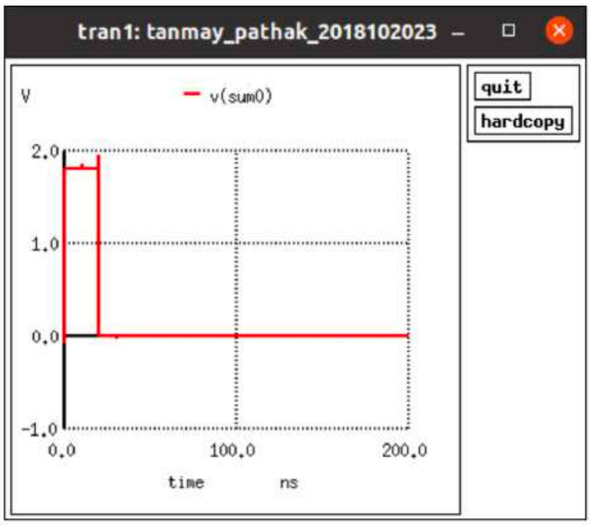


OUTPUT Plots - SUM Block after FlipFlop









Delay - This can be calculated using transient analysis

Gate / block		tpd	targ	trig
Flip Flop		1.014270E-08	3.019270E-08	2.005000E-08
Propagate and Generate	Propagate	1.313327E-10	3.032419E-08	3.019286E-08
	Generate	1.483470E-10	3.034105E-08	3.019270E-08
Carry Look Ahead		6.554758E-10	3.097967E-08	3.032419E-08
SUM		1.426915E-10	2.026890E-08	2.012621E-08
CLA Adder		2.011583E-08	4.016583E-08	2.005000E-08
MAX DELAY		**2.020175E-08**	**4.025175E-08**	**2.005000E-08**

1) FlipFlop - The delay will be equal to the clock pulse duration.
2) Propagate and Generate - Since both propagate and generate blocks run in parallel, delay will be max of delay in both of them which is equal to 0.14ns
3) Carry Look Ahead - One bit goes through a maximum of 5 gates (2 AND gates and 3 OR gates) and thus we estimate the delay to be 5 x delay for one gate. The actual delay was 0.65ns
4) SUM - This is a simple XOR block and the delay was 0.14ns
5) CLA Adder - We estimate the delay to be 2 x (delay in input and output flip-flop) + the delay in other blocks. The delay in this case was 20.11ns
6) MAX DELAY - The worst case delay was 20.201ns

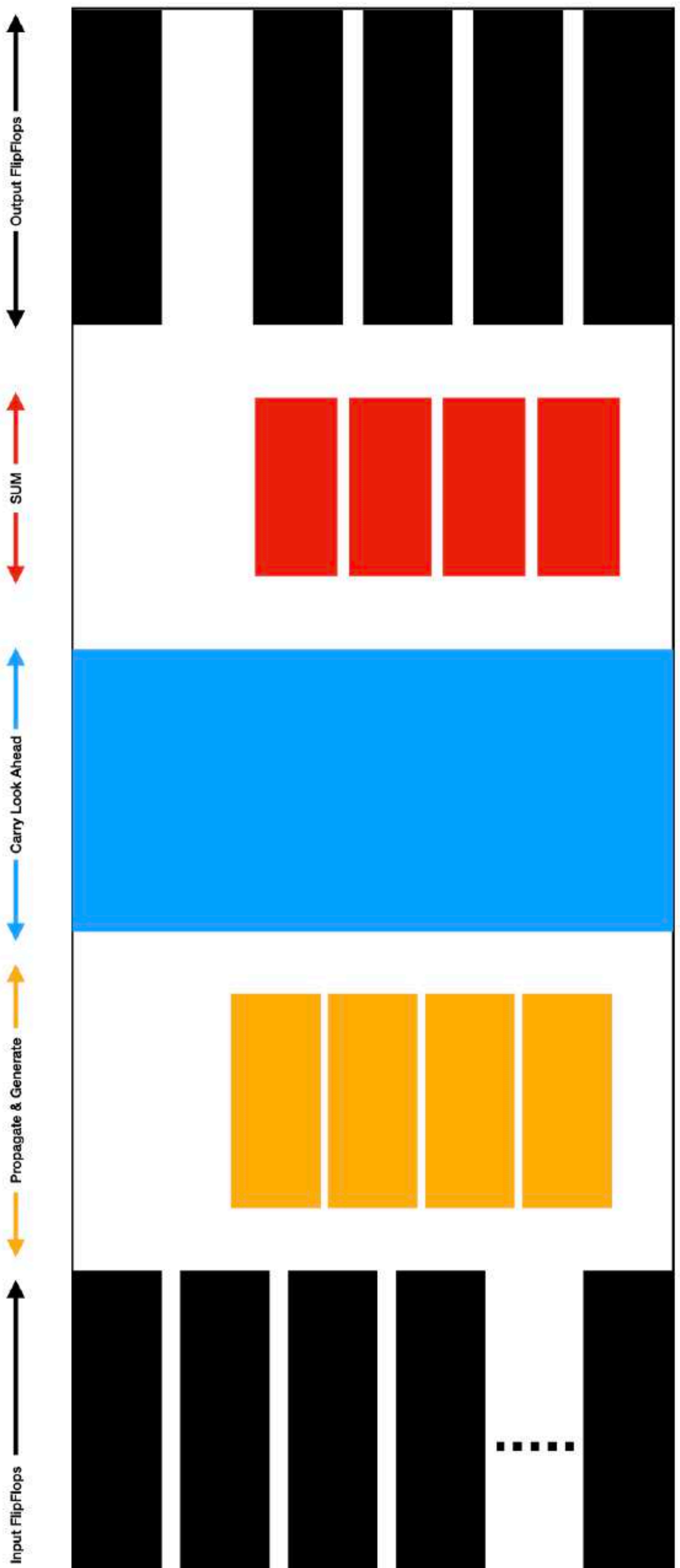
Question 8) Floor Planning

By performing floor planning we aim to minimize the area used while retaining the easy routability (refers to the maintenance of proper spacing between the connecting metals keeping in mind the losses).

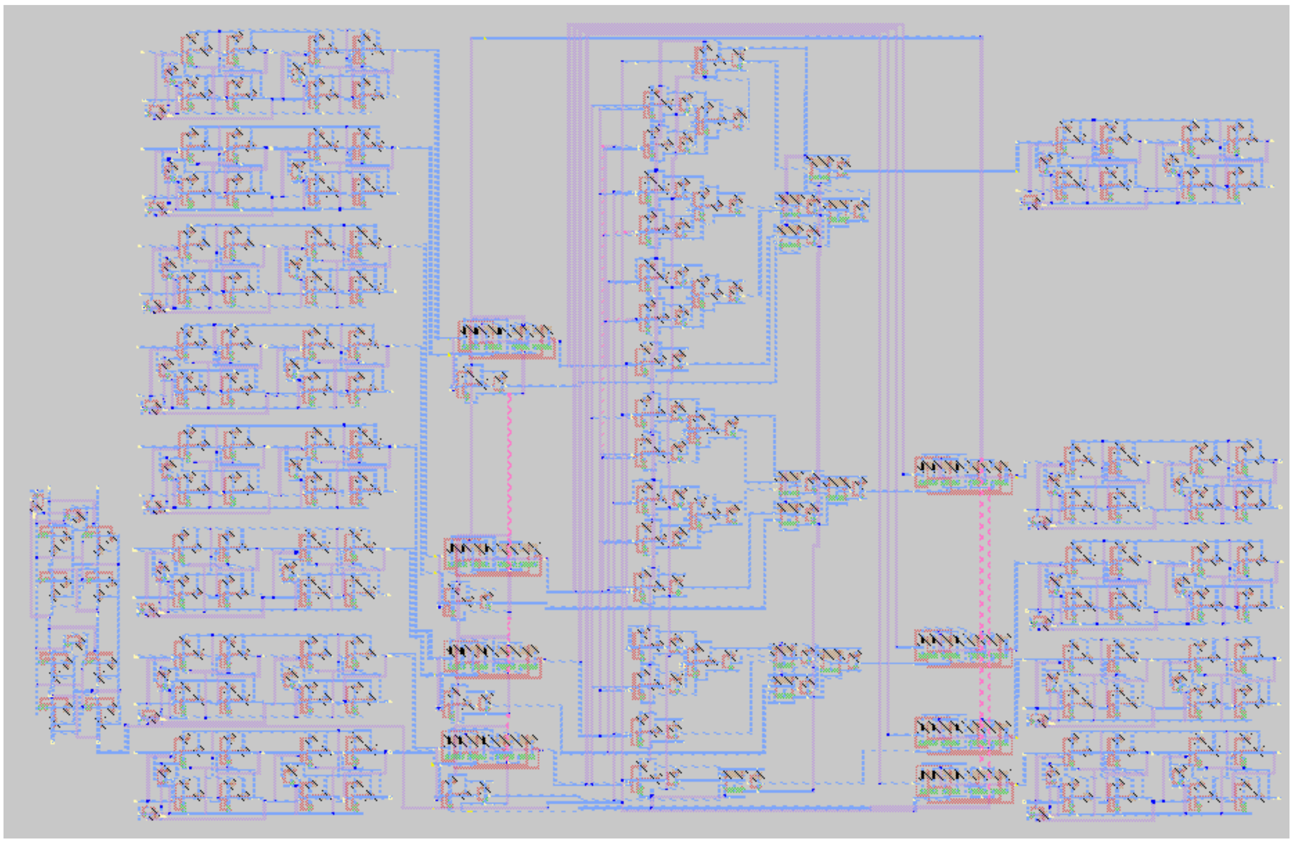
We maintain similarity in shape of all blocks and gates and keep them rectangularly shaped. To maintain flow of the diagram, the inputs are taken towards the left side and the output is given at the right hand side of the diagram

The biggest non-breakable component is the CLA Block thus this block should determine the vertical size of the floor plan. But stacking all the flip flops exceeds the height of the CLA block, thus we place some (n) of the flip flops adjacent to the flip flop stack(total-n).

The Floor plan is given on the next page


Output FlipFlops
SUM
Carry Look Ahead
Propagate & Generate
Input FlipFlops

Question 9) MAGIC Layout of the entire circuit



This is the MAGIC layout of the complete circuit. The leftmost vertical flip flop takes c_0. Adjacent to it an 8 bit Flip flop block, taking two 4-Bit inputs. Next comes a Propagate and Generate block which also takes in two 4-Bit inputs and gives out two 4-Bit outputs. Next, the CLA block takes in two 4 Bit inputs and gives a 4-Bit output. Next the SUM Block takes in two 4-Bit inputs and gives a 4-Bit output. Finally, we have the last Flip Flop block, termed as output flip flop block that takes in a 4-Bit input and gives a 4-Bit output.

The block wise simulations were shown before.

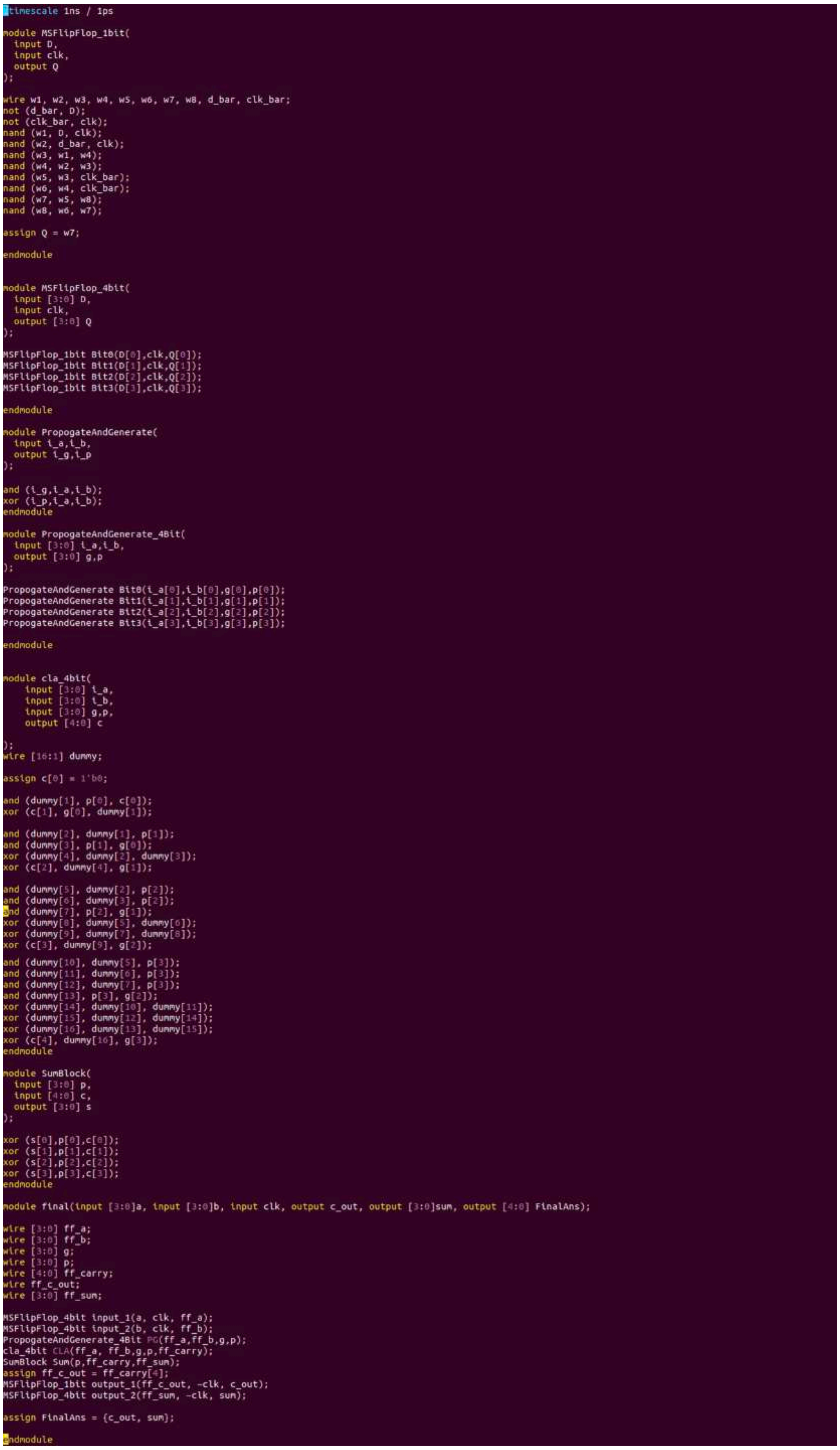
Question 10) Delay in CLA adder

The delay for the CLA adder was calculated by performing transient analysis and was also reported previously. **Delay = 20.11ns**

	CLA adder
tpd	2.011583E-08
targ	4.016583E-08
trig	2.005E-08

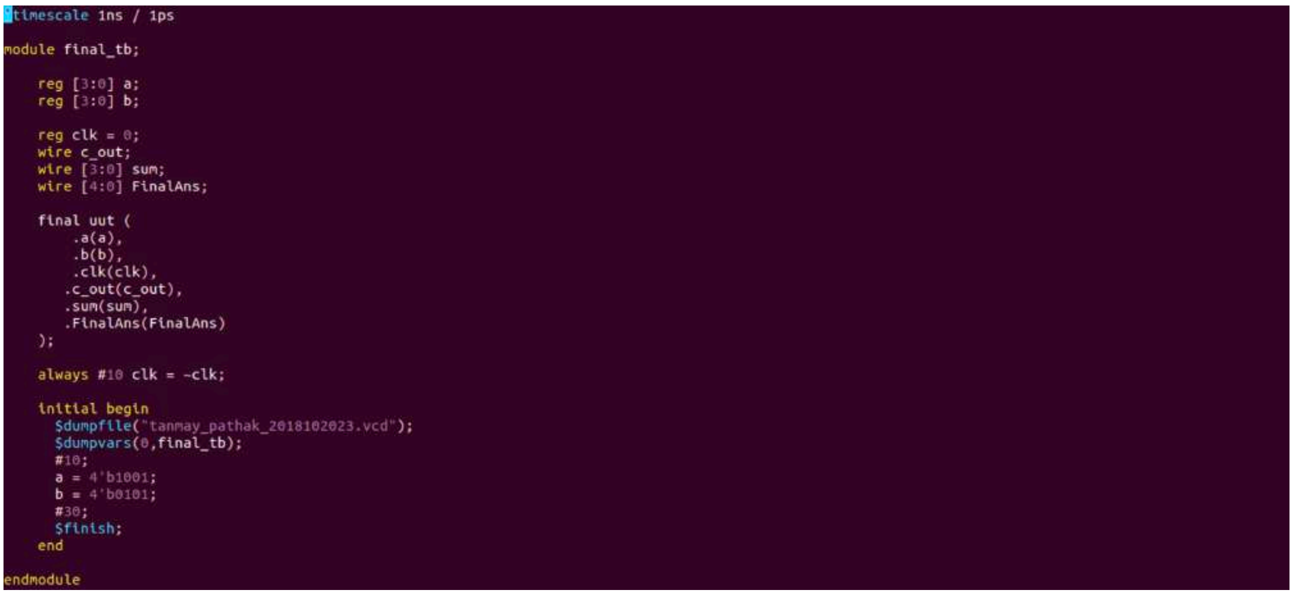
Question 11) Verilog Simulation and verification

The Verilog code used is given below



```verilog
`timescale 1ns / 1ps

module MSFlipFlop_1bit(
  input D,
  input clk,
  output Q
);

wire w1, w2, w3, w4, w5, w6, w7, w8, d_bar, clk_bar;
not (d_bar, D);
not (clk_bar, clk);
nand (w1, D, clk);
nand (w2, d_bar, clk);
nand (w3, w1, w4);
nand (w4, w2, w3);
nand (w5, w3, clk_bar);
nand (w6, w4, clk_bar);
nand (w7, w5, w8);
nand (w8, w6, w7);

assign Q = w7;

endmodule


module MSFlipFlop_4bit(
  input [3:0] D,
  input clk,
  output [3:0] Q
);

MSFlipFlop_1bit Bit0(D[0],clk,Q[0]);
MSFlipFlop_1bit Bit1(D[1],clk,Q[1]);
MSFlipFlop_1bit Bit2(D[2],clk,Q[2]);
MSFlipFlop_1bit Bit3(D[3],clk,Q[3]);

endmodule

module PropogateAndGenerate(
  input i_a,i_b,
  output i_g,i_p
);

and (i_g,i_a,i_b);
xor (i_p,i_a,i_b);
endmodule

module PropogateAndGenerate_4Bit(
  input [3:0] i_a,i_b,
  output [3:0] g,p
);

PropogateAndGenerate Bit0(i_a[0],i_b[0],g[0],p[0]);
PropogateAndGenerate Bit1(i_a[1],i_b[1],g[1],p[1]);
PropogateAndGenerate Bit2(i_a[2],i_b[2],g[2],p[2]);
PropogateAndGenerate Bit3(i_a[3],i_b[3],g[3],p[3]);

endmodule


module cla_4bit(
    input [3:0] i_a,
    input [3:0] i_b,
    input [3:0] g,p,
    output [4:0] c

);
wire [16:1] dummy;

assign c[0] = 1'b0;

and (dummy[1], p[0], c[0]);
xor (c[1], g[0], dummy[1]);

and (dummy[2], dummy[1], p[1]);
and (dummy[3], p[1], g[0]);
xor (dummy[4], dummy[2], dummy[3]);
xor (c[2], dummy[4], g[1]);

and (dummy[5], dummy[2], p[2]);
and (dummy[6], dummy[3], p[2]);
and (dummy[7], p[2], g[1]);
xor (dummy[8], dummy[5], dummy[6]);
xor (dummy[9], dummy[7], dummy[8]);
xor (c[3], dummy[9], g[2]);

and (dummy[10], dummy[5], p[3]);
and (dummy[11], dummy[6], p[3]);
and (dummy[12], dummy[7], p[3]);
and (dummy[13], p[3], g[2]);
xor (dummy[14], dummy[10], dummy[11]);
xor (dummy[15], dummy[12], dummy[14]);
xor (dummy[16], dummy[13], dummy[15]);
xor (c[4], dummy[16], g[3]);
endmodule

module SumBlock(
  input [3:0] p,
  input [4:0] c,
  output [3:0] s
);

xor (s[0],p[0],c[0]);
xor (s[1],p[1],c[1]);
xor (s[2],p[2],c[2]);
xor (s[3],p[3],c[3]);
endmodule

module final(input [3:0]a, input [3:0]b, input clk, output c_out, output [3:0]sum, output [4:0] FinalAns);

wire [3:0] ff_a;
wire [3:0] ff_b;
wire [3:0] g;
wire [3:0] p;
wire [4:0] ff_carry;
wire ff_c_out;
wire [3:0] ff_sum;

MSFlipFlop_4bit input_1(a, clk, ff_a);
MSFlipFlop_4bit input_2(b, clk, ff_b);
PropogateAndGenerate_4Bit PG(ff_a,ff_b,g,p);
cla_4bit CLA(ff_a, ff_b,g,p,ff_carry);
SumBlock Sum(p,ff_carry,ff_sum);
assign ff_c_out = ff_carry[4];
MSFlipFlop_1bit output_1(ff_c_out, ~clk, c_out);
MSFlipFlop_4bit output_2(ff_sum, ~clk, sum);

assign FinalAns = {c_out, sum};

endmodule
```

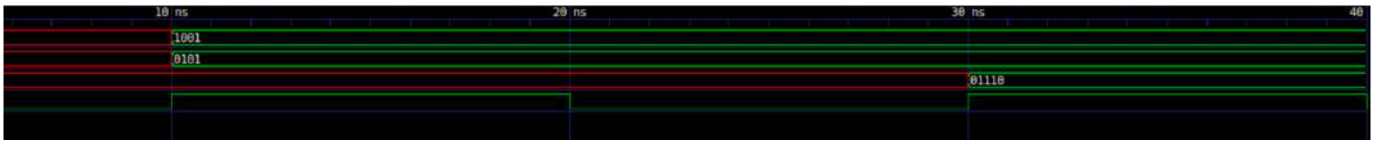
The test bench used is given below



```
`timescale 1ns / 1ps

module final_tb;

    reg [3:0] a;
    reg [3:0] b;

    reg clk = 0;
    wire c_out;
    wire [3:0] sum;
    wire [4:0] FinalAns;

    final uut (
        .a(a),
        .b(b),
        .clk(clk),
       .c_out(c_out),
       .sum(sum),
       .FinalAns(FinalAns)
    );

    always #10 clk = ~clk;

    initial begin
      $dumpfile("tanmay_pathak_2018102023.vcd");
      $dumpvars(0,final_tb);
      #10;
      a = 4'b1001;
      b = 4'b0101;
      #30;
      $finish;
    end

endmodule
```

GTKWave Output

(Inputs)
a = 1001 = 9
b = 0101 = 5



(Output - Expected)

Final_Ans = 01110 = 14

The inputs are given at the first positive edge of the clock while the answer is output at the second positive wave.

The output in the waves is 01110 = 14