

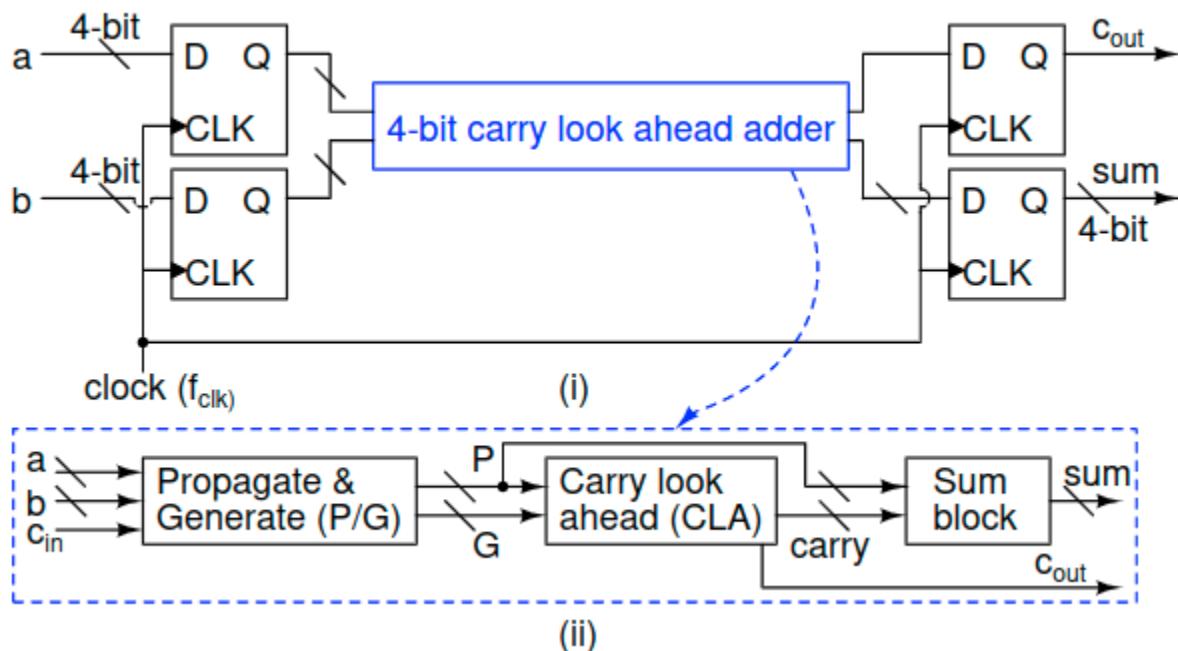
VLSI Project

Design a 4-bit carry look ahead (CLA) adder

Aravind Narayanan

2019102014

Block Structure:



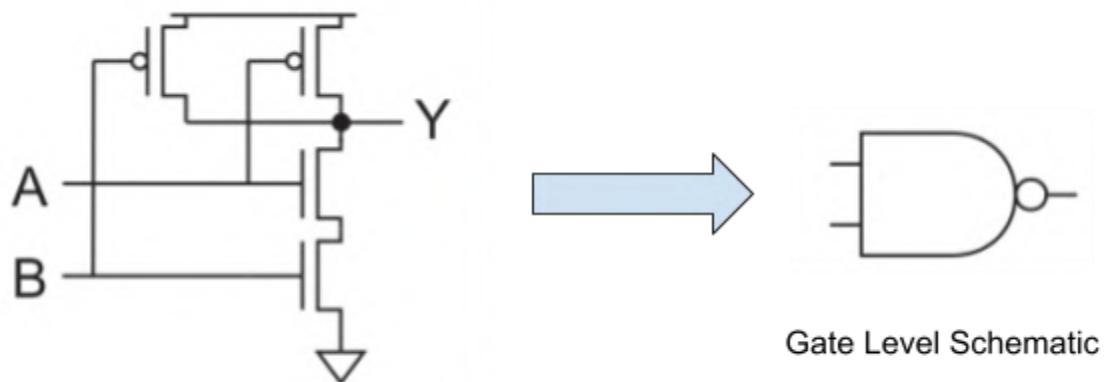
Inverter Gate Transistor Level:



Transistor Level of
Inverter(NOT) Gate

Gate-Level Symbol of
NOT Gate

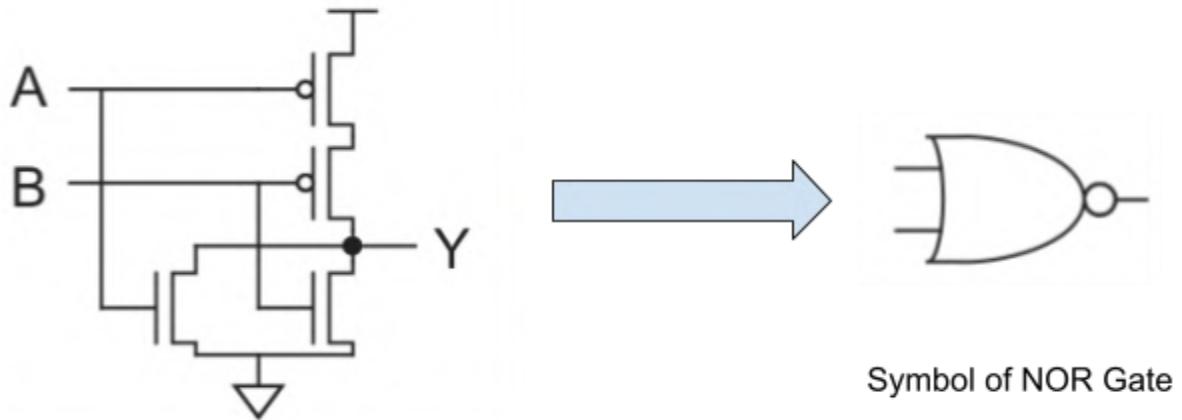
NAND Gate Transistor Level:



Transistor Level of NAND Gate

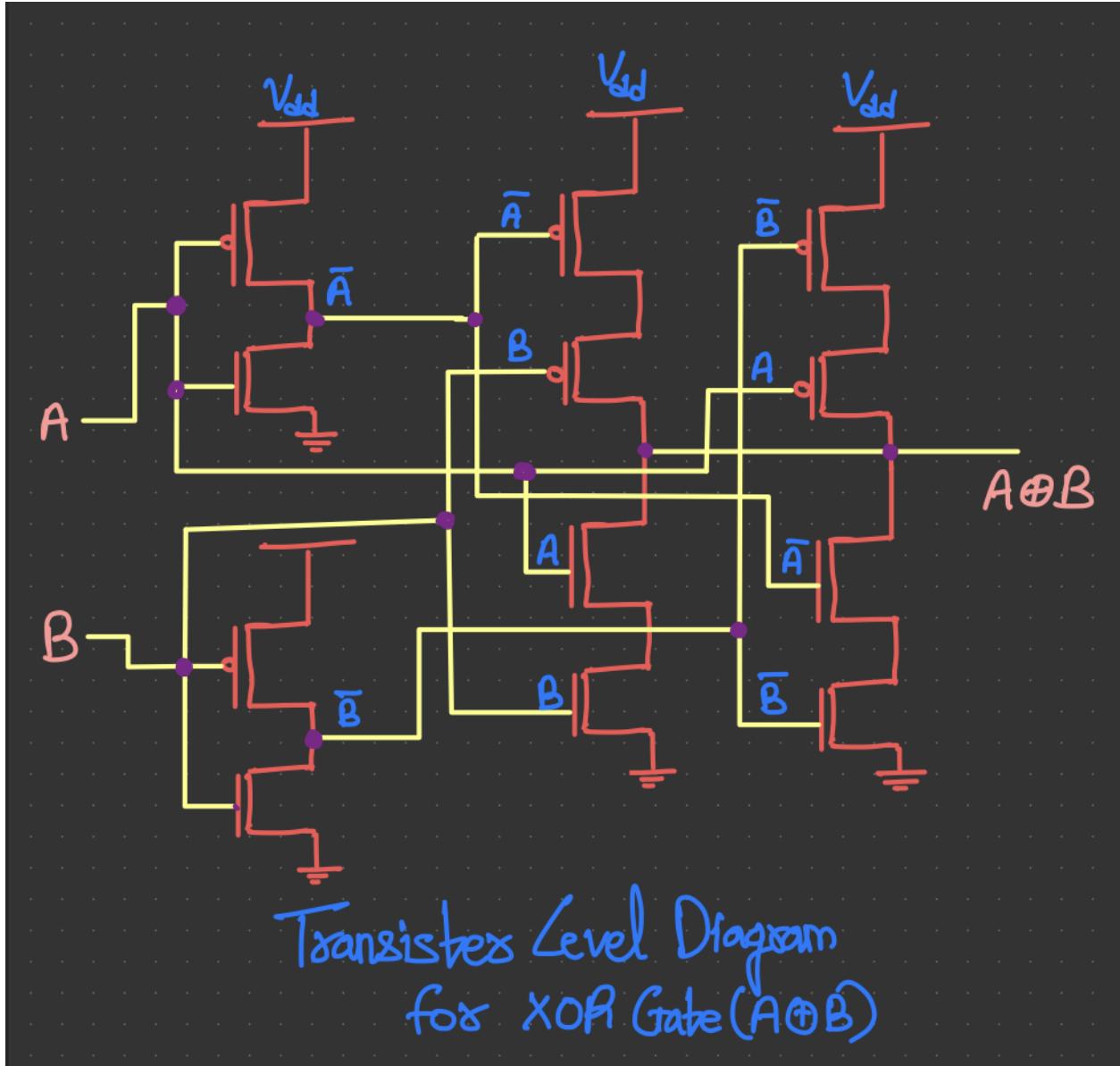
Gate Level Schematic

NOR Gate Transistor Level:



Transistor Level Diagram of NOR Gate

XOR Gate Transistor Level:



Question 1:

The adder used to compute sum is a Carry Look Ahead adder. This consists of 3 major components namely:

- Propagate & Generate for i^{th} bit
 - Propagate $p_i = a_i \text{ XOR } b_i, i = 1,2,3,4$
 - Generate $g_i = a_i \text{ AND } b_i, i = 1,2,3,4$

The above values are first generated using the XOR and 2-bit AND blocks.

- Carry Out of i^{th} bit position:

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

Using the Propagate and Generate bits generated, we precalculate the carry-out bits for each $i = 1,2,3,4$ as follows:

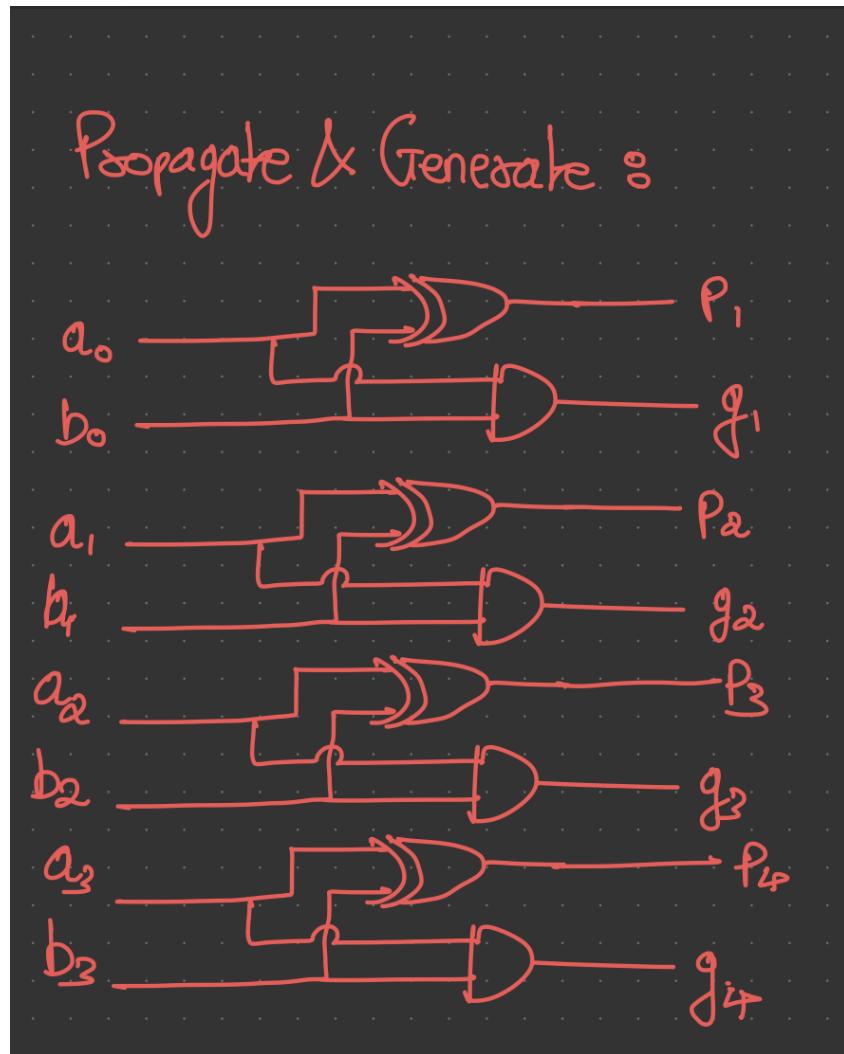
- 1st bit: $c_1 = g_1$
- 2nd bit: $c_2 = g_2 + p_2 g_1$
- 3rd bit: $c_3 = g_3 + p_3 g_2 + p_3 p_2 g_1$
- 4th bit: $c_4 = g_4 + p_4 g_3 + p_4 p_3 g_2 + p_4 p_3 p_2 g_1$

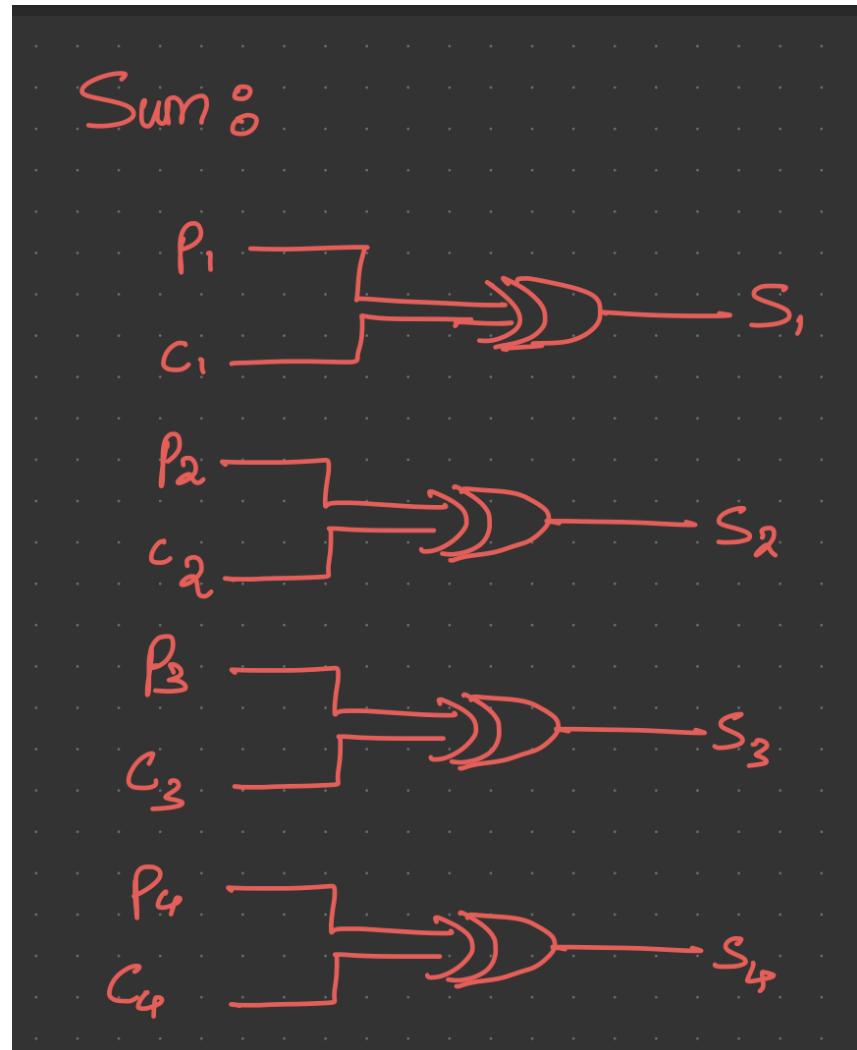
NOTE: The above mentioned carry values are taken such that c-

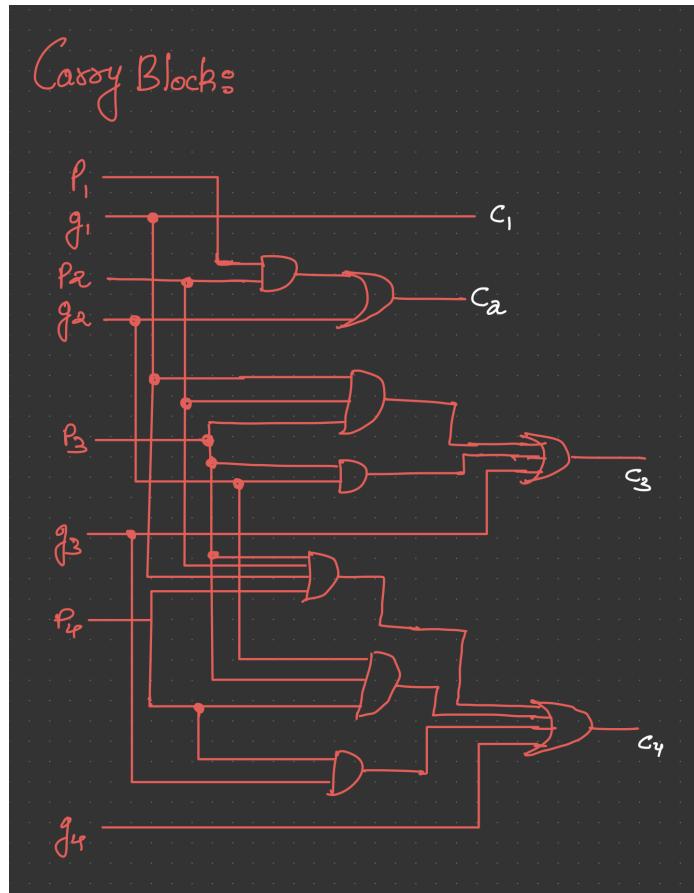
- Sum of i^{th} bit:

$$s_i = p_i \text{ XOR } c_i, i = 1,2,3,4$$

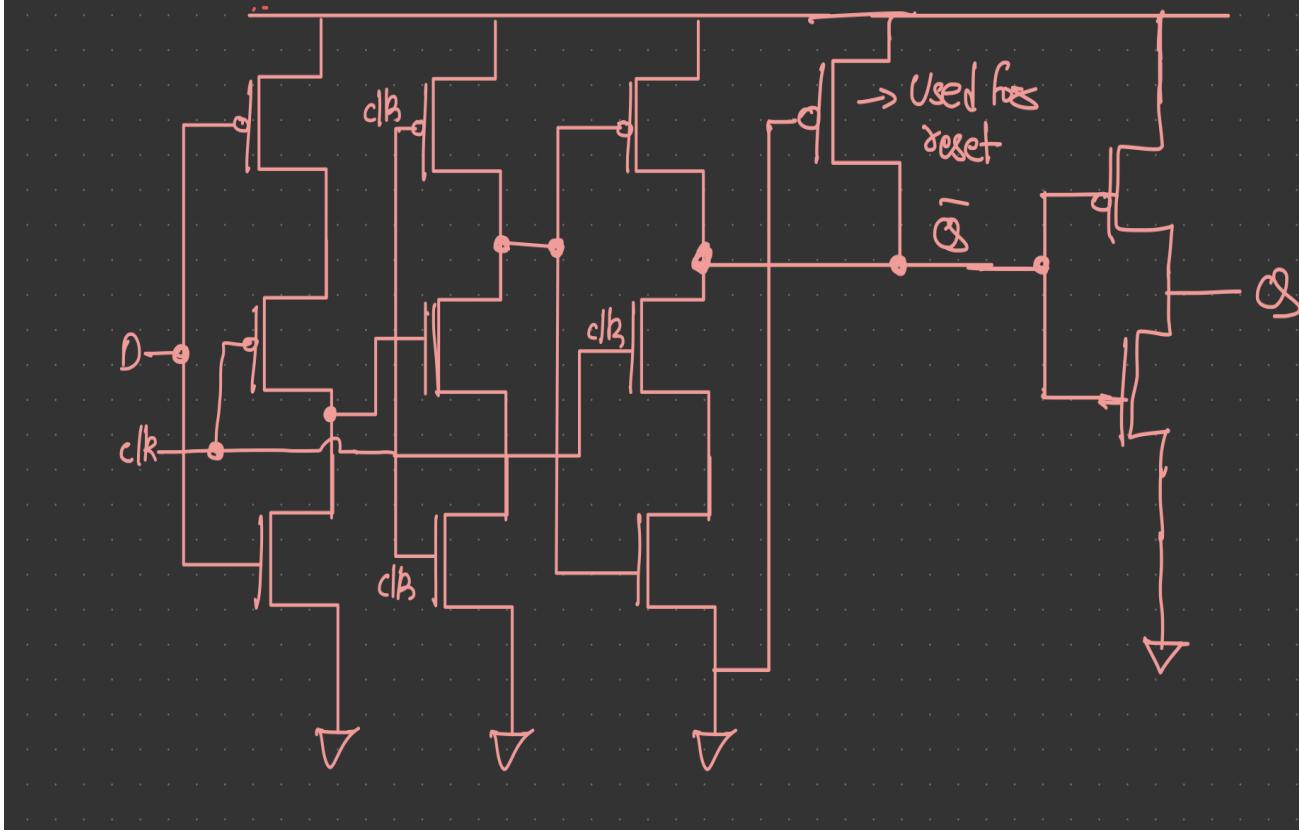
Unlike the normal adders where we send the carry of a bit to the next bit, we pre-compute the carry values for all 4 including the final carry out. So, these values are directly sent to the SUM calculators instead of having to wait for the previous bit's sum calculation. All these happen simultaneously which make the entire CLA-Adder an efficient system.





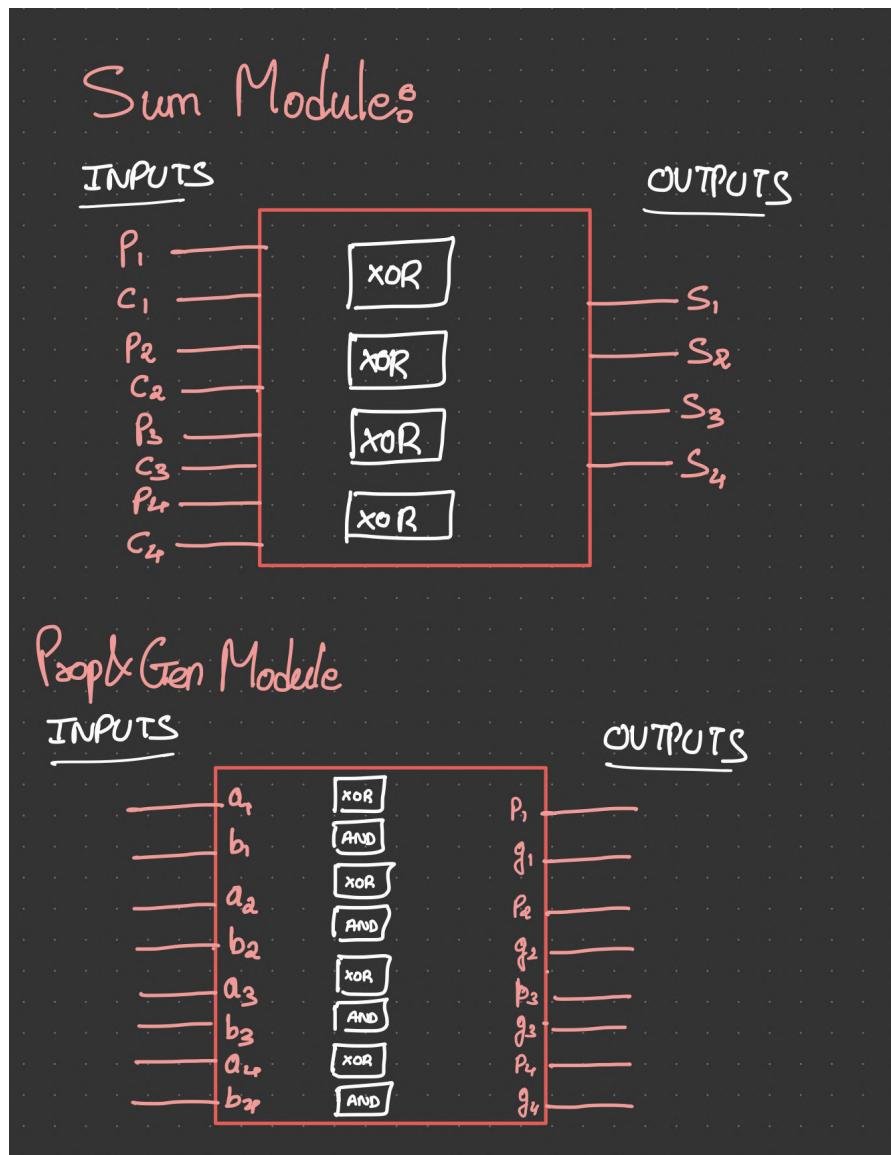


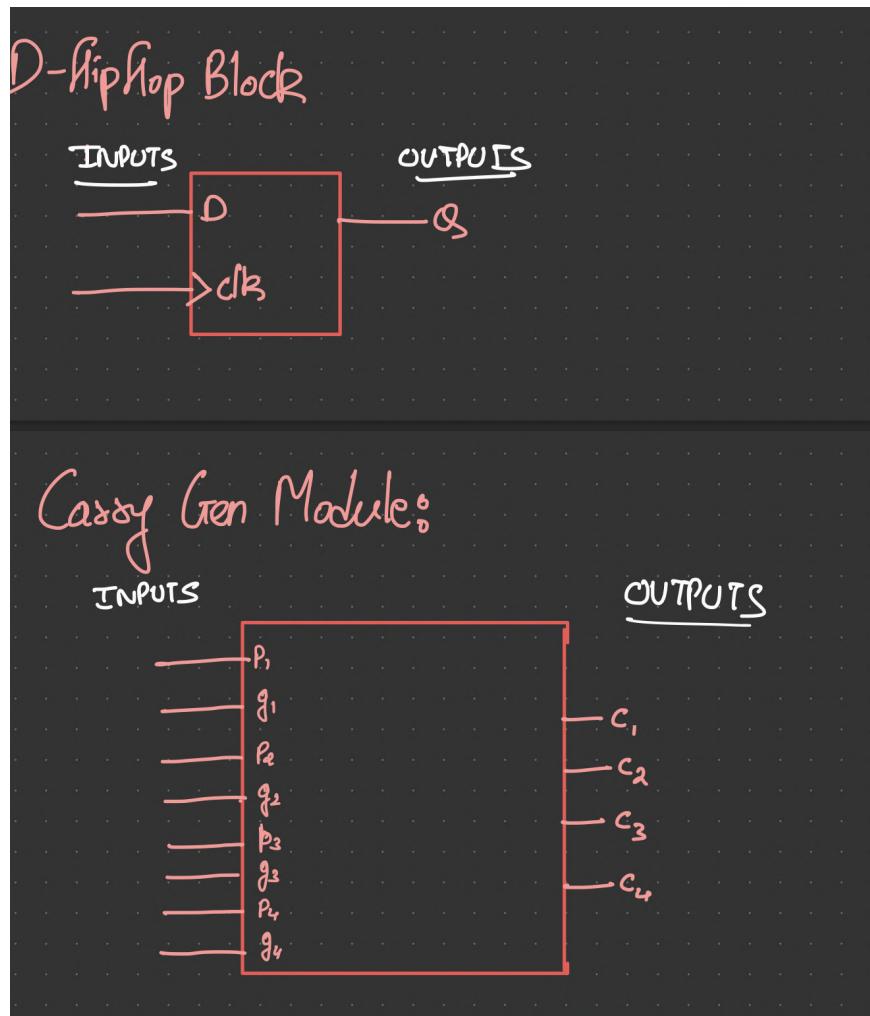
D-flip flop:



Question 2

Modules and their topology:





SIZES:

Take $\lambda = 0.09\mu$ and $W = 10^* \lambda$. The ratios are chosen such that the output capacitance sees the 2:1 ratio for optimal rise and fall delay purposes.

Module	PMOS SIZE	NMOS SIZE
D Flip Flop	4W	W
XOR	4W	2W
2-Input AND	2W	2W
2-Input OR	4W	W

3-Input AND	2W	3W
3-Input OR	6W	W
4-Input AND	2W	4W
4-Input OR	8W	W

NOTE: For OR, AND, these values belong to the NAND,NOR parts combined with an inverter.

Question 3

D-Flip Flop:

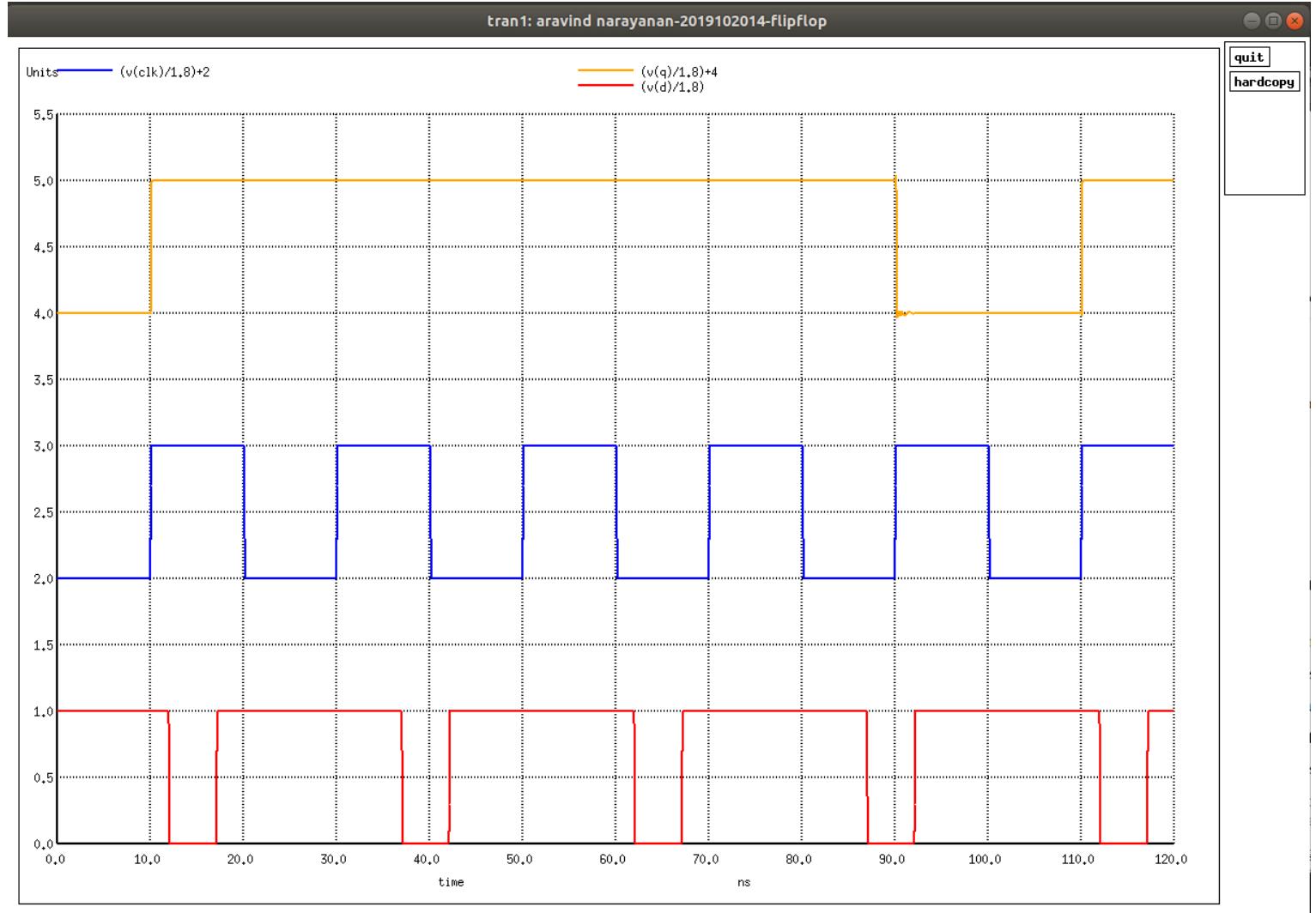
Netlist:

```

VLSI-Project > SPICE SIMS > FlipFlop.cir
1 .include TSMC_180nm.txt
2 .param SUPPLY=1.8
3 .param LAMBDA=0.09u
4 .param width_N={20*LAMBDA}
5 .param width_P={40*LAMBDA}
6 .global gnd vdd
7
8 Vdd vdd gnd 'SUPPLY'
9 vd d 0 pulse 1.8 0 12ns 100ps 100ps 5ns 25ns
10 vclk clk 0 pulse 0 1.8 10ns 100ps 100ps 10ns 20ns
11
12 .subckt FlipFlop d clk q vdd gnd
13 // Layer-1
14 M1 J1 d vdd vdd CMOSP W={width_P} L={2*LAMBDA}
15 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
16
17 M2 J2 clk J1 J1 CMOSP W={width_P} L={2*LAMBDA}
18 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
19
20 M3 J2 d gnd gnd CMOSN W={width_N} L={2*LAMBDA}
21 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
22 // Layer-2
23 M4 J3 clk vdd vdd CMOSP W={width_P} L={2*LAMBDA}
24 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
25
26 M5 J3 J2 J4 J4 CMOSN W={width_N} L={2*LAMBDA}
27 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
28
29 M6 J4 clk gnd gnd CMOSN W={width_N} L={2*LAMBDA}
30 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
31 // Layer-3
32 M7 q_bar J3 vdd vdd CMOSP W={width_P} L={2*LAMBDA}
33 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
34
35 M8 q_bar clk J5 J5 CMOSN W={width_N} L={2*LAMBDA}
36 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
37
38 M9 J5 J3 gnd gnd CMOSN W={width_N} L={2*LAMBDA}
39 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
40 // Layer-4
41 M10 q_bar vdd vdd vdd CMOSP W={width_P} L={2*LAMBDA}
42 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
43 // Layer-5
44 M11 q q_bar vdd vdd CMOSP W={width_P} L={2*LAMBDA}
45 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
46 M12 q q_bar gnd gnd CMOSN W={width_N} L={2*LAMBDA}
47 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
48 .ends FlipFlop
49
50
51 xl d clk q vdd gnd FlipFlop
52 .tran 1n 120n
53 .control
54 set hcopypscolor = 1
55 set color0=white
56 set color1=black
57 run
58 set curplottitle= "Aravind Narayanan-2019102014-FlipFlop"
59 plot (v(d)/1.8) (v(clk)/1.8)+2 (v(q)/1.8)+4
60 .endc

```

Simulation Output:



Explanation:

- RED: Input(d)
- BLUE: Input(clk)
- YELLOW: Output(q)

1. Initially till the first positive edge of clk arrives, the output is LOW regardless of the input 'd'.
2. At the first positive edge of clk, the input 'd' is HIGH, so the output becomes HIGH.
3. For the next 3 positive edges of clk, the input 'd' is always HIGH, so the Output

does not change and stays HIGH.

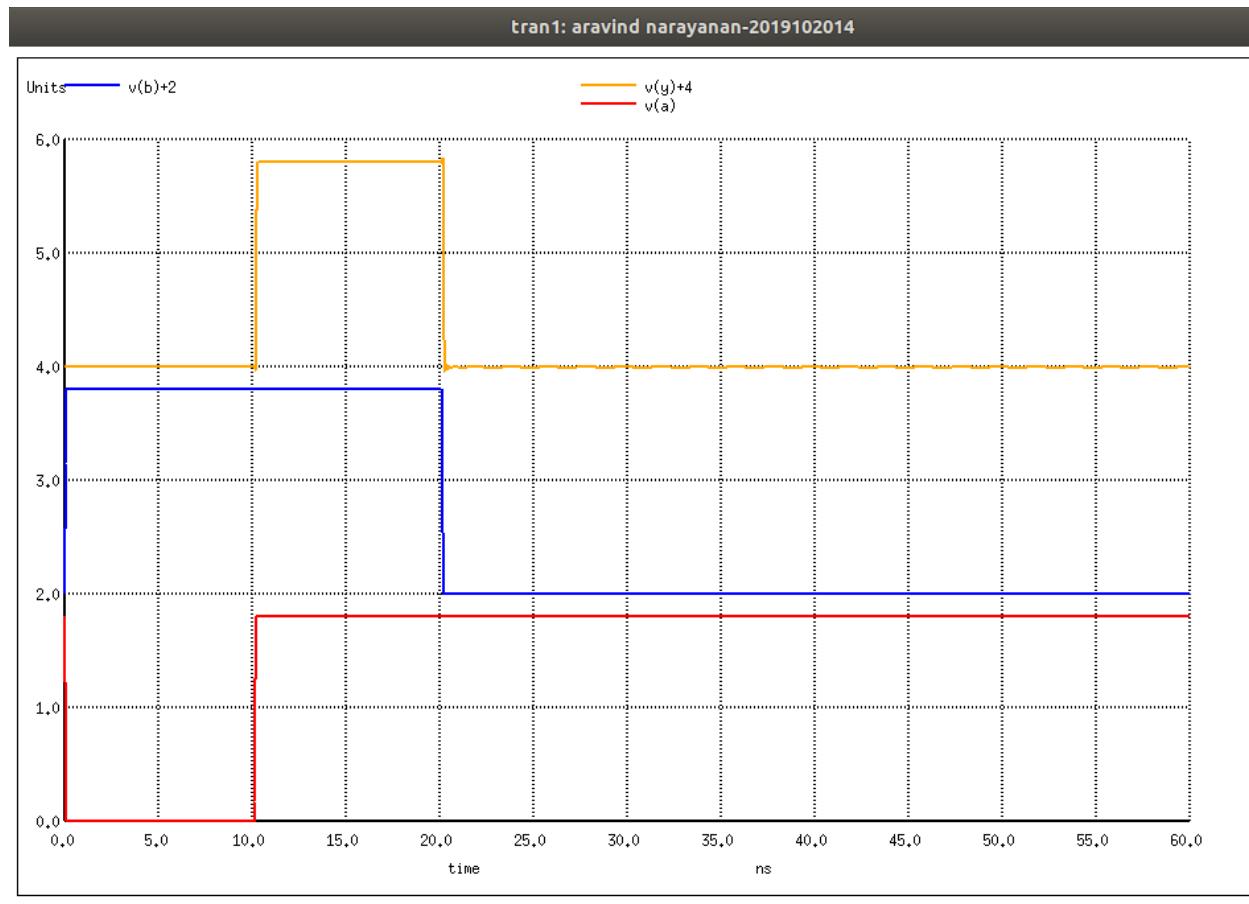
4. At the 4th positive edge of clk, the input 'd' is LOW, so the output changes to LOW.

2-Input AND Gate:

Netlist:

```
VLSI-Project > SPICE SIMS > 2bit_AND_subckt.cir
1 .include TSMC_180nm.txt
2 .param SUPPLY=1.8
3 .param LAMBDA=0.09u
4 .param width_N={20*LAMBDA}
5 .param width_P={40*LAMBDA}
6 .global gnd vdd
7
8 Vdd vdd gnd 'SUPPLY'
9 vd1 a 0 pulse 1.8 0 0ns 100ps 100ps 10ns 60ns
10 vd2 b 0 pulse 0 1.8 0ns 100ps 100ps 20ns 60ns
11
12 .subckt and_subckt a b y vdd gnd
13 * Layer-1
14 M1 nand b vdd vdd CMOSP W={width_P} L={2*LAMBDA}
15 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
16
17 M2 nand a vdd vdd CMOSP W={width_P} L={2*LAMBDA}
18 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
19
20 M3 nand a J J CMOSN W={width_N} L={2*LAMBDA}
21 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
22
23 M4 J b gnd gnd CMOSN W={width_N} L={2*LAMBDA}
24 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
25
26 M5 y nand vdd vdd CMOSP W={width_P} L={2*LAMBDA}
27 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
28
29 M6 y nand gnd gnd CMOSN W={width_N} L={2*LAMBDA}
30 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
31 .ends and_subckt
32
33 x1 a b y vdd gnd and_subckt
34
35 .tran 1n 60n
36 .control
37 set hcopypscolor = 1
38 set color0=white
39 set color1=black
40 run
41 set curplottitle= "Aravind Narayanan-2019102014"
42 plot v(a) v(b)+2 v(y)+4
43 .endc
```

Simulation Output:



Explanation:

The simulation follows the expected gate outputs when compared to theoretical outputs.

2-Input OR Gate:

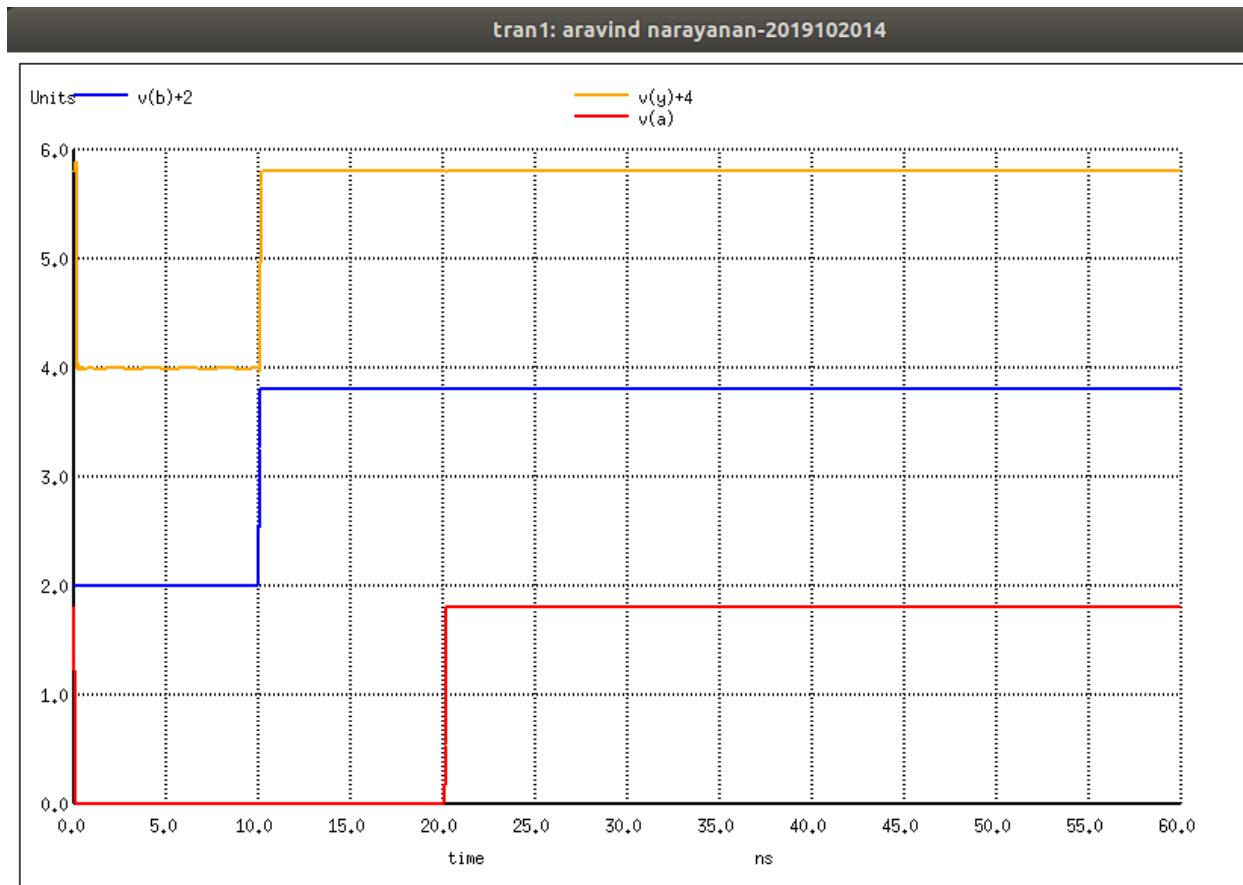
Netlist:

```

VLSI-Project > SPICE SIMS > 2bit_OR_subckt.cir
1 .include TSMC_180nm.txt
2 .param SUPPLY=1.8
3 .param LAMBDA=0.09u
4 .param width_N={20*LAMBDA}
5 .param width_P={40*LAMBDA}
6 .global gnd vdd
7
8 Vdd vdd gnd 'SUPPLY'
9 vd1 a 0 pulse 1.8 0 0ns 100ps 100ps 20ns 60ns
10 vd2 b 0 pulse 0 1.8 10ns 100ps 100ps 50ns 60ns
11
12 .subckt or_subckt a b y vdd gnd
13 M1 J1 a vdd vdd CMOSP W={width_P} L={2*LAMBDA}
14 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
15
16 M2 nor b J1 J1 CMOSP W={width_P} L={2*LAMBDA}
17 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
18
19 M3 nor a gnd gnd CMOSN W={width_N} L={2*LAMBDA}
20 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
21
22 M4 nor b gnd gnd CMOSN W={width_N} L={2*LAMBDA}
23 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
24
25 M5 y nor vdd vdd CMOSP W={width_P} L={2*LAMBDA}
26 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
27
28 M6 y nor gnd gnd CMOSN W={width_N} L={2*LAMBDA}
29 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
30 .ends or_subckt
31
32 xl a b y vdd gnd or_subckt
33
34 .tran 1n 60n
35 .control
36 set hcopypscolor = 1
37 set color0=white
38 set color1=black
39
40 set curplottitle= "Aravind Narayanan-2019102014"
41 plot v(a) v(b)+2 v(y)+4
42 .endc

```

Simulation Output:



Explanation:

The simulation follows the expected gate outputs when compared to theoretical outputs.

3-Input AND Gate:

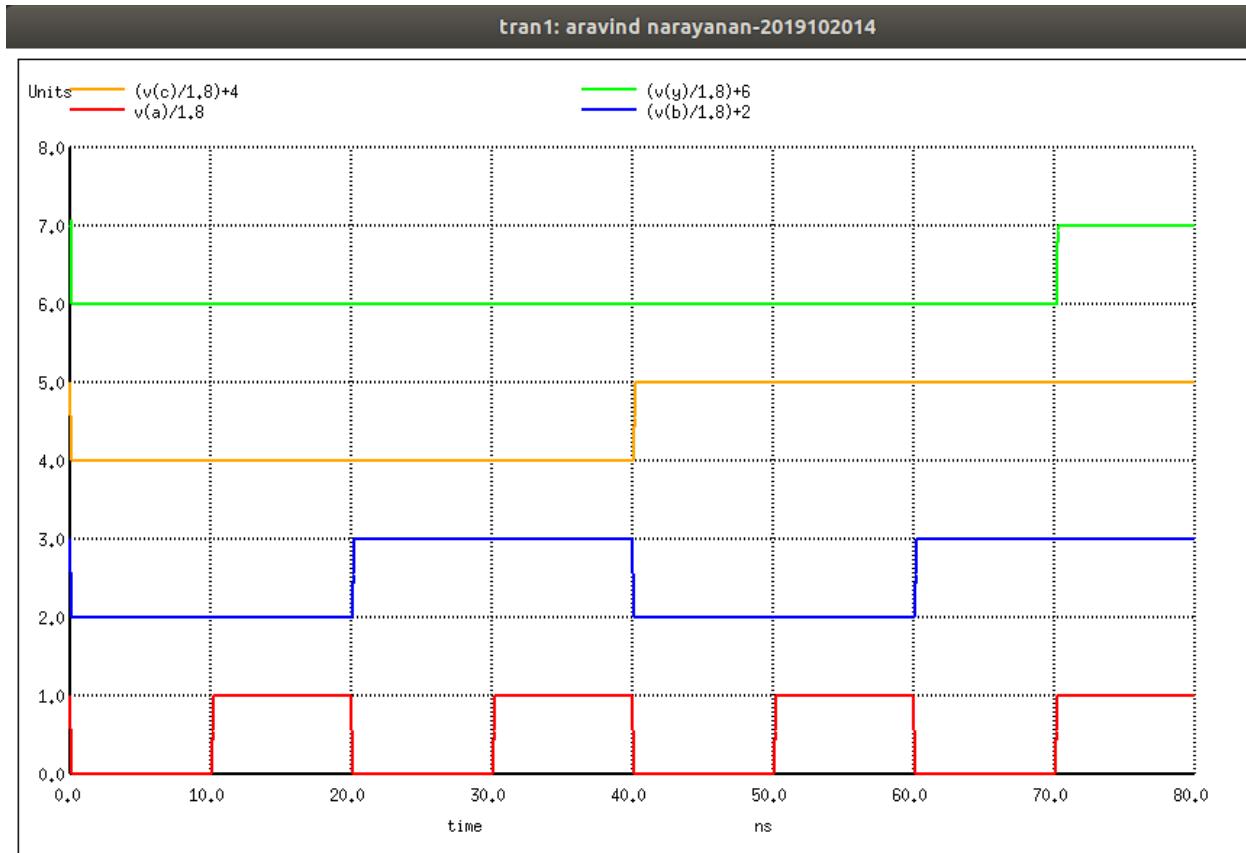
Netlist:

```

VLSI-Project > SPICE SIMS > 3bit_AND_subckt.cir
1 .include TSMC_180nm.txt
2 .param SUPPLY=1.8
3 .param LAMBDA=0.09u
4 .param width_N={10*LAMBDA}
5 .param width_P={20*LAMBDA}
6 .global gnd vdd
7
8 Vdd vdd gnd 'SUPPLY'
9 vd1 a 0 pulse 1.8 0 0ns 100ps 100ps 10ns 20ns
10 vd2 b 0 pulse 1.8 0 0ns 100ps 100ps 20ns 40ns
11 vd3 c 0 pulse 1.8 0 0ns 100ps 100ps 40ns 80ns
12
13 .subckt and3_subckt a b c y vdd gnd
//NAND
15 M1 nand a vdd vdd CMOSP W={width_P} L={2*LAMBDA}
+ AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
17
18 M2 nand b vdd vdd CMOSP W={width_P} L={2*LAMBDA}
+ AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
20
21 M3 nand c vdd vdd CMOSP W={width_P} L={2*LAMBDA}
+ AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
23
24 M4 nand a J1 J1 CMOSN W={width_N} L={2*LAMBDA}
+ AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
26
27 M5 J1 b J2 J2 CMOSN W={width_N} L={2*LAMBDA}
+ AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
29
30 M6 J2 c gnd gnd CMOSN W={width_N} L={2*LAMBDA}
+ AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
32 //Inverter
33 M7 y nand vdd vdd CMOSP W={width_P} L={2*LAMBDA}
+ AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
35
36 M8 y nand gnd gnd CMOSN W={width_N} L={2*LAMBDA}
+ AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
38 .ends and3_subckt
39
40 xl a b c y vdd gnd and3_subckt
41
42 .tran 0.01n 80n
43 .control
44 set hcopypscolor = 1
45 set color0=white
46 set color1=black
47 run
48 | set curplottitle= "Aravind Narayanan-2019102014"
49 plot v(a)/1.8 (v(b)/1.8)+2 (v(c)/1.8)+4 (v(y)/1.8)+6
50 .endc

```

Simulation Output:



Explanation:

The simulation follows the expected gate outputs when compared to theoretical outputs.

3-Input OR Gate:

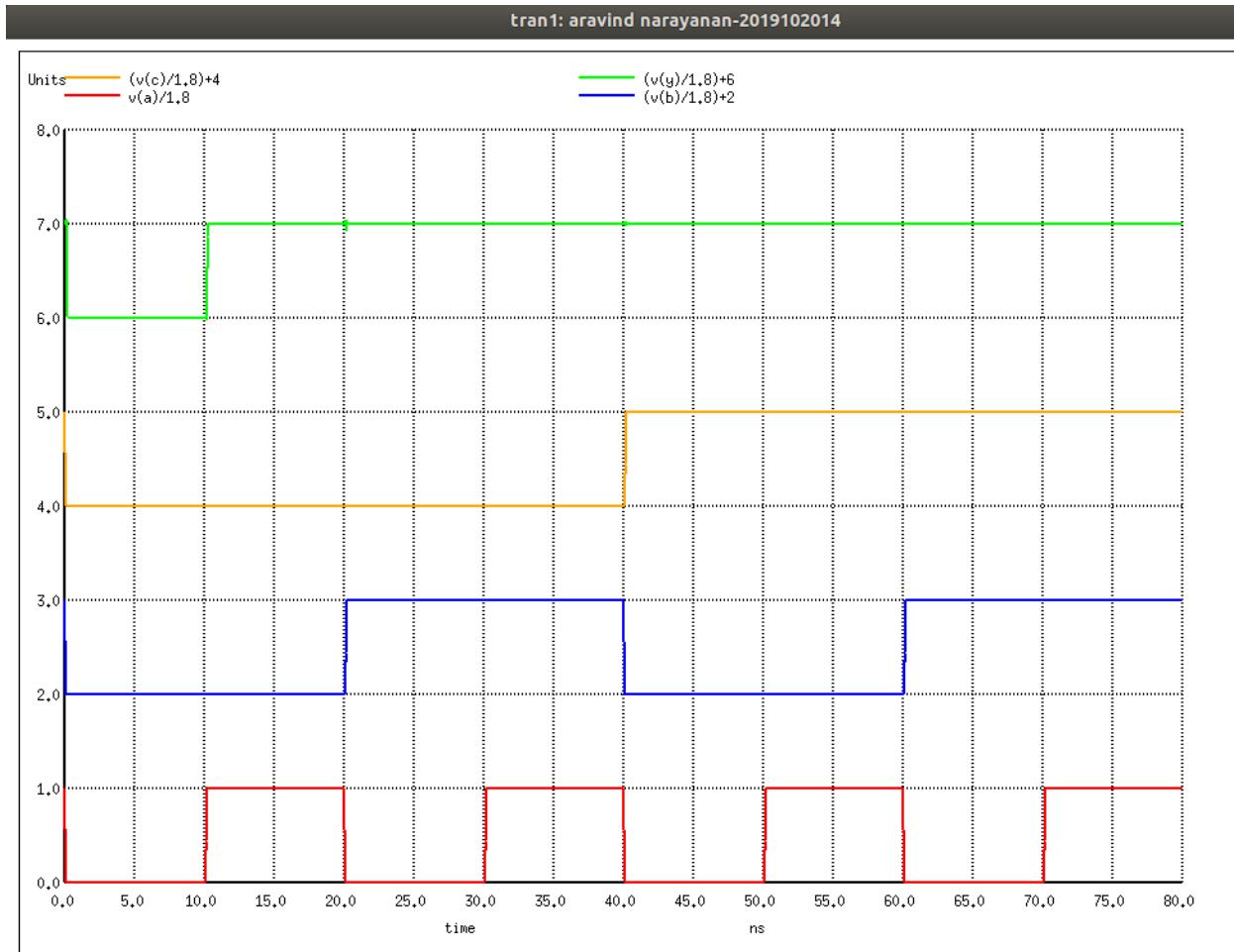
Netlist:

```

VLSI-Project > SPICE SIMS > 3bit_OR_subckt.cir
1 .include TSMC_180nm.txt
2 .param SUPPLY=1.8
3 .param LAMBDA=0.09u
4 .param width_N={20*LAMBDA}
5 .param width_P={40*LAMBDA}
6 .global gnd vdd
7
8 Vdd vdd gnd 'SUPPLY'
9 vd1 a 0 pulse 1.8 0 0ns 100ps 100ps 10ns 20ns
10 vd2 b 0 pulse 1.8 0 0ns 100ps 100ps 20ns 40ns
11 vd3 c 0 pulse 1.8 0 0ns 100ps 100ps 40ns 80ns
12
13 .subckt or3_subckt a b c y vdd gnd
14 // NOR
15 M1 J1 a vdd vdd CMOSP W={width_P} L={2*LAMBDA}
16 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
17
18 M2 J2 b J1 J1 CMOSP W={width_P} L={2*LAMBDA}
19 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
20
21 M3 nor c J2 J2 CMOSP W={width_P} L={2*LAMBDA}
22 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
23
24 M4 nor a gnd gnd CMOSN W={width_N} L={2*LAMBDA}
25 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
26
27 M5 nor b gnd gnd CMOSN W={width_N} L={2*LAMBDA}
28 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
29
30 M6 nor c gnd gnd CMOSN W={width_N} L={2*LAMBDA}
31 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
32 // Inverter
33 M7 y nor vdd vdd CMOSP W={width_P} L={2*LAMBDA}
34 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
35
36 M8 y nor gnd gnd CMOSN W={width_N} L={2*LAMBDA}
37 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
38 .ends or3_subckt
39
40 xl a b c y vdd gnd or3_subckt
41
42 .tran 0.01n 80n
43 .control
44 set hcopypscolor = 1
45 set color0=white
46 set color1=black
47
48 run
49 | set curplottitle= "Aravind Narayanan-2019102014"
50 | plot v(a)/1.8 (v(b)/1.8)+2 (v(c)/1.8)+4 (v(y)/1.8)+6
51 .endc

```

Simulation Output:



Explanation:

The simulation follows the expected gate outputs when compared to theoretical outputs.

4-Input AND Gate:

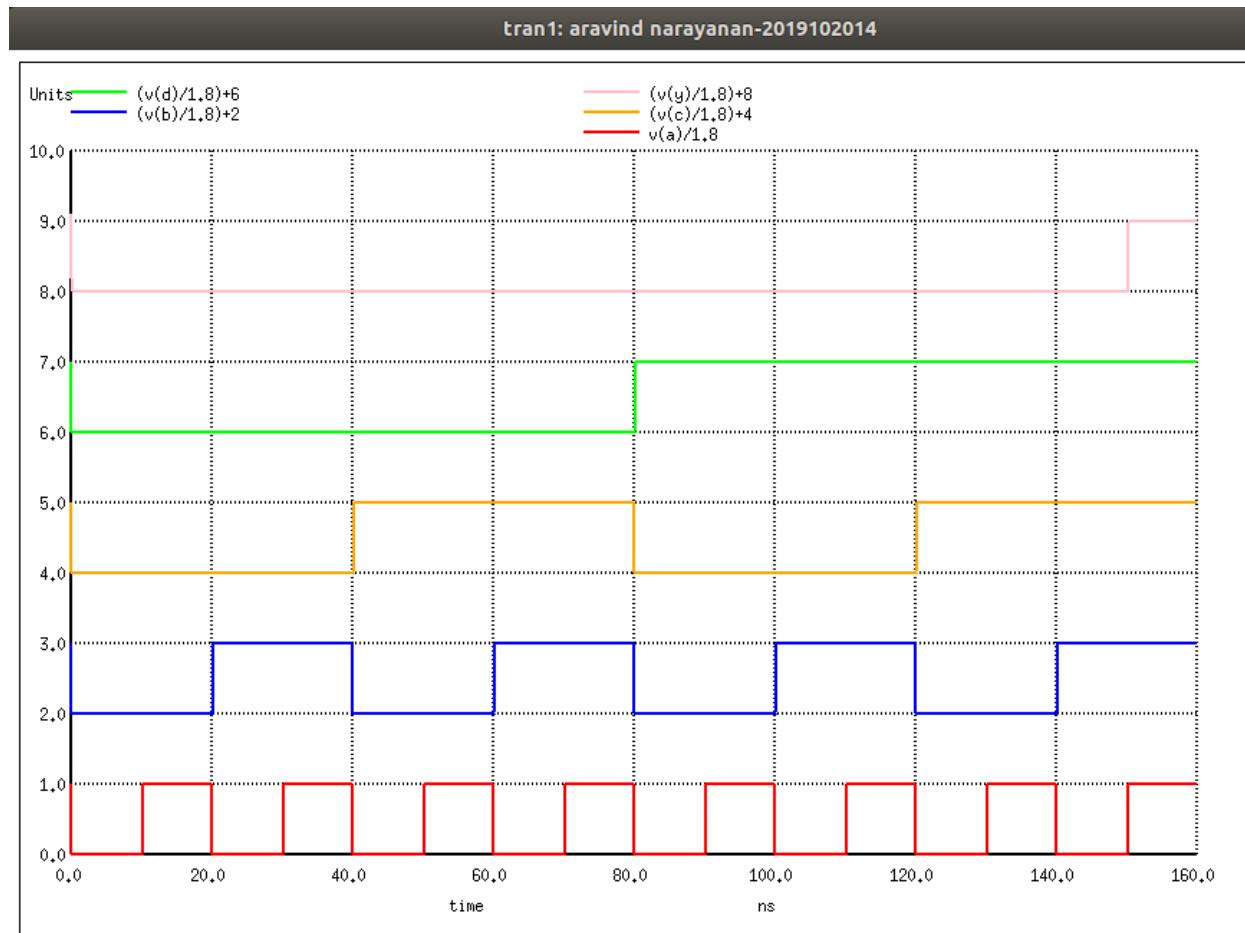
Netlist:

```

VLSI-Project > SPICE SIMS > 4bit_AND_subckt.cir
1 .include TSMC_180nm.txt
2 .param SUPPLY=1.8
3 .param LAMBDA=0.09u
4 .param width_N={10*LAMBDA}
5 .param width_P={20*LAMBDA}
6 .global gnd vdd
7
8 Vdd vdd gnd 'SUPPLY'
9 vd1 a 0 pulse 1.8 0 0ns 100ps 100ps 10ns 20ns
10 vd2 b 0 pulse 1.8 0 0ns 100ps 100ps 20ns 40ns
11 vd3 c 0 pulse 1.8 0 0ns 100ps 100ps 40ns 80ns
12 vd4 d 0 pulse 1.8 0 0ns 100ps 100ps 80ns 160ns
13
14 .subckt and4_subckt a b c d y vdd gnd
15 //NAND
16 M1 nand a vdd vdd CMOSP W={width_P} L={2*LAMBDA}
17 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
18
19 M2 nand b vdd vdd CMOSP W={width_P} L={2*LAMBDA}
20 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
21
22 M3 nand c vdd vdd CMOSP W={width_P} L={2*LAMBDA}
23 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
24
25 M4 nand d vdd vdd CMOSP W={width_P} L={2*LAMBDA}
26 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
27
28 M5 nand a J1 J1 CMOSN W={width_N} L={2*LAMBDA}
29 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
30
31 M6 J1 b J2 J2 CMOSN W={width_N} L={2*LAMBDA}
32 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
33
34 M7 J2 c J3 J3 CMOSN W={width_N} L={2*LAMBDA}
35 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
36
37 M8 J3 d gnd gnd CMOSN W={width_N} L={2*LAMBDA}
38 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
39 //Inverter
40 M9 y nand vdd vdd CMOSP W={width_P} L={2*LAMBDA}
41 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
42
43 M10 y nand gnd gnd CMOSN W={width_N} L={2*LAMBDA}
44 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
45 .ends and4_subckt
46
47 xl a b c d y vdd gnd and4_subckt
48
49 .tran 0.01n 160n
50 .control
51 set hcopypscolor = 1
52 set color0=white
53 set color1=black
54
55 run
56 set curplottitle= "Aravind Narayanan-2019102014"
57 plot v(a)/1.8 (v(b)/1.8)+2 (v(c)/1.8)+4 (v(d)/1.8)+6 (v(y)/1.8)+8
58 .endc

```

Simulation Output:



Explanation:

The simulation follows the expected gate outputs when compared to theoretical outputs.

4-Input OR Gate:

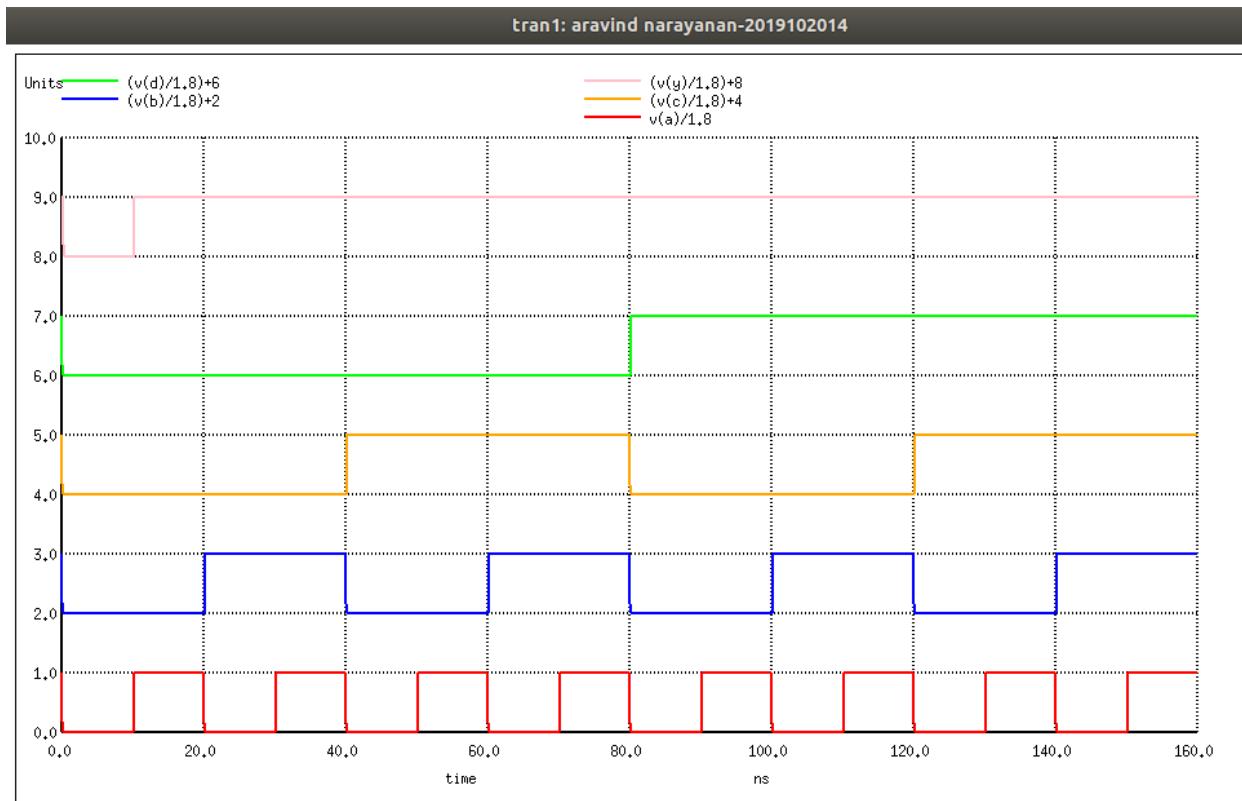
Netlist:

```

VLSI-Project > SPICE SIMS > 4bit_OR_subckt.cir
1 .include TSMC_180nm.txt
2 .param SUPPLY=1.8
3 .param LAMBDA=0.09u
4 .param width_N={20*LAMBDA}
5 .param width_P={40*LAMBDA}
6 .global gnd vdd
7
8 Vdd vdd gnd 'SUPPLY'
9 vd1 a 0 pulse 1.8 0 0ns 100ps 100ps 10ns 20ns
10 vd2 b 0 pulse 1.8 0 0ns 100ps 100ps 20ns 40ns
11 vd3 c 0 pulse 1.8 0 0ns 100ps 100ps 40ns 80ns
12 vd4 d 0 pulse 1.8 0 0ns 100ps 100ps 80ns 160ns
13
14 .subckt or4_subckt a b c d y vdd gnd
15 // NOR
16 M1      J1      a      vdd      vdd CMOSP    W={width_P}    L={2*LAMBDA}
17 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
18
19 M2      J2      b      J1      J1 CMOSP    W={width_P}    L={2*LAMBDA}
20 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
21
22 M3      J3      c      J2      J2 CMOSP    W={width_P}    L={2*LAMBDA}
23 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
24
25 M4      nor      d      J3      J3 CMOSP    W={width_P}    L={2*LAMBDA}
26 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
27
28 M5      nor      a      gnd     gnd CMOSN    W={width_N}    L={2*LAMBDA}
29 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
30
31 M6      nor      b      gnd     gnd CMOSN    W={width_N}    L={2*LAMBDA}
32 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
33
34 M7      nor      c      gnd     gnd CMOSN    W={width_N}    L={2*LAMBDA}
35 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
36
37 M8      nor      d      gnd     gnd CMOSN    W={width_N}    L={2*LAMBDA}
38 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
39 // Inverter
40 M9      y      nor      vdd     vdd CMOSP    W={width_P}    L={2*LAMBDA}
41 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
42
43 M10     y      nor      gnd     gnd CMOSN    W={width_N}    L={2*LAMBDA}
44 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
45 .ends or4_subckt
46
47 x1 a b c d y vdd gnd or4_subckt
48
49 .tran 0.01n 160n
50 .control
51 set hcopypscolor = 1
52 set color0=white
53 set color1=black
54
55 run
56 set curplottitle= "Aravind Narayanan-2019102014"
57 plot v(a)/1.8 (v(b)/1.8)+2 (v(c)/1.8)+4 (v(d)/1.8)+6 (v(y)/1.8)+8
58 .endc

```

Simulation Output:



Explanation:

The simulation follows the expected gate outputs when compared to theoretical outputs.

2-Input XOR Gate:

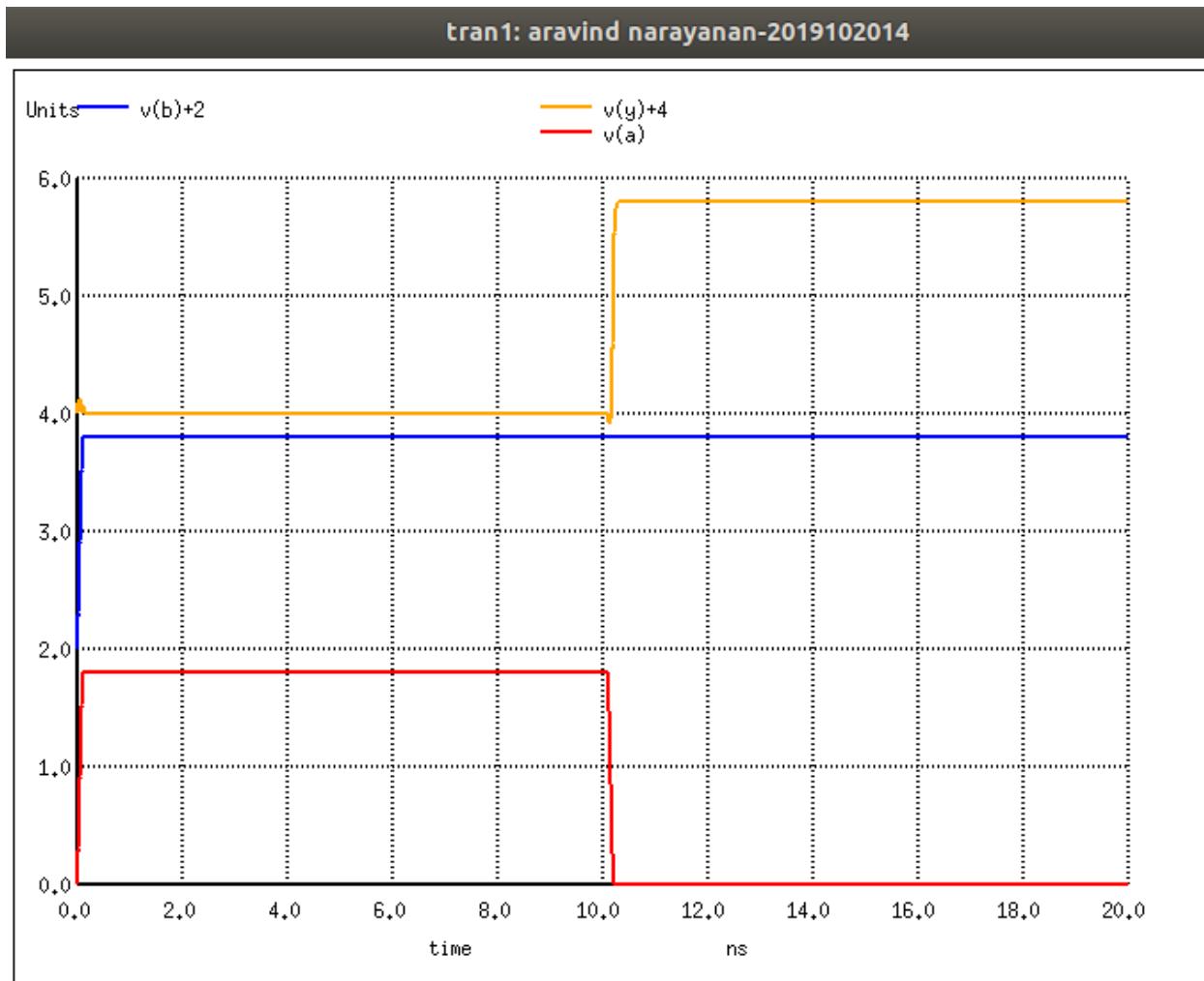
Netlist:

```

VLSI-Project > SPICE SIMS > XOR_subckt.cir
1 .include TSMC_180nm.txt
2 .param SUPPLY=1.8
3 .param LAMBDA=0.09u
4 .param width_N={20*LAMBDA}
5 .param width_P={40*LAMBDA}
6 .global gnd vdd
7
8 Vdd vdd gnd 'SUPPLY'
9 vd1 a 0 pulse 0 1.8 0ns 100ps 100ps 10ns 20ns
10 vd2 b 0 pulse 0 1.8 0ns 100ps 100ps 20ns 40ns
11
12 .subckt xor_subckt a b y vdd gnd
13 //Top inverter
14 M1 a_bar a vdd vdd CMOSP W={width_P} L={2*LAMBDA}
15 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
16
17 M2 a_bar a gnd gnd CMOSN W={width_N} L={2*LAMBDA}
18 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
19 //Bottom Inverter
20 M3 b_bar b vdd vdd CMOSP W={width_P} L={2*LAMBDA}
21 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
22
23 M4 b_bar b gnd gnd CMOSN W={width_N} L={2*LAMBDA}
24 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
25 //Layer 2
26 M5 J1 a_bar vdd vdd CMOSP W={width_P} L={2*LAMBDA}
27 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
28
29 M6 y b J1 J1 CMOSP W={width_P} L={2*LAMBDA}
30 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
31
32 M7 y a J2 J2 CMOSN W={width_N} L={2*LAMBDA}
33 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
34
35 M8 J2 b gnd gnd CMOSN W={width_N} L={2*LAMBDA}
36 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
37 //Layer 3
38 M9 J11 b_bar vdd vdd CMOSP W={width_P} L={2*LAMBDA}
39 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
40
41 M10 y a J11 J11 CMOSP W={width_P} L={2*LAMBDA}
42 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
43
44 M11 y a_bar J22 J22 CMOSN W={width_N} L={2*LAMBDA}
45 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
46
47 M12 J22 b_bar gnd gnd CMOSN W={width_N} L={2*LAMBDA}
48 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
49 .ends xor_subckt
50
51 xl a b y vdd gnd xor_subckt
52
53 .tran 0.001n 20n
54 .control
55 set hcopypscolor = 1
56 set color0=white
57 set color1=black
58 run
59 set curplottitle= "Aravind Narayanan-2019102014"
60 plot v(a) v(b)+2 v(y)+4
61 .endc

```

Simulation Output:



Explanation:

CLA Block:

(I) Propagate:

Netlist:

```

VLSI-Project > SPICE SIMS > PropGen.cir
1 .include TSMC_180nm.txt
2 .param SUPPLY=1.8
3 .param LAMBDA=0.09u
4 .param width_N={20*LAMBDA}
5 .param width_P={40*LAMBDA}
6 .global gnd vdd
7
8 Vdd vdd gnd 'SUPPLY'
9 vd4a a4 0 pulse 1.8 0 0ns 100ps 100ps 60ns 320ns
10 vd4b b4 0 pulse 1.8 0 0ns 100ps 100ps 160ns 320ns
11
12
13 .subckt xor_subckt a b y vdd gnd
//Top inverter
14
15 M1      a_bar      a      vdd      vdd  CMOSP  W={width_P}  L={2*LAMBDA}
16 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
17 M2      a_bar      a      gnd      gnd  CMOSN  W={width_N}  L={2*LAMBDA}
18 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
19
20 //Bottom Inverter
21 M3      b_bar      b      vdd      vdd  CMOSP  W={width_P}  L={2*LAMBDA}
22 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
23 M4      b_bar      b      gnd      gnd  CMOSN  W={width_N}  L={2*LAMBDA}
24 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
25
26 //Layer 2
27 M5      J1      a_bar      vdd      vdd  CMOSP  W={width_P}  L={2*LAMBDA}
28 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
29 M6      y      b      J1      J1  CMOSP  W={width_P}  L={2*LAMBDA}
30 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
31 M7      y      a      J2      J2  CMOSN  W={width_N}  L={2*LAMBDA}
32 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
33 M8      J2      b      gnd      gnd  CMOSN  W={width_N}  L={2*LAMBDA}
34 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
35
36 //Layer 3
37 M9      J11      b_bar      vdd      vdd  CMOSP  W={width_P}  L={2*LAMBDA}
38 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
39 M10     y      a      J11      J11  CMOSP  W={width_P}  L={2*LAMBDA}
40 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
41 M11     y      a_bar      J22      J22  CMOSN  W={width_N}  L={2*LAMBDA}
42 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
43 M12     J22      b_bar      gnd      gnd  CMOSN  W={width_N}  L={2*LAMBDA}
44 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
45 .ends xor_subckt
46
47
48 .subckt and_subckt a b y vdd gnd
* Layer 1
49
50 M1      nand      b      vdd      vdd  CMOSP  W={width_P}  L={2*LAMBDA}
51 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
52 M2      nand      a      vdd      vdd  CMOSP  W={width_P}  L={2*LAMBDA}
53 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
54 M3      nand      a      J      J  CMOSN  W={width_N}  L={2*LAMBDA}
55 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
56 M4      J      b      gnd      gnd  CMOSN  W={width_N}  L={2*LAMBDA}
57 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
58 M5      y      nand      vdd      vdd  CMOSP  W={width_P}  L={2*LAMBDA}
59 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
60 M6      y      nand      gnd      gnd  CMOSN  W={width_N}  L={2*LAMBDA}
61 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
62 .ends and_subckt

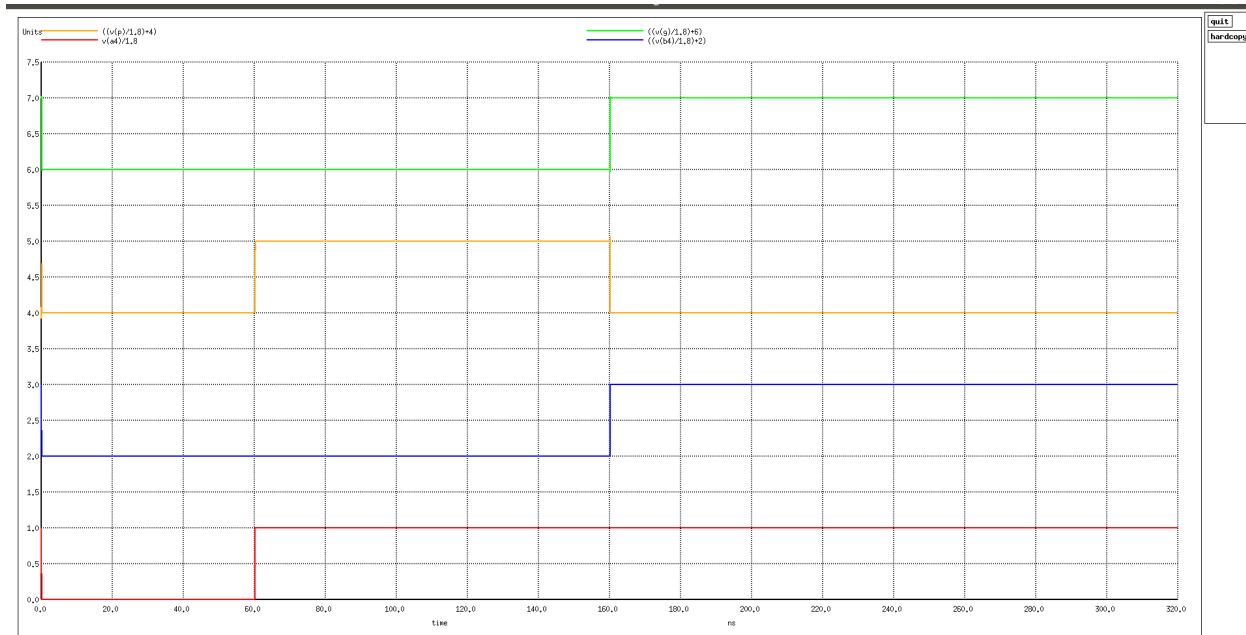
```

```

63
64 // Propagate
65 xl a4 b4 P vdd gnd xor_subckt
66
67 // Generate
68
69 x2 a4 b4 G vdd gnd and_subckt
70
71
72 .tran 1n 320n
73 .control
74 set hcopypscolor = 1
75 set color0=white
76 set color1=black
77
78 run
79 plot v(a4)/1.8 ((v(b4)/1.8)+2) ((v(P)/1.8)+4) ((v(G)/1.8)+6)
80 set curplottitle= "Aravind Narayanan-2019102014"
81
82 .endc

```

Simulation Output:



Delay:

Propagate: 2.99844e-11secs => 0.029ns

Generate: 4.37242e-11secs => 0.043ns

(III) Carry:

NETLIST:

```
VLSI-Project > SPICE SIMS > CarryGen.cir
1 .include TSMC_180nm.txt
2 .param SUPPLY=1.8
3 .param LAMBDA=0.09u
4 .param W = {10*LAMBDA}
5 .param width_N={10*LAMBDA}
6 .param width_P={20*LAMBDA}
7 .global gnd vdd
8
9 Vdd vdd gnd 'SUPPLY'
10 vd10 P1 0 pulse 1.8 0 0ns 100ps 100ps 20ns 40ns
11 vd20 G1 0 pulse 1.8 0 0ns 100ps 100ps 30ns 40ns
12 vd11 P2 0 pulse 0 1.8 10ns 100ps 100ps 10ns 40ns
13 vd21 G2 0 pulse 1.8 0 0ns 100ps 100ps 30ns 40ns
14 vd12 P3 0 pulse 1.8 0 0ns 100ps 100ps 50ns 60ns
15 vd22 G3 0 pulse 1.8 0 0ns 100ps 100ps 20ns 30ns
16 vd13 P4 0 pulse 1.8 0 0ns 100ps 100ps 20ns 30ns
17 vd23 G4 0 pulse 1.8 0 0ns 100ps 100ps 40ns 60ns
18
19 vdC Ci 0 pulse 0 0 0ns 100ps 100ps 20ns 60ns|
20
21 .subckt xor_subckt a b y vdd gnd
//Top inverter
22 M1 a_bar a vdd vdd CMOSP W={width_P} L={2*LAMBDA}
23 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
24 M2 a_bar a gnd gnd CMOSN W={width_N} L={2*LAMBDA}
25 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
26
//Bottom Inverter
27 M3 b_bar b vdd vdd CMOSP W={width_P} L={2*LAMBDA}
28 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
29 M4 b_bar b gnd gnd CMOSN W={width_N} L={2*LAMBDA}
30 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
31
//Layer 2
32 M5 J1 a_bar vdd vdd CMOSP W={width_P} L={2*LAMBDA}
33 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
34 M6 y b J1 J1 CMOSP W={width_P} L={2*LAMBDA}
35 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
36 M7 y a J2 J2 CMOSN W={width_N} L={2*LAMBDA}
37 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
38 M8 J2 b gnd gnd CMOSN W={width_N} L={2*LAMBDA}
39 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
40
//Layer 3
41 M9 J11 b_bar vdd vdd CMOSP W={width_P} L={2*LAMBDA}
42 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
43 M10 y a J11 J11 CMOSP W={width_P} L={2*LAMBDA}
44 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
45 M11 y a_bar J22 J22 CMOSN W={width_N} L={2*LAMBDA}
46 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
47 M12 J22 b_bar gnd gnd CMOSN W={width_N} L={2*LAMBDA}
48 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
49
.ends xor_subckt
50
51
52
53
54
```

```

/LSI-Project > SPICE SIMS > CarryGen.cir
56 .subckt and_subckt a b y vdd gnd
57 .param w_AND_P = {2*W}
58 .param w_AND_N = {2*W}
59 * Layer-1
60 M1 nand b vdd vdd CMOSP W={w_AND_P} L={2*LAMBDA}
61 + AS={5*w_AND_P*LAMBDA} PS={10*LAMBDA+2*w_AND_P} AD={5*w_AND_P*LAMBDA} PD={10*LAMBDA+2*w_AND_P}
62 M2 nand a vdd vdd CMOSP W={w_AND_P} L={2*LAMBDA}
63 + AS={5*w_AND_P*LAMBDA} PS={10*LAMBDA+2*w_AND_P} AD={5*w_AND_P*LAMBDA} PD={10*LAMBDA+2*w_AND_P}
64 M3 nand a J CMOSN W={w_AND_N} L={2*LAMBDA}
65 + AS={5*w_AND_N*LAMBDA} PS={10*LAMBDA+2*w_AND_N} AD={5*w_AND_N*LAMBDA} PD={10*LAMBDA+2*w_AND_N}
66 M4 J b gnd gnd CMOSN W={w_AND_N} L={2*LAMBDA}
67 + AS={5*w_AND_N*LAMBDA} PS={10*LAMBDA+2*w_AND_N} AD={5*w_AND_N*LAMBDA} PD={10*LAMBDA+2*w_AND_N}
68
69 M5 y nand vdd vdd CMOSP W={width_P} L={2*LAMBDA}
70 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
71 M6 y nand gnd gnd CMOSN W={width_N} L={2*LAMBDA}
72 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
73 .ends and_subckt
74
75 .subckt or_subckt a b y vdd gnd
76 .param w_OR_P = {4*W}
77 .param w_OR_N = {W}
78 * Layer-1
79 M1 J1 a vdd vdd CMOSP W={w_OR_P} L={2*LAMBDA}
80 + AS={5*w_OR_P*LAMBDA} PS={10*LAMBDA+2*w_OR_P} AD={5*w_OR_P*LAMBDA} PD={10*LAMBDA+2*w_OR_P}
81 M2 nor b J1 J1 CMOSP W={w_OR_P} L={2*LAMBDA}
82 + AS={5*w_OR_P*LAMBDA} PS={10*LAMBDA+2*w_OR_P} AD={5*w_OR_P*LAMBDA} PD={10*LAMBDA+2*w_OR_P}
83 M3 nor a gnd gnd CMOSN W={w_OR_N} L={2*LAMBDA}
84 + AS={5*w_OR_N*LAMBDA} PS={10*LAMBDA+2*w_OR_N} AD={5*w_OR_N*LAMBDA} PD={10*LAMBDA+2*w_OR_N}
85 M4 nor b gnd gnd CMOSN W={w_OR_N} L={2*LAMBDA}
86 + AS={5*w_OR_N*LAMBDA} PS={10*LAMBDA+2*w_OR_N} AD={5*w_OR_N*LAMBDA} PD={10*LAMBDA+2*w_OR_N}
87
88 M5 y nor vdd vdd CMOSP W={width_P} L={2*LAMBDA}
89 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
90 M6 y nor gnd gnd CMOSN W={width_N} L={2*LAMBDA}
91 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
92 .ends or_subckt
93
94 .subckt or3_subckt a b c y vdd gnd
95 .param w_OR_P3 = {6*W}
96 .param w_OR_N3 = {W}
97 // NOR
98 M1 J1 a vdd vdd CMOSP W={w_OR_P3} L={2*LAMBDA}
99 + AS={5*w_OR_P3*LAMBDA} PS={10*LAMBDA+2*w_OR_P3} AD={5*w_OR_P3*LAMBDA} PD={10*LAMBDA+2*w_OR_P3}
100
101 M2 J2 b J1 J1 CMOSP W={w_OR_P3} L={2*LAMBDA}
102 + AS={5*w_OR_P3*LAMBDA} PS={10*LAMBDA+2*w_OR_P3} AD={5*w_OR_P3*LAMBDA} PD={10*LAMBDA+2*w_OR_P3}
103
104 M3 nor c J2 J2 CMOSP W={w_OR_P3} L={2*LAMBDA}
105 + AS={5*w_OR_P3*LAMBDA} PS={10*LAMBDA+2*w_OR_P3} AD={5*w_OR_P3*LAMBDA} PD={10*LAMBDA+2*w_OR_P3}
106
107 M4 nor a gnd gnd CMOSN W={w_OR_N3} L={2*LAMBDA}
108 + AS={5*w_OR_N3*LAMBDA} PS={10*LAMBDA+2*w_OR_N3} AD={5*w_OR_N3*LAMBDA} PD={10*LAMBDA+2*w_OR_N3}
109
110 M5 nor b gnd gnd CMOSN W={w_OR_N3} L={2*LAMBDA}
111 + AS={5*w_OR_N3*LAMBDA} PS={10*LAMBDA+2*w_OR_N3} AD={5*w_OR_N3*LAMBDA} PD={10*LAMBDA+2*w_OR_N3}

```

```

114 + AS={5*w_OR_N3*LAMBDA} PS={10*LAMBDA+2*w_OR_N3} AD={5*w_OR_N3*LAMBDA} PD={10*LAMBDA+2*w_OR_N3}
115
116 // Inverter
117 M7      y      nor      vdd      vdd  CMOSP   W={width_P}   L={2*LAMBDA}
118 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
119
120 M8      y      nor      gnd      gnd  CMOSN   W={width_N}   L={2*LAMBDA}
121 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
122 .ends or3_subckt
123
124 .subckt or4_subckt a b c d y vdd gnd
125 .param w_OR_P4 = {8*W}
126 .param w_OR_N4 = {W}
127 // NOR
128 M1      J1      a      vdd      vdd  CMOSP   W={w_OR_P4}   L={2*LAMBDA}
129 + AS={5*w_OR_P4*LAMBDA} PS={10*LAMBDA+2*w_OR_P4} AD={5*w_OR_P4*LAMBDA} PD={10*LAMBDA+2*w_OR_P4}
130
131 M2      J2      b      J1      J1  CMOSP   W={w_OR_P4}   L={2*LAMBDA}
132 + AS={5*w_OR_P4*LAMBDA} PS={10*LAMBDA+2*w_OR_P4} AD={5*w_OR_P4*LAMBDA} PD={10*LAMBDA+2*w_OR_P4}
133
134 M3      J3      c      J2      J2  CMOSP   W={w_OR_P4}   L={2*LAMBDA}
135 + AS={5*w_OR_P4*LAMBDA} PS={10*LAMBDA+2*w_OR_P4} AD={5*w_OR_P4*LAMBDA} PD={10*LAMBDA+2*w_OR_P4}
136
137 M4      nor      d      J3      J3  CMOSP   W={w_OR_P4}   L={2*LAMBDA}
138 + AS={5*w_OR_P4*LAMBDA} PS={10*LAMBDA+2*w_OR_P4} AD={5*w_OR_P4*LAMBDA} PD={10*LAMBDA+2*w_OR_P4}
139
140 M5      nor      a      gnd      gnd  CMOSN   W={w_OR_N4}   L={2*LAMBDA}
141 + AS={5*w_OR_N4*LAMBDA} PS={10*LAMBDA+2*w_OR_N4} AD={5*w_OR_N4*LAMBDA} PD={10*LAMBDA+2*w_OR_N4}
142
143 M6      nor      b      gnd      gnd  CMOSN   W={w_OR_N4}   L={2*LAMBDA}
144 + AS={5*w_OR_N4*LAMBDA} PS={10*LAMBDA+2*w_OR_N4} AD={5*w_OR_N4*LAMBDA} PD={10*LAMBDA+2*w_OR_N4}
145
146 M7      nor      c      gnd      gnd  CMOSN   W={w_OR_N4}   L={2*LAMBDA}
147 + AS={5*w_OR_N4*LAMBDA} PS={10*LAMBDA+2*w_OR_N4} AD={5*w_OR_N4*LAMBDA} PD={10*LAMBDA+2*w_OR_N4}
148
149 M8      nor      d      gnd      gnd  CMOSN   W={w_OR_N4}   L={2*LAMBDA}
150 + AS={5*w_OR_N4*LAMBDA} PS={10*LAMBDA+2*w_OR_N4} AD={5*w_OR_N4*LAMBDA} PD={10*LAMBDA+2*w_OR_N4}
151
152 // Inverter
153 M9      y      nor      vdd      vdd  CMOSP   W={width_P}   L={2*LAMBDA}
154 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
155
156 M10     y      nor      gnd      gnd  CMOSN   W={width_N}   L={2*LAMBDA}
157 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
158 .ends or4_subckt
159
160 .subckt and3_subckt a b c y vdd gnd
161 .param w_AND_P3 = {2*W}
162 .param w_AND_N3 = {3*W}
163 //NAND
164 M1      nand      a      vdd      vdd  CMOSP   W={w_AND_P3}   L={2*LAMBDA}
165 + AS={5*w_AND_P3*LAMBDA} PS={10*LAMBDA+2*w_AND_P3} AD={5*w_AND_P3*LAMBDA} PD={10*LAMBDA+2*w_AND_P3}
166
167 M2      nand      b      vdd      vdd  CMOSP   W={w_AND_P3}   L={2*LAMBDA}
168 + AS={5*w_AND_P3*LAMBDA} PS={10*LAMBDA+2*w_AND_P3} AD={5*w_AND_P3*LAMBDA} PD={10*LAMBDA+2*w_AND_P3}
169

```

```

177 // AS={5*w_AND_N3*LAMBDA} PS={10*LAMBDA+2*w_AND_N3} AD={5*w_AND_N3*LAMBDA} PD={10*LAMBDA+2*w_AND_N3}
178 M6 J2 c gnd gnd CMOSN W={w_AND_N3} L={2*LAMBDA}
179 + AS={5*w_AND_N3*LAMBDA} PS={10*LAMBDA+2*w_AND_N3} AD={5*w_AND_N3*LAMBDA} PD={10*LAMBDA+2*w_AND_N3}
180
181 //Inverter
182 M7 y nand vdd vdd CMOSP W={width_P} L={2*LAMBDA}
183 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
184
185 M8 y nand gnd gnd CMOSN W={width_N} L={2*LAMBDA}
186 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
187 .ends and3_subckt
188
189 .subckt and4_subckt a b c d y vdd gnd
190 .param w_AND_P4 = {2*W}
191 .param w_AND_N4 = {4*W}
192 //NAND
193 M1 nand a vdd vdd CMOSP W={w_AND_P4} L={2*LAMBDA}
194 + AS={5*w_AND_P4*LAMBDA} PS={10*LAMBDA+2*w_AND_P4} AD={5*w_AND_P4*LAMBDA} PD={10*LAMBDA+2*w_AND_P4}
195
196 M2 nand b vdd vdd CMOSP W={w_AND_P4} L={2*LAMBDA}
197 + AS={5*w_AND_P4*LAMBDA} PS={10*LAMBDA+2*w_AND_P4} AD={5*w_AND_P4*LAMBDA} PD={10*LAMBDA+2*w_AND_P4}
198
199 M3 nand c vdd vdd CMOSP W={w_AND_P4} L={2*LAMBDA}
200 + AS={5*w_AND_N4*LAMBDA} PS={10*LAMBDA+2*w_AND_N4} AD={5*w_AND_N4*LAMBDA} PD={10*LAMBDA+2*w_AND_N4}
201
202 M8 J3 d gnd gnd CMOSN W={w_AND_N4} L={2*LAMBDA}
203 + AS={5*w_AND_N4*LAMBDA} PS={10*LAMBDA+2*w_AND_N4} AD={5*w_AND_N4*LAMBDA} PD={10*LAMBDA+2*w_AND_N4}
204
205 //Inverter
206 M9 y nand vdd vdd CMOSP W={width_P} L={2*LAMBDA}
207 + AS={5*width_P*LAMBDA} PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}
208
209 M10 y nand gnd gnd CMOSN W={width_N} L={2*LAMBDA}
210 + AS={5*width_N*LAMBDA} PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}
211 .ends and4_subckt
212
213
214
215
216
217 // Carry Calculation
218 // Cout1
219 // Cout2
220 | x9 P2 G1 T2_1 vdd gnd and_subckt
221 | x10 G2 T2_1 Cout2 vdd gnd or_subckt
222 // Cout3
223 | x11 P3 G2 T3_1 vdd gnd and_subckt
224 | x12 P3 P2 G1 T3_2 vdd gnd and3_subckt
225 | x13 G3 T3_1 T3_2 Cout3 vdd gnd or3_subckt
226 // Cout4
227 | x14 P4 G3 T4_1 vdd gnd and_subckt
228 | x15 P4 P3 G2 T4_2 vdd gnd and3_subckt
229 | x16 P4 P3 P2 G1 T4_3 vdd gnd and4_subckt
230 | x17 G4 T4_1 T4_2 T4_3 Cout4 vdd gnd or4_subckt
231
232

```

Live Share

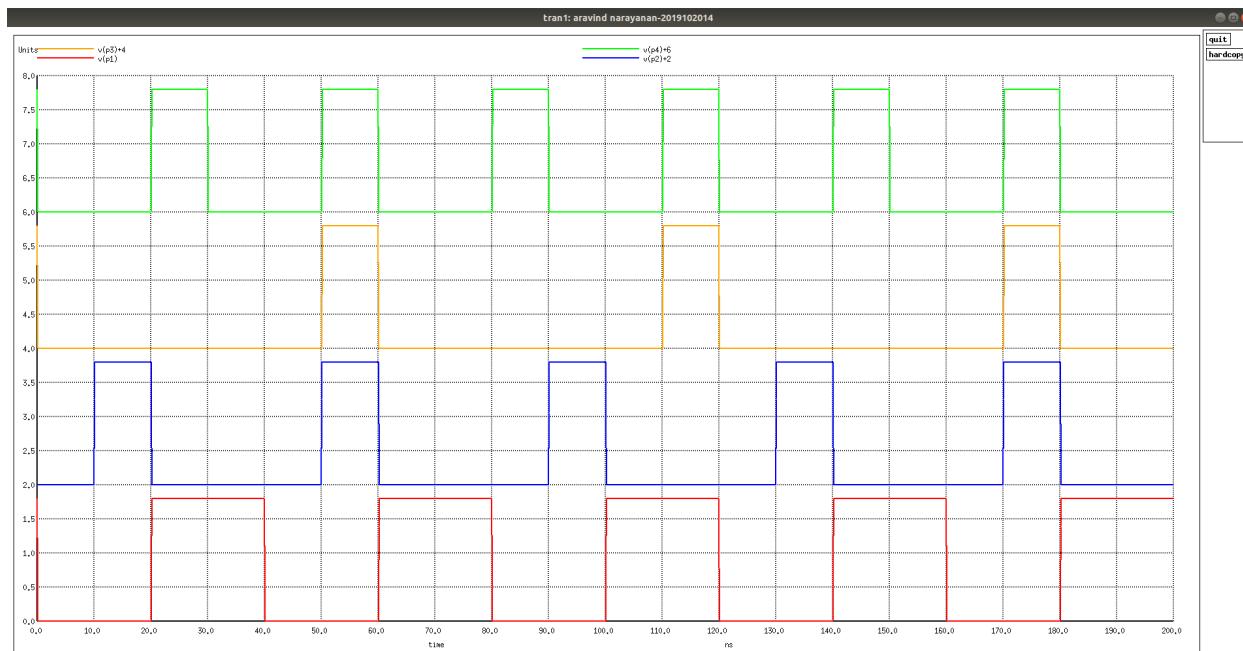
```

232
233
234 .tran 1n 200n
235 .control
236 set hcopypscolor = 1
237 set color0=white
238 set color1=black
239
240 run
241 plot v(G1) v(Cout2)+2 v(Cout3)+4 v(Cout4)+6
242 set curplottitle= "Aravind Narayanan-2019102014"
243 plot v(G1) v(G2)+2 v(G3)+4 v(G4)+6
244 set curplottitle= "Aravind Narayanan-2019102014"
245 plot v(P1) v(P2)+2 v(P3)+4 v(P4)+6
246 set curplottitle= "Aravind Narayanan-2019102014"
247 .endc

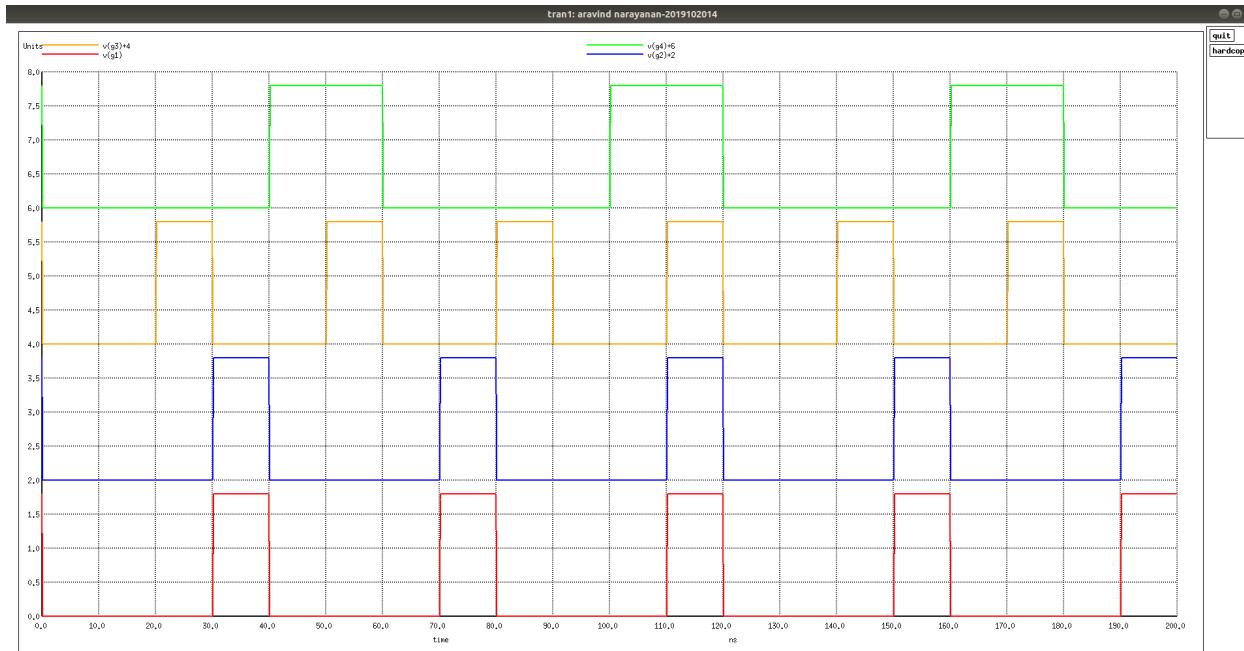
```

Simulation Output:

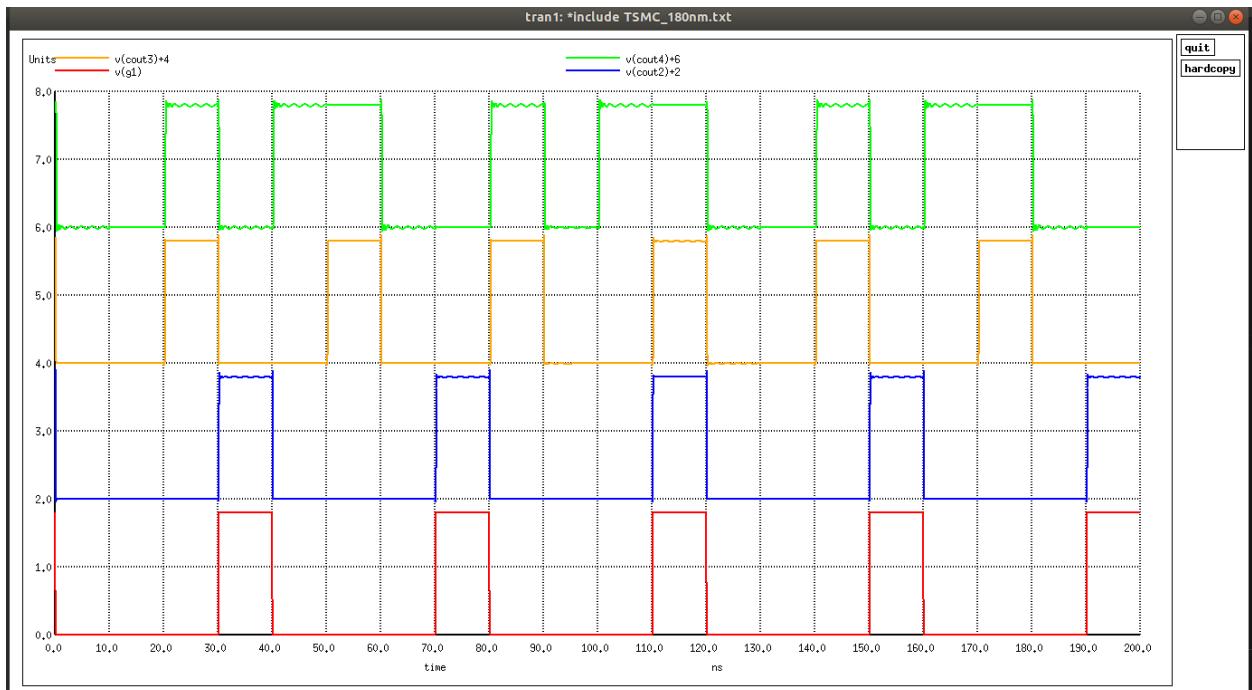
Propagate



Generate



Carry:



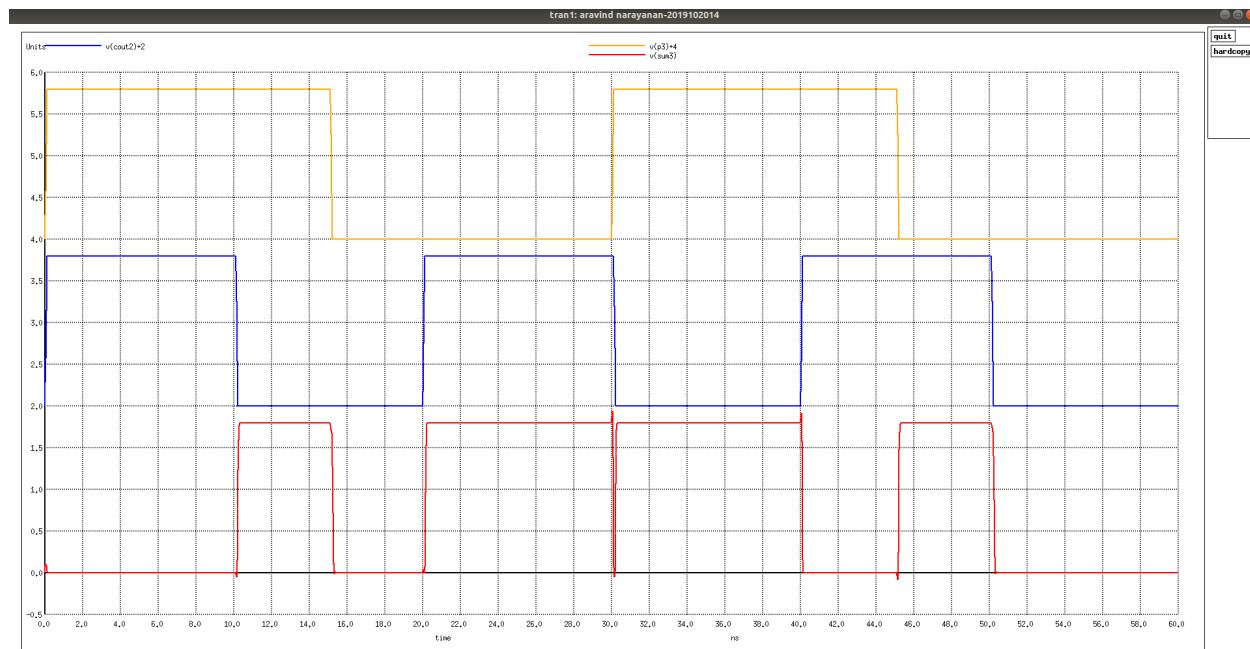
Delay:

The delay is measured for the final Carry Out(Cout4) and is obtained as follows as $2.17518e-10s \Rightarrow 0.22\text{ns}$.

This delay is also considered as the final delay of the Carry Generator Block.

(IV) Sum:

Simulation Output:



Explanation:

NOTE: The red color line shows the XOR function implemented and proper sum is obtained.

Delay:

The delay of the sum block = delay of XOR gate i.e similar to propagate module calculated earlier i.e. 0.029ns.

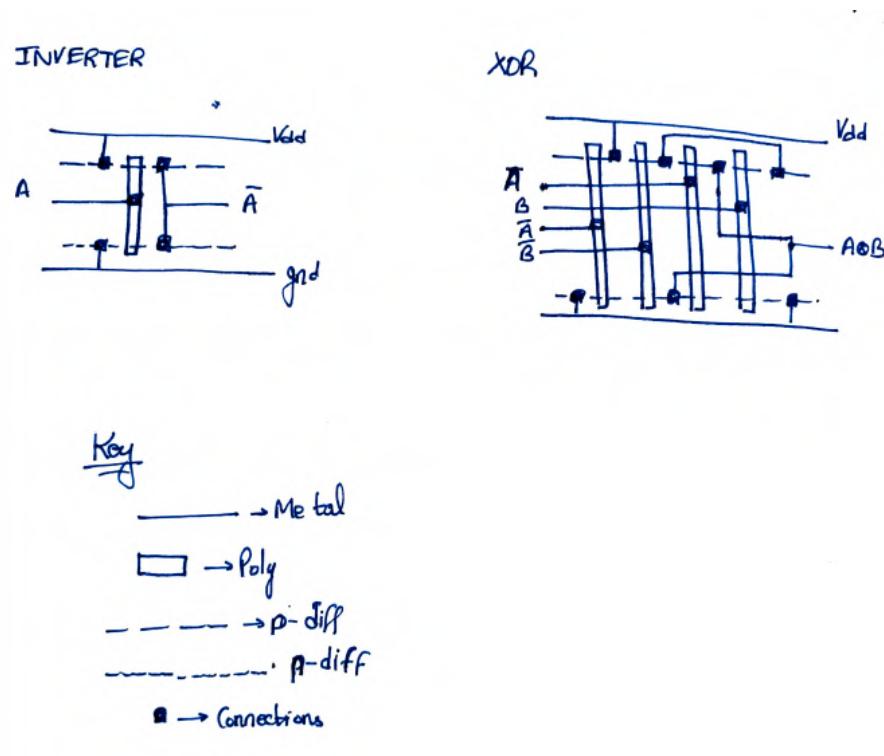
Question 4

We find the setup time by seeing how close the input can be to the clock trigger such that a proper output is maintained. As, if we go too close, we will get incorrect output or instability.

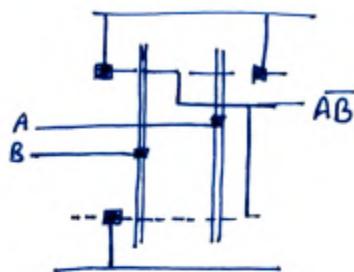
SETUP TIME: 0.23ns

Hold time is calculated by changing outputs immediately after the clock trigger to see how close we can get the output properly even after the change. It is found that the hold time is nearly **0 (around 0.004ns)** as the flip flop taken here is a TSPC latch based flip flop.

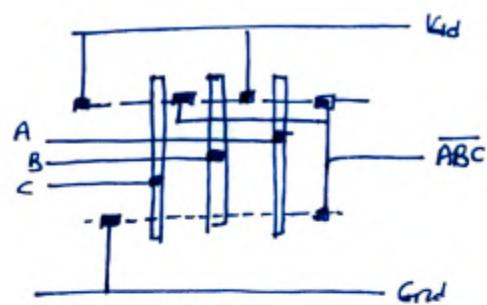
Question 5:



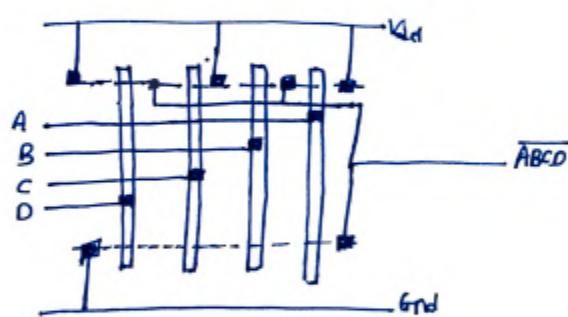
2-INPUT NAND



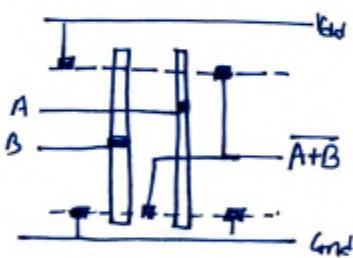
3 INPUT NAND



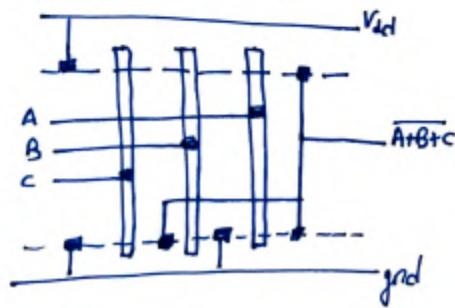
4 INPUT NAND



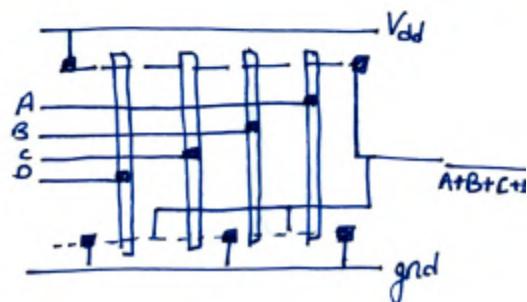
2 INPUT NOR



3-INPUT NOR

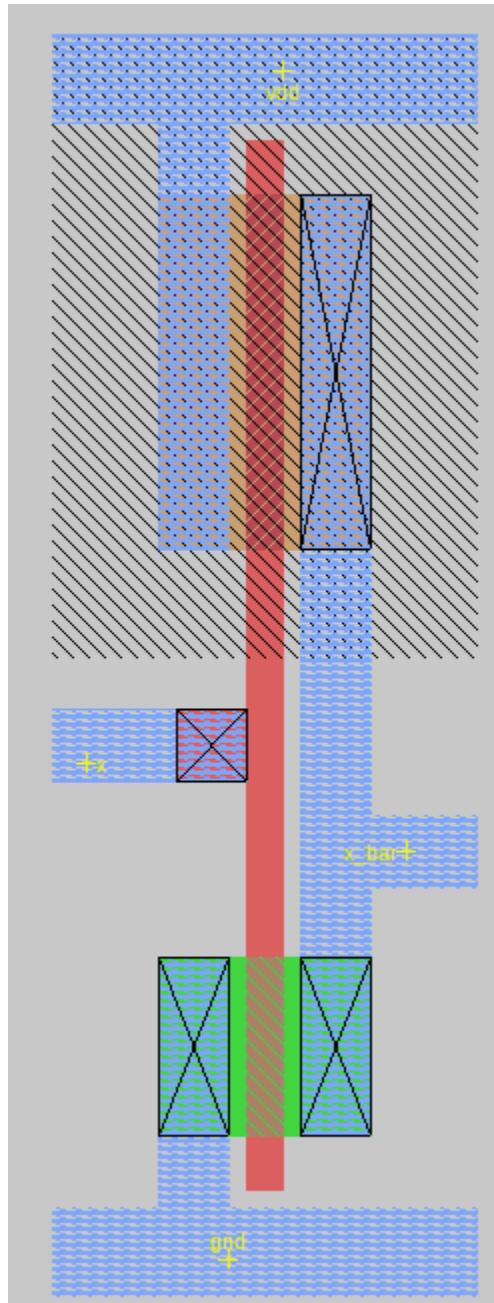


4 INPUT NOR

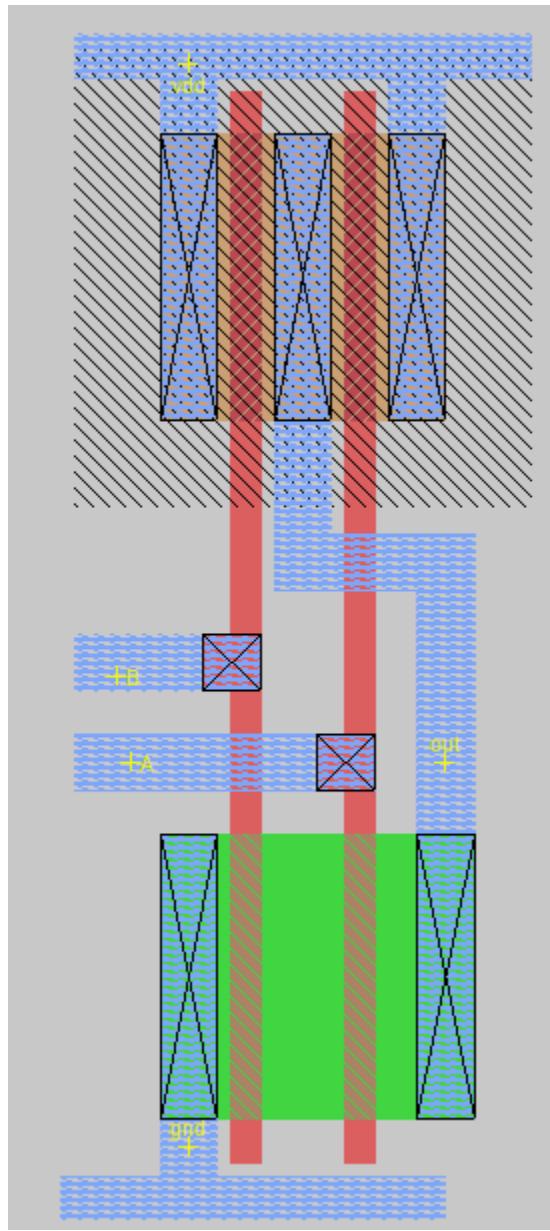


Question 6:

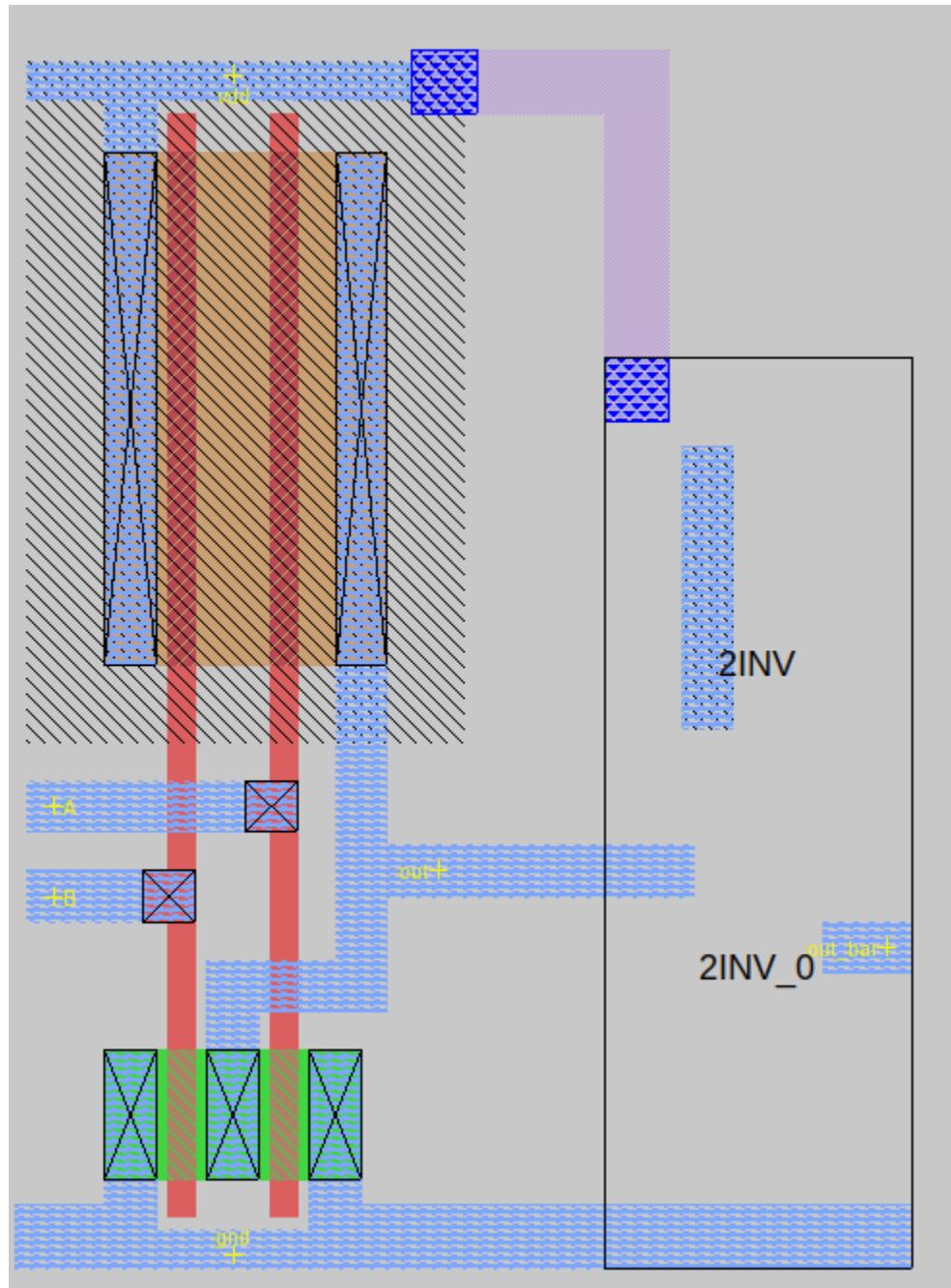
1) Inverter:



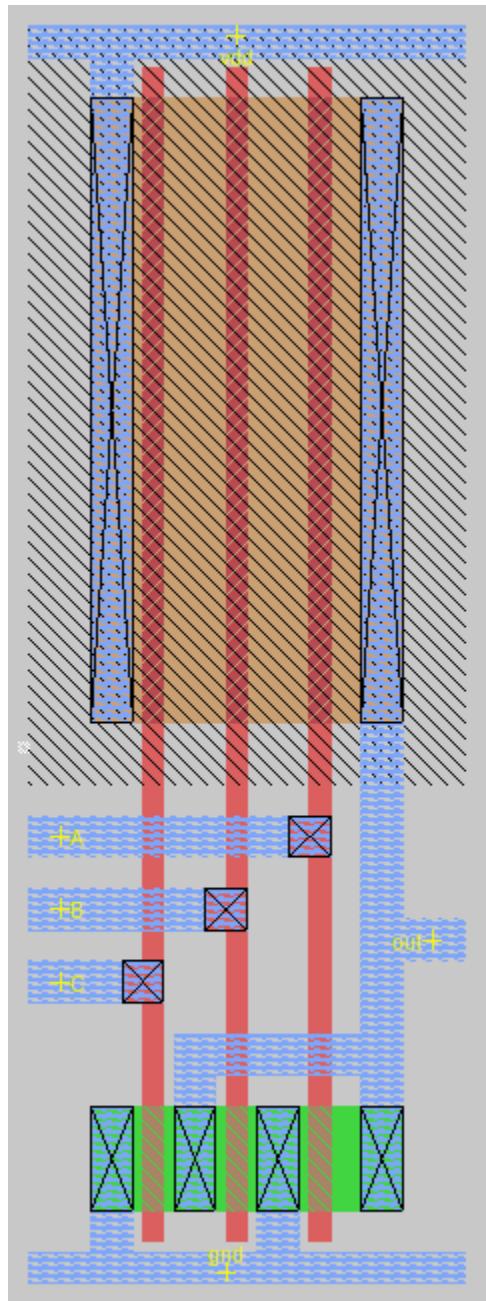
2) 2 INPUT NAND:



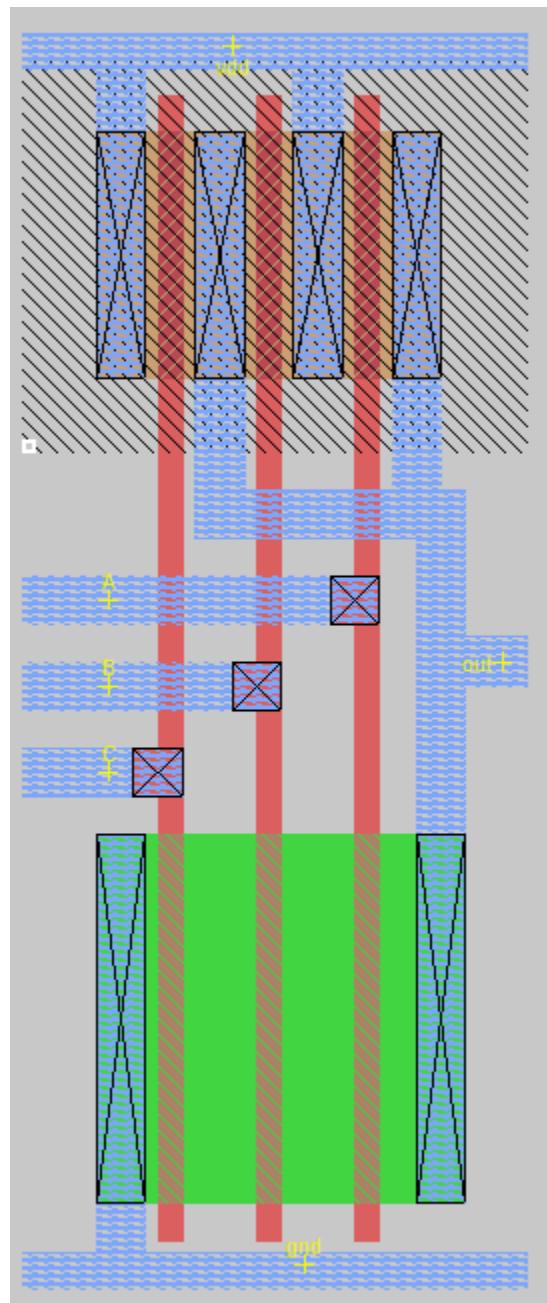
3) 2 INPUT OR



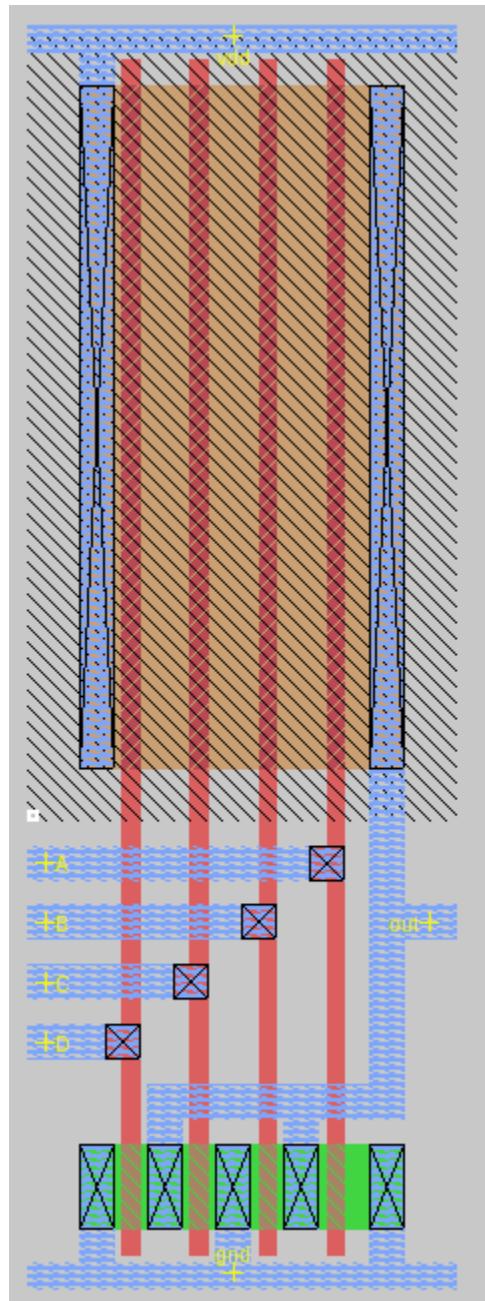
4) 3 INP NOR



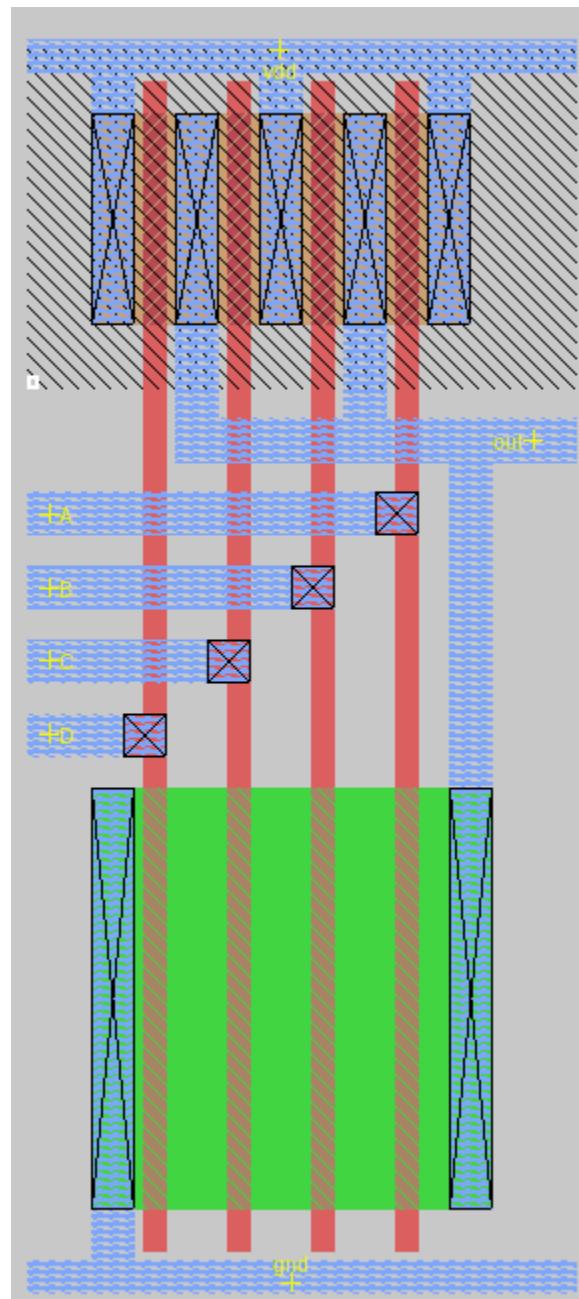
3) 3 INP NAND



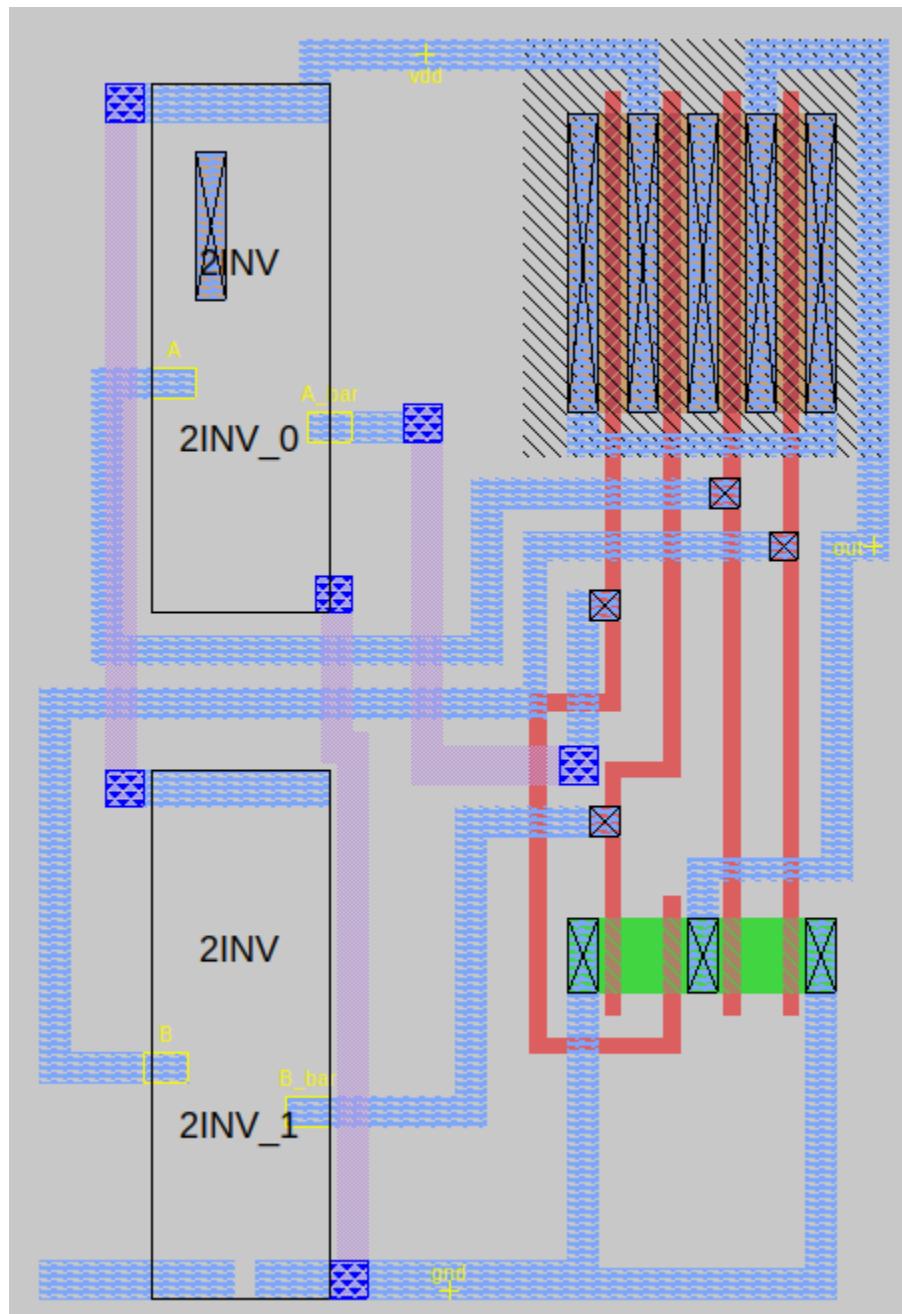
4) 4 INPUT NOR



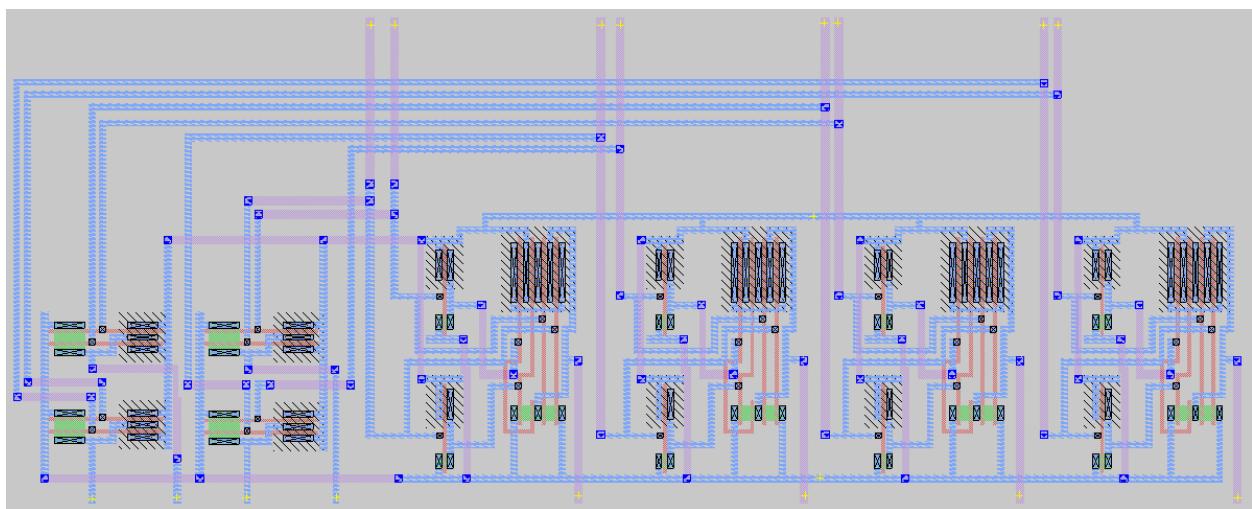
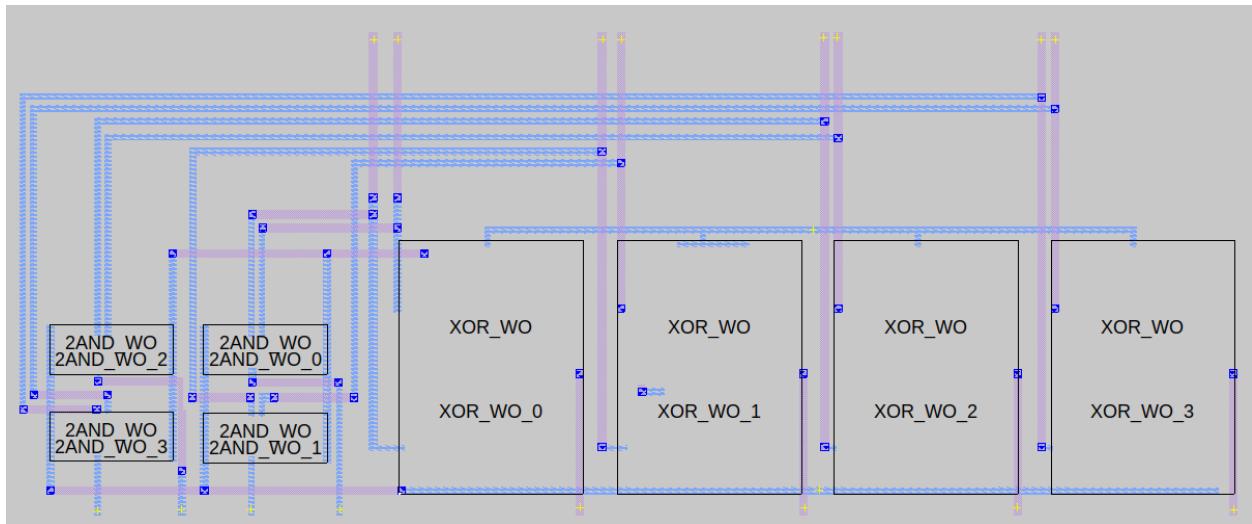
5) 4 INPUT NAND



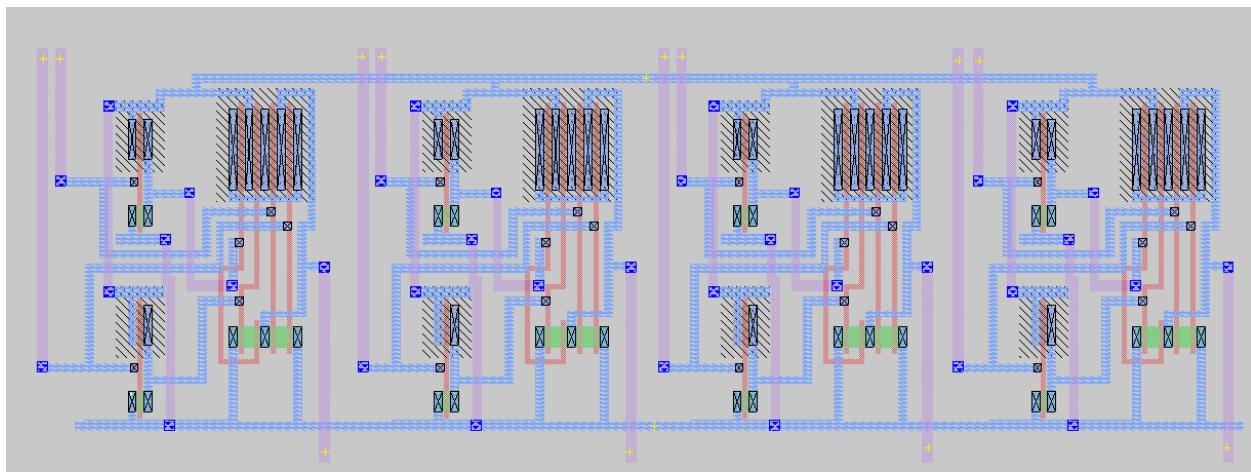
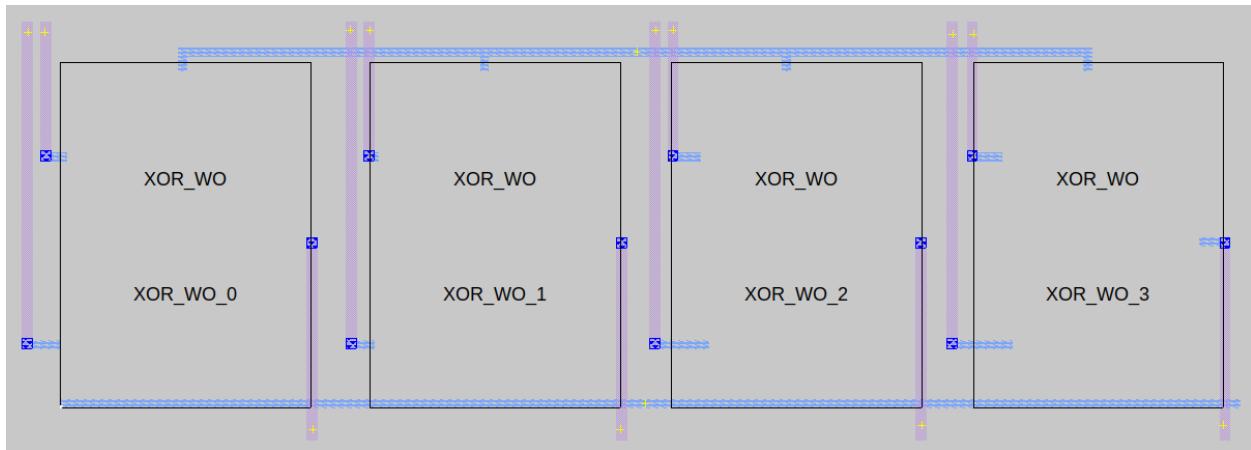
6) XOR



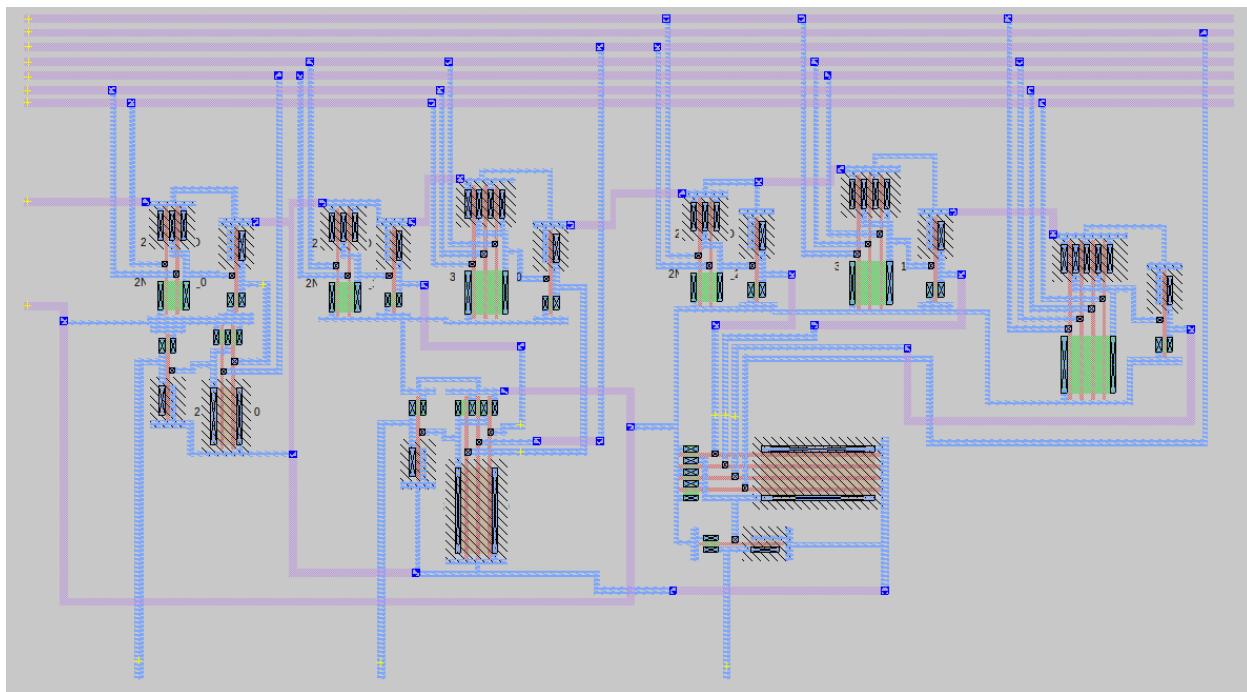
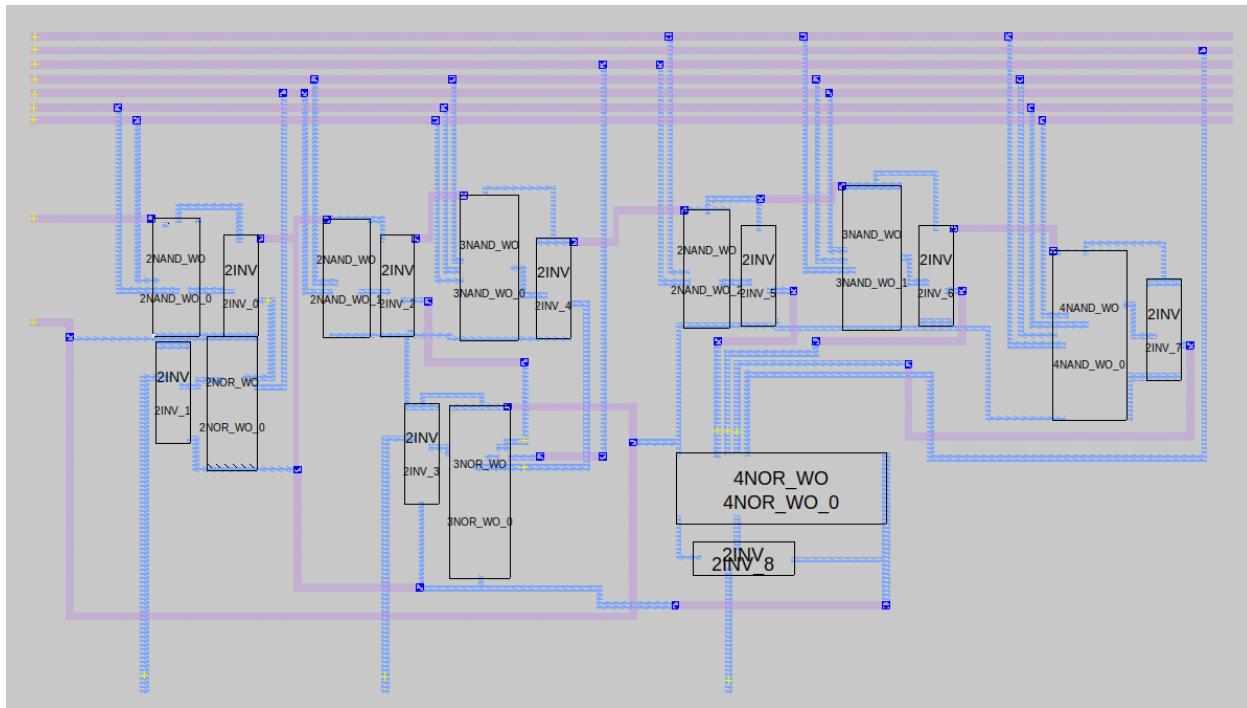
Propagate and Generate Block:



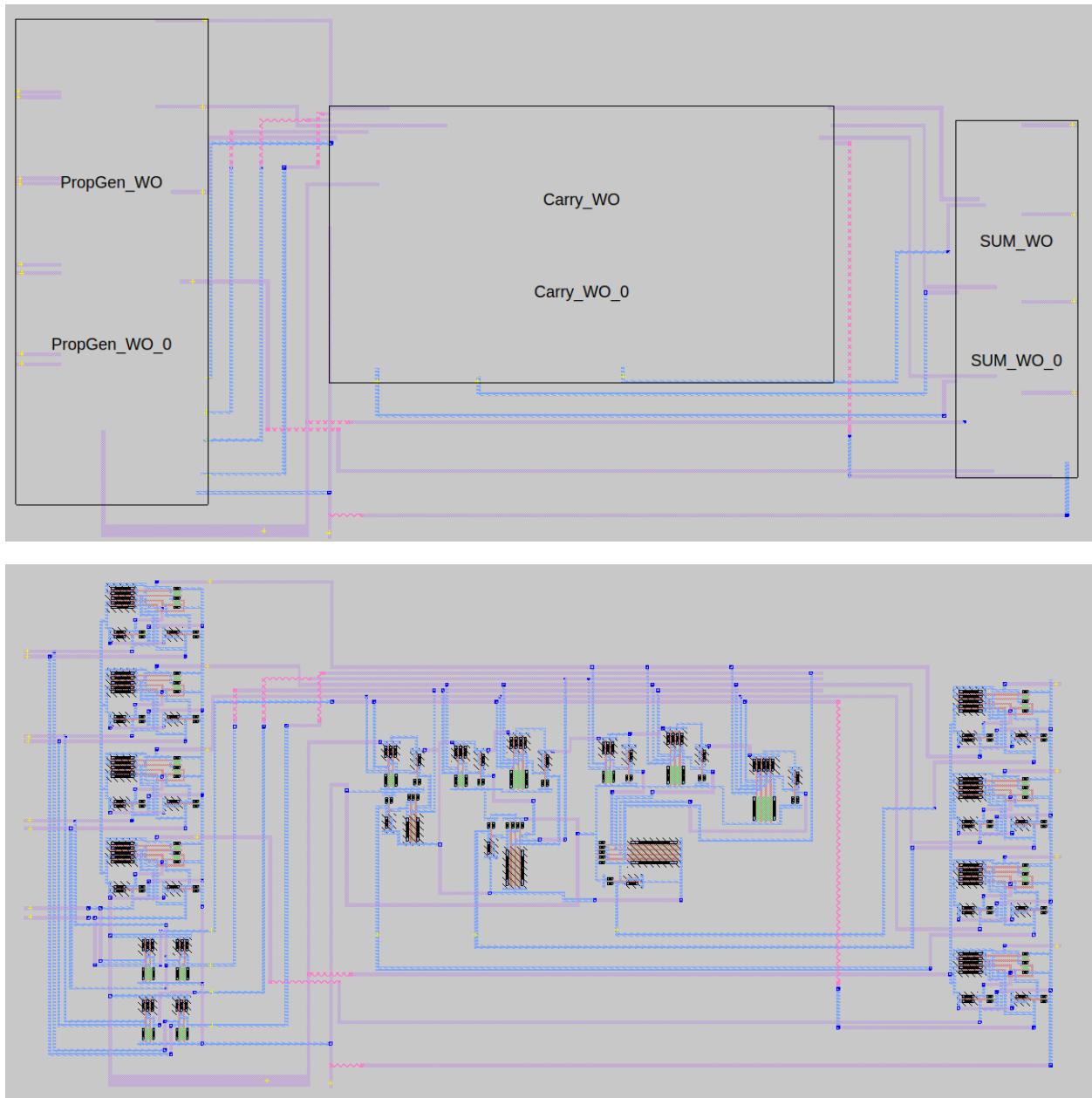
SUM Block:



Carry Block:

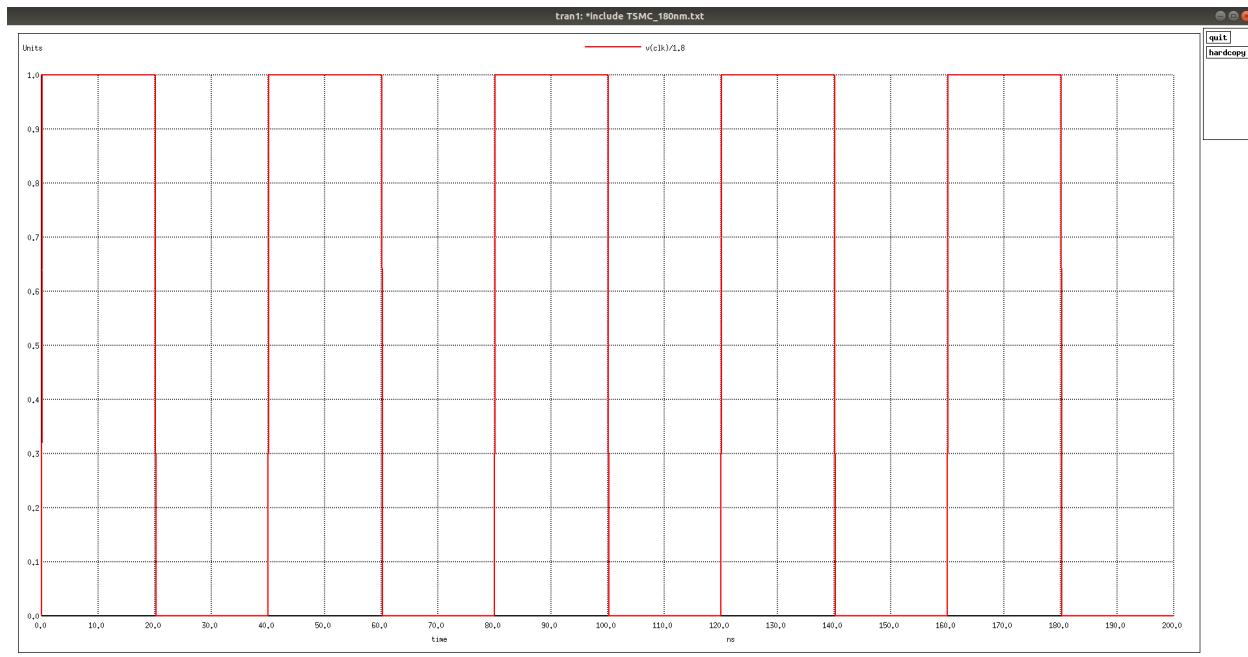


OVERALL CLA:

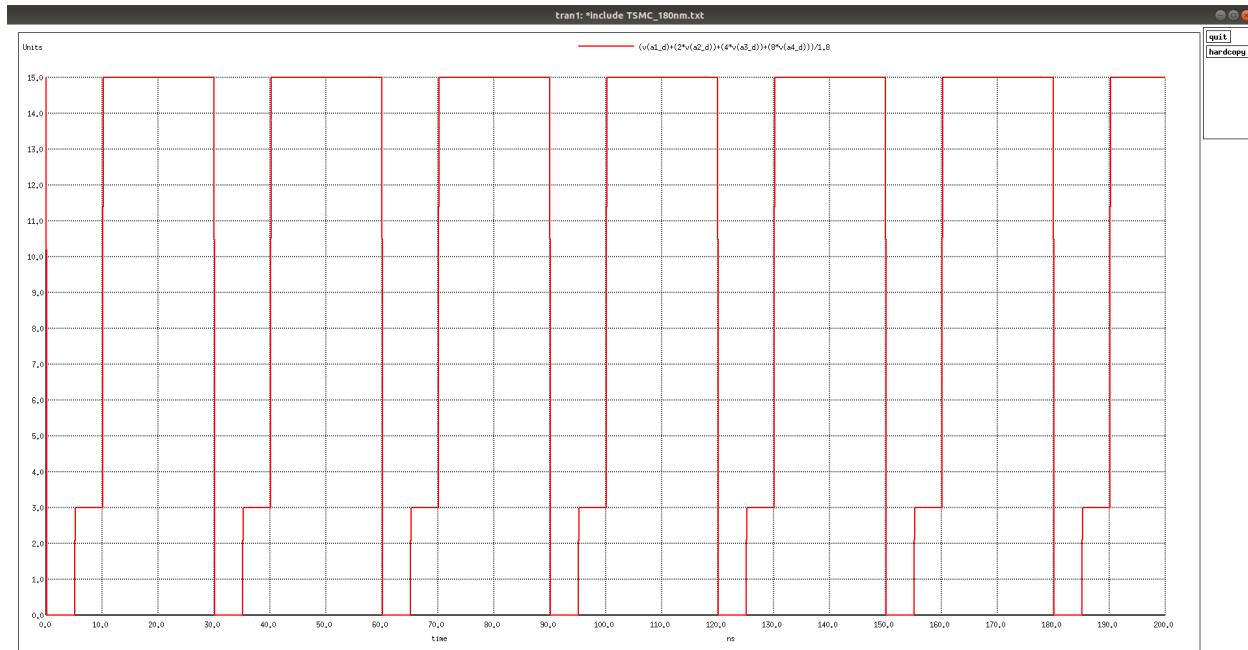


Question 7:

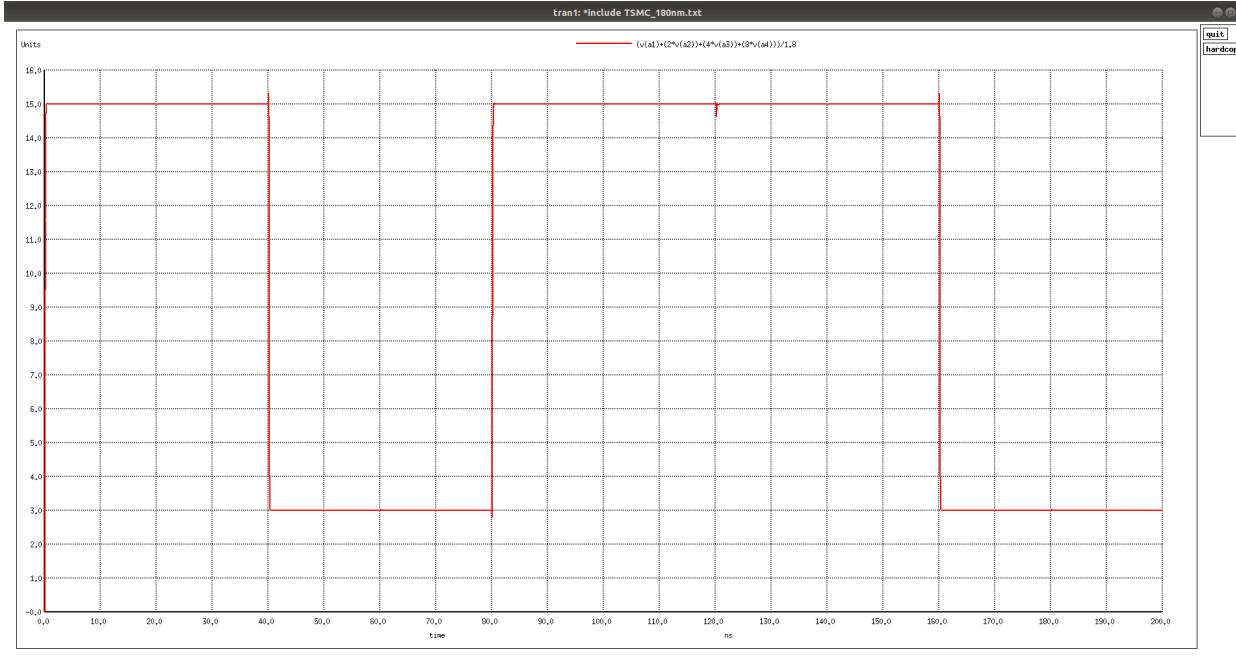
1) Clock:



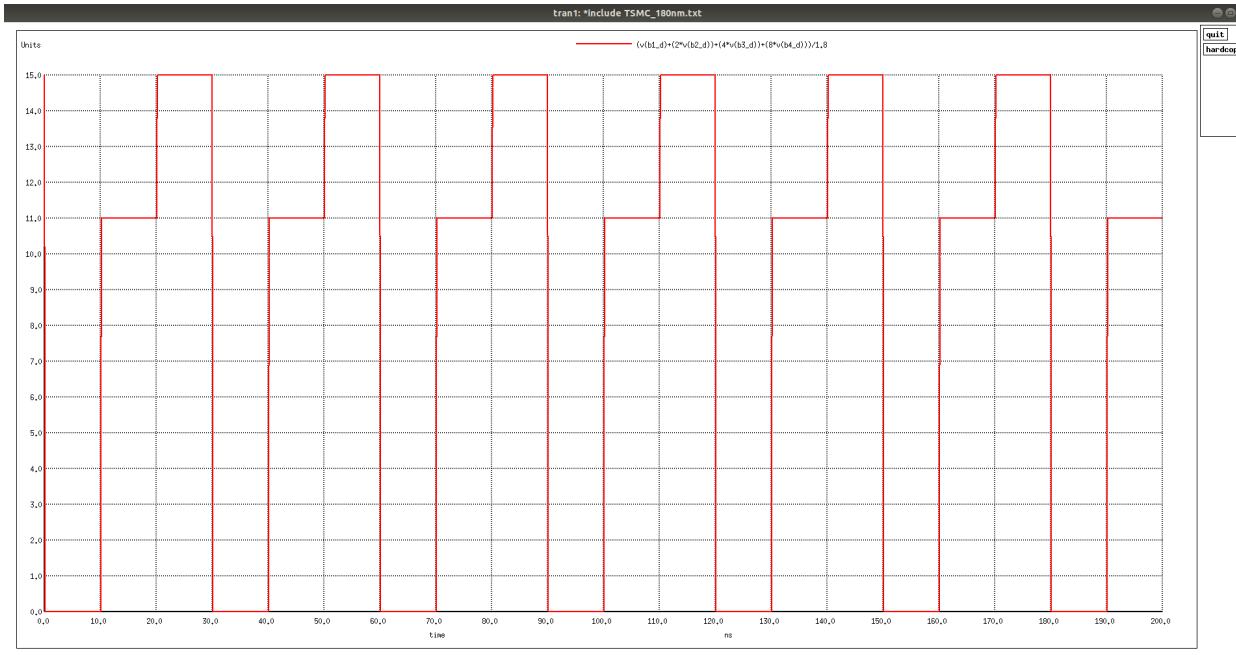
2) Input A to flip flop:



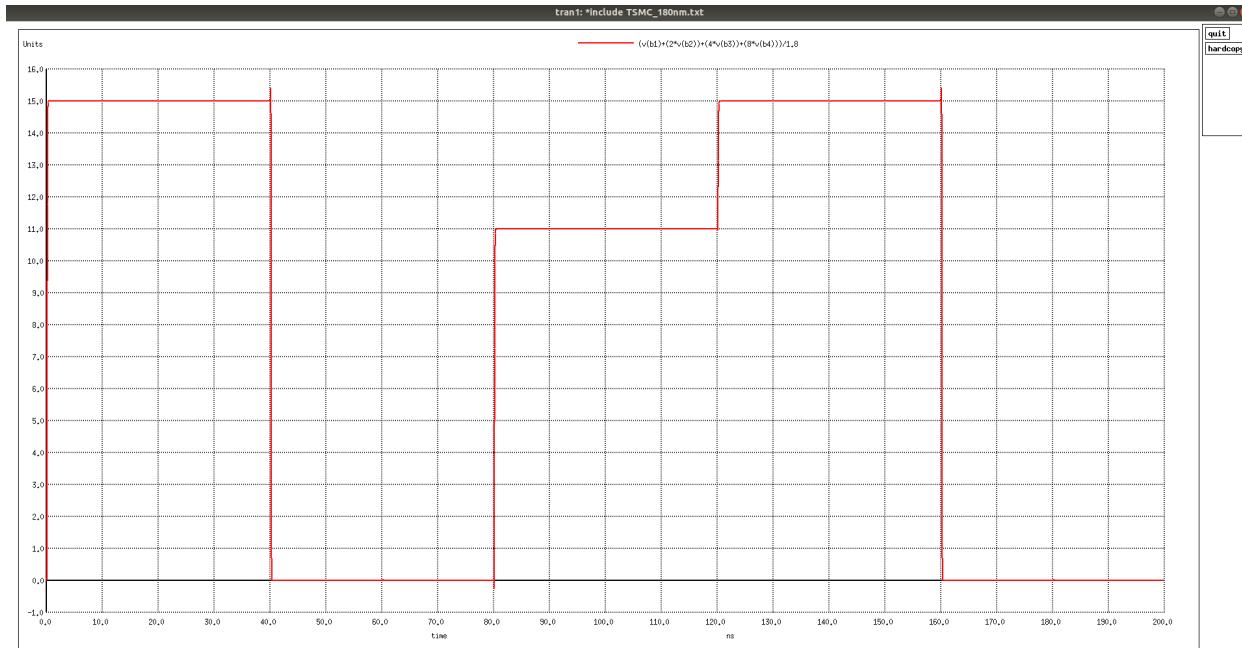
3) Input A After FlipFlop:



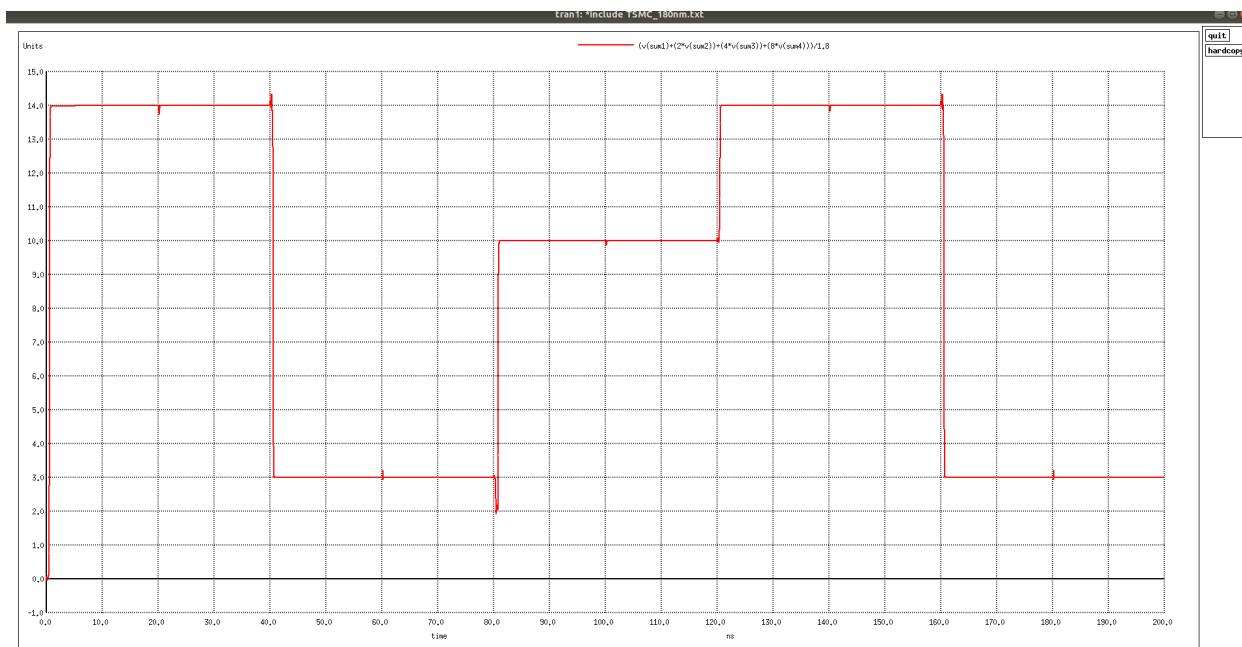
4) Input B to flip flop:



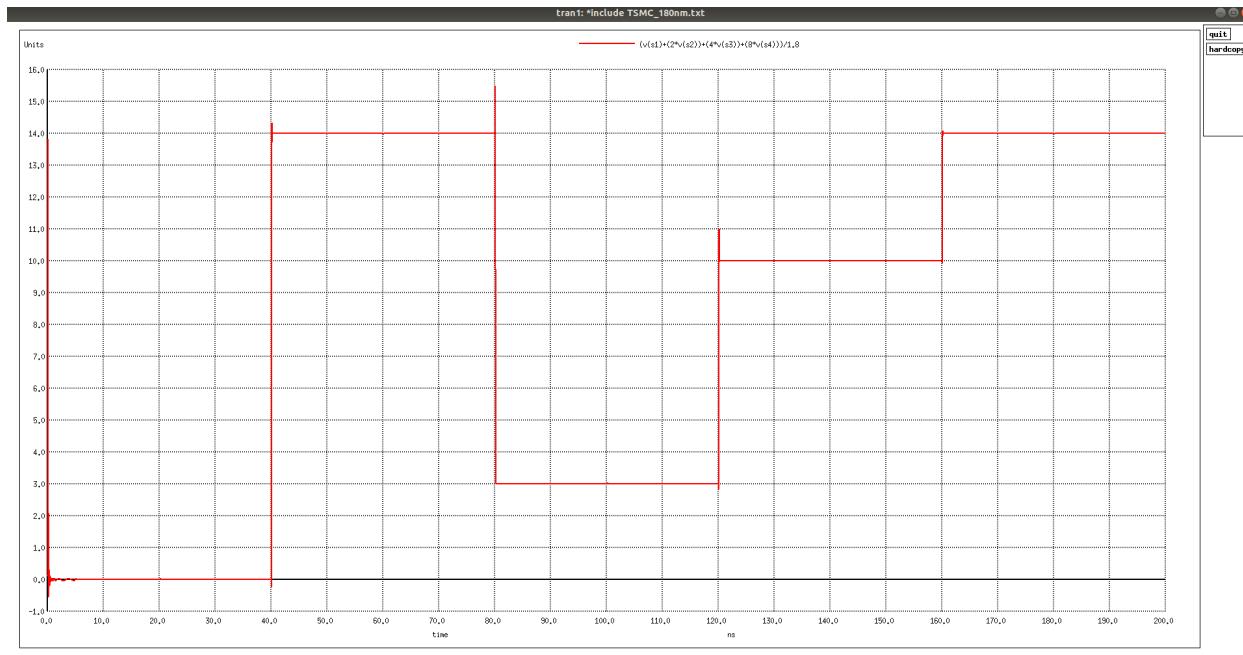
5) Input A After FlipFlop:



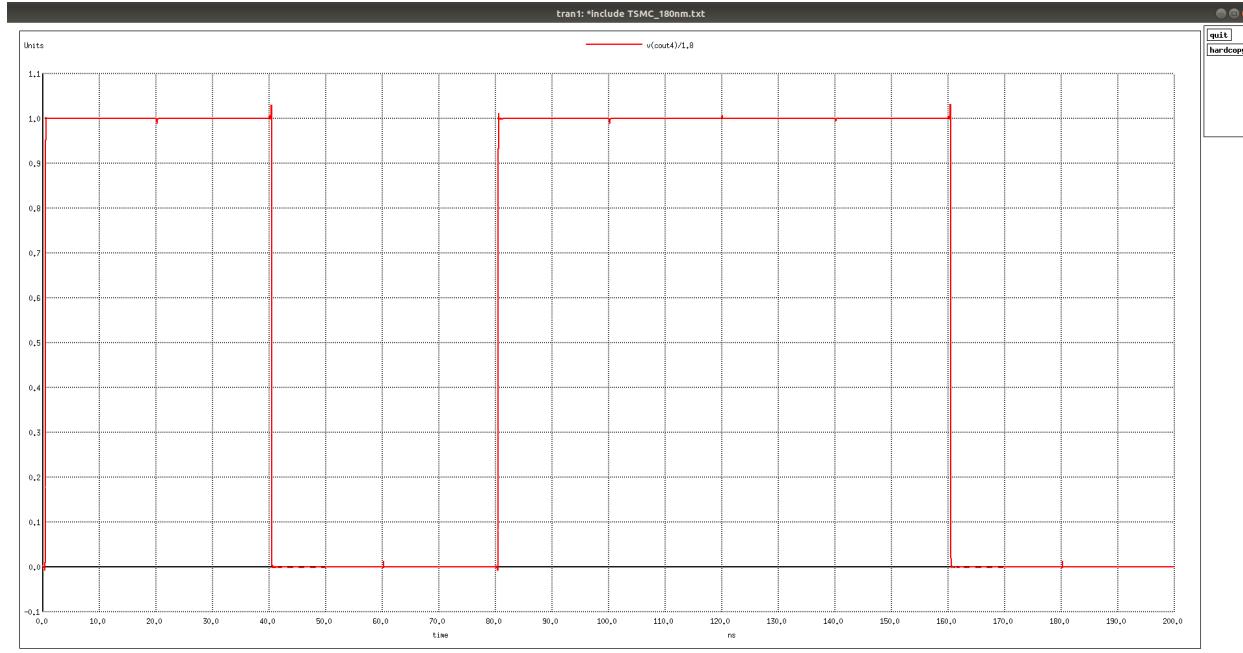
6) Sum from CLA:



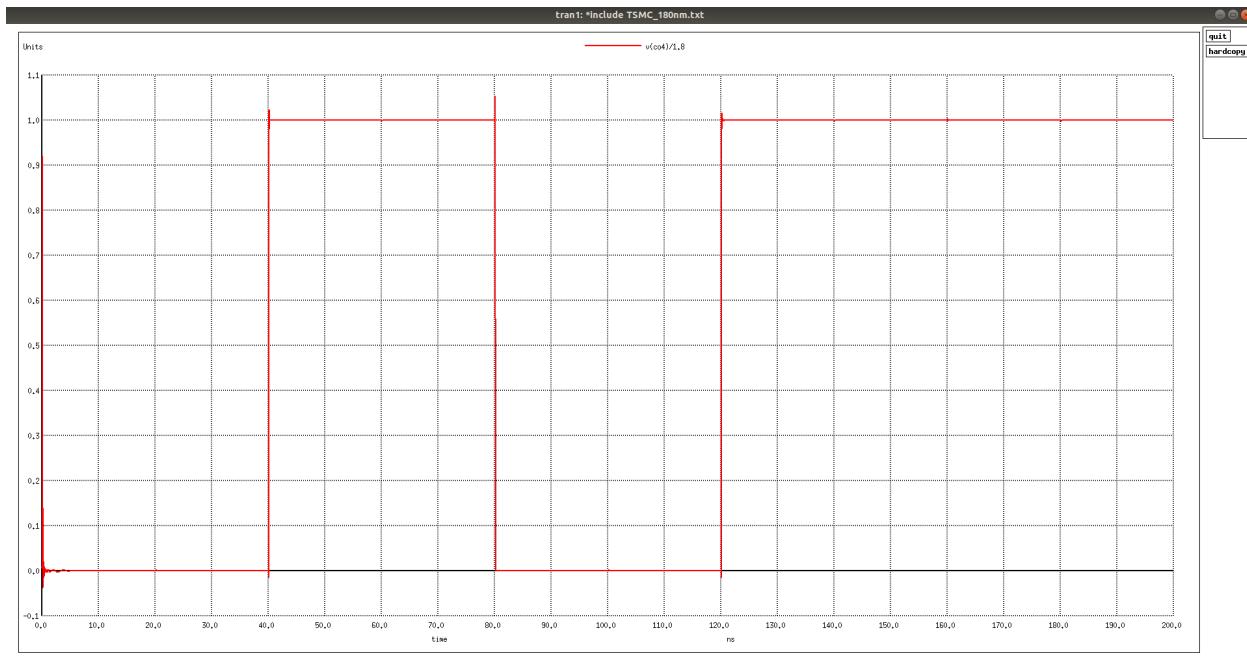
7) Sum after passing through flip-flop:



8) Cout of CLA:

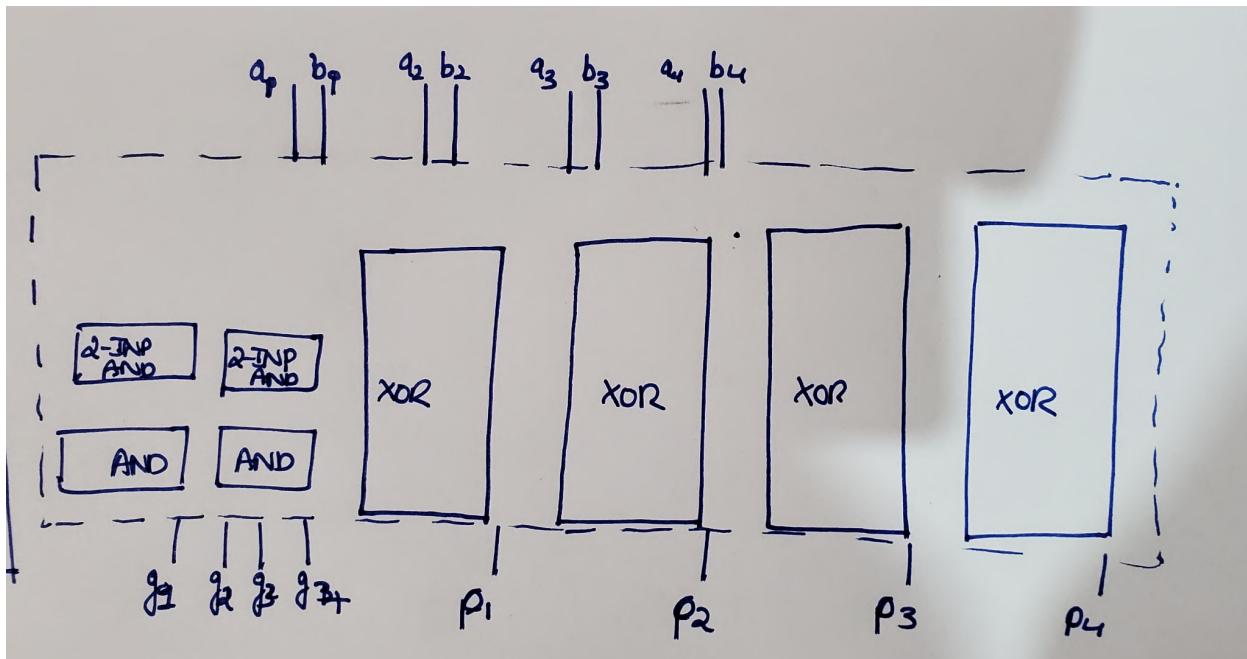


9) Cout after flip flop:



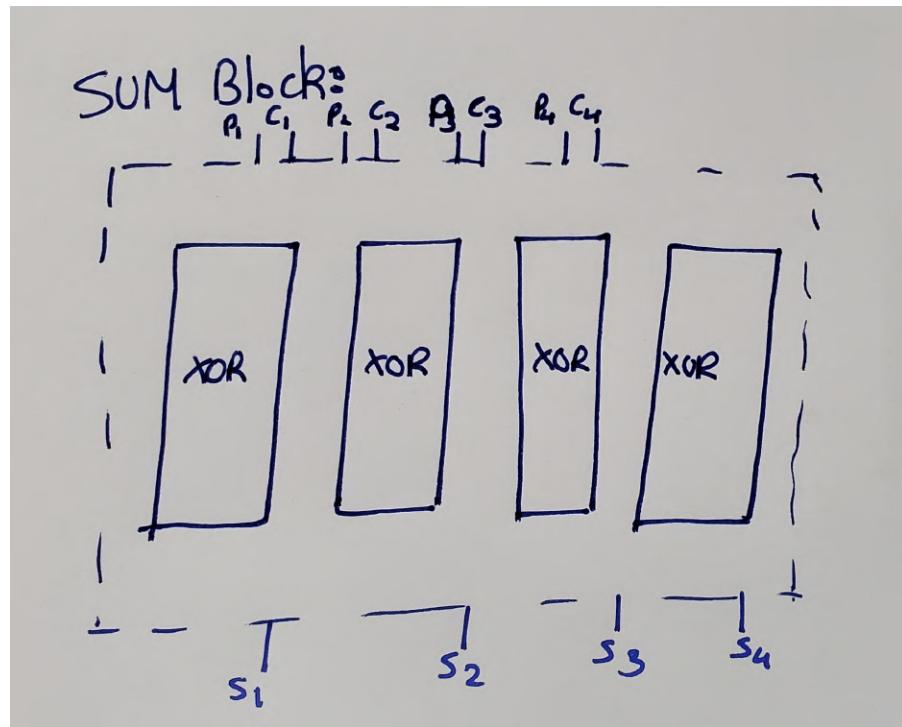
Question 8:

1) Prop and Generate:



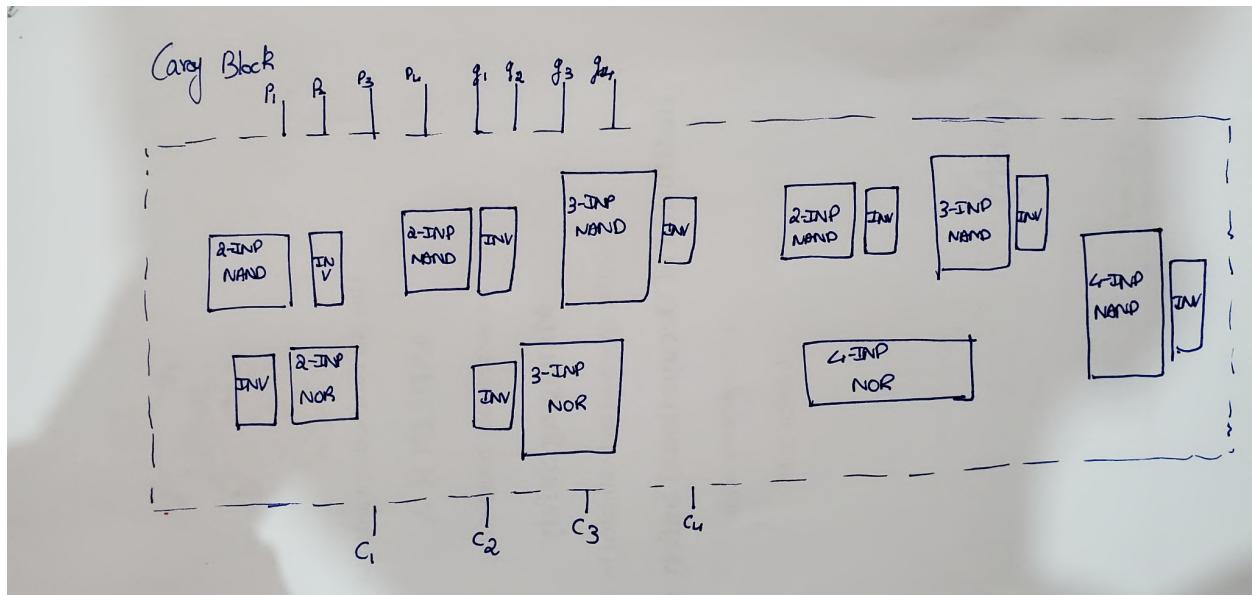
SIZE: 815 x 325 in terms of lambda

2) SUM BLOCK:



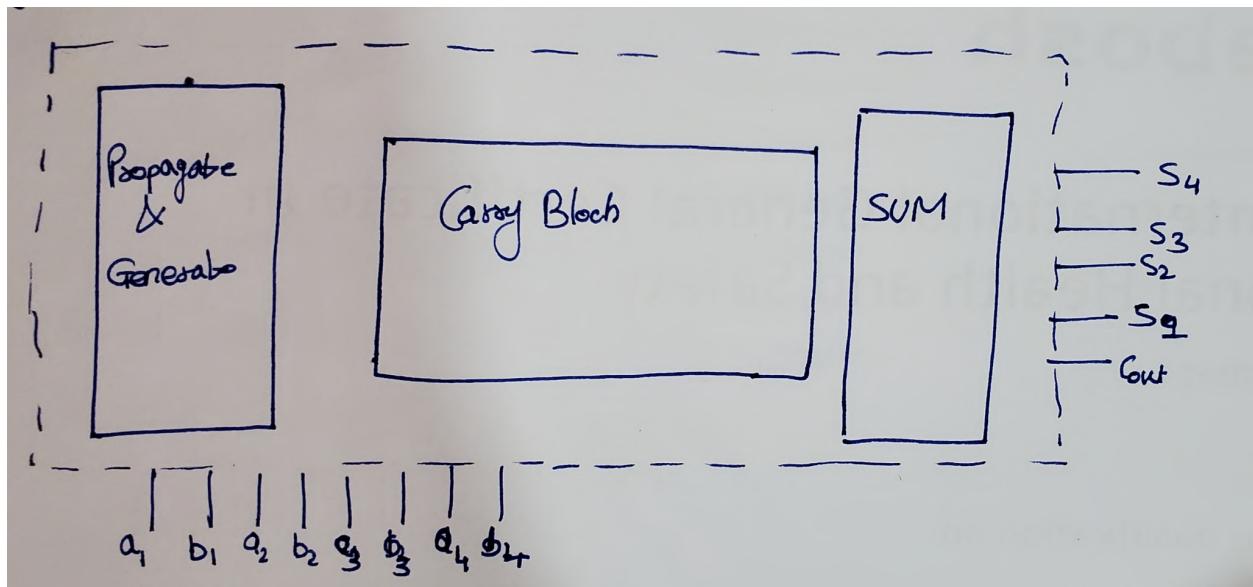
SIZE: 604 x 206 in terms of lambda

3)



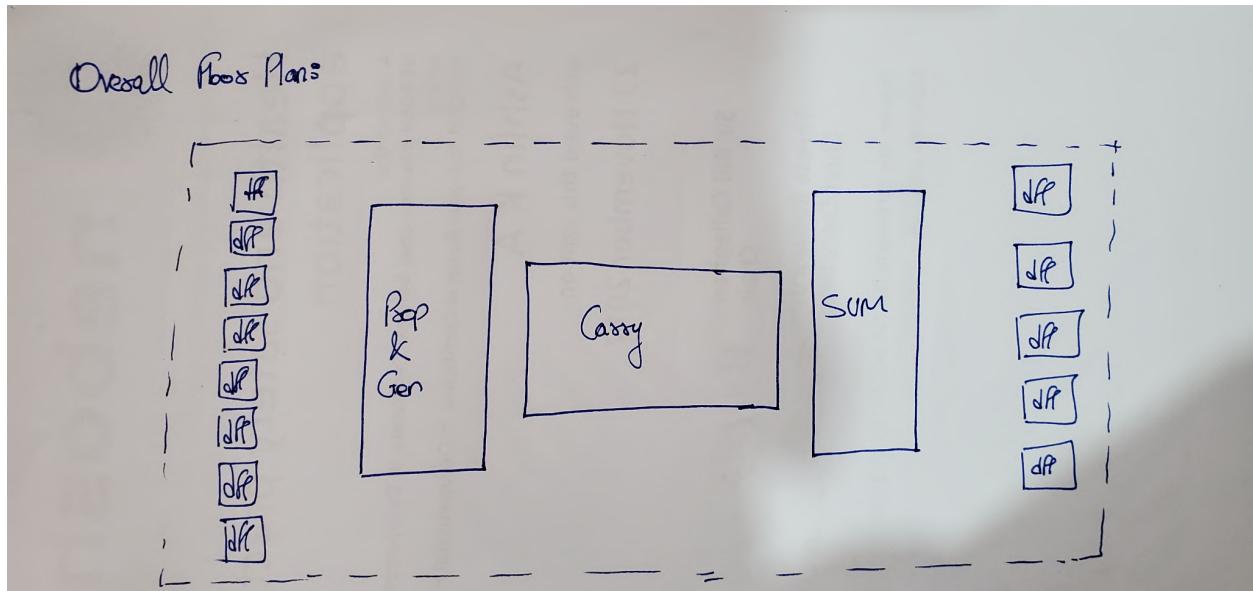
SIZE: 467 x 845 in terms of lambda

4) CLA



SIZE: 866 x 1806 in terms of lambda

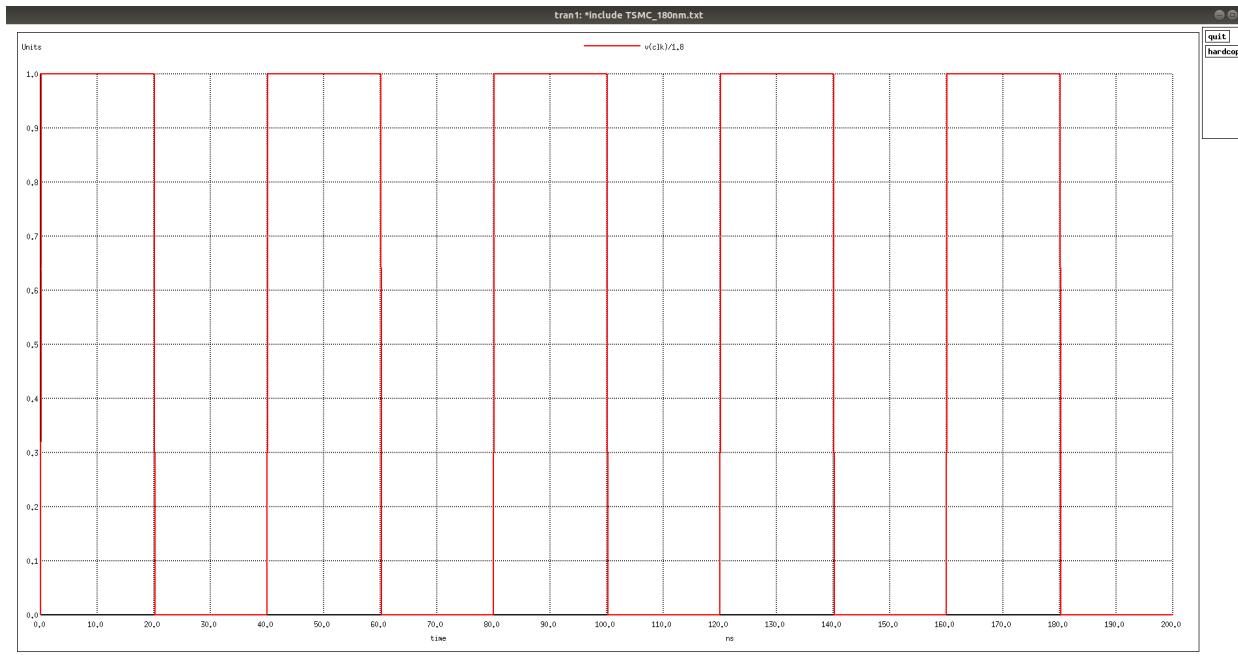
5) CLA with FlipFlops:



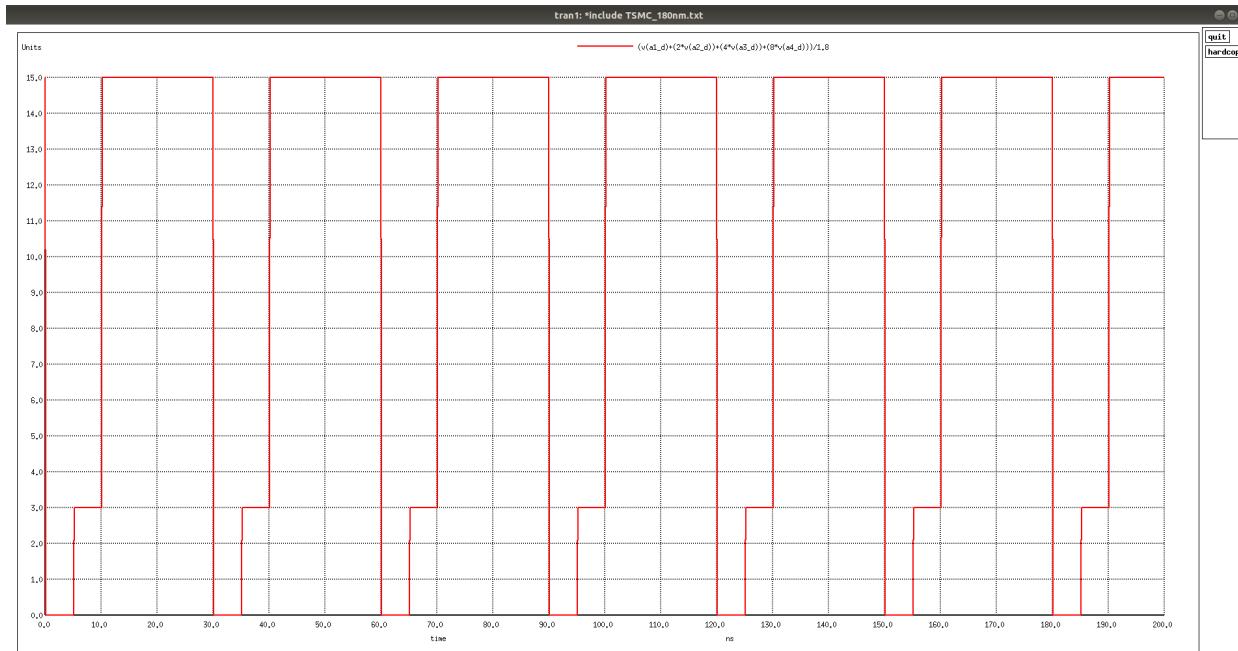
Question 9

Simulations:

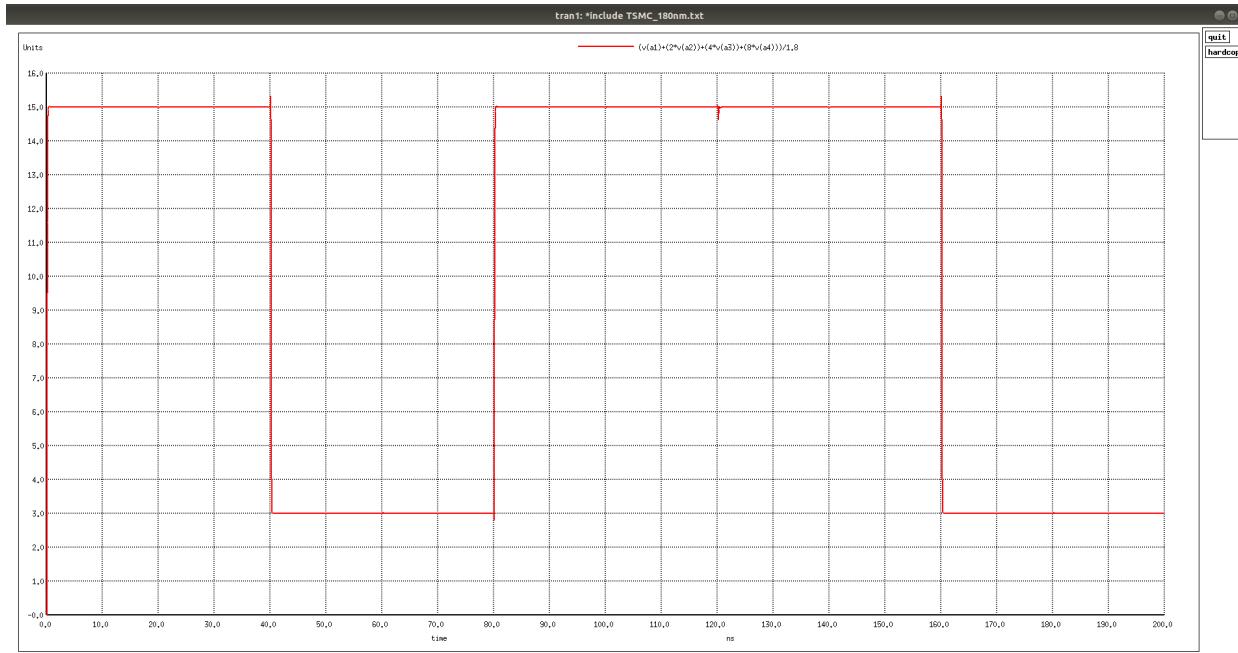
1) Clock:



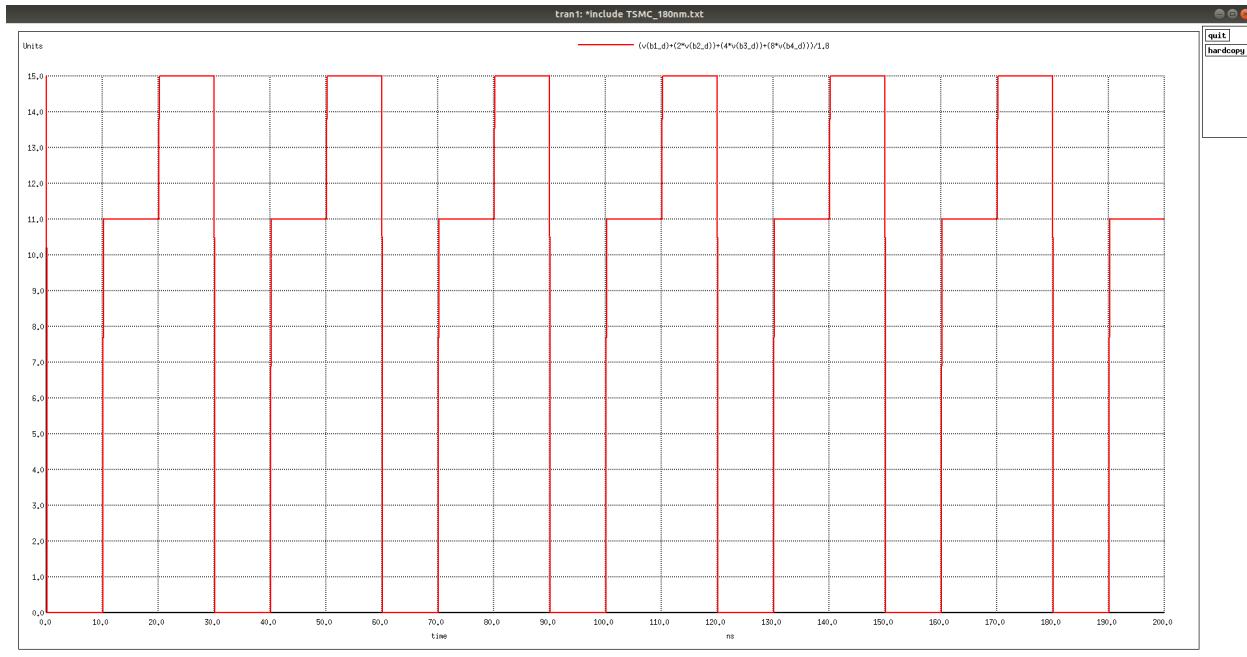
2) Input A to flip flop:



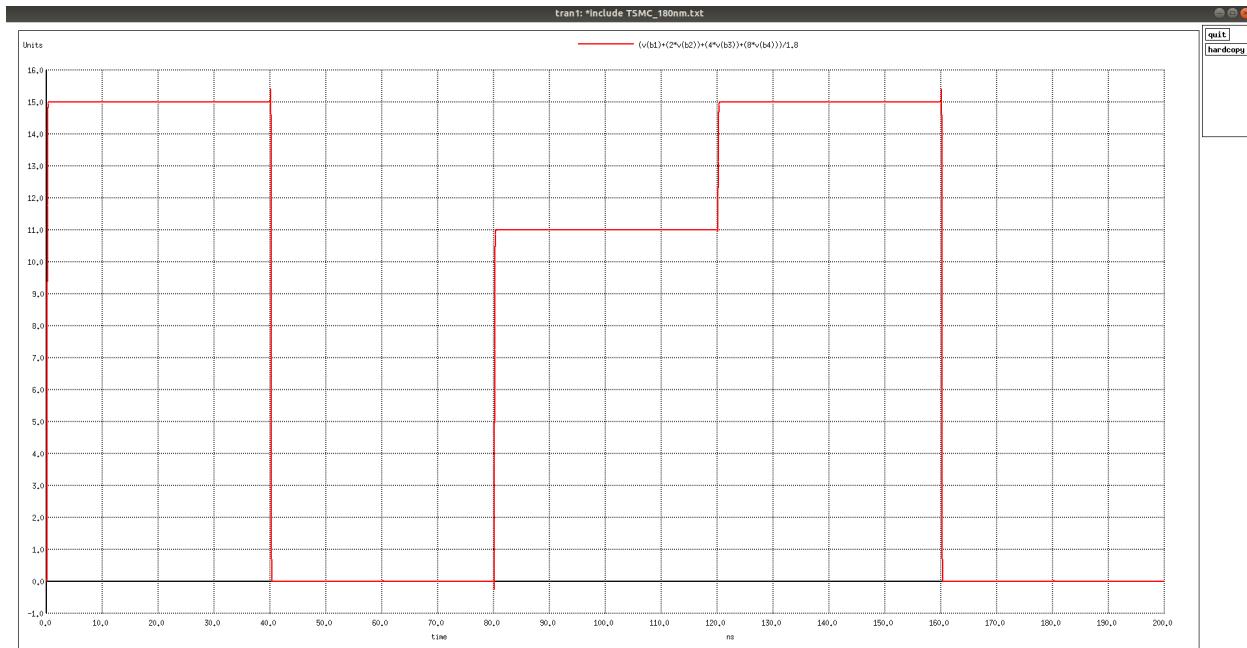
3) Input A After FlipFlop:



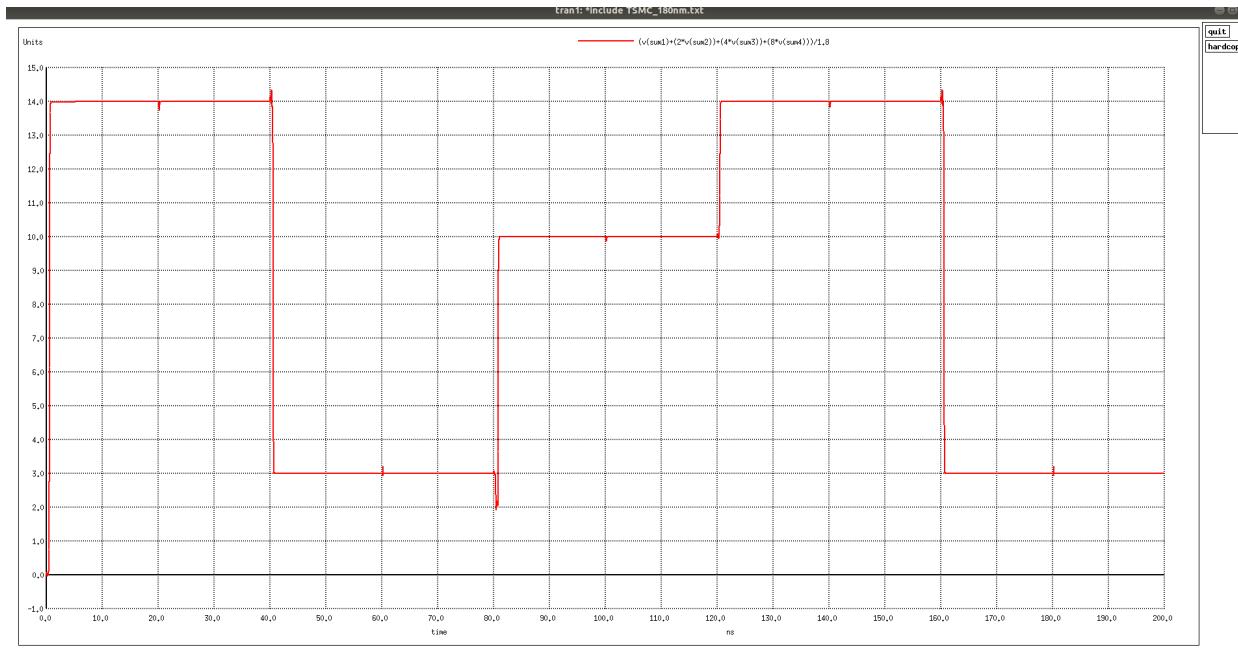
4)Input B to flip flop:



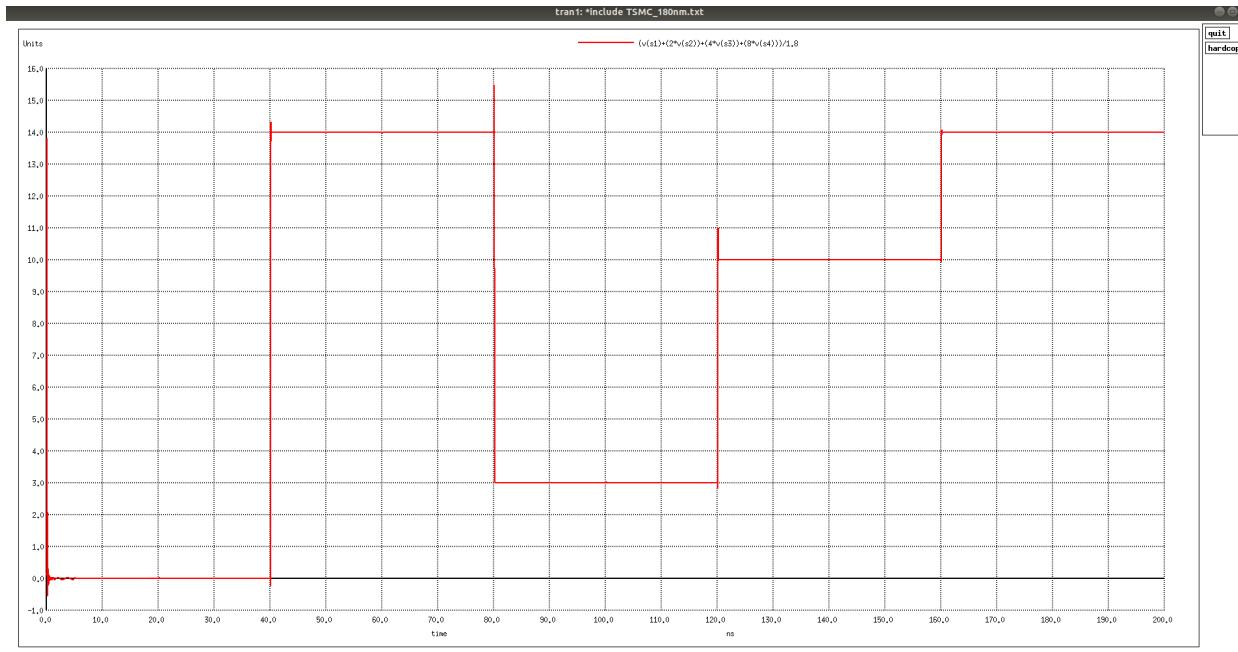
5) Input A After FlipFlop:



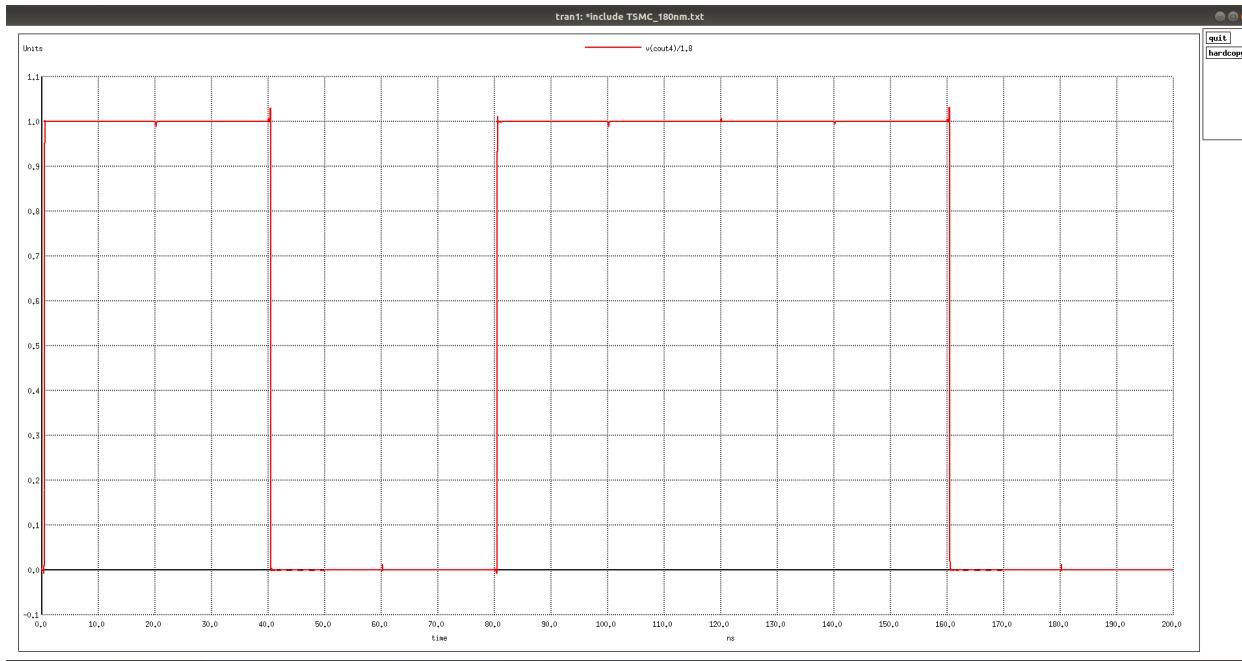
6) Sum from CLA:



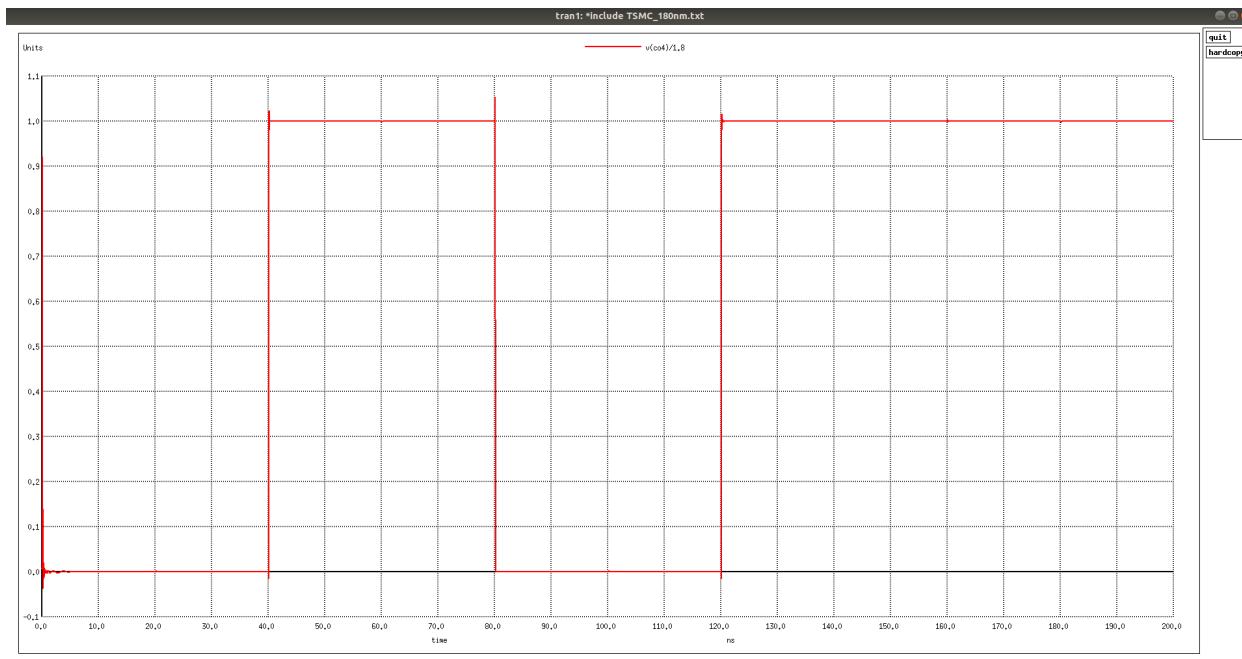
7) Sum after passing through flip-flop:



8) Cout of CLA:



9) Cout after flip flop:



	Pre Layout	Post Layout
CLA Delay(including All 3)	0.59ns	0.87ns

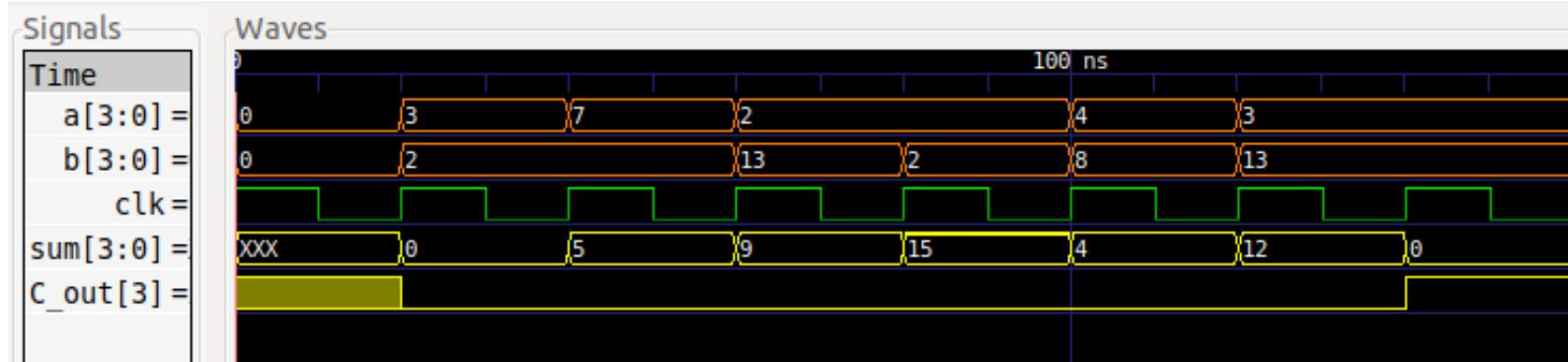
blocks)		
D flip flop(c to q) delay	0.19ns	0.27ns

Question 10

From question 9, we find the maximum delay of the adder to maximise the clock frequency while maintaining reliable inputs.

So, overall we find max delay of CLA by giving input such that the first bit input is the sole reason for the change in the last bit from 0 to 1. By doing so, we obtained the maximum delay possible as approximately **0.87ns**. Then we add the setup time, hold time and D(c to q) delay i.e gives a total delay of **1.31ns**. Therefore, the **maximum possible frequency of flip flop is equal to 0.763358GHz** maximum possible frequency of flip flop is equal to **0.763358GHz**.

Question 11



Clock Cycle	A	B	Sum	Output Carry
0	0	0	X	X
1	3	2	0	0

2	7	2	5	0
3	2	13	9	0
4	2	2	15	0
5	4	8	4	0
6	3	13	12	0
7	3	13	0	1

Testbench:

```
VLSI-Project > Verilog Implementation > Carry Look-Ahead Adder > tb_CLA.v
 1  `timescale 1ns/1ps
 2  module tb;
 3    reg [3:0] a;
 4    reg [3:0] b;
 5    wire [3:0] sum;
 6    wire [3:0] C_out;
 7    reg clk;
 8    wire [3:0] Q_a;
 9    wire [3:0] Q_b;
10
11  FF CLA_timed(.a(a),
12                .b(b),
13                .clk(clk),
14                .C_out(C_out),
15                .sum(sum));
16
17 initial clk = 1;
18 initial begin
19   $monitor("\ta=%d,b=%d,sum=%d,cout=%d",a,b,sum,C_out[3]);
20   $dumpfile("DUMP_CLA.vcd");
21   $dumpvars(0,tb);
22   a = 0;
23   b = 0;
24   #20 a = 4'd3; b = 4'd2;
25   #20 a = 4'd7; b = 4'd2;
26   #20 a = 4'd2; b = 4'd13;
27   #20 a = 4'd2; b = 4'd2;
28   #20 a = 4'd4; b = 4'd8;
29   #20 a = 4'd3; b = 4'd13;
30   #20;
31   #20;
32   $finish;
33 end
34
35   always #10 clk = ~clk;
36 endmodule
```

Wrapper Module:

```
VLSI-Project > Verilog Implementation > Carry Look-Ahead Adder > Wrapper.v
 1  `timescale 1ns/1ps
 2  module FF (
 3    input [3:0] a,
 4    input [3:0] b,
 5    input clk,
 6    output reg [3:0] C_out,
 7    output reg [3:0] sum
 8
 9  );
10   reg Cin;
11   wire [3:0] Q_C_out;
12   wire [3:0] Q_sum;
13   reg [3:0] Q_a;
14   reg [3:0] Q_b;
15   initial Cin = 0;
16
17   //Flip-flop for A
18   always @(posedge clk) begin
19     |   Q_a = a;
20   end
21   //Flip-flop for B
22   always @(posedge clk) begin
23     |   Q_b = b;
24   end
25   //4-bit carry lookahead block module
26   CLA cl(.a(Q_a),
27           .b(Q_b),
28           .Cin(Cin),
29           .sum(Q_sum),
30           .C_out(Q_C_out));
31
32
33   //Flip-flop for Carry-Out
34   always @(posedge clk) begin
35     |   C_out = Q_C_out;
36   end
37   //Flip-flop for Sum-Bits
38   always @(posedge clk) begin
39     |   sum = Q_sum;
40   end
41 endmodule
```

Carry Look Ahead Adder Module:

```
VLSI-Project > Verilog Implementation > Carry Look-Ahead Adder > CLA.v
1 `timescale 1ns/1ps
2 module CLA (
3     input [3:0] a,
4     input [3:0] b,
5     input Cin,
6     output reg [3:0] sum,
7     output reg [3:0] C_out);
8
9     reg [3:0] p;
10    reg [3:0] g;
11    always @(*) begin
12        //Propagate pi = ai xor bi
13        p[0] = a[0] ^ b[0];
14        p[1] = a[1] ^ b[1];
15        p[2] = a[2] ^ b[2];
16        p[3] = a[3] ^ b[3];
17        //Generate gi = ai and bi
18        g[0] = a[0] & b[0];
19        g[1] = a[1] & b[1];
20        g[2] = a[2] & b[2];
21        g[3] = a[3] & b[3];
22        //Carry-Out c(i+1) = (pi and ci) + gi
23        C_out[0] = (p[0] & Cin) | g[0];
24        C_out[1] = (p[1] & C_out[0]) | g[1];
25        C_out[2] = (p[2] & C_out[1]) | g[2];
26        C_out[3] = (p[3] & C_out[2]) | g[3];
27        //Sum sum_i = pi xor ci
28        sum[0] = p[0] ^ Cin;
29        sum[1] = p[1] ^ C_out[0];
30        sum[2] = p[2] ^ C_out[1];
31        sum[3] = p[3] ^ C_out[2];
32    end
33 endmodule
```