

# *MINI PROJECT*

## **BIT1\_BLOCK**

Team Members:

1. Leelavati B
2. Shrivatsa P
3. Kanchana M

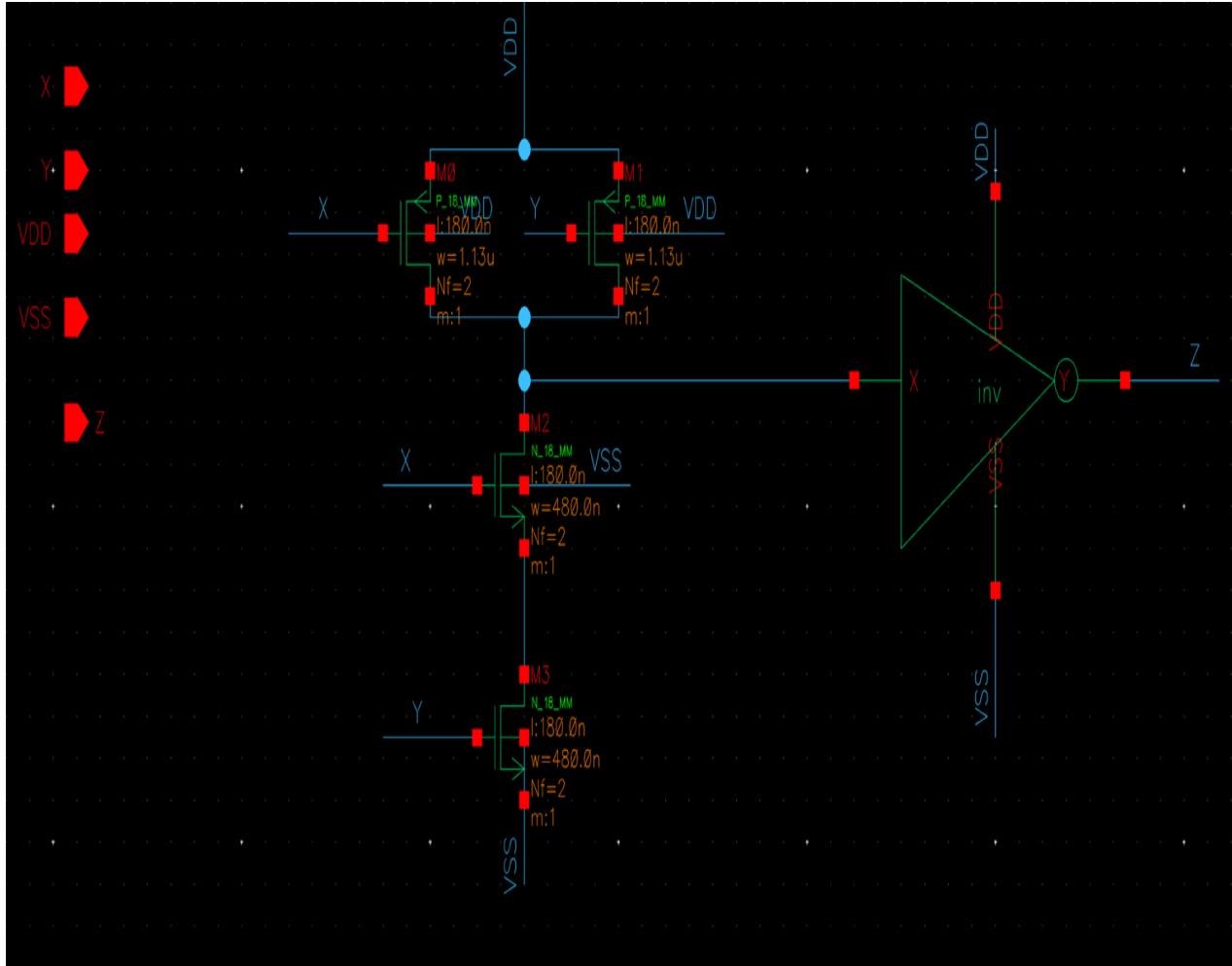
# INTRODUCTION

The Bit1 block is a basic digital circuit that counts the number of ones present in its input bits and gives the binary equivalent of that count at the output.

## **Subblocks are:**

1. AND Gate
2. AND\_3 Gate
3. OR Gate
4. XOR Gate
5. FULL ADDER
6. 3BIT\_ONES

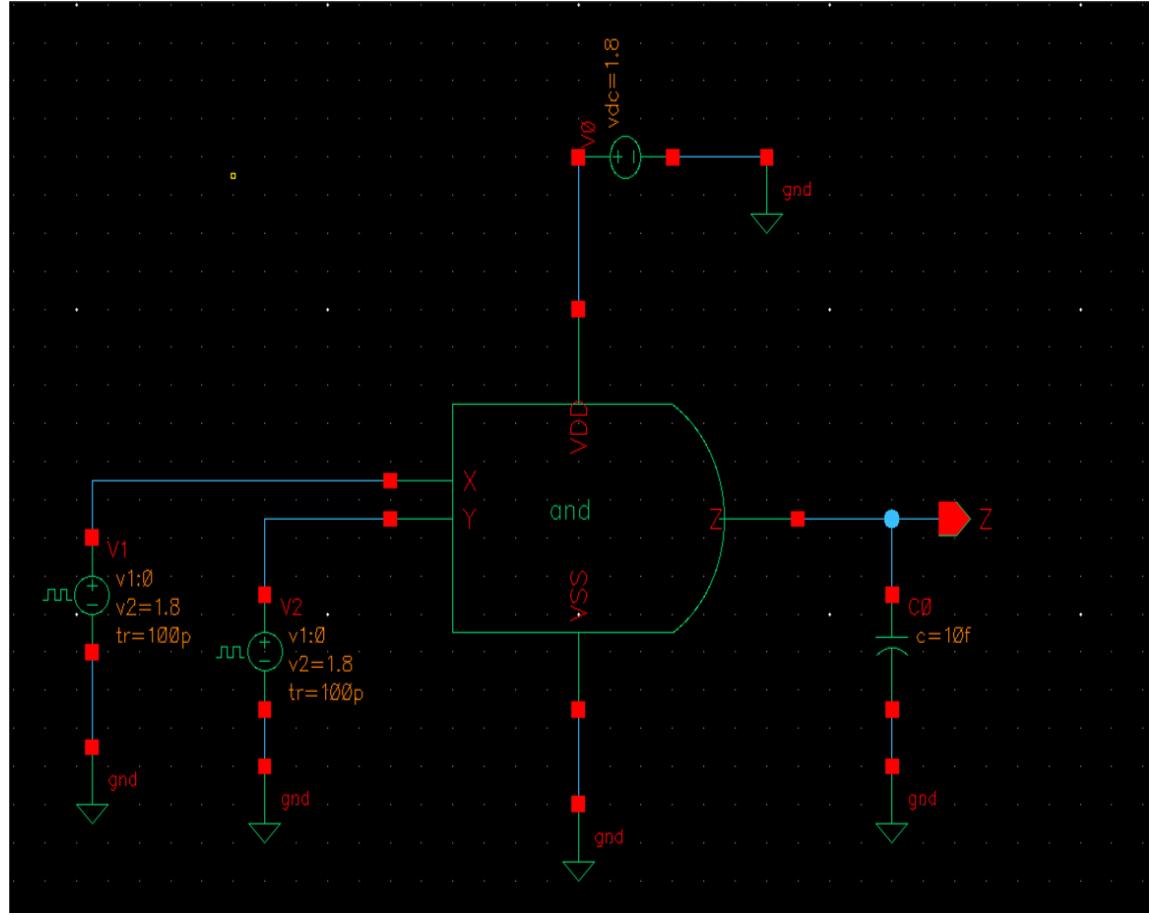
## 1.2\_INPUT\_AND\_GATE\_SCHEMATIC



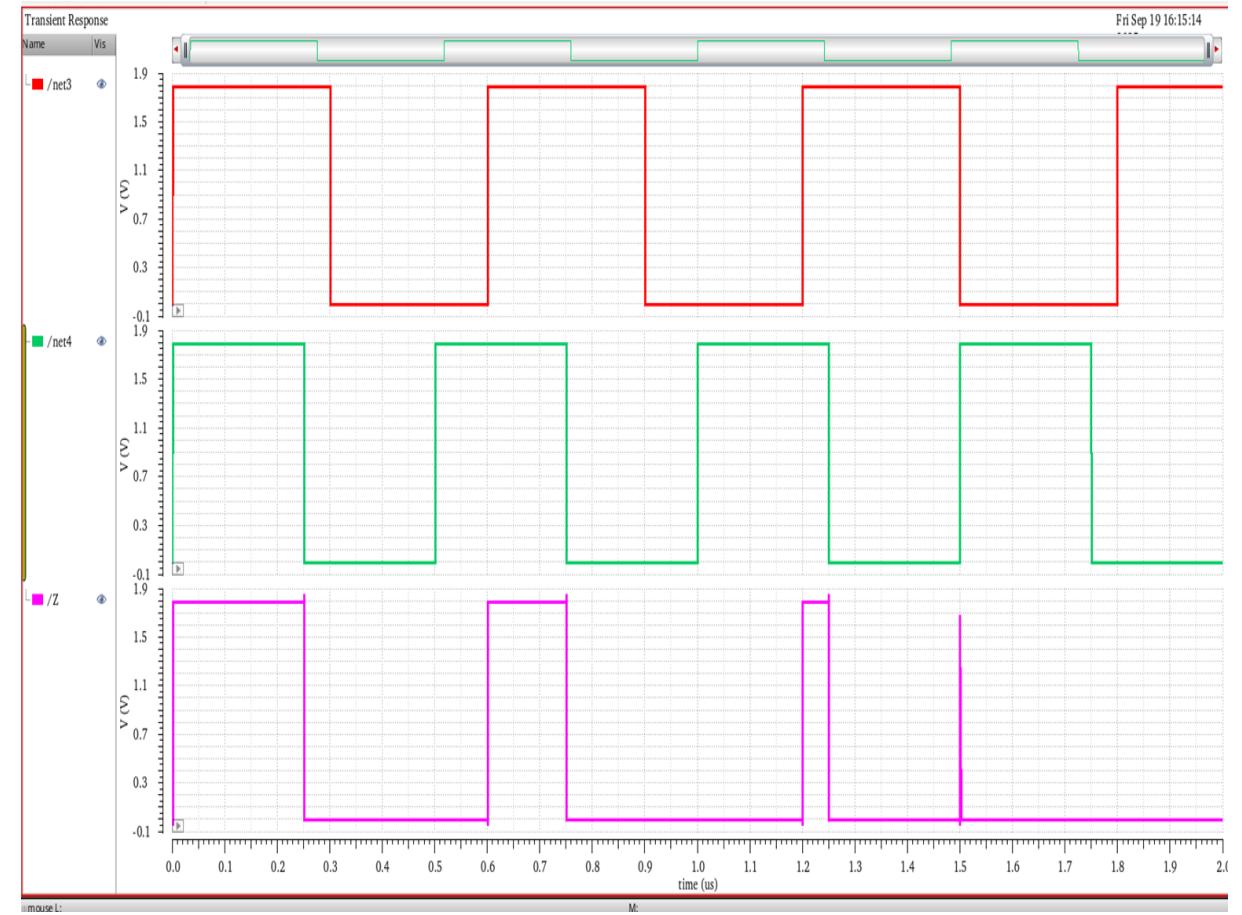
## TRUTH TABLE

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

# TESTBENCH

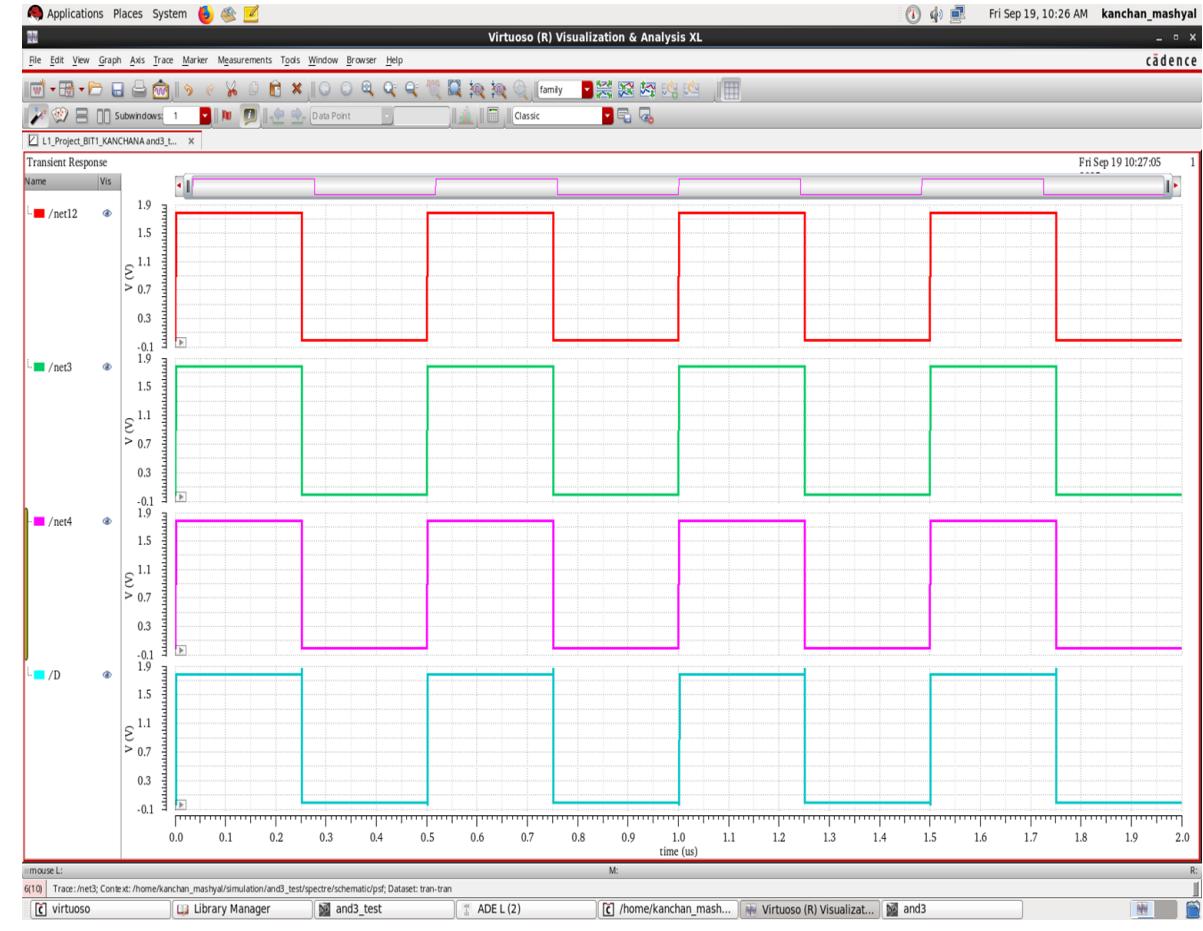
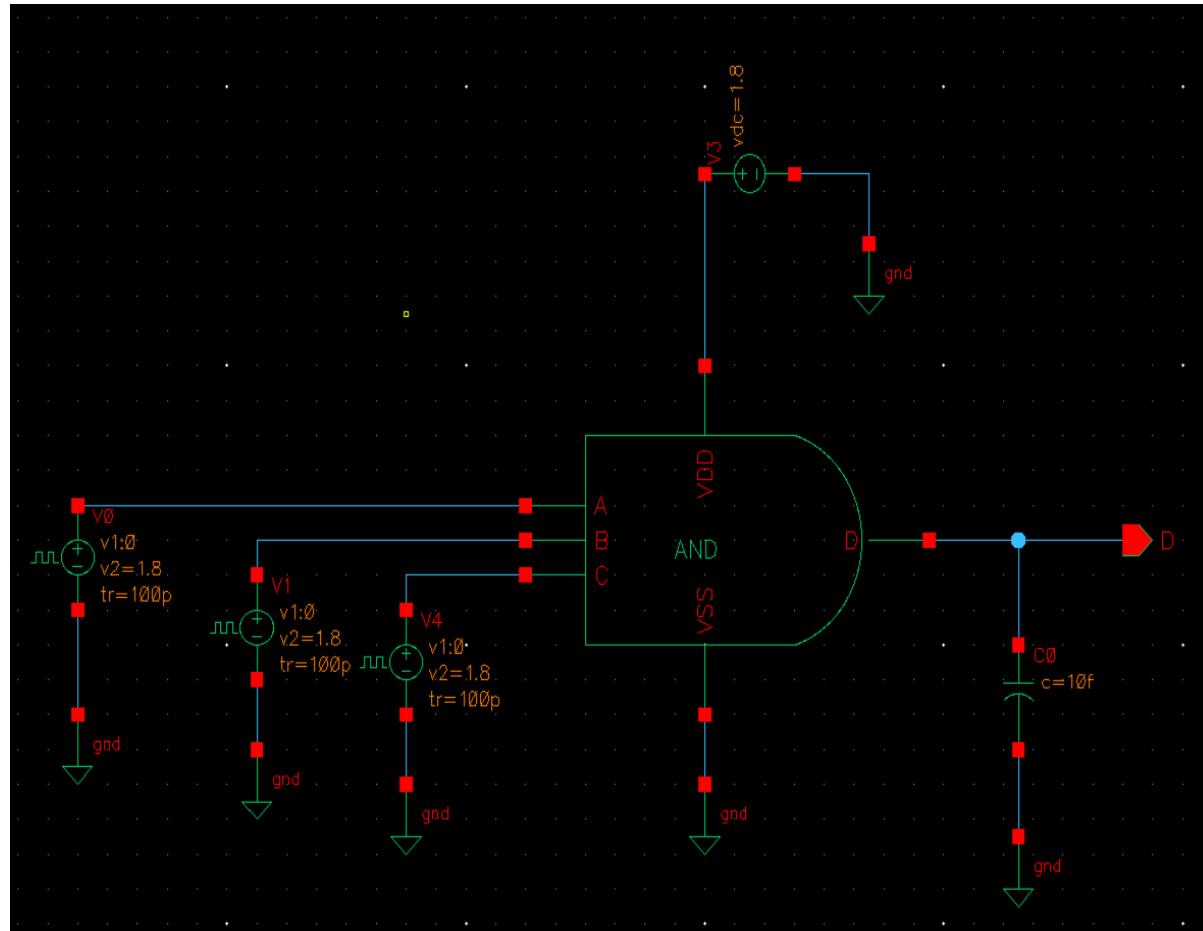


# WAVEFORM

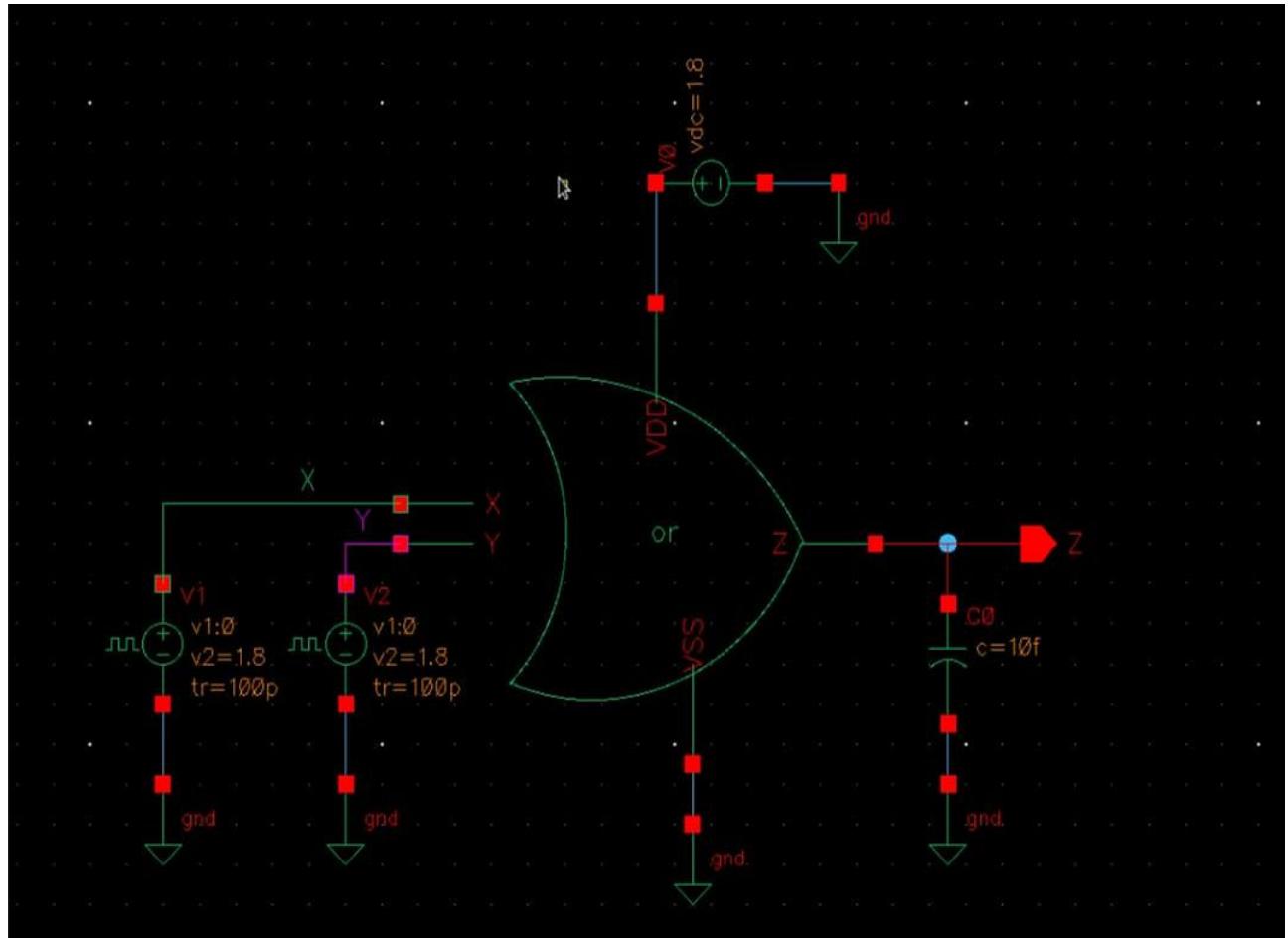


## 2. 3\_INPUT\_AND\_GATE\_TESTBENCH

## WAVEFORM



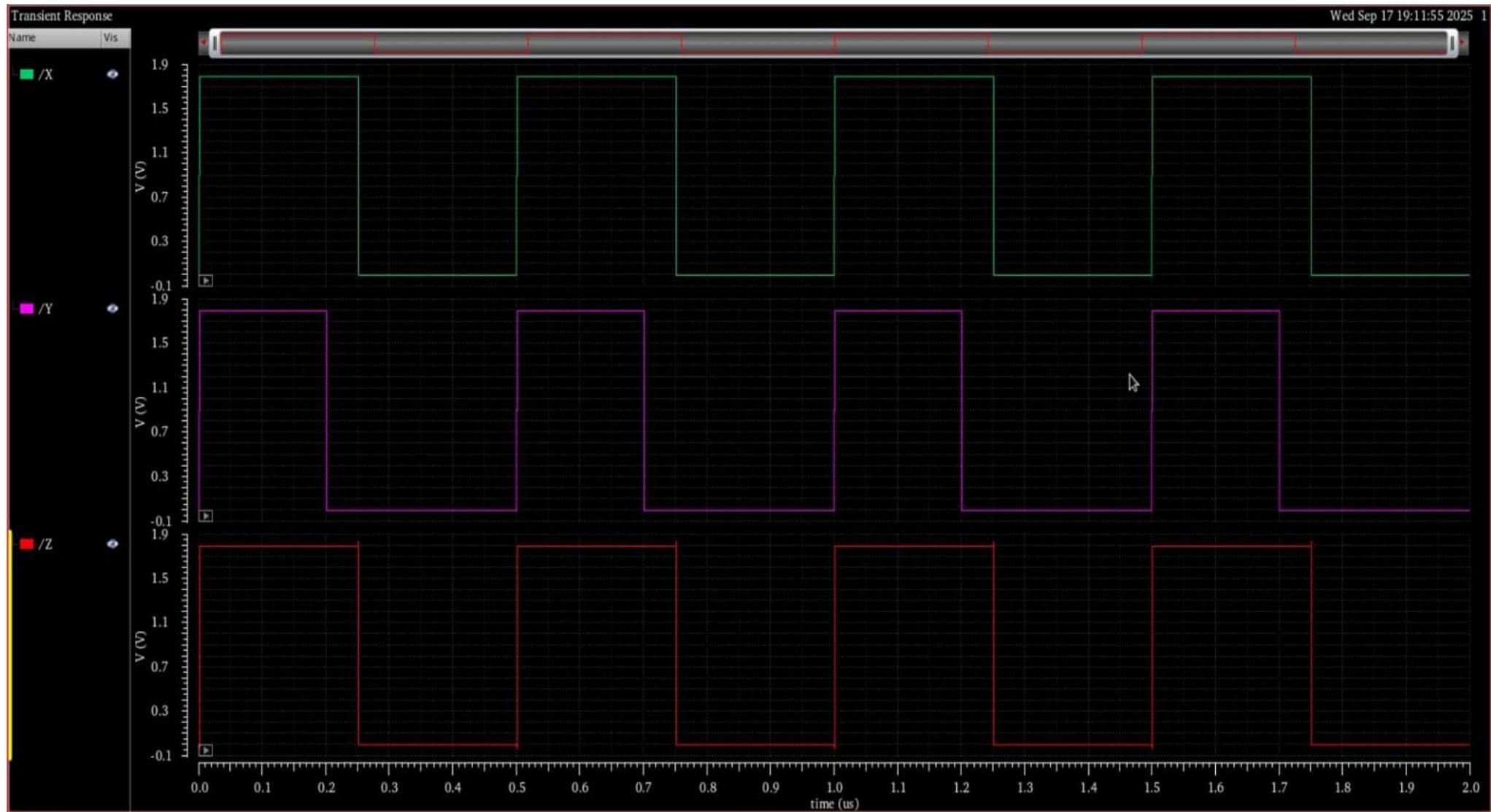
### 3. OR\_GATE\_TESTBENCH



### TRUTH TABLE

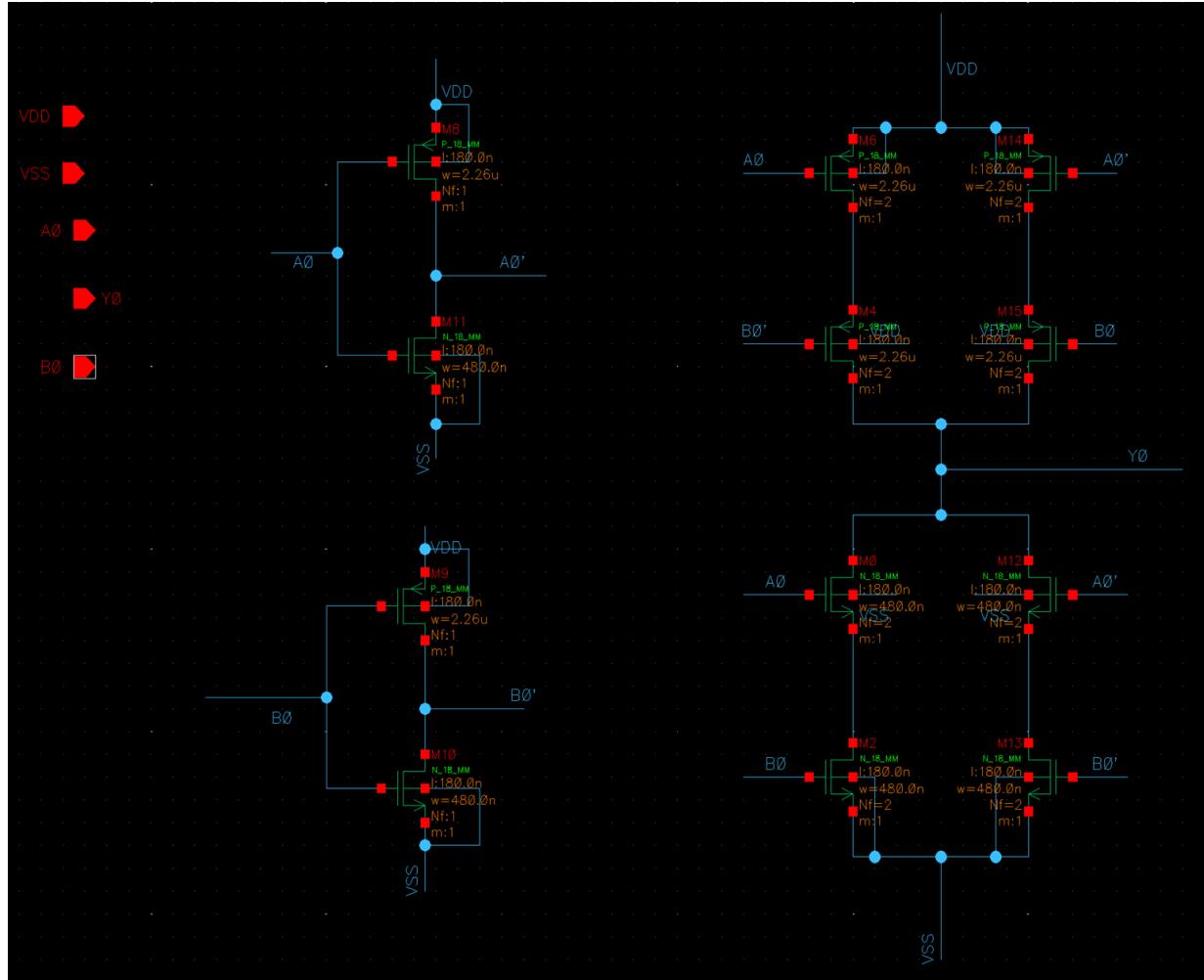
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

# WAVEFORM



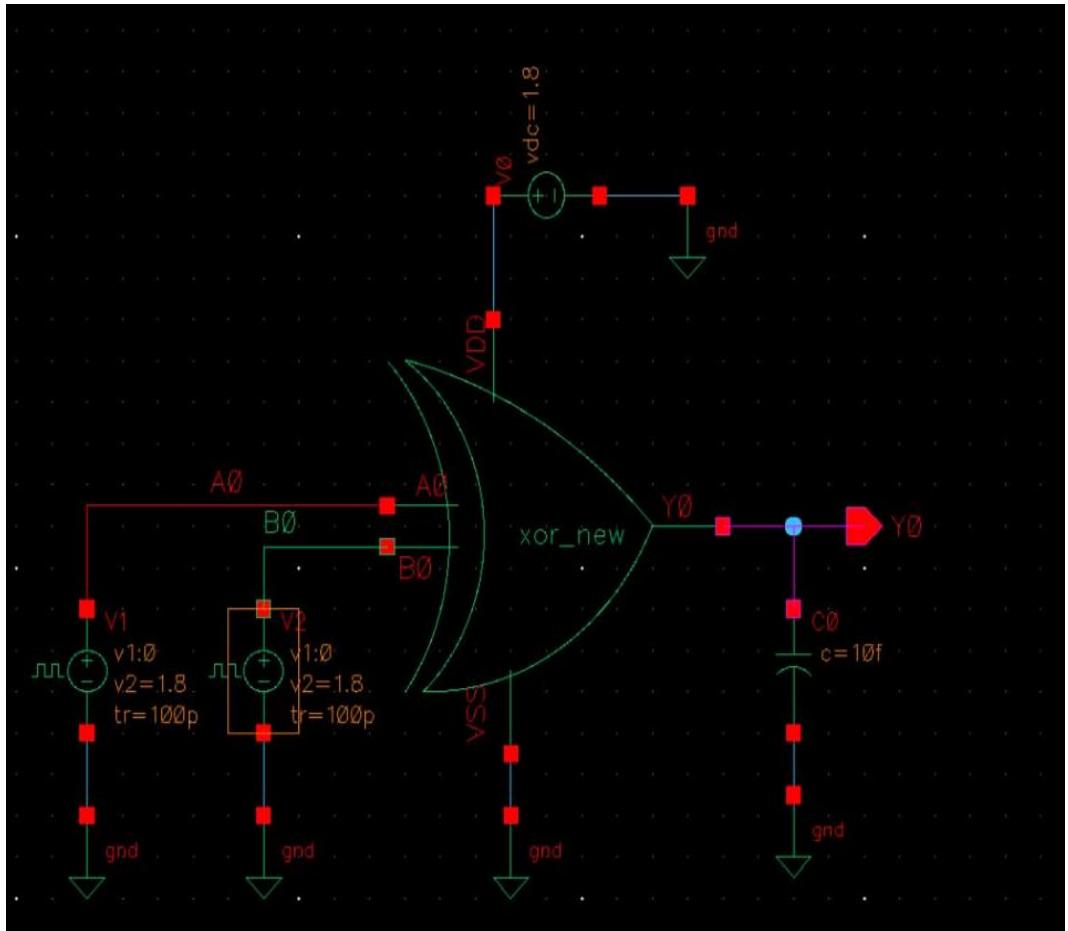
## 4. XOR\_SCHEMATIC

## TRUTH TABLE

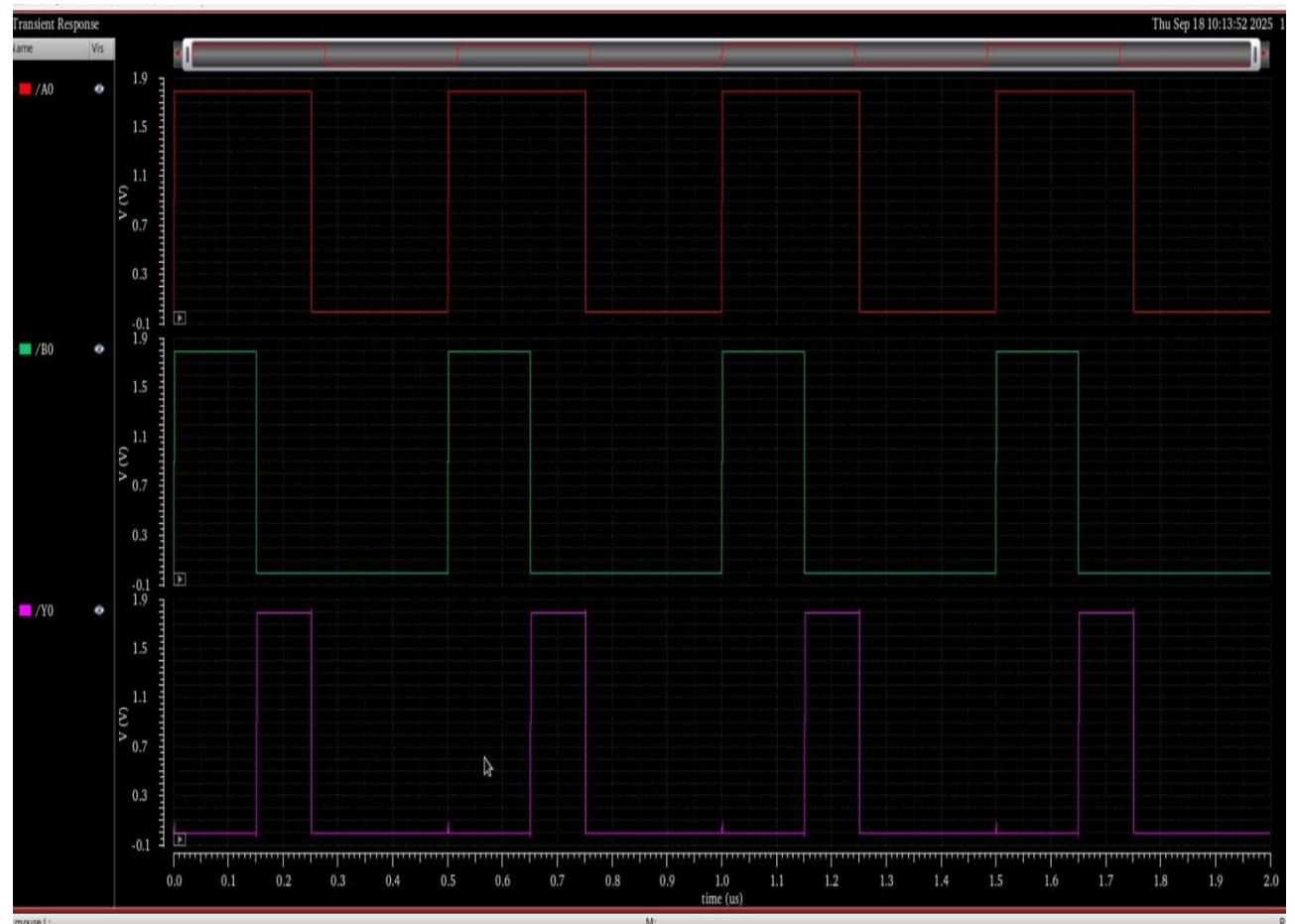


X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

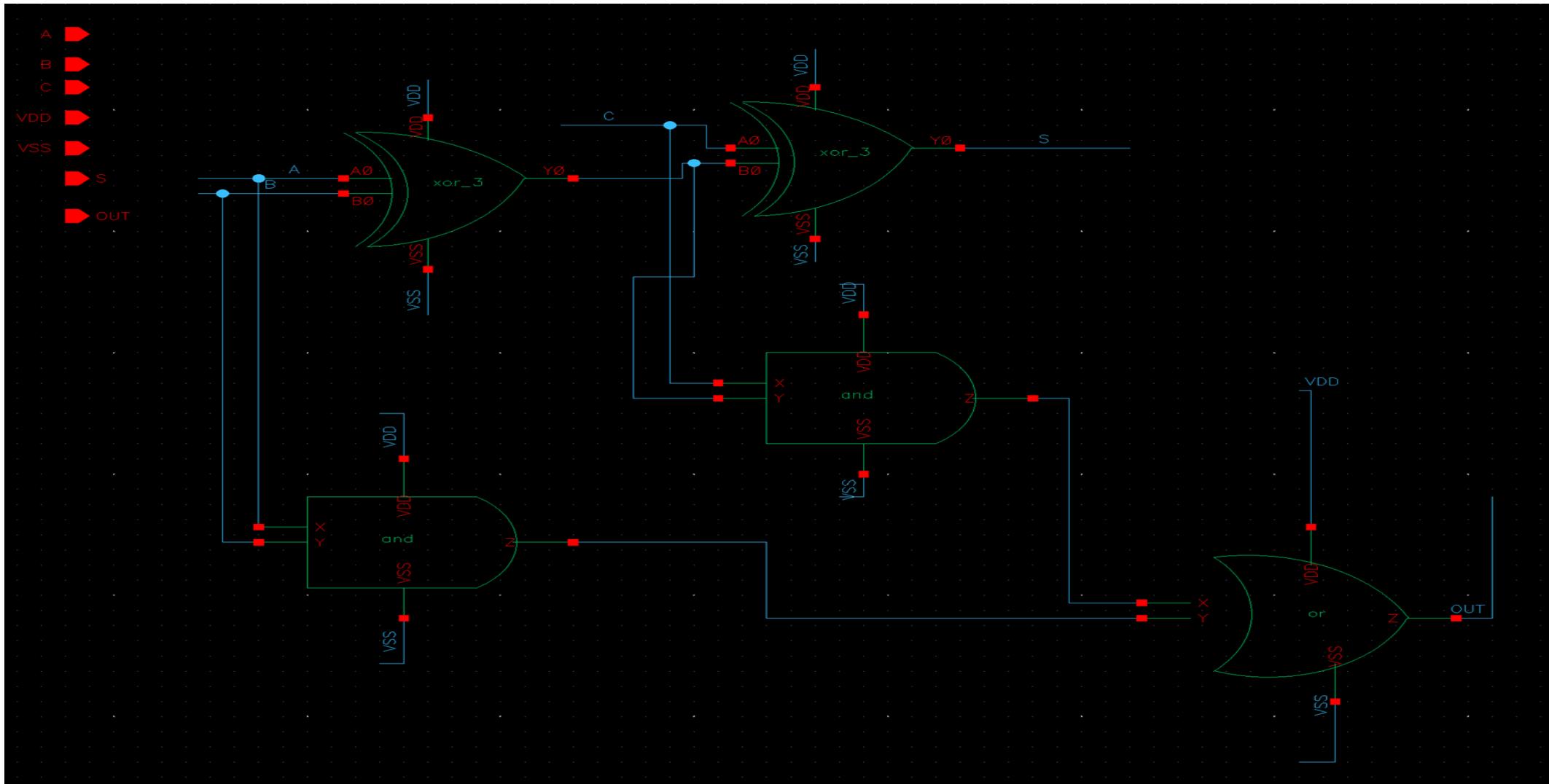
# TESTBENCH



# WAVEFORM



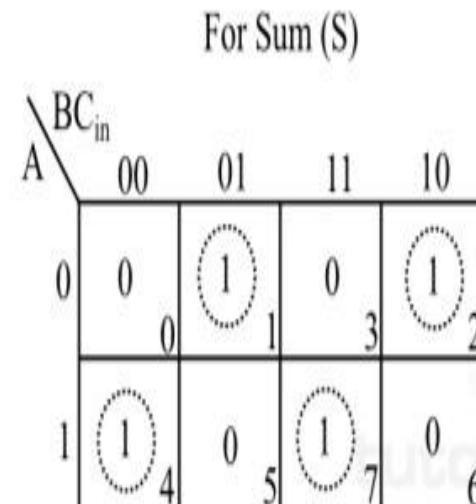
## 5. FULL\_ADDER\_SCHEMATIC



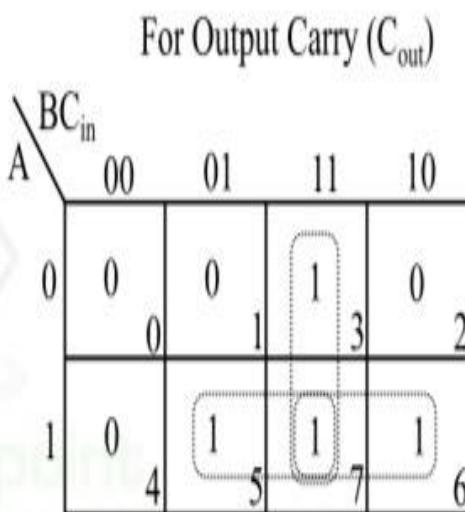
## TRUTH TABLE

Inputs			Outputs	
A	B	C <sub>in</sub>	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

## K MAP FOR FULL ADDER

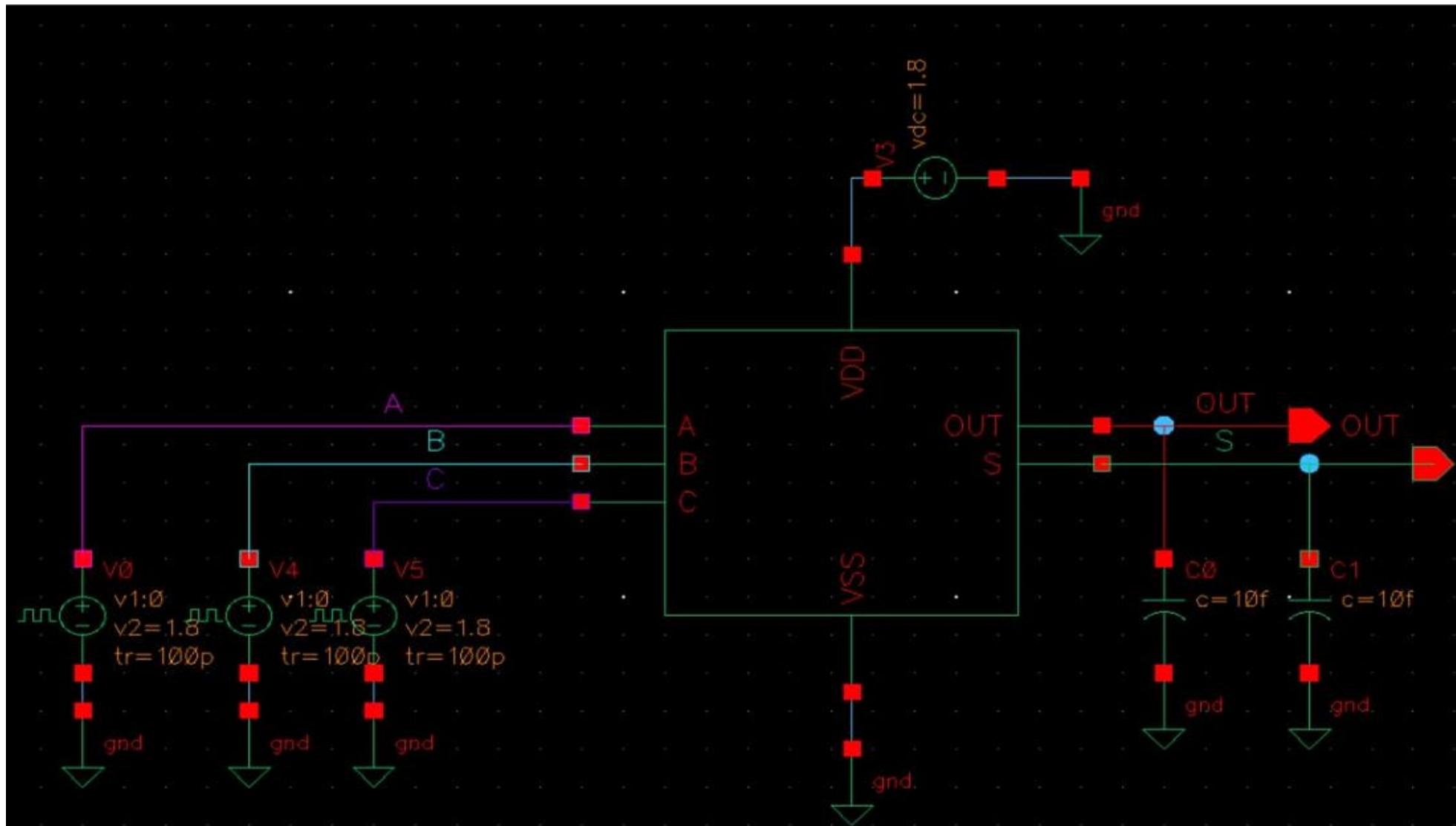


$$S = A'B'C_{in} + A'BC'_{in} + AB'C_{in} + ABC_{in}$$

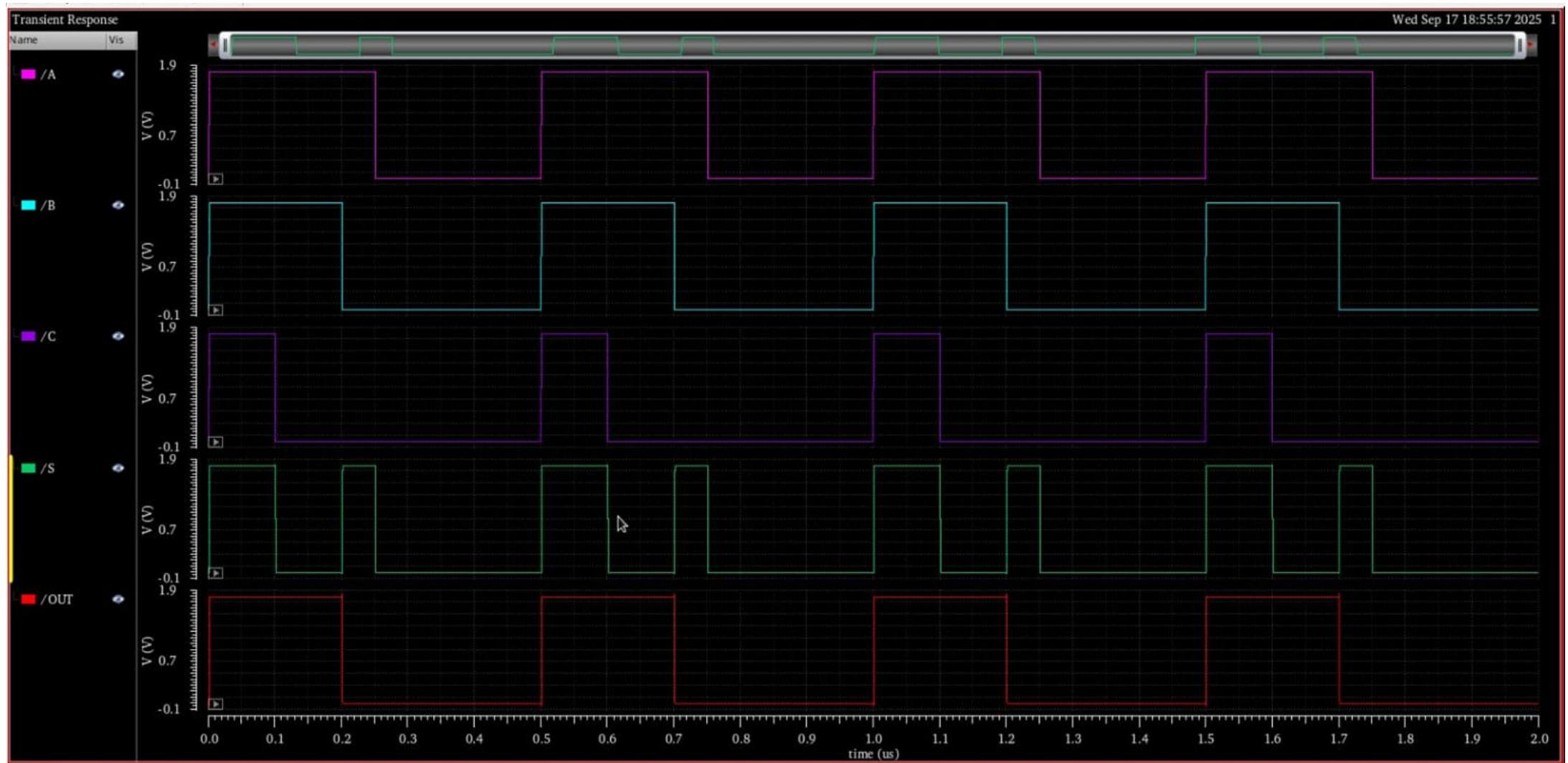


$$C_{out} = AB + AC_{in} + BC_{in}$$

# FULL\_ADDER\_TESTBENCH

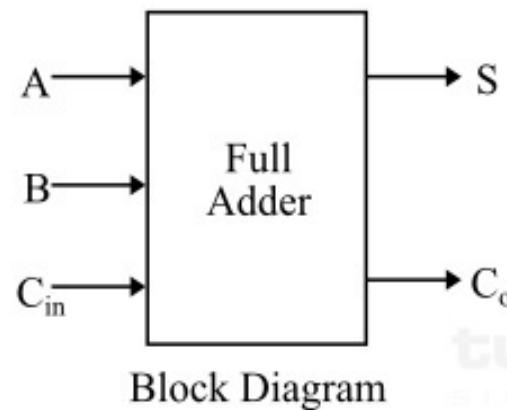


# WAVEFORM



## 6. 3\_BIT\_ONES

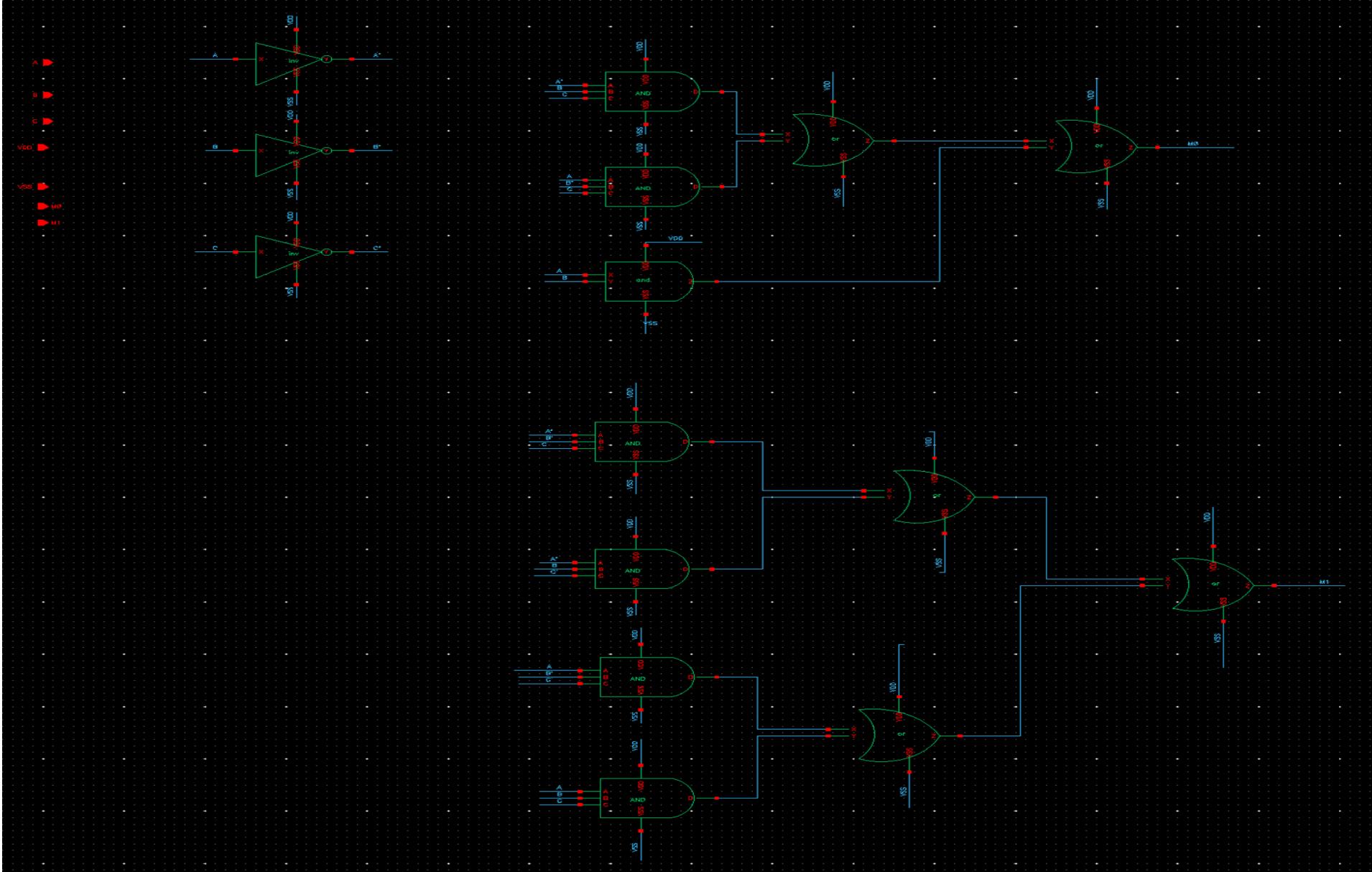
The 3-bit ONES circuit acts like a Full Adder, because both perform the same operation: adding three 1-bit values.



## TRUTH TABLE

Inputs			Outputs	
A	B	C <sub>in</sub>	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

# SCHEMATIC



## EXPRESSION

$$M_0 = A'BC + AB'C + AB(C + C') = ABC + ABC'$$

$$M_1 = A'B'C + A'BC' + AB'C' + ABC$$

## K MAP FOR M0

		BC <sub>in</sub>			
		00	01	11	10
A	0	0	0	1	0
	1	0	1	1	1
		4	5	7	6

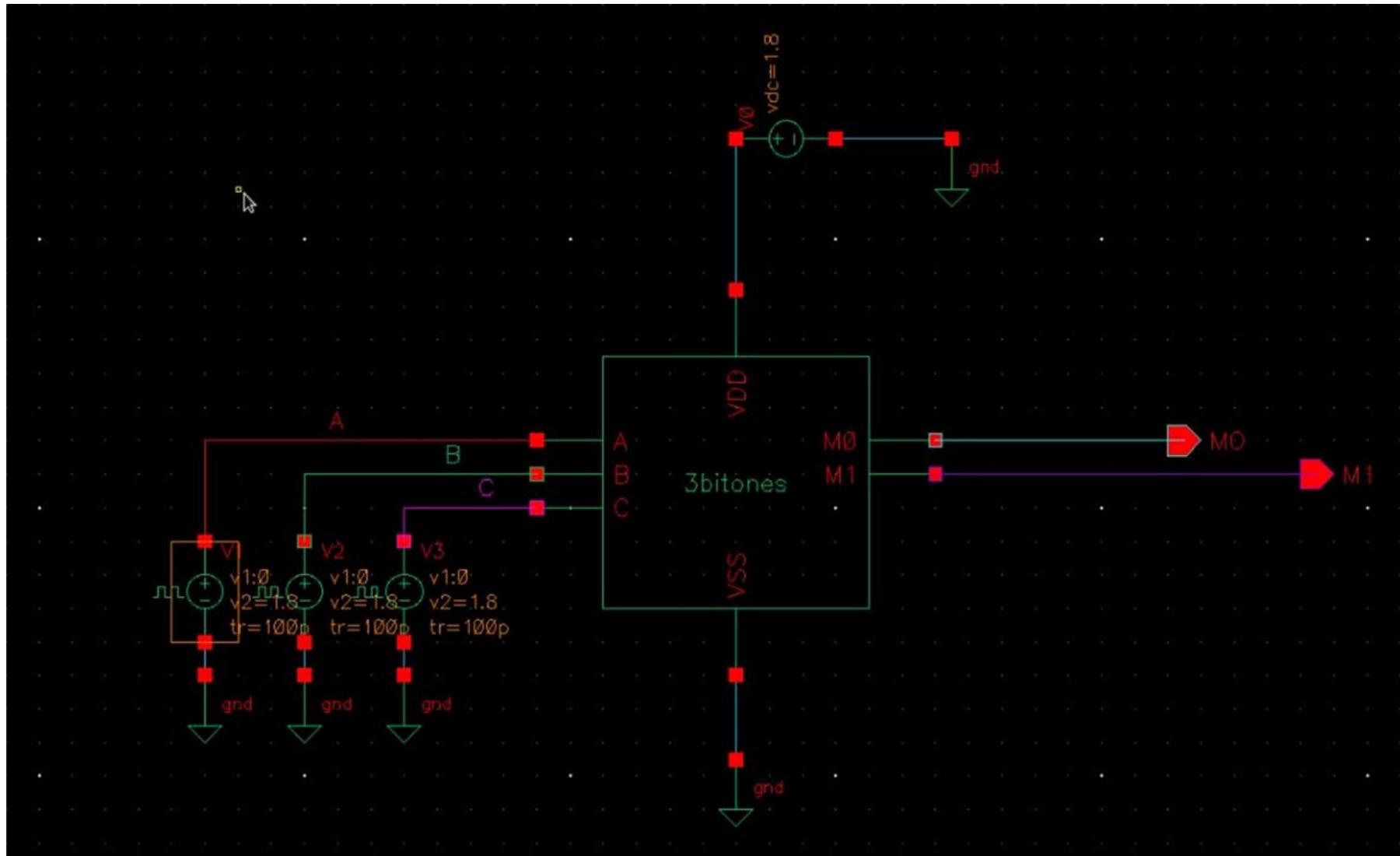
$$C_{out} = AB + AC_{in} + BC_{in}$$

## K MAP FOR M1

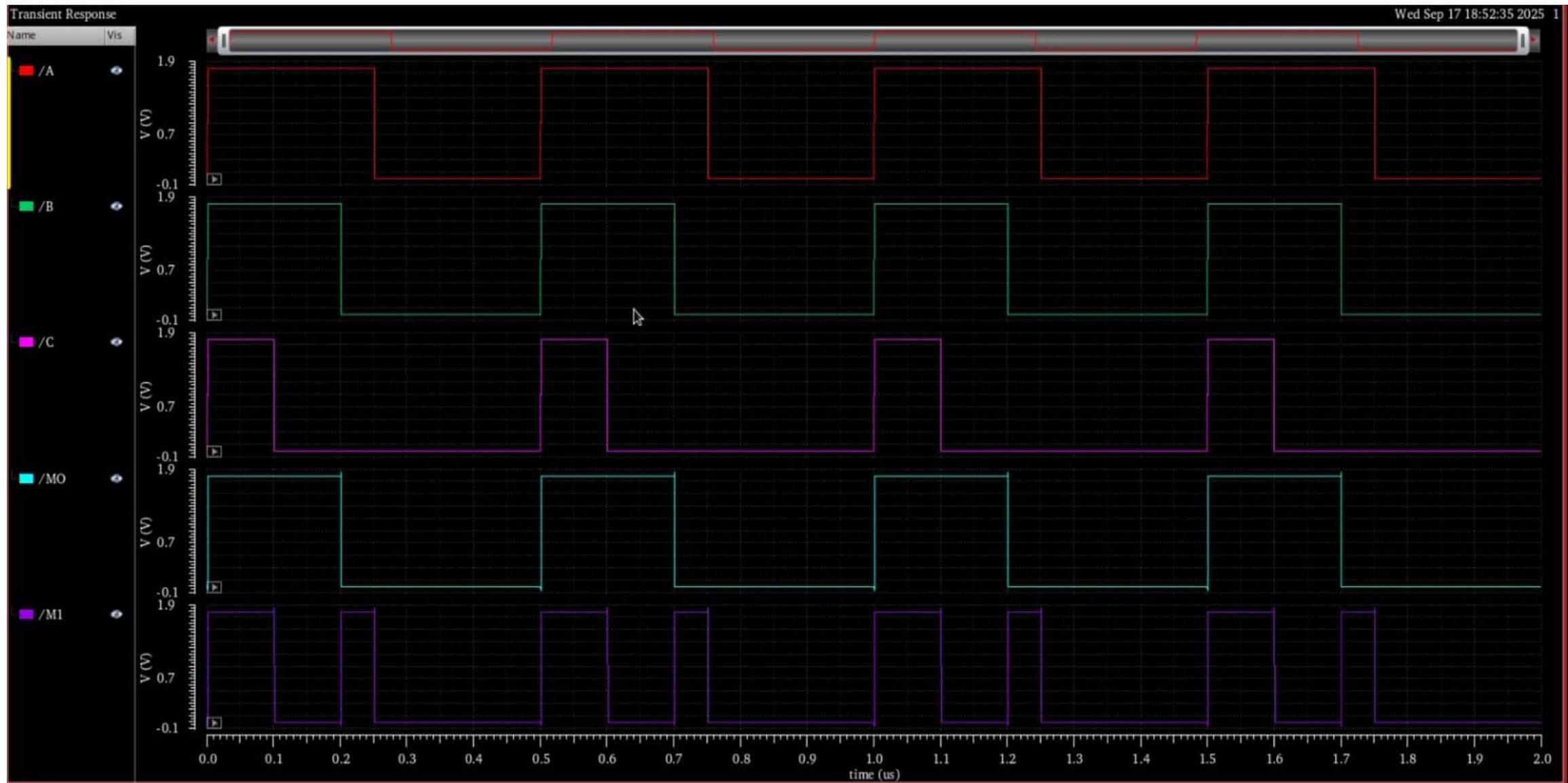
		BC <sub>in</sub>			
		00	01	11	10
A	0	0	1	0	1
	1	1	0	1	0
		4	5	7	6

$$S = A'B'C_{in} + A'BC'_{in} + AB'C'_{in} + ABC_{in}$$

# TESTBENCH



# WAVEFORM



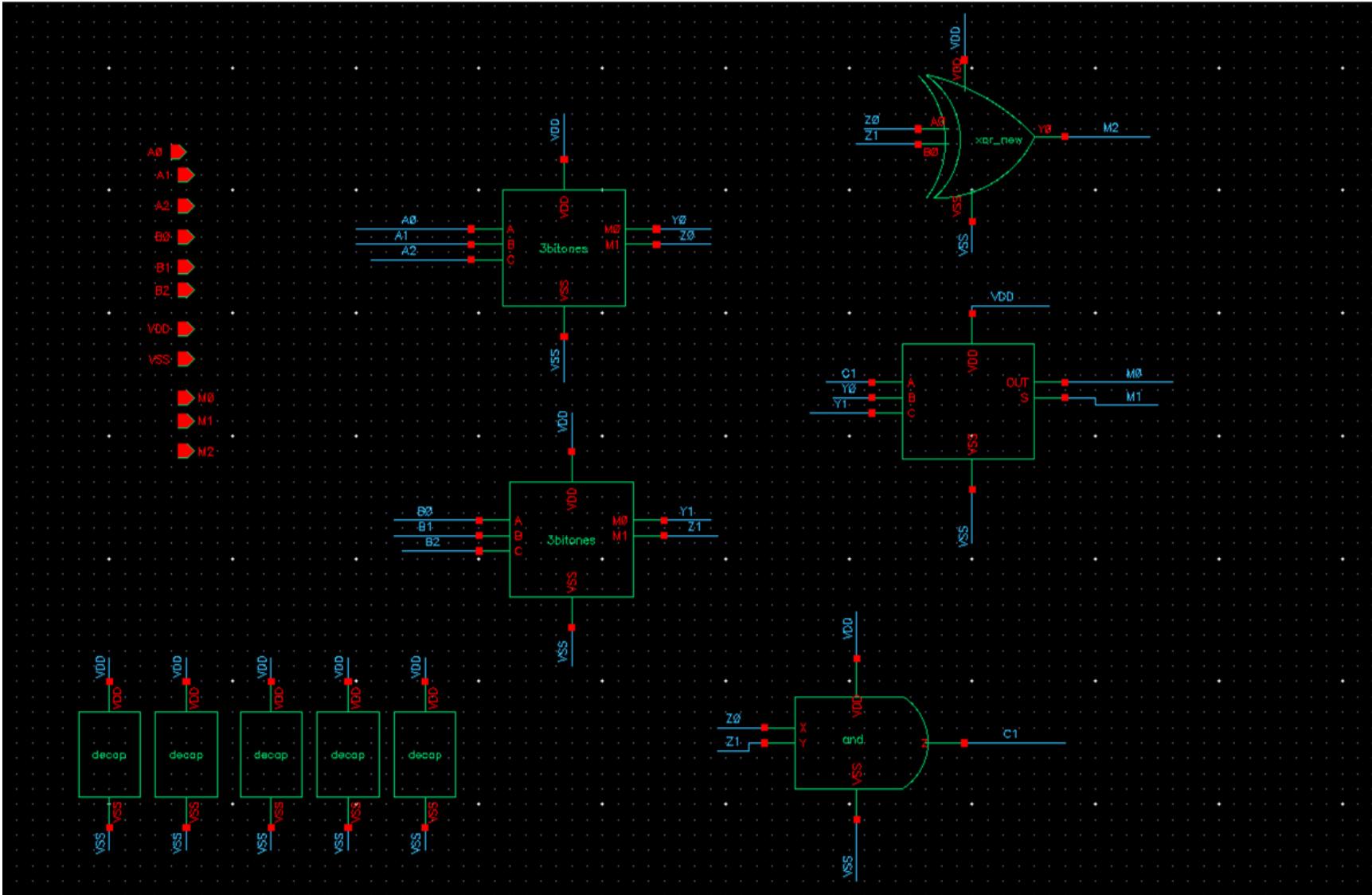
## **BIT1\_BLOCK**

- ❖ Bit-1 block counts how many logic ‘1’s are present in the inputs and gives the binary equivalent at the output.
- ❖ Since it is a 6-input it has  $2^6=64$  possible input combinations, as represented in the truth table

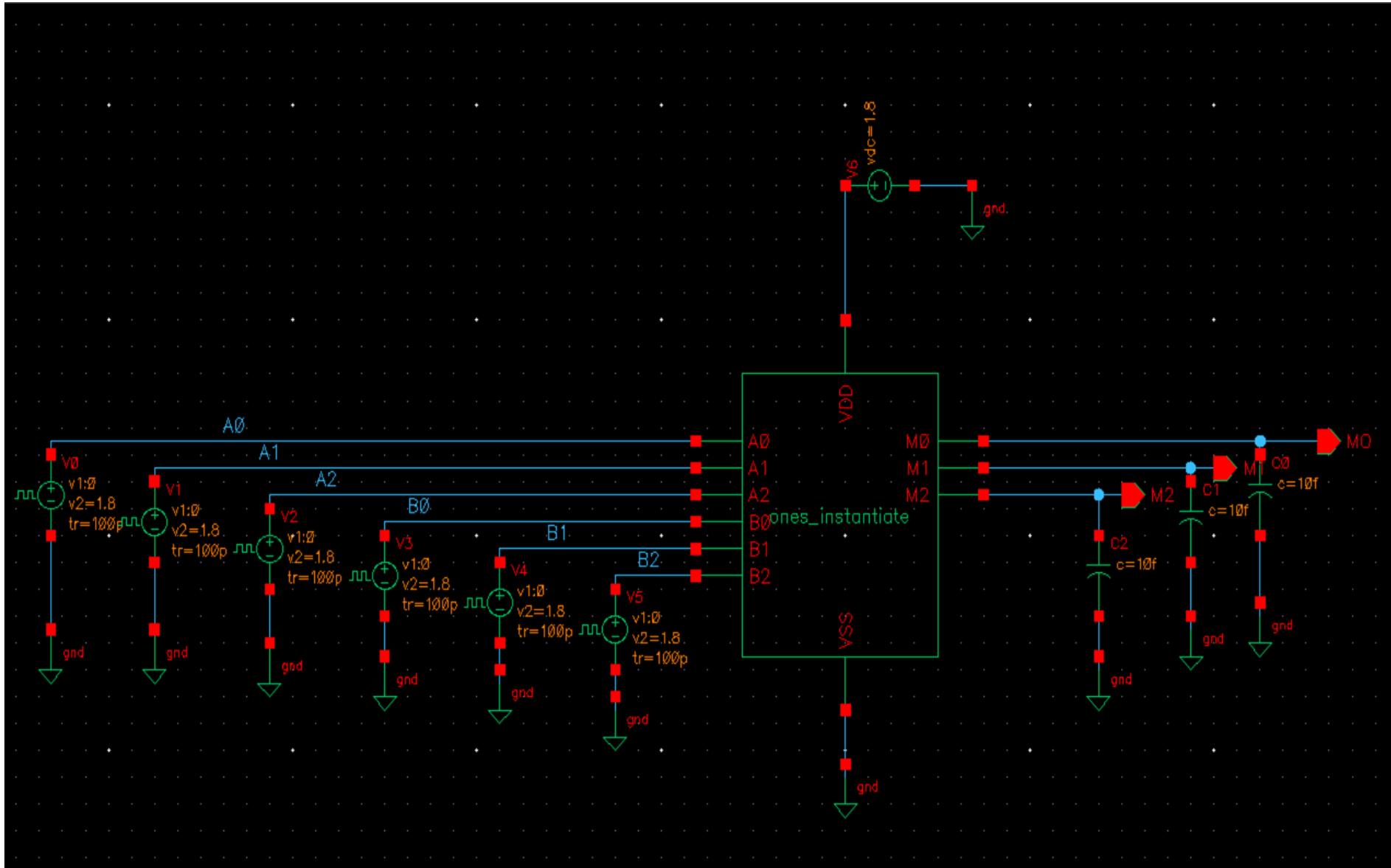
## **TRUTH TABLE**

A0	A1	A2	B0	B1	B2	M0	M1	M2
0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	1
0	0	0	0	1	0	0	0	1
0	0	0	0	1	1	0	1	0
0	0	0	1	0	0	0	0	1
0	0	0	1	0	1	0	1	0
0	0	0	1	1	0	0	1	0
0	0	1	1	1	1	0	1	1
0	0	1	0	0	0	0	0	1

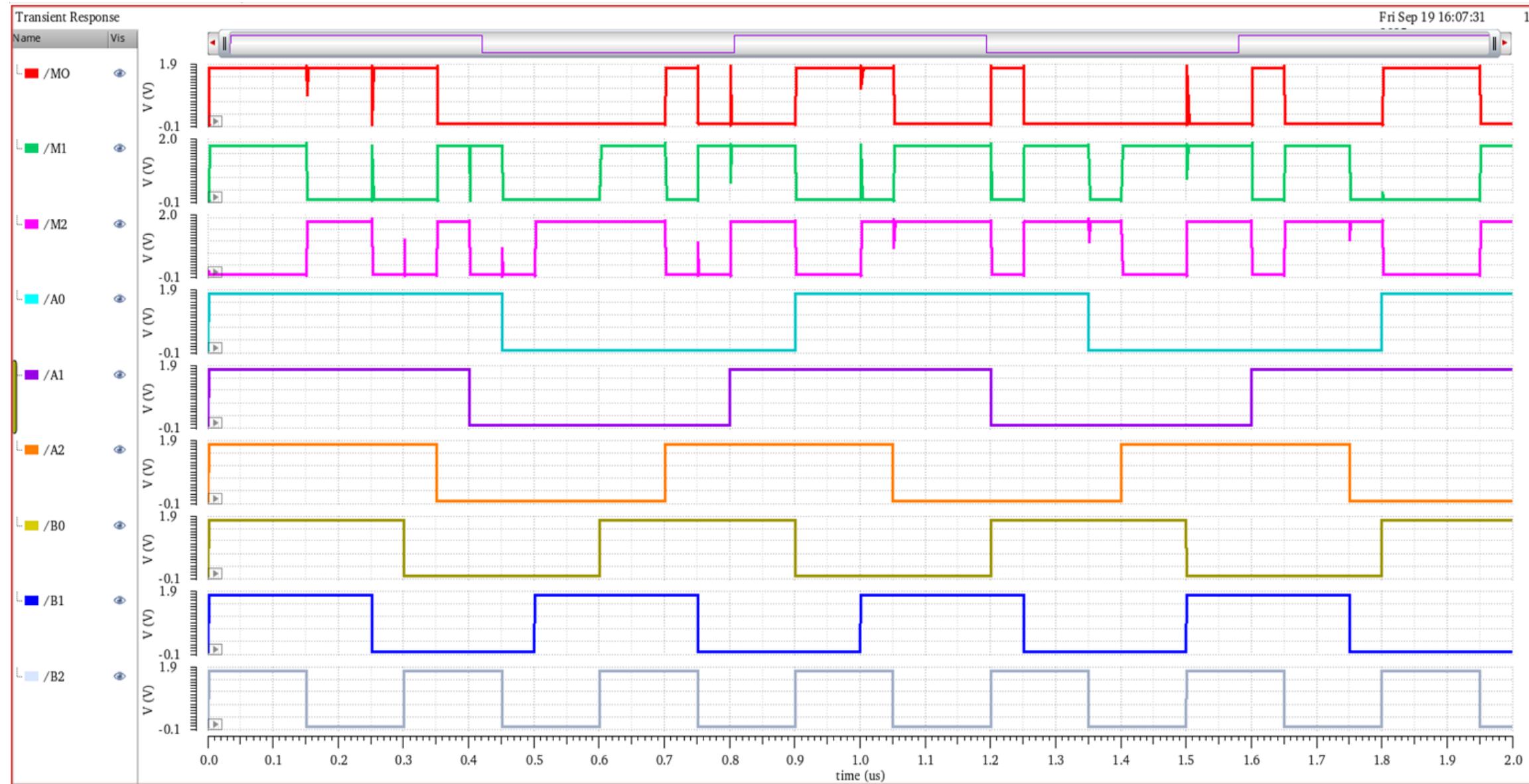
# SCHEMATIC\_OF\_BIT1\_BLOCK



# TESTBENCH



# WAVEFORM



# DELAY CALCULATION AND CURRENT

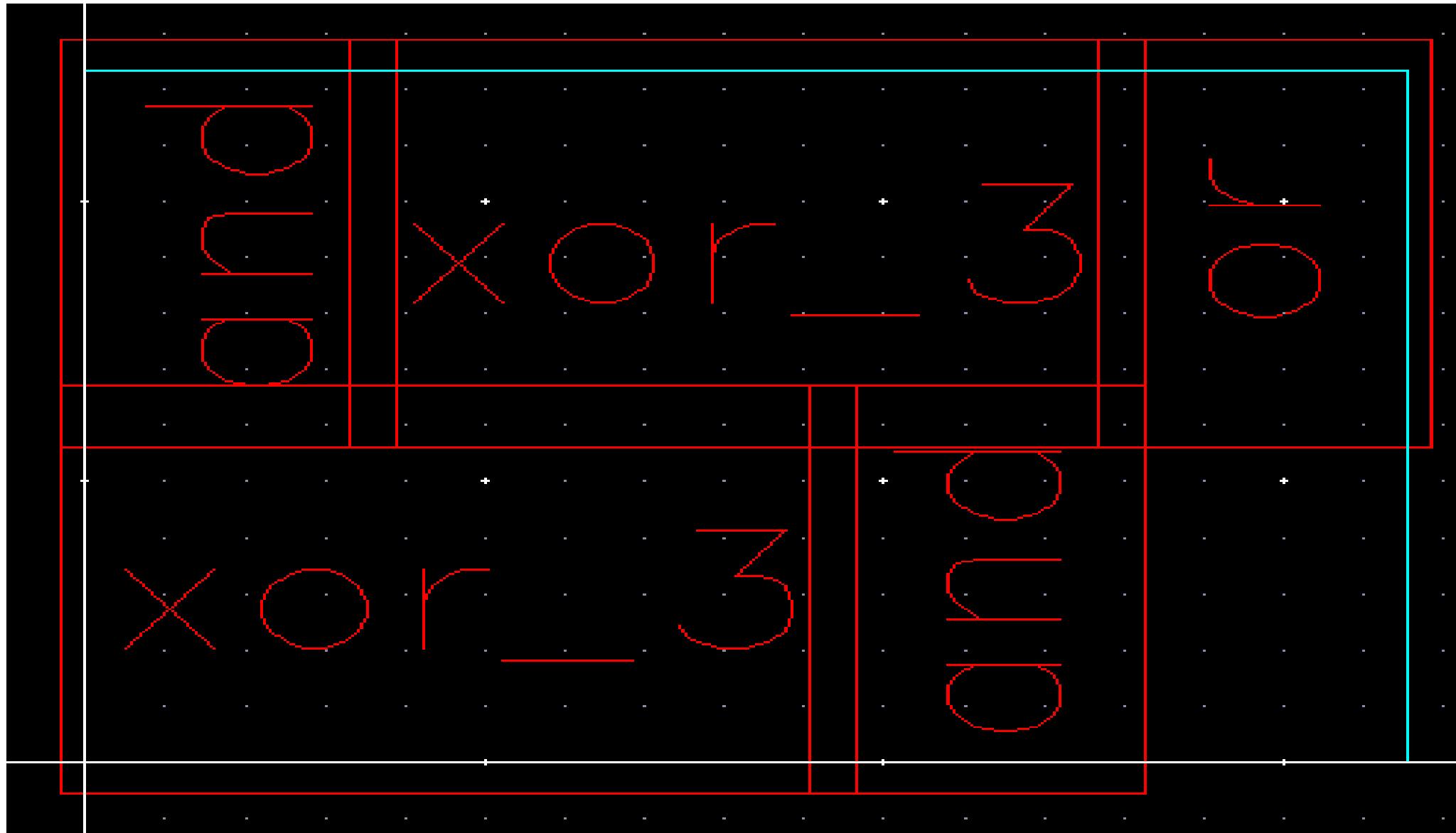
Process	Voltage	Temp	TPLH	TPHL	Current
ff	1.98v	0	618.87ps	696.24ps	1.52uA
tt	1.8v	27	849.22ps	927.601ps	1.336uA
ss	1.62v	100	1.138fs	1.918fs	1.219uA

Number Of Transistor: 228

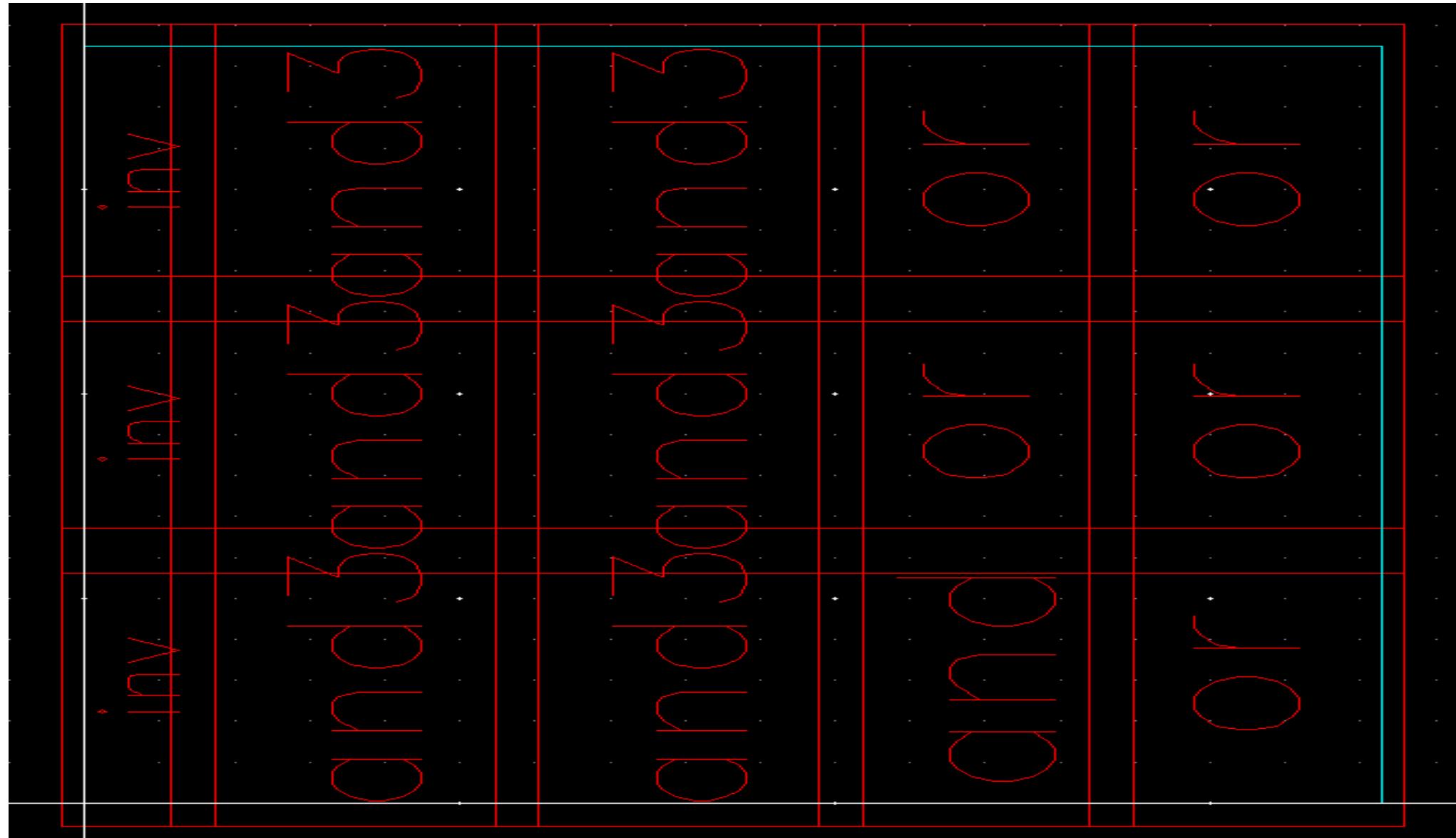
# APPLICATIONS

- Used in digital signal processing(DSP)
- Used in arithmetic logic unit (ALU)
- Parity and error detection

# FLOORPLAN OF FULL ADDER

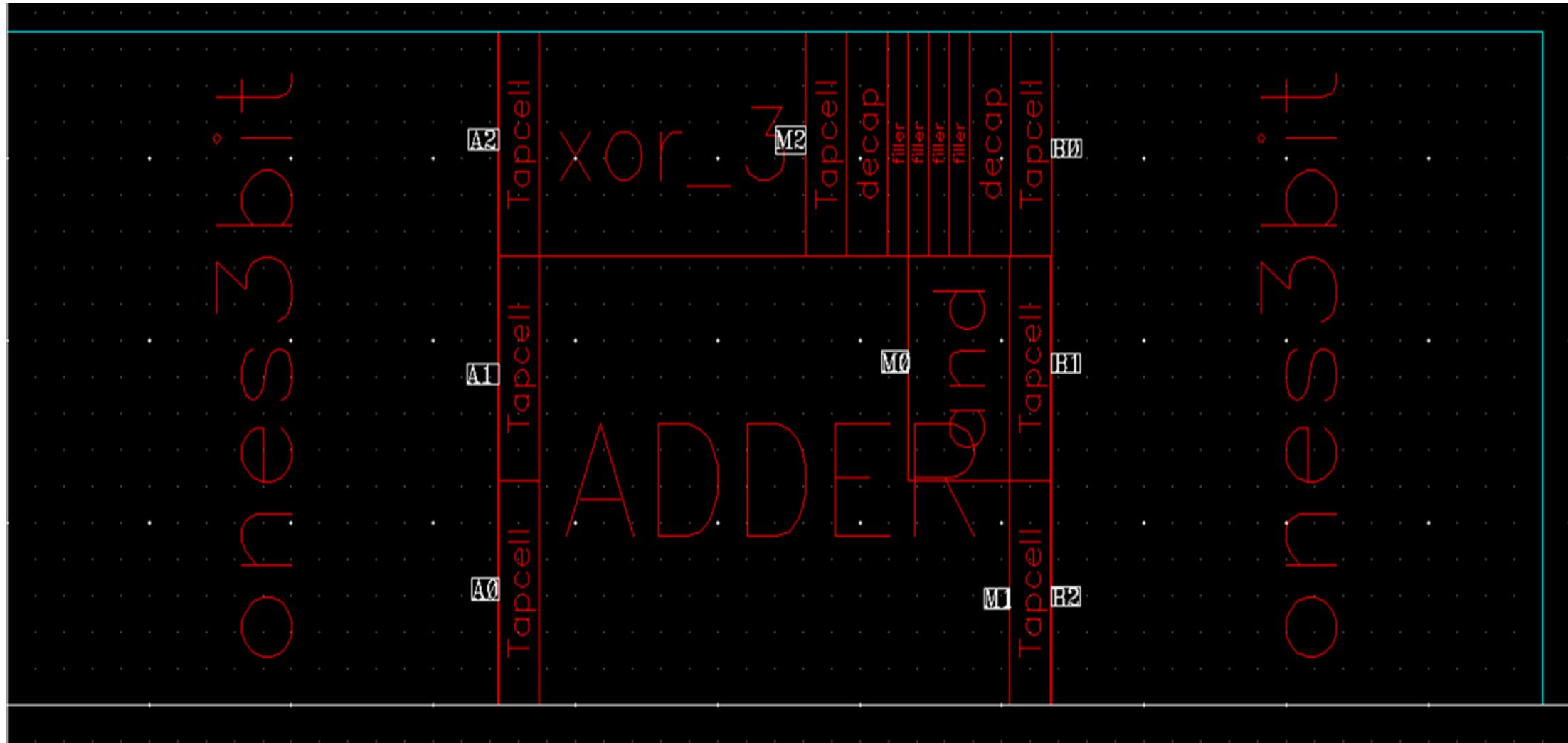


# FLOORPLAN OF ONES\_3\_BIT

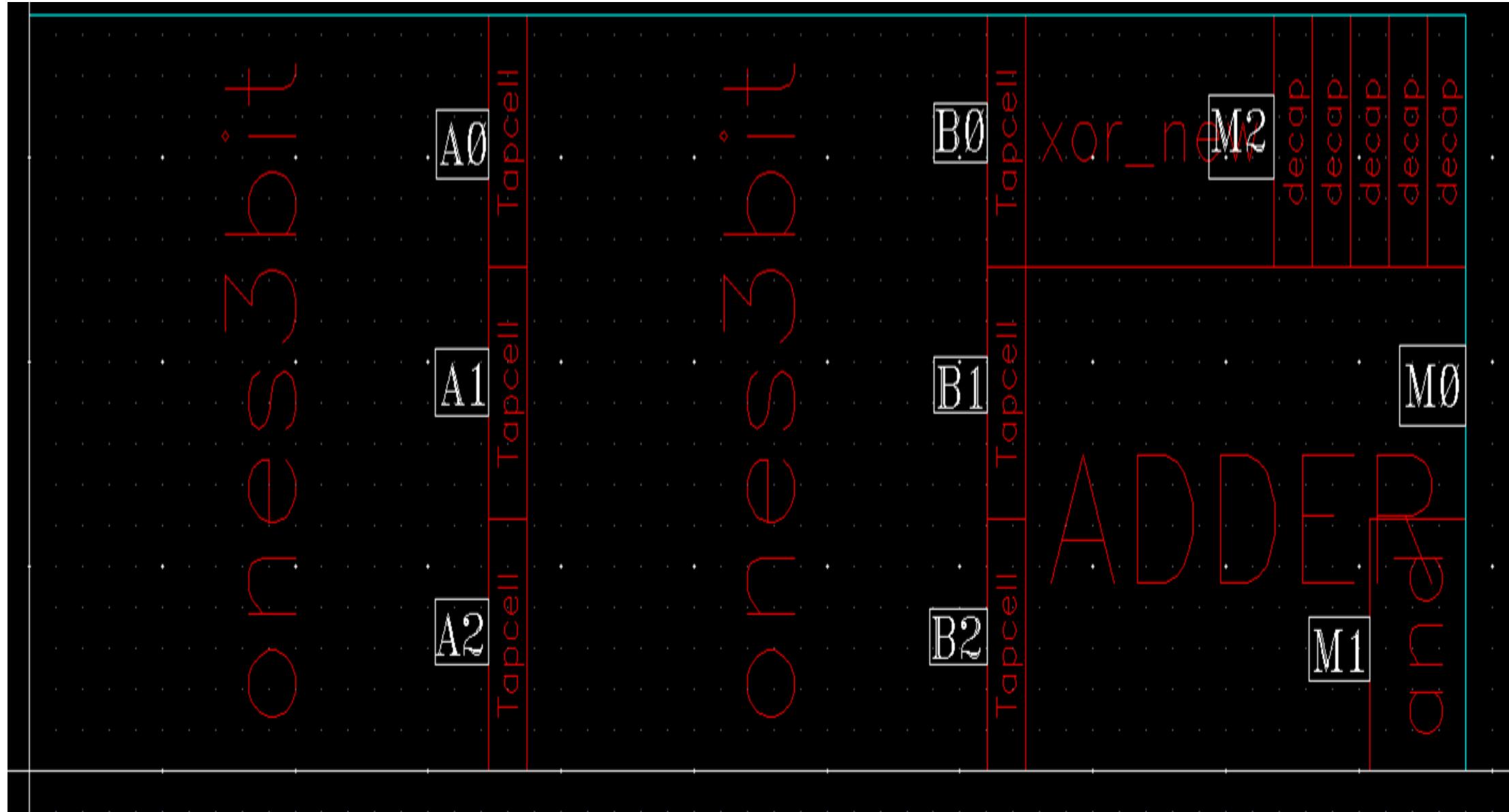


# FLOORPLAN OF BIT1

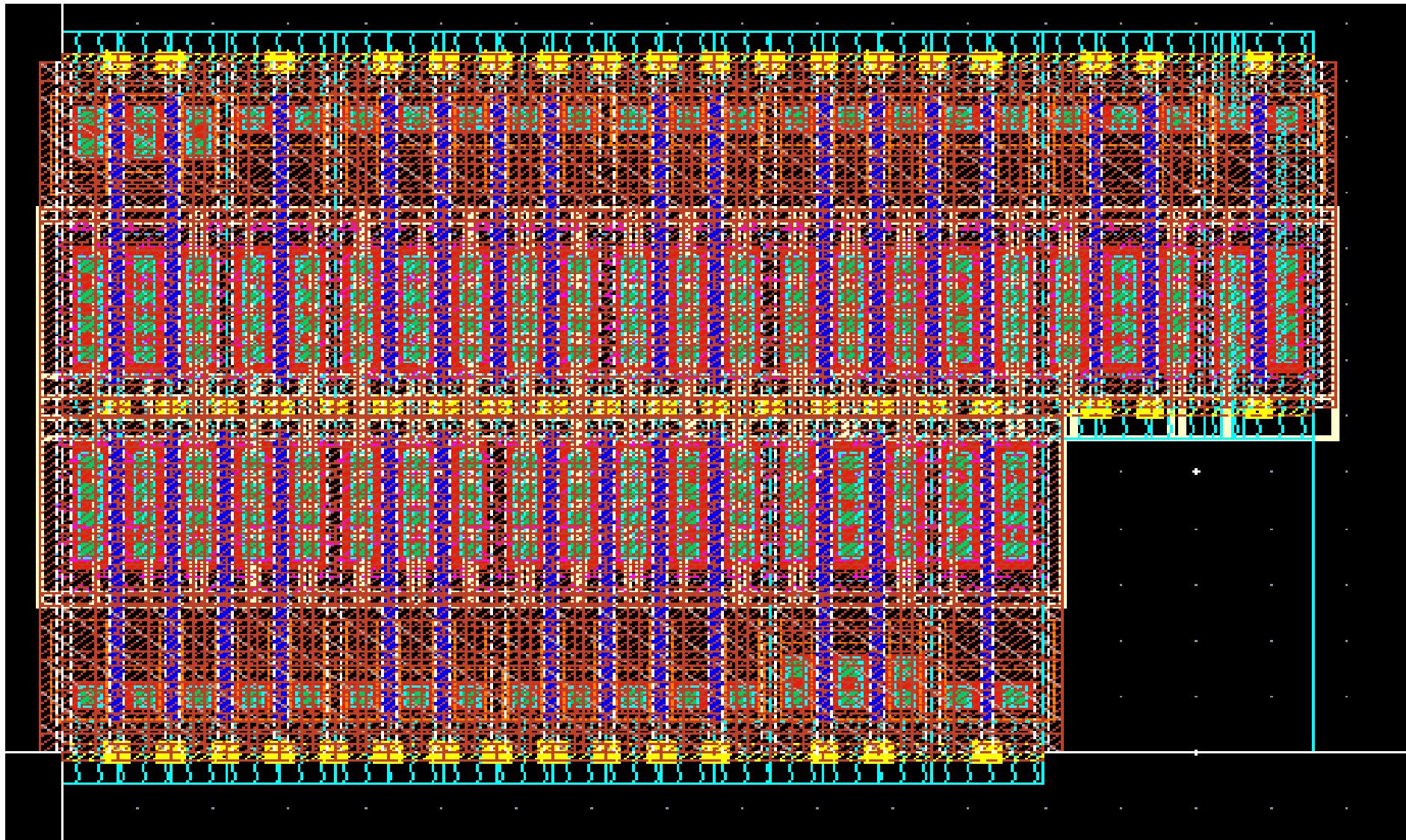
TOTAL AREA = 997.92  $\mu\text{m}^2$



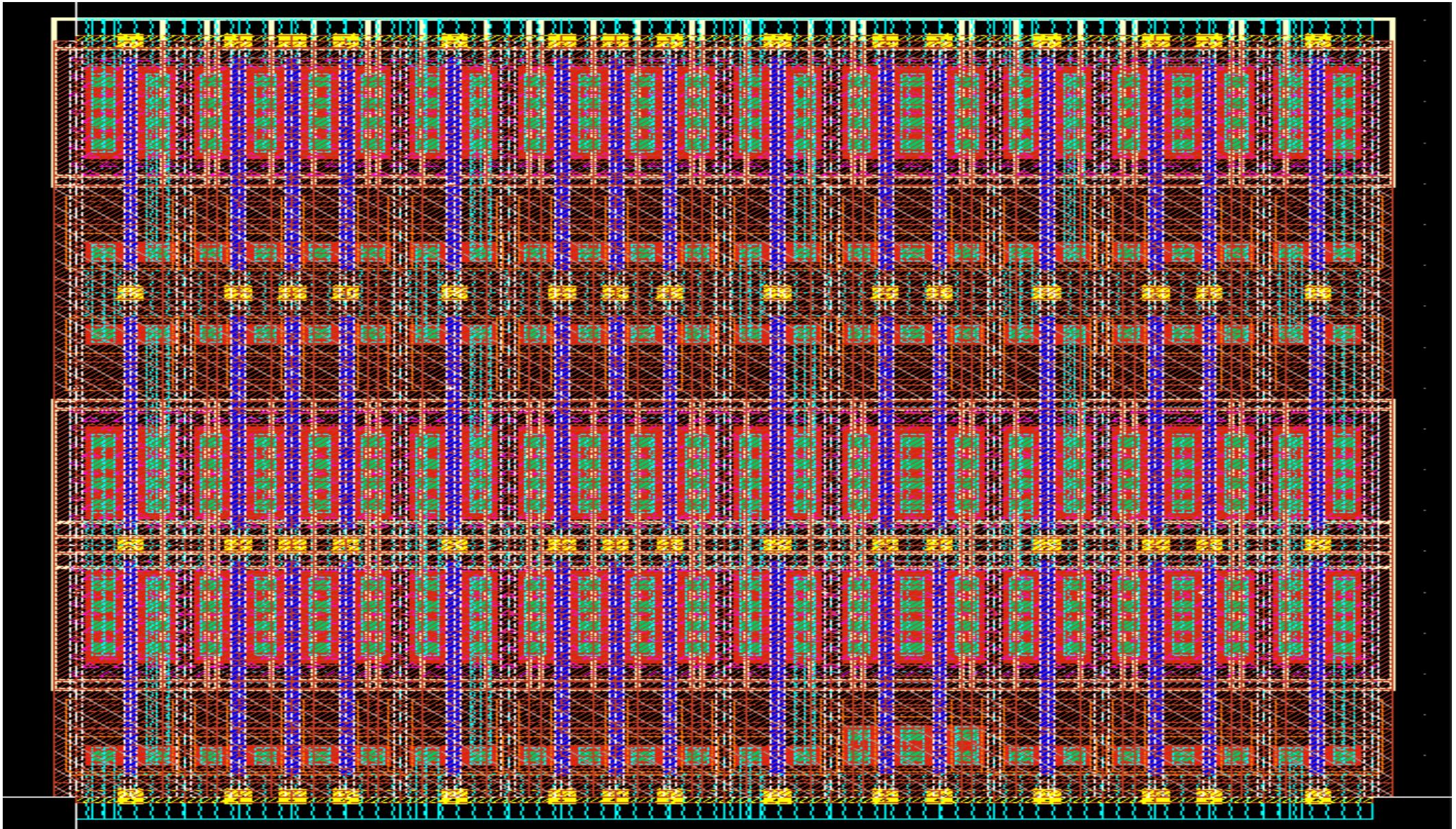
# FINAL\_FLOORPLAN OF BIT1



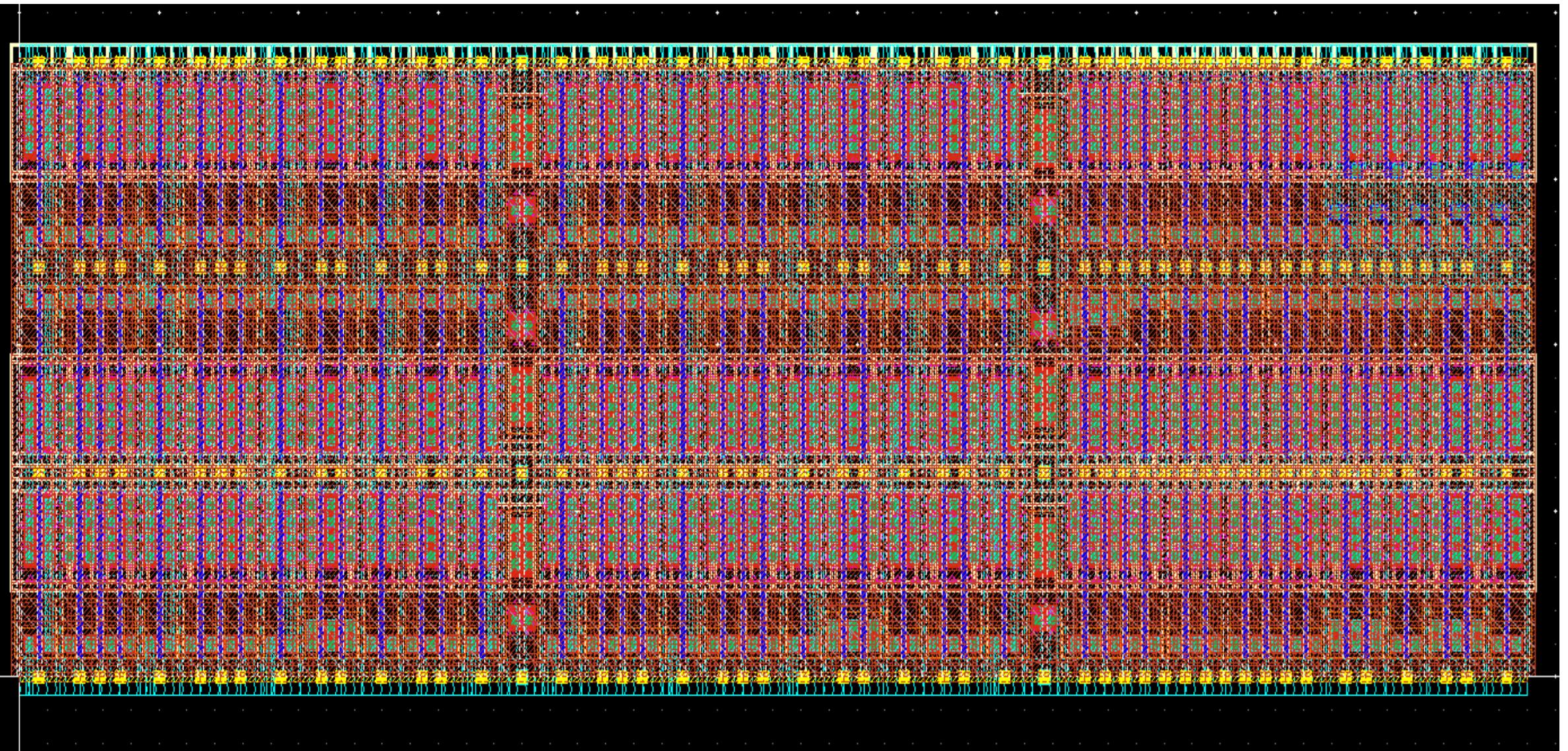
# PLACEMENT OF FULL ADDER



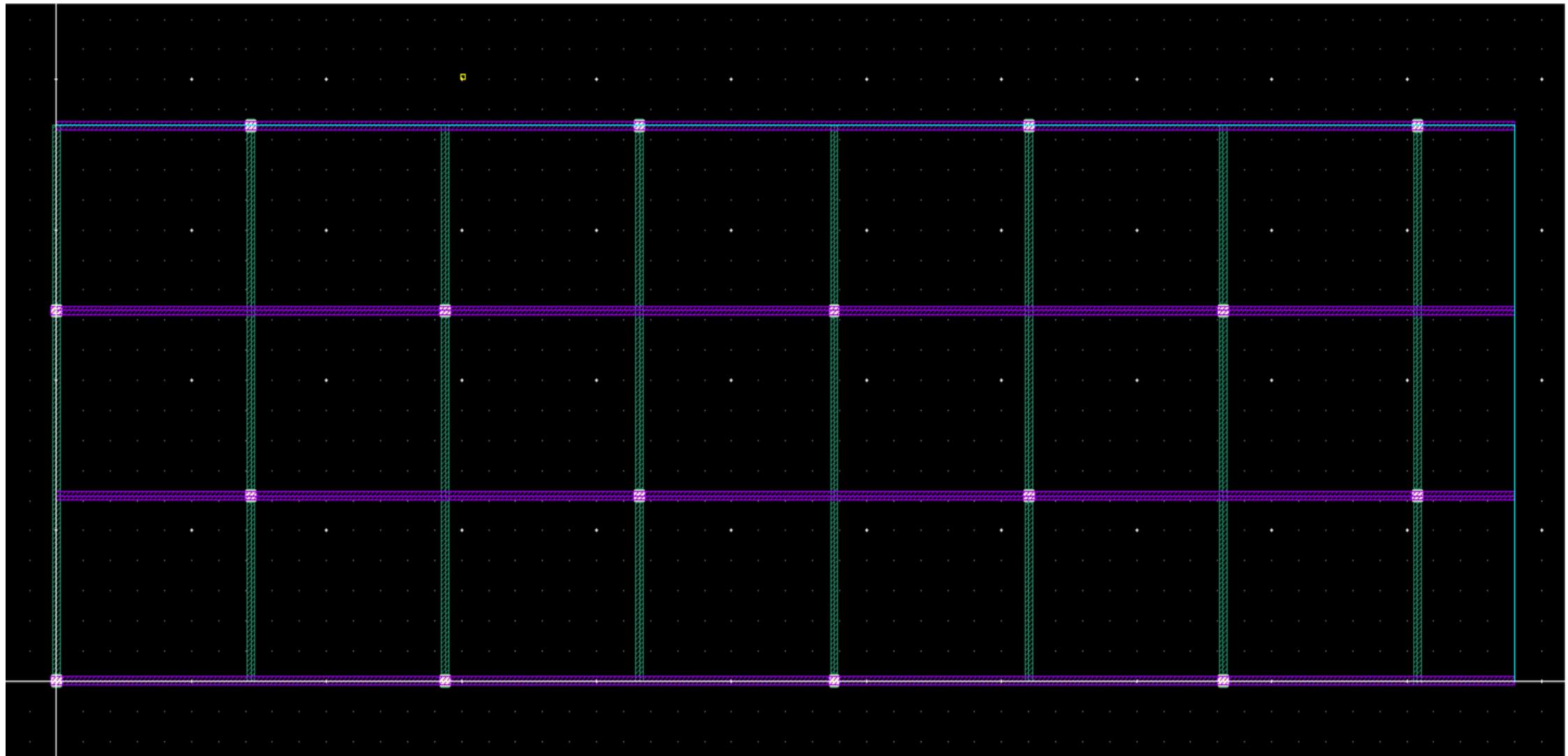
# PLACEMENT OF ONES\_3\_BIT



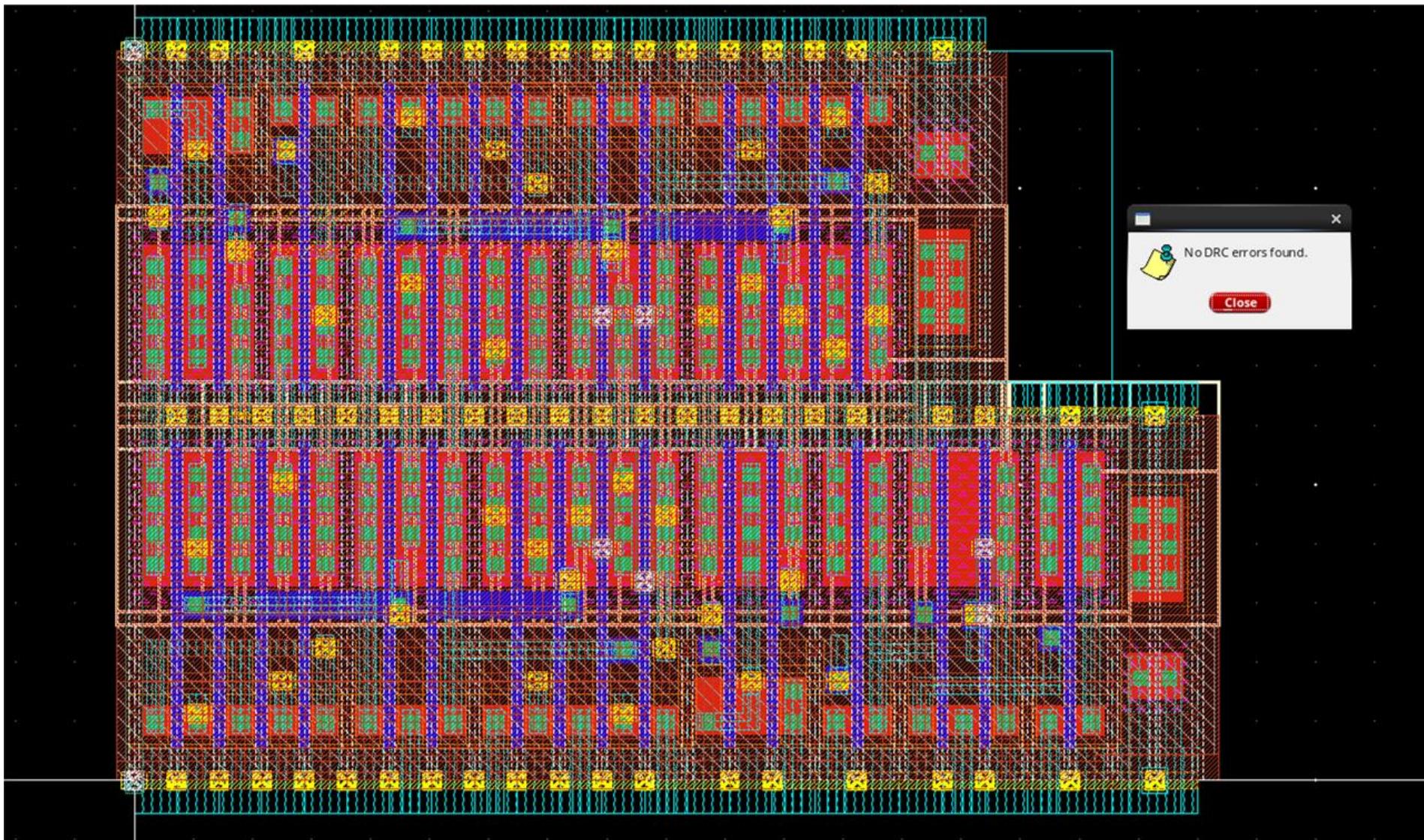
# PLACEMENT OF BIT1

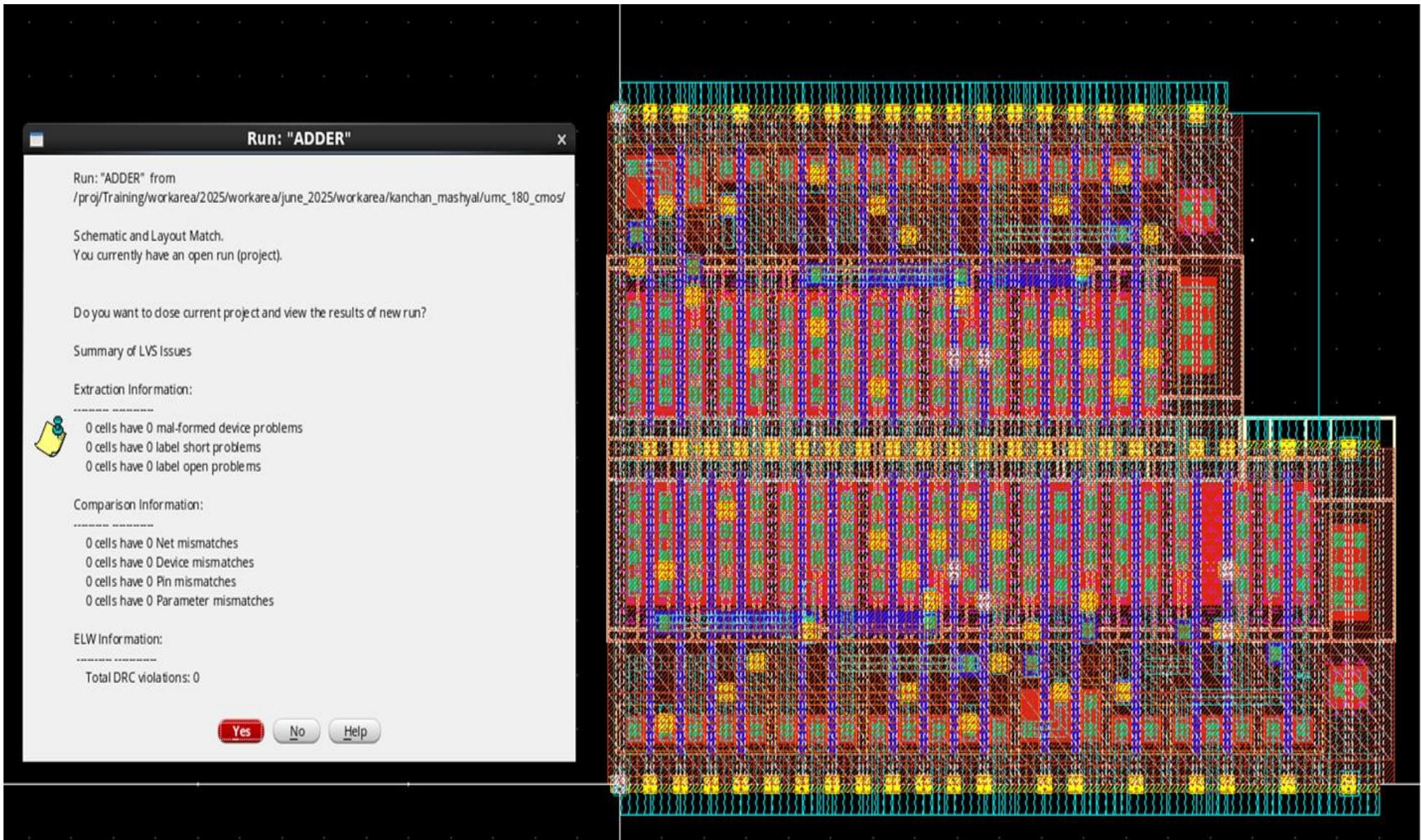


# **POWER\_MESH**

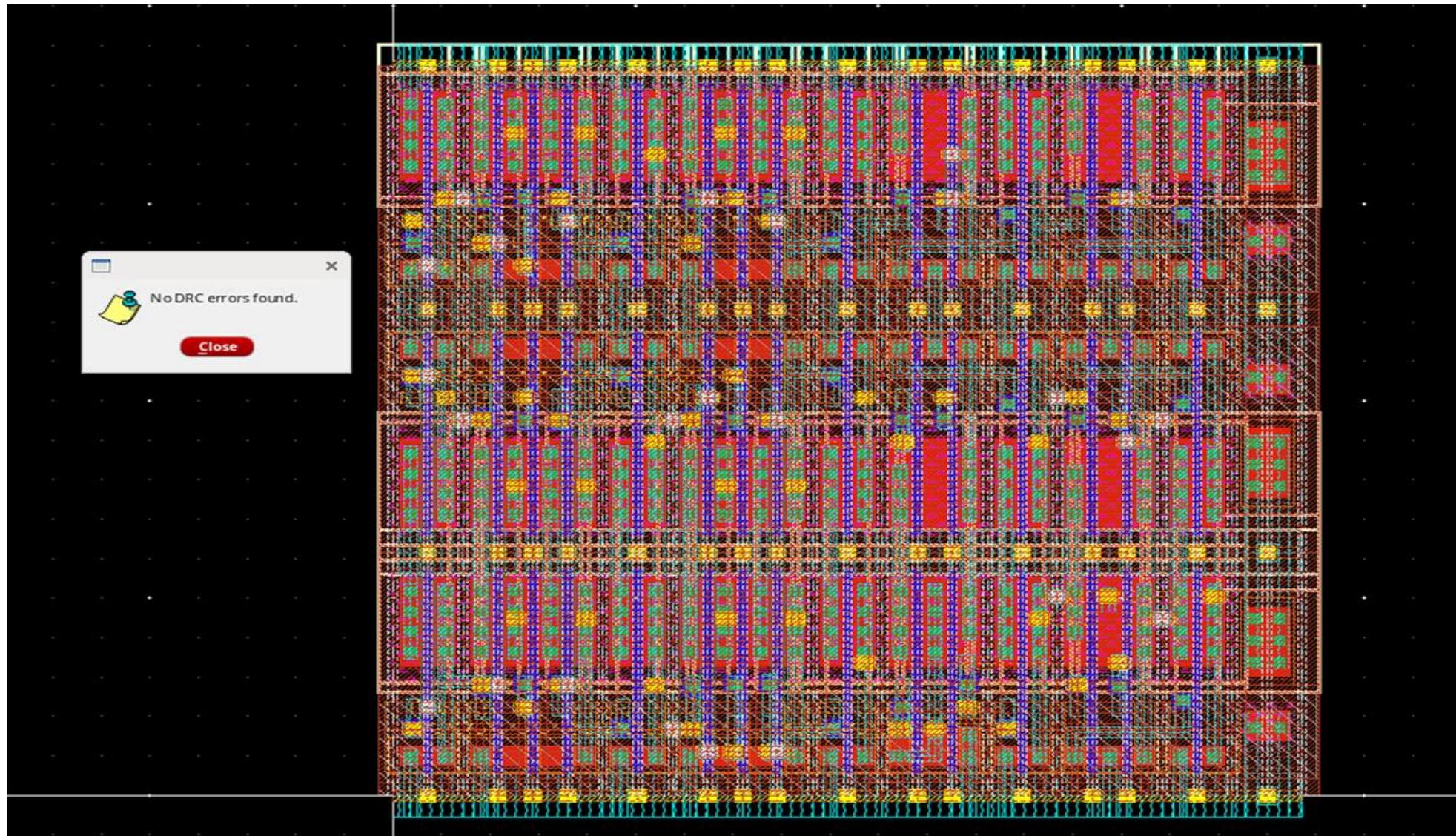


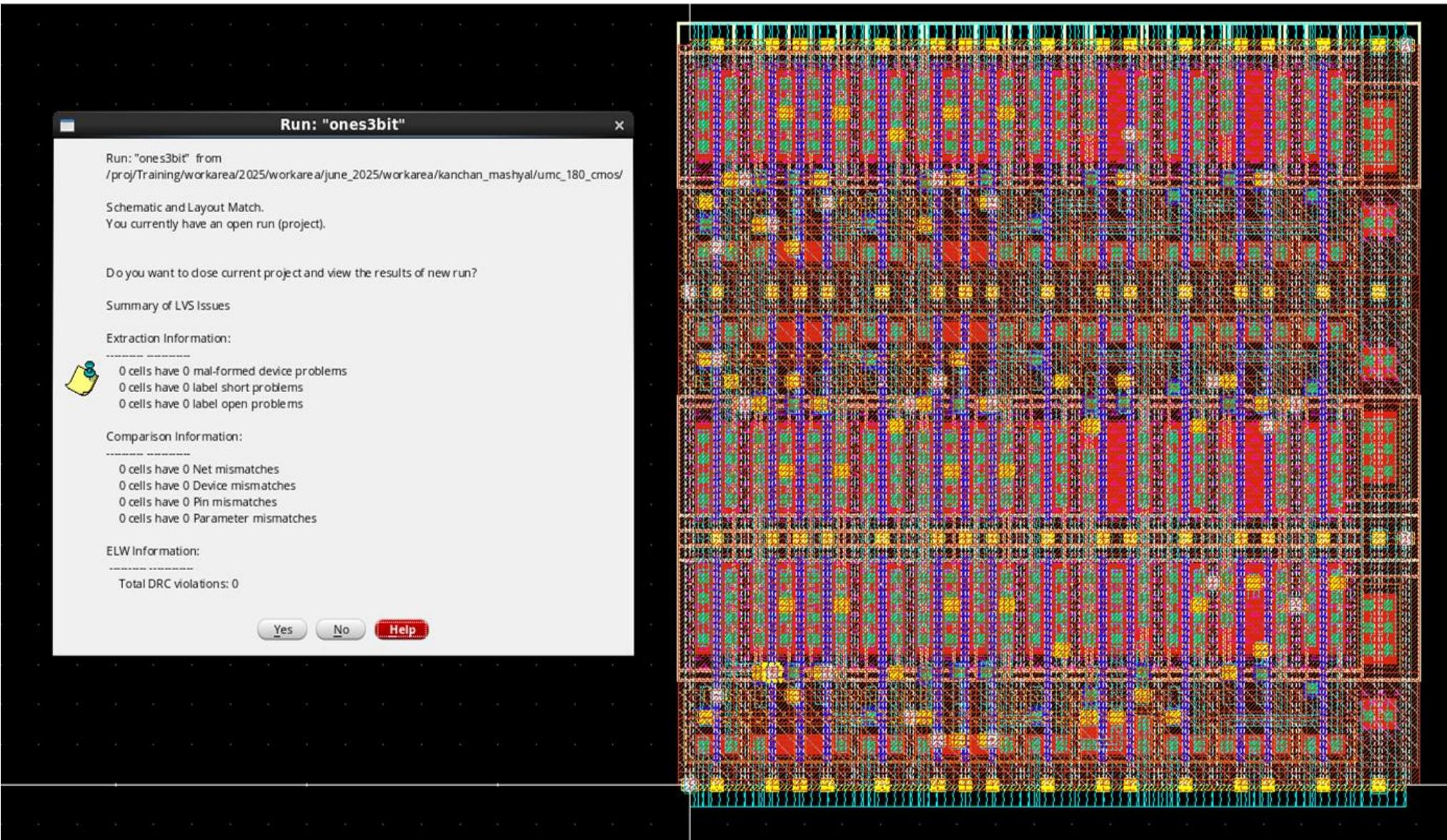
# FULL ADDER DRC & LVS CLEAN



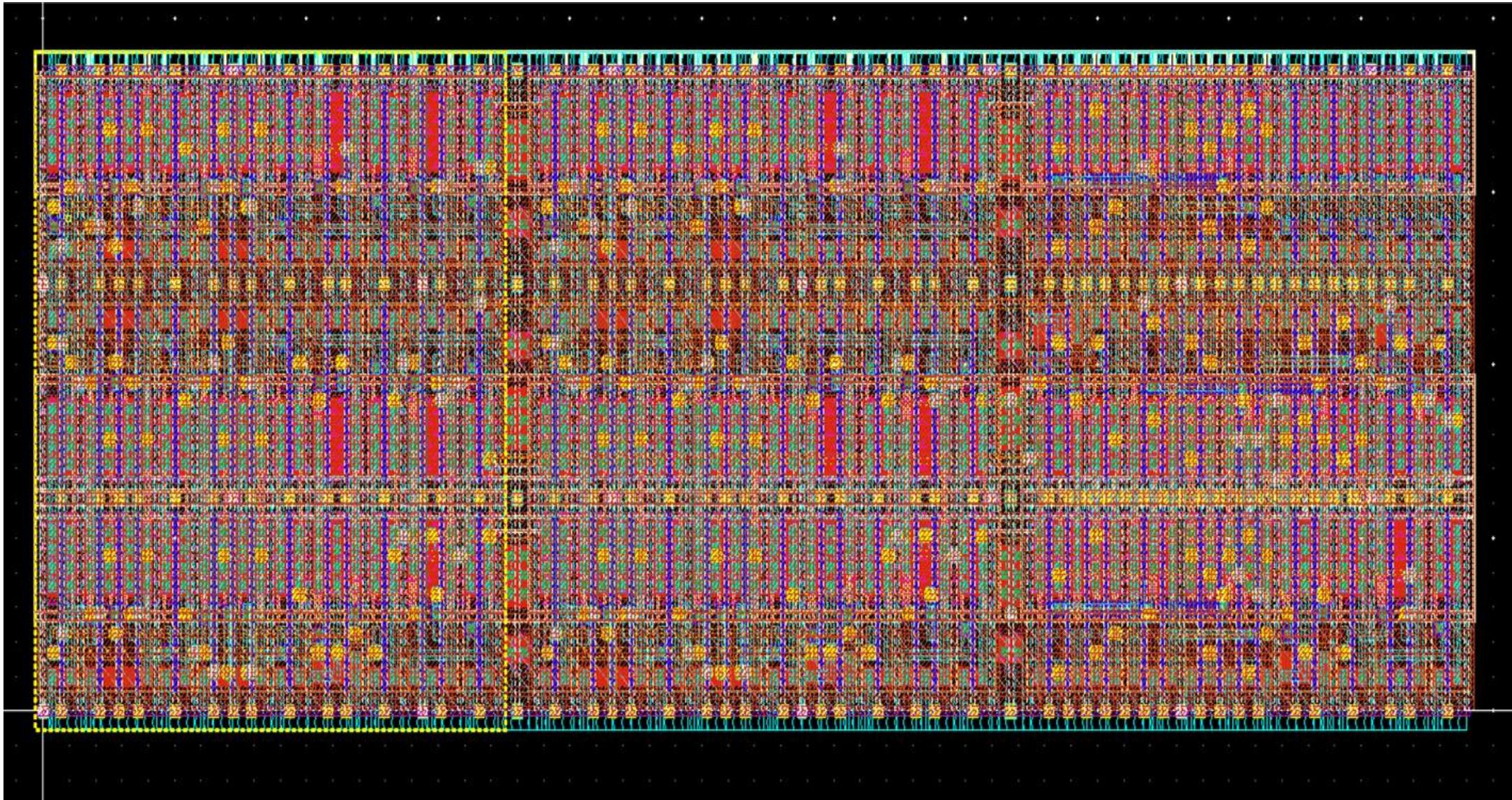


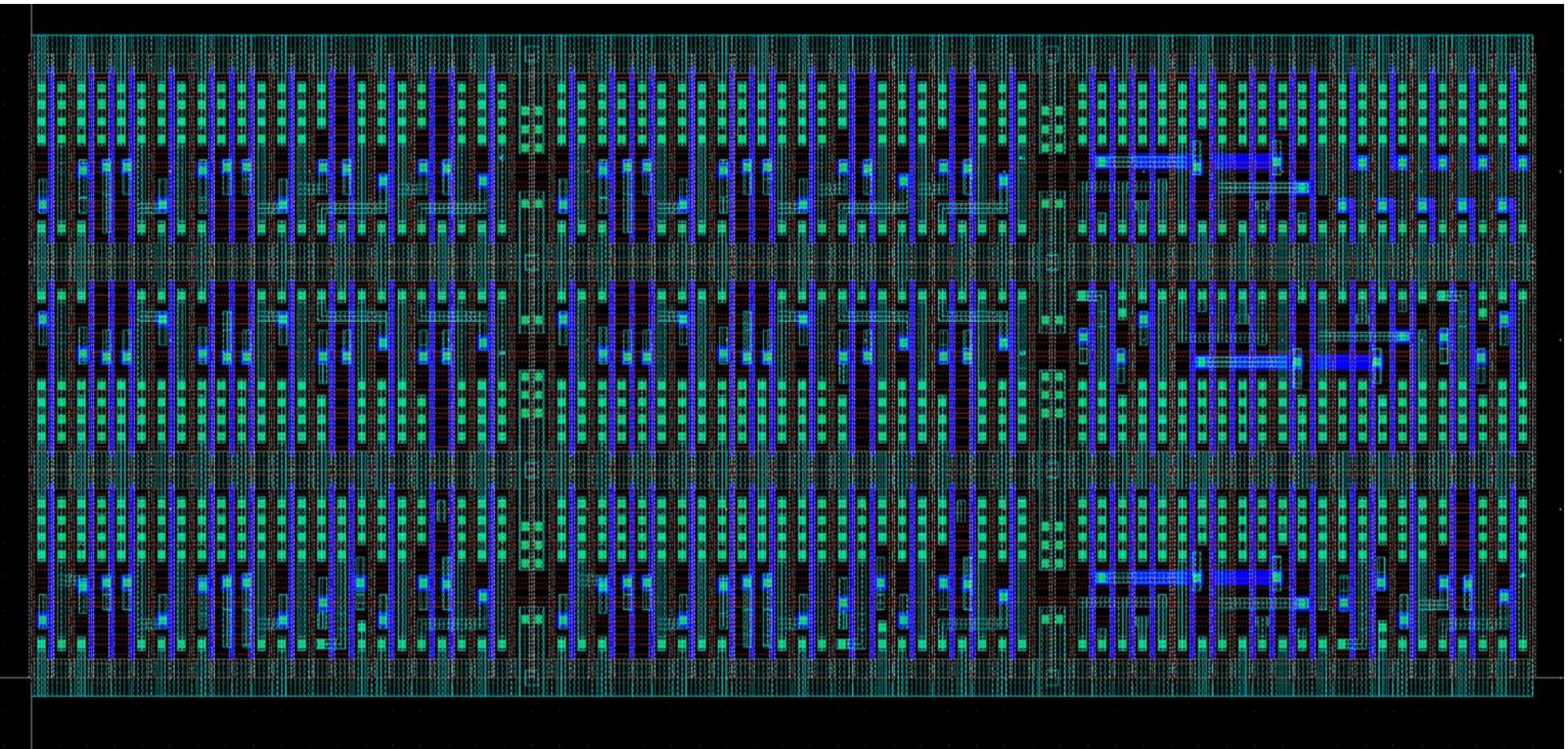
# **ONES\_3\_BIT DRC & LVS CLEAN**

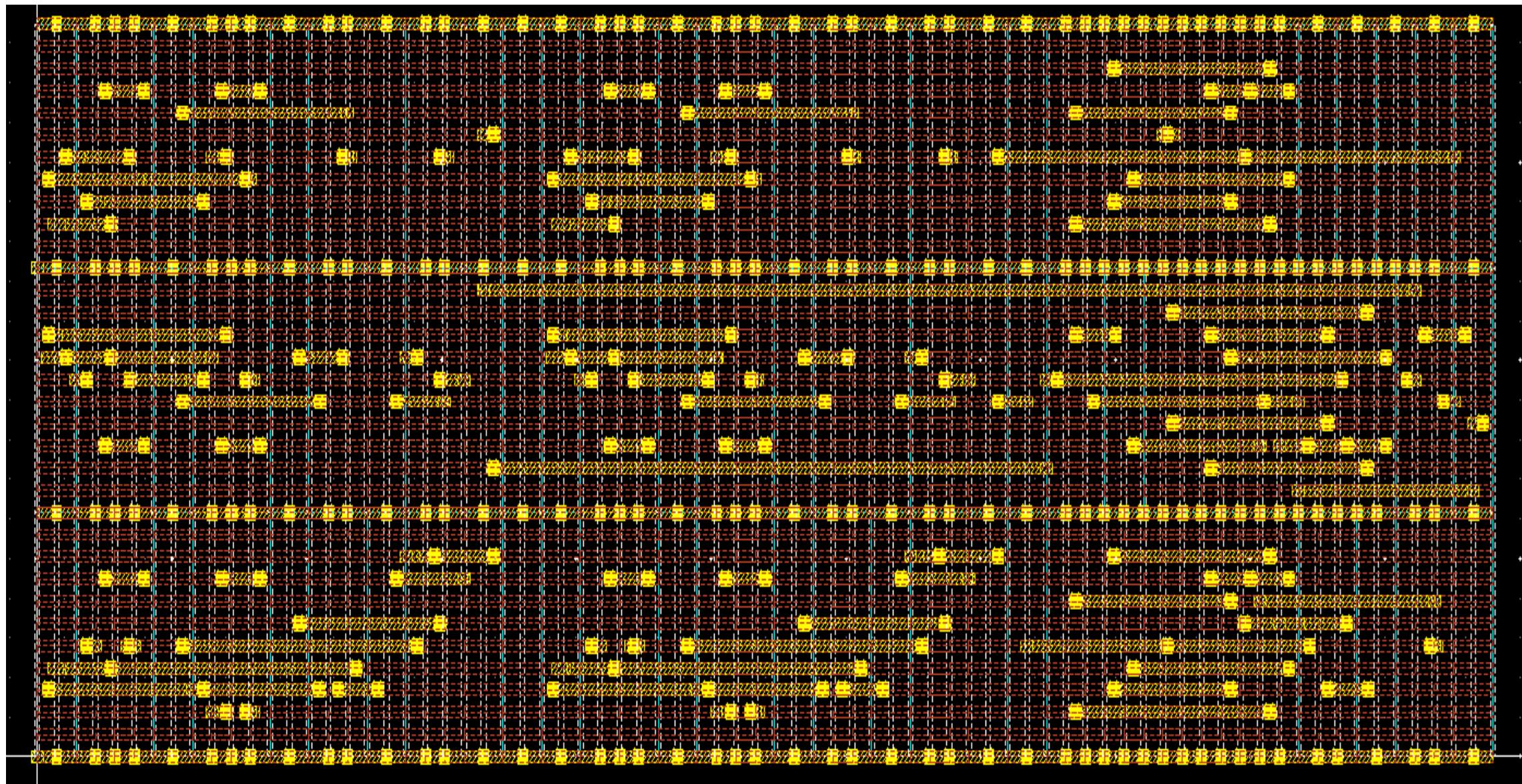


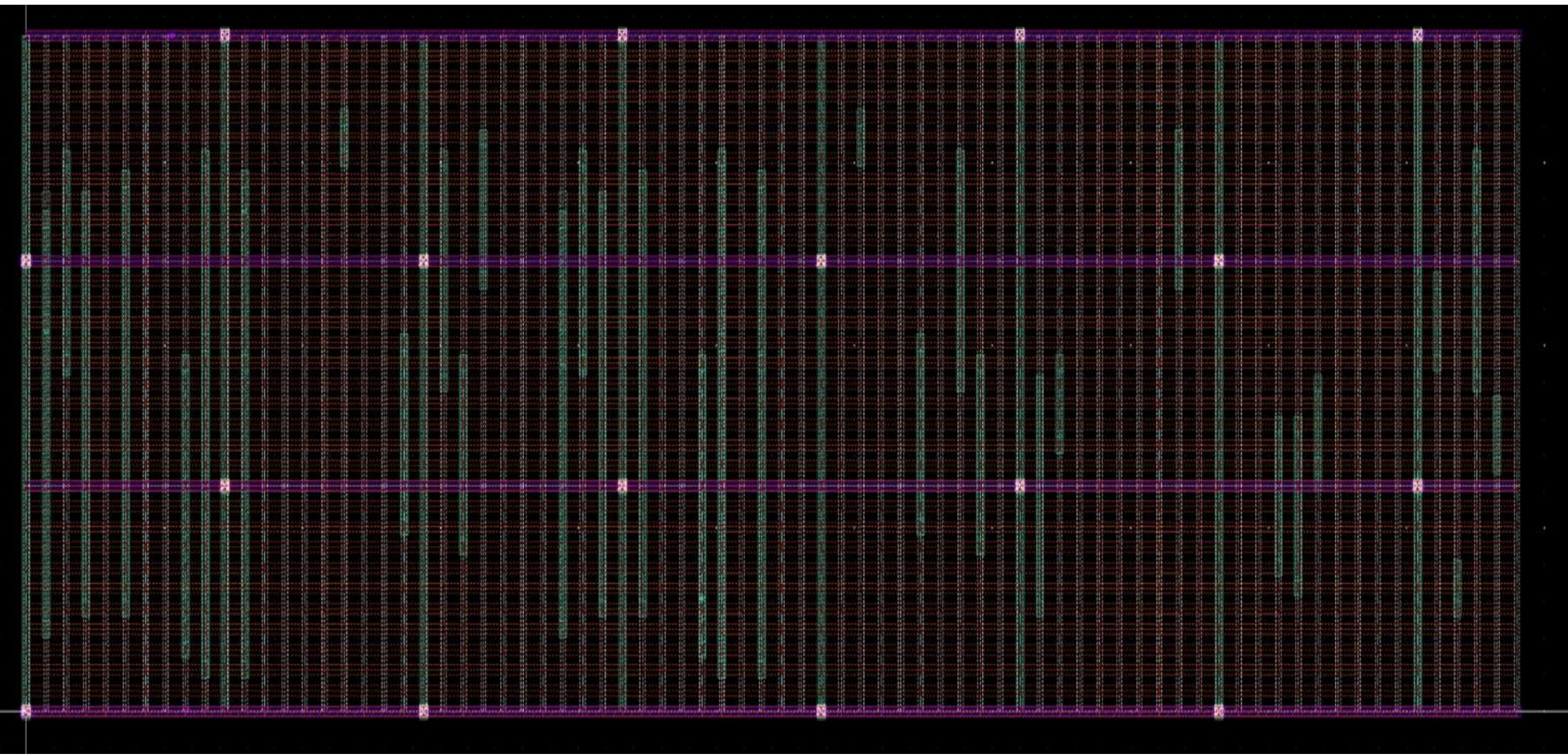


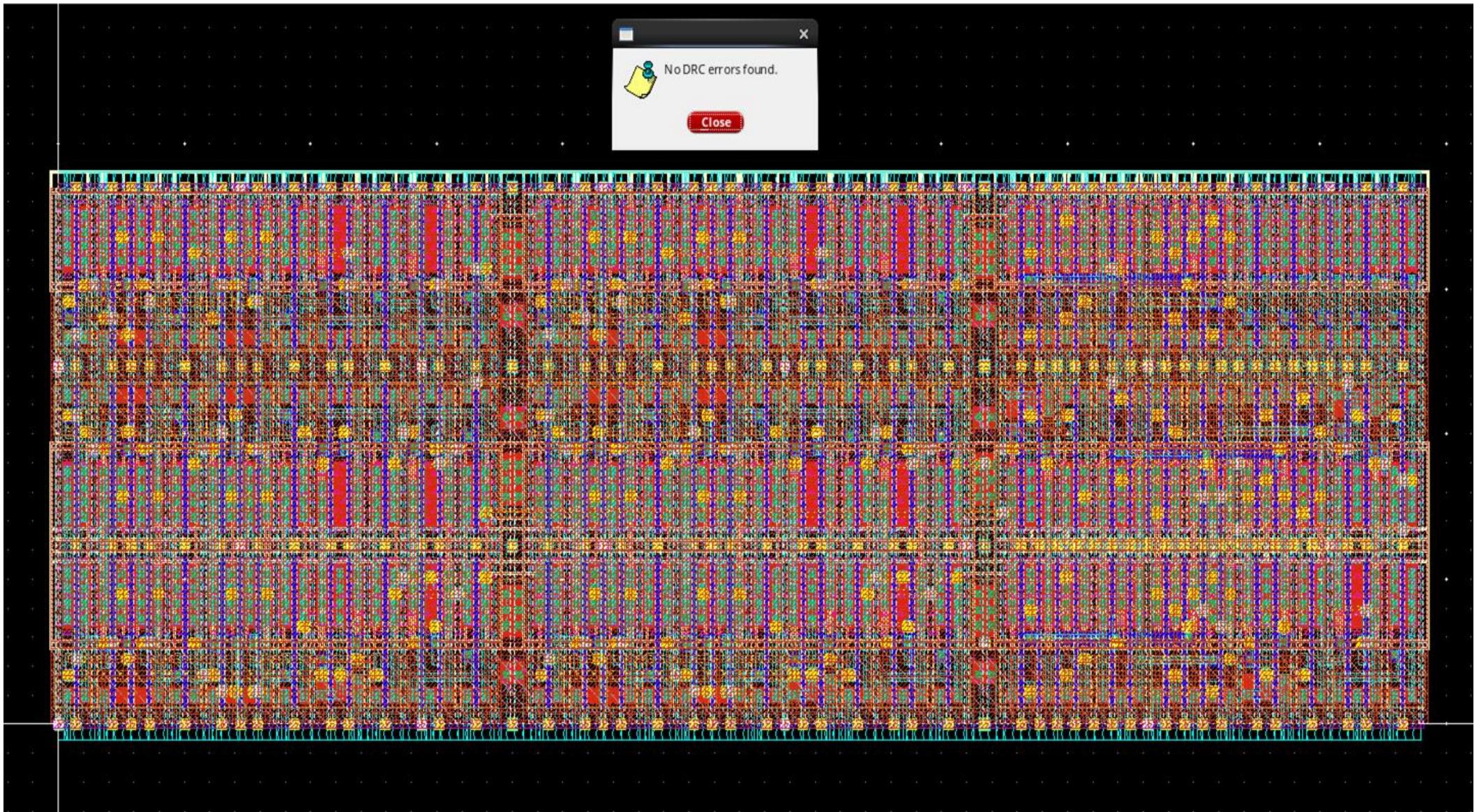
# BIT1 BLOCK ROUTING

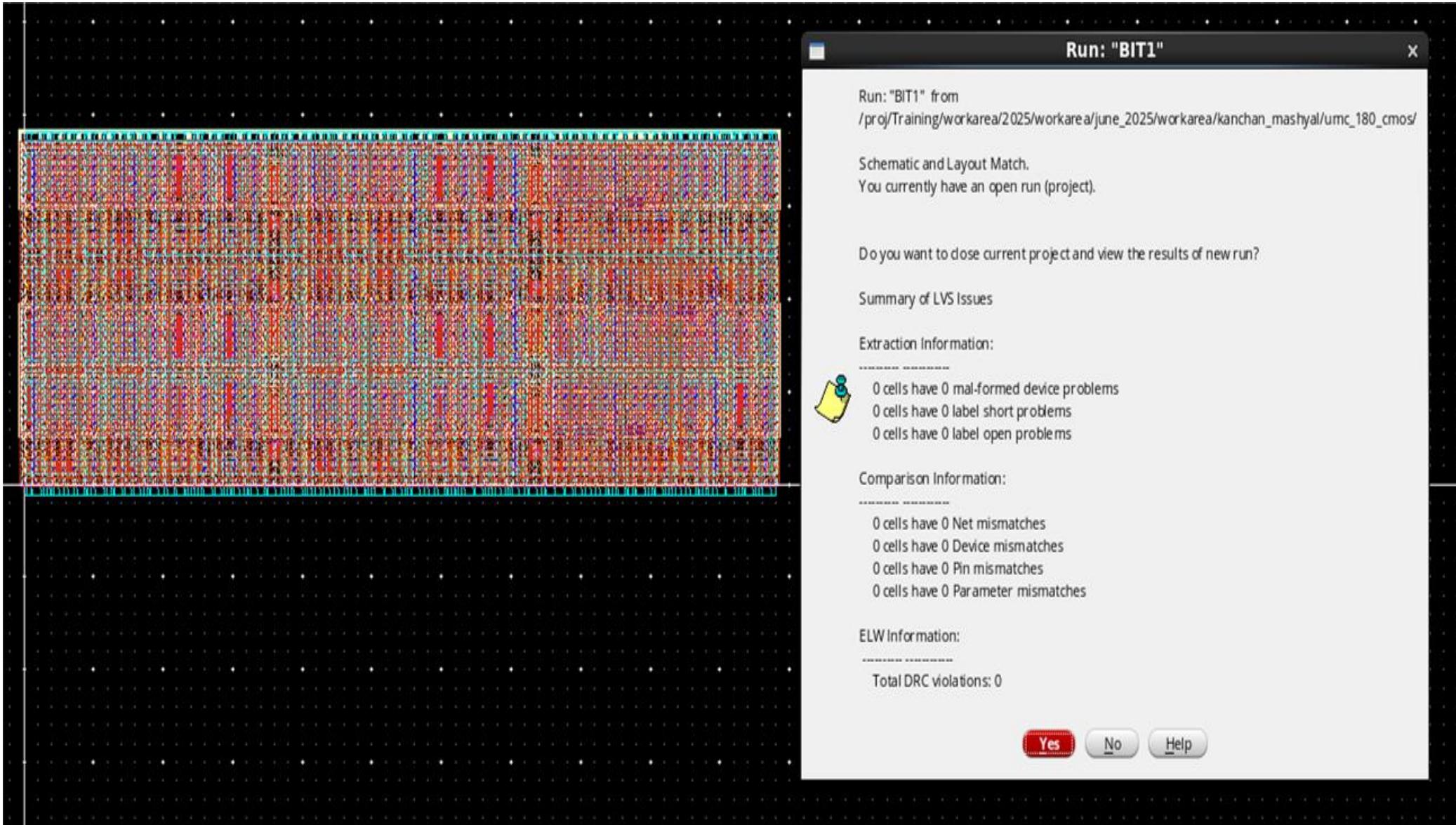












## **OVERALL LEARNING OUTCOMES BY TEAMATES**

- 1.Learned how to do floorplan and placement
- 2.Routing of sub-block
- 3.Top level routing process
- 4.Learned the method of creating power mesh
- 5.Learned to identify various LVS & DRC errors and debug them effectively.

## **CHALLENGES FACED**

- 1.Decap
- 2.EXOR layout in 13 CPP
- 3.During the routing process of ones-3-bit-block
- 4.Changing routing according to power mesh
- 5.Finding longest path in delay

**THANK YOU**