ELL 734-MOS VLSI DESIGN

"Design a 8 bit Multiplier in 65nm technology "

Submitted By:

Kushagra Sharma – 2023EEN2754

Abhilash – 2022JTM2391

Arjit Awadhiya-2023EEN2762

Abhijeet Kumar-2023EEN2722

1.Introduction

1.1 Referenced documents

1. Y. -J. Chang, Y. -C. Cheng, S. -C. Liao and C. -H. Hsiao, "**A Low Power Radix-4 Booth Multiplier With Pre-Encoded Mechanism**," in IEEE Access, vol. 8, pp. 114842-114853, 2020, doi: 10.1109/ACCESS.2020.3003684.

In this paper, a new radix-4 Booth pre-encoded mechanism is proposed to reduce the power consumption of the Booth multiplier. The proposed design can effectively reduce the power of the Booth multiplier dissipated in the redundant activities by disabling the Booth encoders and decoders from unnecessary working. Particularly, since the control signals are generated early at the pipeline input register before the multiplier, the performance of our design is better than the traditional Booth multiplier.

2. Adiono, Trio & Herdian, Hans & Harimurti, Suksmandhira & Putra, Tengku. (2020). **Design of Compact Modified Radix-4 8-Bit Booth Multiplier.** International Journal on Electrical Engineering and Informatics. 12. 228-241. 10.15676/ijeei.2020.12.2.4.

The conventional serial booth multiplier generates full partial product (2n+1) and it require a 2n bit adder to perform the operation. Because of that the area of the circuit will be inefficient. In this paper, Modified booth algorithm is used to reduce the number of partial product calculation
to only half of the serial-parallel multiplier implementation, while keeping the area is still in acceptable level. In order to optimize the area, we propose a design by modifying the architecture of radix-4 modified booth multiplier.

3. **CMOS VLSI Design A Circuits and Systems Perspective**
   Authors: Neil H. E. Weste Macquarie University and The University of Adelaide
   David Money Harris Harvey Mudd College

1.2 Design library name

In this assignment, we have used umc library. The technology we have used here is umc 65nm.

Mosfet we have used are :

N_12_llrvt for N-Mosfet and P_12_llrvt for P-Mosfet.

2.Function

Radix-4 Booth Algorithm

Radix-4 :

The radix-4 Booth algorithm is widely algorithm used to improve the performance of multiplier because it can reduce the number of partial products by half.
Radix 2^r multiplier produce N/r partial products, where N denotes the bits in the number, each of which depends on r bits of the multiplier.

In a typical radix-4 Booth-encoded multiplier design, each group of 3 bits (a pair, along with the most significant bit of the previous pair) is encoded into several select lines ($SINGLE_i$, $DOUBLE_i$, and NEG_i shown in the table below).

Radix-4 produces N/2 partial products. Each partial product is 0, Y, 2Y,-Y,-2Y depending on the pair of bits of X as shown in table.

For our assignment we have taken 8 bit multiplier and multiplicand, so in our case 4 partial products were formed.

X [0:7] = 106 Binary equivalent = 01101010

Y [0:7] = 93 Binary equivalent = 01011101s

X is divided into four groups 3 pairs each with MSB of last pair become LSB of consecutive pair.

A is added at the position of X_{-1} without distorting the original bits combination.

X7,X6,X5 X5,X4,X3 X3,X2,X1 X1,X0,0

They are then passed on to the booth encoder which produces the outputs according to the operations mentioned in the table below:

X_{2i+1}	X_{2i}	X_{2i-1}	Operation	Neg_i	$Double_i$	$Single_i$
0	0	0	0.Y	0	0	0
0	0	1	Y	0	0	1
0	1	0	Y	0	0	1
0	1	1	2Y	0	1	0
1	0	0	-2Y	1	1	0
1	0	1	-Y	1	0	1
1	1	0	-Y	1	0	1
1	1	1	-0=0	1	0	0

For the above example:

X [0:7] = 106 Binary equivalent = 01101010

Y [0:7] = 93 Binary equivalent = 01011101

A zero bit is added at the position before LSB in X

X= 01101010 0

According to above table:

100 -> **-2Y**

101 -> **-Y**

101 -> **-Y**

011 -> **2Y**

Here,

Y means the bits were taken as it is.

-Y means taking the 2's complement of the bits.

2Y means simply shifting all the bits of Y to the left by one.

-2Y means taking the 2's complement and then shift the bits to the left by one.

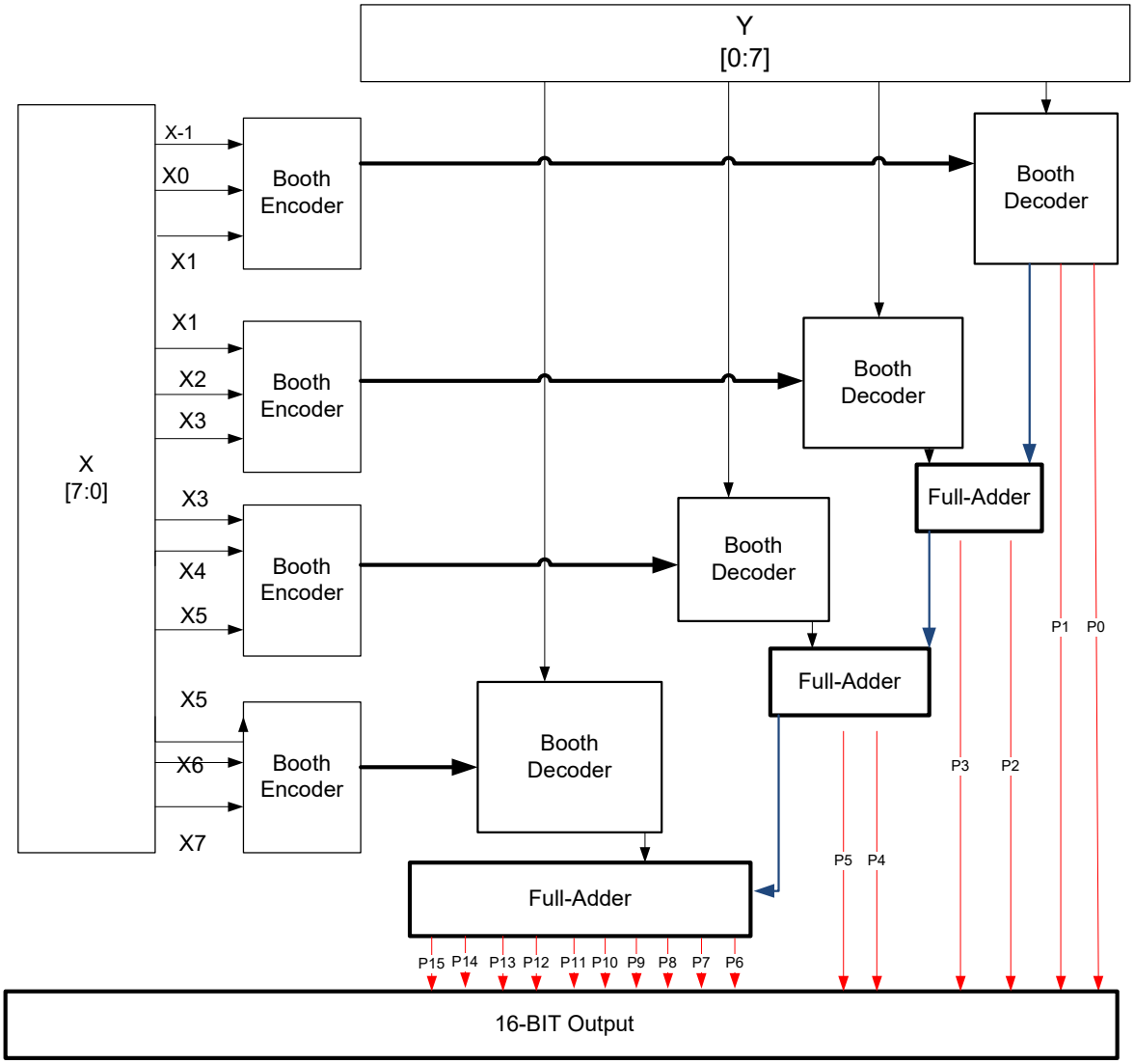
Y	0	0	0	0	0	0	1	0	1	1	1	1	0	1
-2Y	1	1	1	1	1	1	0	1	0	0	0	1	1	0
-Y	1	1	1	1	1	0	1	0	0	0	1	1		
-Y	1	1	0	1	0	0	0	0	1	1				
2Y	1	0	1	1	1	0	1	0						
Result	1	0	0	1	1	0	1	0	0	0	0	0	1	0

106 X 93 =9858 -> 10011010000010

The output of multiplication two 8-bit number would be a 16-bit product.

The maximum 8-bit number is 255. So, the product of 255 x 255 is 65025. Its binary equivalent is 1111111000000001 it is a 16-bit number.

2.3 Architecture



Modified Booth Algorithm :

We have given the 3-pair of bits to each of the four booth encoder which produces the operation output according to the table shown above.

Then the output of the encoder is given to the booth decoder which produces the partial products based on operations.

Then the arithmetic shift and adding of the partial products take place using the 16-bit full adder.

Design of Gates and Simulation:

The logical effort is used to determine the delay of the logic gate with formula:

$$d = gh + p$$

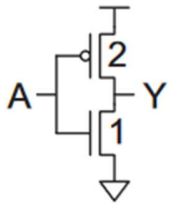
d is the delay, g is the logical effort of the gates, h is the electrical effort, p is the parasitic capacitance.

The logical effort of the gate is dependent on the aspect ratio of the transistors.

Here, we have taken NOT gate or inverter as an reference for all the gates. So, we have the taken pmos to be double in width that of nmos for unit resistance in inverter.

Logical effort is the ratio of the input capacitance of the gate to the input capacitance of the inverter.

$g_{NOT} = 3/3 = 1$, $C_{in} = 3$

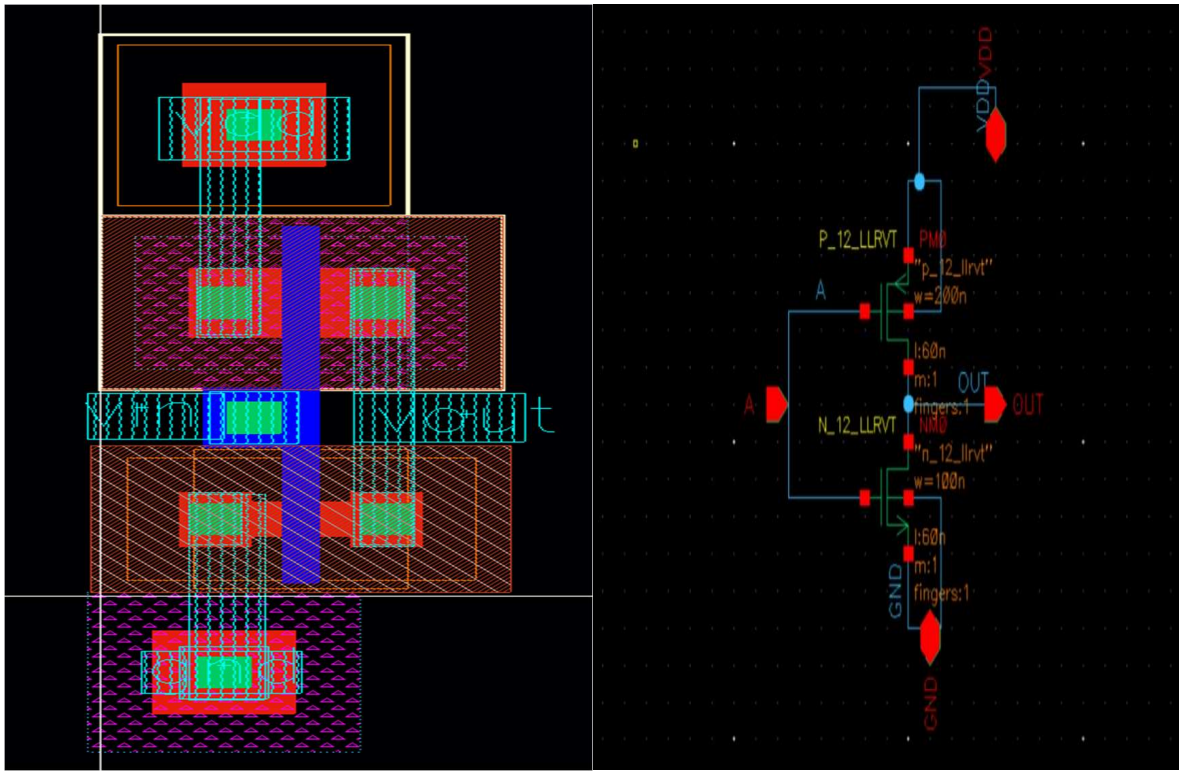


Gates	Inputs=2
NAND	$C_{in}=4$, $g=4/3$
NOR	$C_{in}=5$, $g=5/3$
XOR	$C_{in}=12$, $g=4$

2.4 Detail Functional Description

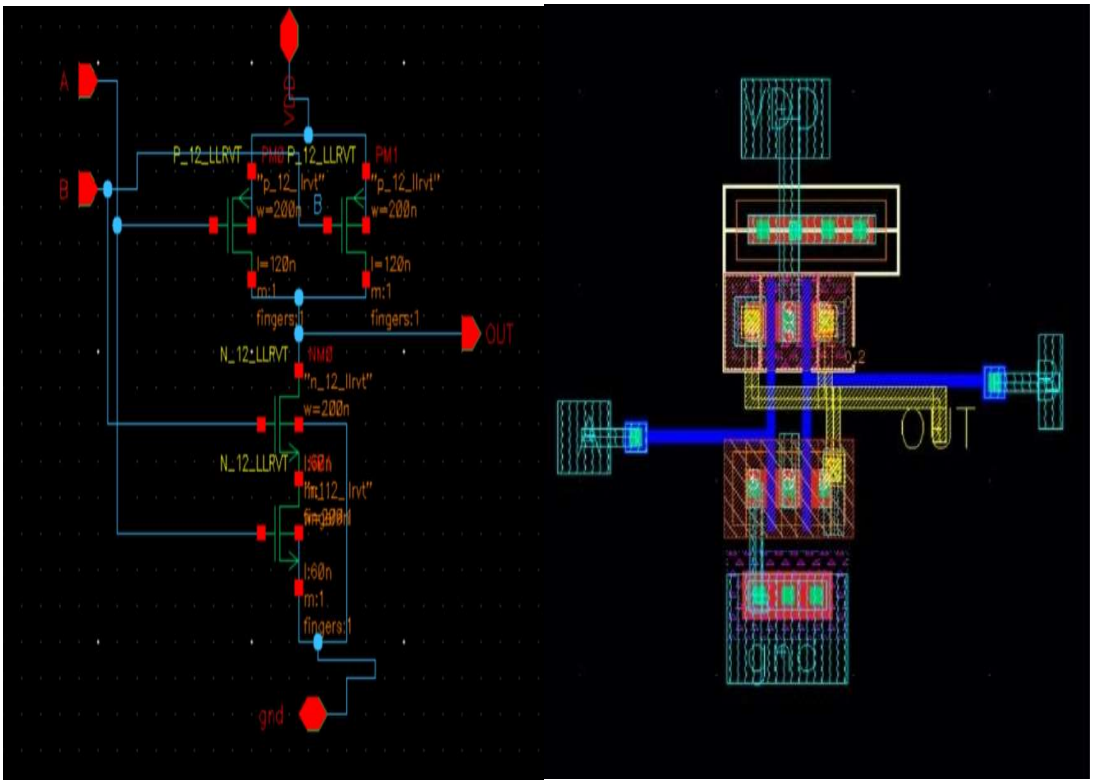
1. Logic gates (NOT, NAND2, NOR2, EXOR2)
2. Full-Adder
3. Booth encoder
4. Booth Decoder
5. Encoder-Decoder
6. 16-bit full adder
7. Unsigned 8x8 Modified Booth Multiplier

NOT GATE



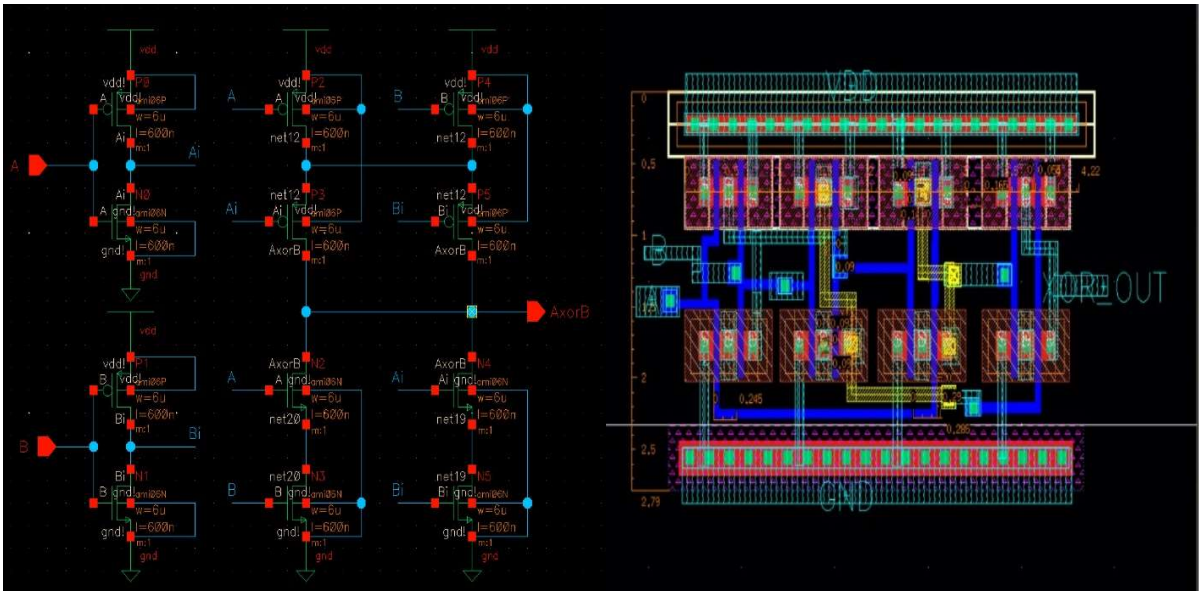
Schematic and layout of NOT GATE

NAND GATE



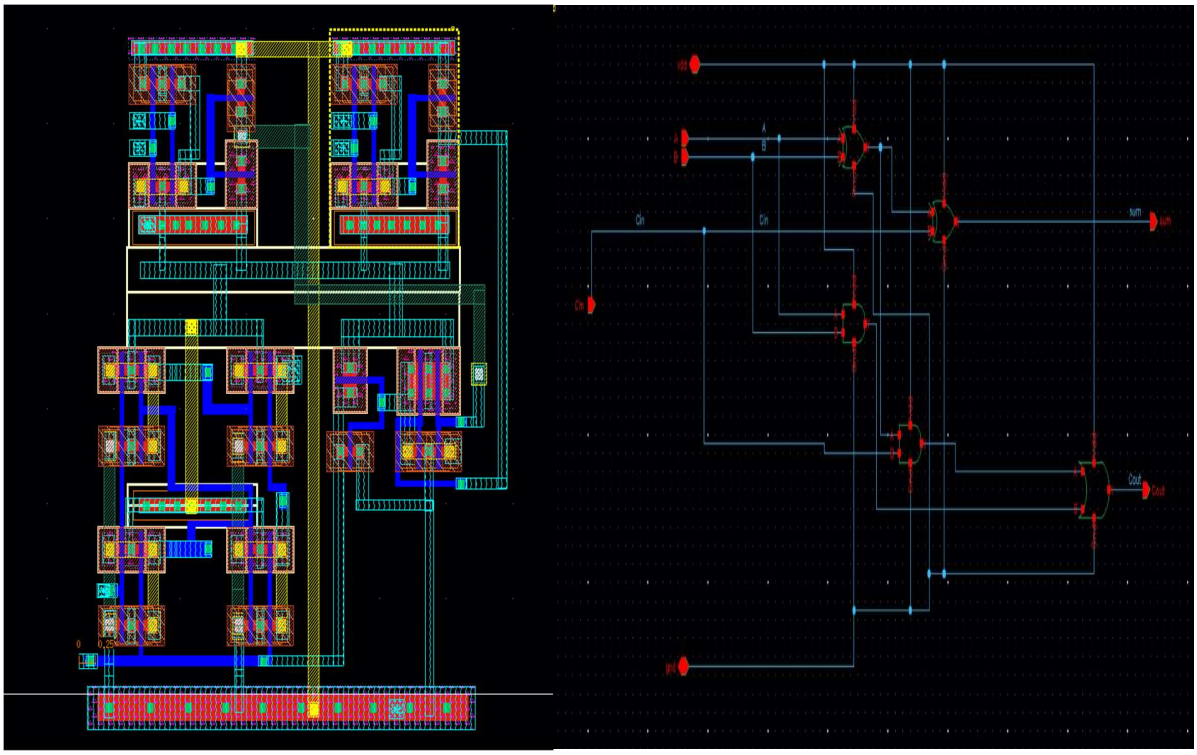
Schematic and layout of NAND GATE

XOR GATE



Schematic and layout of XOR GATE

Full Adder



Schematic and layout of Full Adder

Booth Encoder

The diagram shown below is the booth encoder. The function of booth encoder is to generate multiplication operation 0, Y, 2Y, -Y, -2Y based on the inputs of the multiplier X, according to the table shown in the starting.

Then the output of the encoder goes to the decoder which will then generates partial products.

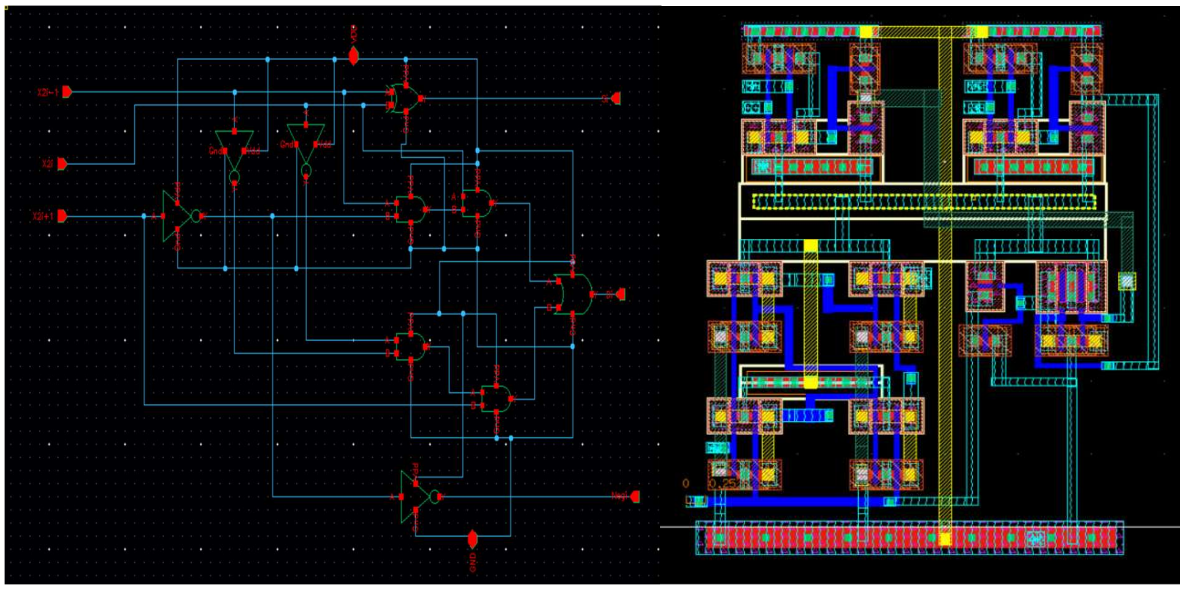


Fig. Booth Encoder schematic and layout

Booth Decoder

The diagram shown below is our booth decoder. The inputs to this are the outputs of booth encoder and pair of bits of multiplicand Y. It will generates partial products.

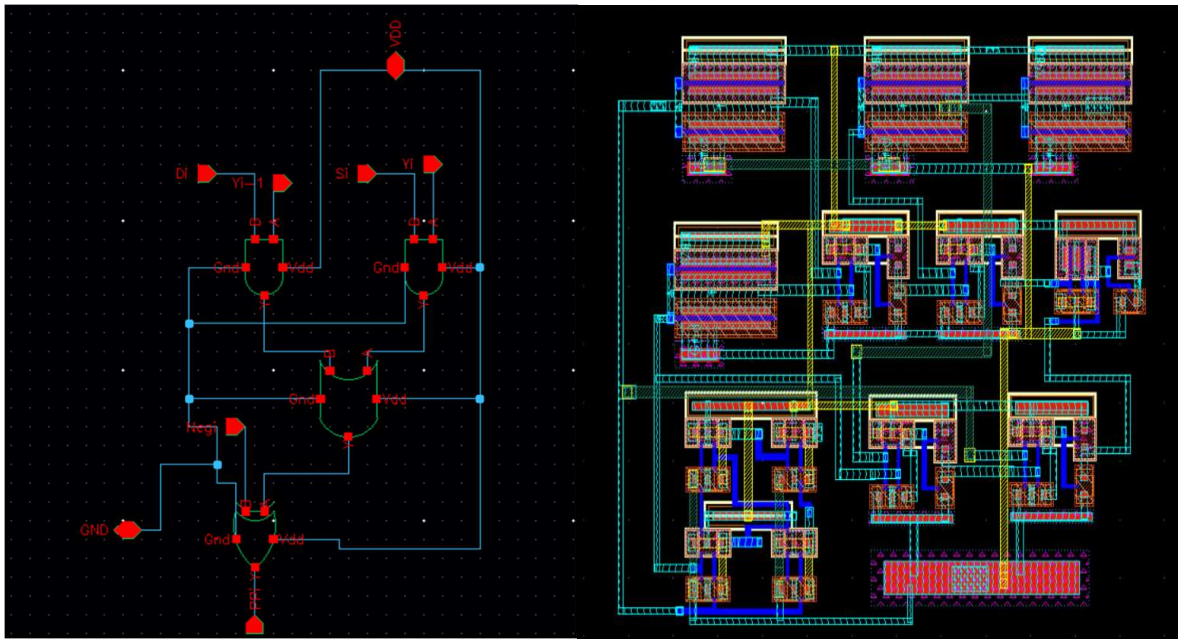


Fig. Booth Decoder schematic and layout

Encoder-Decoder

This is the combined block of a single booth encoder and 8-bit decoder.

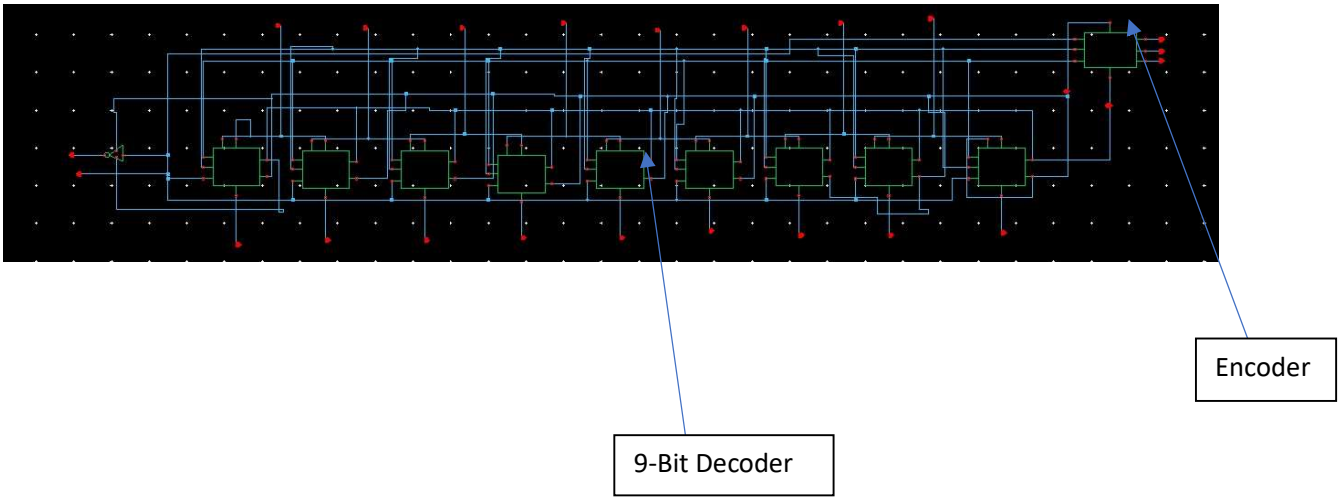
We have used 4 encoder-decoder pair in our assignment.

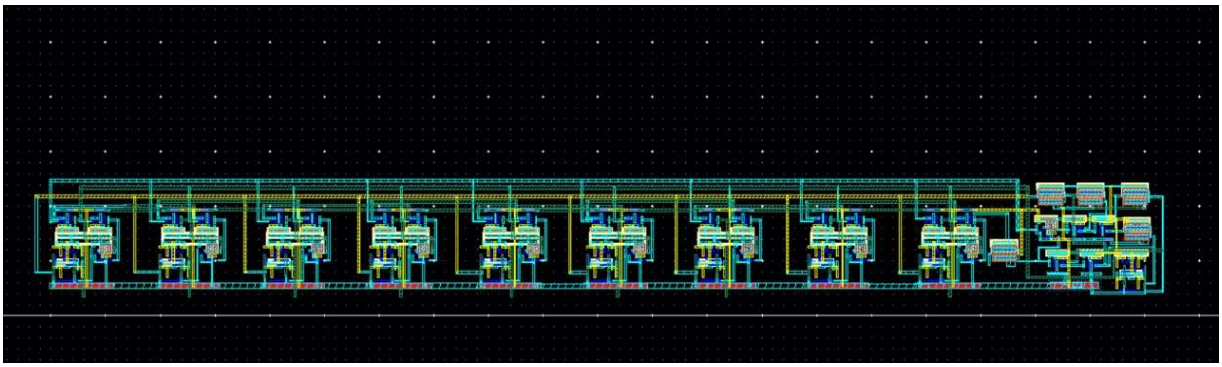
As there are four partial product stages for radix-4 booth multiplication of 8-bit numbers.

This is the combined block of a single booth encoder and 8-bit decoder.

We have used 4 encoder-decoder pair in our assignment.

As there are four partial product stages for radix-4 booth multiplication of 8-bit numbers.



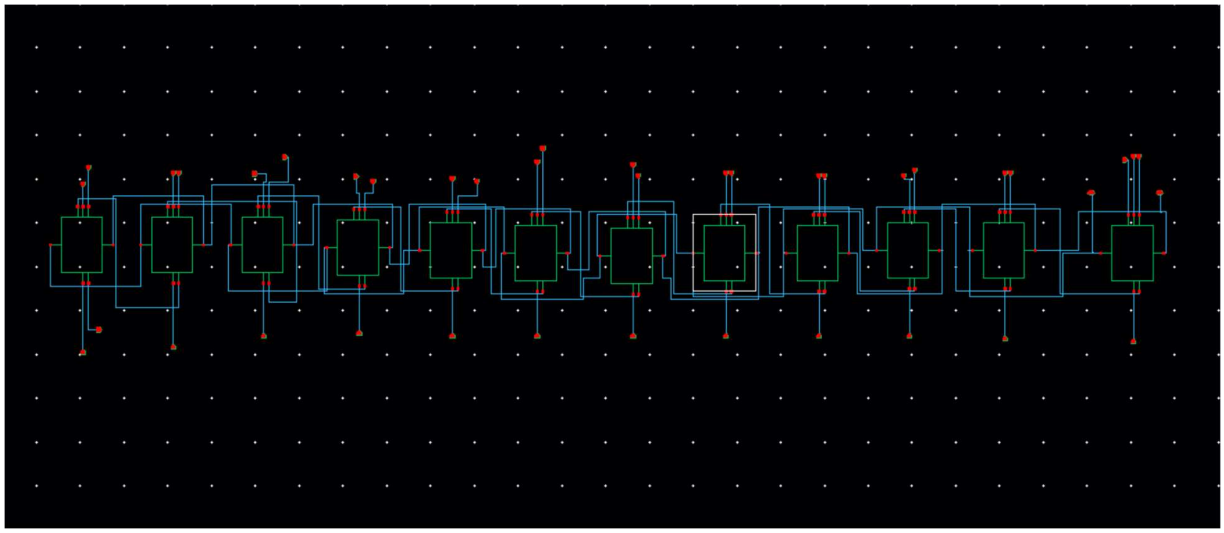


ENCODER-DECODER schematic and layout

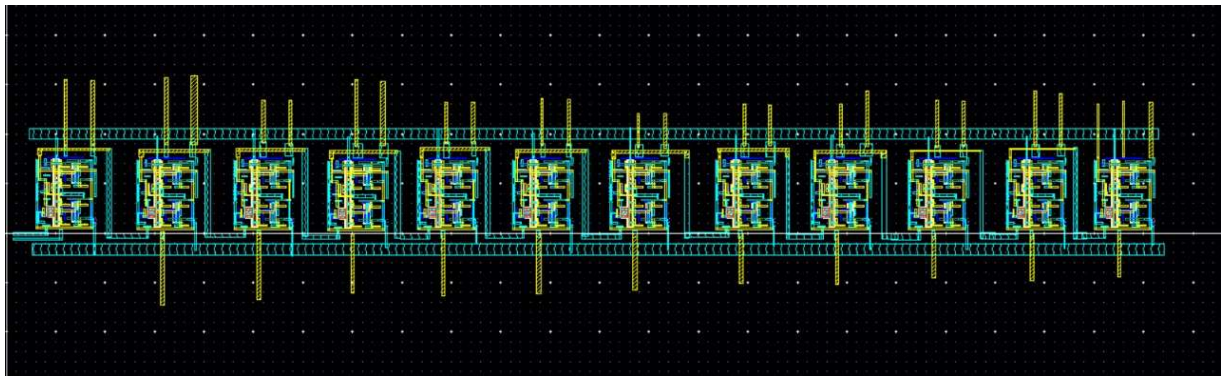
12-Bit Ripple Carry Adder

This is the schematic of a 12 bit ripple carry adder constructed using 12 full-adder connected in parallel fashion, each of the carry output of a full-adder is the carry input to the next full-adder.

The partial products generated from decoder goes into the ripple carry adder for final summation and the multiplication result generates.

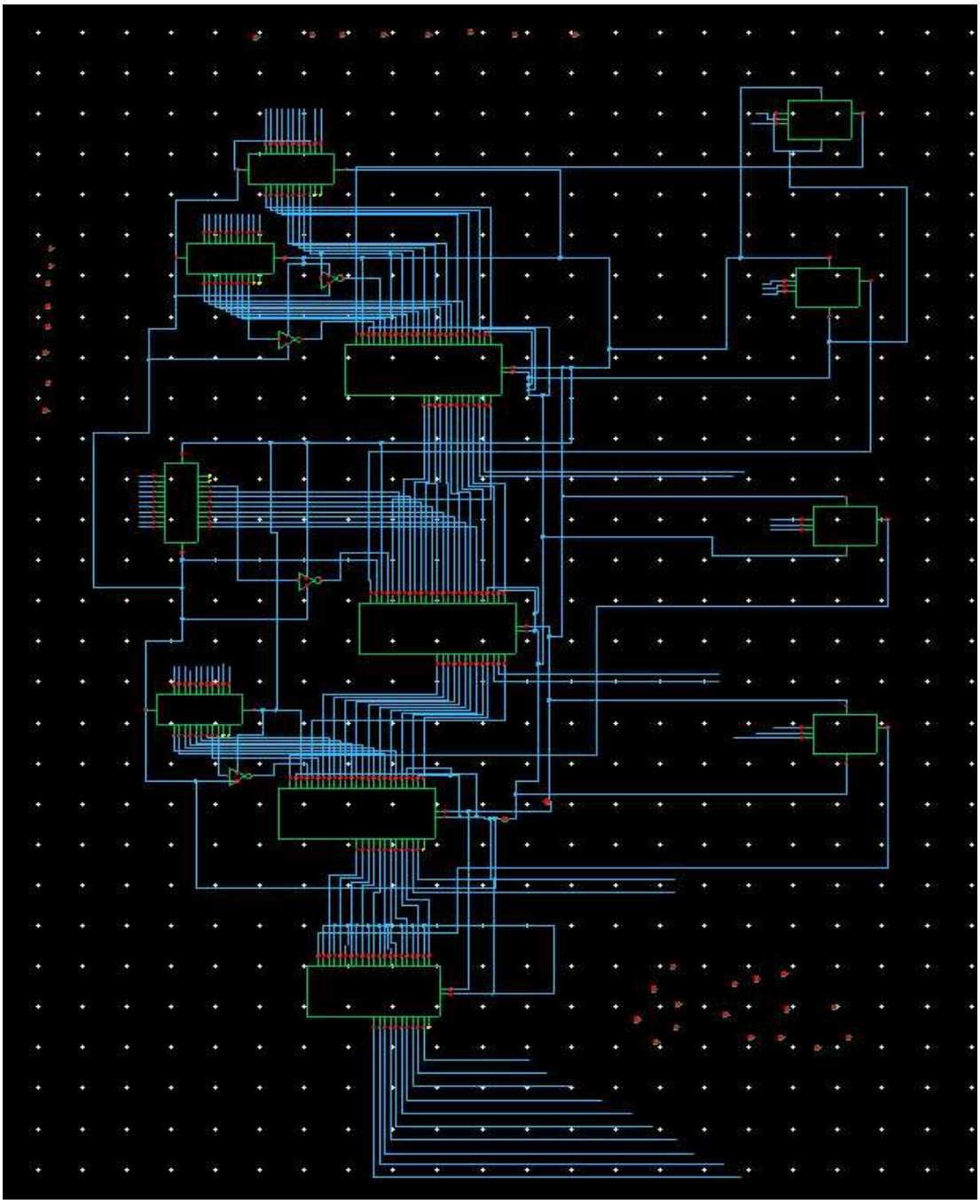


Schematic of 12-bit Adder

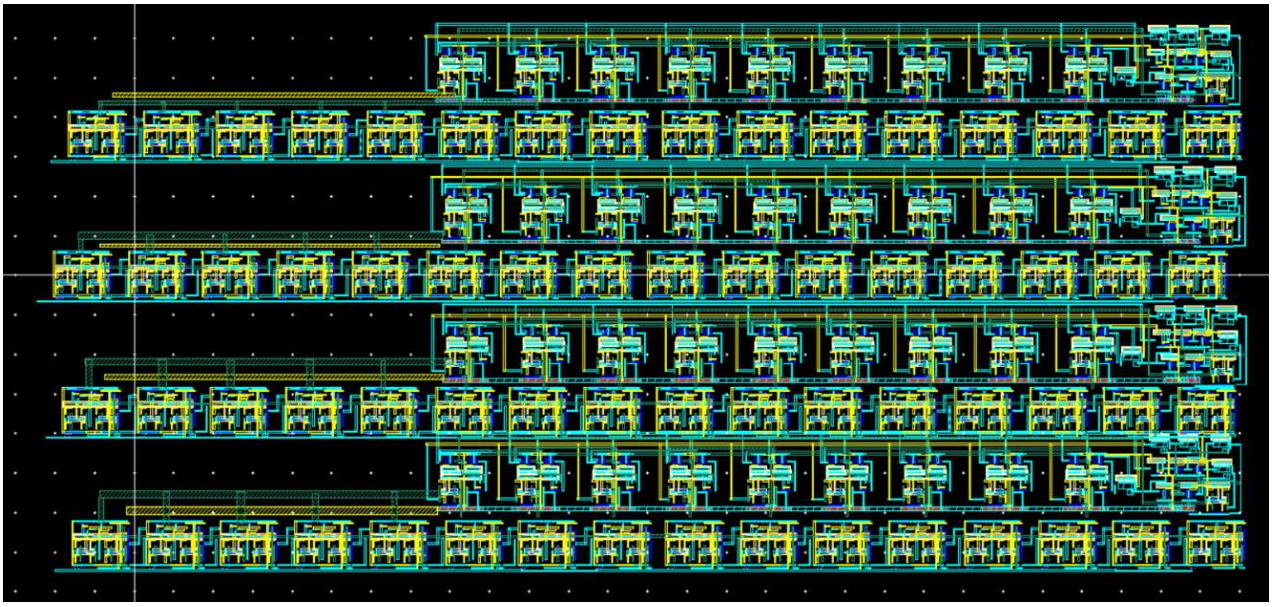


Layout of 12-bit Adder

8x8 Modified Booth Multiplier



Schematic of 8X8 BIT BOOTH MULTIPLIER



Layout of 8x8 BIT BOOTH MULTIPLIER

1. Design parameters

N/A

2. Verification Strategy

4.1 Objectives

Aim for minimum possible area with rectangular shaped layout and please Clear the DRC errors, verify the LVS and perform a PEX analysis. you need to put screenshots for that . Report the comparison of post-layout extracted circuit with the original schematics.

4.2 Tools and Version

Cadence Virtuoso, Mentor, and Spectre are all prominent software tools used in the field of electronic design automation (EDA) for integrated circuit (IC) design. Here's a brief description of each:

1. Cadence Virtuoso:

- Cadence Virtuoso is a comprehensive EDA tool suite developed by Cadence Design Systems. It is widely used for designing and verifying complex analog, digital, and mixed-signal ICs. Virtuoso provides a range of tools and capabilities for schematic capture, layout design, simulation, and physical verification.

- Key features of Cadence Virtuoso include a user-friendly graphical interface, support for advanced semiconductor process technologies, integrated design rule checking (DRC), layout versus schematic (LVS) verification, and powerful simulation capabilities. It is a favored tool among IC designers for its versatility and extensive library of design components.

2. Mentor (now part of Siemens EDA):

- Mentor Graphics, now part of Siemens EDA, offers a suite of EDA tools for various aspects of IC design, including digital and analog/mixed-signal design. One of its popular tools is the Mentor Graphics tool suite, which includes tools like Calibre for physical verification and design rule checking (DRC).

- Calibre is known for its accuracy and reliability in ensuring that IC layouts comply with semiconductor foundry rules. It plays a crucial role in the fabrication of integrated circuits, helping designers avoid costly manufacturing errors.

3. Spectre:

- Spectre is a simulation tool developed by Cadence Design Systems for analyzing and simulating analog, digital, and mixed-signal circuits. It is commonly used for transistor-level circuit simulation, which is essential for verifying the functionality and performance of IC designs.

- Spectre supports various simulation techniques, including transient analysis, AC analysis, and DC analysis. It is capable of handling complex circuit topologies and models the behavior of semiconductor devices accurately. Designers use Spectre to predict how their circuits will perform under different operating conditions and to identify potential issues early in the design process.

In summary, Cadence Virtuoso is a comprehensive IC design suite, Mentor offers tools like Calibre for physical verification, and Spectre is a simulation tool for analyzing the electrical behavior of integrated circuits. These tools are essential for designing and validating ICs in the semiconductor industry.

4.3 Checking mechanisms

Design Rule Checking (DRC) and Layout Versus Schematic (LVS) are essential steps in the integrated circuit (IC) design and verification process. They are used to ensure the correctness and manufacturability of semiconductor layouts. Let's take a closer look at each of them:

1. Design Rule Checking (DRC):

- DRC is a process that checks whether the layout of an integrated circuit adheres to a set of predefined design rules specified by the semiconductor foundry or design team. These design rules encompass parameters like minimum feature sizes, spacing, overlap, and other geometric constraints.

- DRC tools analyze the layout data and compare it to the specified design rules, flagging any violations. Common violations include metal traces too close together, insufficient spacing between transistors, or irregularities in shapes.

- DRC helps ensure that the IC layout can be manufactured without defects that might lead to electrical or performance issues. It's a critical step in preventing costly manufacturing errors.

2. Layout Versus Schematic (LVS):

- LVS is a process that verifies whether the physical layout of an integrated circuit matches the intended schematic or logical representation. It checks that the connections, device sizes, and component placements in the layout correspond accurately to the design described in the schematic.

- LVS tools read both the layout data and the schematic data and compare them. They look for discrepancies such as missing or extra components, incorrect connections, or incorrect transistor sizes.

- LVS is crucial for ensuring that the manufactured IC will function as intended based on the original design. It helps identify discrepancies that could lead to functional errors or performance issues.

The process of checking DRC and LVS typically involves the following steps:

1. Layout Extraction: The layout data is extracted from the IC design database and prepared for analysis.

2. DRC Check: The layout is checked against the design rules to identify any violations. Violations are flagged for review and correction.

3. LVS Check: The layout is compared to the schematic to ensure they are consistent. Any discrepancies are identified and reported.

4. Error Resolution: Designers review the DRC and LVS results and make necessary corrections to the layout or schematic to resolve any issues.

5. Reiteration: Steps 1-4 may need to be repeated until both DRC and LVS checks pass successfully without errors or discrepancies.

6. Final Verification: Once DRC and LVS checks pass, the layout is considered ready for manufacturing, knowing that it adheres to design rules and accurately reflects the schematic.

These checks are fundamental in ensuring the quality, manufacturability, and reliability of integrated circuits, from small-scale digital chips to complex mixed-signal or analog designs.

3. Functional Checklist

NA

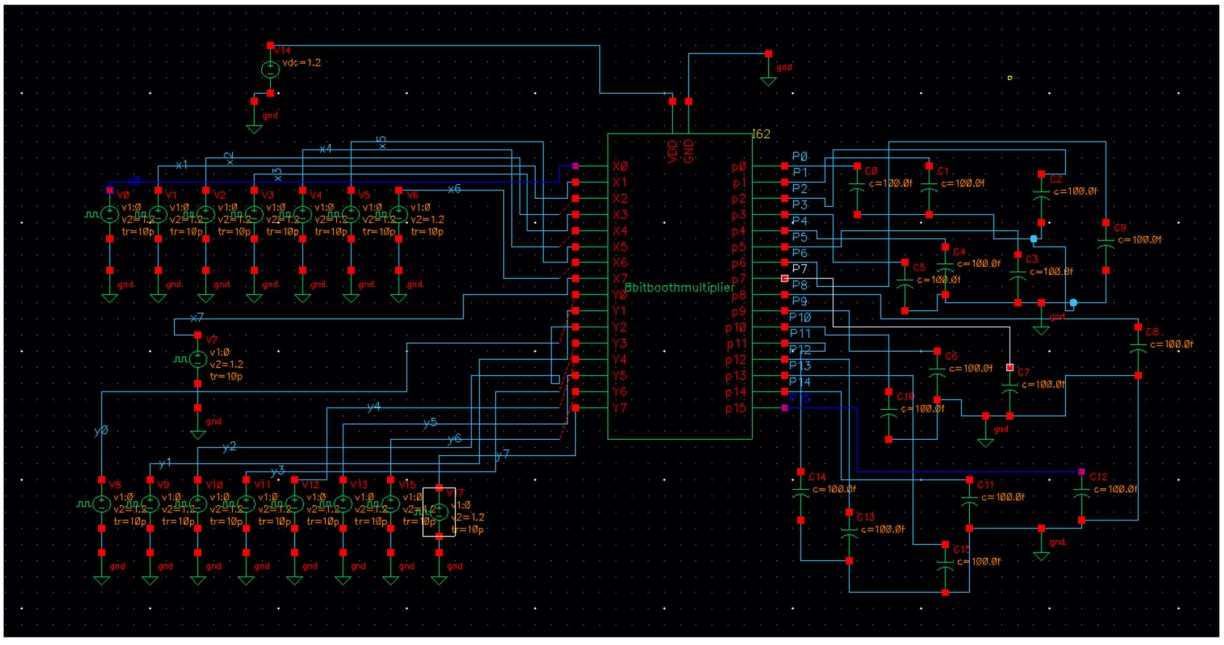
1. Testbench-

A testbench consists of simulation models, test vectors, and stimulus generation mechanisms that are used to simulate the behavior of your IC design.

1.1 Overview

Load of Every input is connected to 0.1 pF of capacitor as It helps capture the transient response, time constants, and frequency-dependent behavior of the circuit

1.2 Architecture



2. Test Specification

9-Corner Analysis-

TEMPERATURE	TT (delay in ns)	FF (delay in ns)	SS (delay in ns)
27	3.373	2.547	4.6037
-40	3.1169	2.2164	4.285
+140	3.7783	2.8975	5.106

3. Design Microarchitecture

NA

8. Physical hierarchy

NA

9. Results

9.1 Area

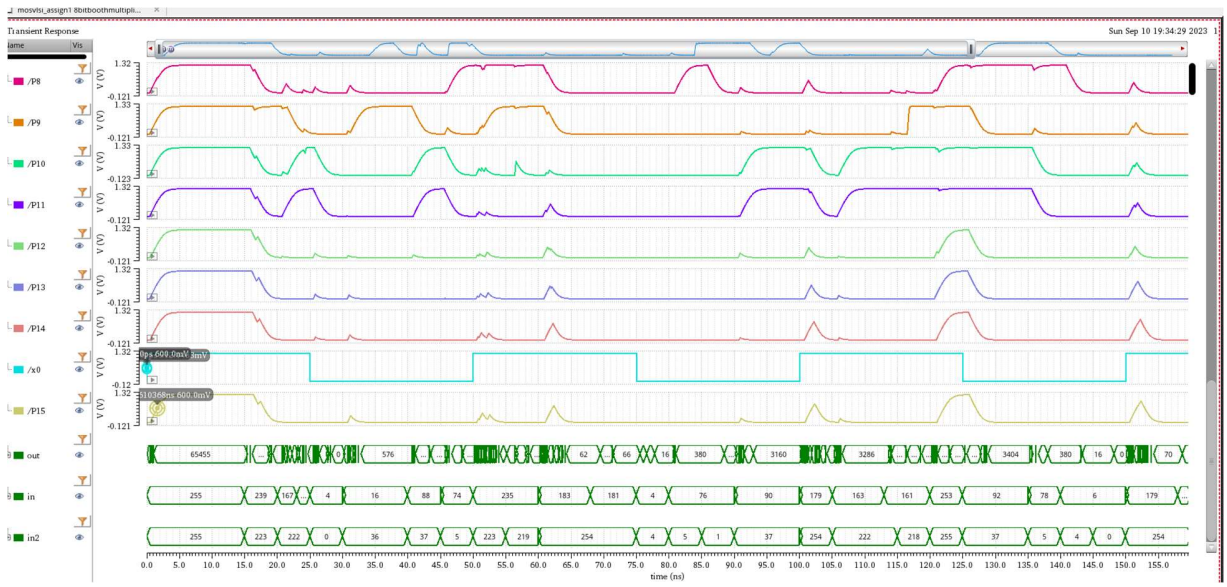
Block Name	ND-2 equivalent Gate count	% utilization	Dimension (W x L) (in um^2)	Layout Area (micro-meter square)
8x8 booth multiplier	3136	100	176x70	12320
Encoder-Decoder	252	10.85	103x10.65	1096.95
12Bit Adder	432	47.78	114.5x55.52	2578.54

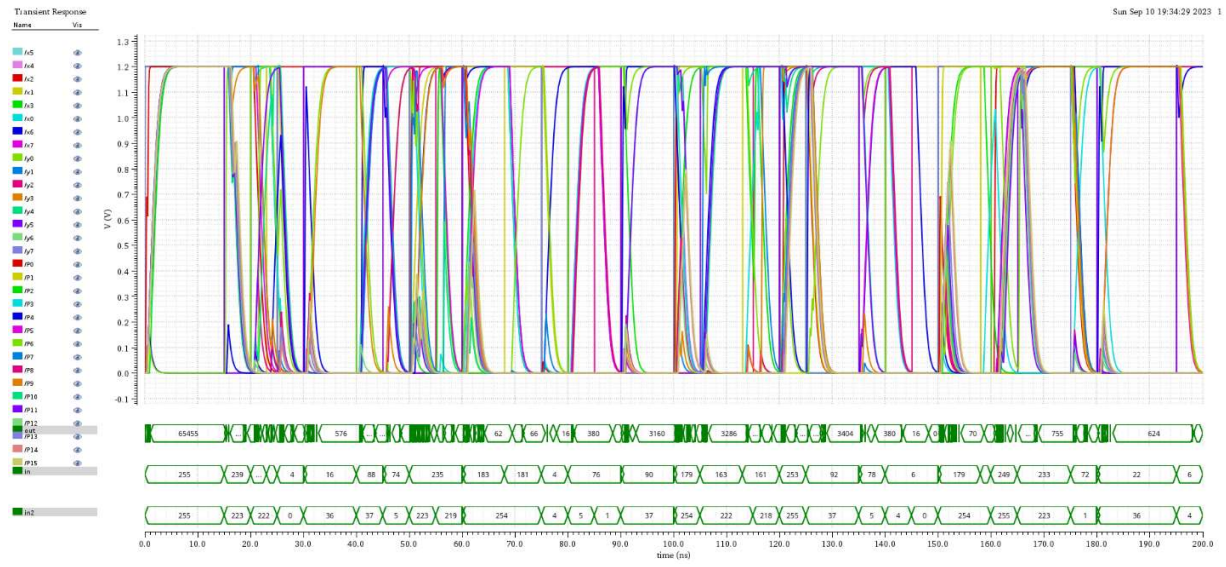
9.2 Timing

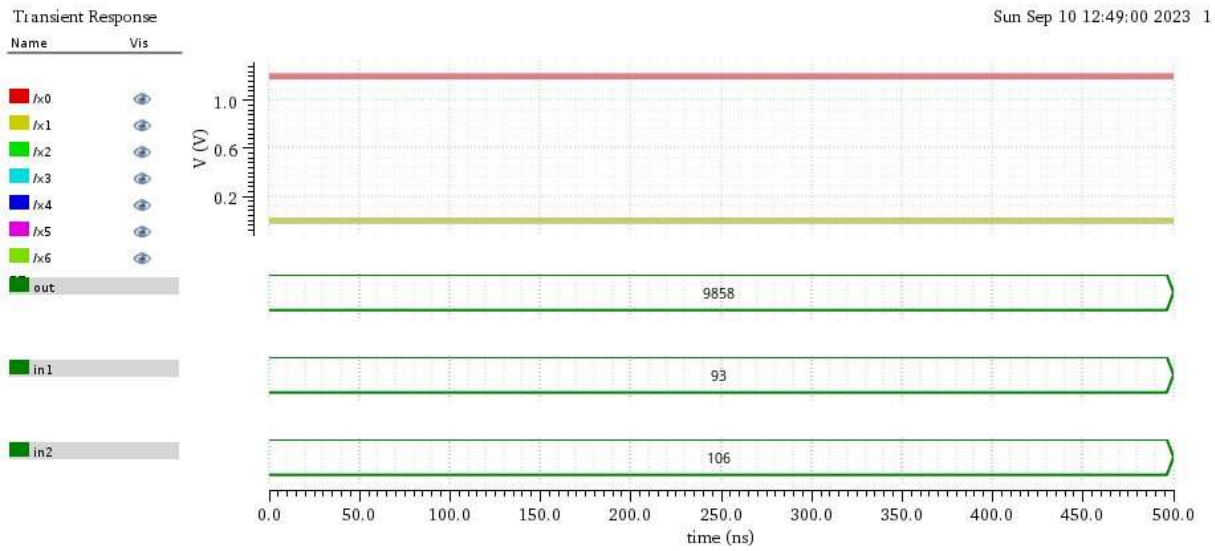
na

9.3 Testability analysis

Test signals







9.4 DRC rule violations

- 8x8 booth multiplier drc is not cleared .
- Minimum poly density over 1mm x 1mm area.
- L2.D1 not cleared

11. Bugs known at submission date

For some numbers booth multiplier gives absurd value.