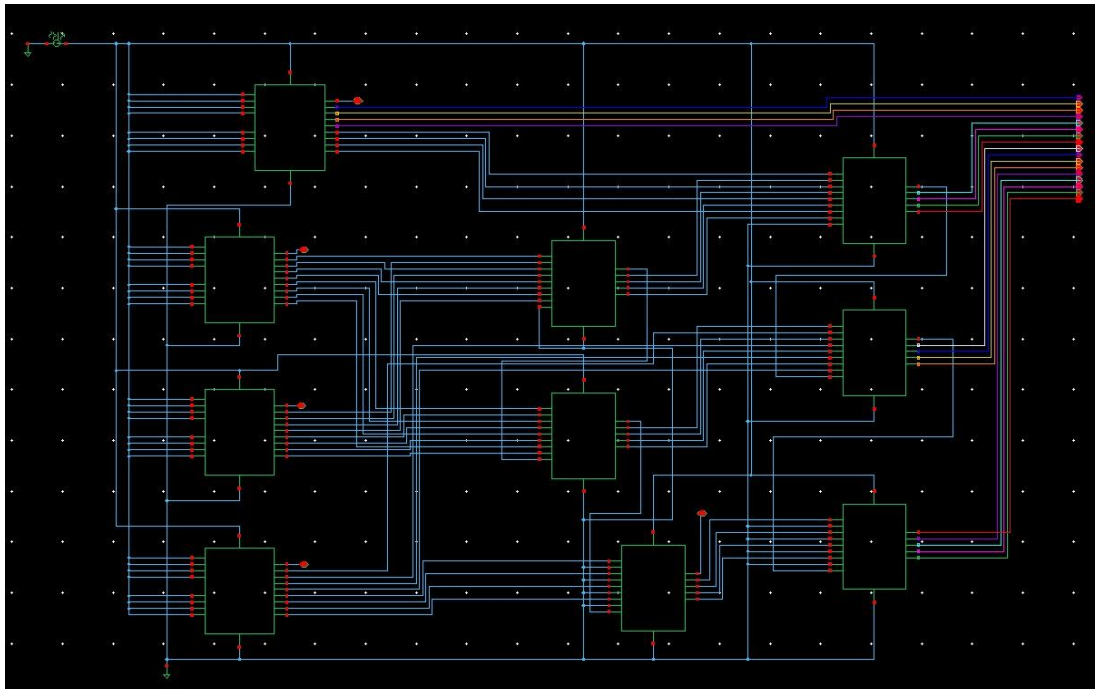
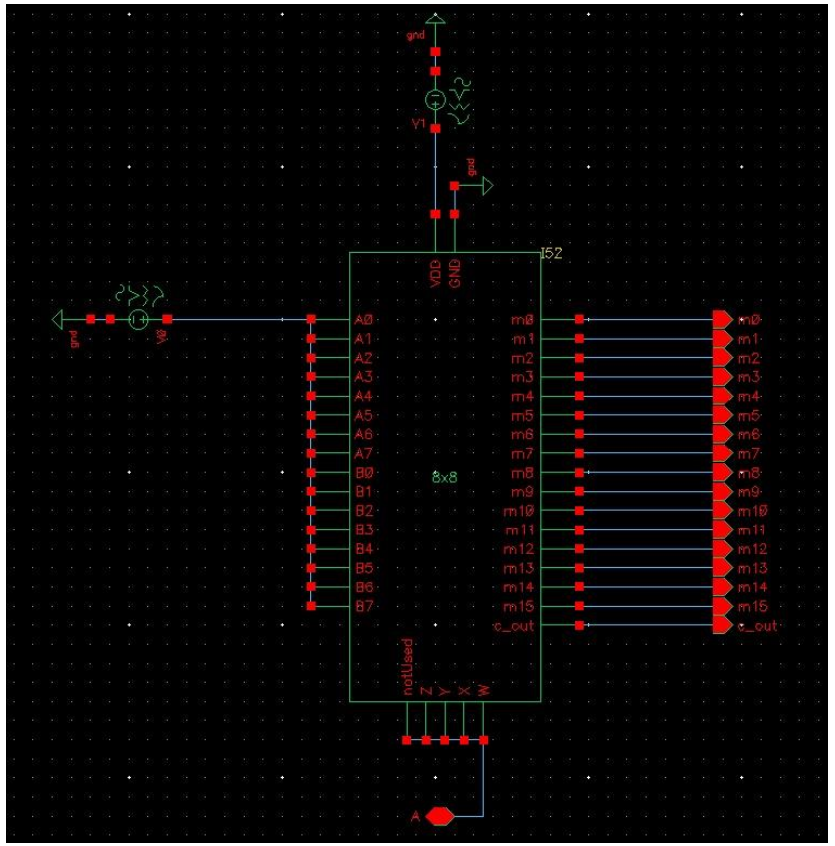
Low Power Approximate Multipliers using Compressors

- Objective: Design and implement an energy efficient multiplier using approximate compressors and encoded partial products.
- Motivation: Approximate computing reduces power and area with minimal accuracy loss.
- Block diagram of the 8x8 multiplier.





Designed by:- Deepak Kumar

Exact 4x4 Multiplier

- In the multiplication of 2 4-bit terms the product is coming in stages
- In stage 1, half adder can simply use to produce sum as (γ1) and carry (c1) for next stage.
- In stage 2, there are already 3 partial product terms present, so we require 4:2 compressor that takes c1 and 3 pp terms as input and gives sum bit (γ2) and carry (c2) for next stage.

ORIGINAL PARTIAL PRODUCT OF THE MULTIPLICATION

Stage 7	Stage 6	Stage 5	Stage 4	Stage 3	Stage 2	Stage 1	Stage 0
	$pp_{3,3}$	$pp_{3,2}$	$pp_{3,1}$	$pp_{3,0}$	$pp_{2,0}$	$pp_{1,0}$	$pp_{0,0}$
		$pp_{2,3}$	$pp_{2,2}$	$pp_{2,1}$	$pp_{1,1}$	$pp_{0,1}$	
			$pp_{1,3}$	$pp_{1,2}$	$pp_{0,2}$		
				$pp_{0,3}$			
γ_7	γ_6	γ_5	γ_4	γ_3	γ_2	γ_1	γ_0

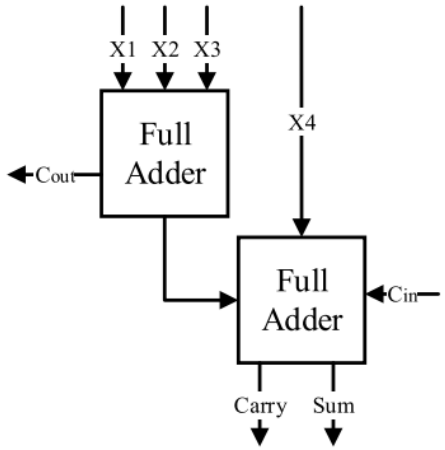
Designed by:- Deepak Kumar

Theoretical Background

- Overview of Approximate 4:2 Compressor :

a . Exact 4:2 Compressor uses XOR, AND, OR that gives high power & delay.

$$Sum = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus C_{in},$$
$$C_{out} = (x_1 \oplus x_2)\,.x_3 + \left(\overline{x_1 \oplus x_2}\right).x_1,$$
$$Carry = (x_1 \oplus x_2 \oplus x_3 \oplus x_4)\,.C_{in} + \left(\overline{x_1 \oplus x_2 \oplus x_3 \oplus x_4}\right).x_4.$$

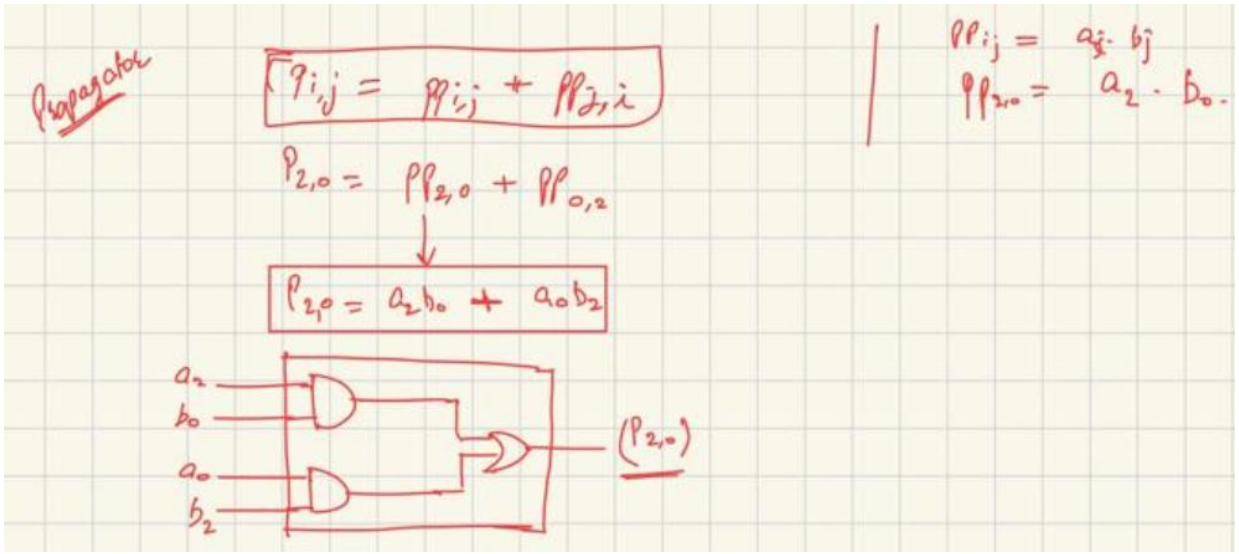


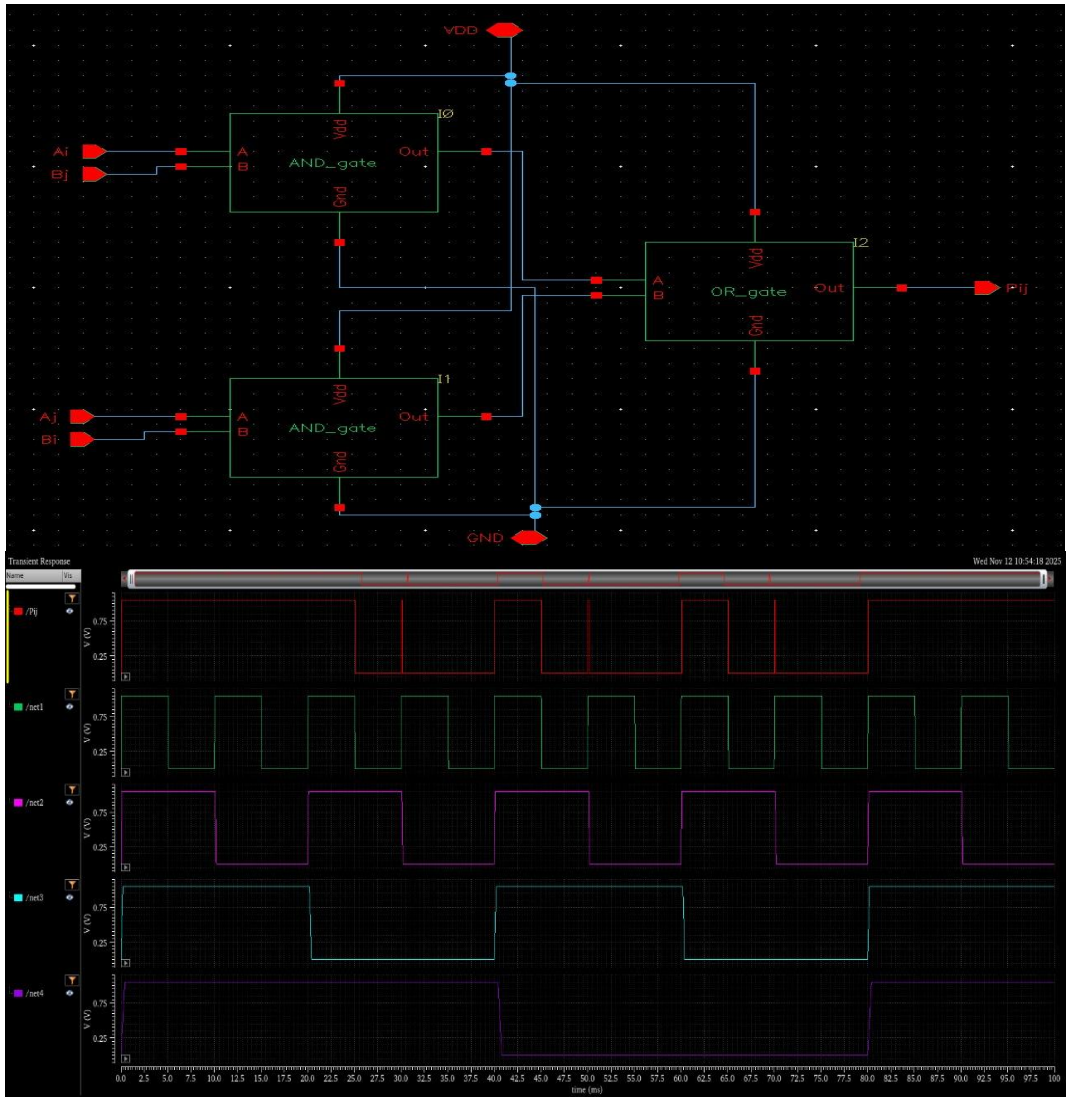
b. Modified compressor : Ignoring XOR gates(high delay) and Cout(doesn't impact much on accuracy)is our goal.

By ignoring this there are 5/7 error values in carry/sum; to incorporate this error we give the inputs to compressor using propagate and generate signals.

Propagators

It performs logical OR of two partial products.

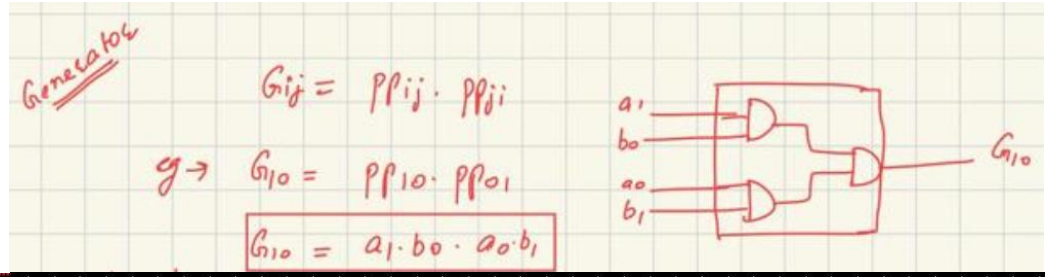


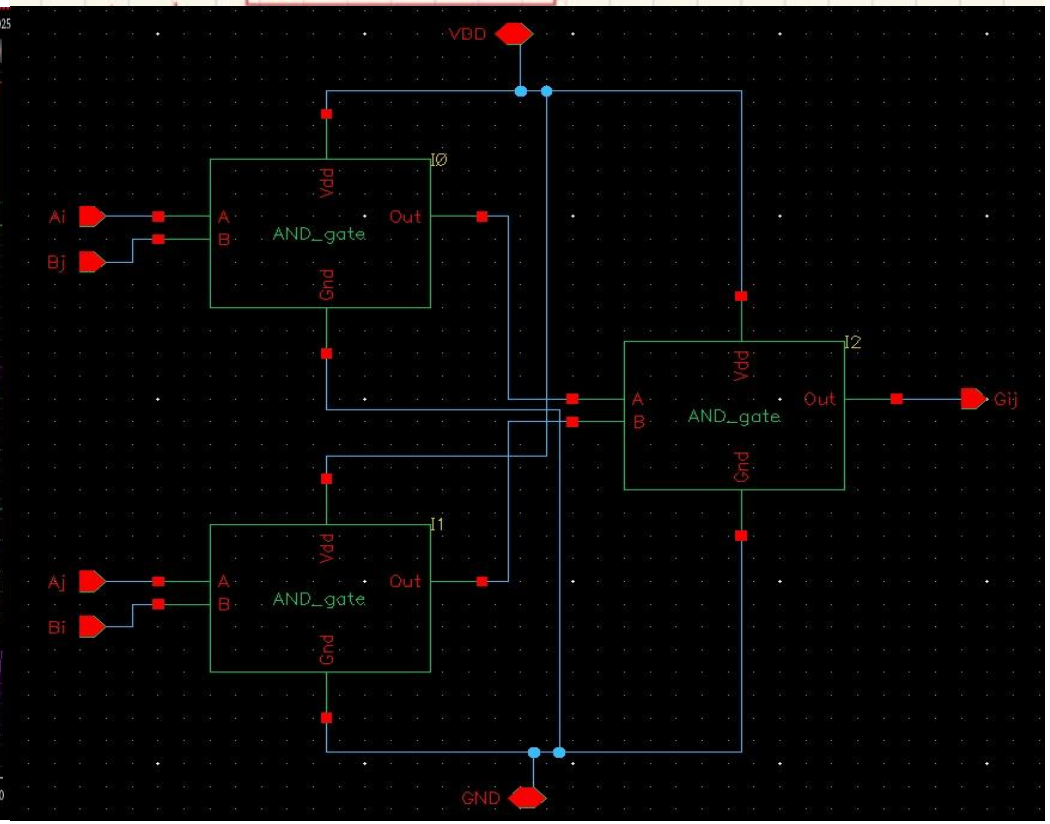


Designed by:- Deepak Kumar

Generators

It performs logical AND of two partial products.

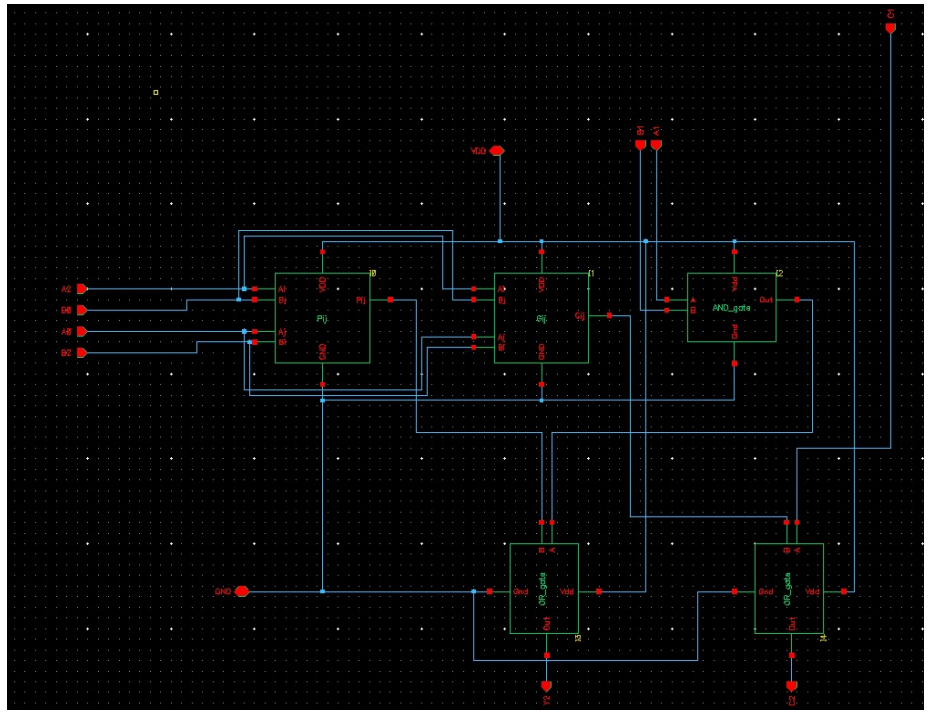




Designed by:- Deepak Kumar

Stage 2 Compressor

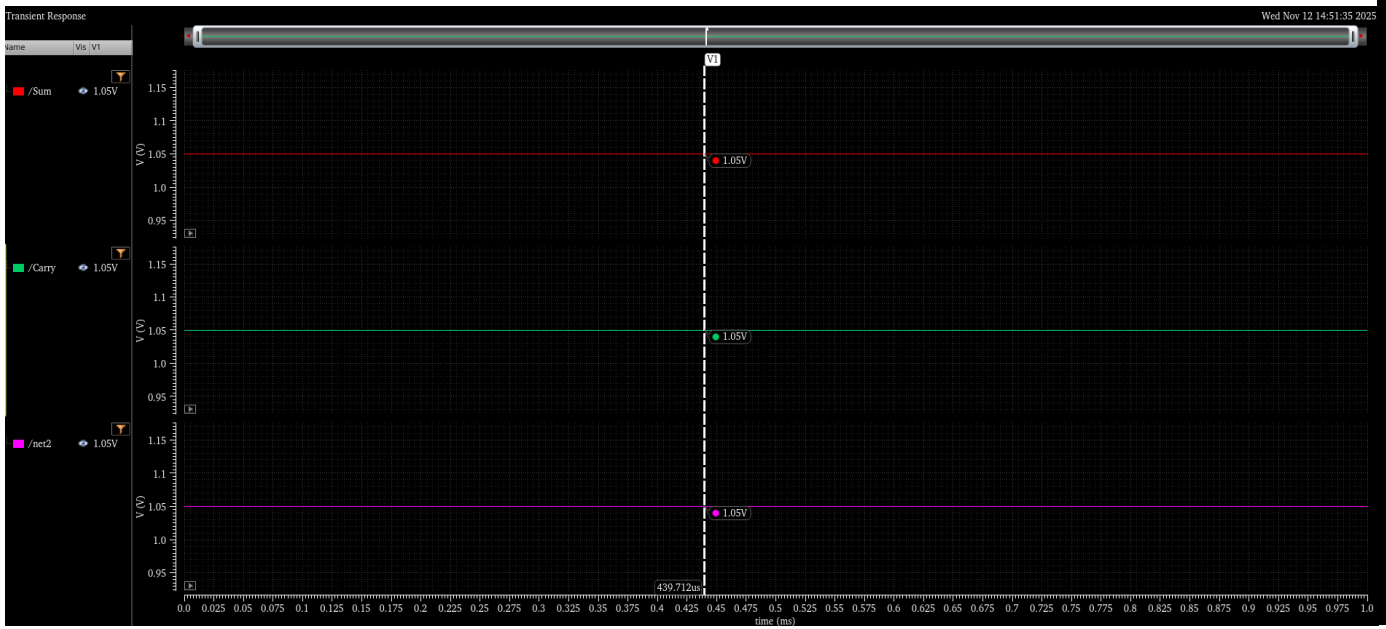
- Inputs: G_{20} , P_{20}, pp_{11} and carry c_1.
- Simplified Logic (from paper eq. 3):
- Sum=x1+x3, Carry=x2+x4
- Reduced from 5/7 faulty outputs to 2/4 faulty outputs.

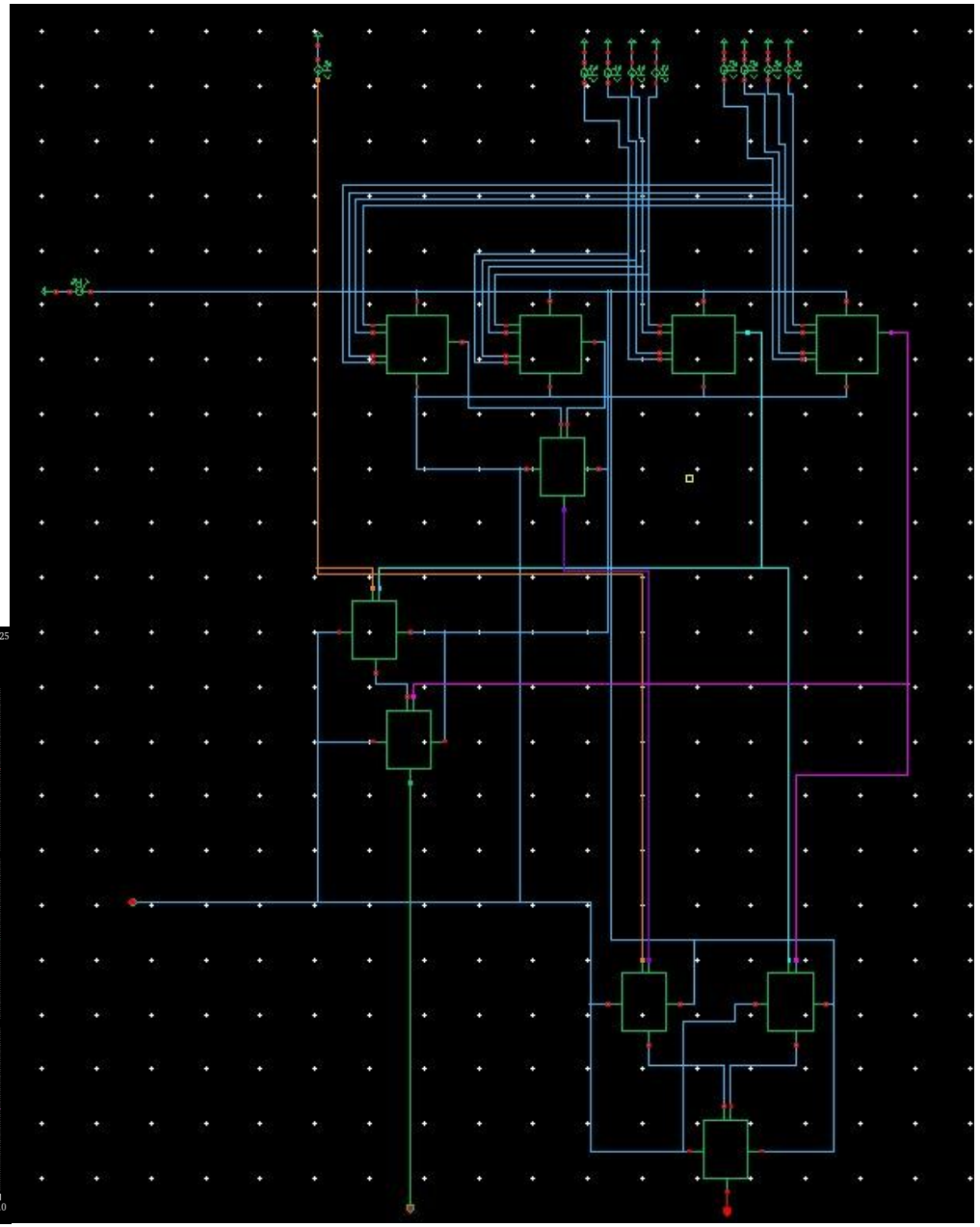


Designed by:- Deepak Kumar

Stage 3 Compressor

- Inputs: c_2, $G_{30}+G_{21}$, P_{21}, P_{30}
- Logic Simplification (from paper Eq. 5):
- Sum = x1+x3+x4, Carry = (x1.x2) + (x3.x4)
- Acts like a 5:2 compressor with reduced complexity.

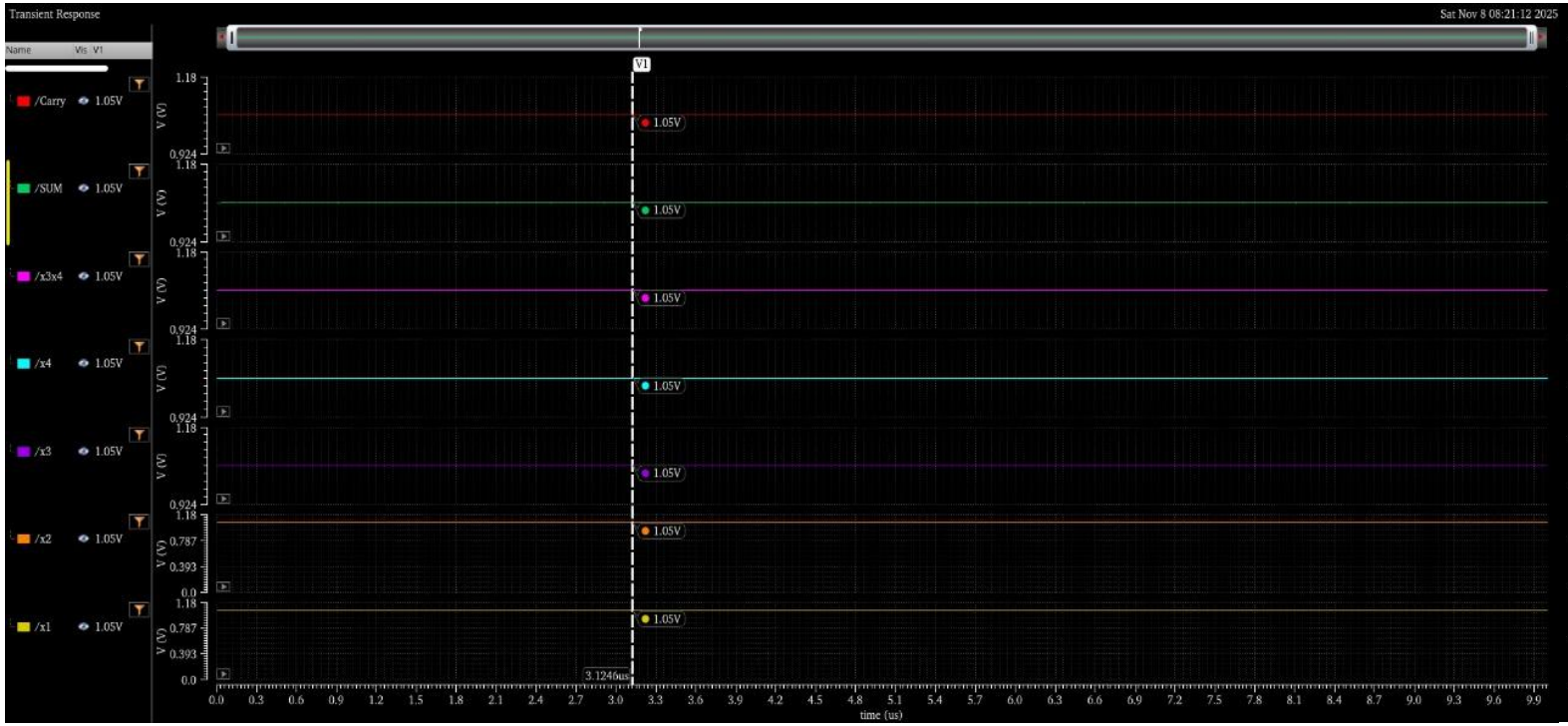


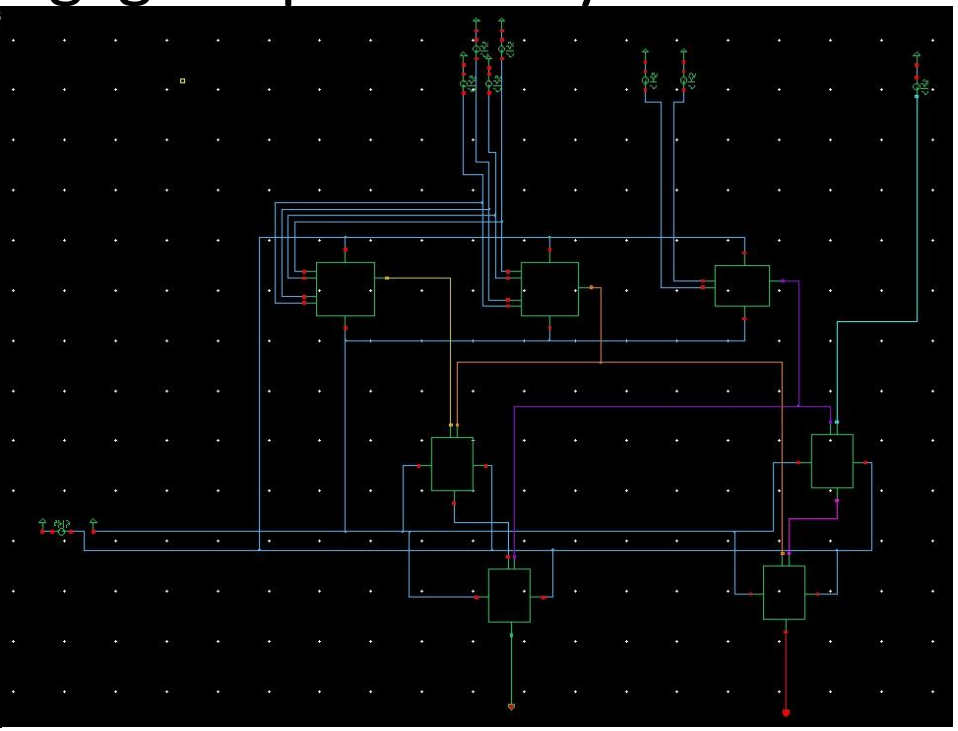


Designed by:- Deepak Kumar

Stage 4 Compressor

- Inputs: G_{31}, P_{31}, pp_{22}, and c_3.
- Simplified logic (from paper Eq. 6):
- Sum = x1+x2+x3, Carry = x2 + (x3.x4)
- Number of error cases further reduced with negligible probability.

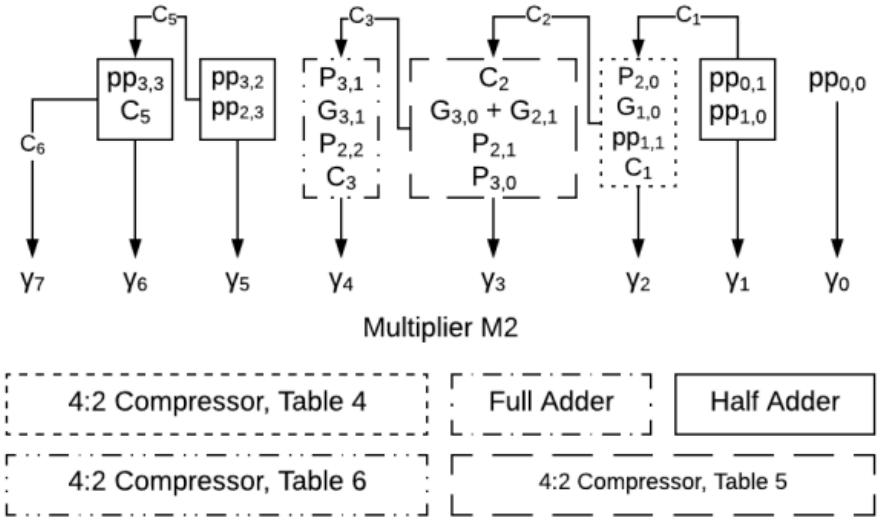


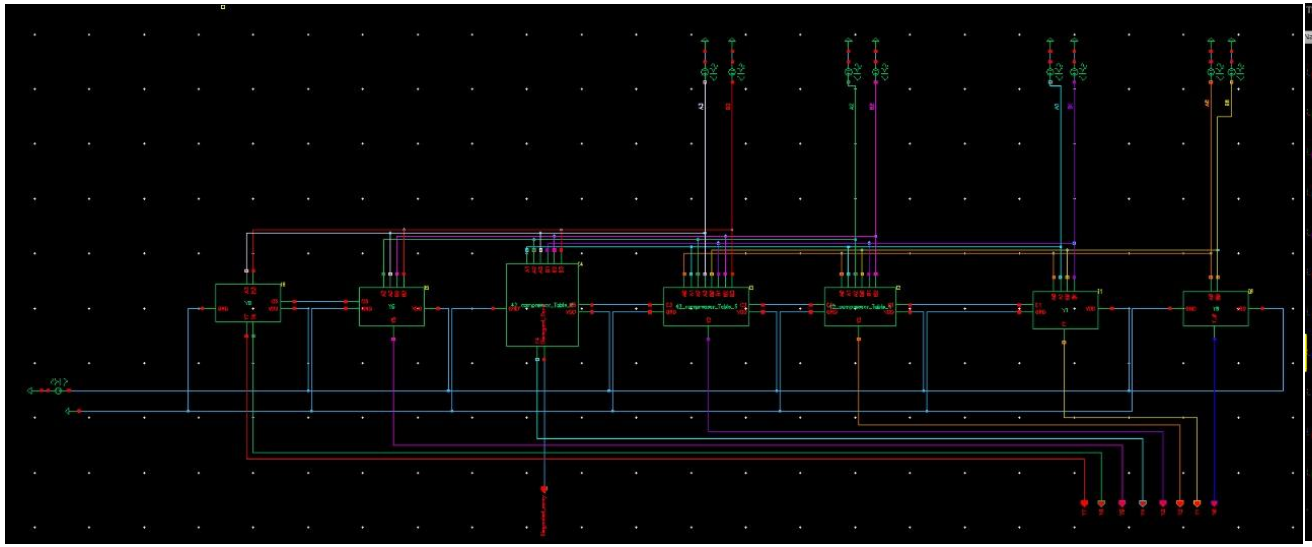


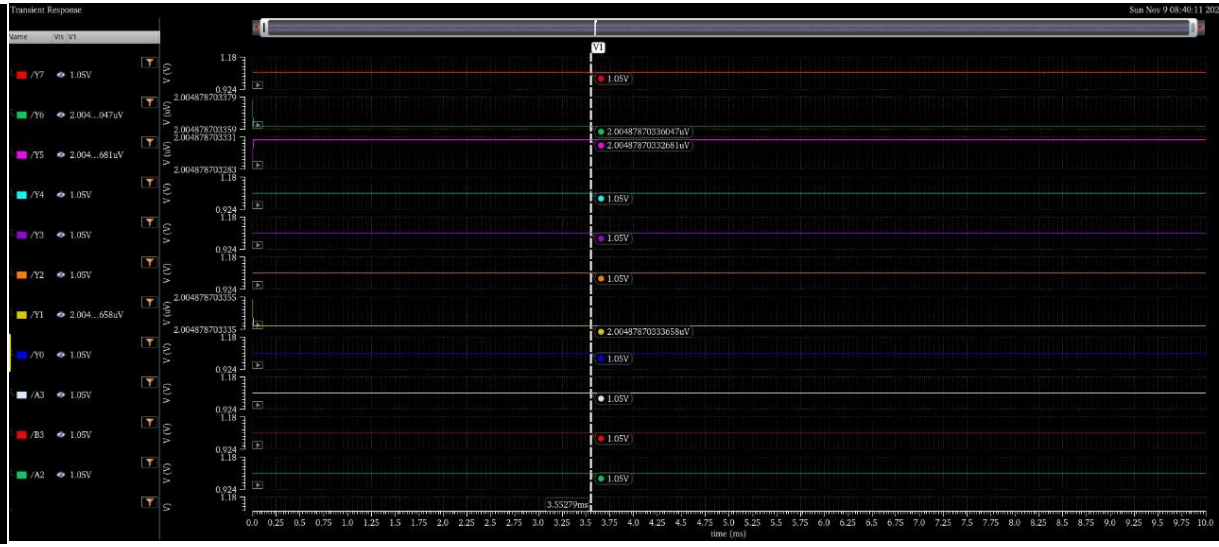
Designed by:- Deepak Kumar

Final Implementation of 4x4

- Combined the Stage 2–4 compressors and Gij/Pij generators.
- Architecture follows the paper's **M2 design** (rejecting c_4 carry).
- As speed is high in case of M2 design compared to M1(where c4 is fed into stage 5 and hence reducing the speed).





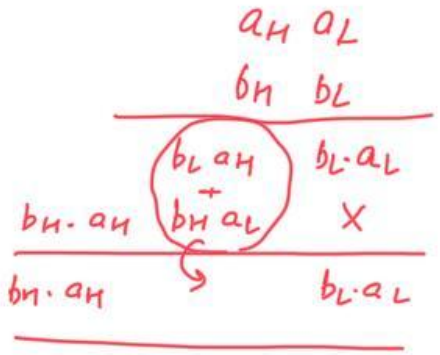


Designed by:- Deepak Kumar

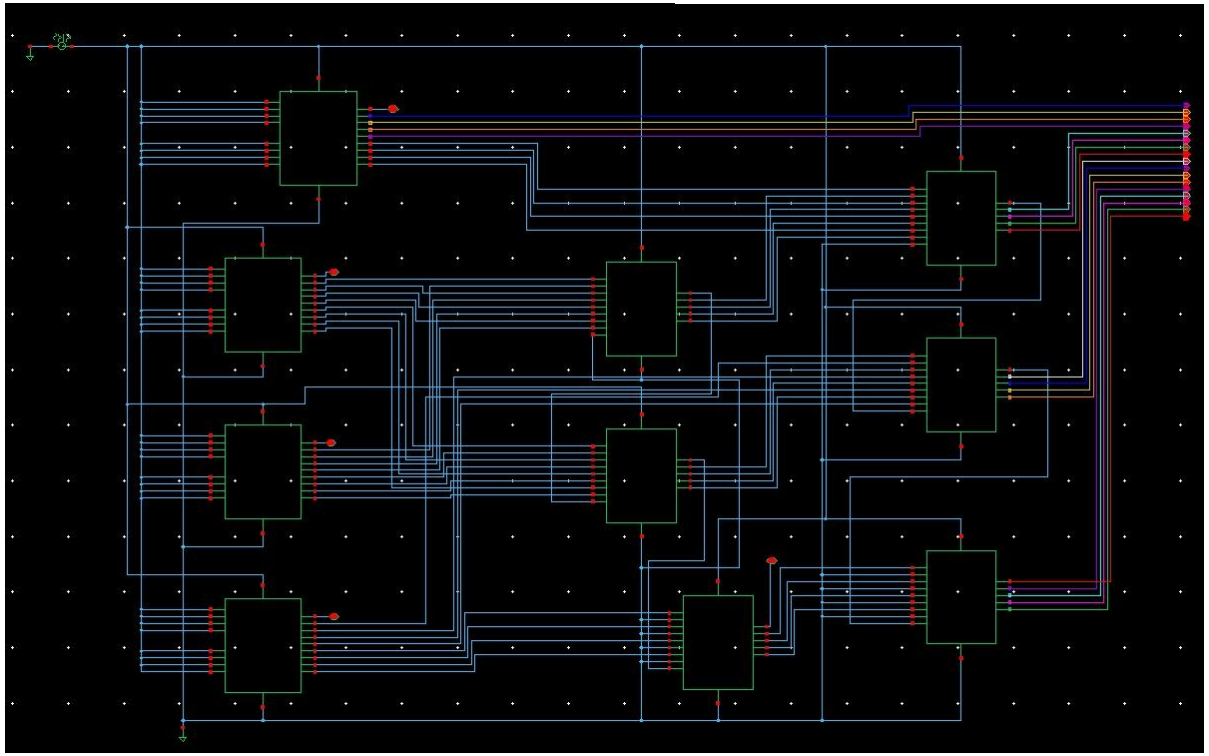
Building 8×8 Multiplier Using 4×4 Blocks

The 8-bit operands A and B are divided into two 4-bit halves:

- A = {AH, AL} → upper 4 bits (AH) and lower 4 bits (AL)
- B = {BH, BL} → upper 4 bits (BH) and lower 4 bits (BL)

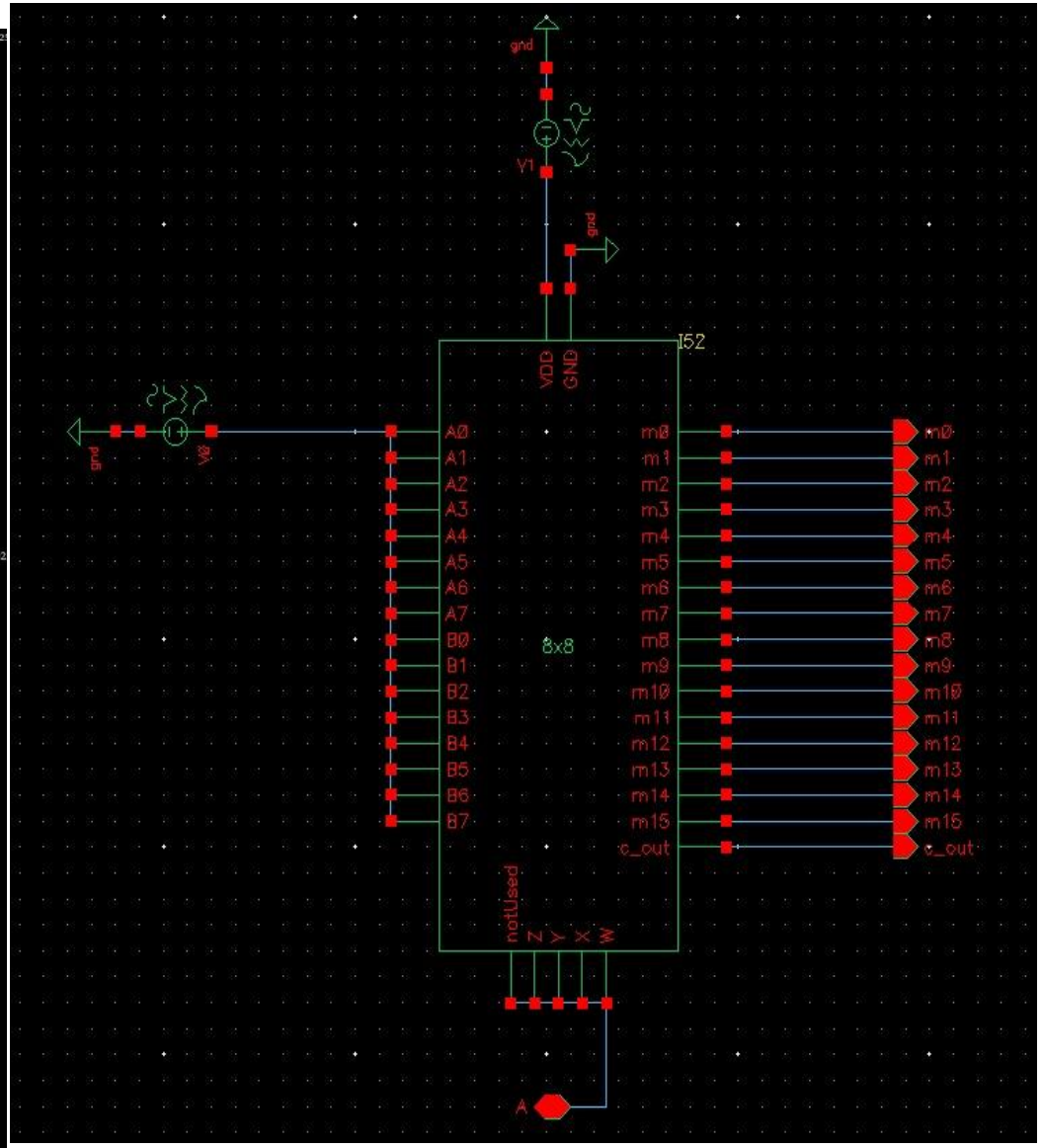


- Each combination of these halves forms **four 4×4 multipliers:**
- **P_1 = αH × βH** → Most significant partial product
- **P_2 = αL × βH**
- **P_3 = αH × βL**
- **P_4 = αL × βL** → Least significant partial product
- These four 4×4 products are then shifted and added to form the final 8×8 result:



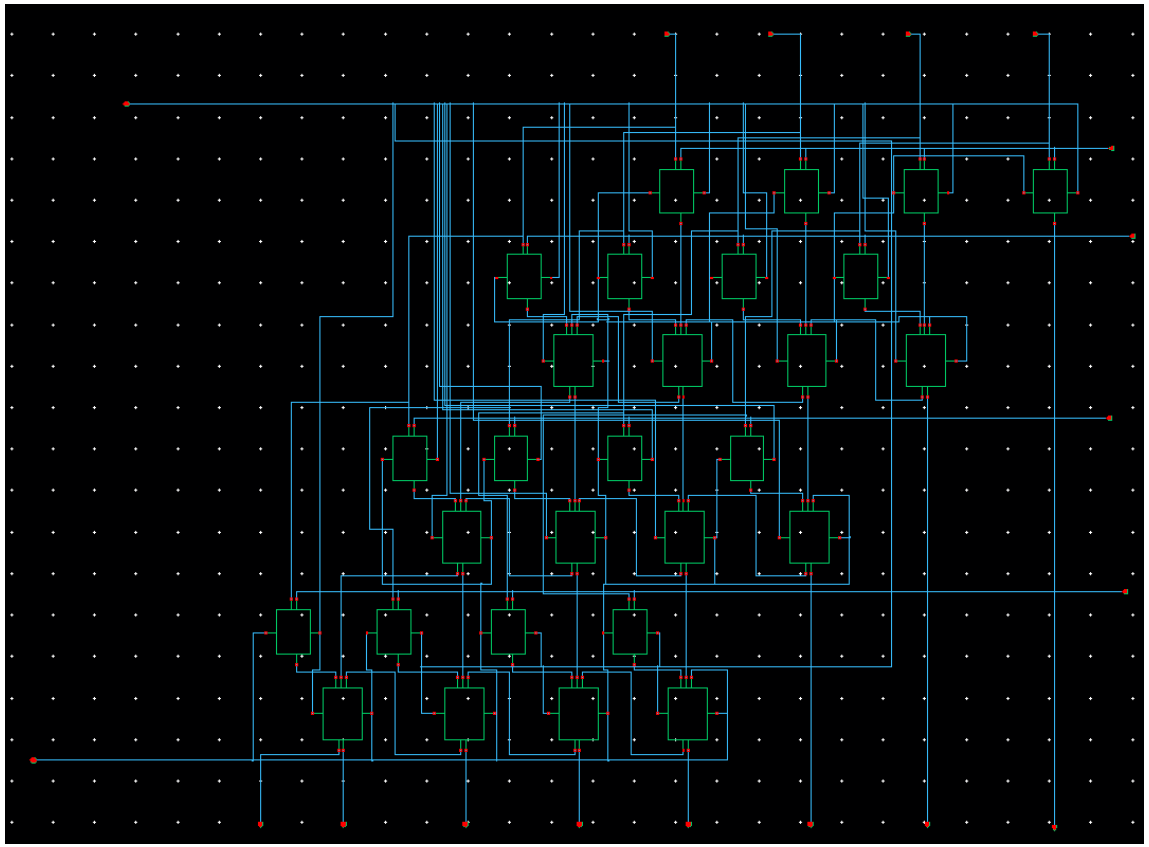
Designed by:- Deepak Kumar

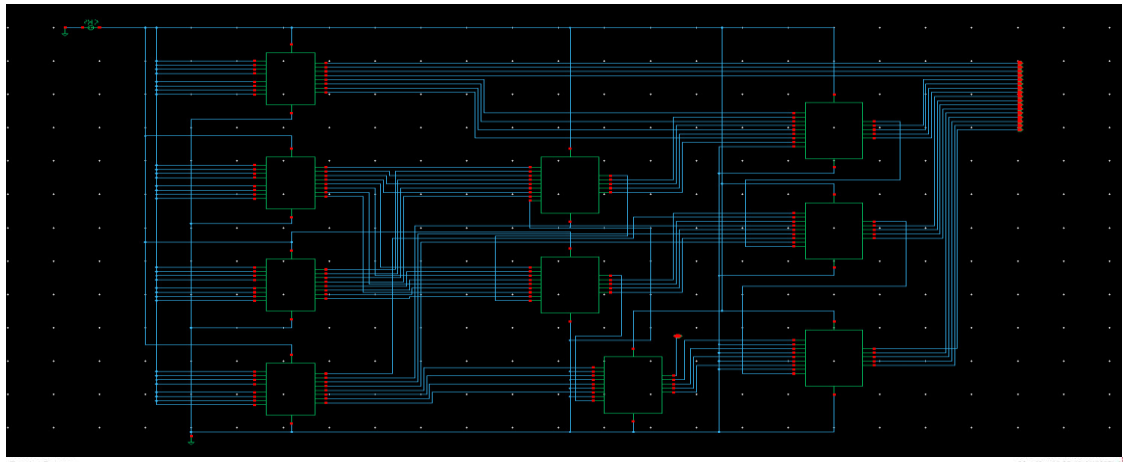
The Final Block

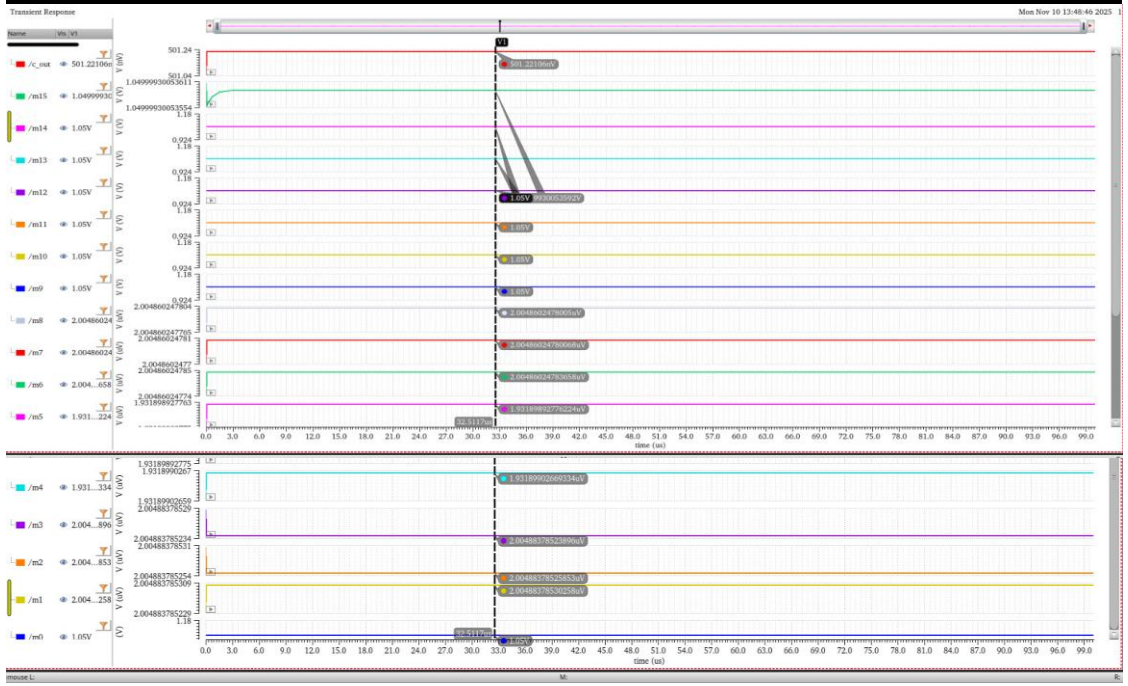


Designed by:- Deepak Kumar

Exact 8x8 Multiplier







Designed by:- Deepak Kumar

Comparison Table

All the multipliers were tested with 8bit ones (255) as A and B.

Exact	Carry Disregard Approximate Multiplier	Low Power Approximate Multipliers using Compressors	Hybrid Parallel-Product High Performance Approximate Recursive Multipliers
65025	40661	45373	62241
0111111100000001	0100111101010101	0101100010011101	0111100110010001
Error Distance	6	9	4

Designed by:- Jay, Deepak, Vishwas

Performance Comparison Table

We tested these multipliers for many different inputs and took average power and delay of them..

8x8 Multiplier	Average Delay	Average Power
Exact Multiplier	5.9986775 ns	477.63918 µW
Carry Disregard Approximate Multiplier	**5.7585925 ns**	232.48725 µW
Low Power Approximate Multipliers using Compressors	5.802965 ns	**228.0562 µW**
Hybrid Parallel-Product High Performance Approximate Recursive Multipliers	5.916225 ns	289.4195 µW

Designed by:- Jay, Deepak, Vishwas