RV College of Engineering®
Autonomous institution affiliated to Visvesvaraya Technological University, Belagavi
Approved by AICTE, New Delhi, Accredited by NAAC, Bengaluru.
Go, change the world

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

Digital VLSI Design Assignment

(18EC54)

Digital Multiplier Design

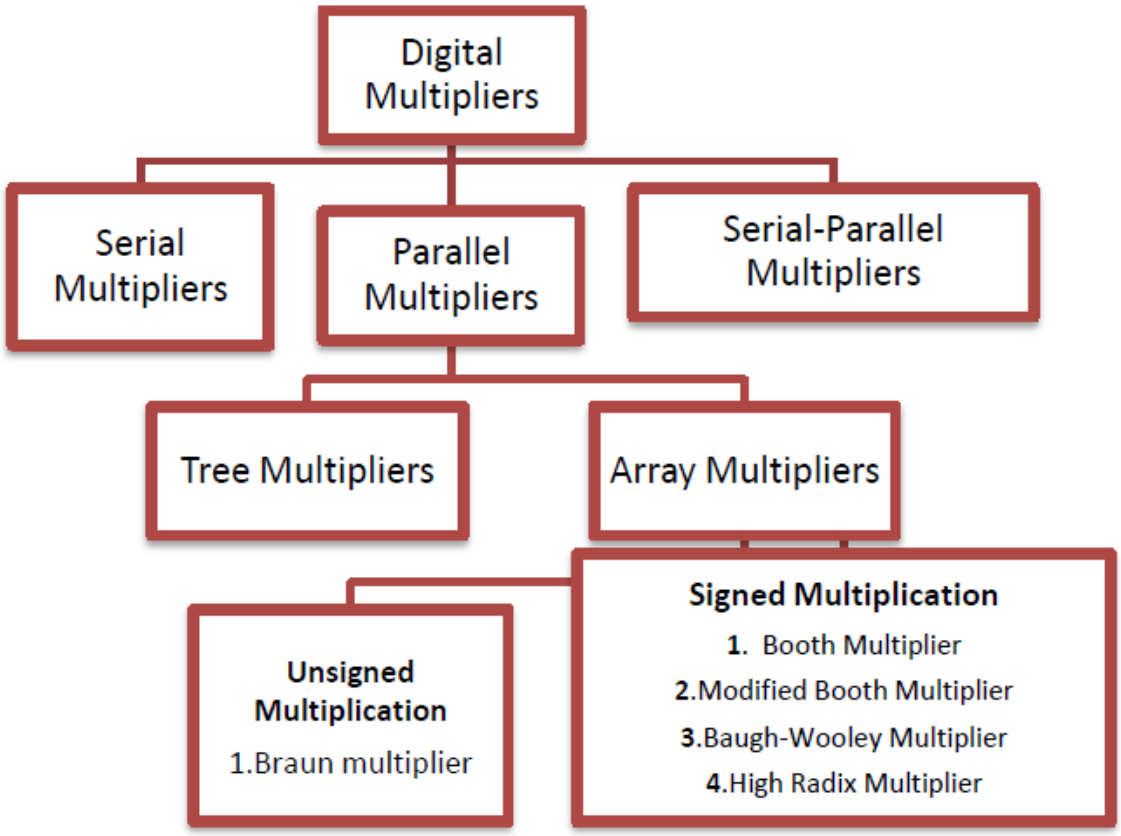
Submitted by,

Student name	USN
NEHA DAOO	**1RV18EC096**

Faculty Incharge:
Dr.Srividya P
Assistant Professor,
Department of Electronics & Communication Engineering
RVCE, Bangalore

2020-2021

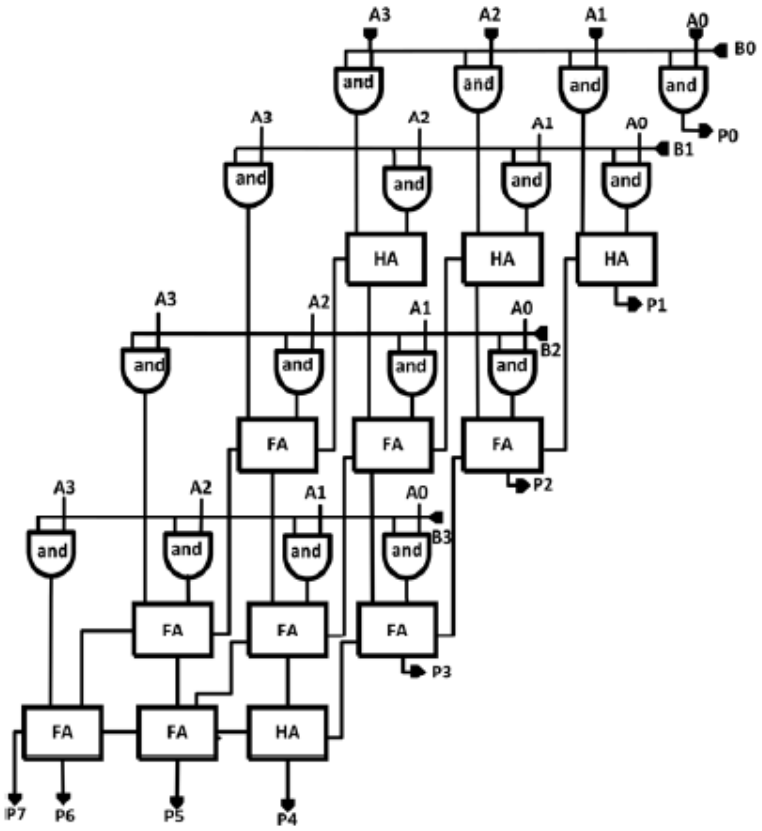
INTRODUCTION:



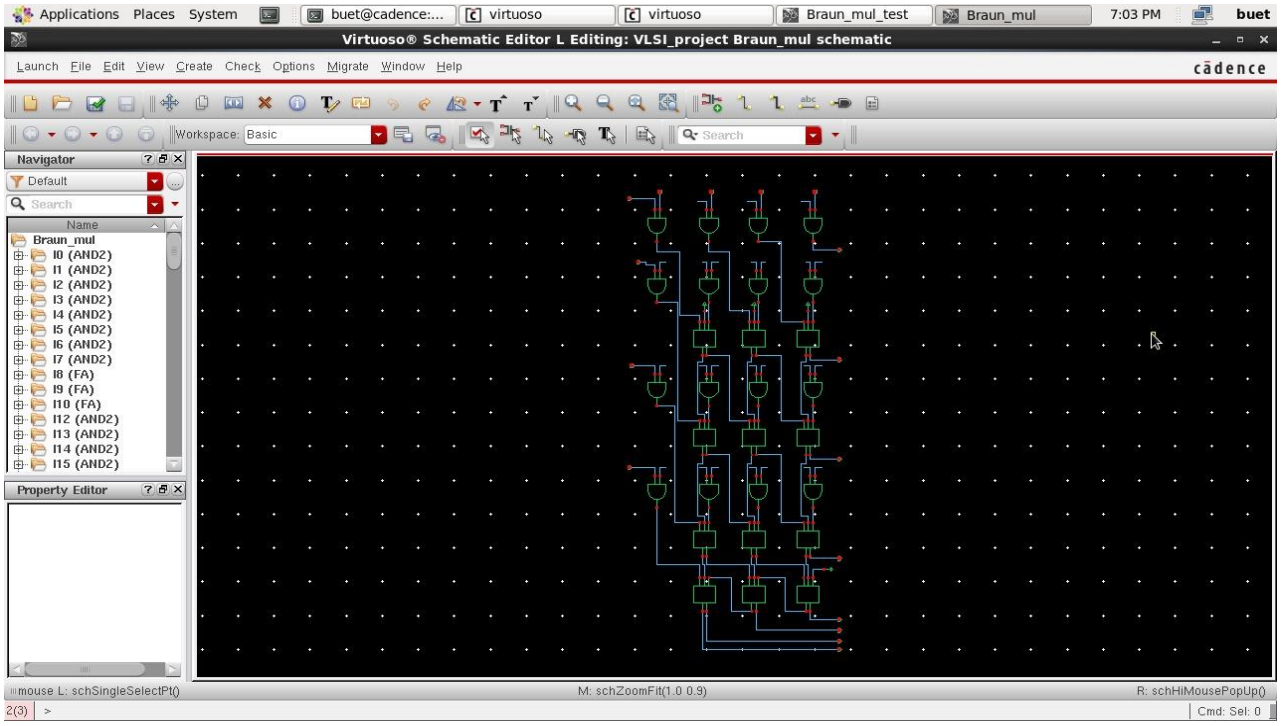
- Multiplication is a heavily used arithmetic operation that figures distinguished in signal processing and scientific applications.
- Typical DSP applications where a multiplier plays an important role include digital filtering, digital communications and spectral analysis.
- Many current DSP applications are aimed at portable, battery-operated systems, so that power dissipation becomes one of the primary design limitations.
- Meanwhile multipliers are quite complex circuits and must typically operate at a high system clock rate, dropping the delay of a multiplier is a vital part of satisfying the overall design.

BRAUN MULTIPLIER:

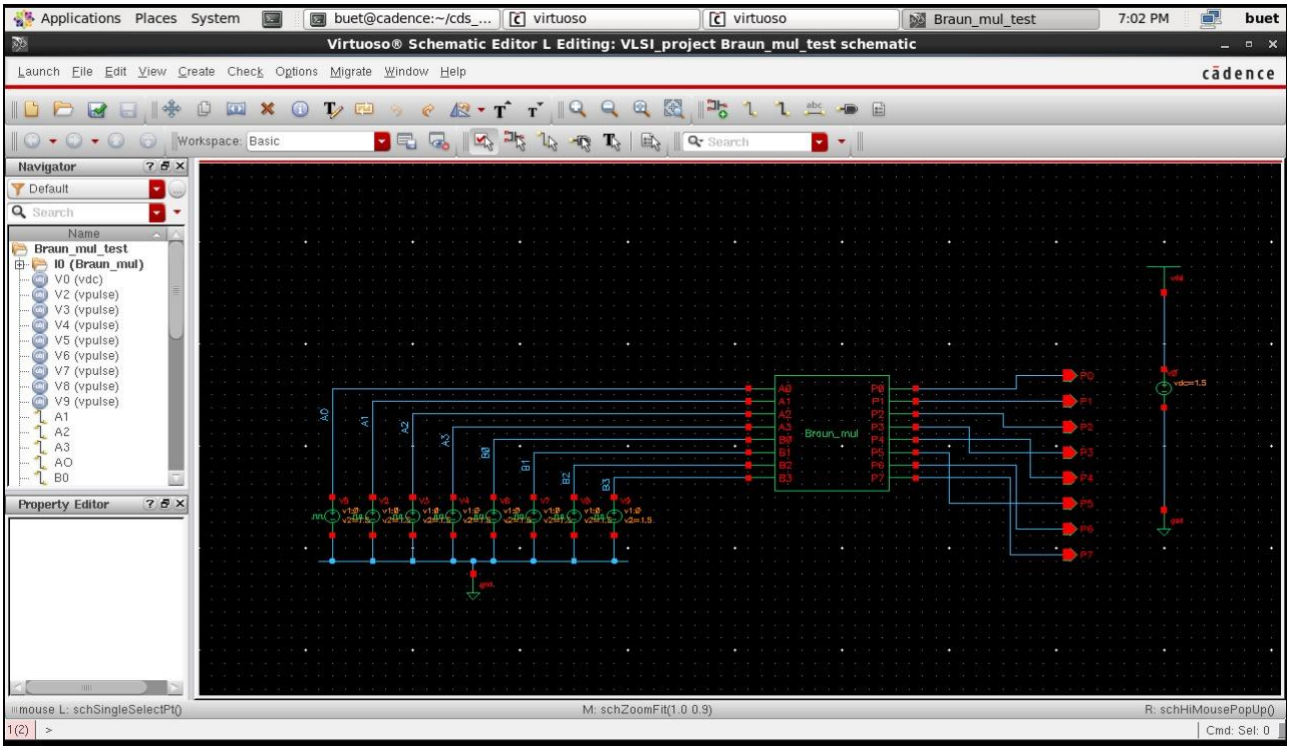
- It is a simple parallel multiplier generally called as carry save array multiplier. It has been restricted to perform signed bits.
- The structure consists of array of AND gates and adders arranged in the iterative manner and no need of logic registers.
- This can be called as non –addictive multipliers. Each product can be generated in parallel with the AND gates.
- Each partial product can be added with the sum of partial product which has previously produced by using the row of adders.
- The carry out will be shifted one bit to the left or right and then it will be added to the sum which is generated by the first adder and the newly generated partial product. The shifting would carry out with the help of Carry Save Adder (CSA) and the Ripple carry adder should be used for the final stage of the output.
- Braun multiplier performs well for the unsigned operands that are less than 16 bits in terms of speed, power and area. But it is simple structure when compared to the other multipliers.
- The main advantage of the Barun multiplier that it has only one critical path rather than many paths found in the array multiplier and this is the most widely used in DSP applications due to consuming low power.



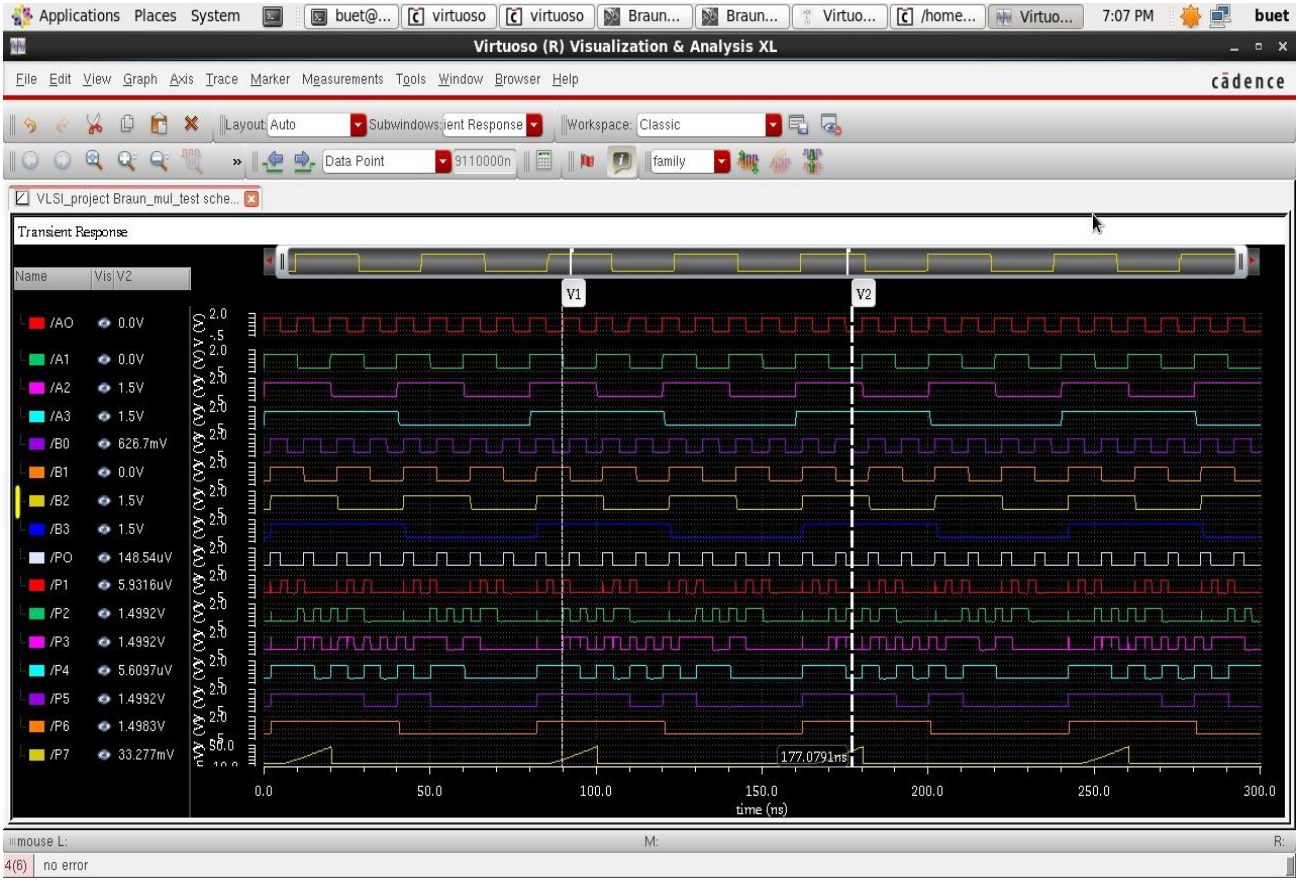
Schematic:



Test Schematic:



Transient analysis:

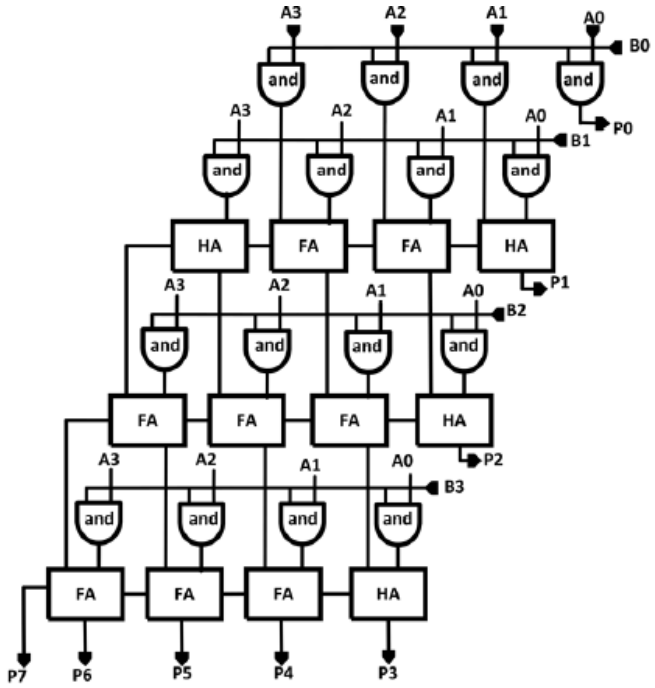


Testing:

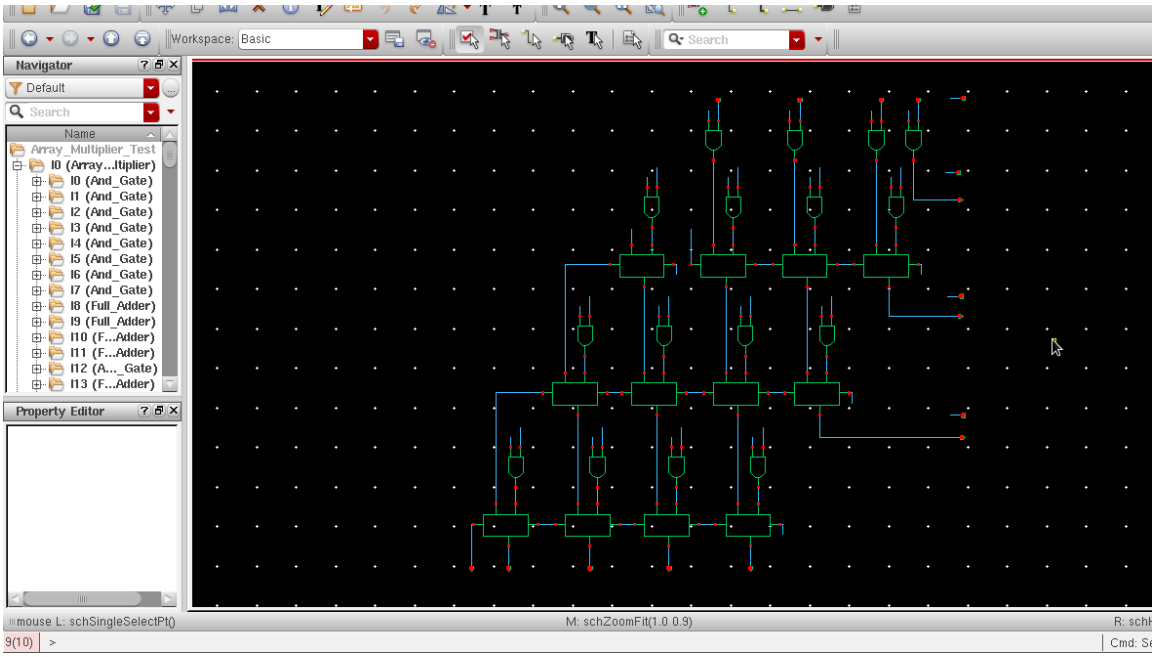
				1	1	0	0	(Multiplicand)
				0	0	1	1	(Multiplier)
				1	1	0	0	
			1	1	0	0		
		0	0	0	0			
	0	0	0	0				
0	0	1	0	0	1	0	0	(Product)

ARRAY MULTIPLIER

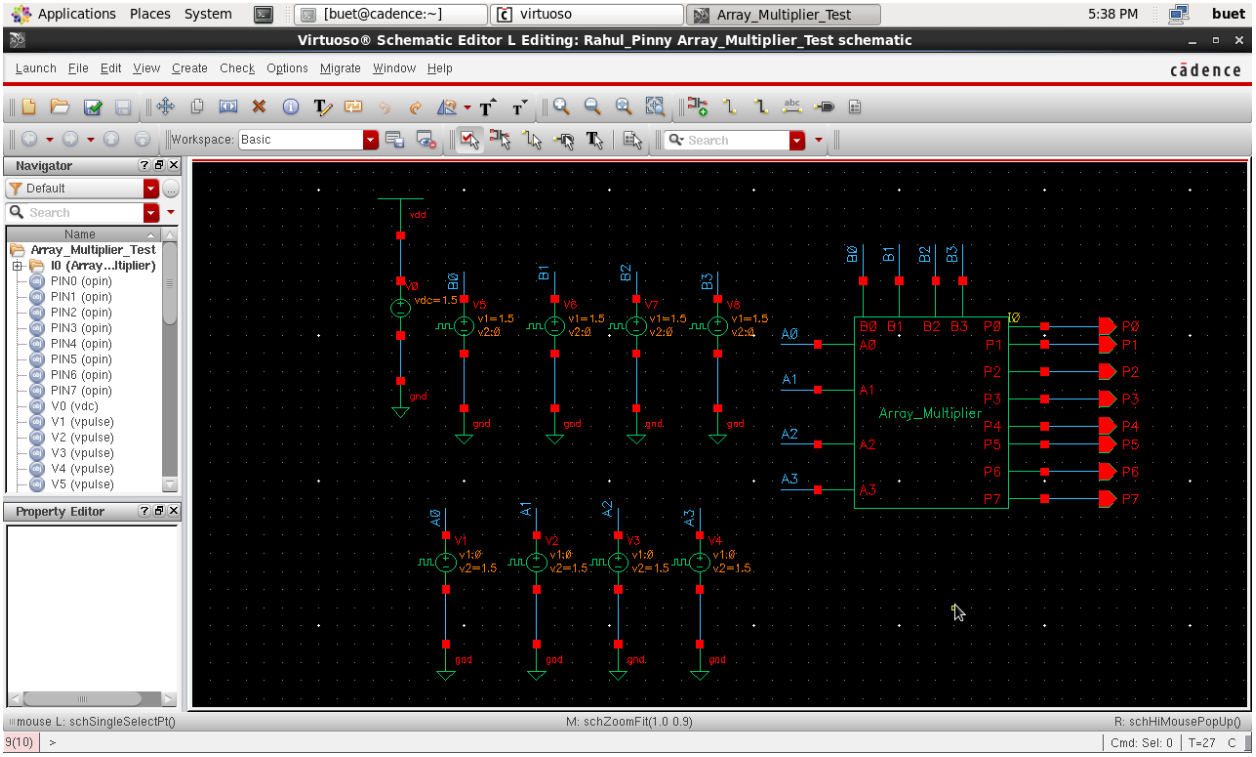
- Array multiplier is the simplest structure of parallel multiplier.
- This multiplier using the standard adds and shift operation based on 'add and shift' algorithms to perform a multiplication operation. The structure of 4-bit array multiplier is presented in figure.
- The partial products generator consists of n number of 'AND' gates to multiply the multiplicand with each bit of the multiplier and then these partial products are shifted depending on their order and this summation operation can be performed by using full adder and a half adder.
- In 4x4 array multiplier, 4x4 AND gates used to generate partial products and 4x (4-2) full adders and 4 half adders used to generate.



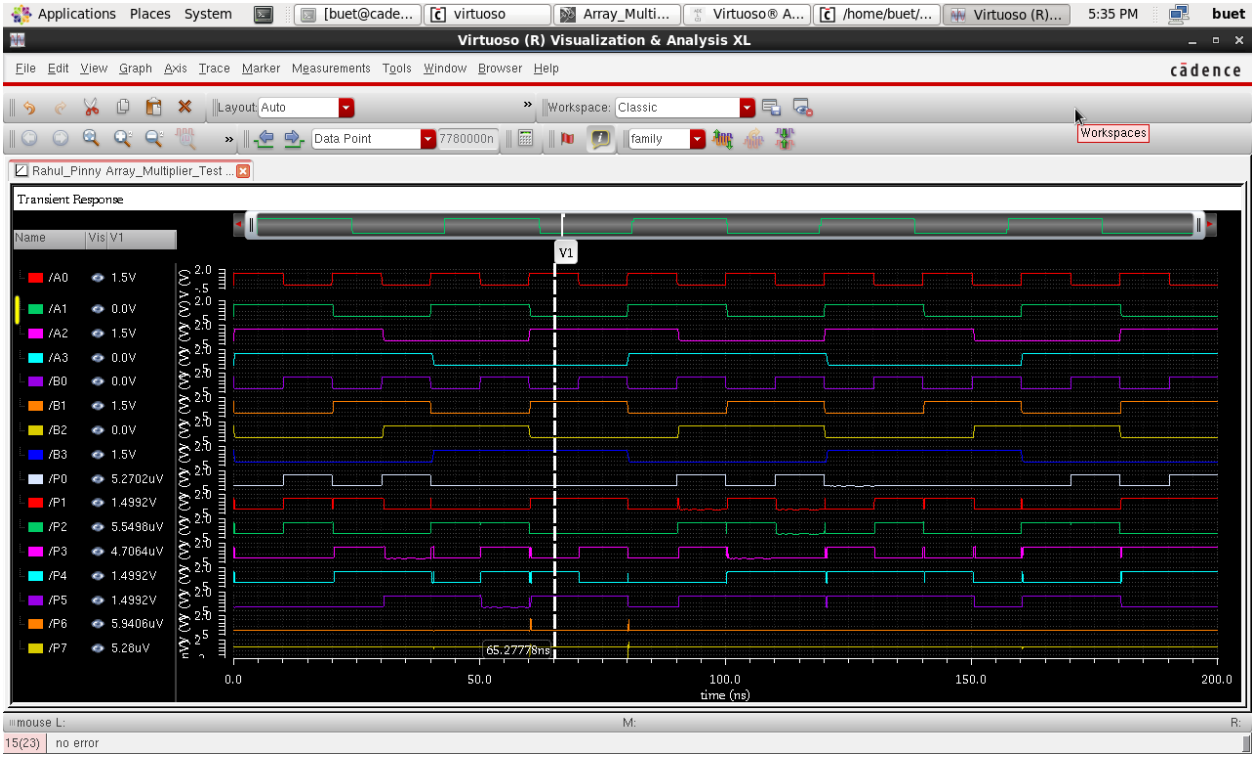
Schematic:



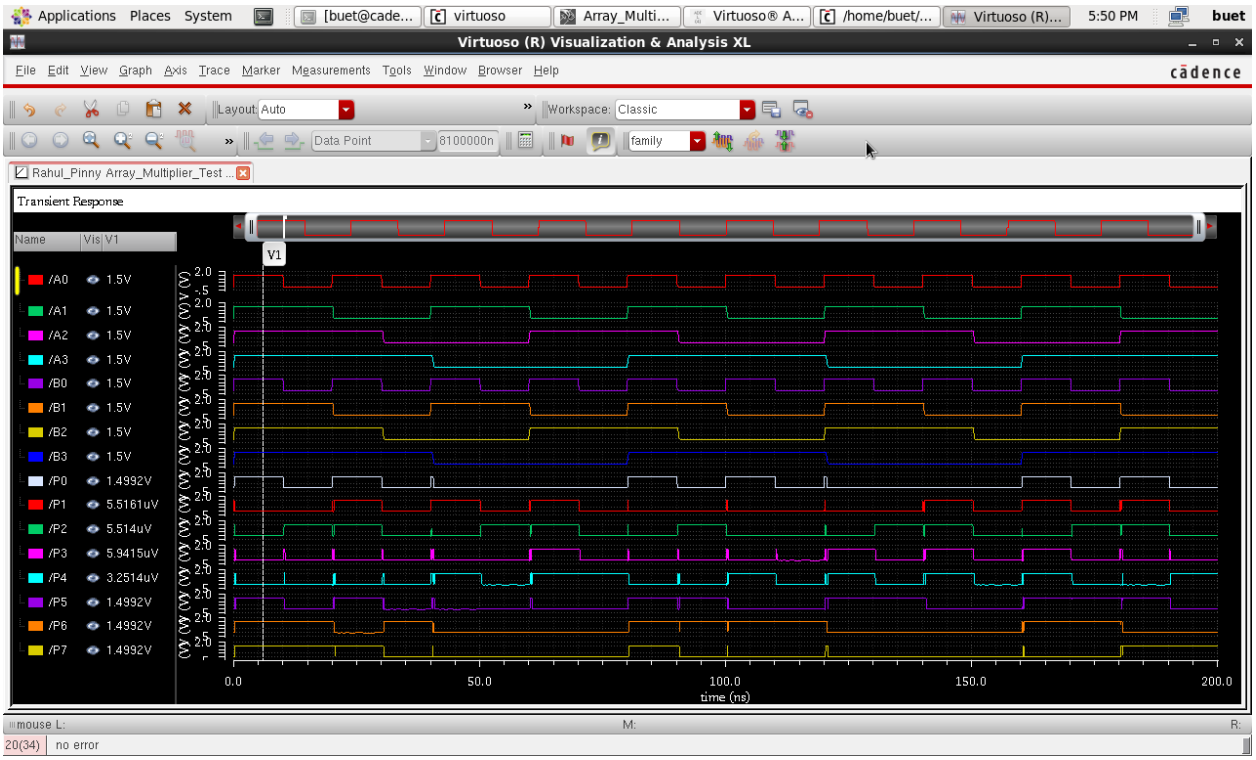
Test Schematic:



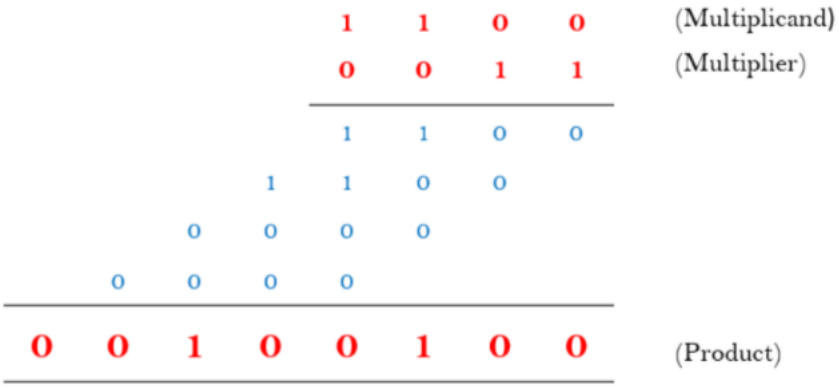
Transient analysis 1:



Transient analysis 2:

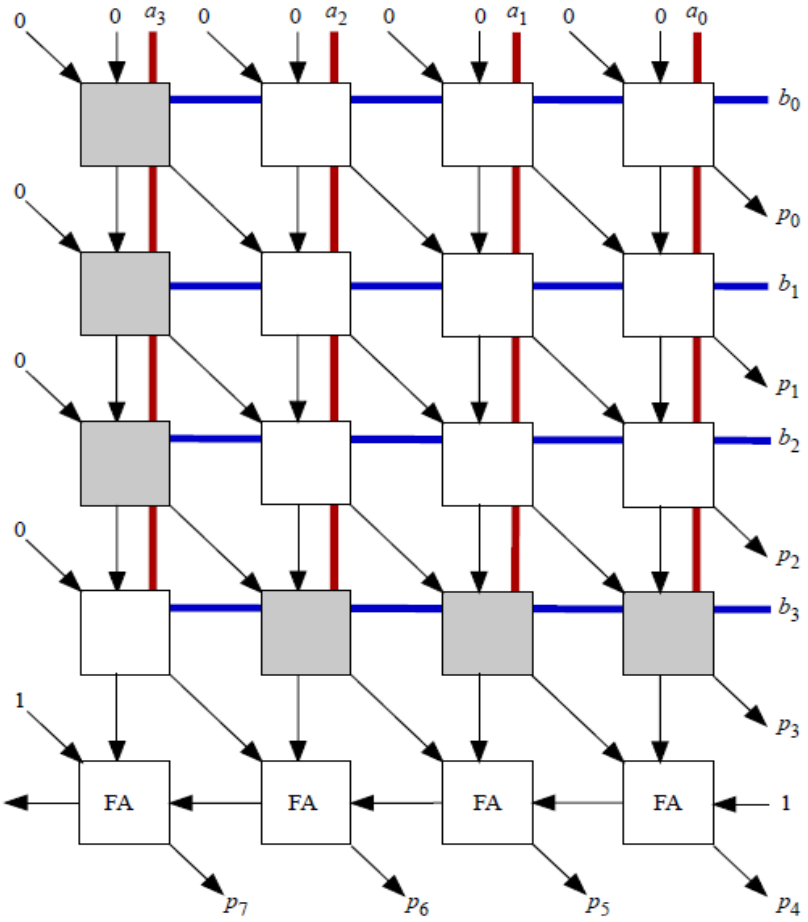


Testing:

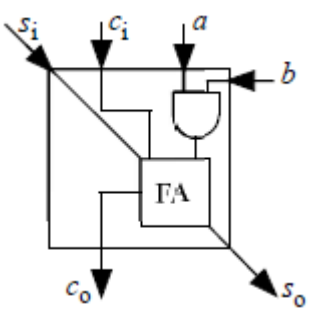


BAUGH WOOLEY MULTIPLIER

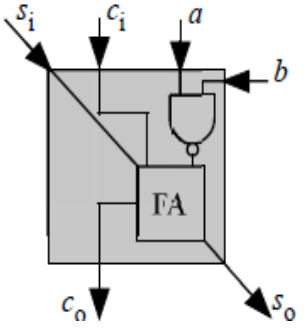
- A Baugh Wooley multiplier based on parallel array architecture.
- This multiplier is used for both unsigned and signed number multiplication.
- Signed number operands which are represented in 2's complemented form to make sure that the signs of all partial products are positive.
- The 4x4 Baugh Wooley multiplier is shown in figure.



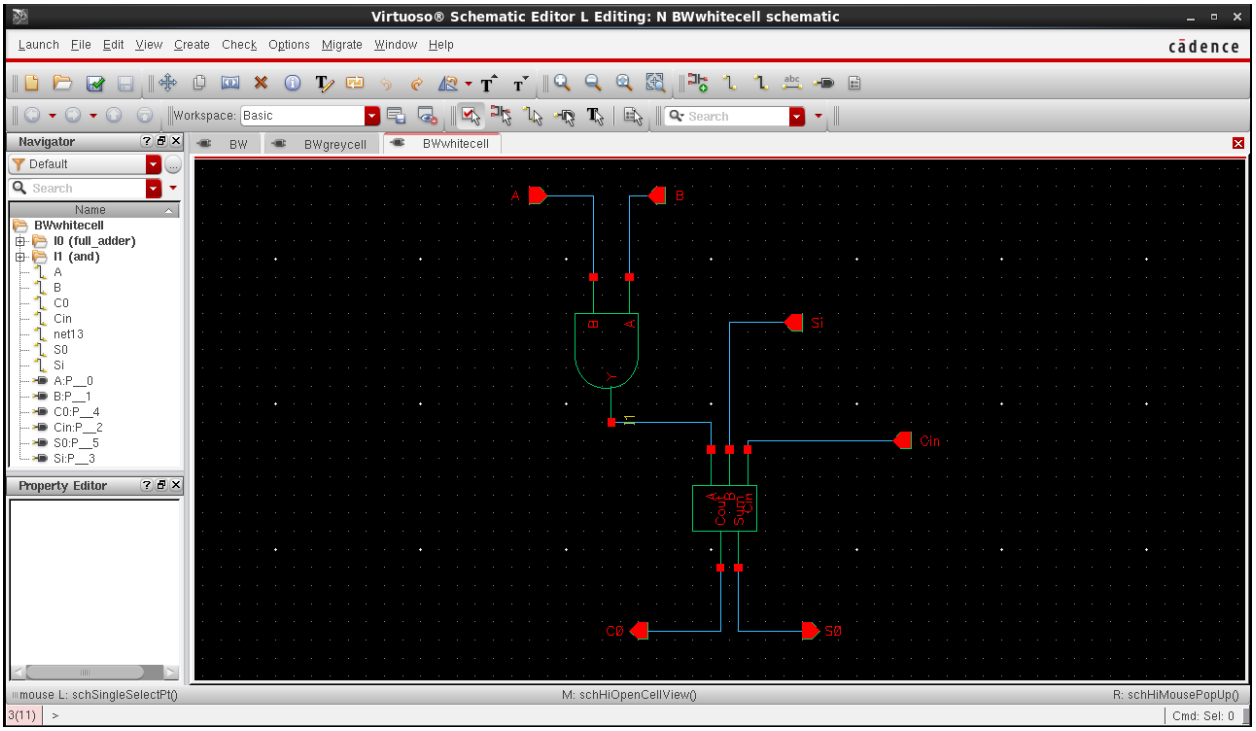
White cell



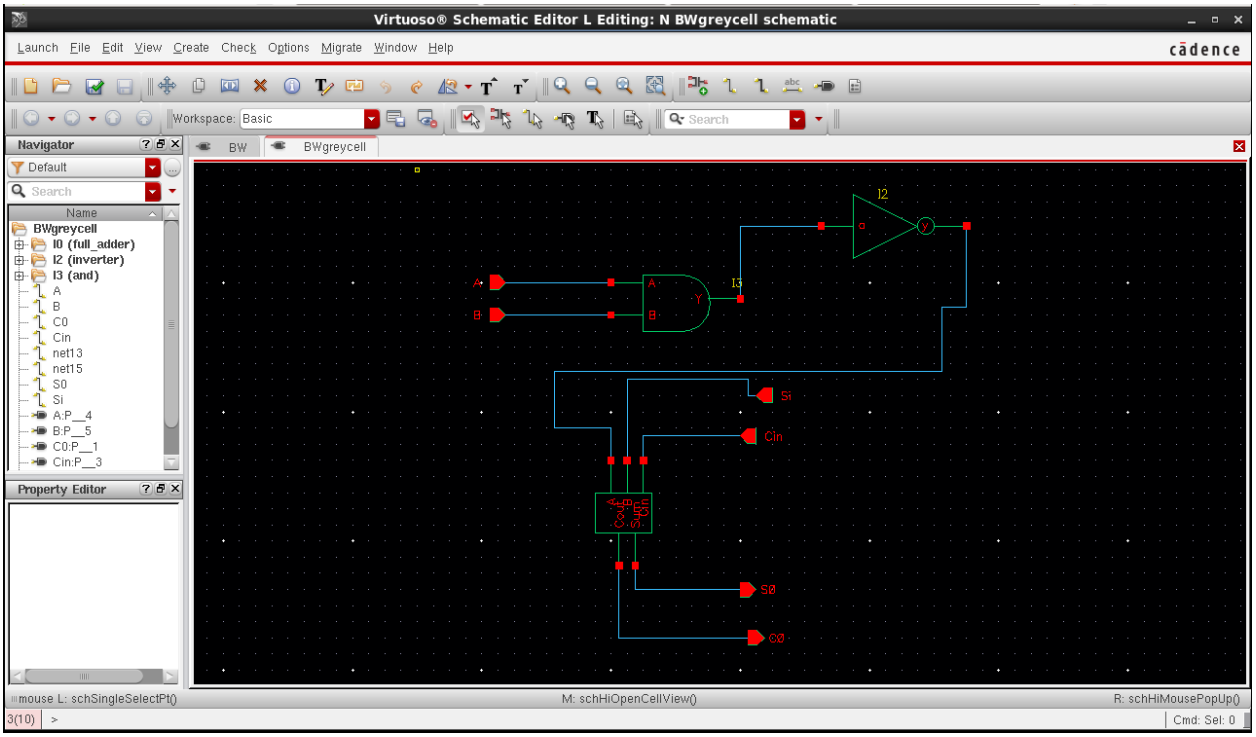
Grey cell



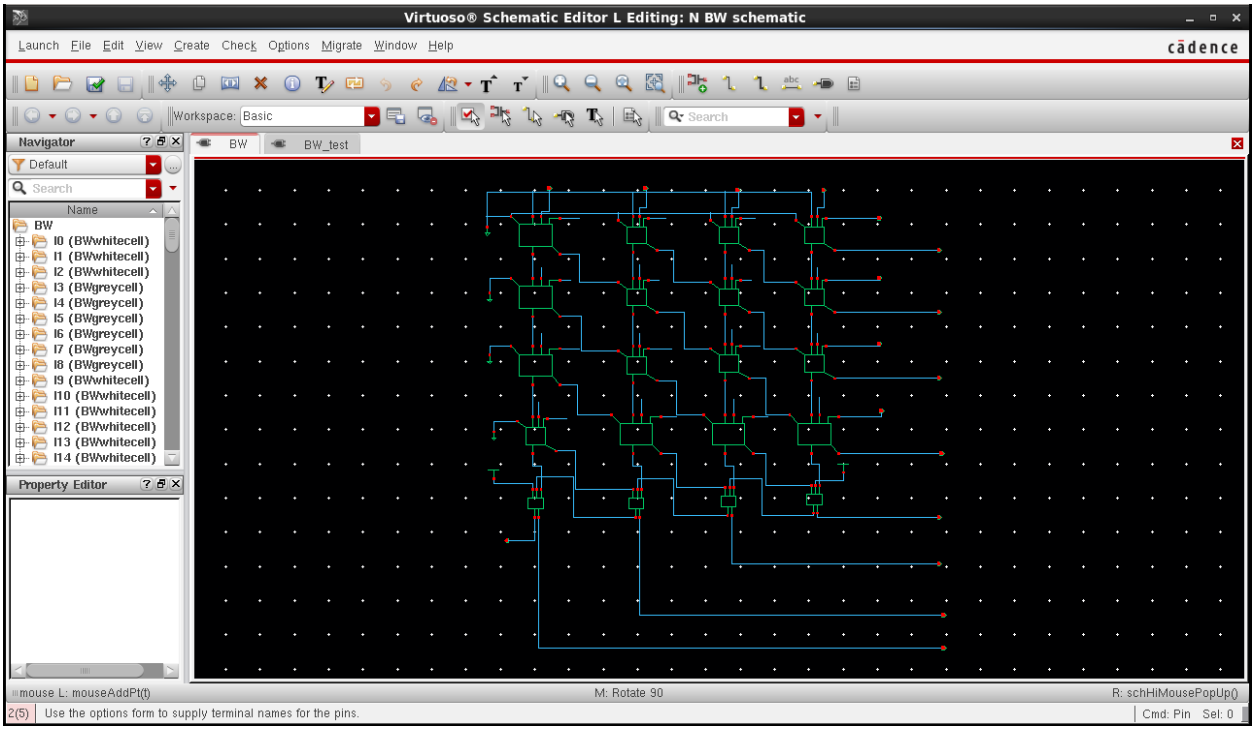
White cell Schematic



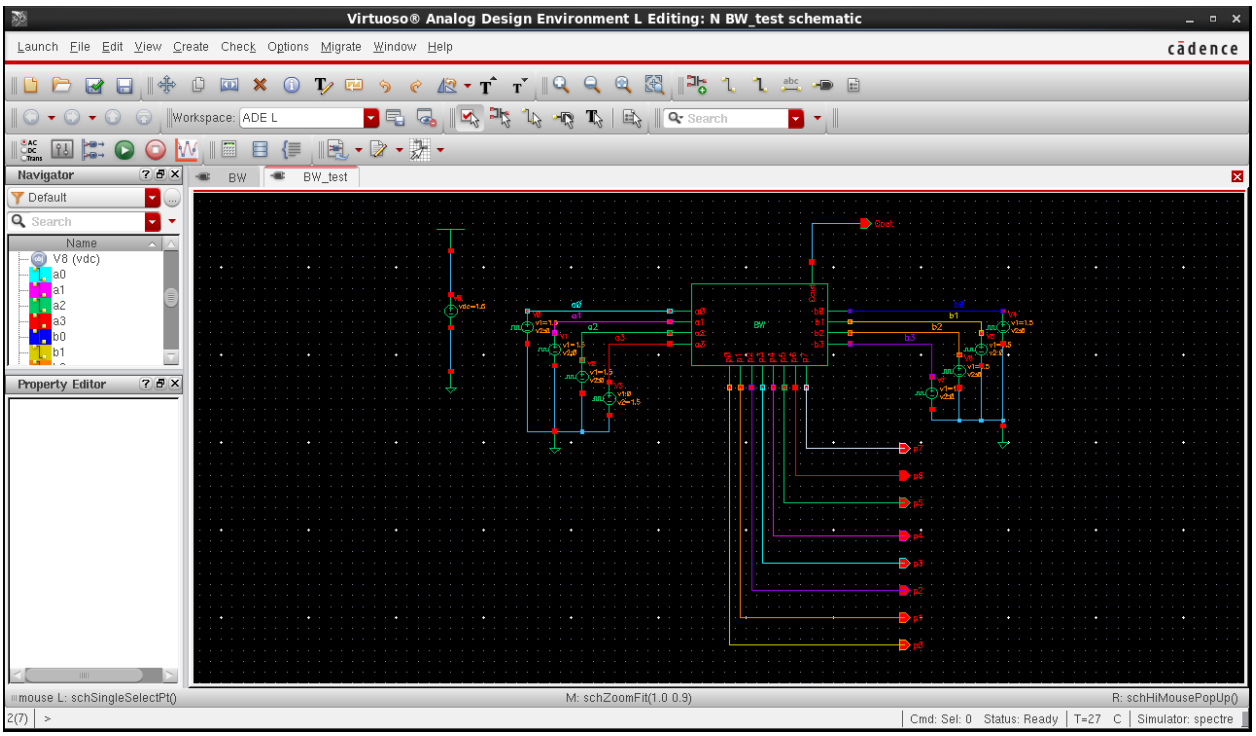
Grey cell Schematic



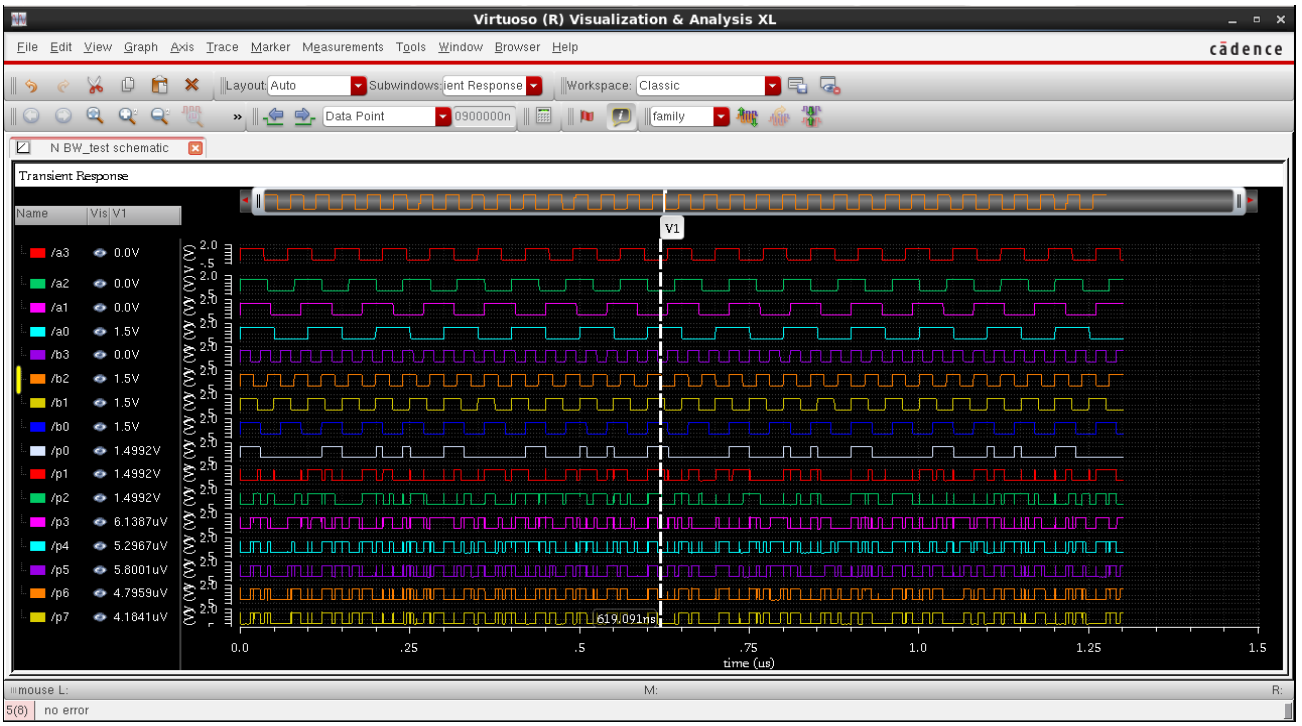
Multiplier schematic:



Test schematic:



Simulation results:



Testing:

(Sign Bits)

	1	0	0	1
	0	0	1	1
		0	0	1
	0	0	1	-
0	0	0	-	-
- 0	-	-	-	-
1 1	1	1	1	1
- -	-	-	-	1
1 1	1	0	0	1
- -	-	-	-	1
(Sign Bit) 1	1	0	1	0

(Signed Multiplicand = -7)

(Signed Multiplier = 3)

(Signed Product = -21)