8 BIT MULTIPLIER DESIGN

AKSHAY K (EE17D410)

BIJAY J (EE16D414)

KASYAP V KARUN (EE17M038)

TABLE OF CONTENTS

Problem Statement

Design an 8 bit multiplier with and without pipelining. Also find:

- Area estimate of the multiplier without pipeline
- Maximum operating frequency of the multiplier (without pipeline)
- Location of the flip flop to double the frequency of operation
- SPICE simulation showing the frequency of operation with and without pipelining

1 Eight Bit Multiplier without Pipelining

1.1 Circuit Diagram

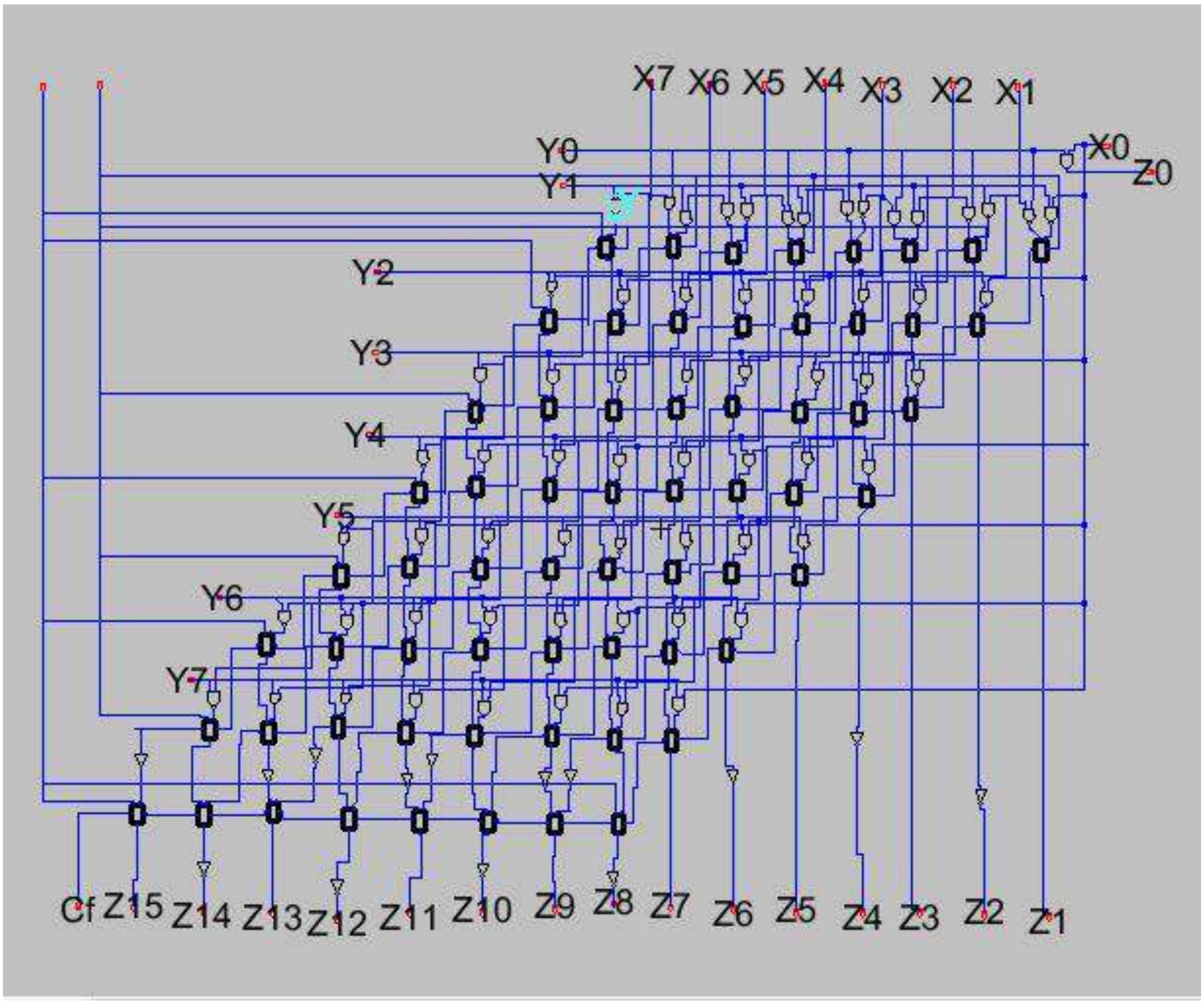


Figure 1 Schematic diagram of 8-bit multiplier without pipelining

1.2 Circuit Design

The entire carry save multiplier system was realised with inverting adders alone, without using any non-inverting adder. With this design, it is possible to eliminate the need for any inverter in the critical carry propagation path. The inverter-less critical path offers minimum delay for signals to propagate from input to output and fastens up the circuit. The choice of 'inverting adders only' based design offers additional advantages other than delay reduction. As an inverting adder doesn't need inverter at the sum and carry out path, we can significantly reduce the total number of inverters needed in the entire circuit and thereby save some space. Lesser the gates needed to achieve the same functionality, lesser the area needed and lesser the power dissipation. The catch in this design is that, while finally taking the output bits, we need to place inverters at the output of alternate adders (see Fig.1, inverters along Z_2, Z_4, Z_6, Z_8, Z_{10}, Z_{12} and Z_{14}). Also, while feeding the partial products to a non-inverting adder in the circuit, care must be taken whether to feed it directly (i.e through an AND gate) or to invert and feed (i.e. through a NAND gate). This choice depends on whether the carry input to the adder from an adder in the previous stage is non-inverted or inverted.

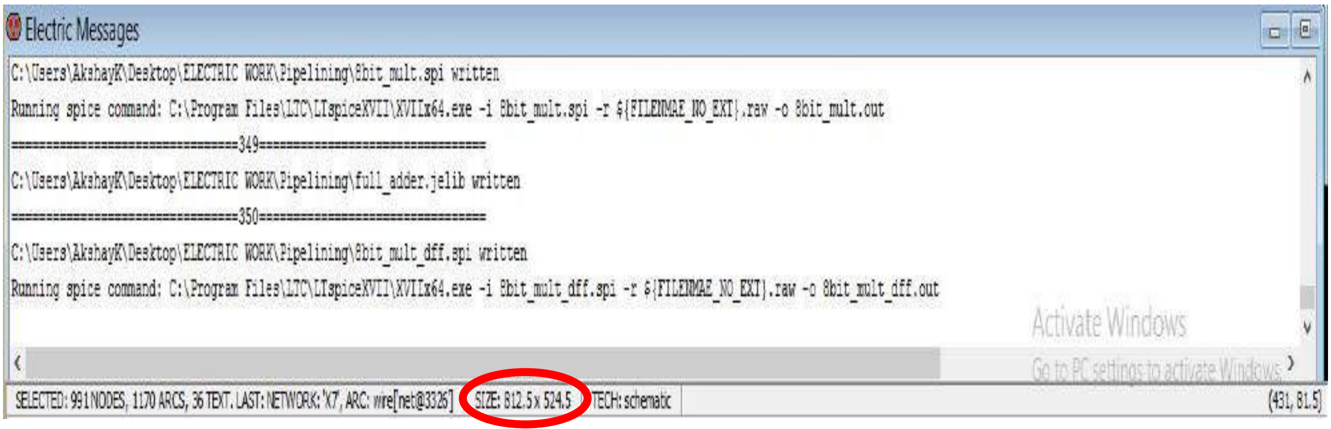
Implementation of CSM entirely with inverted full adder blocks enabled us to eliminate 128 inverters from the circuit compared to implementation using full adders only. Moreover it will decrease the number of AND gates used to generate partial products from 56 to 33 and the number of NAND gates used increases from 8 to 31. Since the AND gate is made of NAND gate cascaded with inverter, decrease in the number of AND gates will reduce delay in partial product generation and in the subsequent stages also.

1.3 Area Estimate and Floor Plan of CSM

$$Area\ of\ CMS = \frac{Area\ of\ CMS\ Schematic}{\begin{array}{c}Area\ of\ Inverting\ FA\ in\ the\\ corresponding\ schematic\end{array}} X\ (Area\ of\ Inverting\ FA\ in\ its\ layout)$$

Component	Size of Schematic	Size of Inverted FA in schematic	Area of Component (m^2)
Inverted Full Adder	437 x 236	437 x 236	9.282 x 10^{-9}
CSM (Without pipelining and D flip flop)	812.5 x 524.5	12 x 16	2.06x 10^{-5}
CSM (Without pipelining and with D flip flop)	877 x 459	12 x 16	1.946 x 10^{-5}

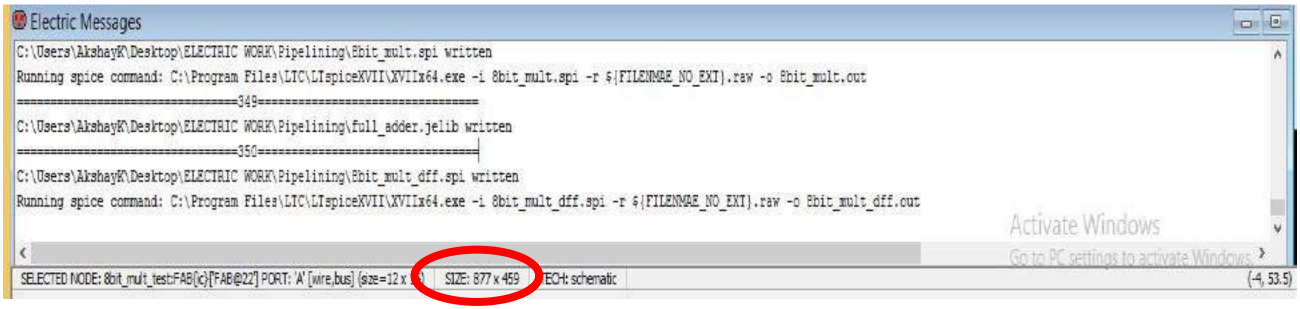
The screen shots taken for the area estimation of the CSM as tabulated above is given below;



```
C:\Users\AkshayK\Desktop\ELECTRIC WORK\Pipelining\8bit_mult.spi written
Running spice command: C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe -i 8bit_mult.spi -r ${FILENMAE_NO_EXT}.raw -o 8bit_mult.out
================================349================================
C:\Users\AkshayK\Desktop\ELECTRIC WORK\Pipelining\full_adder.jelib written
================================350================================
C:\Users\AkshayK\Desktop\ELECTRIC WORK\Pipelining\8bit_mult_dff.spi written
Running spice command: C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe -i 8bit_mult_dff.spi -r ${FILENMAE_NO_EXT}.raw -o 8bit_mult_dff.out
```

SELECTED: 991 NODES, 1170 ARCS, 36 TEXT. LAST: NETWORK: 'X7', ARC: wire[net@3326] | SIZE: 812.5 x 524.5 | TECH: schematic | (431, 81.5)

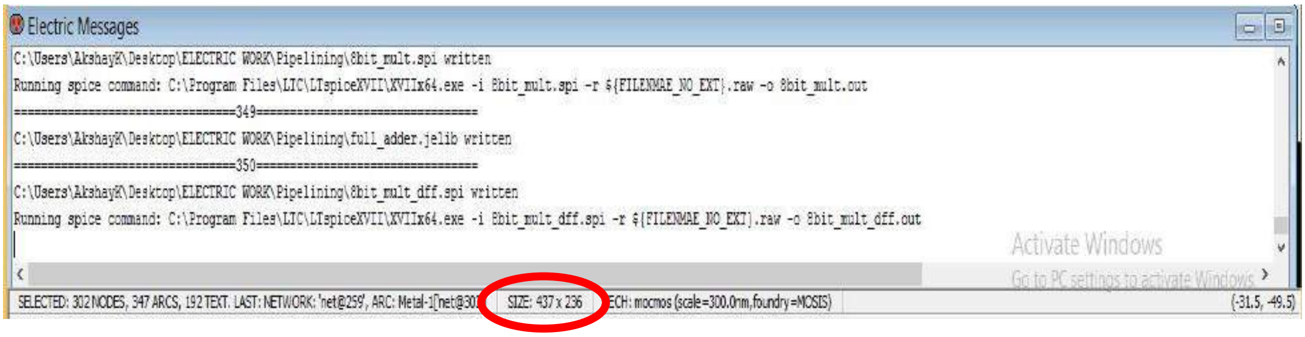
Figure 2 CSM without pipelining and with D flip flop



```
C:\Users\AkshayK\Desktop\ELECTRIC WORK\Pipelining\8bit_mult.spi written
Running spice command: C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe -i 8bit_mult.spi -r ${FILENMAE_NO_EXT}.raw -o 8bit_mult.out
================================349================================
C:\Users\AkshayK\Desktop\ELECTRIC WORK\Pipelining\full_adder.jelib written
================================350================================
C:\Users\AkshayK\Desktop\ELECTRIC WORK\Pipelining\8bit_mult_dff.spi written
Running spice command: C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe -i 8bit_mult_dff.spi -r ${FILENMAE_NO_EXT}.raw -o 8bit_mult_dff.out
```

SELECTED NODE: 8bit_mult_test:FAB{ic}['FAB@22'] PORT: 'A' [wire,bus] (size=12 x 16) | SIZE: 877 x 459 | TECH: schematic | (-4, 53.5)

Figure 3 Inverted Full Adder size in CSM Schematic



```
C:\Users\AkshayK\Desktop\ELECTRIC WORK\Pipelining\8bit_mult.spi written
Running spice command: C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe -i 8bit_mult.spi -r ${FILENMAE_NO_EXT}.raw -o 8bit_mult.out
================================349================================
C:\Users\AkshayK\Desktop\ELECTRIC WORK\Pipelining\full_adder.jelib written
================================350================================
C:\Users\AkshayK\Desktop\ELECTRIC WORK\Pipelining\8bit_mult_dff.spi written
Running spice command: C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe -i 8bit_mult_dff.spi -r ${FILENMAE_NO_EXT}.raw -o 8bit_mult_dff.out
```

```
SELECTED: 302 NODES, 347 ARCS, 192 TEXT. LAST: NETWORK: 'net@259', ARC: Metal-1['net@30   SIZE: 437 x 236   ECH: mocmos (scale=300.0nm,foundry=MOSIS)   (-31.5, -49.5)
```

Figure 4 Inverted Full Adder Size in its Layout

1.4 Maximum Operating Frequency (w/o Pipeline)

The propagation delay, T_d, of the 8 bit multiplier circuit without pipelining and without D flip flops, as extracted using the schematic is 5.46ns while that using the RC extracted netlist is 7.56ns. Hence, the maximum operating frequency;

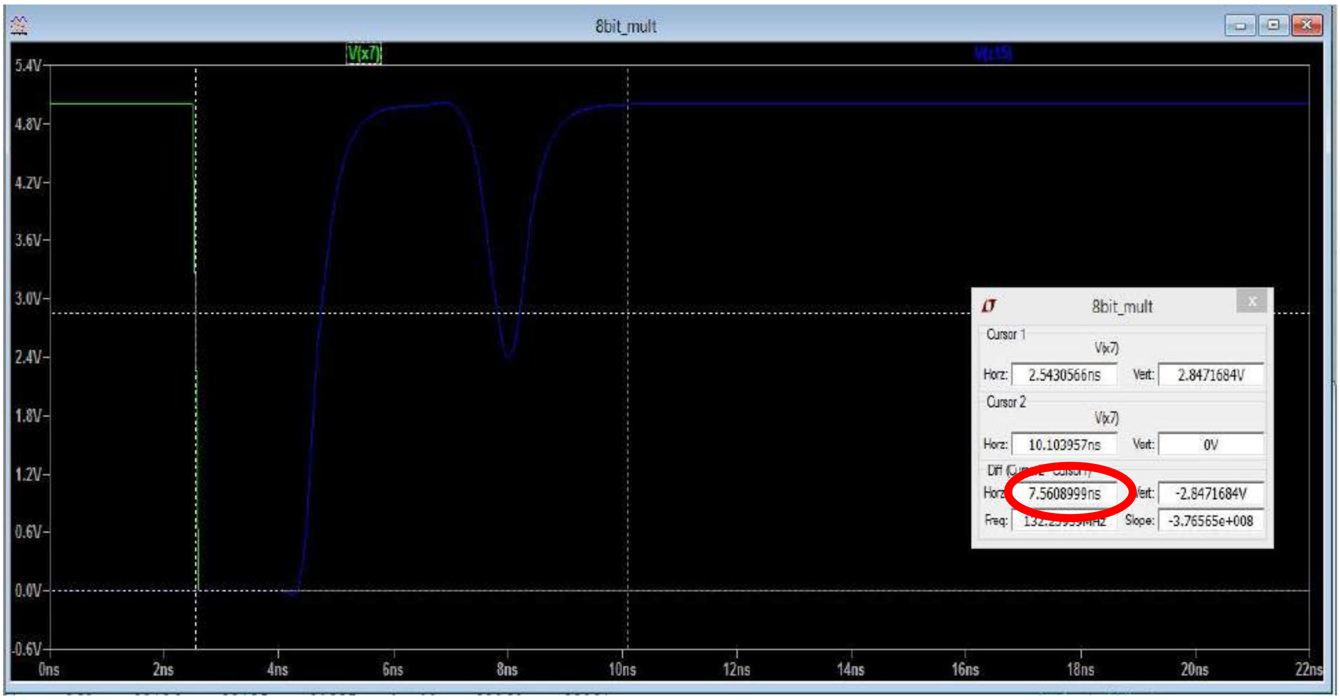
$$f = \frac{1}{Td} \qquad (1)$$

is 183.15 MHz and 132.28MHz as extracted from the schematic and netlist respectively.

Similarly the minimum clock period for CSM with input-output flip flop and without pipelining are 5.665ns and 7.765ns respectively for schematic driven and RC extracted simulations. So the maximum clock frequencies are 176.55MHz and 128.7MHz respectively.

Schematic Type	Extraction Type	Minimum Clock Period	Maximum Clock Frequency(MHz)
CSM without flip flop and Pipelining	Schematic	5.46ns	183.15
	RC extracted	7.56ns	132.275
CSM with input-output flip flops and without Pipelining	Schematic	5.665ns	176.55
	RC extracted	7.765ns	128.7

The simulated plots for the above calculations are given below;



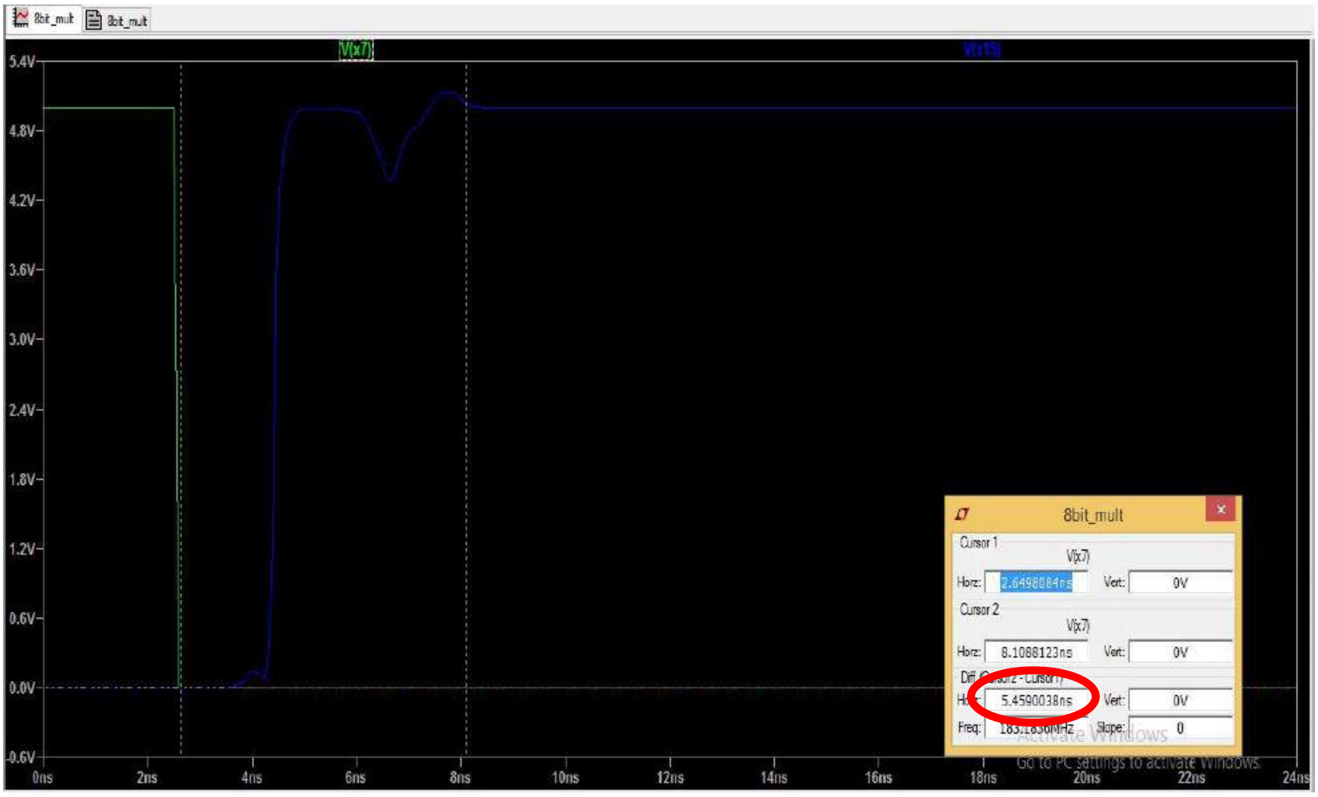
8bit_mult

Cursor 1
V(x7)
Horz: 2.5430566ns Vert: 2.8471684V

Cursor 2
V(x7)
Horz: 10.103957ns Vert: 0V

Diff (Cursor2 - Cursor1)
Horz: 7.5608999ns Vert: -2.8471684V
Freq: 132.2[illegible]MHz Slope: -3.76565e+008

Figure 5 Propagation delay of CSM without flip flops with RC extraction netlist



8bit_mult

Cursor 1
V(x7)
Horz: 2.6498084ns Vert: 0V

Cursor 2
V(x7)
Horz: 8.1088123ns Vert: 0V

Diff (Cursor2 - Cursor1)
Horz: 5.4590038ns Vert: 0V
Freq: 183.1836MHz Slope: 0

Figure 6 Propagation delay of CSM without flip flops by schematic driven simulation

1.5 Technique to Double Operating Frequency

In order to double the frequency of operation, we need to identify the critical path and observe the signal propagating in that path. Ideal placement of flip flop, for doubling the operating frequency, is exactly at the point where signal is half way away from the input and output, i.e. at the middle of the propagation delay. If the flops are placed close to the input or output, there is some time wasted for waiting for the clock edge and thereby degrading the maximum operating frequency. In our design, this middle of the propagation delay is approximately just before the vector merge stage. This is illustrated in Fig. 2. So we place flip flops between the vector merge stage and the second last

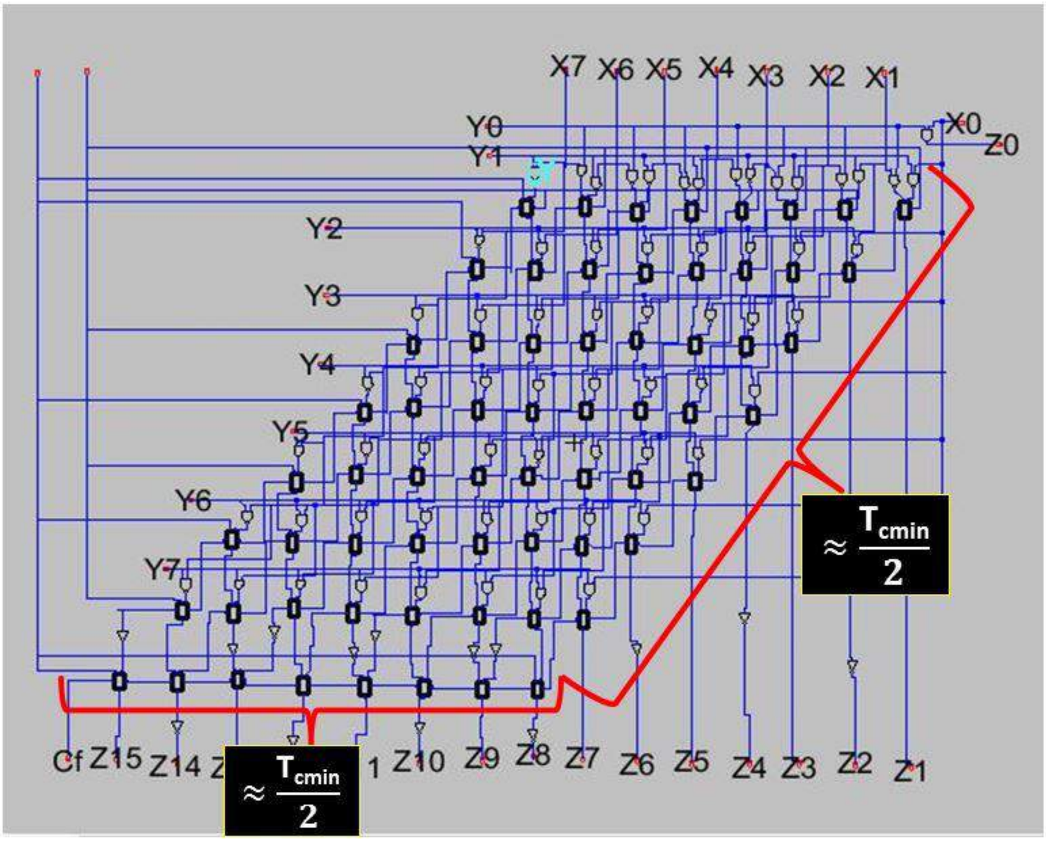
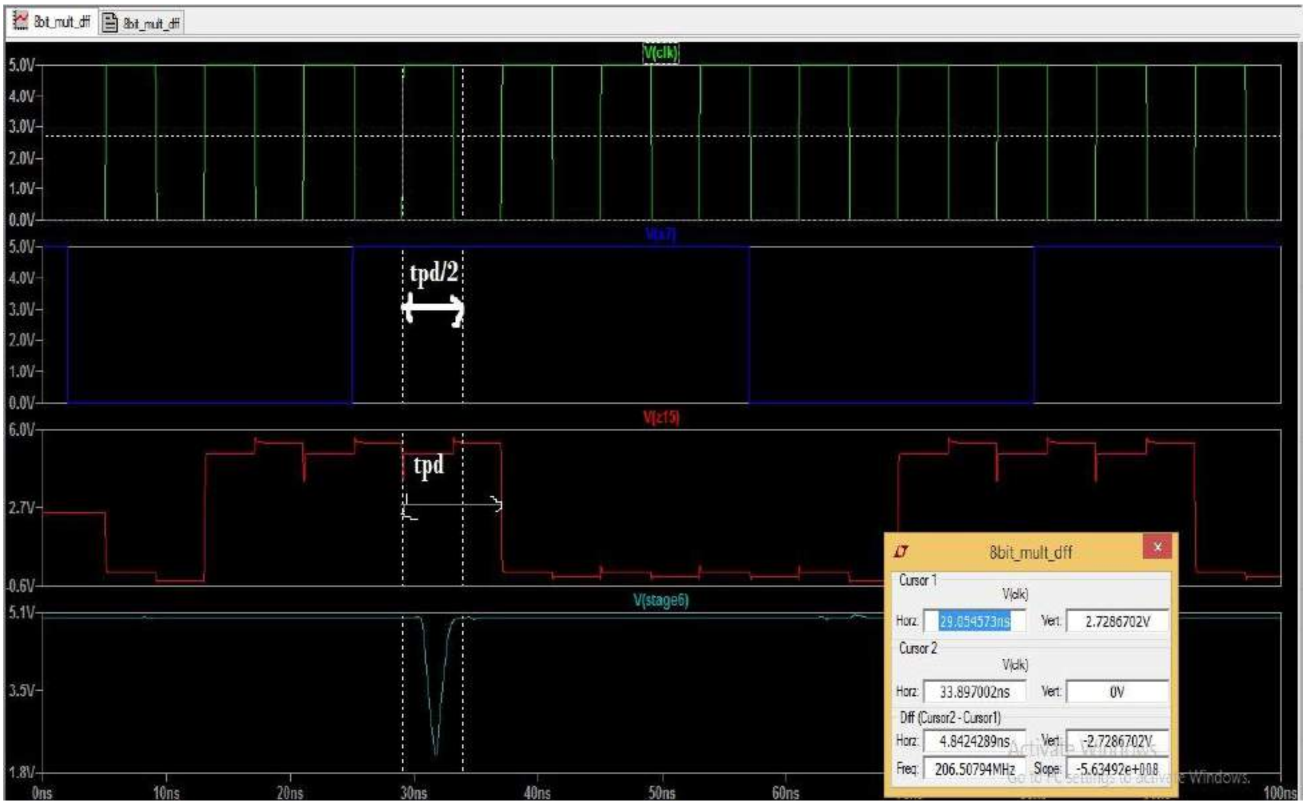

X7 X6 X5 X4 X3 X2 X1
X0
Y0
Y1
Z0
Y2
Y3
Y4
Y5
Y6
Y7
$\approx \frac{T_{cmin}}{2}$
$\approx \frac{T_{cmin}}{2}$
Cf Z15 Z14 Z
1 Z10 Z9 Z8 Z7 Z6 Z5 Z4 Z3 Z2 Z1

Figure 7 Identifying optimum location for the placement of D flip flops.

The simulated plot which shows the propagation delay of the halfway point is given below;



8bit_mult_dff

Cursor 1
V(clk)
Horz: 29.054573ns Vert: 2.7286702V

Cursor 2
V(clk)
Horz: 33.897002ns Vert: 0V

Diff (Cursor2 - Cursor1)
Horz: 4.8424289ns Vert: -2.7286702V
Freq: 206.50794MHz Slope: -5.63492e+008

The label stage6 denotes the half way point and has a propagation delay half that of the output. Stage6 node is the node at which the Full adders in the vector merging array receives inputs from.

1.6 Clock Frequency Comparison

Block Type	Simulation Type	Minimum Clock Period-Simulated Value	Minimum Clock Period-Theoretical Value	Maximum Clock Frequency-Simulated	Maximum Clock Frequency-Theoretical
CSM without pipelining	Schematic driven	6ns	5.665ns	166.67MHz	176.52MHz
	RC Extracted	8ns	7.765ns	125MHz	128.78MHz
CSM with Pipelining	Schematic Driven	4ns	2.832ns	250MHz	353MHz
	RC Extracted	6ns	3.8825ns	166.67MHz	257.56MHz

2 Eight Bit Multiplier with Flip flop & without Pipelining

2.1 Circuit Diagram

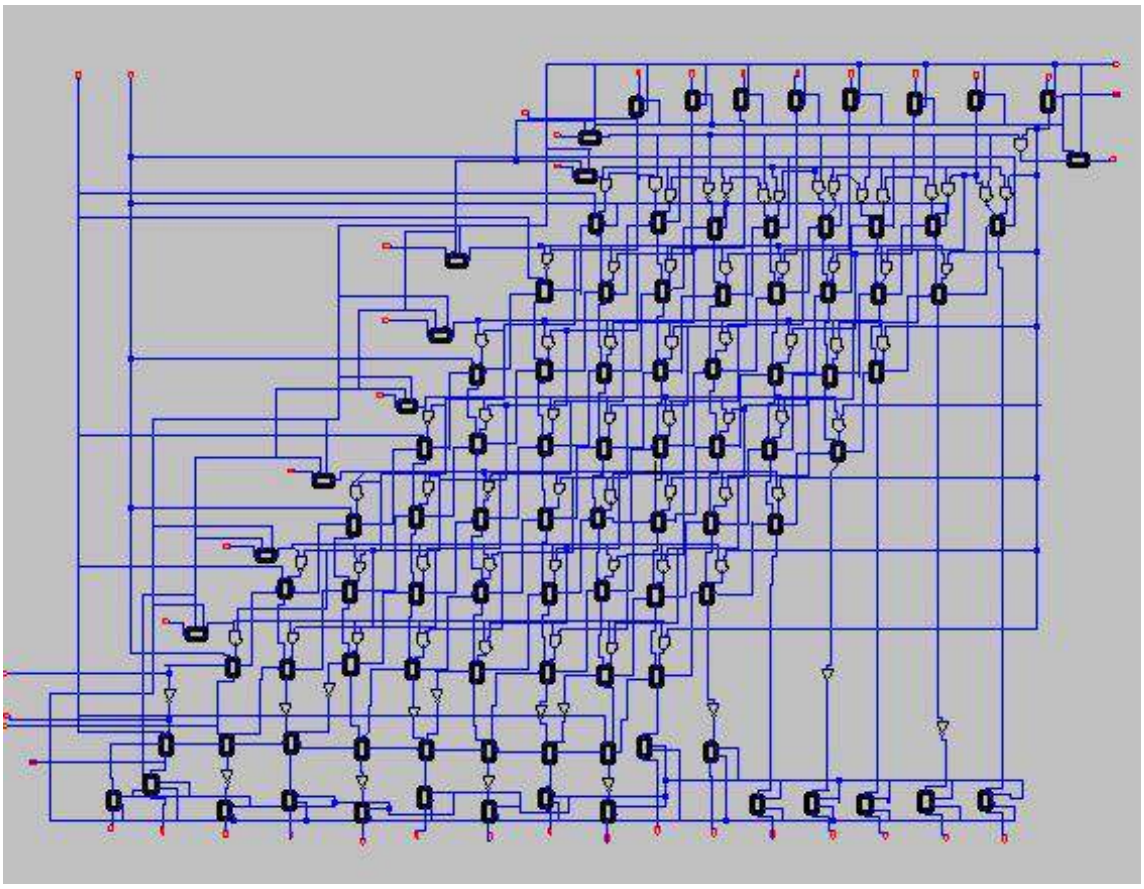


Figure 8 Circuit Diagram for non-pipelined CSM with flip flops

2.2 Maximum Operating Frequency

To determine the maximum operating frequency of the circuit, we increase the clock frequency in steps and check the output and see whether the correct output is obtained at all the rising edge in the entire cycle, for that particular frequency. The simulations done using schematic for clock period, 6.5ns and 5.5ns are shown in Fig. 4 and Fig. 5 respectively. The simulations done using RC extracted netlist for clock period, 20ns, 8ns and 6ns are shown in Fig. 6, Fig. 7 and Fig. 8 respectively. The minimum clock time period, T_{Cmin}, of the 8 bit multiplier circuit with inputs given and outputs taken through flipflops, without pipelining, as extracted using the schematic is 6ns while that using the RC extracted netlist is 8ns. Hence, the maximum operating frequency,

$$f = \frac{1}{TCmin} \qquad (2)$$

is 167MHz and 125MHz as extracted from the schematic and netlist respectively.

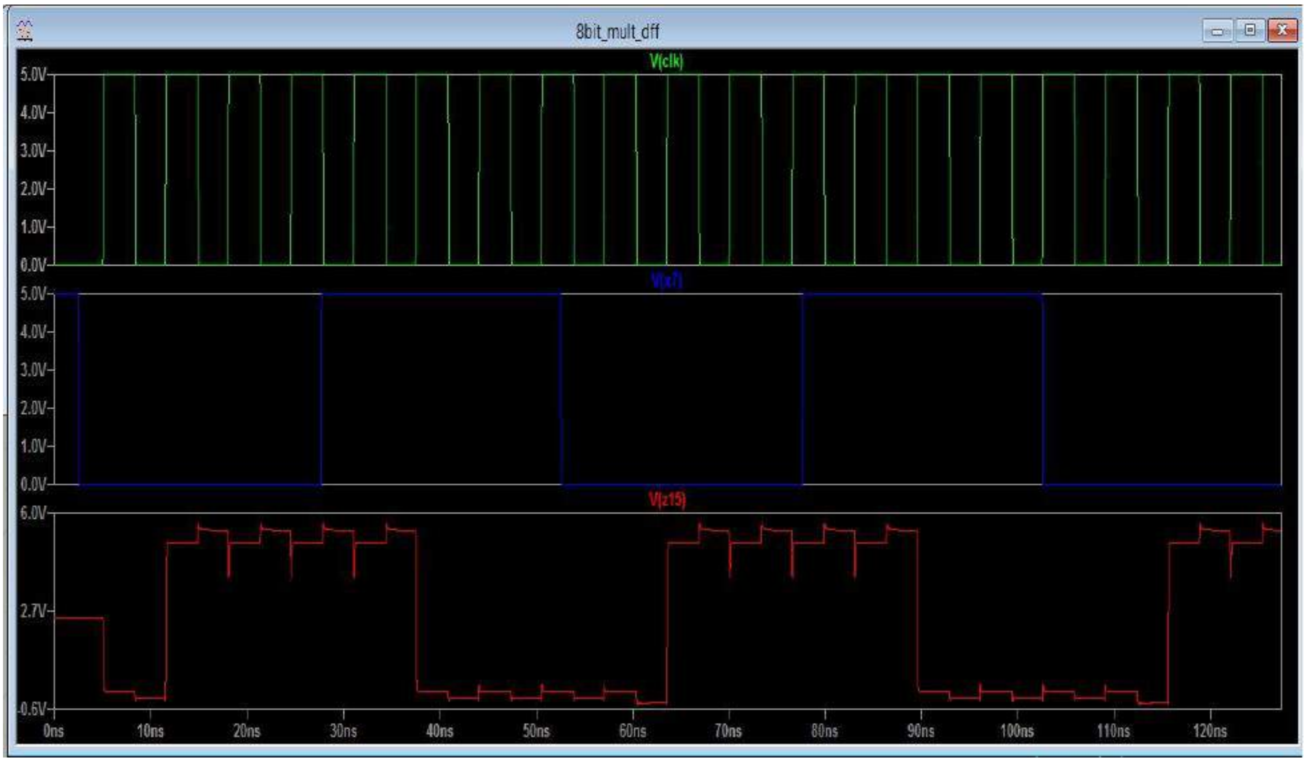


Figure 9 Schematic simulation output for clock period = 6.5ns

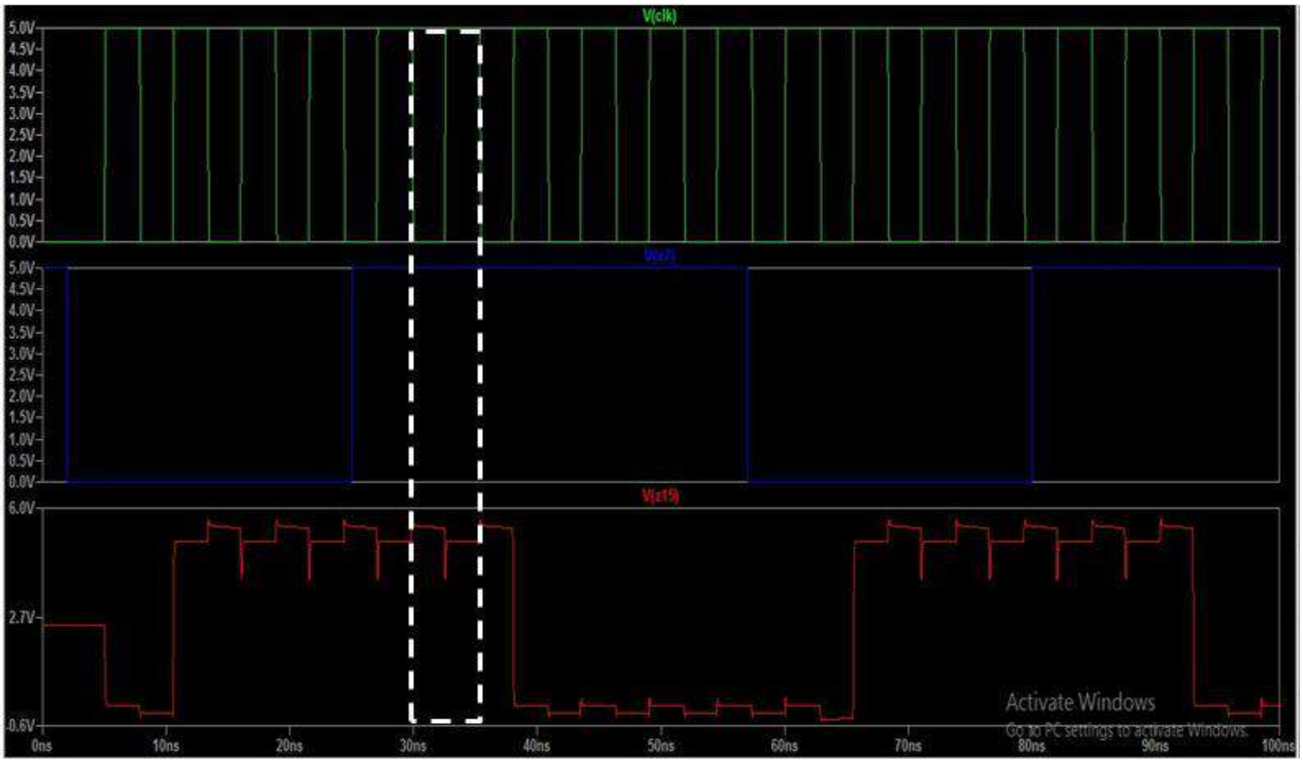


Figure 10 Schematic simulation output for clock period=5.5ns (erroneous output highlighted)

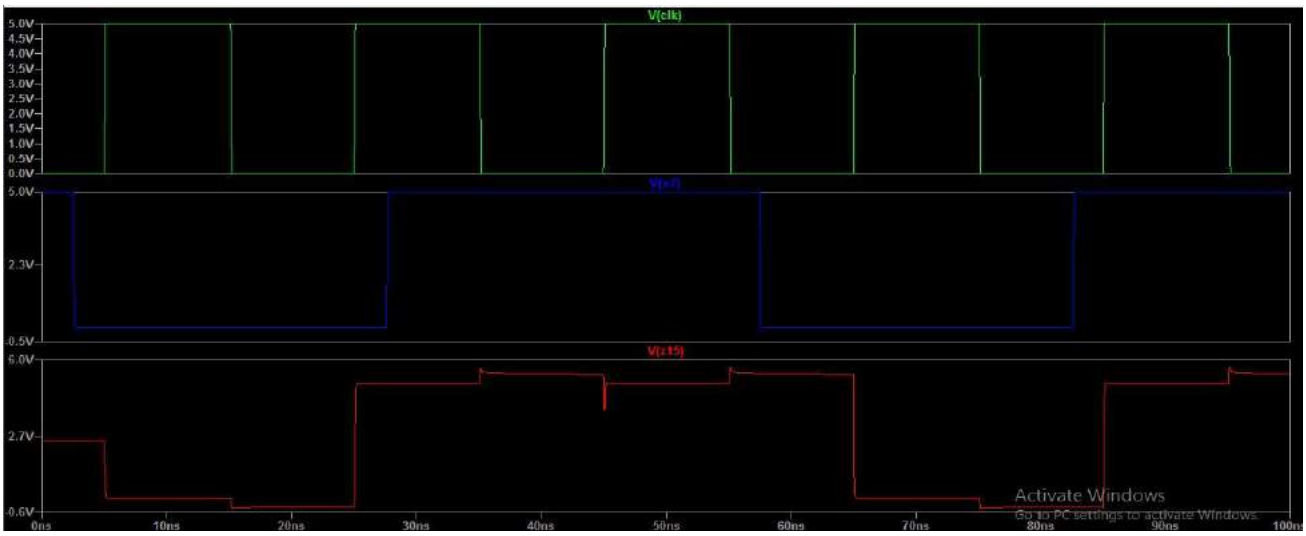


Figure 11 RC extracted netlist simulation for clock period=20ns

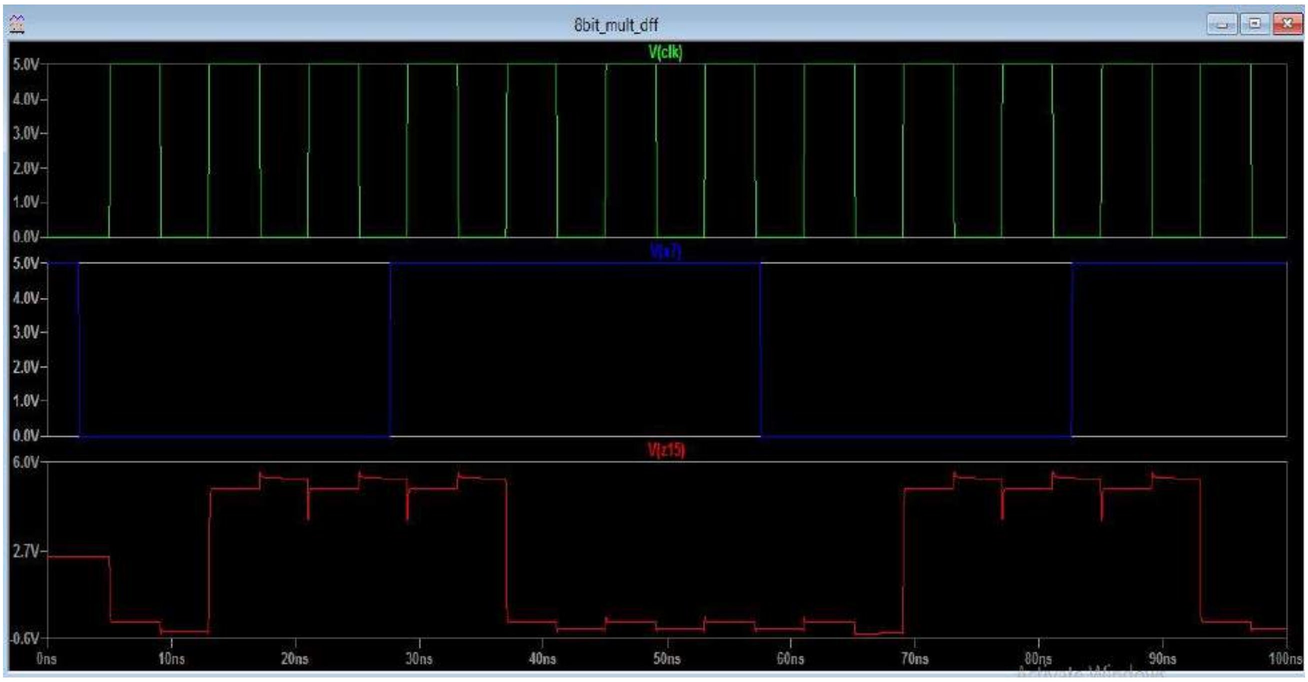


Figure 12 RC extracted netlist simulation for clock period = 8ns

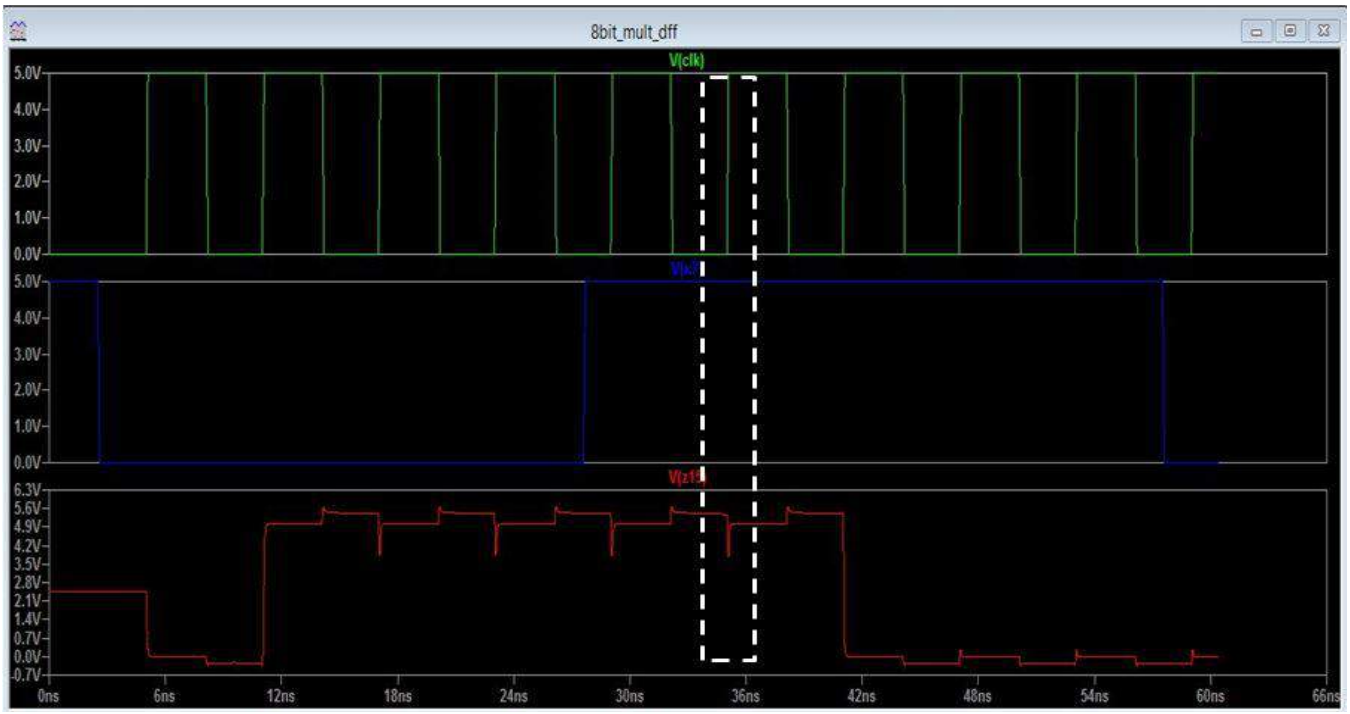


Figure 13 RC extracted netlist simulation for clock period=6ns(erroneous output highlighted)

3 Eight Bit Multiplier With Pipelining

3.1 Circuit Diagram

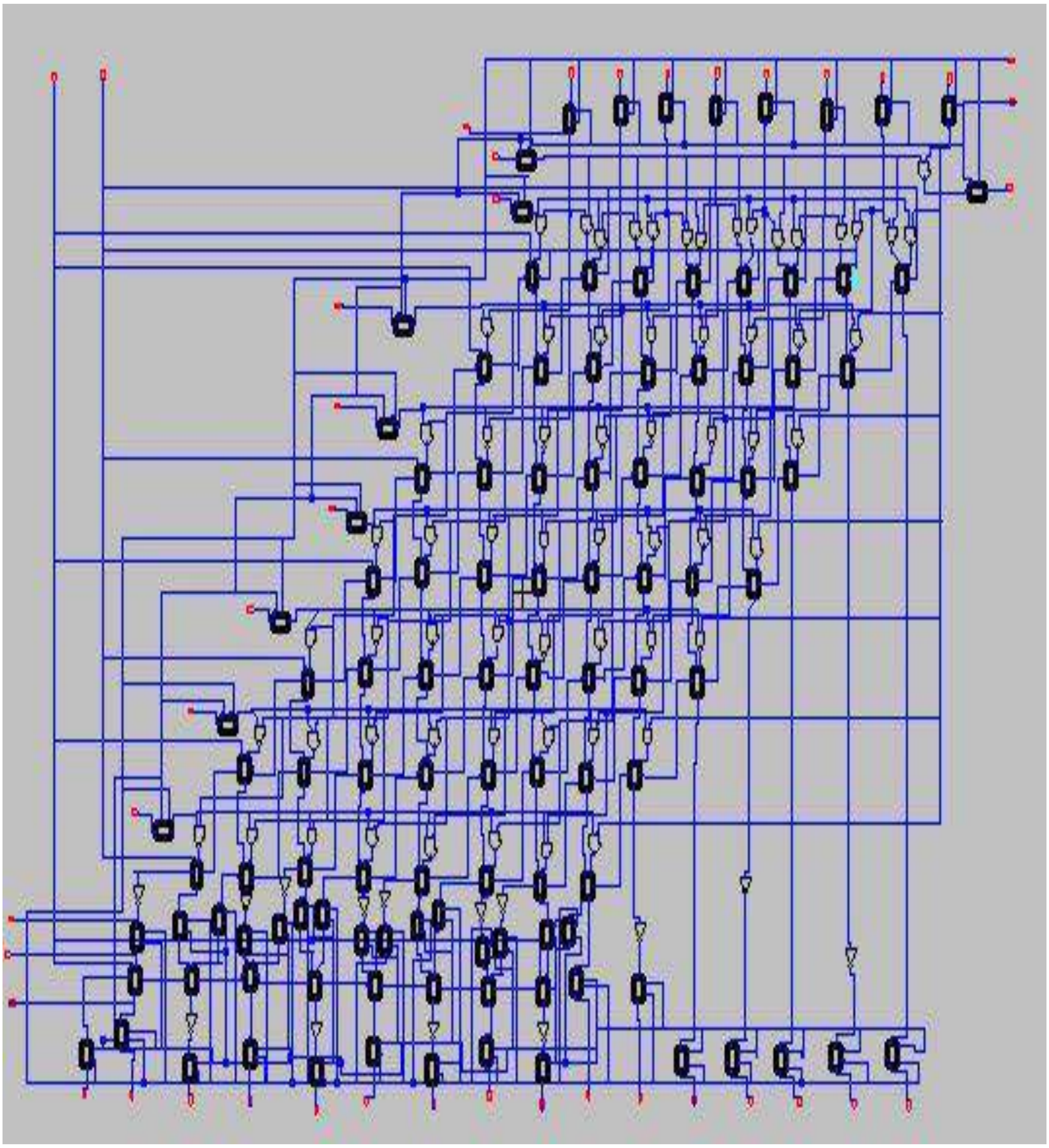


Figure 14 Schematic diagram of 8-bit multiplier with pipelining

3.2 Maximum Operating Frequency

To determine the maximum operating frequency of the circuit, we increase the clock frequency in steps and check the output and see whether the correct output is obtained at all the rising edge in the entire cycle, for that particular frequency. The simulations done for clock period, 5.5ns, 4ns and 3.5ns are shown in Fig. 9, Fig. 10 and Fig. 11 respectively. The minimum clock time period, T_{Cmin}, of the 8 bit multiplier circuit with pipelining, as extracted using the schematic is 4ns while that using the RC extracted netlist is 8ns. Hence, the maximum operating frequency,

$$f = \frac{1}{TCmin} \qquad (2)$$

is 167MHz and 125MHz as extracted from the schematic and netlist respectively.

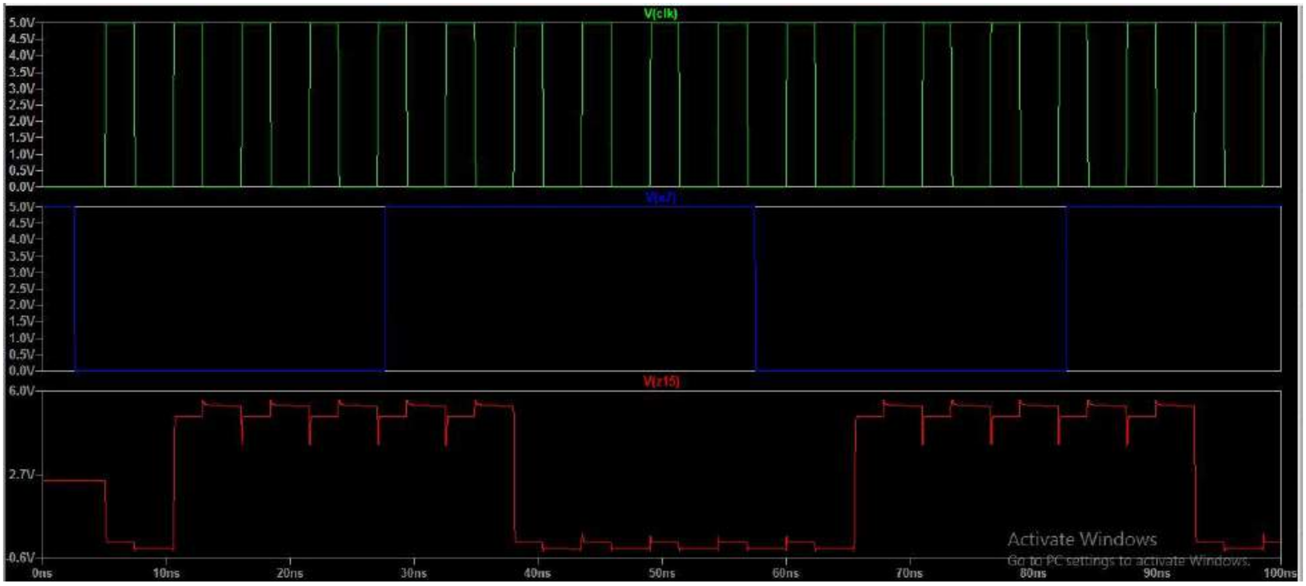


Figure 15 Schematic simulation output for clock period = 5.5ns

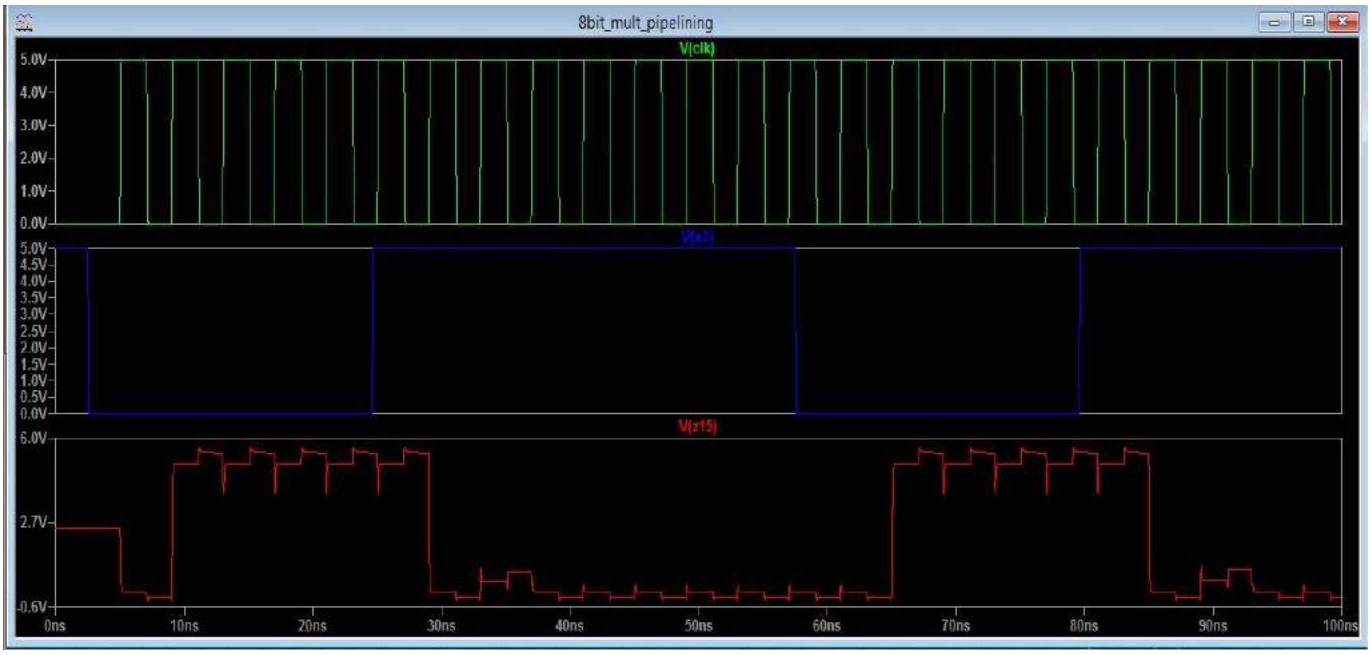


Figure 16 Schematic simulation output for clock period = 4ns

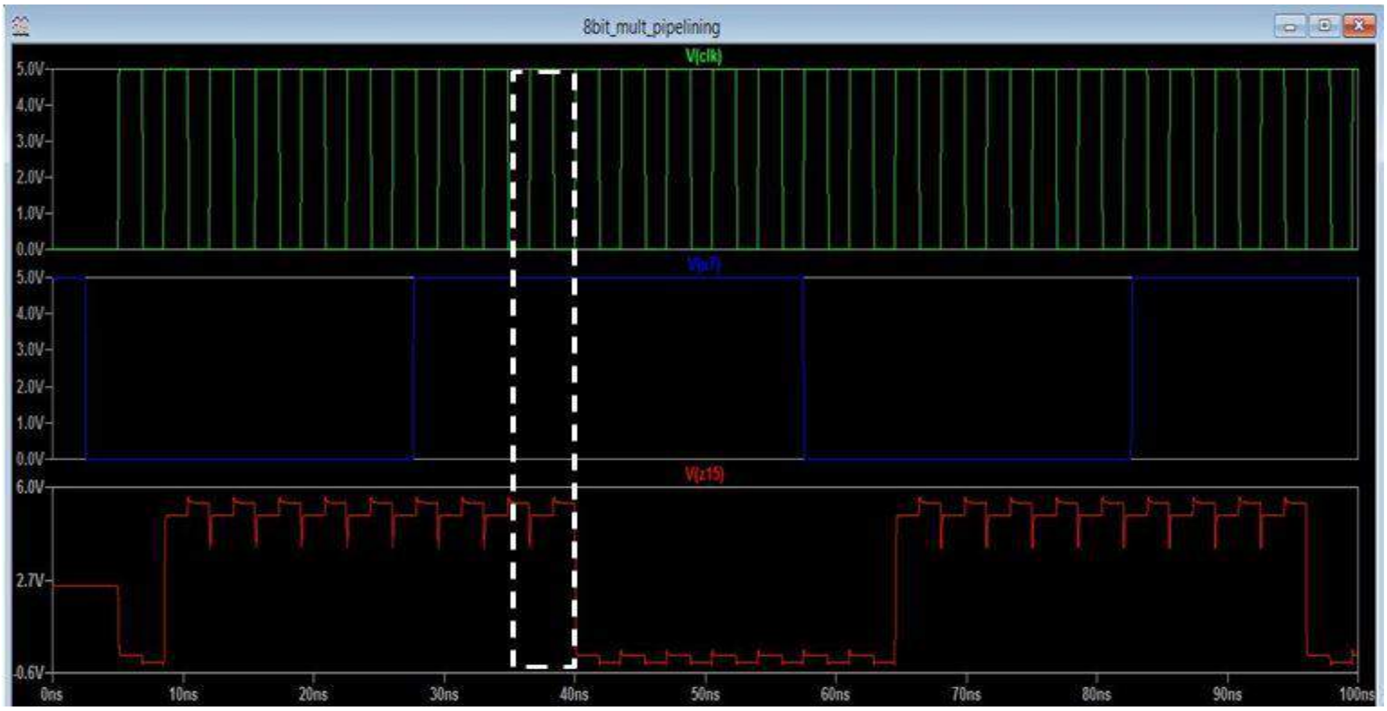


Figure 17 Schematic simulation for clock period=3.5ns(erroneous output highlighted)

4. Input Combinations Tested

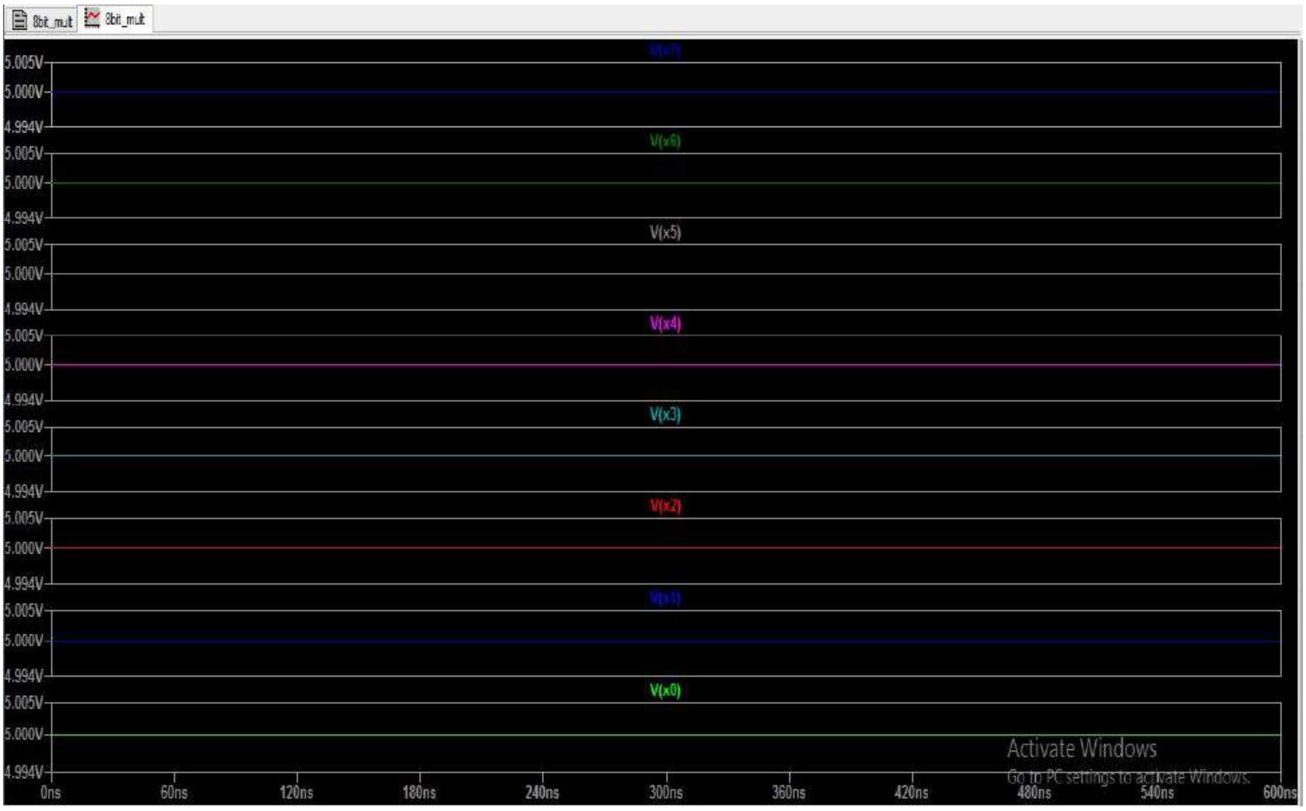


Figure 18 X = 1111 1111

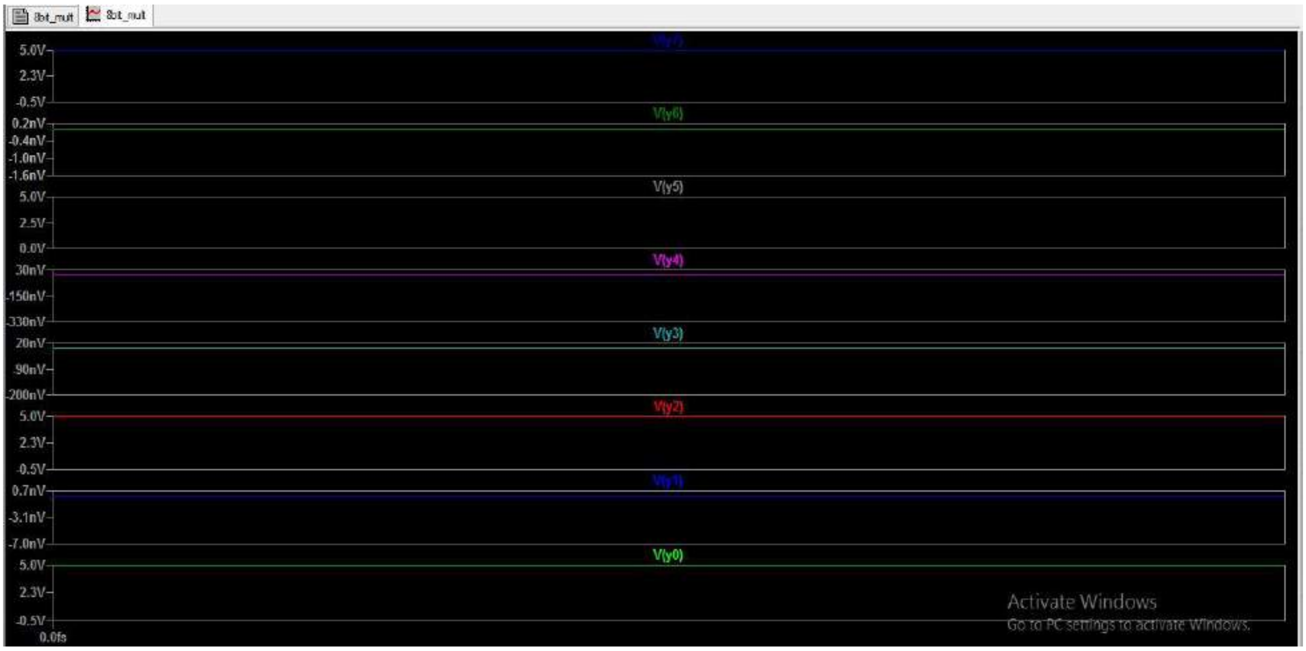


Figure 19 Y = 1010 0101

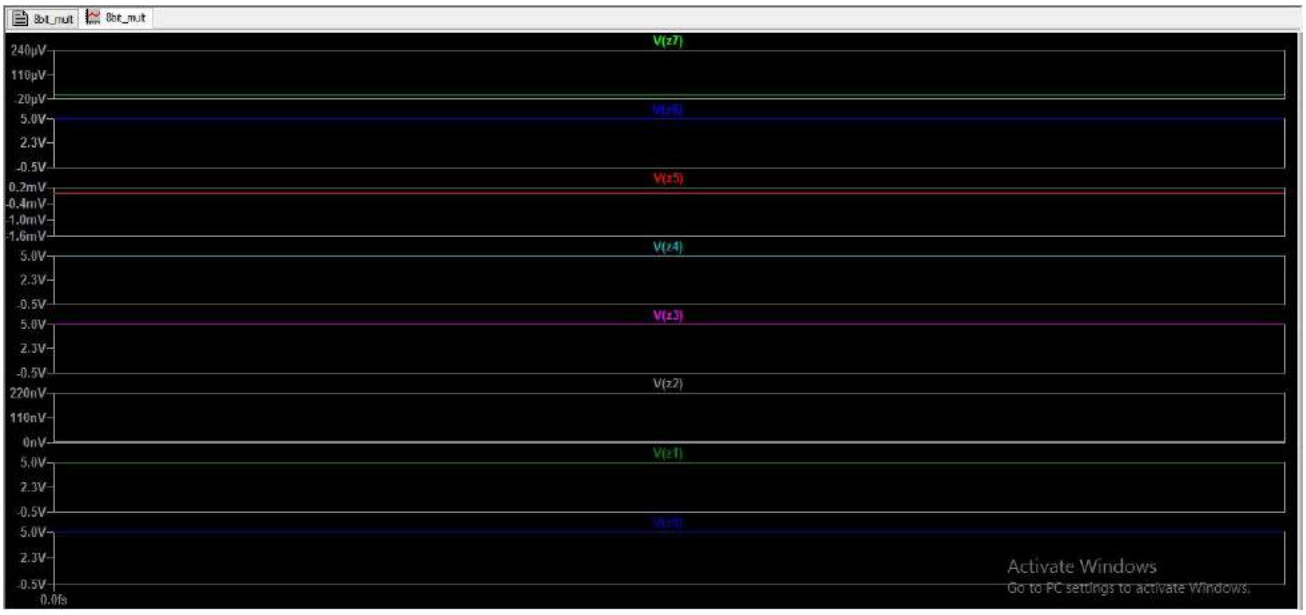


Figure 20 Z7-0 = 0101 1011

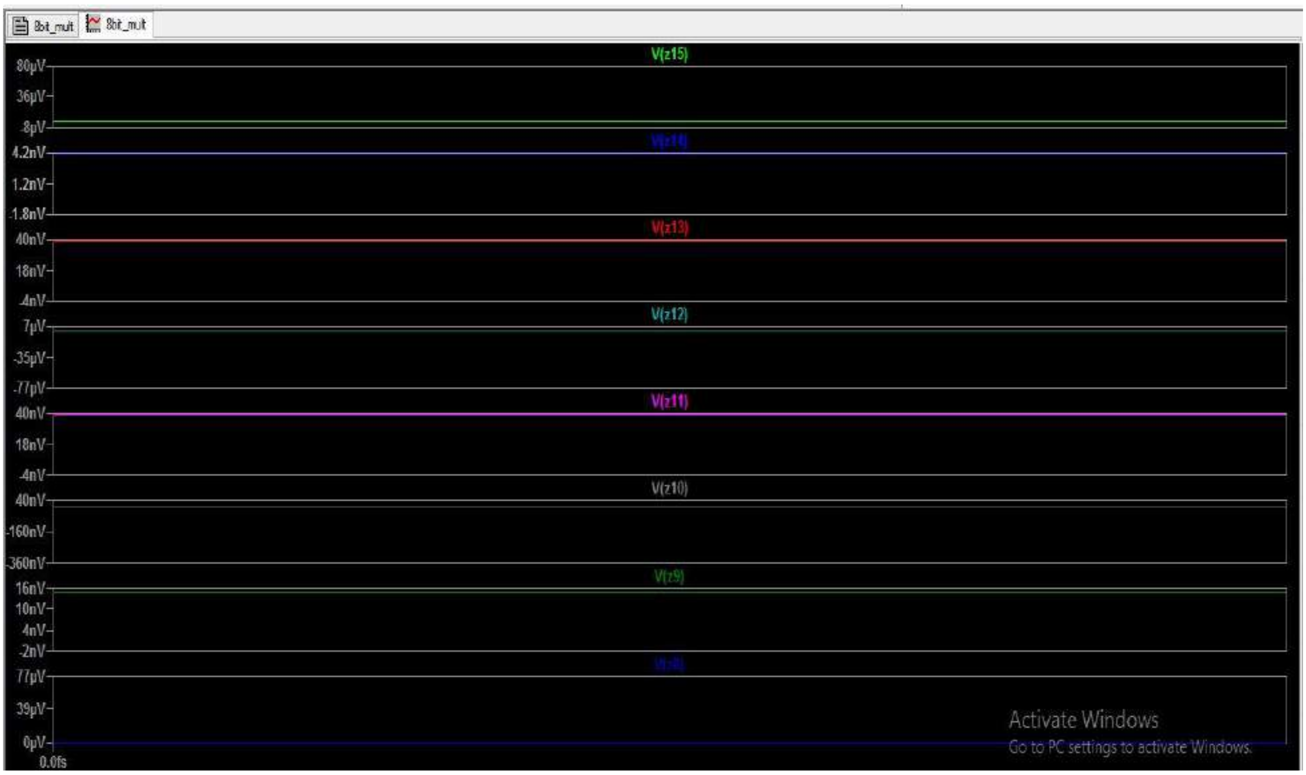


Figure 21 Z8-Z15 = 0000 0000

5. APPENDIX

5.1 8bit Multiplier_Propagation delay Code

```
vdd vdd 0 DC 5
vd0 V0 0 DC 0
vd1 V1 0 DC 5
vX0 X0 0 PULSE(0 5 2.5n 0.1n 0.1n 25n 55n)
vX1 X1 0 PULSE(0 5 2.5n 0.1n 0.1n 25n 55n)
vX2 X2 0 PULSE(5 0 2.5n 0.1n 0.1n 25n 55n)
vX3 X3 0 PULSE(0 5 2.5n 0.1n 0.1n 25n 55n)
vX4 X4 0 PULSE(0 5 2.5n 0.1n 0.1n 25n 55n)
vX5 X5 0 PULSE(5 0 2.5n 0.1n 0.1n 25n 55n)
vX6 X6 0 PULSE(0 5 2.5n 0.1n 0.1n 25n 55n)
vX7 X7 0 PULSE(5 0 2.5n 0.1n 0.1n 25n 55n)
vY0 Y0 0 DC 5
vY1 Y1 0 DC 0
vY2 Y2 0 DC 5
vY3 Y3 0 DC 0
vY4 Y4 0 DC 0
vY5 Y5 0 DC 5
vY6 Y6 0 DC 0
vY7 Y7 0 DC 5
.measure tran tf trig v(X7) val=2.5 fall=1 td=0 targ v(Z15) val=2.5 rise=1
.tran 600ns
```

5.2 8bit Multiplier without D flip flop minimum clock Code

```
vdd vdd 0 DC 5
vd0 V0 0 DC 0
vd1 V1 0 DC 5
vb clk 0 PULSE(0 5 5n 0.1n 0.1n 3n 6n)
vc nclk 0 PULSE(5 0 5n 0.1n 0.1n 3n 6n)
vX0 X0 0 PULSE(0 5 2n 0.01n 0.01n 23n 55n)
vX1 X1 0 PULSE(0 5 2n 0.01n 0.01n 23n 55n)
vX2 X2 0 PULSE(5 0 2n 0.01n 0.01n 23n 55n)
vX3 X3 0 PULSE(0 5 2n 0.01n 0.01n 23n 55n)
vX4 X4 0 PULSE(0 5 2n 0.01n 0.01n 23n 55n)
vX5 X5 0 PULSE(5 0 2n 0.01n 0.01n 23n 55n)
vX6 X6 0 PULSE(0 5 2n 0.01n 0.01n 23n 55n)
vX7 X7 0 PULSE(5 0 2n 0.01n 0.01n 23n 55n)
```

```
vY0 Y0 0 DC 5
vY1 Y1 0 DC 0
vY2 Y2 0 DC 5
vY3 Y3 0 DC 0
vY4 Y4 0 DC 0
vY5 Y5 0 DC 5
vY6 Y6 0 DC 0
vY7 Y7 0 DC 5
.tran 100ns
```

5.3 8bit Multiplier with pipeline minimum clock Code

```
vdd vdd 0 DC 5
vd0 V0 0 DC 0
vd1 V1 0 DC 5
vb clk 0 PULSE(0 5 5n 0.1n 0.1n 1.975n 3.95n)
vc nclk 0 PULSE(5 0 5n 0.1n 0.1n 1.975n 3.95n)
vX0 X0 0 PULSE(0 5 2n 0.01n 0.01n 23n 55n)
vX1 X1 0 PULSE(0 5 2n 0.01n 0.01n 23n 55n)
vX2 X2 0 PULSE(5 0 2n 0.01n 0.01n 23n 55n)
vX3 X3 0 PULSE(0 5 2n 0.01n 0.01n 23n 55n)
vX4 X4 0 PULSE(0 5 2n 0.01n 0.01n 23n 55n)
vX5 X5 0 PULSE(5 0 2n 0.01n 0.01n 23n 55n)
vX6 X6 0 PULSE(0 5 2n 0.01n 0.01n 23n 55n)
vX7 X7 0 PULSE(5 0 2n 0.01n 0.01n 23n 55n)
vY0 Y0 0 DC 5
vY1 Y1 0 DC 0
vY2 Y2 0 DC 5
vY3 Y3 0 DC 0
vY4 Y4 0 DC 0
vY5 Y5 0 DC 5
vY6 Y6 0 DC 0
vY7 Y7 0 DC 5
.tran 100ns
```