8 BIT MULTIPLIER DESIGN WITH AND WITHOUT PIPELINING

EE5311 Digital IC Design [illegible]

Group-35

V. Kalavathi(EE20M043)

P. Sai Sumanth(EE20M050)

D. Maheswara Reddy(EE20M045)

Surya Prakash Yadav

INTRODUCTION:

The carry-save array multiplier uses an array of carry-save adders for the accumulation of partial products. It uses a carry-propagate adder for the generation of the final product. This reduces the critical path delay of the multiplier since the carry-save adders pass the carry to the next level of adders rather than the adjacent ones. Figure 1 shows the structure of the carry save array multiplier.

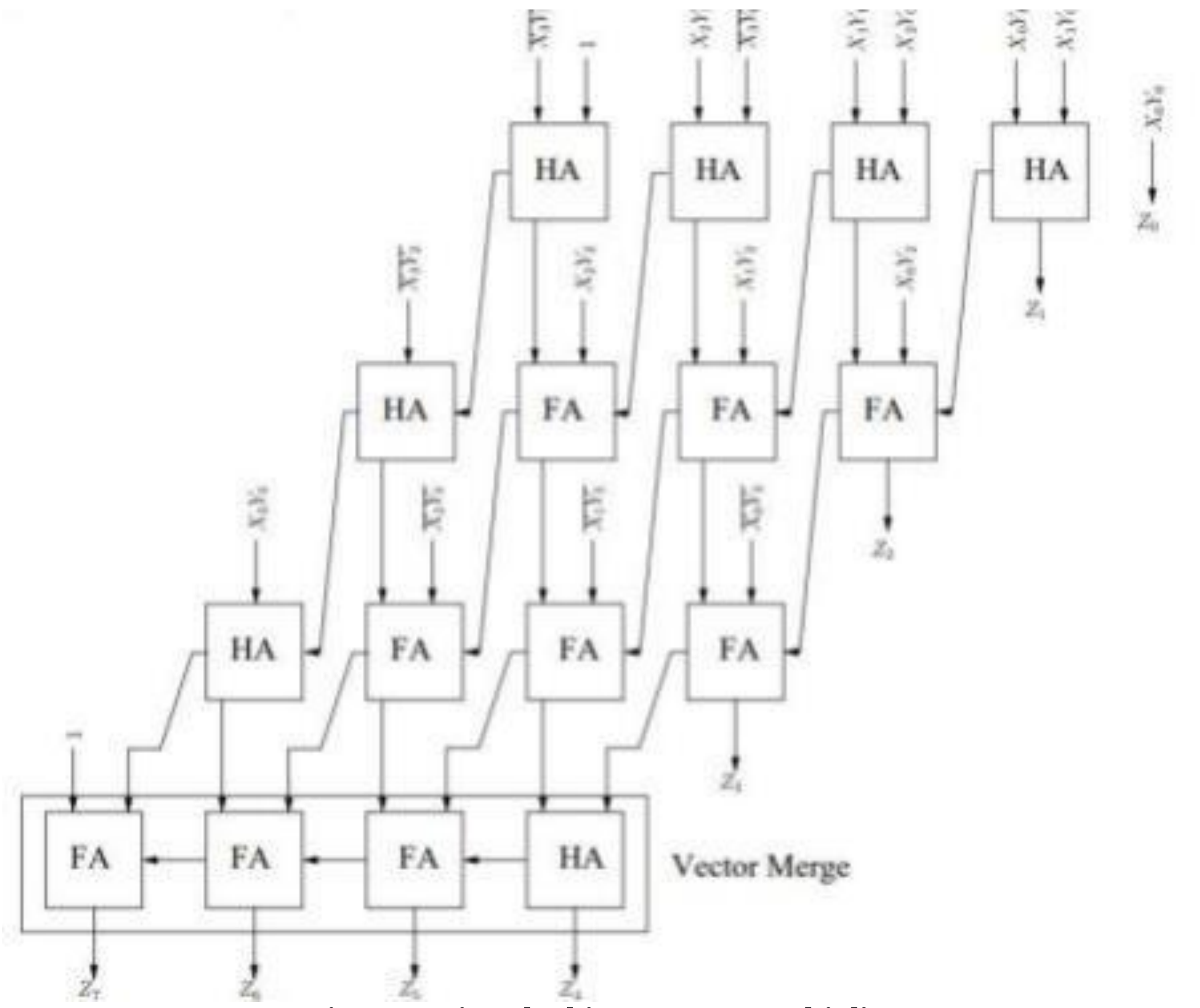


Figure 1: Signed 4-bit carry save multiplier

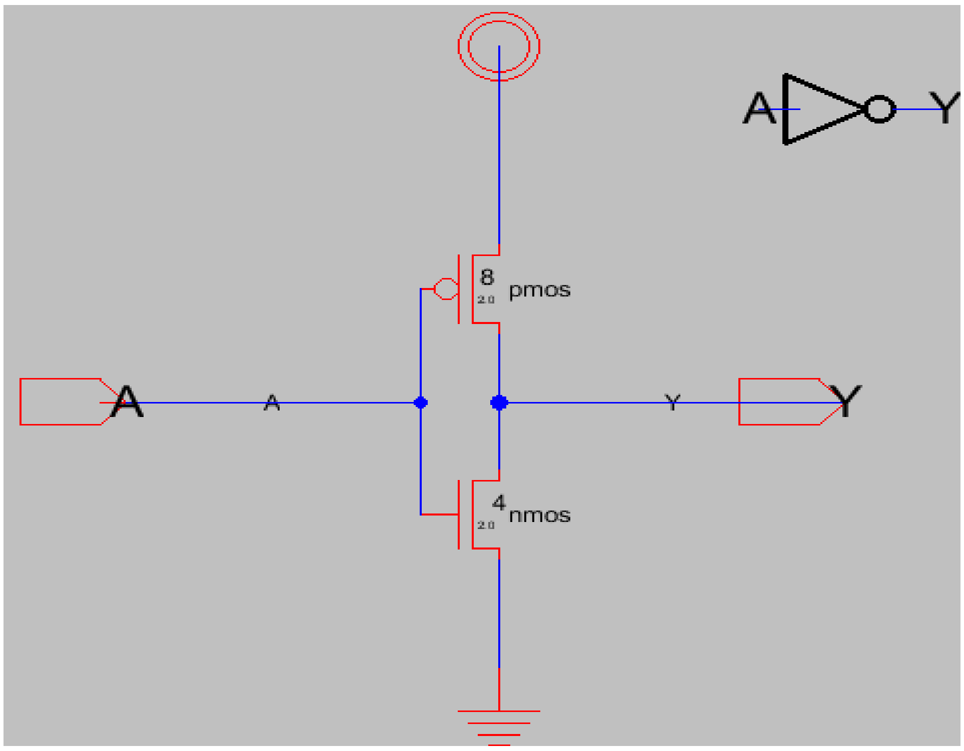
Working of 8 bit signed multiplier:

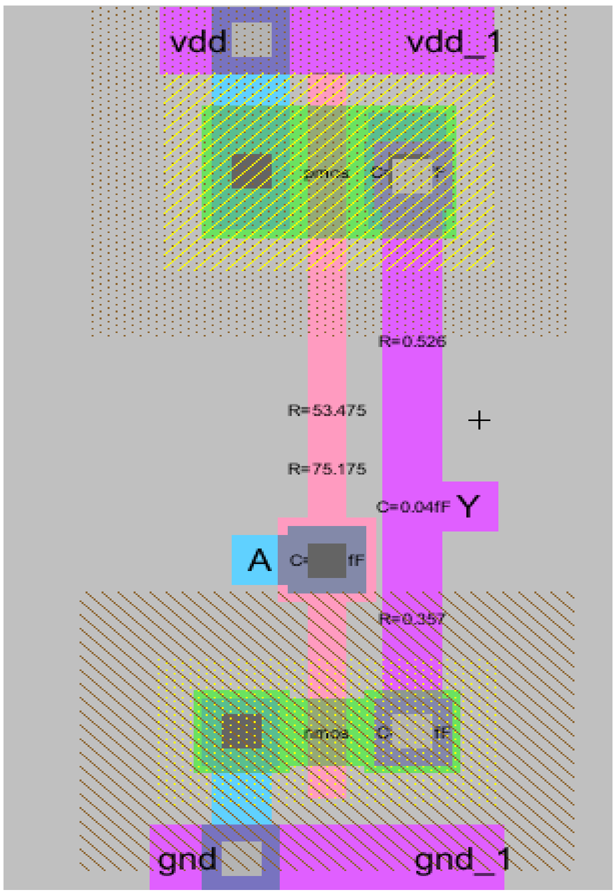
We are using Inverting full adders and nand for 8-bit signed multipliers. We are using inverting full adders since sum and carry have mirroring property. If we give inverted input to inverting full adder, we will get sum and carry as output. Similarly, if we give normal input, we will get inverted output. Now if we observe our multiplier at the first stage (odd stages – Z1, Z3, Z5, Z7) we are giving inverted input and we will get sum, so there is no need of applying inverter at the sum output. Now if we again observe our multiplier at second stage (even stages - Z2, Z4, Z6) we are giving normal input and we will get inverted sum, so there is need of applying inverter at the sum output. Here at every even stage of full adder there is a need of inverter at output for sum (Z2, Z4, Z6).

From Z8 to Z15 we get output from final vector merge ripple adder and the ripple adder inputs are from odd stage of inverted full adders. The inputs to vector merge stage will be normal, that will yield to inverted sum and inverted carry, therefore there is a need of inverter at sum as well as inverter at carry before rippling carry to another inverting full adder.

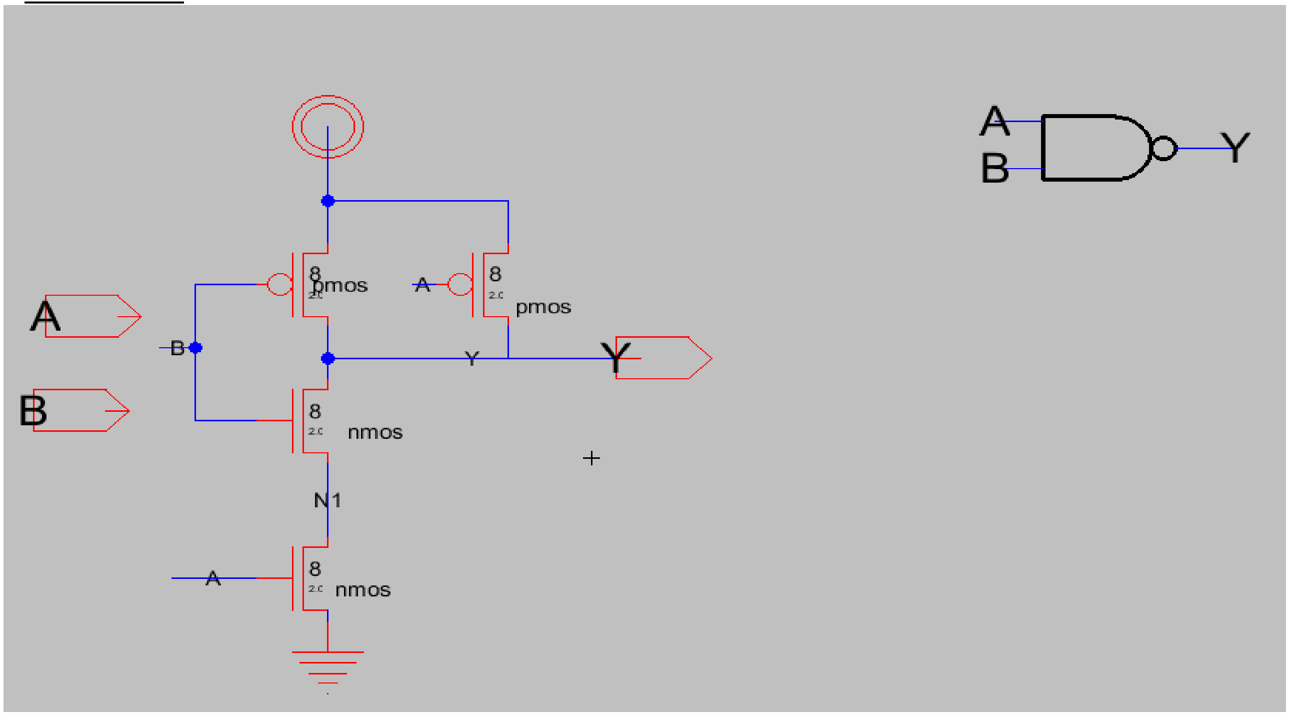
Some basic gates used in this project are

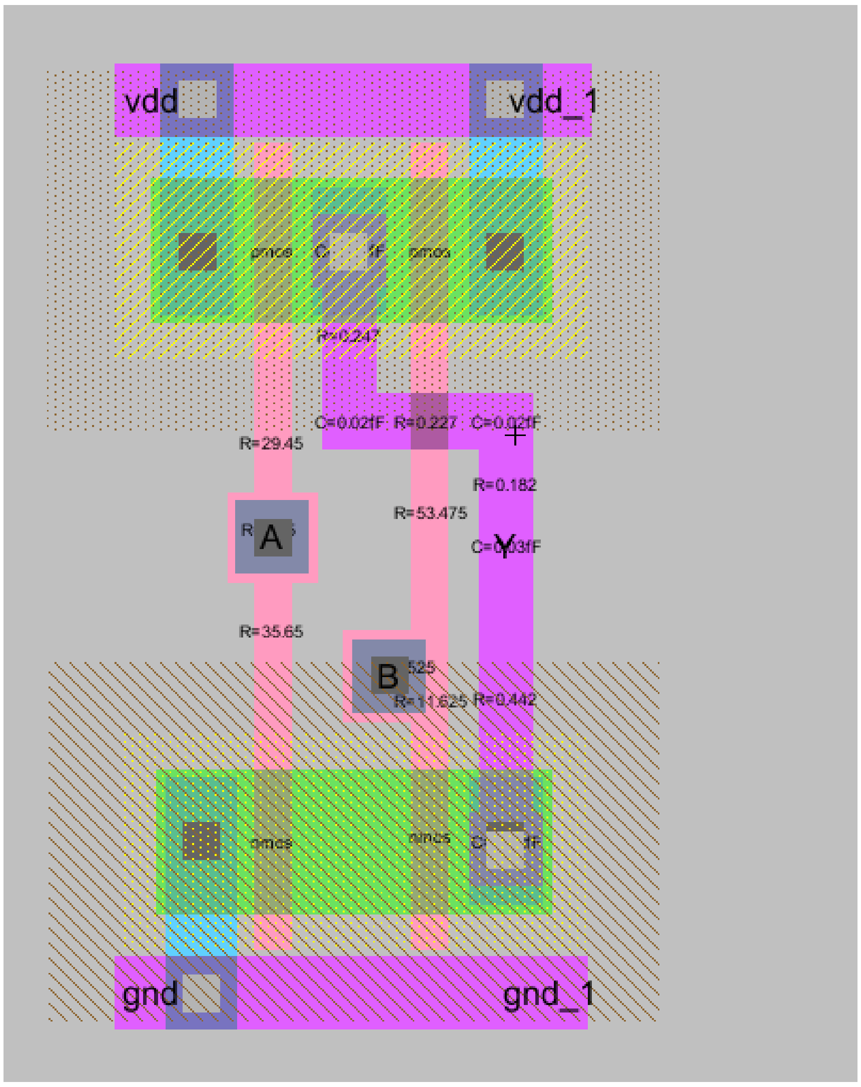
N<u>OT gate:</u>





NAND Gate:

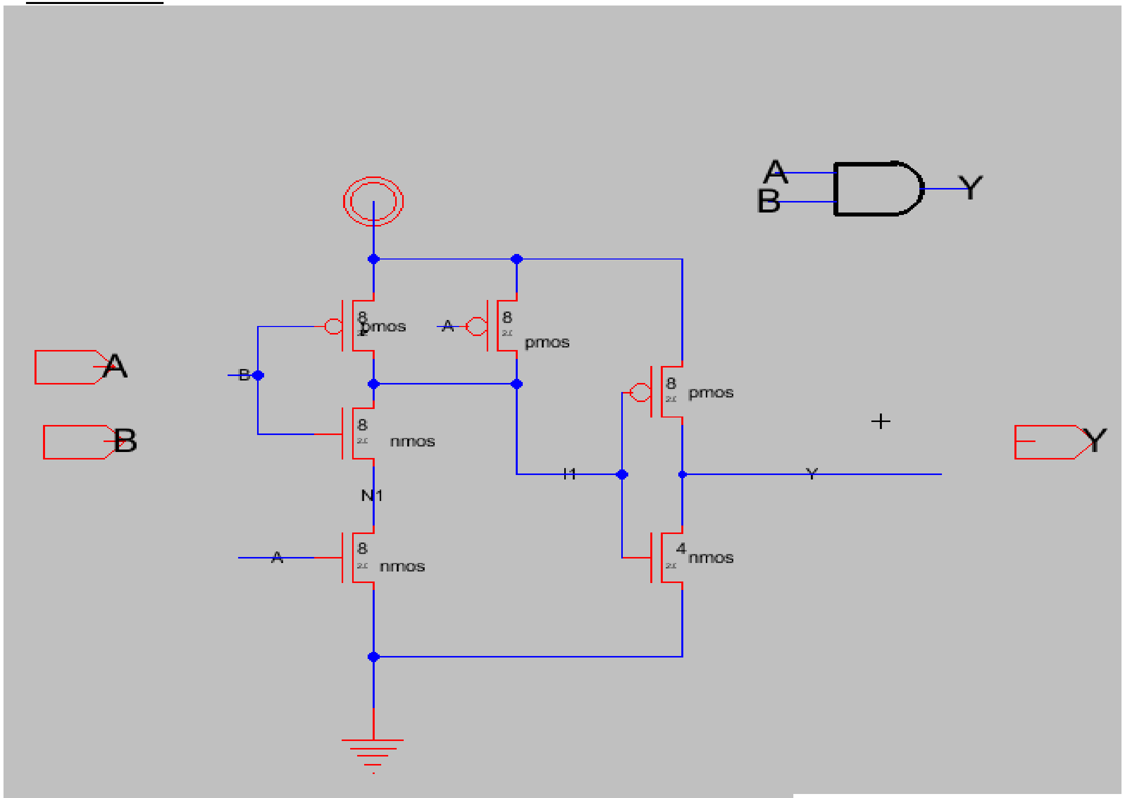


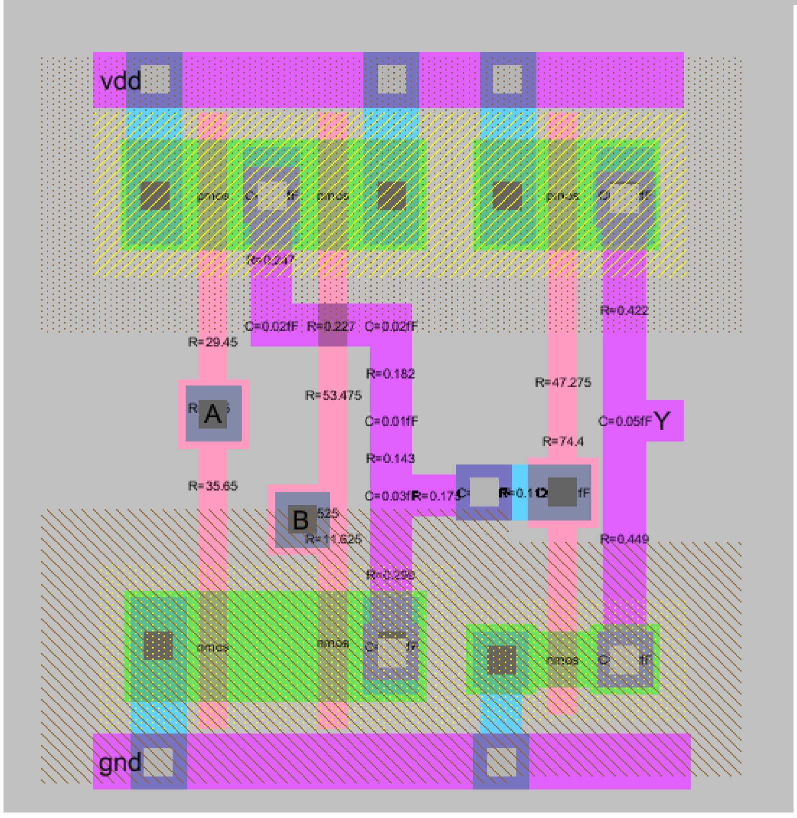


Used in CSM (to give inputs to Full adders) at alternate stages of CSM

NAND gate schematic and layout

AND Gate:



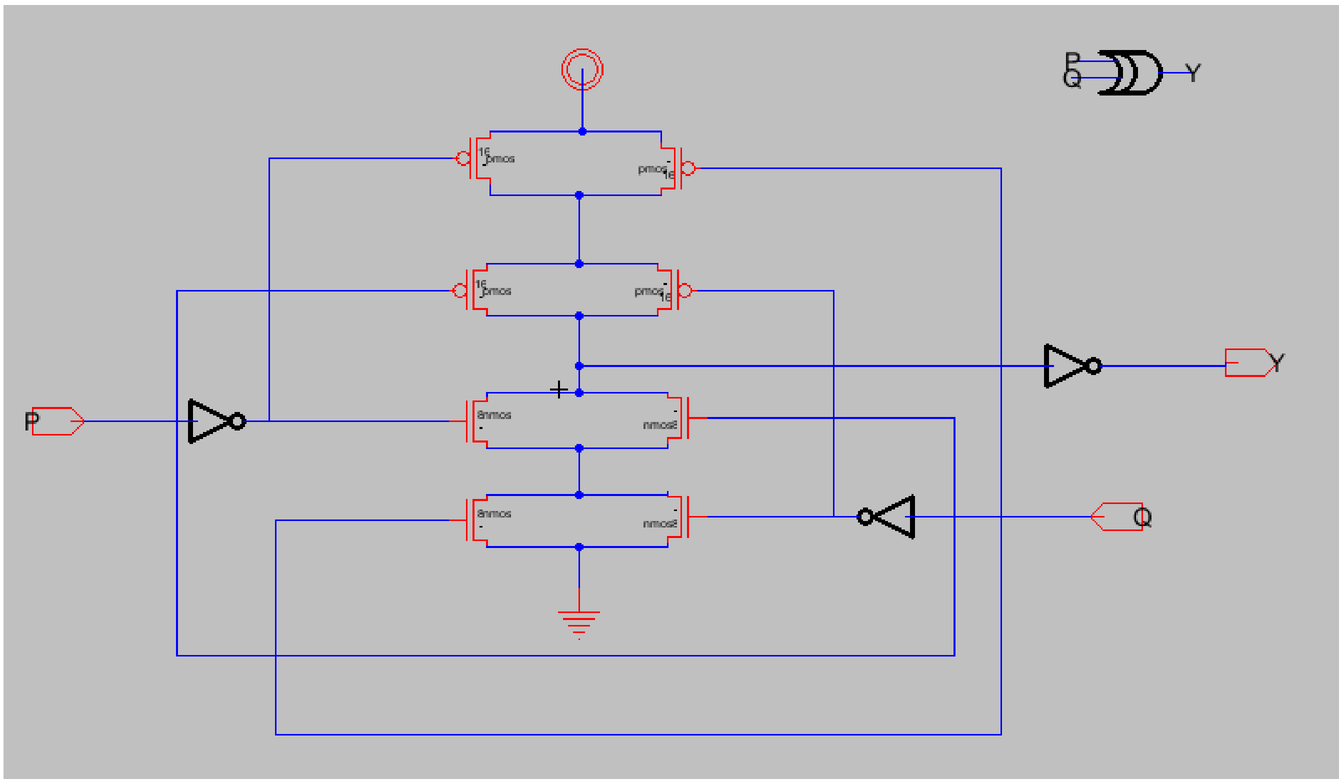


Used in CSM and in Carry look ahead adder (for generate (Gi) signal

XOR gate

In order to calculate sum using carry look ahead adder S = **A** (XOR) **B** (XOR) **C** we need XOR gate.

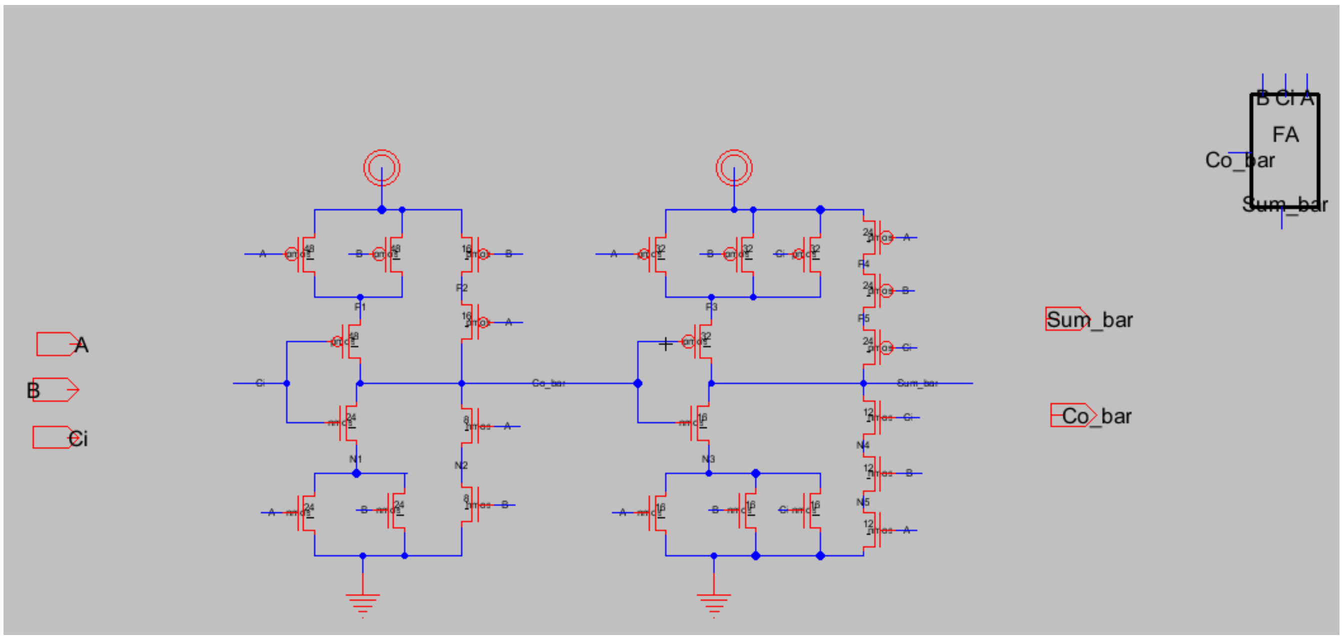
The schematic of XOR gate designed in this project is shown below

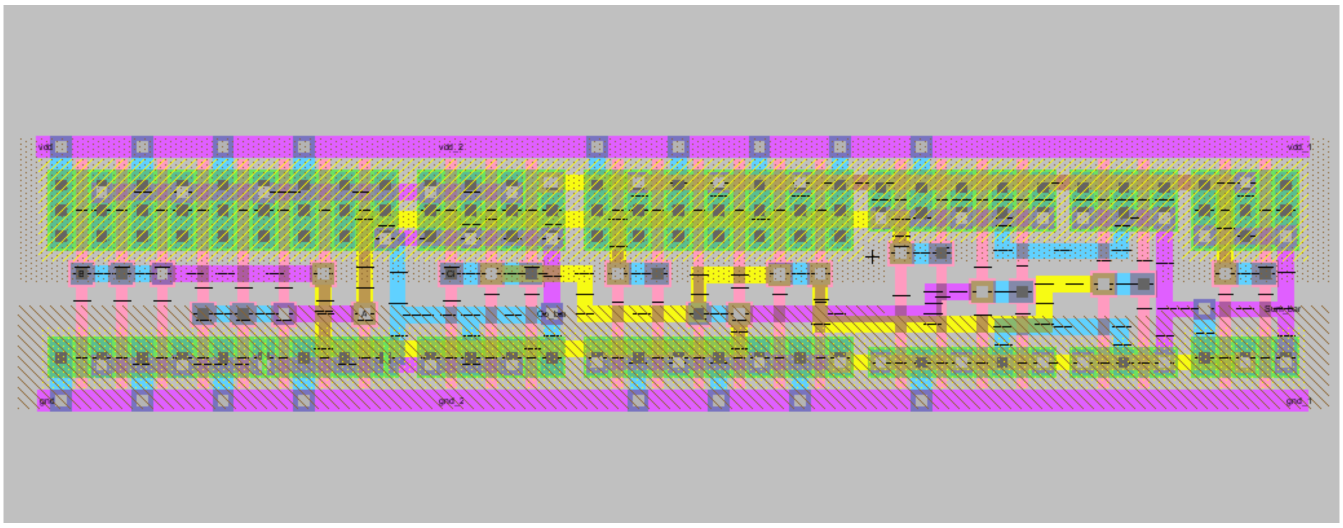


Sizing is done as per unit reference inverter.

(***** as CLA layout is not needed, XOR gate layout is not done)

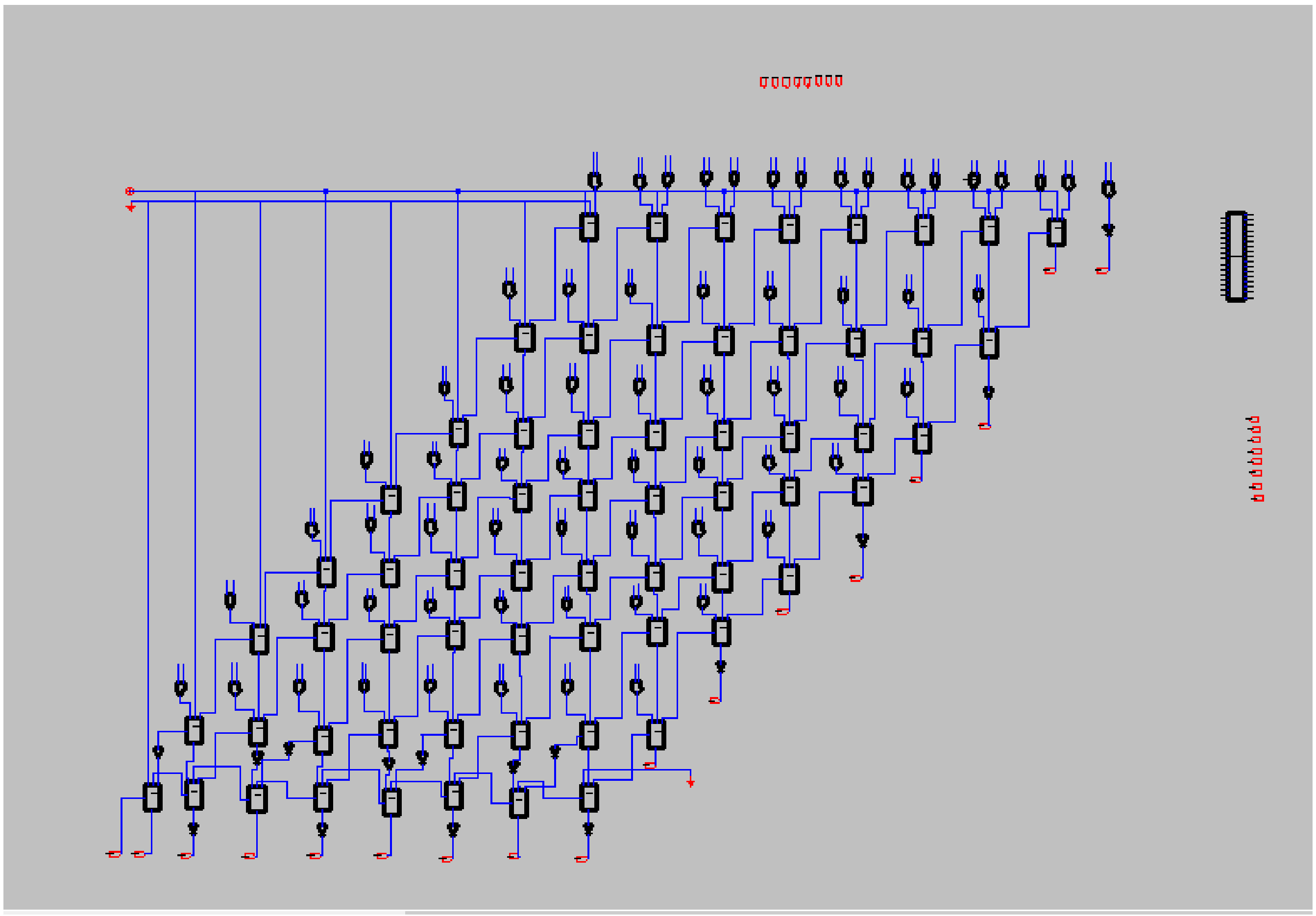
Full adder block layout





The forms the a sub block in the CSM

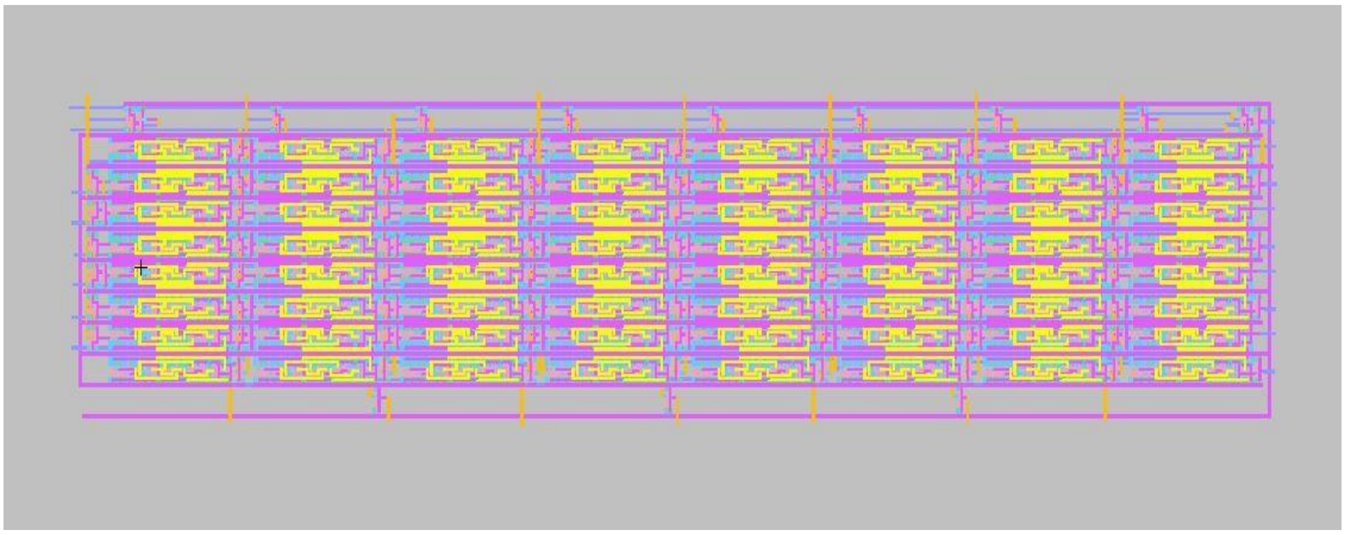
CSM (Carry Save Multiplier):

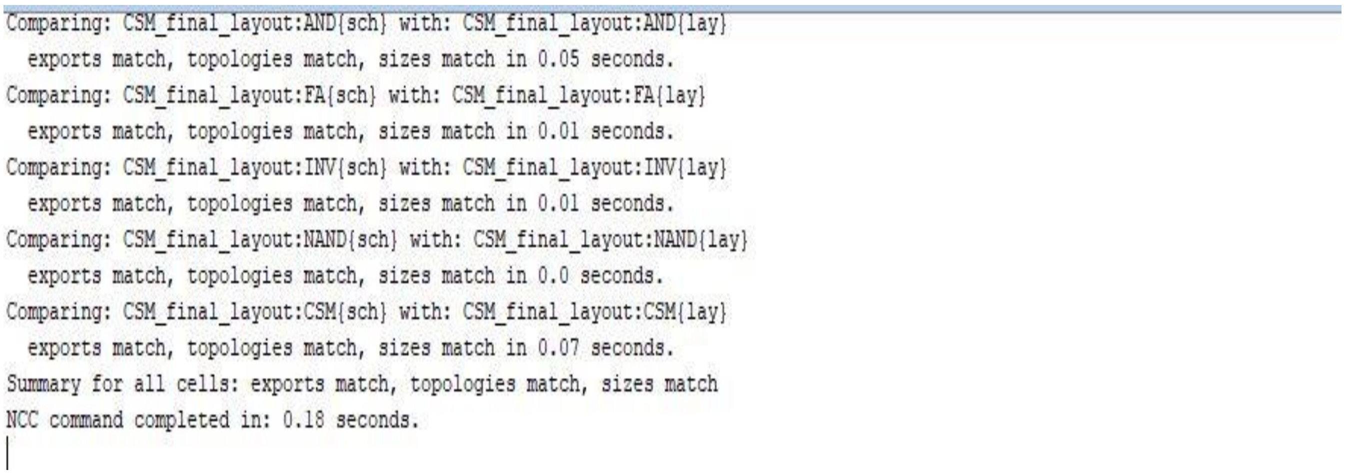


CSM schematic

- Complemented inputs are given at alternate stages to take advantage of **mirror implemented adder** schematic.(**this is done to avoid usage of inverters in between so that delay is optimized**)
- Last stage is vector merge stage implemented using **ripple carry adder** this generates the carry out that is required to ensure the sign of the result of multiplication.

CSM Layout





```
Comparing: CSM_final_layout:AND{sch} with: CSM_final_layout:AND{lay}
  exports match, topologies match, sizes match in 0.05 seconds.
Comparing: CSM_final_layout:FA{sch} with: CSM_final_layout:FA{lay}
  exports match, topologies match, sizes match in 0.01 seconds.
Comparing: CSM_final_layout:INV{sch} with: CSM_final_layout:INV{lay}
  exports match, topologies match, sizes match in 0.01 seconds.
Comparing: CSM_final_layout:NAND{sch} with: CSM_final_layout:NAND{lay}
  exports match, topologies match, sizes match in 0.0 seconds.
Comparing: CSM_final_layout:CSM{sch} with: CSM_final_layout:CSM{lay}
  exports match, topologies match, sizes match in 0.07 seconds.
Summary for all cells: exports match, topologies match, sizes match
NCC command completed in: 0.18 seconds.
```

- LVS clean result is shown above

Pipelining of an 8 bit signed multiplier:

In order to do pipelining we use D Flip flop as a register.

<u>Construction of D Flip-flop:</u>

- D Flip flop is constructed with two latches which are low level triggered makes the flop as a positive edge triggered one.

Figure.1 D flip-flop schematic

- Its working is similar to a master slave configuration, in which slave(second) latch gets an inverted clock signal compare to master(first)latch.
- Sizes of the transmission gate is same for both pMOS and nMOS

<u>Characterisation of D flip-flop:</u>

- **Setup time** for the flop is calculated as by varying the input transition very close to the clock edge and observing the transition at the output.
- Whenever there is a 5% rise (as per RABEY text book) in the Tcq delay, value is noted down. And the observed values as follows

For 0 to 1 rising (at the output) = 37.68ps

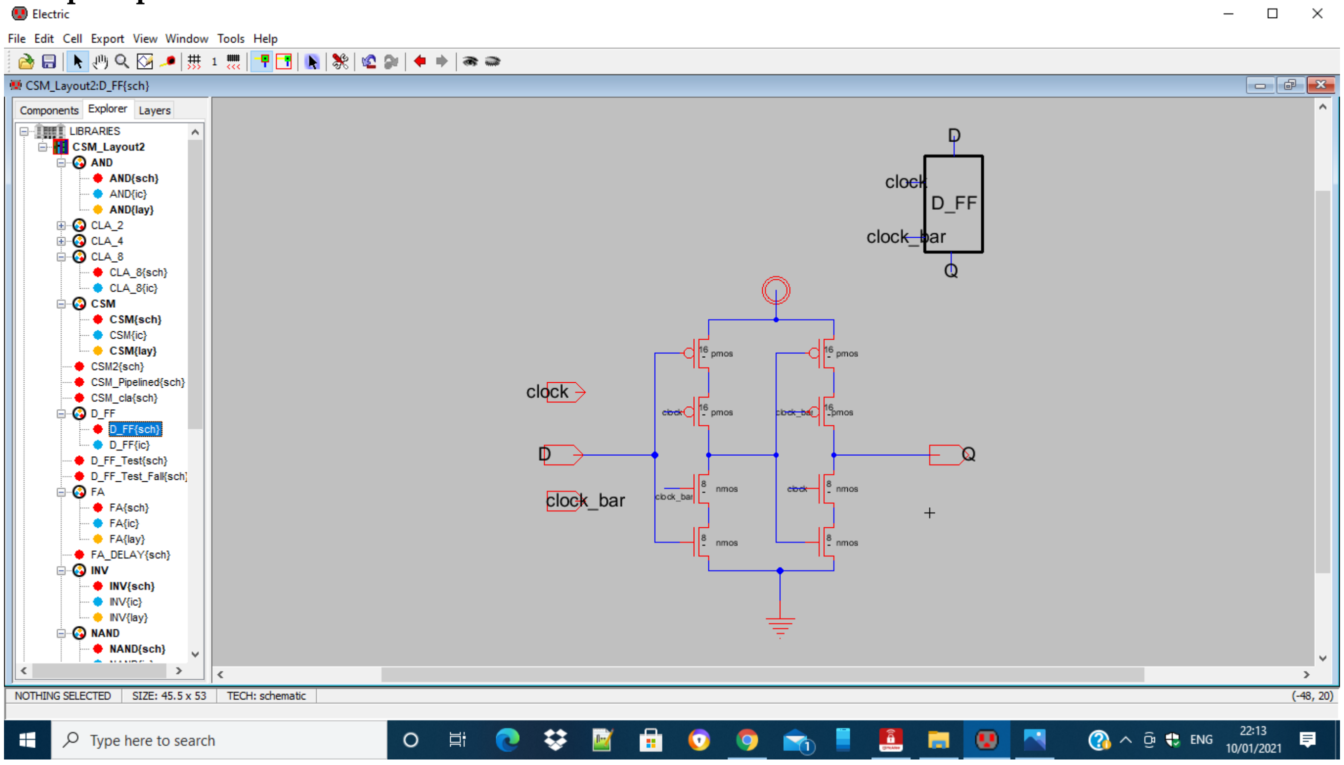
For 1 to 0 falling (at the output) =32.1335ps

Clock to Q (output) delay (T_{cq} value):

(Here it is calculated with input settled much before clock edge arrives)

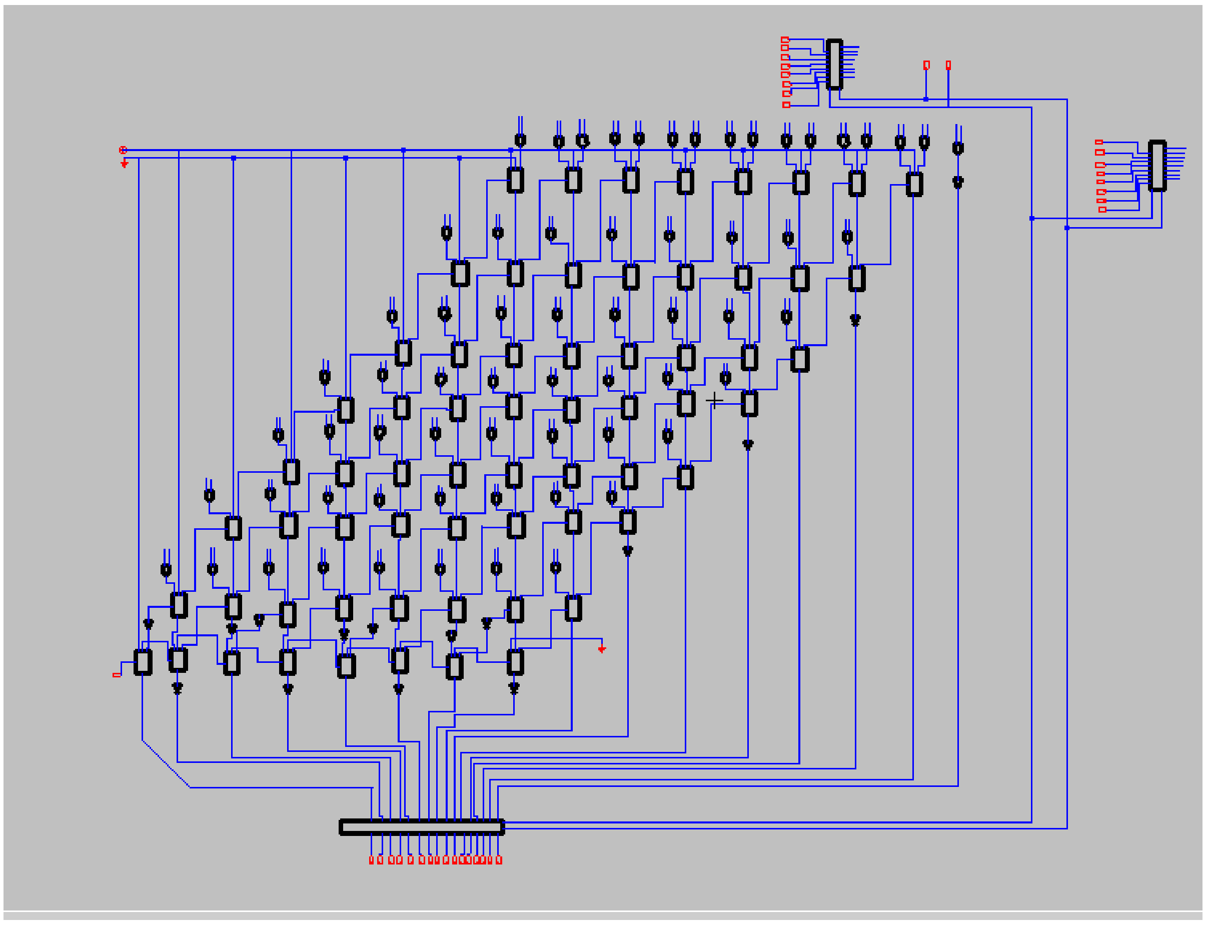
The delay value is found to 66.191ps by measuring the input to output (of the D flip-flop) $V_{dd}/2$ crossings.

D flip-flop


Electric
File Edit Cell Export View Window Tools Help
CSM_Layout2:D_FF{sch}
Components Explorer Layers
LIBRARIES
CSM_Layout2
AND
AND{sch}
AND{ic}
AND{lay}
CLA_2
CLA_4
CLA_8
CLA_8{sch}
CLA_8{ic}
CSM
CSM{sch}
CSM{ic}
CSM{lay}
CSM2{sch}
CSM_Pipelined{sch}
CSM_cla{sch}
D_FF
D_FF{sch}
D_FF{ic}
D_FF_Test{sch}
D_FF_Test_Fall{sch}
FA
FA{sch}
FA{ic}
FA{lay}
FA_DELAY{sch}
INV
INV{sch}
INV{ic}
INV{lay}
NAND
NAND{sch}
D
clock
D_FF
clock_bar
Q
pmos
pmos
pmos
pmos
nmos
nmos
nmos
nmos
clock
D
clock_bar
Q
NOTHING SELECTED
SIZE: 45.5 x 53
TECH: schematic
(-48, 20)
Type here to search
ENG
22:13
10/01/2021

.

Finding max frequency of operation (without pipelining)



For the calculation of delay,

Static Timing Analysis:

Using Static Timing Analysis, for a single full adder block, tsum = 67.79ps, tcarry = 47.252ps and and gate delay is tand = 28.7021

Delay of combinational circuit is tcomb = 901.6995ps

Minimum time period is T = tsetup + tcq + tcomb

Minimum time period T = 1.005nsec

Therefore the maximum frequency of operation is 0.995GHz

If we use layout extracted Rc values for the delay

Minimum time period is T = 1.2938nsec

So the delay here is 1.2938 times the delay of the schematic.

Simulated values:

But in the simulation the propogation delay of the Carry Save Multiplier is tcomb = **302.01psec**

The variation is due to the inputs we have provided. While doing STA we have provided inputs to the input A of the full adder which gives the worst case delay but in the CSM for fast execution we have given inputs to the input C of the Full Adder.

The Minimum time period is T = tcomb + tsetup + tcq

T = 405.8819psec

So the **Maximum frequency of operation** in the simulation is **2.46GHz**

In the same way if we use RC extracted netlist the delay is found to be

Minimum time period is T = tcomb + tsetup + tcq

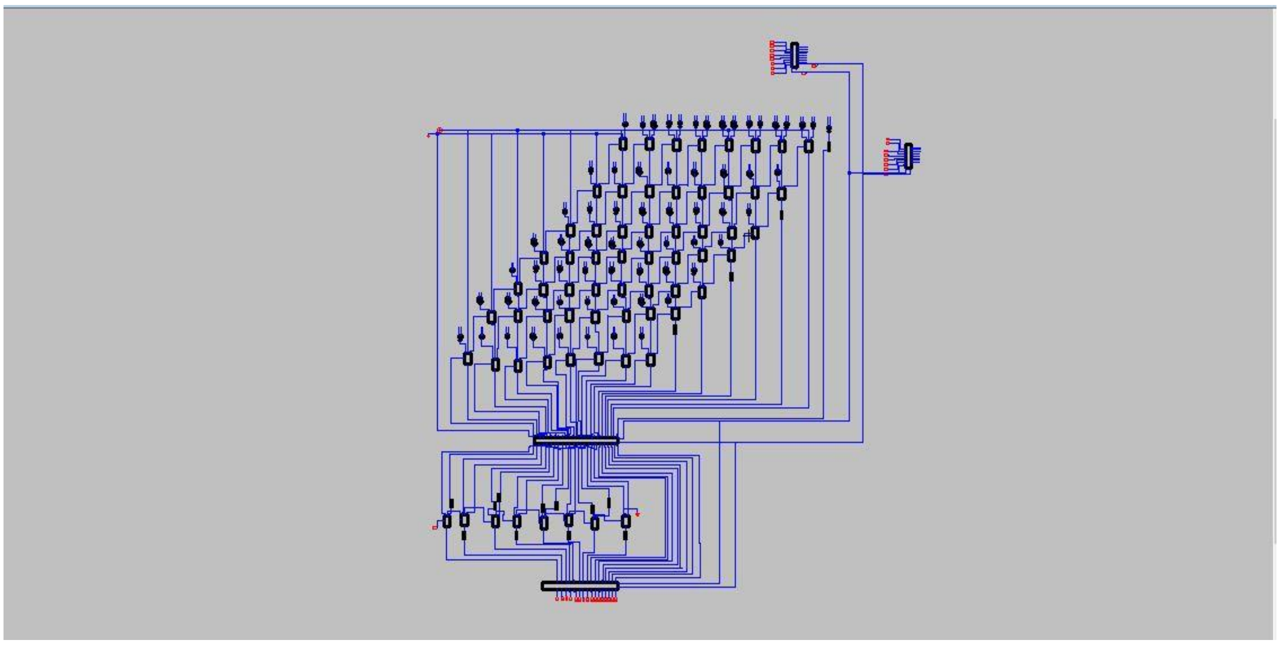
T = 425.687 + 66.191 + 37.6809

T = 529.5589

So the **Maximum frequency of operation** in the simulation is **1.883GHz**

So the delay here is **1.48 times** the delay of the schematic.

CSM (with pipelining):



By introducing a flip-flop just **above the vector merge stage** so that the critical path can be broken in such a way so as to increase the frequency of operation.

Using the Static Timing Analysis T = 674.8052

So the delay is 1.48 times the delay of the schematic without pipelining.

In the Simulation the delay is found to be

CARRY LOOK AHEAD ADDER:

Sum = Pi (XOR) Cin

Carry = Gi + Pi (XOR) Cin
Here Cin is the input carry to entire circuit and Ai and Bi are the input to adders at ith bit sum calculation.

Where Pi is propagate signal = Ai (XOR) Bi

Gi is generate signal = Ai (AND) Bi

Similarly, for structure is same for 3,4,5,6 bits adder.

FUNCTIONALITY

Input given :

A: -2 B: -2 and now A change to 1 A : 11111110 B: 11111110 Z (A*B): 00000100 A -> 00000001 B: 11111110 Z:11111110

As here in the above 2nd test case as both x0 and z15 is changing its value, we can treat that as the worst case input test case.

Summary of results

	Without pipelining	With pipelining	Conclusion
Max frequency	2.46GHz	6.188GHz	Frequency increased by 2.51 times
AREA of CSM layout	2381 x 506	-----------	----------
No .of test patterns	2	2	----------
Architecture of vector merge	Ripple carry adder, CLA	Ripple Carry Adder CLA	Frequency can be increased

Schematic delay VS Extracted Netilist delay

Schematic delay	Extracted Netlist delay	Conclusion
405.8819psec	529.5589psec	Extracted Netlist delay is 1.48 times the schematic delay

***note all time units in this report are expressed in terms of ps (Pico seconds).