Design of 8-bit Signed Pipelined Carry Save Multiplier
Course Project: EE5311 Digital IC Design

Arjun Menon V   Srinivas R   Ashwanth S
EE18B104   EE18B136   EE18B040
Electrical Engineering
IIT Madras

August- November, 2021

The drive folder containing the libraries for the designs can be accessed using this link.

1 Summary

SUMMARY	
Maximum Operating Frequency (w/o pipelining)	2.062 GHz(RC extracted = 1.64 GHz)
Maximum Operating Frequency (with pipelining)	3.704 GHz(RC extracted = 2.618 GHz)
Area of Layout	111.251μm^2(L*W=919,432)
Aspect Ratio	1:5.50

Table 1: Summary of Carry Save Multiplier characteristics

FUNCTIONAL SIMULATION RESULTS		
X	**Y**	**Z**
0000 1001 (9)	0000 1111 (15)	0000 0000 1000 0111 (135)
1111 0110 (-10)	0000 1100 (12)	1111 1111 1000 1000 (-120)
1010 1001 (-87)	1001 1011 (-101)	0010 0010 0101 0011 (8787)
1111 1111 (-1)	1111 1111 (-1)	0000 0000 0000 0001 (1)
0111 1111 (127)	1000 0000 (-128)	1100 0000 1000 0000 (-16256)
0011 1111 (63)	1110 0100 (-28)	1111 1001 0001 1100 (-1764)

Table 2: Summary of Functional Simulation of Carry Save Multiplier

2 Introduction

A carry-save multiplier (schematic + layout) is designed using the cells designed in assignments 1-4. In order to minimise both delay and area occupied by the layout of the CSM, the following cell sizes are used:

- **INV:** 1×
- **NAND:** 1×
- **Full Adder:** Carry-Out- 3×, Sum-generation- 1×

To simplify Layout v/s Schematic checks and make the design process modular, the CSM schematic was split into the following units that are built using the above cells:

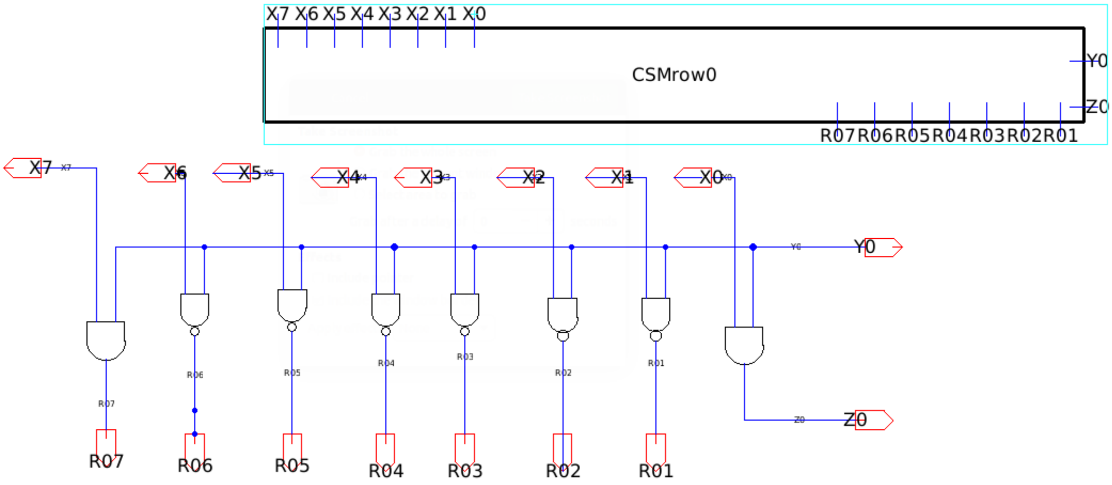


Figure 1: Row 0 of CSM

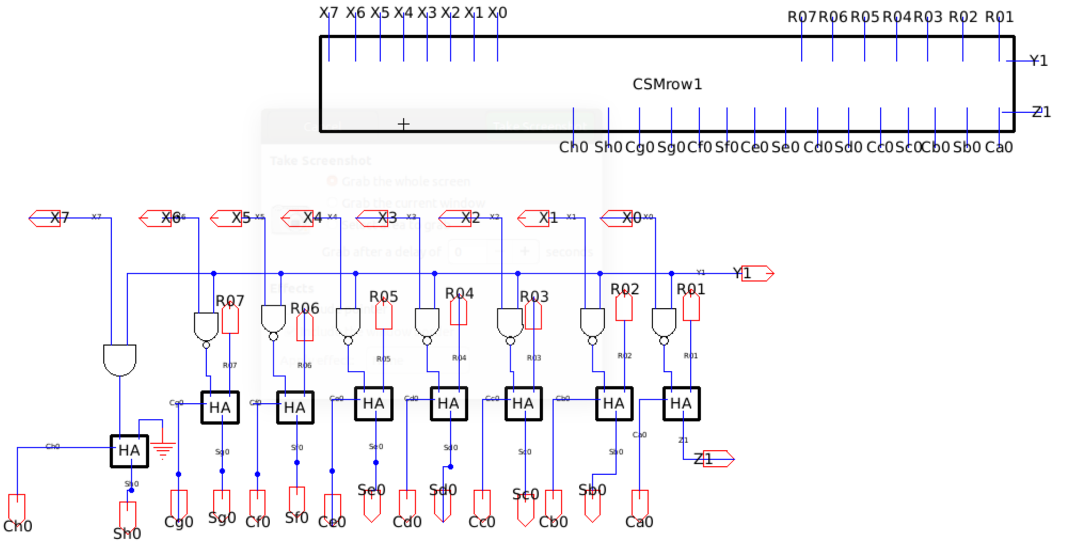


Figure 2: Row 1 of CSM

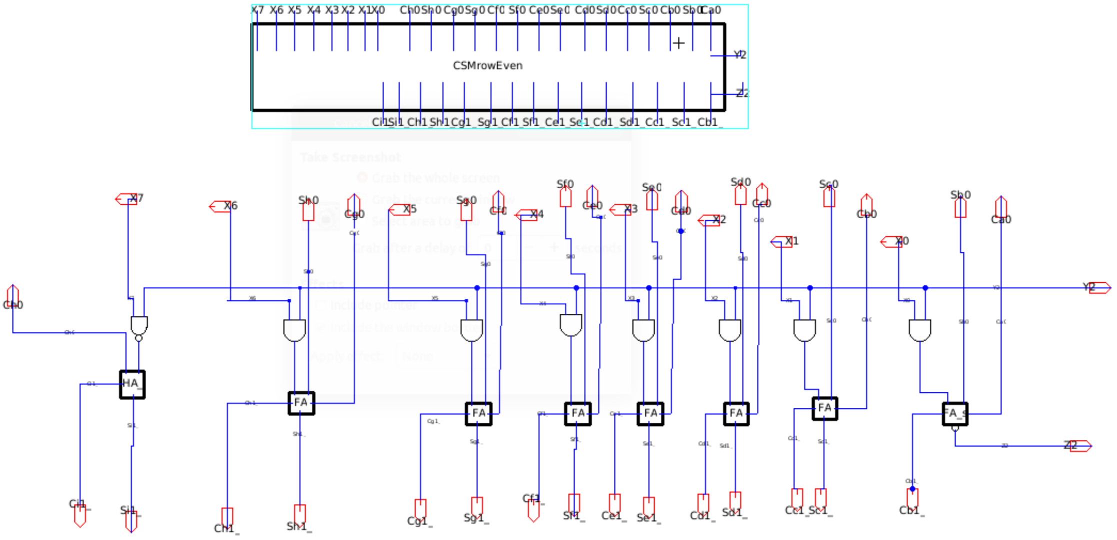

X7 X6 X5 X4 X3 X2 X1 X0 Ch0 Sh0 Cg0 Sg0 Cf0 Sf0 Ce0 Se0 Cd0 Sd0 Cc0 Sc0 Cb0 Sb0 Ca0
CSMrowEven
Y2
Z2
Ci1 Si1 Ch1 Sh1 Cg1 Sg1 Cf1 Sf1 Ce1 Se1 Cd1 Sd1 Cc1 Sc1 Cb1
HA
FA
FA_s
Y2
Z2

Figure 3: Rows 2, 4, 6 of CSM

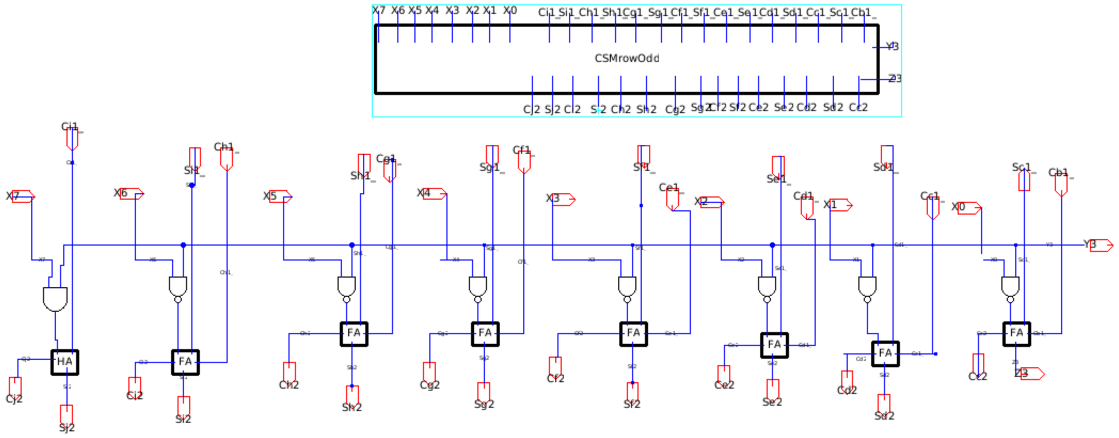

X7 X6 X5 X4 X3 X2 X1 X0 Ci1 Si1 Ch1 Sh1 Cg1 Sg1 Cf1 Sf1 Ce1 Se1 Cd1 Sd1 Cc1 Sc1 Cb1
CSMrowOdd
Y3
Z3
Cj2 Sj2 Ci2 Si2 Ch2 Sh2 Cg2 Sg2 Cf2 Sf2 Ce2 Se2 Cd2 Sd2 Cc2
HA
FA
Y3
Z3

Figure 4: Rows 3 and 5 of CSM

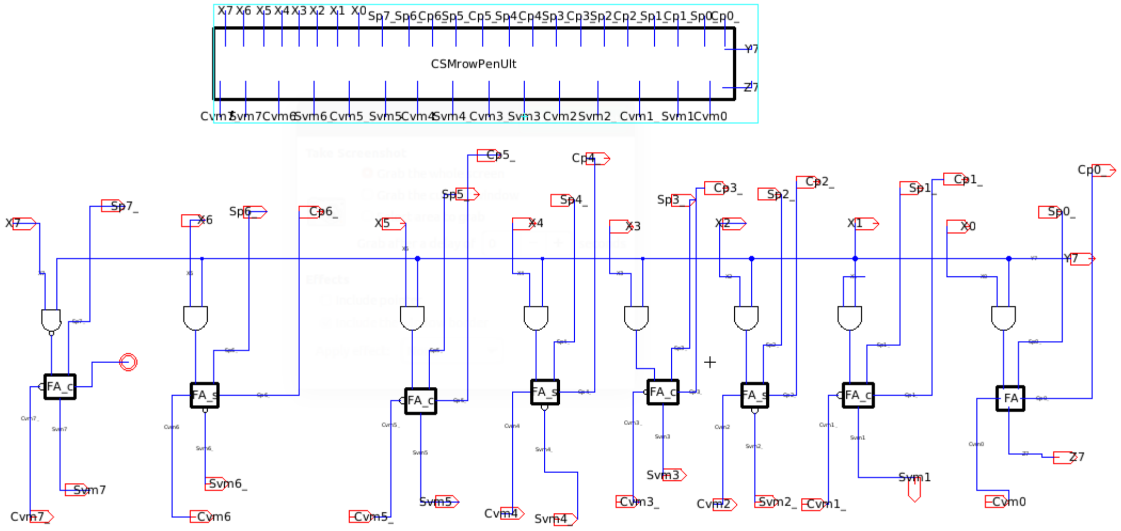

X7 X6 X5 X4 X3 X2 X1 X0 Sp7 Sp6 Cp6 Sp5 Cp5 Sp4 Cp4 Sp3 Cp3 Sp2 Cp2 Sp1 Cp1 Sp0 Cp0
CSMrowPenUlt
Y7
Z7
Cvm7 Svm7 Cvm6 Svm6 Cvm5 Svm5 Cvm4 Svm4 Cvm3 Svm3 Cvm2 Svm2 Cvm1 Svm1 Cvm0
FA_c
FA_s
FA
Y7
Z7

Figure 5: Row 7 (penultimate row) of CSM

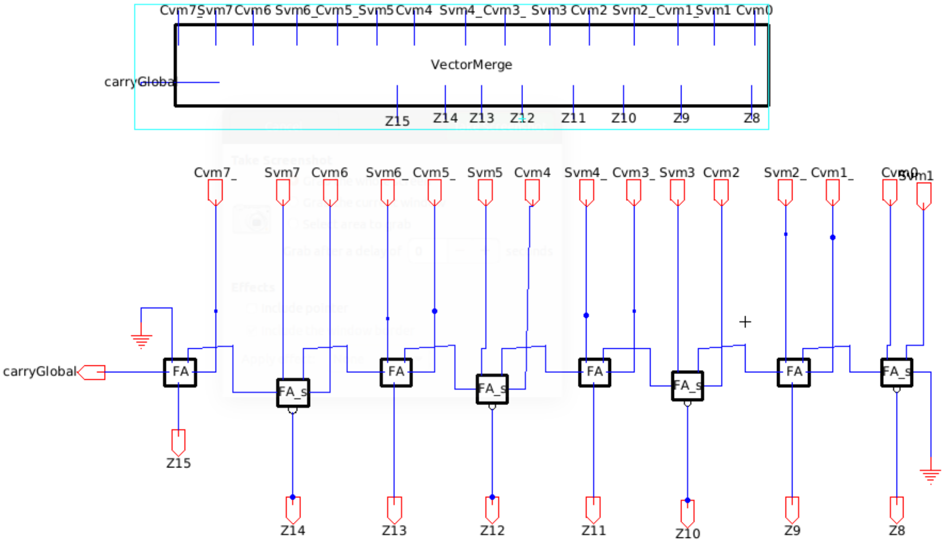

Cvm7 Svm7 Cvm6 Svm6_ Cvm5_ Svm5 Cvm4 Svm4_ Cvm3_ Svm3 Cvm2 Svm2_ Cvm1_ Svm1 Cvm0
carryGlobal
VectorMerge
Z15 Z14 Z13 Z12 Z11 Z10 Z9 Z8
Cvm7_ Svm7 Cvm6 Svm6_ Cvm5_ Svm5 Cvm4 Svm4_ Cvm3_ Svm3 Cvm2 Svm2_ Cvm1_ Cvm0 Svm1
carryGlobal
FA
FA_s
FA
FA_s
FA
FA_s
FA
FA_s
Z15
Z14 Z13 Z12 Z11 Z10 Z9 Z8

Figure 6: Ripple Carry Adder used as Vector Merge in baseline CSM

Use of Inverting Adders: Making use of the mirroring property of the components of a Full Adder circuit, every alternate array of Full Adders in the multiplier takes in inverted inputs- feeding non-inverting outputs to the next array of adders which further give out inverting outputs. This is illustrated by the schematic diagrams above- for example, the even rows take in non inverting inputs and give out inverting outputs (vice versa for odd rows). *Notation: an '_' is added to a node's name to indicate inversion.*

3 Baseline Carry Save Multiplier

3.1 Schematic

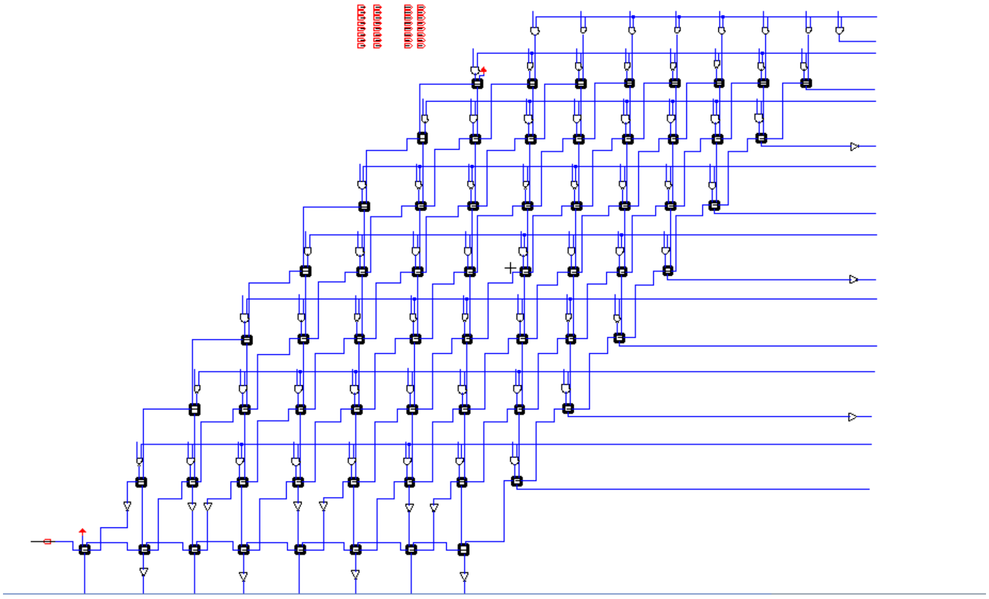


Figure 7: Schematic of CSM with Ripple Carry Adder

3.2 Layout

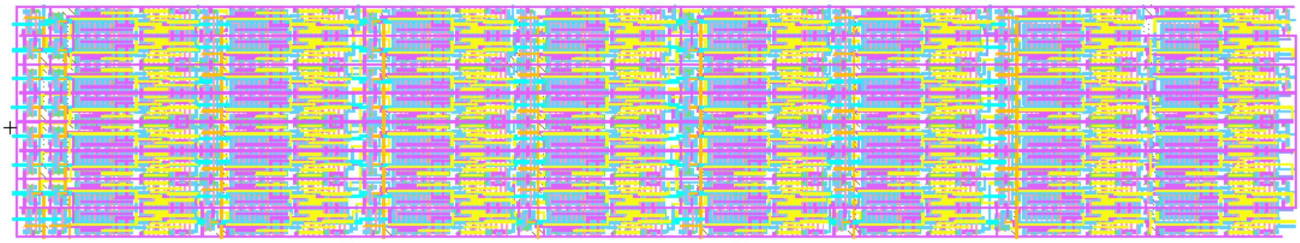
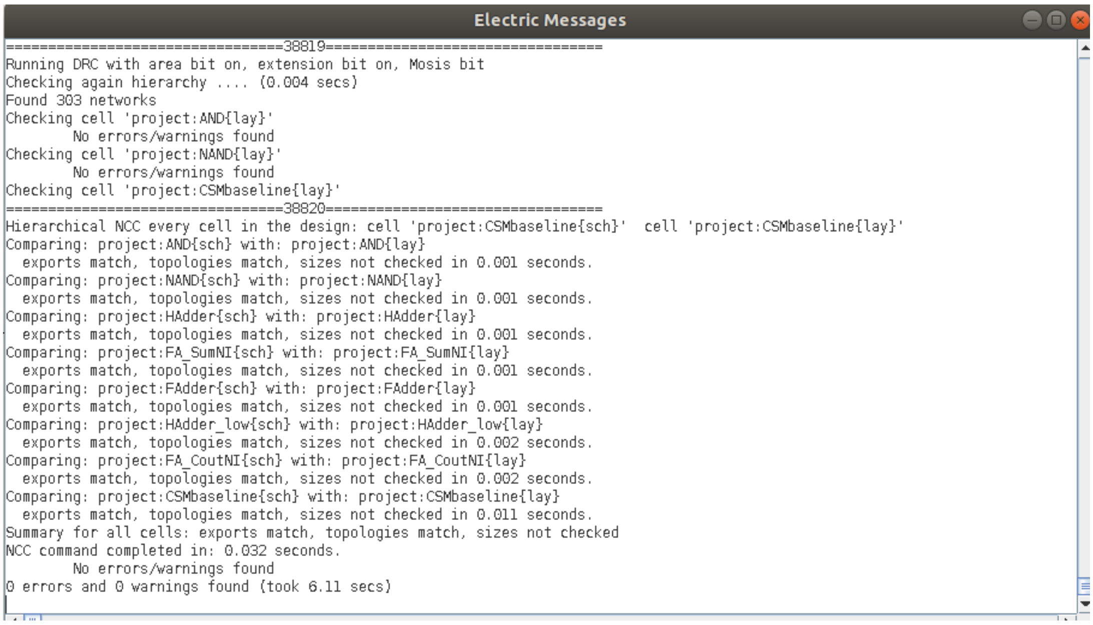


Figure 8: Layout of CSM with Ripple Carry Adder



```
=================================38819================================
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.004 secs)
Found 303 networks
Checking cell 'project:AND{lay}'
        No errors/warnings found
Checking cell 'project:NAND{lay}'
        No errors/warnings found
Checking cell 'project:CSMbaseline{lay}'
=================================38820================================
Hierarchical NCC every cell in the design: cell 'project:CSMbaseline{sch}'  cell 'project:CSMbaseline{lay}'
Comparing: project:AND{sch} with: project:AND{lay}
  exports match, topologies match, sizes not checked in 0.001 seconds.
Comparing: project:NAND{sch} with: project:NAND{lay}
  exports match, topologies match, sizes not checked in 0.001 seconds.
Comparing: project:HAdder{sch} with: project:HAdder{lay}
  exports match, topologies match, sizes not checked in 0.001 seconds.
Comparing: project:FA_SumNI{sch} with: project:FA_SumNI{lay}
  exports match, topologies match, sizes not checked in 0.001 seconds.
Comparing: project:FAdder{sch} with: project:FAdder{lay}
  exports match, topologies match, sizes not checked in 0.001 seconds.
Comparing: project:HAdder_low{sch} with: project:HAdder_low{lay}
  exports match, topologies match, sizes not checked in 0.002 seconds.
Comparing: project:FA_CoutNI{sch} with: project:FA_CoutNI{lay}
  exports match, topologies match, sizes not checked in 0.002 seconds.
Comparing: project:CSMbaseline{sch} with: project:CSMbaseline{lay}
  exports match, topologies match, sizes not checked in 0.011 seconds.
Summary for all cells: exports match, topologies match, sizes not checked
NCC command completed in: 0.032 seconds.
        No errors/warnings found
0 errors and 0 warnings found (took 6.11 secs)
```

Figure 9: Layout vs Schematic and Design Rule Check for complete Layout

3.3 Cells Used

3.3.1 NAND

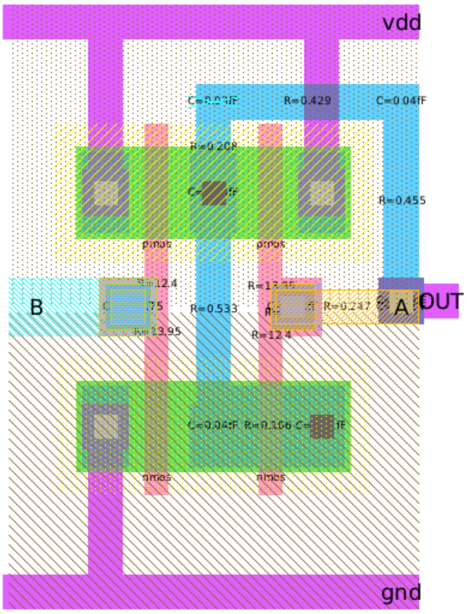
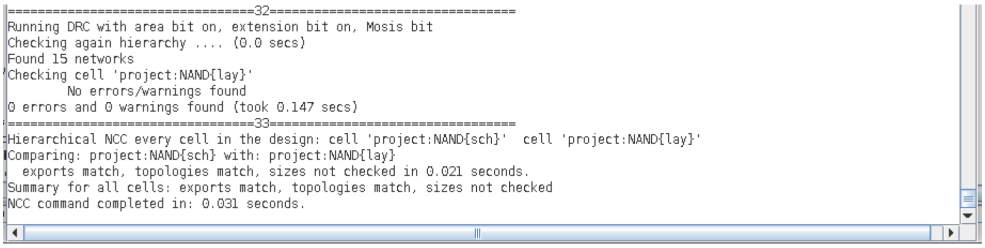


Figure 10: Layout of NAND 1× cell



```
================================32================================
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.0 secs)
Found 15 networks
Checking cell 'project:NAND{lay}'
        No errors/warnings found
0 errors and 0 warnings found (took 0.147 secs)
================================33================================
Hierarchical NCC every cell in the design: cell 'project:NAND{sch}'  cell 'project:NAND{lay}'
Comparing: project:NAND{sch} with: project:NAND{lay}
  exports match, topologies match, sizes not checked in 0.021 seconds.
Summary for all cells: exports match, topologies match, sizes not checked
NCC command completed in: 0.031 seconds.
```

Figure 11: Layout vs Schematic and Design Rule Check for NAND

3.3.2 INV

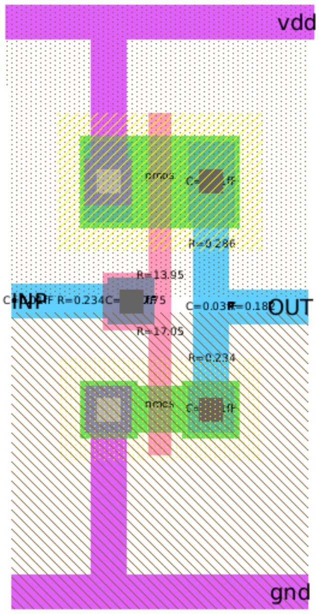
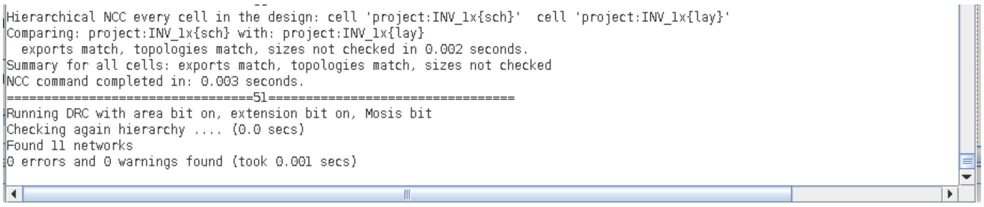


Figure 12: Layout of INV 1× cell



```
Hierarchical NCC every cell in the design: cell 'project:INV_1x{sch}'  cell 'project:INV_1x{lay}'
Comparing: project:INV_1x{sch} with: project:INV_1x{lay}
  exports match, topologies match, sizes not checked in 0.002 seconds.
Summary for all cells: exports match, topologies match, sizes not checked
NCC command completed in: 0.003 seconds.
=================================51=================================
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.0 secs)
Found 11 networks
0 errors and 0 warnings found (took 0.001 secs)
```

Figure 13: Layout vs Schematic and Design Rule Check for INV

3.3.3 Full Adder

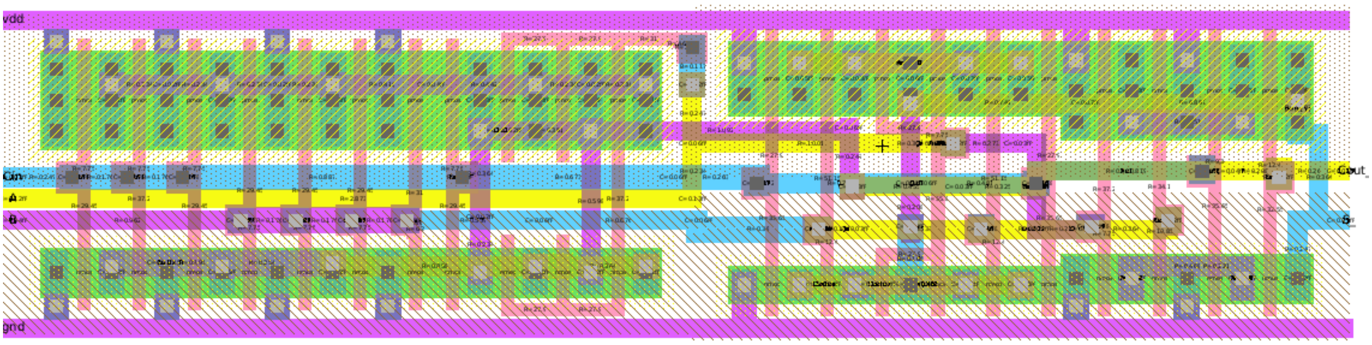
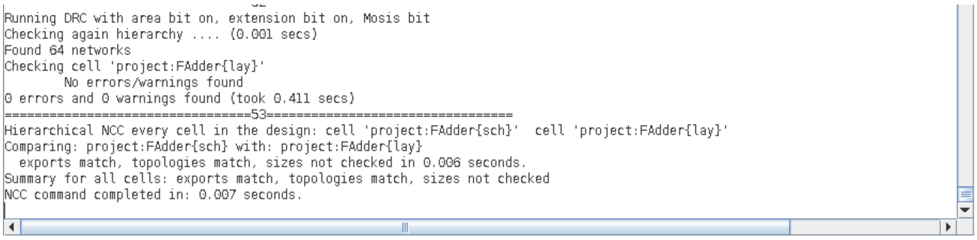


Figure 14: Layout of Full Adder 3×-1× cell



```
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.001 secs)
Found 64 networks
Checking cell 'project:FAdder{lay}'
        No errors/warnings found
0 errors and 0 warnings found (took 0.411 secs)
=================================53=================================
Hierarchical NCC every cell in the design: cell 'project:FAdder{sch}'  cell 'project:FAdder{lay}'
Comparing: project:FAdder{sch} with: project:FAdder{lay}
  exports match, topologies match, sizes not checked in 0.006 seconds.
Summary for all cells: exports match, topologies match, sizes not checked
NCC command completed in: 0.007 seconds.
```

Figure 15: Layout vs Schematic and Design Rule Check for FA

3.4 Layout Characteristics

The layout occupies an area of $2248\lambda \times 409\lambda$, where $\lambda = 11$nm.

Parameter	Value
Length	2248 λ
Width	409 λ
Area	111.251 μm^2
Aspect Ratio	1:5.50
Propagation Delay (schematic)	453 ps
Prop Delay (RC netlist)	575.58 ps
Layout Efficiency	1.271

Layout Characteristics for baseline Carry Save Multiplier

3.5 Functional Simulation

3.5.1 63×-28

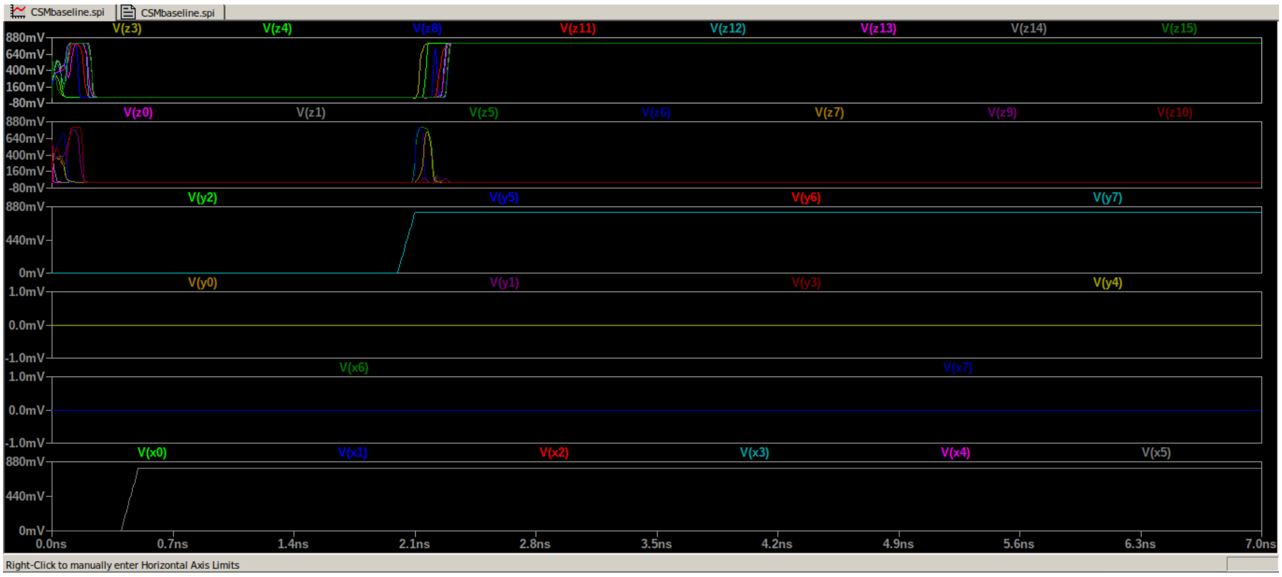


Figure 16: X = 0011 1111, Y = 1110 0100, Z = 1111 1001 0001 1100

3.5.2 127 × −128

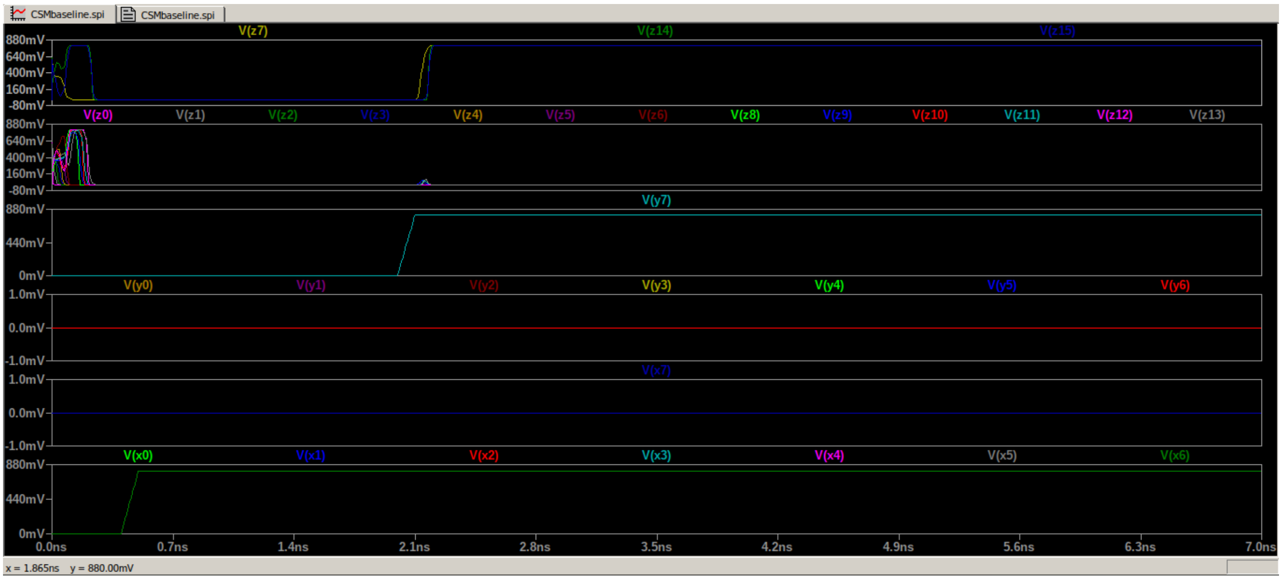


Figure 17: X = 0111 1111, Y = 1000 0000, Z = 1100 0000 1000 0000

3.5.3 9 × 15

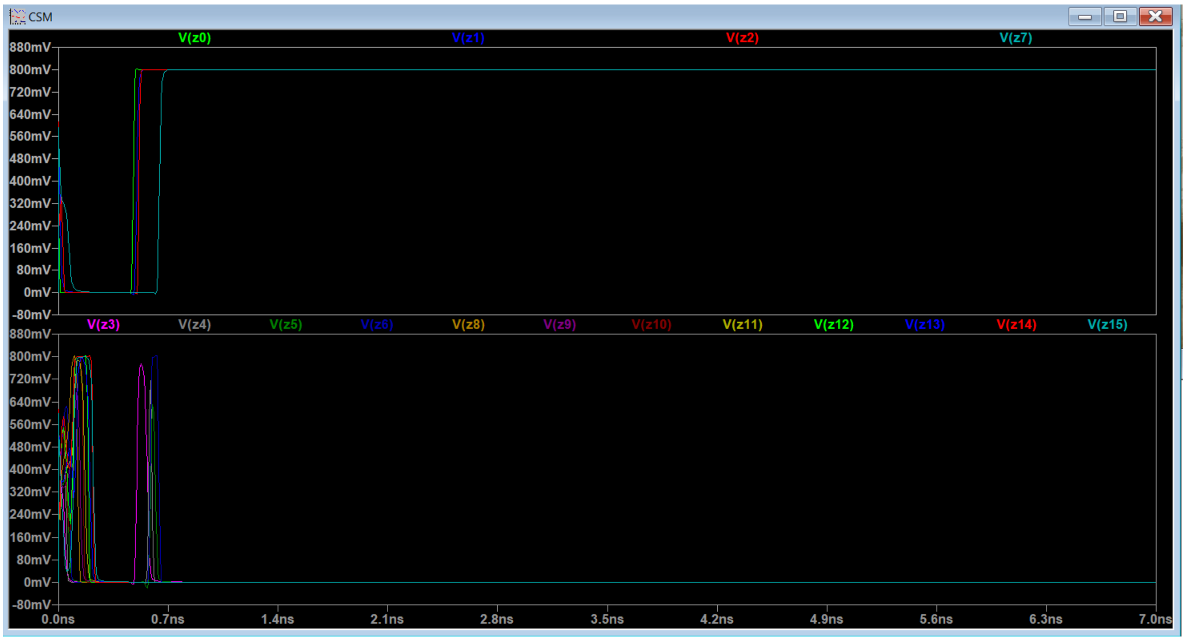


Figure 18: X = 0000 1001, Y = 0000 1111, Z = 0000 0000 1000 0111

3.5.4 -10×12

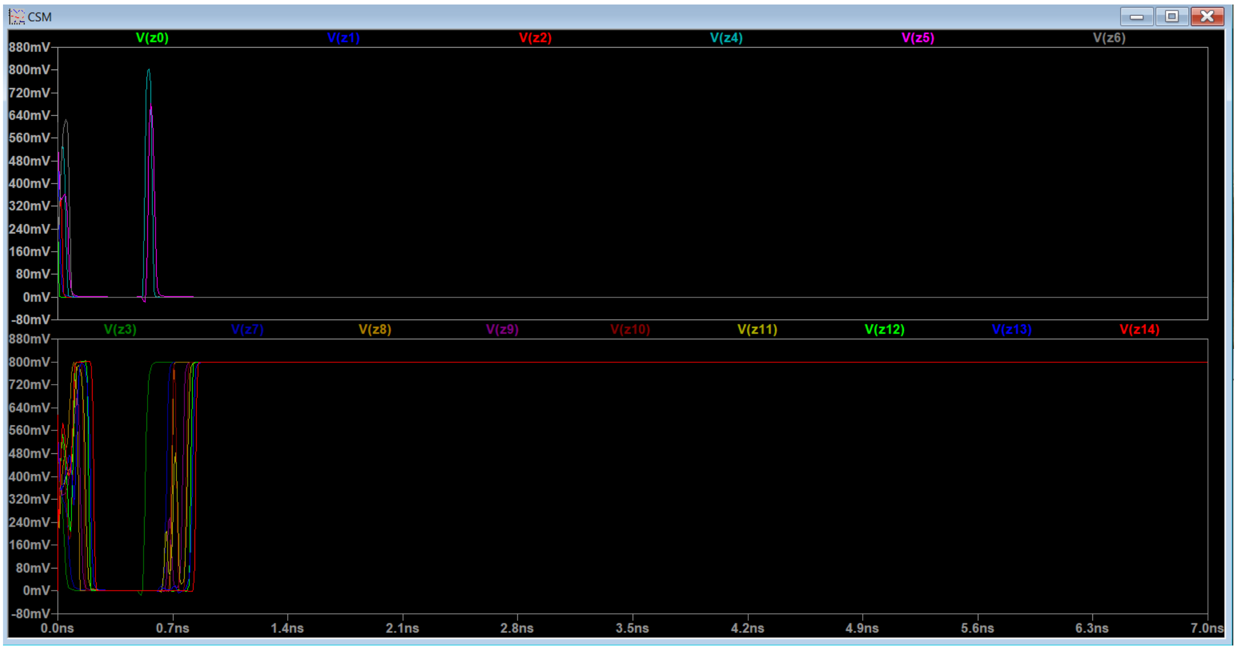


Figure 19: X = 1111 0110, Y = 0000 1100, Z = 1111 1111 1000 1000

3.5.5 -87×-101

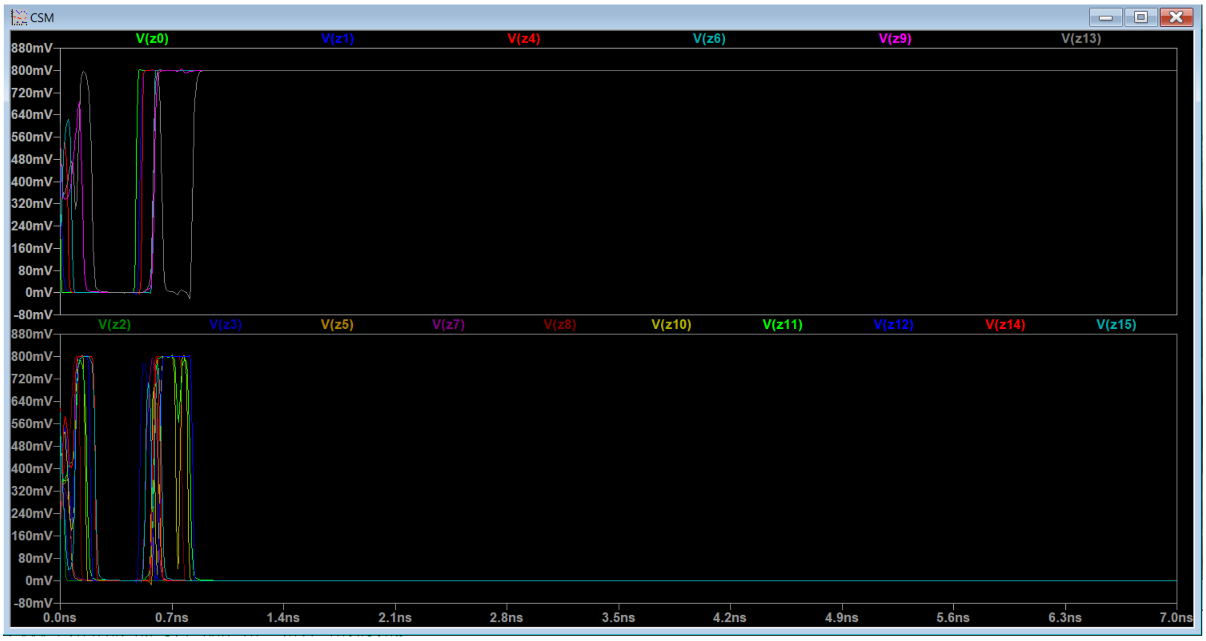


Figure 20: X = 1010 1001, Y = 1001 1011, Z = 0010 0010 0101 0011

4 Pipelined CSM

4.1 Flip-Flop Circuit

A C^2MOS Flip-Flop was designed with sizes chosen appropriately to minimize the setup and clock-to-q delays.

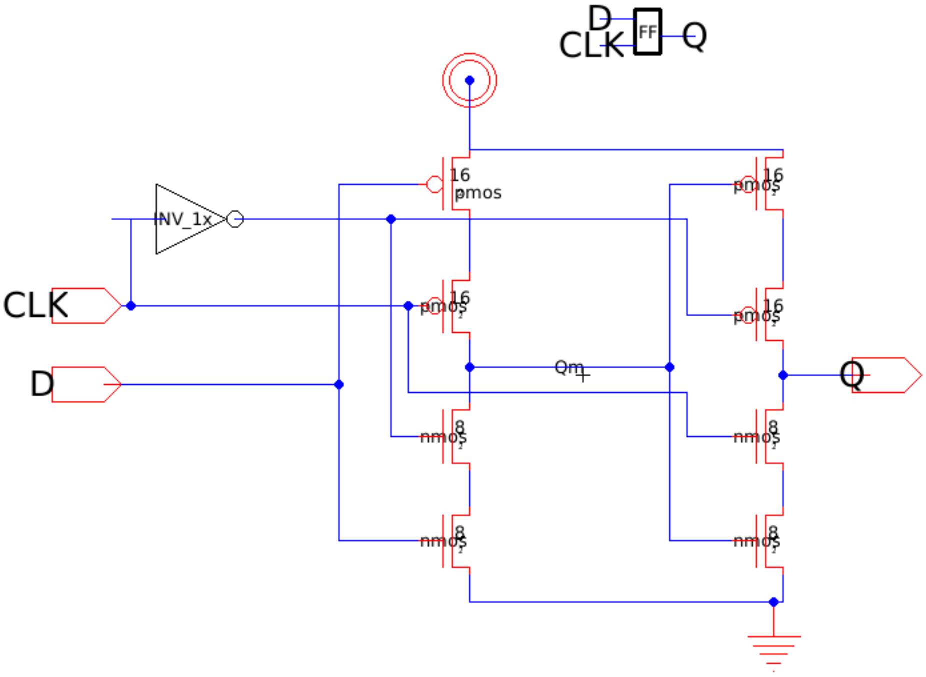


Figure 21: C2MOS Flop

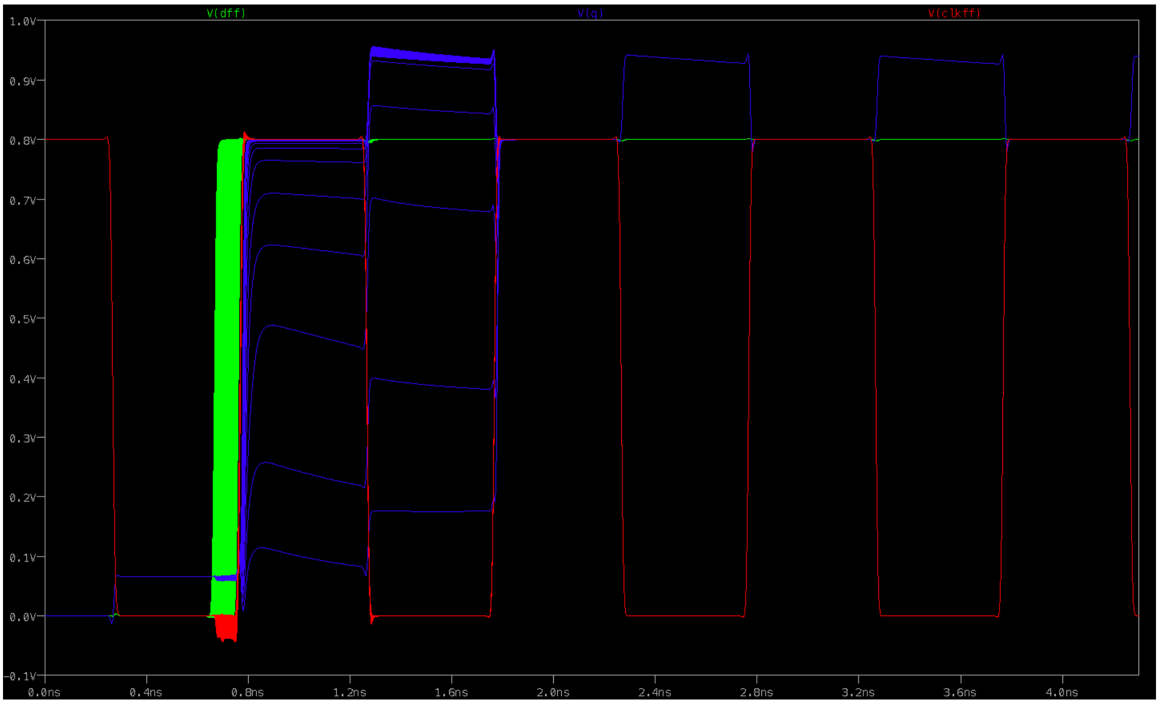


Figure 22: Functional Simulation of Flip Flop

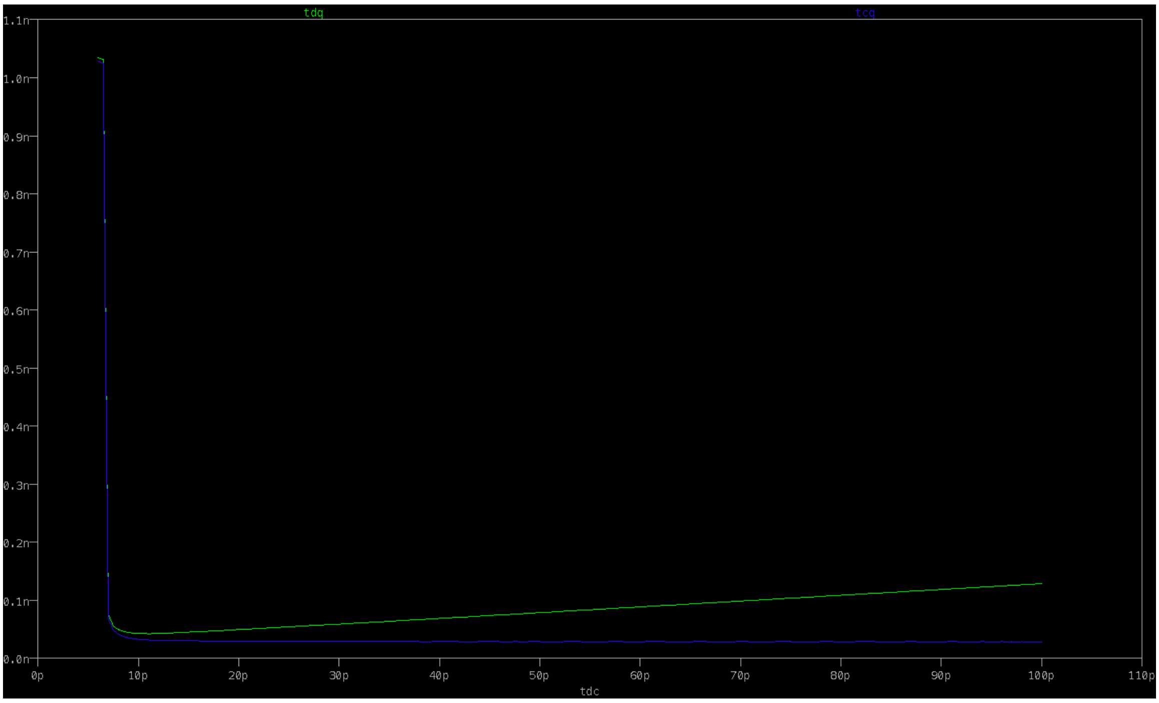


Figure 23: T_{dq} v/s T_{dc} of Flip-Flop

From the graph and results tabulated, the minimum T_{dq} is 43ps which occurs at T_{dc} of 9.3ps. Therefore, propagation delay when using this Flip Flop is 43 ps and the setup time is 9.3 ps.

4.2 Pipelining

An array of Flip-Flops were placed between rows 5 and 6 of the schematic to store the state of the computation. This enable the circuit to operate at a time period that is equal to `min{Input->Pipeline Array delay, Pipeline Array -> Output Delay}`. To maximize the operating frequency, the array is placed midway- this was done by identifying the node in the critical model that was equally far in terms of delay from the input and the output.

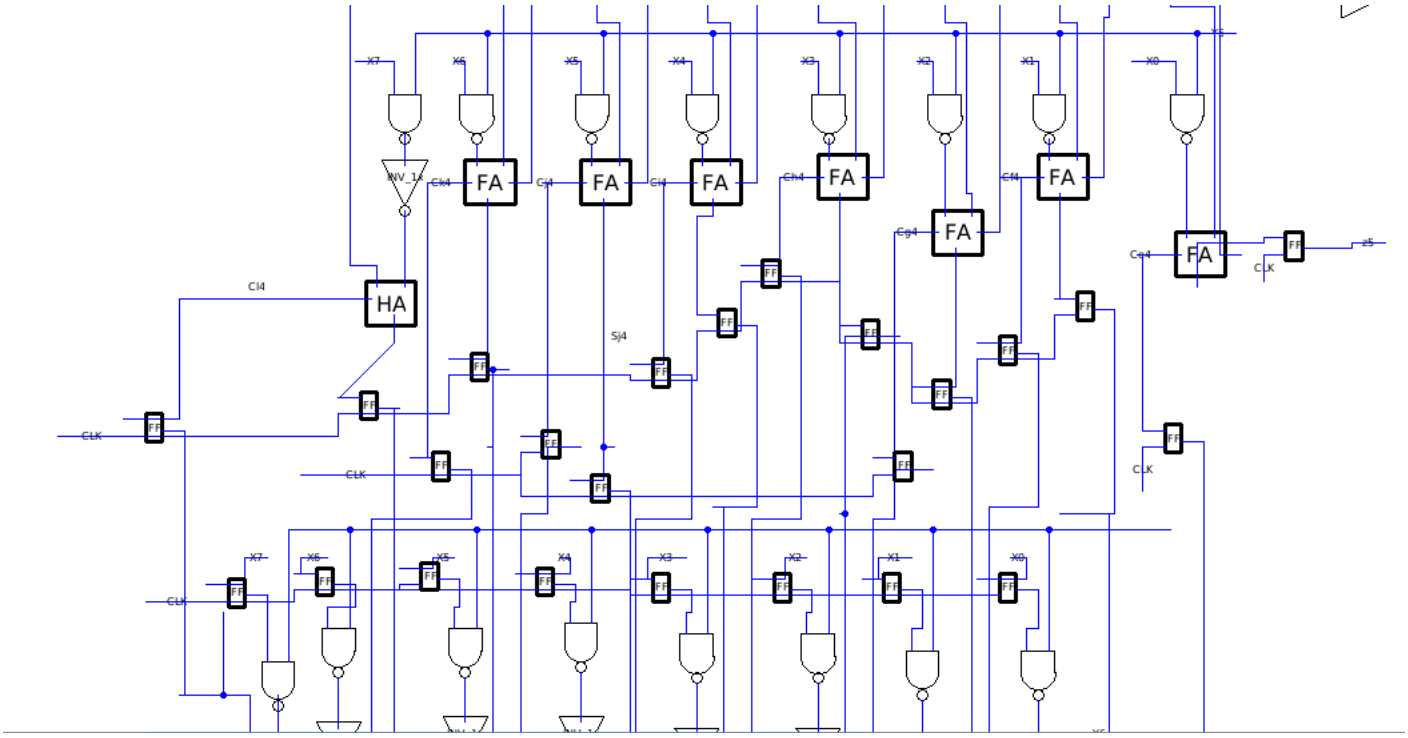


Figure 24: Location of Pipeline FF array in the schematic

4.2.1 Minimum Operating Period

The following table lists the operating delay for pipelined and non-pipelined versions of the circuit. Hence, pipelining enables a speedup of about 1.79×.

CARRY SAVE MULTIPLIER	NON-PIPELINED	PIPELINED
Schematic Netlist	485 ps (2.062 GHz)	270 ps (3.704GHz)
Layout Netlist	610 ps (1.639 GHz)	382 ps (2.618 GHz)

4.3 Functional Simulation

4.3.1 -1×-1

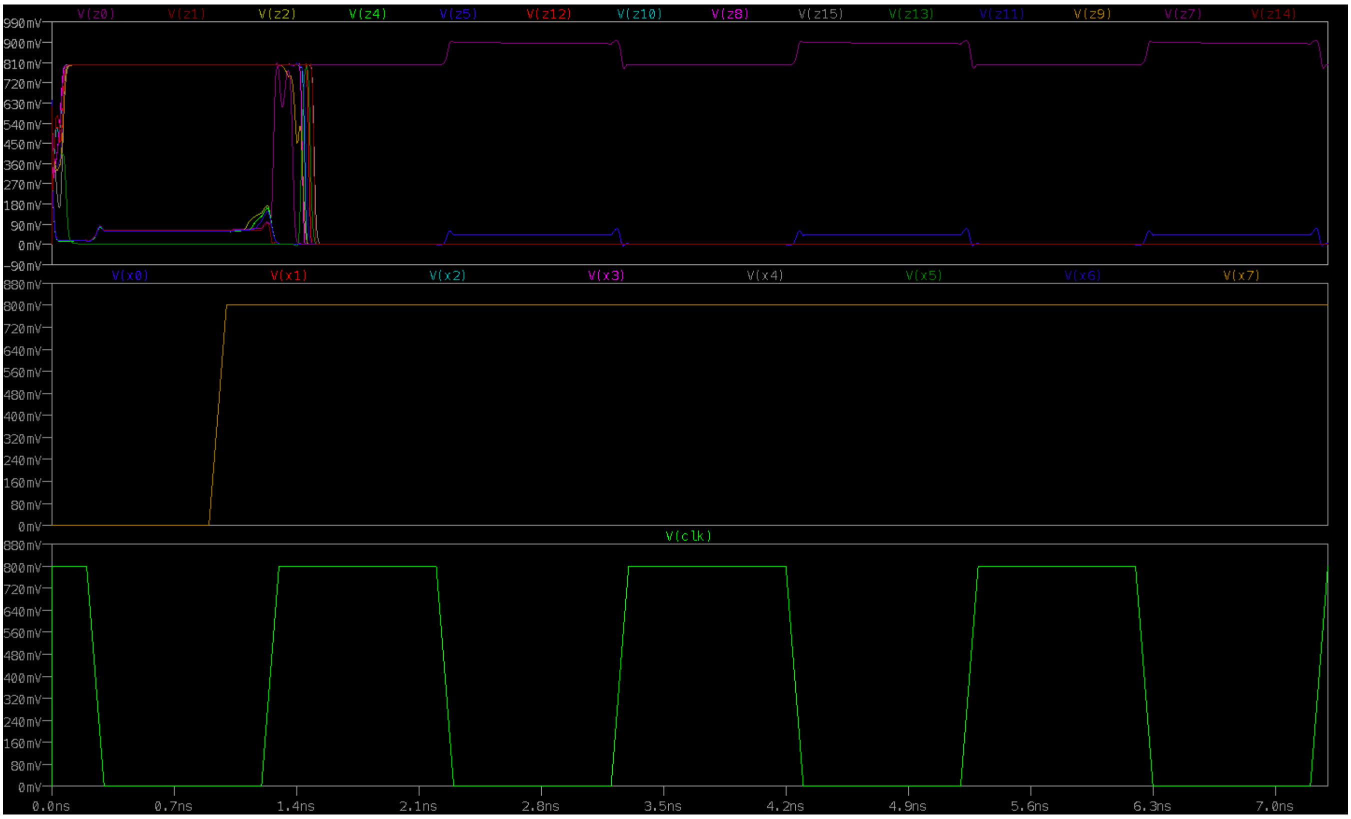


Figure 25: X = 1111 1111, Y = 1111 1111, Z = 0000 0000 0000 0001

4.3.2 -11×-11 with input setup and output capture flops

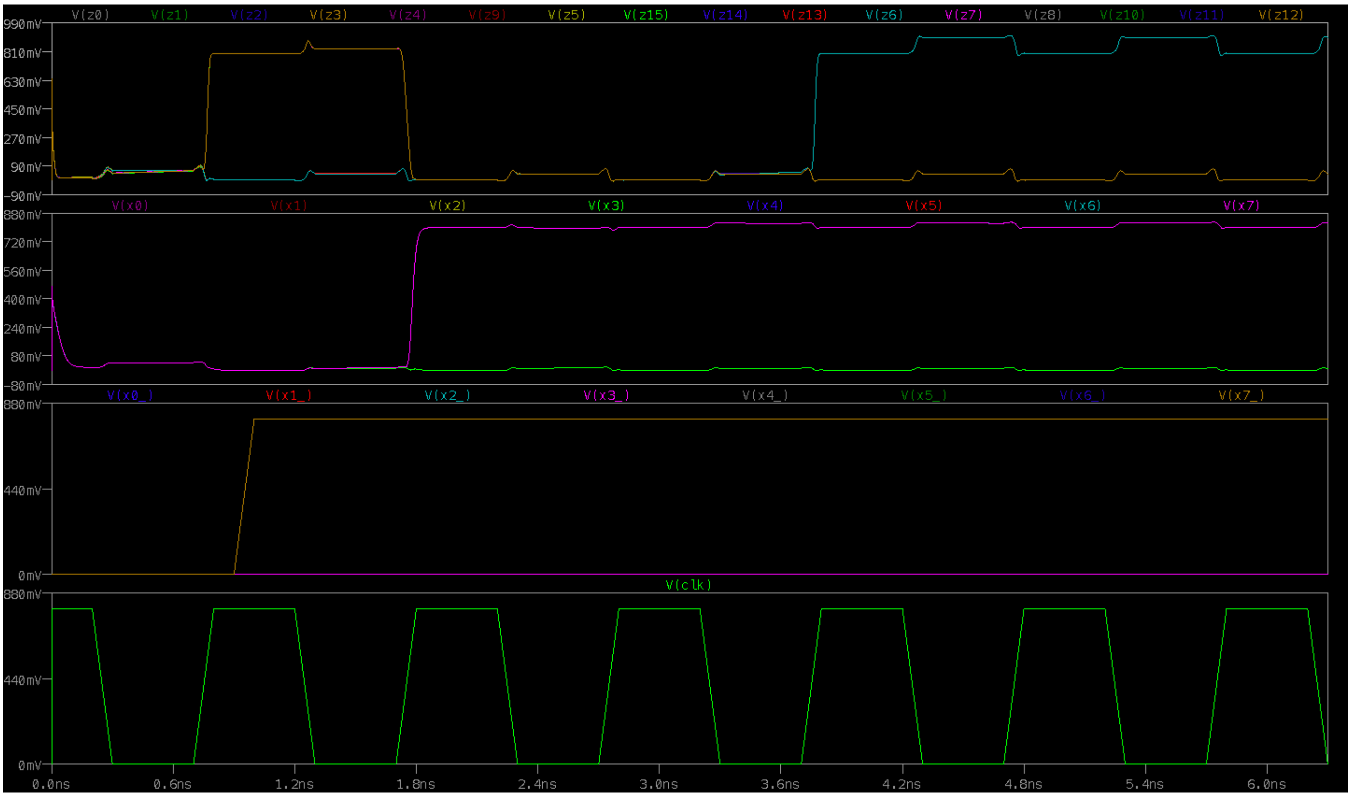


Figure 26: X = 1111 0101, Y = 1110 0101, Z = 0000 0000 0111 1001

5 Alternate Vector Merge Adders

5.1 Carry Skip Adder

Two versions of Carry Skip Adder were tried- N = 4 and N = 2. The N = 4 version gave a performance benefit for the input sequence that triggers the critical path (-1×-1), while N = 2 did not. This is because the net delay due to the MUX and XOR gates is significant in the N = 2 version.

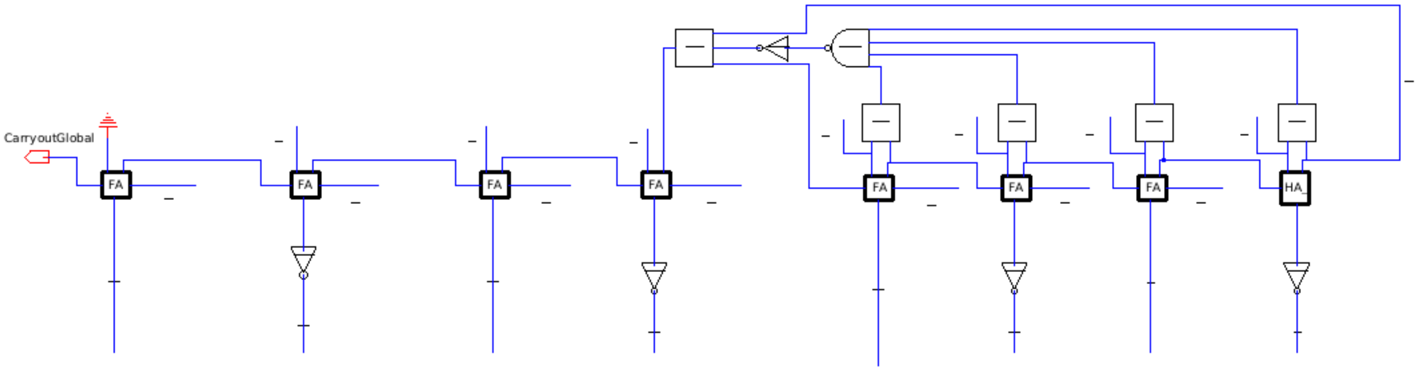


Figure 27: Carry Skip Adder Schematic used in place of Ripple Adder in the Vector Merge Stage

5.1.1 Delay Comparison

Comparison of propagation delays for the Carry-Save Multiplier by using Carry Skip adder.

	CRITICAL DELAY MEAS.	
NETLIST USED	**SCHEME**	**CRITICAL DELAY**
SCHEMATIC	CARRY SAVE MULTIPLIER	417 ps
SCHEMATIC	CARRY SKIP MULTIPLIER (N=4)	420 ps
SCHEMATIC	CARRY SKIP MULTIPLIER (N=2)	442 ps

5.2 Carry Lookahead Adder

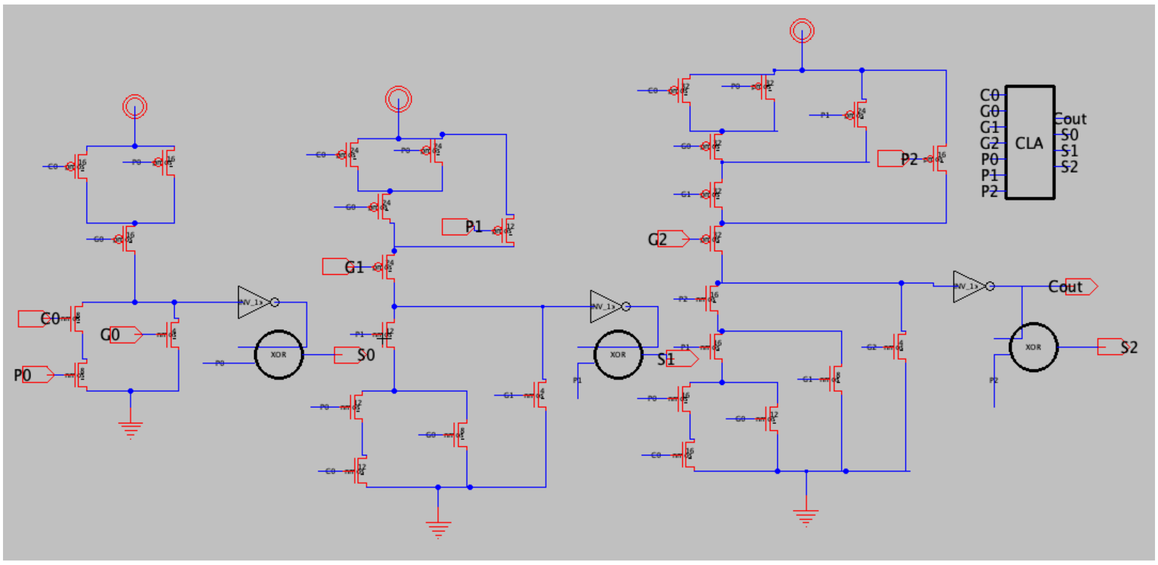


Figure 28: Carry Lookahead Adder Schematic

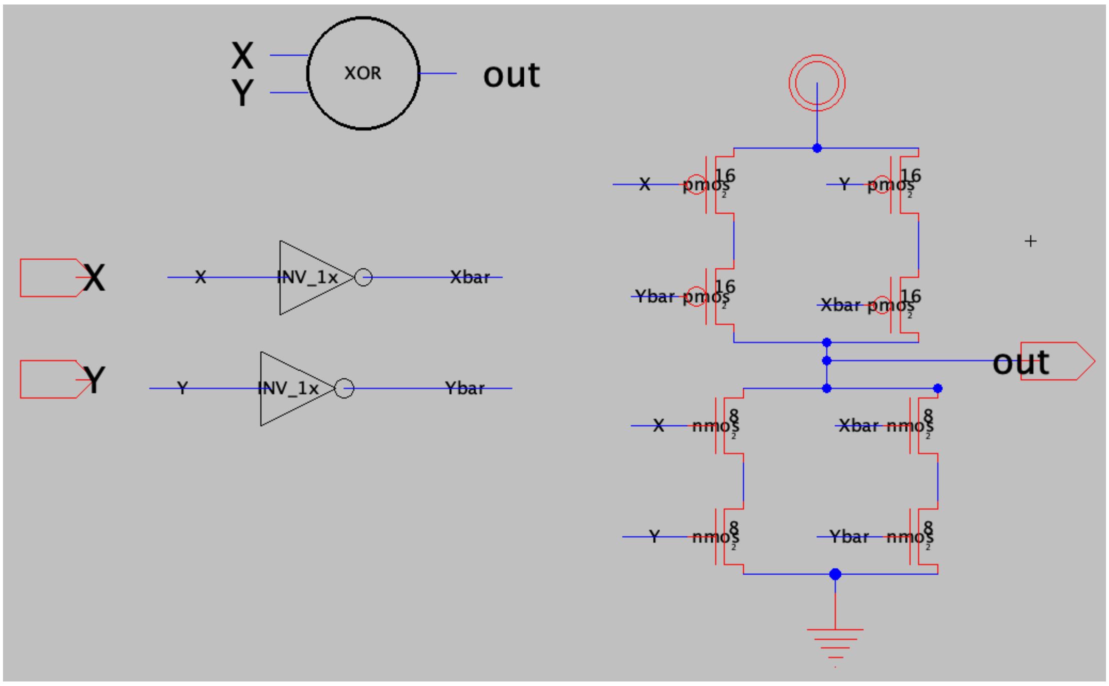


Figure 29: XOR gate used in the Carry Lookahead Adder

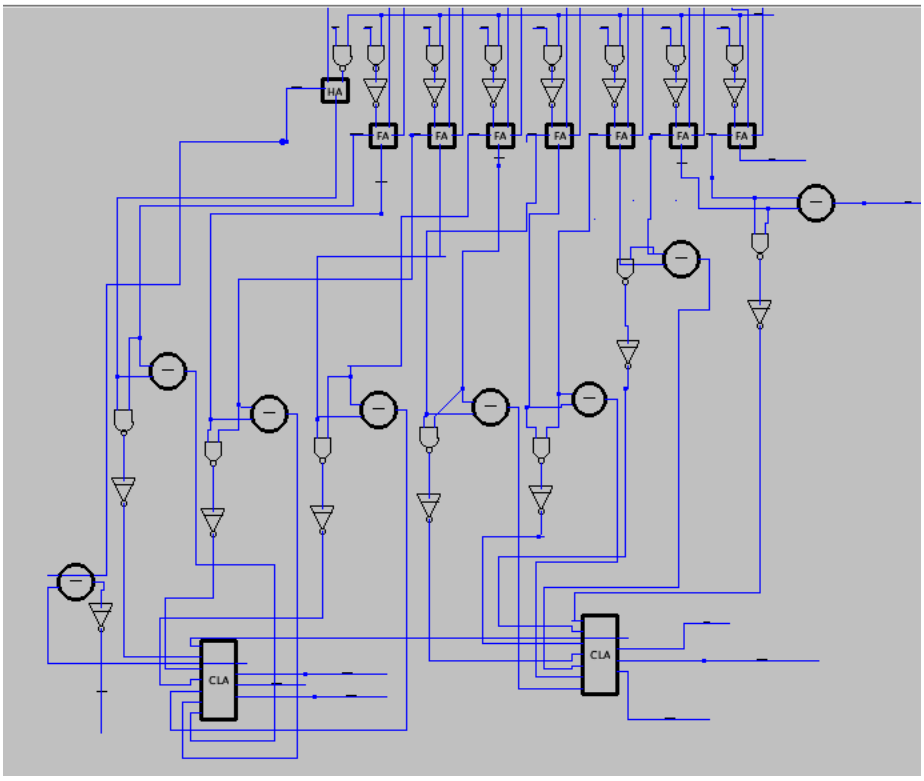


Figure 30: Carry Lookahead Adder used in Vector Merge Stage

5.2.1 -1×1 with 3% decrease

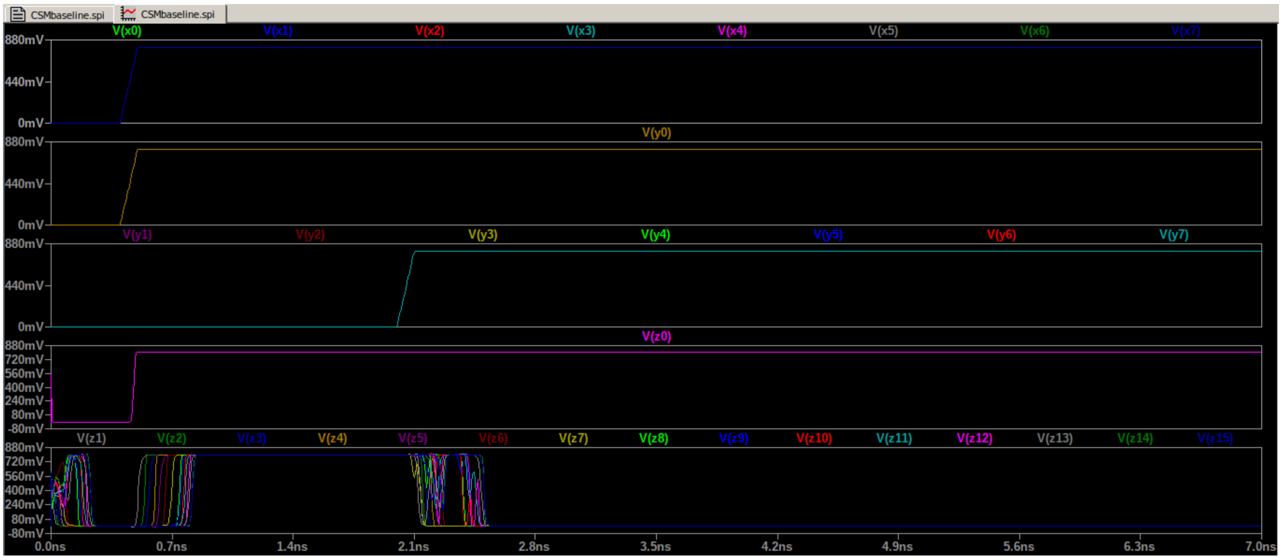


Figure 31: X = 1111 1111, Y = 1111 1111, Z = 0000 0000 0000 0001

Using carry look ahead adder decreases the delay. The decrease in delay however depends on the input and varies from about 3-30%. Other inputs such as -11×11 have a 30% decrease in delay(reduced to 280ps from 400ps).

★ ★ ★