Bruno E. Gracia Villalobos

EE 4513 – Introduction to VLSI Design

Design and Implementation of a Static CMOS 8-bit Ripple Carry Adder

October 7, 2019

INTRODUCTION

This report entails of the design of an 8-bit Ripple Carry Adder (RCA) using 0.6-micron, static CMOS technology. Using λ values for convention, a λ of 0.3 micron is therefore chosen given the standard MOSFET dimensions of 2 λ x 4 λ for width and length respectively. The standard width of each cell is 80 λ for all fundamental gates and 1-bit adders. The length of each NOR, NAND, and INV cells are 50 λ, and the rest of the cells have variable lengths.

The adder cell is developed from the ground up, and the report describes the design process of realizing the fundamental building blocks of the adder: half-adders, full-adders, and 4-bit full-adders—with increasing abstraction. In addition to these, the gates underlying their designs are also developed from scratch and simulated for verified operation. Essentially, a cell-based design approach is used for the author to learn of the increased importance in the industry for saving both p-substrate space, decreased used of layering, and decreased use of metal surface area when available to reduce costs. To begin, the 2-input XOR and 2-input AND gates are both examined as the building blocks of the half-adder.

2-INPUT XOR GATE

Using previously built NAND gates, the 2-input XOR gate is realized using four cells as shown in Figure 1. Metal 1 is used for the power and signal pins, and Metal 2 is used for the interconnects between cells. The yellow outline in Figure 1 represents the use of efficient Metal 1 routing in place of Metal 2 when possible. In retrospect, the pins would have better placement if routed to the outsides of the cell to follow VLSI conventions.

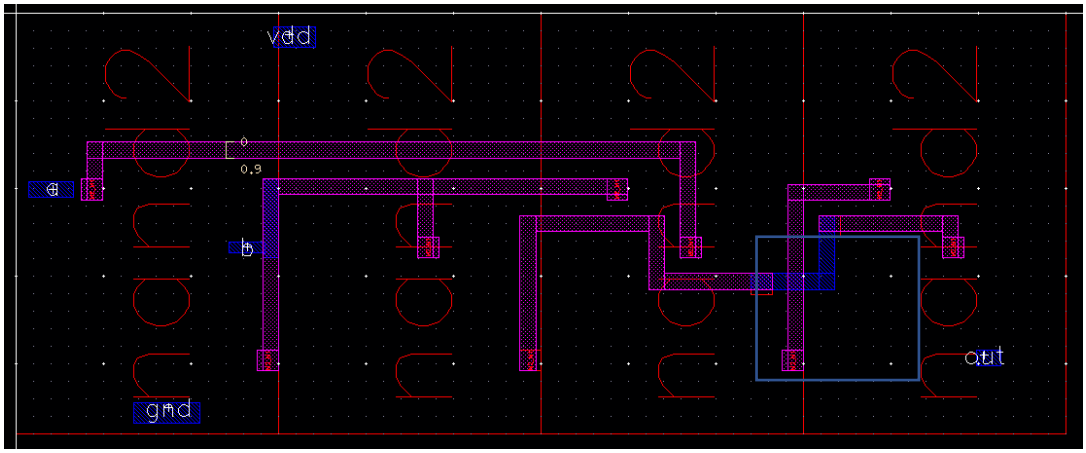


Figure 1: 2-input XOR Gate

The XOR gate uses 16 MOSFETs in total, and these are split evenly between the Pull-Up Network (PUN) and Pull-Down Network (PDN). Figure 2 demonstrates the CMOS layout of the XOR gate.

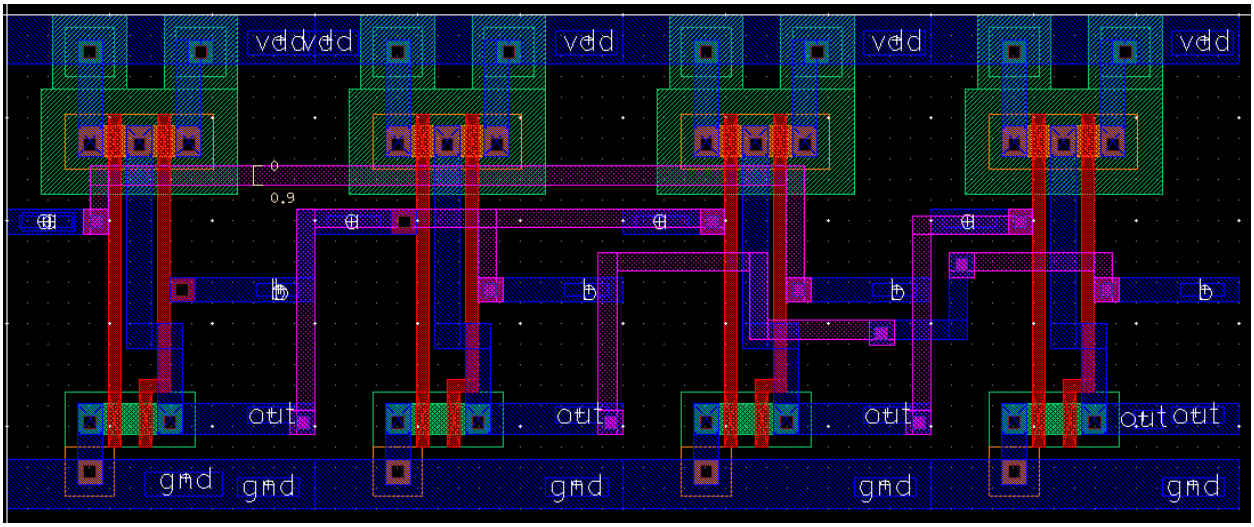


Figure 2: 2-input XOR Gate CMOS Layout

The last step for designing the XOR gate is to create a simulation schematic to test the cell for desired operation. Given the combinational outputs of a 2-input XOR gate amount to four unique values, one square wave for each input is used to verify all possible combinations. To represent the Most-Significant-Bit (MSB), a square wave with a pulse width of 40ns and a period of 80ns is used. To represent the Least-Significant-Bit (LSB), a square wave with a pulse width of 20ns and a period of 40ns is used. These choices allow the simulator to effectively test every possible input into the XOR gate. Figure 3 showcases the full schematic used to test design.

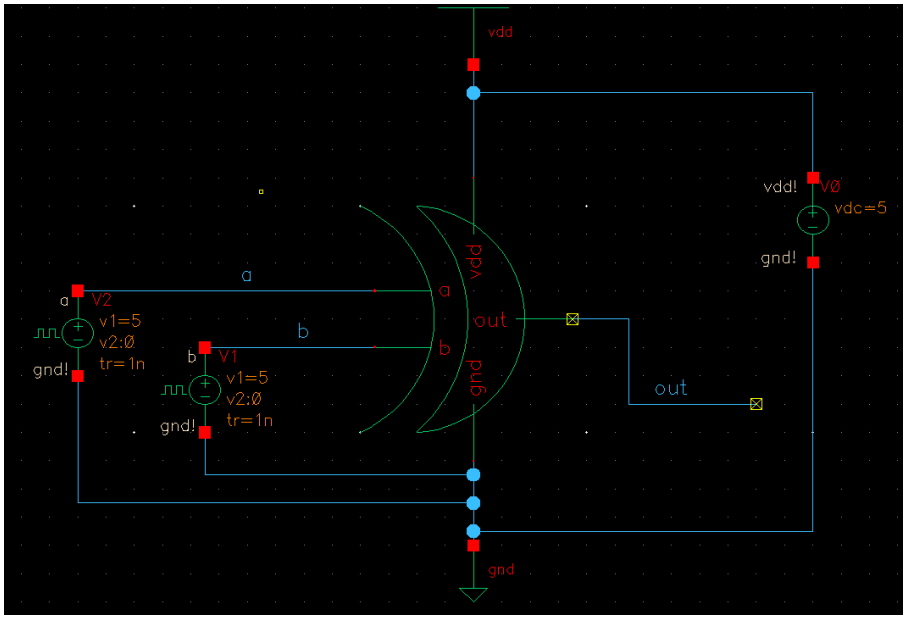
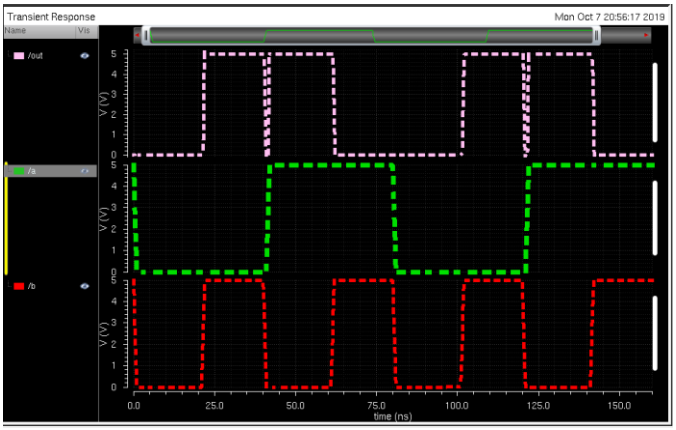


Figure 3: Test schematic for 2-input XOR Gate




```
******  Circuit Statistics  ******
# nodes        =       12 # elements   =       19
# resistors    =        0 # capacitors =        0
# mutual_inds  =        0 # vccs       =        0
# cccs         =        0 # ccvs       =        0
# curr_srcs    =        0 # diodes     =        0
# jfets        =        0 # mosfets    =       16
# T elements   =        0 # W elements =        0
# S elements   =        0 # P elements =        0
# vector_srcs  =        0 # N elements =        0
```

Figure 4: (LEFT) XOR Gate waveforms. (TOP RIGHT) Bill of Materials (BOM) for XOR Gate

The simulation results are presented in Figure 4. As expected, the report verifies the amount of MOSFETs created in the CMOS layout to amount to 16. The output waveform shows the correct values for a XOR gate, which must be equal to 1 unless both inputs are the same value.

HALF-ADDER

With the fundamental building blocks built, 2-input XOR and 2-input AND gates, the half-adder cell can now be designed. The layout consists of a XOR2 gate followed by an AND2 gate, and the dimensions are 300 λ by 80 λ. In this design, up to Metal 2 routing is used for the interconnections. As shown in Figure 5, Metal 1 routing is used whenever possible to tunnel under existing Metal 2 traces. Like the previous designs, the author concedes the pins could have been routed better to the outsides of the cell to follow design conventions. Nevertheless, apart from the power pins, the half-adder consists of two inputs and two outputs, a, b and sum, carry respectively.

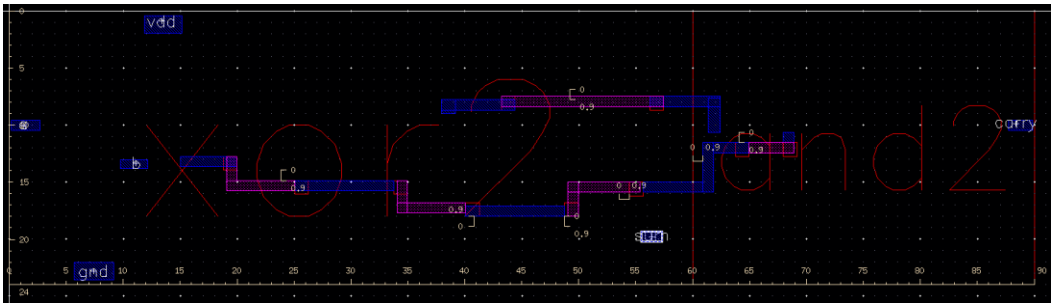


Figure 5: Subdivision of building blocks in the Half-Adder cell

Figure 6 displays the CMOS layout of the Half-Adder cell. The power rails VDD and GND are both 8 λ in width. For this design, 16 MOSFETs from the XOR gate + 6 MOSFETs from the AND gate amount to 22 MOSFETs in total.

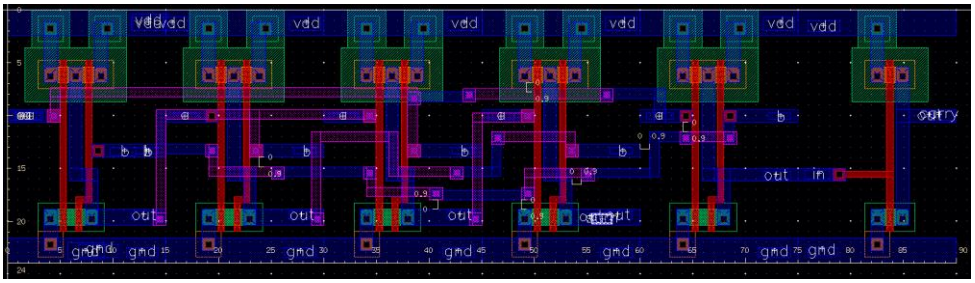
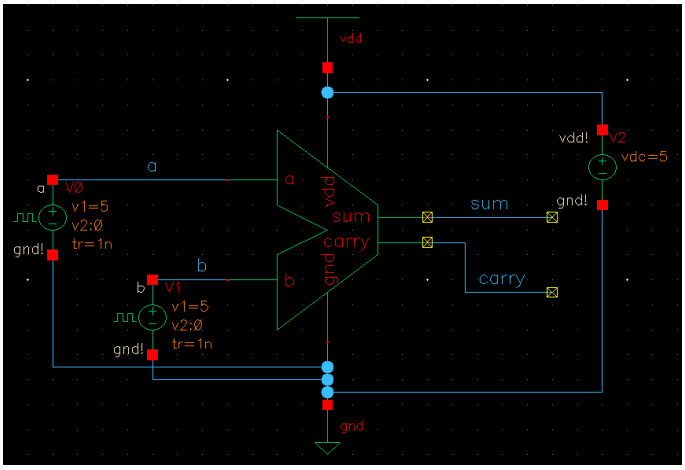


Figure 6: CMOS layout of Half-Adder cell

Given the cell takes in two signals, a total of 2^2 combinations are checked against using two square waves like the testing process of the 2-input XOR gate; their pulse widths and periods are also set accordingly to represent the MSB and LSB of the half-adder. A standard 5V supply and ground nets are added to the power pins of the half-adder as well. The test schematic is crafted to reflect these requirements to verify the cell for desired operation and shown in Figure 7.




```
******  Circuit Statistics  ******
# nodes        =      15 # elements   =      25
# resistors    =       0 # capacitors =       0
# mutual_inds  =       0 # vccs       =       0
# cccs         =       0 # ccvs       =       0
# curr_srcs    =       0 # diodes     =       0
# jfets        =       0 # mosfets    =      22
# T elements   =       0 # W elements =       0
# S elements   =       0 # P elements =       0
# vector_srcs  =       0 # N elements =       0
```

Figure 7: (LEFT) Test schematic for half-adder. (TOP RIGHT) BOM of half-adder

The simulation report in Figure 7 attests to the CMOS layout of the half-adder: only 22 MOSFETs are used, which are divided evenly between the PUN and PDN of the cell as a whole. The half-adder is sucessfully verified in Figure 8 below, and now the full-adder is presented next.

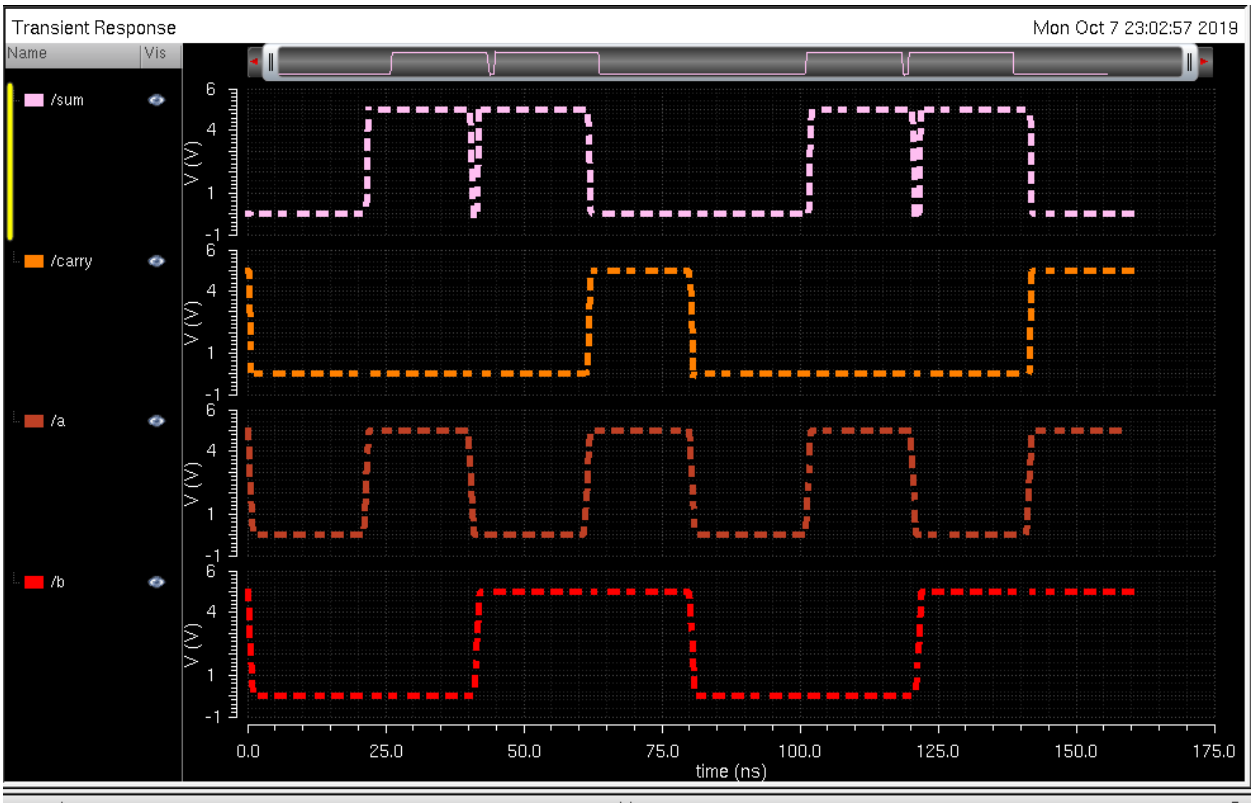
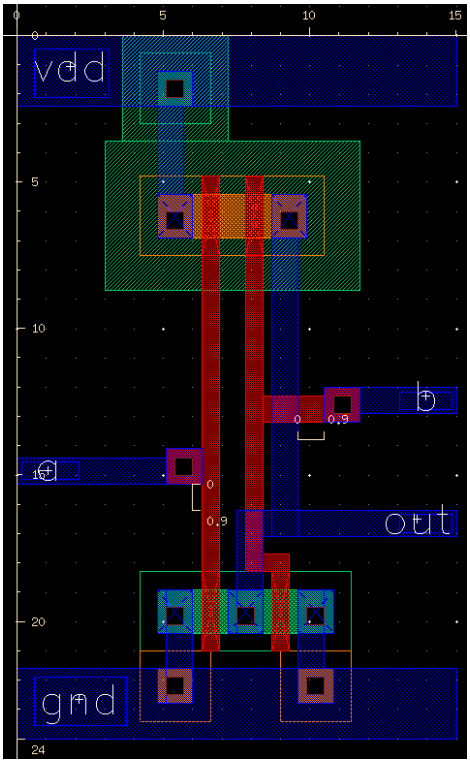


Figure 8: Half-adder simulation results

2-INPUT NOR



Before designing the half-adder, which constitutes of a 2-input XOR gate and a 2-input OR gate, the 2-input NOR cell must be created as the first step to designing the 2-input OR gate given the INV cell has been designed from the previous assignment. Given the triviality of this gate, the information will be presented in the following figures without an analysis in lieu of the purpose of the report.

Figure 10 displays the layout and simulation report of the total amount of MOSFETs used (number of poly gates). Figure 9 displays the test schematic, and Figure 11 the simulation waveforms.

```
******  Circuit Statistics  ******
# nodes        =        6 # elements   =        7
# resistors    =        0 # capacitors =        0 # inductors   =
# mutual_inds  =        0 # vccs       =        0 # vcvs        =
# cccs         =        0 # ccvs       =        0 # volt_srcs   =
# curr_srcs    =        0 # diodes     =        0 # bjts        =
# jfets        =        0 # mosfets    =        4 # U elements  =
# T elements   =        0 # W elements =        0 # B elements  =
# S elements   =        0 # P elements =        0 # va device   =
# vector_srcs  =        0 # N elements =        0
```

Figure 9: (LEFT) 2-input NOR layout, (RIGHT) BOM for 2-input NOR

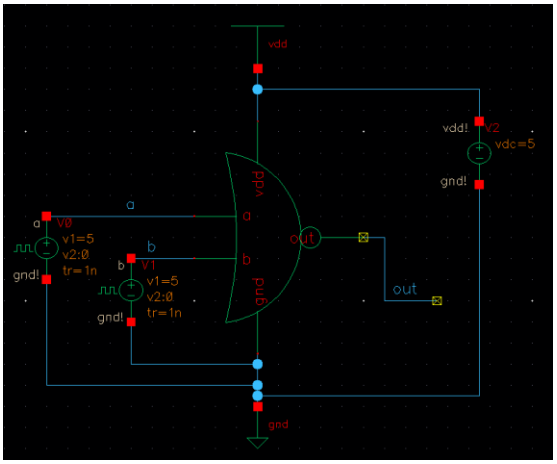


Figure 10: Test schematic for 2-input NOR

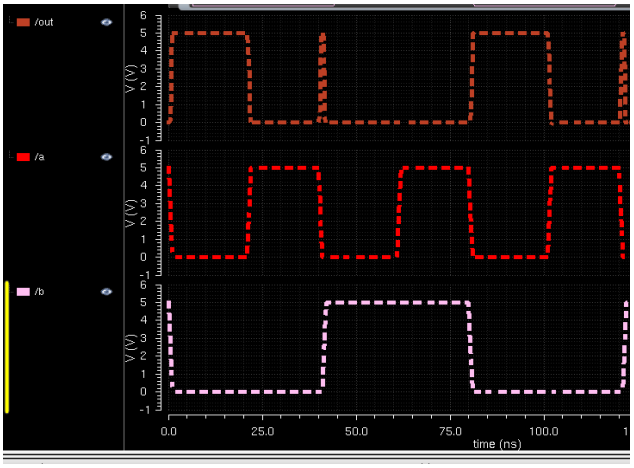
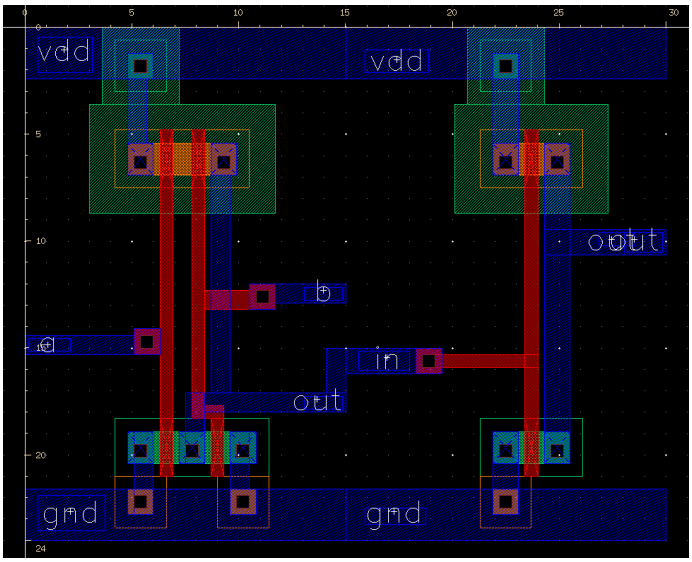


Figure 11: Simulation waveforms for 2-input NOR

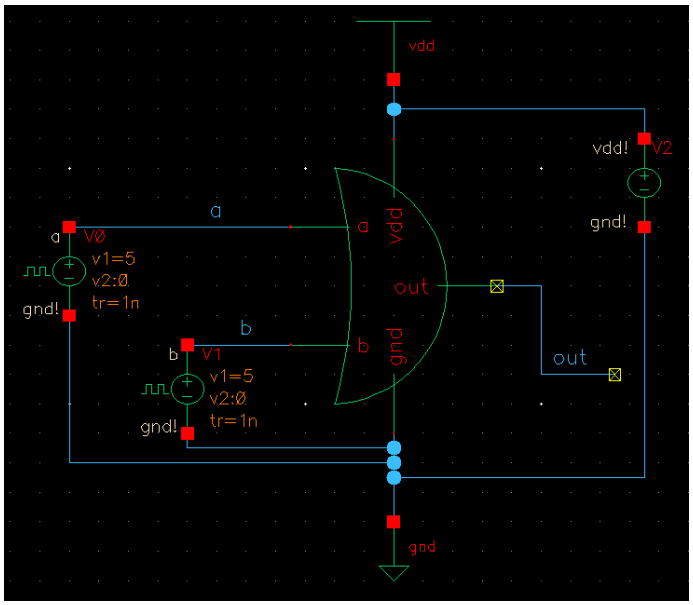
2-INPUT OR



Now, the 2-input OR cell is built using the 2-input NOR and INV cells from the author's Virtuoso library. For the same reasons as delineated in the previous section, this design will be presented without analysis as well.

```
******  Circuit Statistics  ******
# nodes       =        7 # elements   =        9
# resistors   =        0 # capacitors =        0
# mutual_inds =        0 # vccs       =        0
# cccs        =        0 # ccvs       =        0
# curr_srcs   =        0 # diodes     =        0
# jfets       =        0 # mosfets    =        6
# T elements  =        0 # W elements =        0
# S elements  =        0 # P elements =        0
# vector_srcs =        0 # N elements =        0
```

Figure 12: (LEFT) CMOS layout of 2-input OR. (RIGHT) BOM for 2-input OR



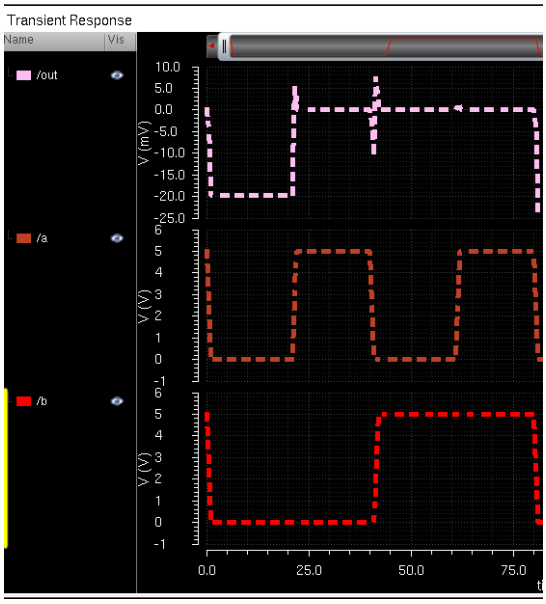


Figure 13: (LEFT) Test schematic for 2-input OR. (RIGHT) Simulation waveforms for 2-input OR.

FULL-ADDER

With all the fundmantal gates necessary, as well as the underlying building block (half-adder) ready, the full-adder is now presented. The strinking difference between the half-adder and the full-adder is the addition of a carry-in (cin) bit which allows the cell to be daisy chained to form higher-order adders. The only other difference is that it takes two half-adders to create the full-adder, hence the name of the former.

First off, it is imporatnt to mention a 10 λ break is inserted between both half-adder cells to break the connecting carry out of the half adder module with the a input of the other half adder module. Figure 14 displays the underlying cells in the full-adder, their pinouts outlined with gold, and the break area outlined in green. To better illustrate the design choice of adding the 10 λ extension, Figure 15 shows the area zoomed in with the underlying CMOS components. Given a cell-based design is used, future implementations should take into account pin locations to reflect the chaining of cells to make larger designs in an effort to save subtrate space and metal routing.

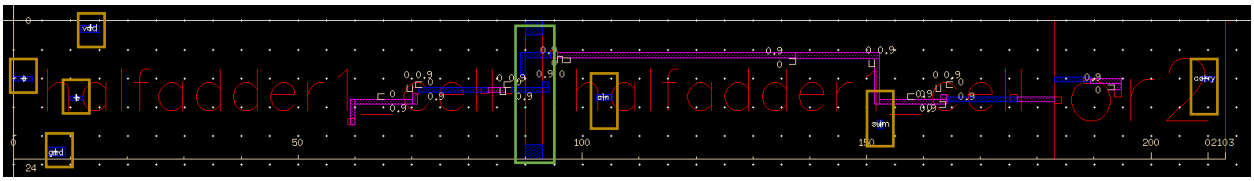


Figure 14: Full-adder cell-based view layout

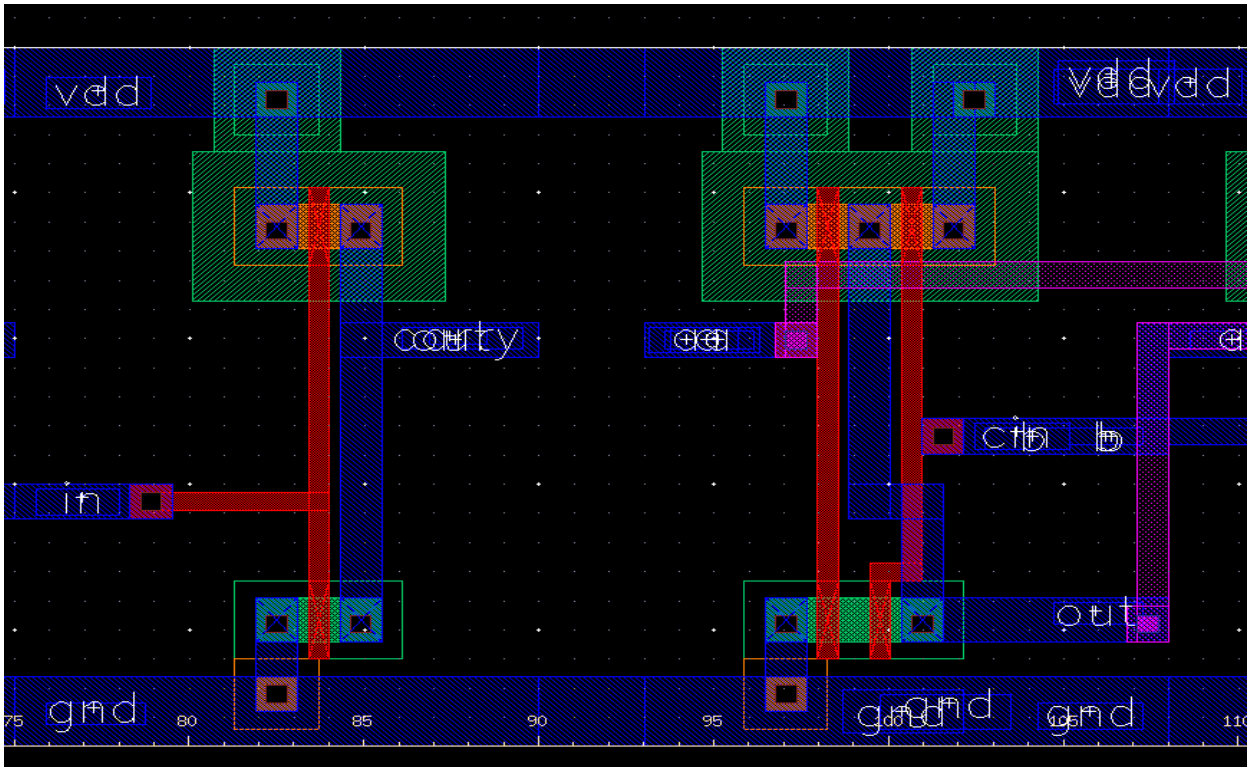


Figure 15: Extension between 90 micron and 93 micron

The full-adder layout is presented in Figure 16. As the rulers showcase on the top section of the figure, the design is 24 micron x 210.3 micron, or 80 λ x 701 λ. The 0.9 micron measurement rulers around the layout are remaining from the design process tailored to save metal surface area. The bottom section of the figure better illustrates the routing practices of the design: inputs and outputs are routed to the edges of the cell to meet convention, and power rails have the same width across the underlying cells for more seamless implementation. To reiterate, the author concedes indeed there is wasted substrate space on this layout, but for learning purposes, it meets the desired goal for learning a cell-based design.

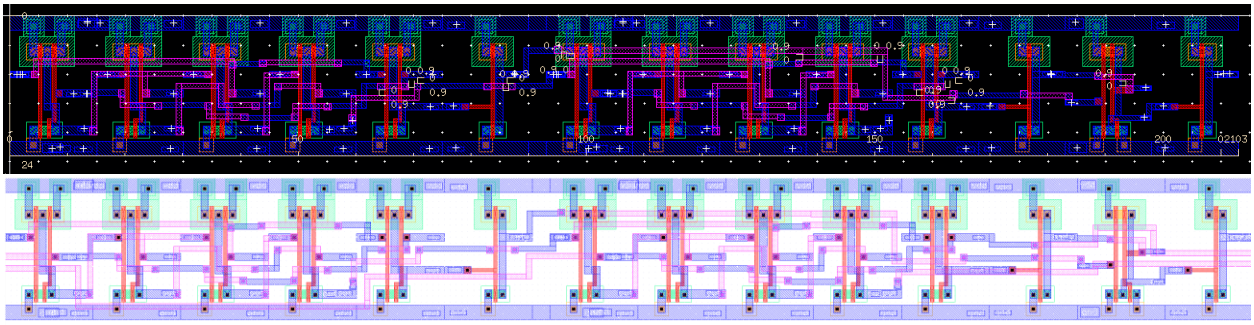
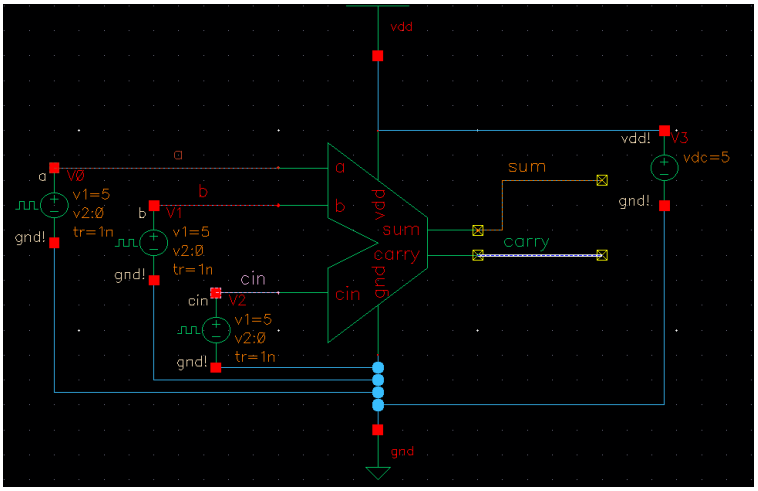


Figure 16: (TOP) Virtuoso screencap of full-adder layout. (BOTTOM) Exported layout from Virtuoso in PNG

Next, the full-adder is simulated to test for desired operation. Figure 17 displays the test schematic used. Given the full-adder takes in 3 signal inputs, a total of 2^3 combinations must be tested against to verify the desired operation of the full-adder; the full-adder can be seen as taking a 3-bit input, and with a little-endian approach, input a is the MSB, and input cin is the LSB. Three square waves are therefore used to probe the circuit, with the MSB having the longest period, and the LSB having the fastest period—similar to the approach used for the previous sections.



Since the half-adder design uses 22 MOSFETs and the OR gate uses 6 MOSFETs, adding up the underlying cells gives a total of 50 MOSFETs needed to realize the design shown in Figure 17.

```
******  Circuit Statistics  ******
# nodes       =      30 # elements   =      54
# resistors   =       0 # capacitors =       0
# mutual_inds =       0 # vccs       =       0
# cccs        =       0 # ccvs       =       0
# curr_srcs   =       0 # diodes     =       0
# jfets       =       0 # mosfets    =      50
# T elements  =       0 # W elements =       0
# S elements  =       0 # P elements =       0
# vector_srcs =       0 # N elements =       0
```

Figure 17: (LEFT) Test schematic for full-adder. (RIGHT) BOM for full-adder.

Figure 18 displays the simulation waveforms for the full-adder and verifies the desired outputs. Now, the 4-bit RCA can be built as the upcoming building block for the 8-bit RCA.

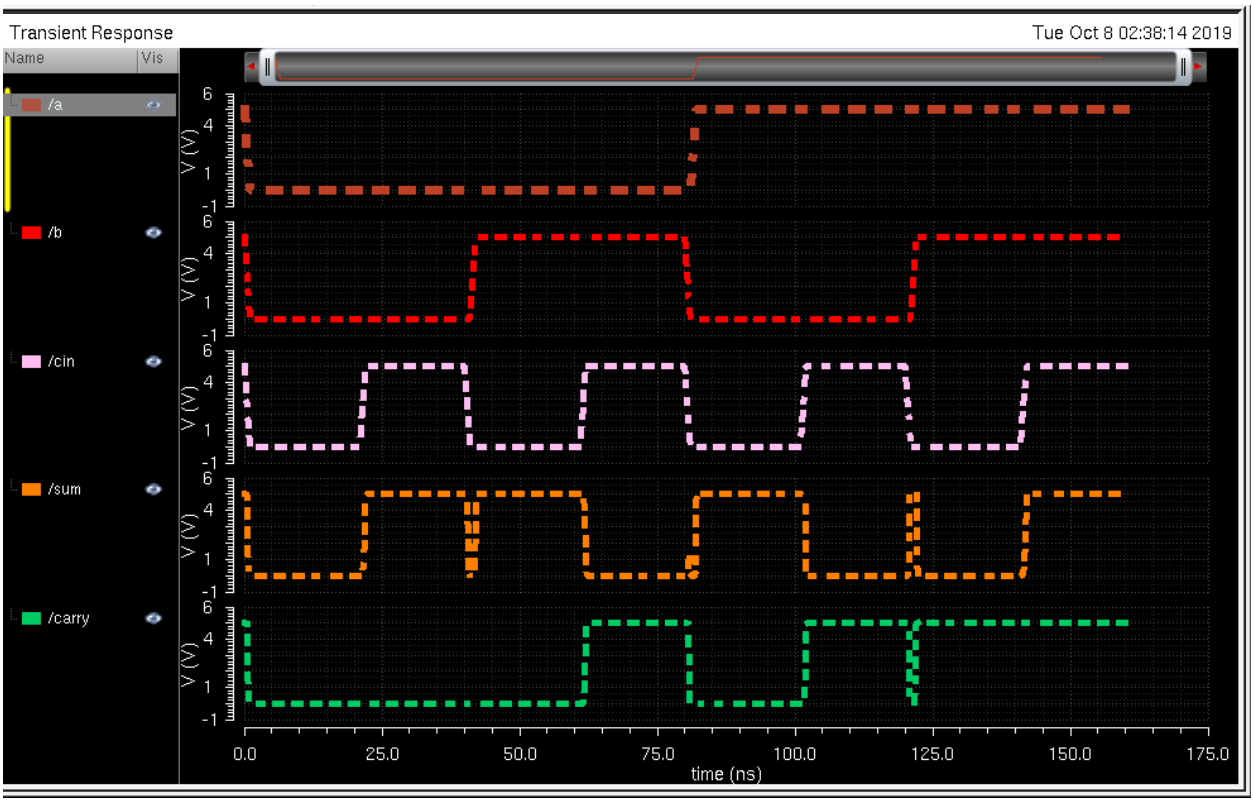


Figure 18: Full-adder simulation waveforms

4-BIT RIPPLE-CARRY ADDER

The fundamental building block of the 4-bit RCA has been built—the full-adder. To build the RCA, four full-adders are daisy chained by propagating the cin from the LSB, all the way to the MSB's adder. The 4-bit RCA takes in two 4-bit numbers as well as a carry-in, and outputs 1 4-bit number in addition to a carry-out.

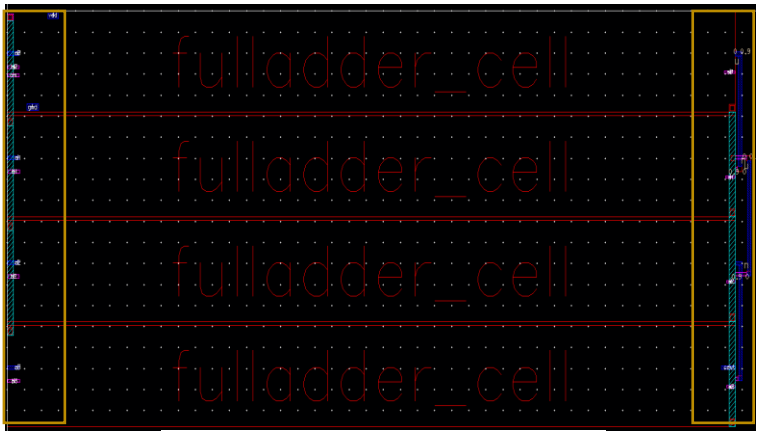


Figure 19: Cell-based view of the 4-bit RCA

In the CMOS layout window, the full-adders are now stacked vertically to better "ripple" the carry across the LSB to the MSB as shown in Figure 19. On the left and right, power and gnd are routed across the cells using the Metal 3 layer and the carry is rippled on the right using the Metal 1 and Metal 2 layers.

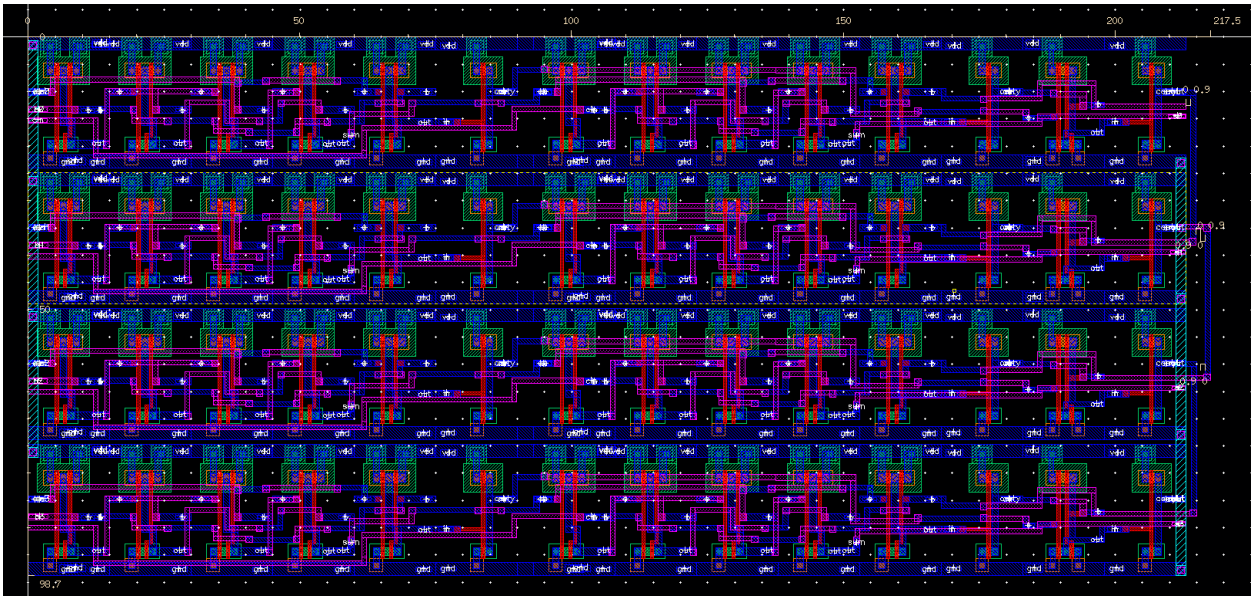


Figure 20: Full layout of 4-bit RCA

It is important to mention the pin routing at the outsides of the half-adder cells surely helped to design the RCA. Otherwise, the layout could not have been neatly stacked and routed through the perimeter of the cell. Figure 20 shows the full layout of the 4-bit RCA. The dimensions of the RCA cell are 98.7 micron x 217.5 micron, or about 328.3 λ x 725 λ—the width is not in discrete measurements of λ and will have to be changed accordingly. Further, between each full-adder cell lies a 0.9 micron or 3 λ spacing to meet the DRC of the 0.6 micron technology. Now the design will be tested using the circuit schematic shown in Figure 21.

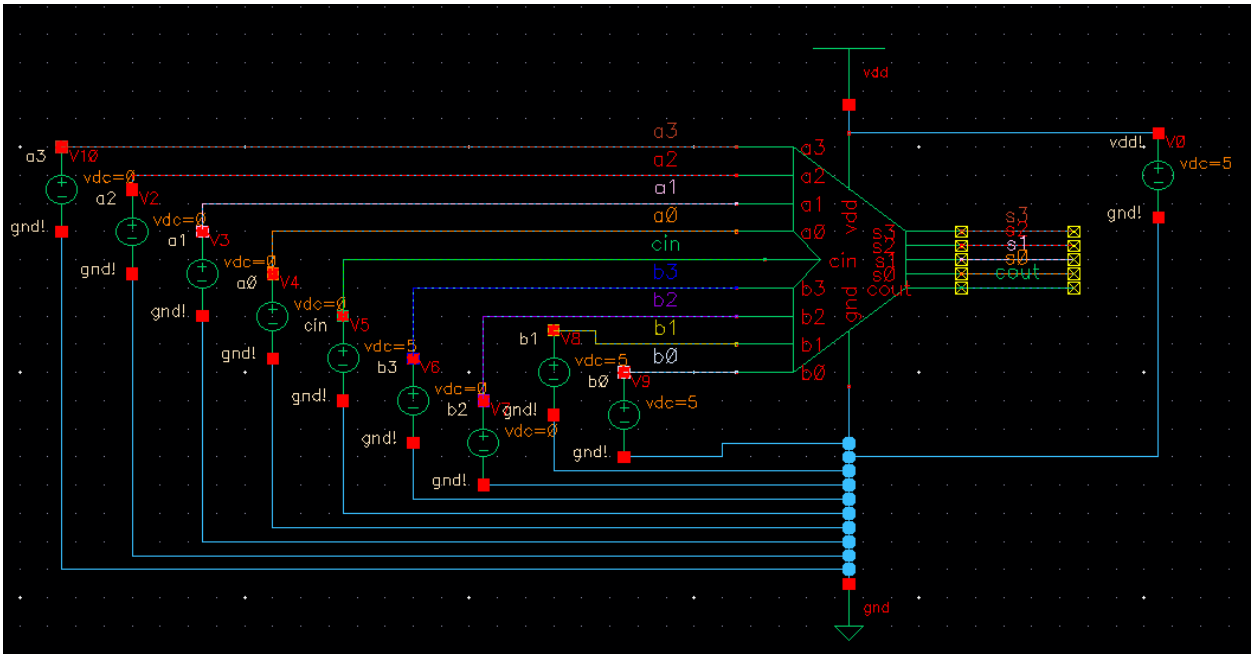


Figure 21: Test schematic for the 4-bit RCA

Compared to the full-adder simulation, the 4-bit RCA does not use square waves for the inputs given the increased complexity of the possible combinations needed to verify the desired operation of the RCA: 2^9 = 512—these are difficult to test for and see at the simulation results window. Therefore, the RCA is tested with five distinct cases, and DC voltage sources are to explicitly defined the bits to input to the RCA. The table below describes the tests used to verify the operation of the RCA, and Figure 22 displays the total amount of MOSFETs used as printed by the simulation report. The following subsections present each of the tests in the table.

TEST#	A<3:0>	B<3:0>	CIN	SUM<3:0>	COUT
0	0000 (0x0)	0011 (0x3)	1	0100 (0x4)	0
1	1000 (0x8)	0001 (0x1)	1	1010 (0xA)	0
2	1011 (0xB)	0100 (0x4)	0	1111 (0xF)	0
3	1110 (0xE)	0001 (0x1)	1	0000 (0x0)	1
4	1110 (0xE)	0001 (0x1)	0	1111 (0xF)	0

```
******  Circuit Statistics  ******
# nodes        =     111 # elements   =     210
# resistors    =       0 # capacitors =       0 # inductors    =
# mutual_inds  =       0 # vccs       =       0 # vcvs         =
# cccs         =       0 # ccvs       =       0 # volt_srcs    =
# curr_srcs    =       0 # diodes     =       0 # bjts         =
# jfets        =       0 # mosfets    =     200 # U elements  =
# T elements   =       0 # W elements =       0 # B elements  =
# S elements   =       0 # P elements =       0 # va device    =
# vector_srcs  =       0 # N elements =       0
```

Figure 22: (TOP) Tests used to verify operation of 4-bit RCA. (BOTTOM) BOM of 4-bit RCA

TEST 0

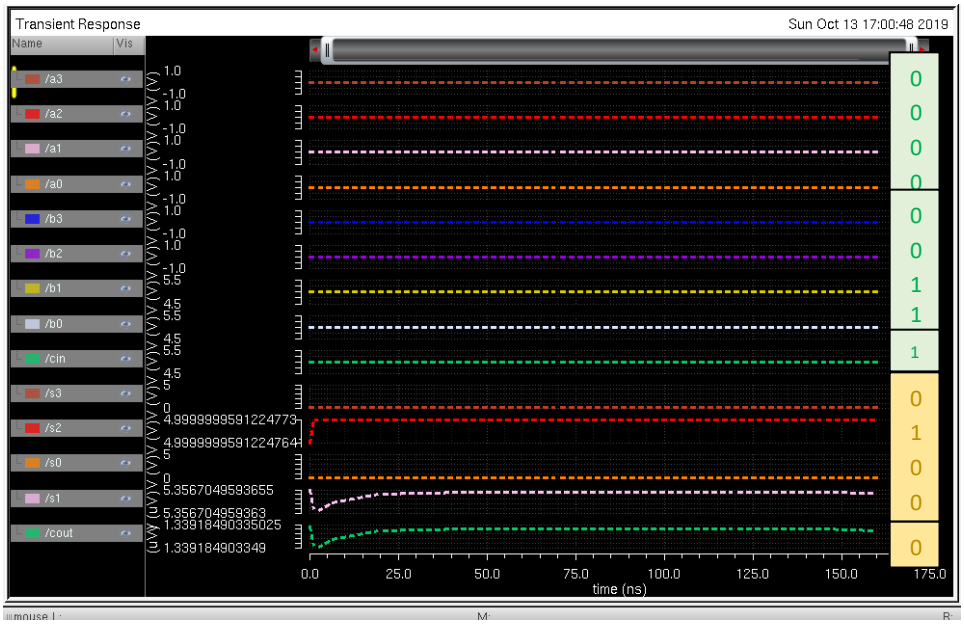


Figure 23: Test 0 for the 4-bit RCA

For this and the following tests, outputs in µV are insignificant for the purposes of this assignment, and these can be defined as logic 0. Figure 23 displays Test 0.

TEST 1

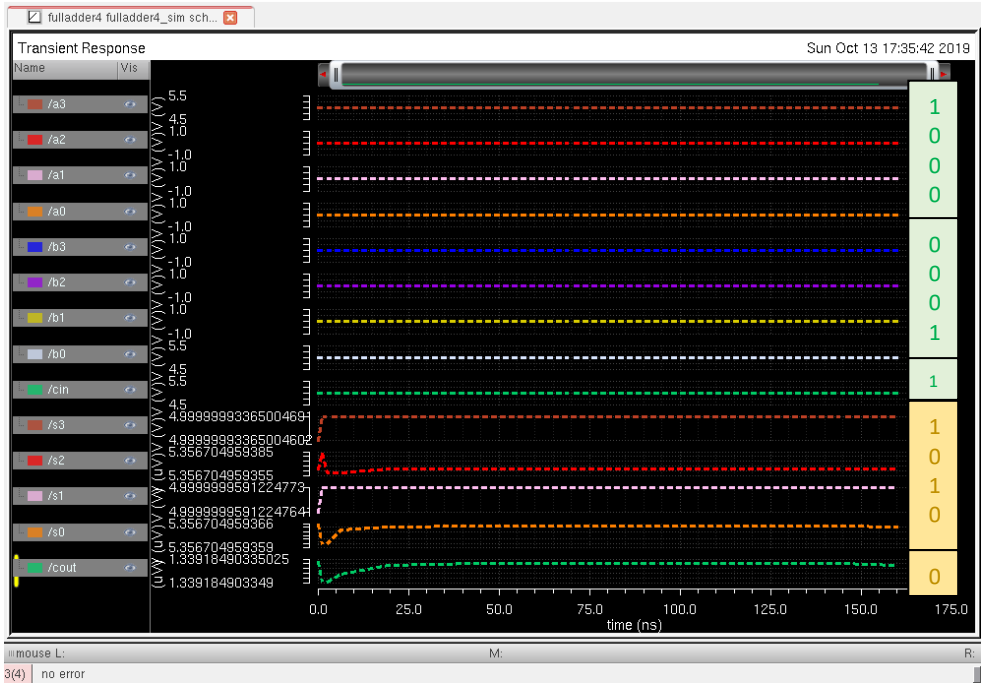


Figure 24: Test 1 for the 4-bit RCA

TEST 2:

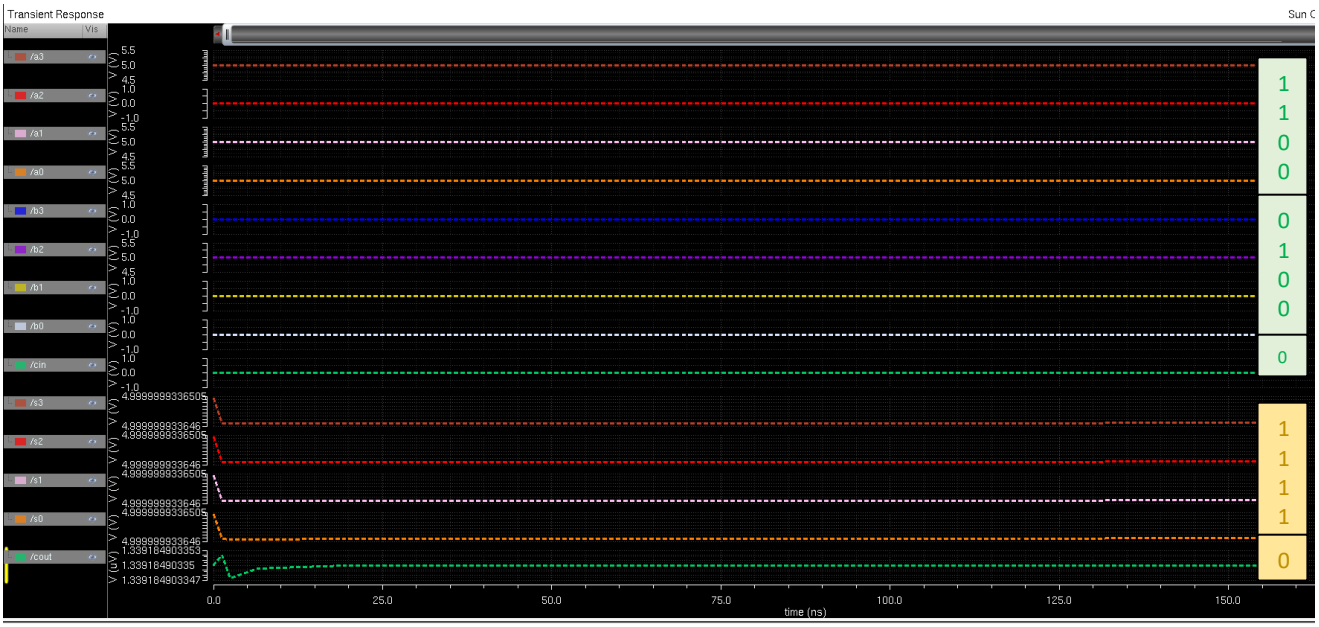


Figure 25: Test 2 for the 4-bit RCA

TEST 3:

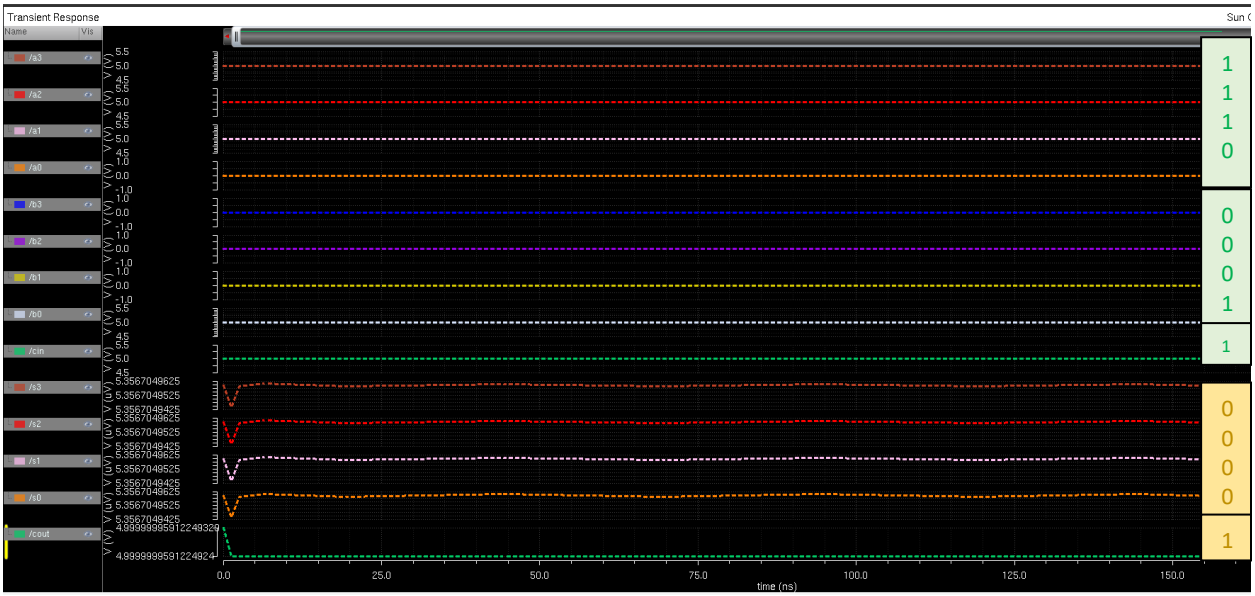


Figure 26: Test 3 for the 4-bit RCA

TEST 4:

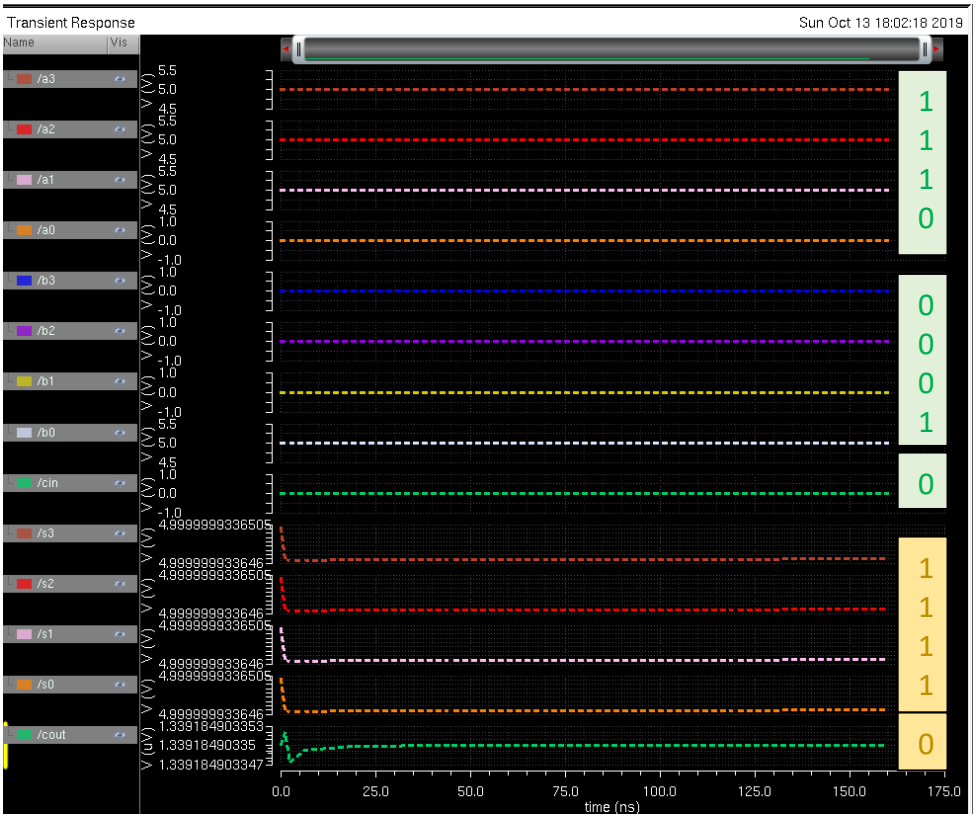


Figure 27: Test 4 for the 4-bit RCA

8-BIT RIPPLE-CARRY ADDER

To build the 8-bit RCA, two 4-bit RCA cells are daisy chained as shown in Figure 28. For this design, the dimensions are 98.7 micron x 439.5 micron, or 329λ x 1465λ. The substrate size has increased significantly, and the empty substrate areas as pointed out in the preliminary sections of this report have enlarged the design more than it needs to be. For an assignment where cell-based design is not the primary focus, the waste of space can be taken into account to create the most efficient design of the full-adder cell to reduce the wafer area significantly.

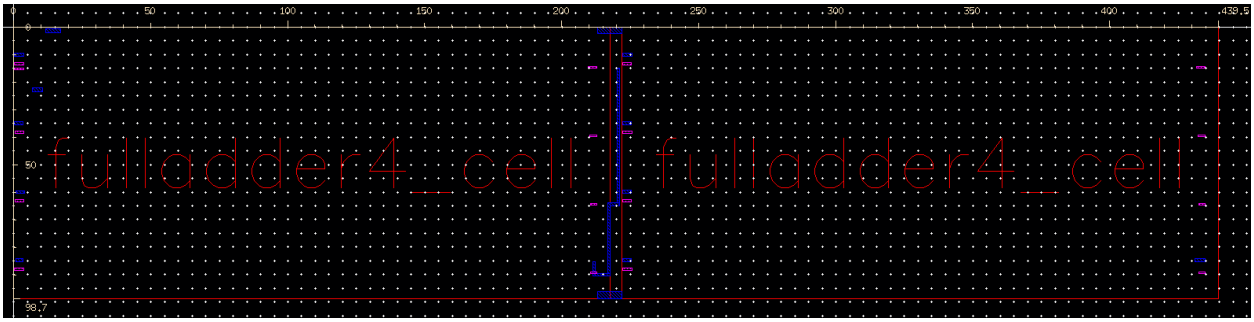


Figure 28: Fundamental cells of the 8-bit RCA

The pink and blue rectangles on the left area of Figure 28 are the input pins for the lower 4-bits of the RCA, as well as the VDD and GND pins. The “ripple” carry propagations can be seen in the middle of Figure 28. Both Metal 1 and Metal 2 layers are used to route the carry from the LSB to the MSB. The right area of Figure 28 shows the pink rectangles as the output bits of Sum<7:4> as well as the cout bit. The 8-bit RCA is presented in Figure 29 at the CMOS layout level in two versions: a screencap, and an export from Cadence Virtuoso. Metal 3 is explicitly used for VDD and GND routing across the cells.

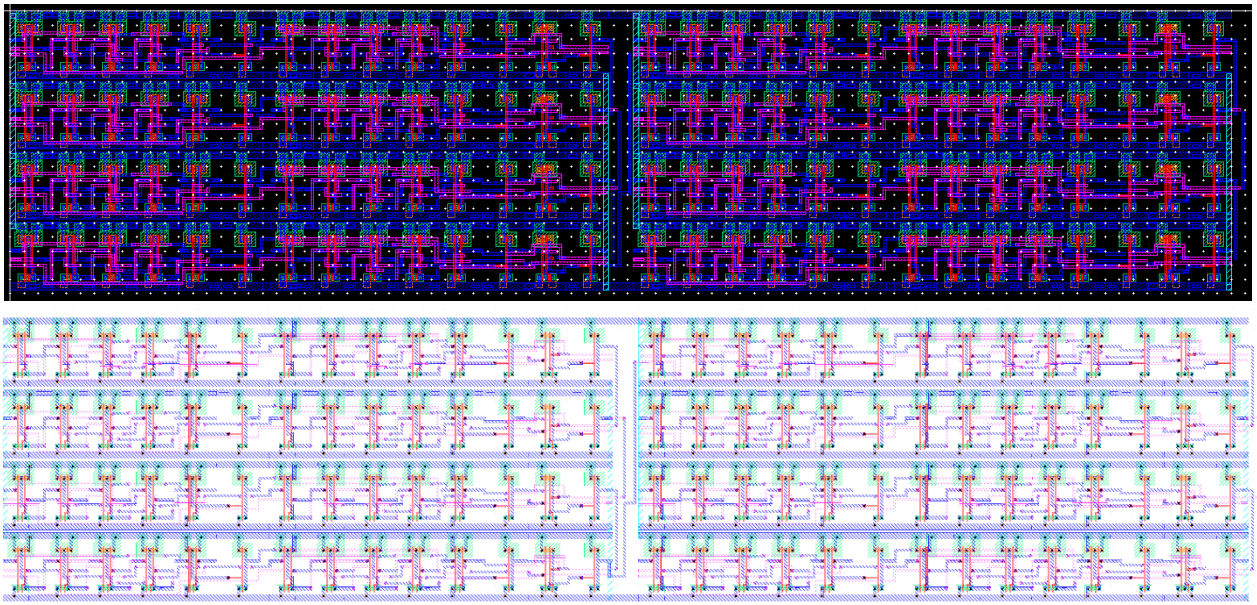
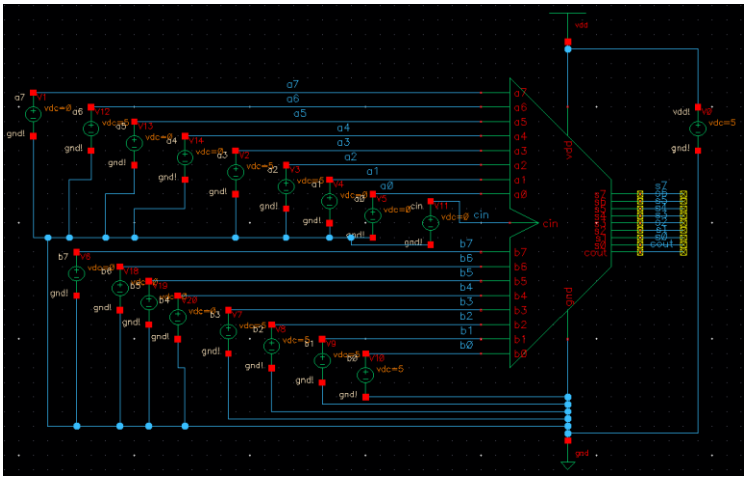


Figure 29: CMOS level layout of the 8-bit RCA



```
******  Circuit Statistics  ******
# nodes       =      219 # elements   =      418
# resistors   =        0 # capacitors =        0
# mutual_inds =        0 # vccs       =        0
# cccs        =        0 # ccvs       =        0
# curr_srcs   =        0 # diodes     =        0
# jfets       =        0 # mosfets    =      400
# T elements  =        0 # W elements =        0
# S elements  =        0 # P elements =        0
# vector_srcs =        0 # N elements =        0
```

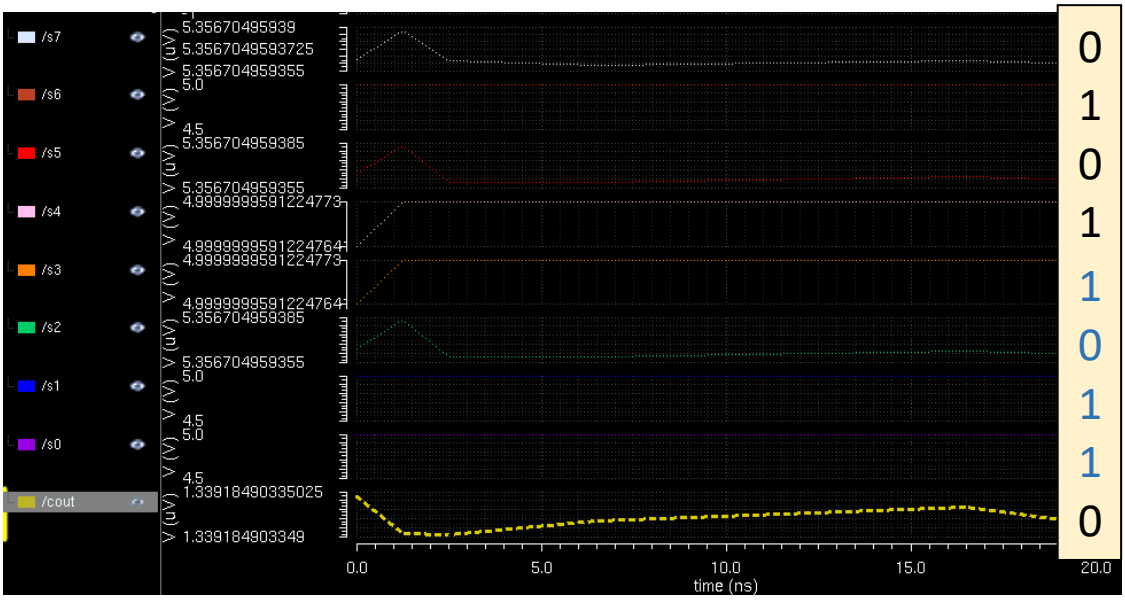
Figure 30: (LEFT) Test schematic for 8-bit RCA. (TOP) BOM for the 8-bit RCA

Compared to the 4-bit RCA, the 8-bit RCA uses double the amount of MOSFETs at a whopping 400. To simulate the RCA, a test schematic is created as shown in Figure 30. Once again, discrete DC voltage sources are used to verify the operation of the RCA given the complexity of the possible input combinations (2^17 = 131072).

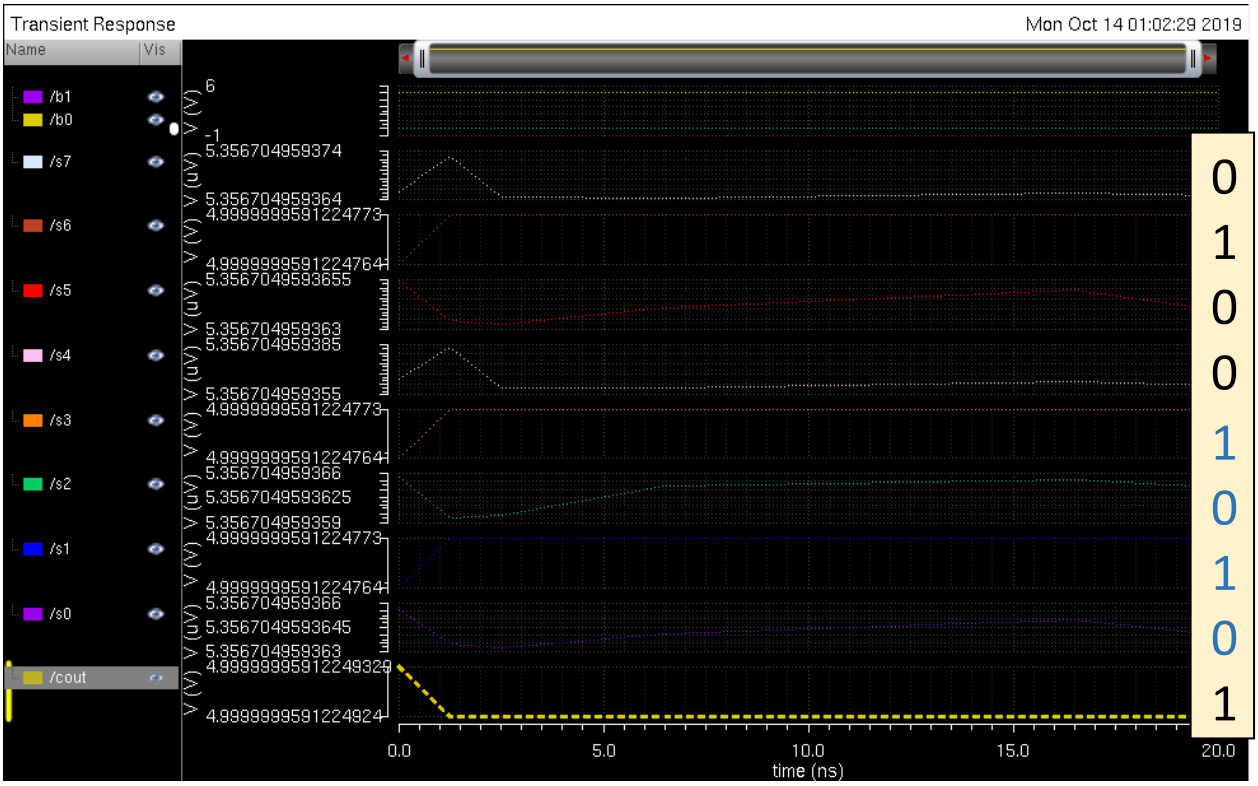
The design is tested against four distinct cases presented below, and the following subsections demonstrate the verification of the design for desired operation. It is important to note that this design will not showcase all of the input waveforms given the limited screensize of Virtuoso—only the 9-bit output will be placed.

TEST#	A<7:0>	B<7:0>	CIN	SUM<7:0>	COUT
0	01001100 (4C)	00001111 (0F)	0	01011011 (5B)	0
1	D2	77	1	4A	1
2	47	3C	1	84	0
3	BD	9F	0	5C	1

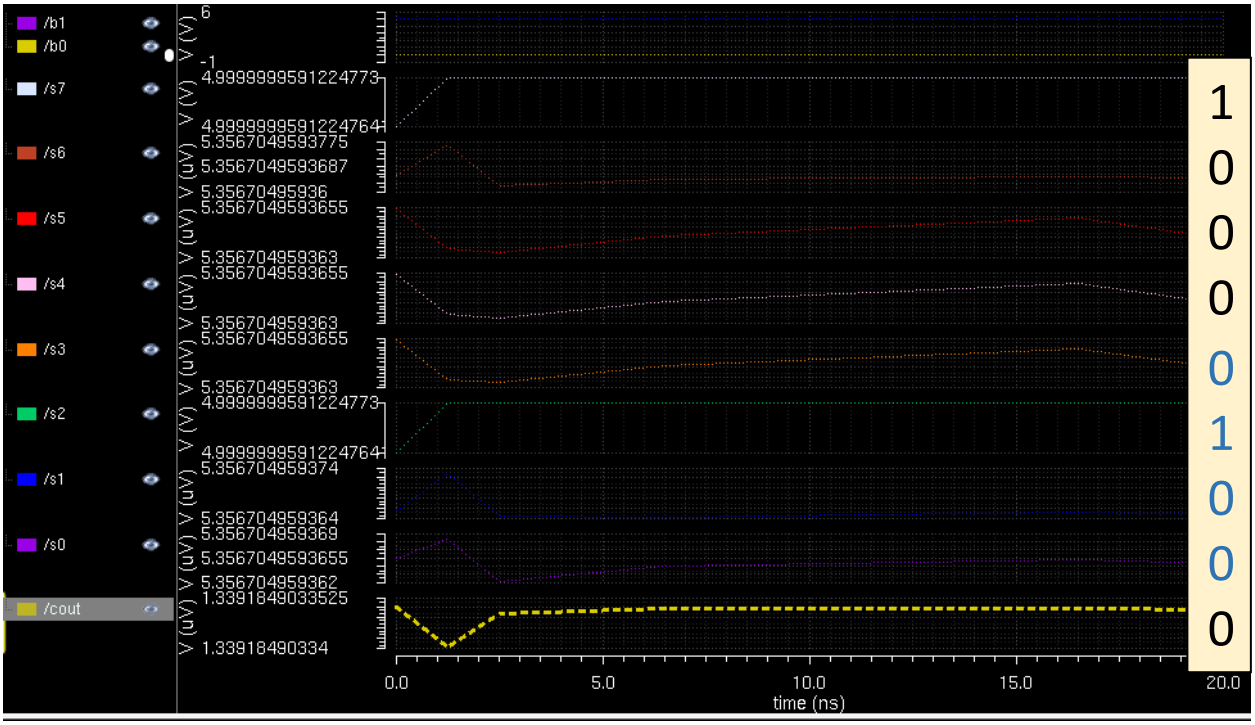
TEST 0



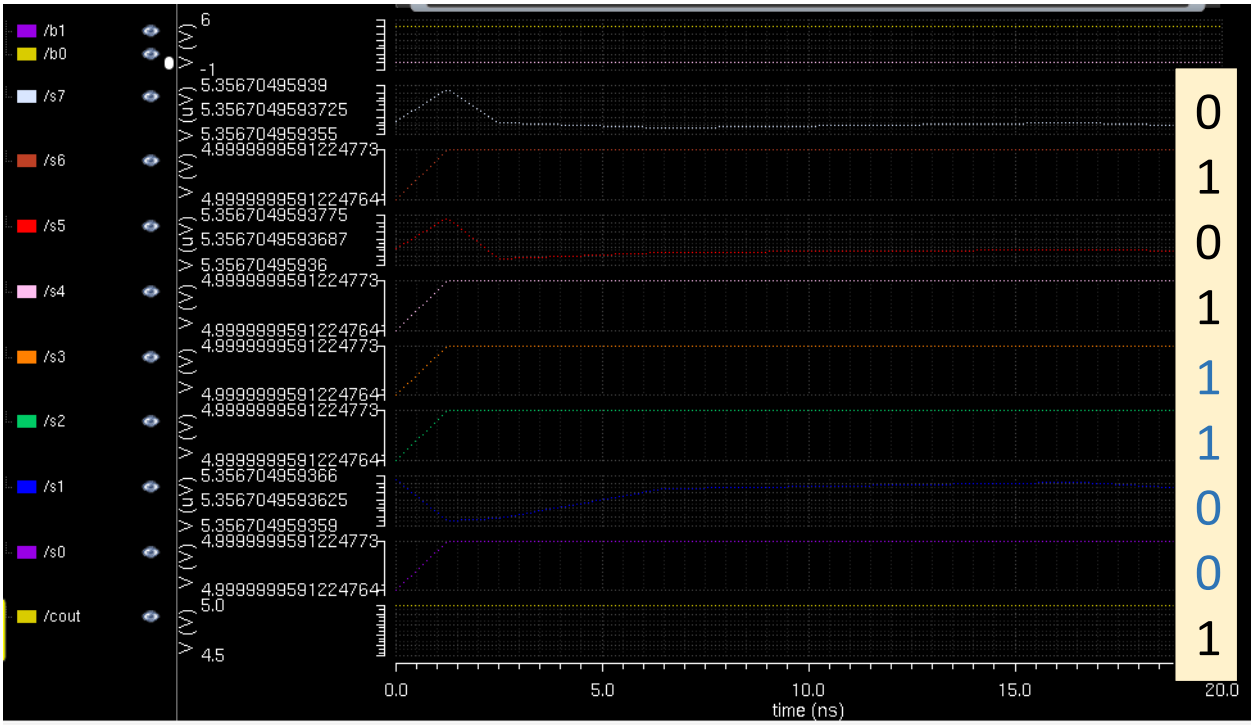
TEST 1


Transient Response
Mon Oct 14 01:02:29 2019
Name
Vis
/b1
/b0
/s7
/s6
/s5
/s4
/s3
/s2
/s1
/s0
/cout
0
1
0
0
1
0
1
0
1
0.0
5.0
10.0
15.0
20.0
time (ns)

TEST 2


/b1
/b0
/s7
/s6
/s5
/s4
/s3
/s2
/s1
/s0
/cout
1
0
0
0
0
1
0
0
0
0.0
5.0
10.0
15.0
20.0
time (ns)

TEST 3



WORST CASE DELAY

The 8-bit RCA has timing constraints whenever the carry bit must “ripple” across all the adders. For example, if the A and B inputs of each adder—from the LSB to the MSB—are both high, then there will be a carry needed to propagate to the next adder. Since the 2-input AND gate needed to propagate the carry to the input of the next adder, this adds a delay to the sum output. Essentially, the nth order adder will have to wait at most n-1 carry propagations, and this can considerably slow down the operation of the adder. Figure 31 shows the worst case delay for the 8-bit RCA.

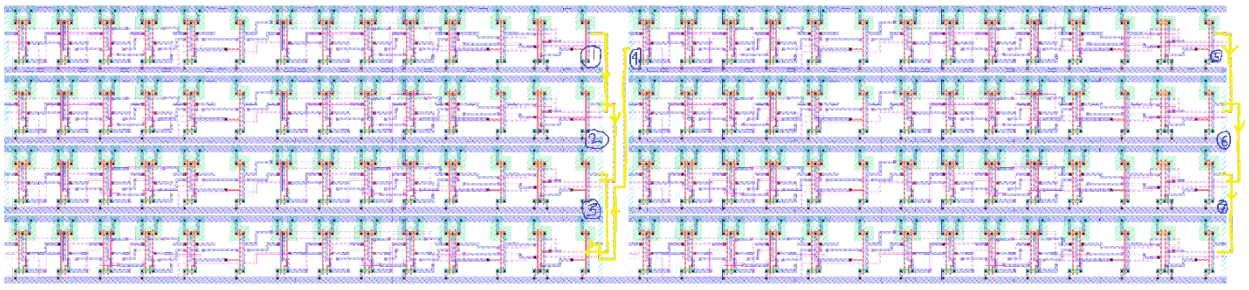


Figure 31: Worst delay for 8-bit RCA takes 7 carry propagations

CONCLUSION

The design of an 8-bit RCA from scratch increased my understanding of cell-based design and what to look out for to increase the efficiency of the layout. For example, the length of each cell that I chose to keep uniform did not matter as much as the empty substrate space did when it came to integrate the entire 8-bit RCA. Further, I learned to space the input and output pins so that the cell can be used in a larger design.

For future implementations, the 8-bit RCA could have all the input and output pins on the perimeter of the design as well as make use of the wasted substrate area. Above all, the assignment allowed me to practice what I learned in class, and I am certain the future assignments will take note of my mistakes and mishaps encountered in this one. I enjoyed building the 8-bit RCA MOSFET by MOSFET.