8-Bit Ripple Carry Adder Design and Simulation

ELEC5503: Digital Microelectronics System Design

Benjamin Riley
21707803

University of Western Australia
June 15, 2020

Contents

List of Figures

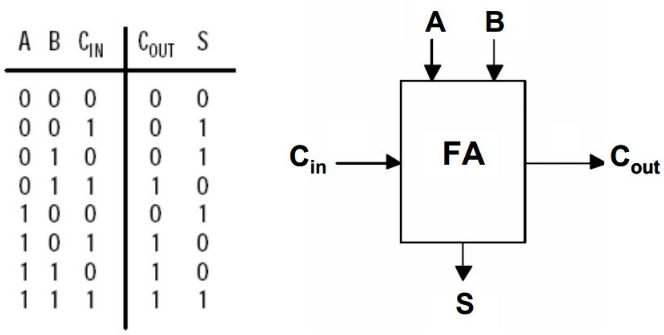
List of Tables

I. Introduction

This project will look at the simulation and design of a full adder integrated circuit and then combine 8 of these blocks to make a 8-Bit ripple carry adder. Cadence Virtuoso is an advanced integrated circuit design suite allowing simulation, layout and post layout simulations to be carried out on custom IC's. This will be used to conduct pre and post layout simulations of a full adder and a 8-bit ripple carry adder.

II. Background

An 8 bit ripple carry adder is an integrated circuit that sums two 8-bit numbers, it achieves this by making use of cascaded full adder circuits. A full adder is a simple circuit that takes 3 inputs; A, B and Carry in. With a 2-bit output consisting of Sum and Carry out, this can be seen in Fig.1. The full adder is cascaded by feeding the carry out of one bit into the carry in of the next bit. This creates a larger circuit with two 8-bit inputs and one carry input, producing a 9-bit output consisting of 8 adder sums and the final carry out.



A	B	C_{IN}	C_{OUT}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Figure 1: *Truth Table and Symbol for a 1-bit Full Adder [1]*

III. Simulation Setups

As specified all simulation inputs use $50ps$ rise and falls times with a $100fF$ load attached to each output. To provide buffered inputs two series unit inverters are used at each input.

IV. Functionality, critical path and delay optimization

i. Full Adder Simulation

The full adder topology and sizing can be seen in Fig.2 this is supplied and needs to be simulated to verify the function. The truth table in Fig.1 shows the expected results. The simulation will use a transient simulation switching A,B and Cin to test all possible inputs in one simulation.

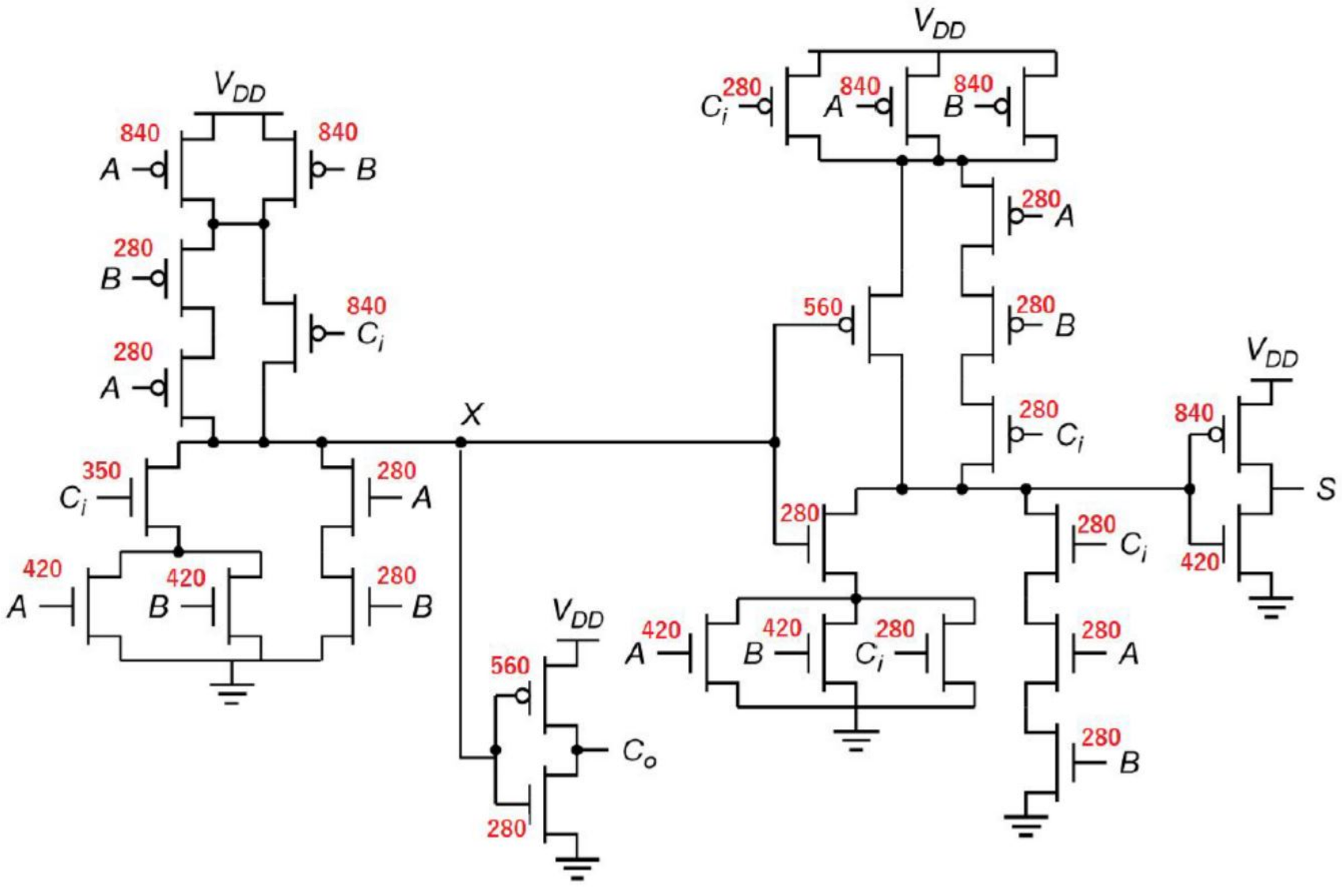


Figure 2: *Supplied Transistor Level Schematic of CMOS Full Adder, Including Transistor Widths in nm. [1]*

The implementation of the supplied full adder topology can be seen in Fig.3. This uses the supplied NMOS ($N\_12\_HSL130E$) and PMOS ($P\_12\_HSL130E$) transistors of United Microelectronics 130nm technology library (umc130mmrf), at the sizes stated in Fig.2.

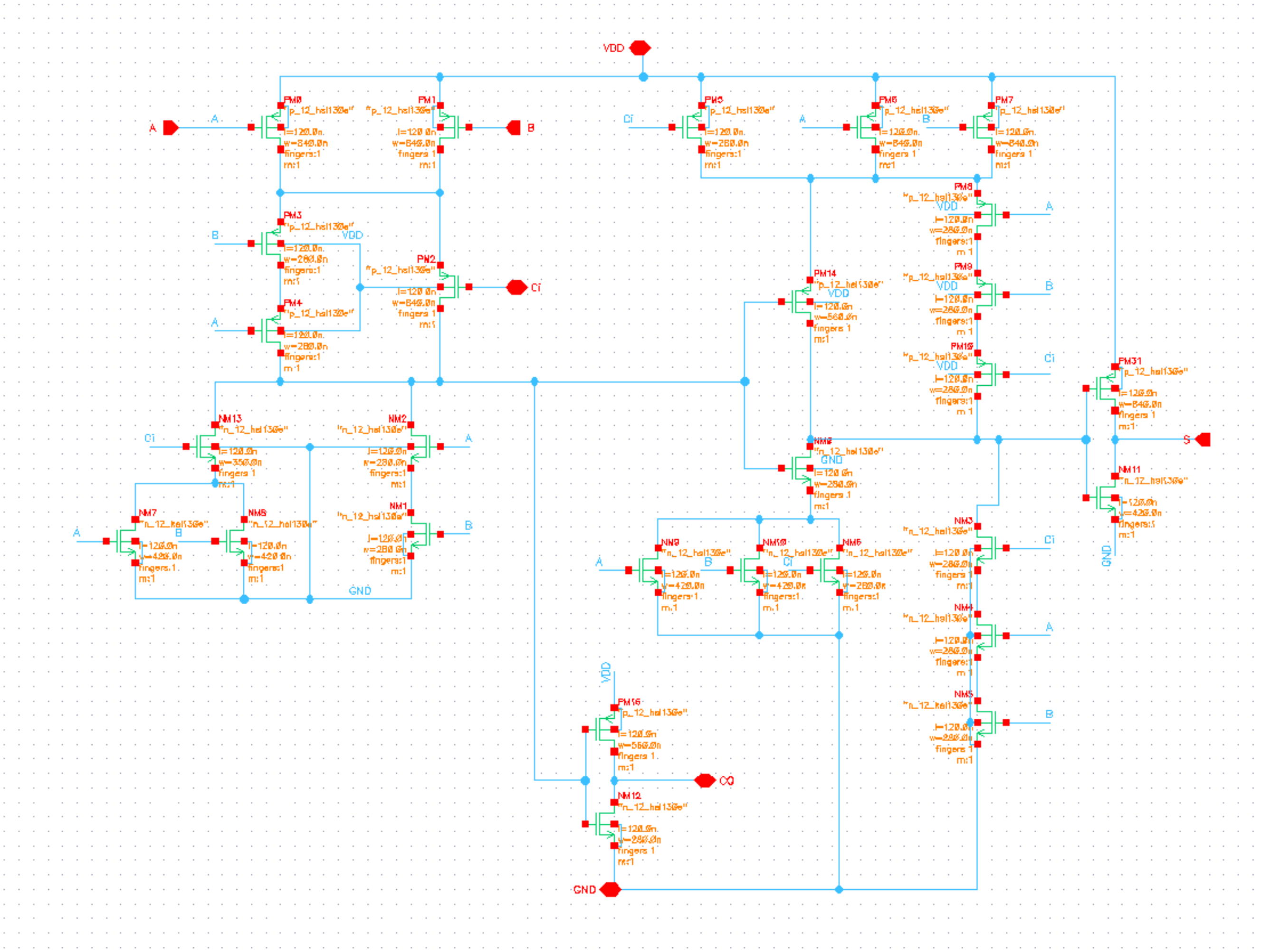


Figure 3: *Cadence Virtuoso Full Adder Schematic.*

All input combinations can be tested in a single simulation using pulse power supplies to drive the three inputs at three different frequencies. Fig.4 shows the simulation schematic in Cadence Virtuoso, it makes use of three pulse generators. Cin has a period of 8ns, B has a period of 16ns and A has a period of 32ns. This allows all 8 input combinations to be tested.

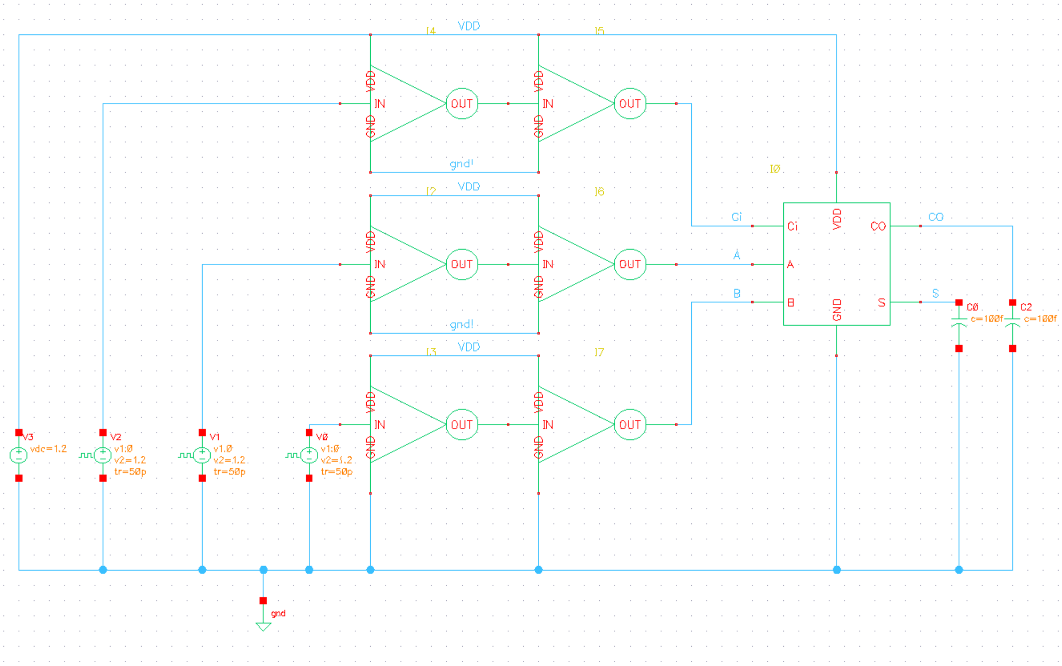


Figure 4: *Cadence Virtuoso Full Adder Simulation Set-Up.*

Fig.5 shows the simulation results of the full adder set-up. It shows the input combinations producing the required outputs seen in Fig.1. This is a theoretical simulation that does not take into account the parasitic capacitances associated with the layout of the full adder.

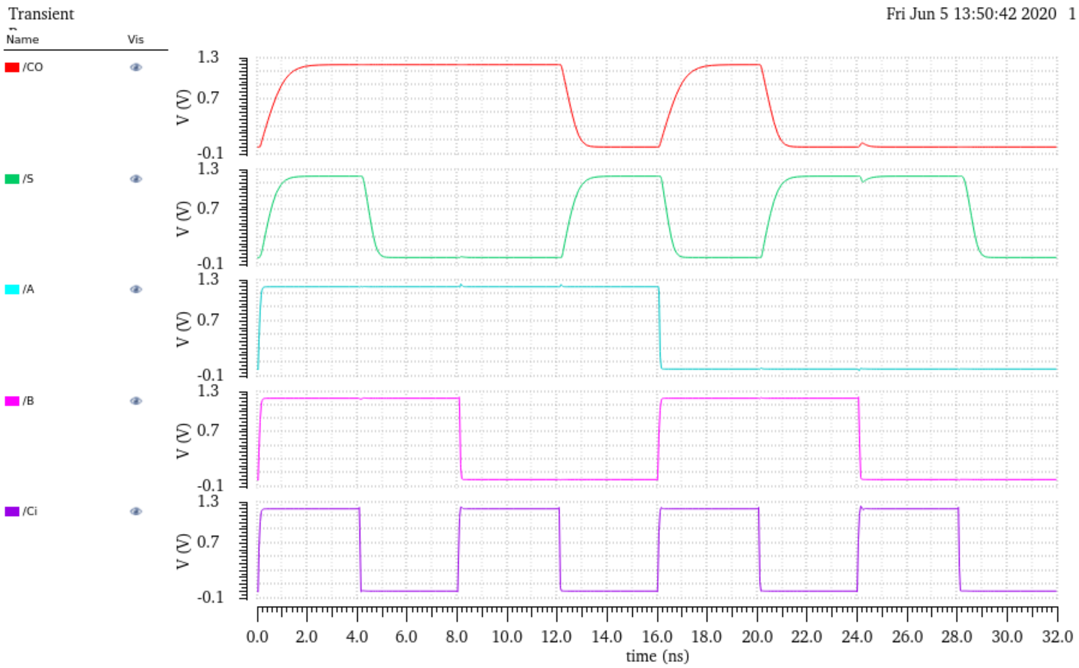


Figure 5: *Transient Simulation Testing All Inputs of Full Adder*

	Delay (ps)	Rise (ps)	Fall (ps)
Sum	497	504	363
Carry Out	553	768	487

Table 1: *Worst Observed Delay, Rise and Fall Times*

Table.1 shows the worst case delays and rise times for both Cout and S from the simulation seen in Fig.5. The worst case delay for S occurs at 28ns and represents a transition from $ABC = 001 \rightarrow 000$. The worse case delay for CO occurs at 16ns and represents a transition from $ABC = 100 \rightarrow 011$. These delays are due to the parasitics associated with the individual transistors used to create the full adder. It is to be expected that when this device is laid out the extra parasitics associated with the wires will increase these observed values.

ii. 8-Bit Adder Simulation

The 8-bit adder is made by cascading 8 full adders together, passing the carry out of the previous section into the carry in of the next full adder. Fig.6 shows a 3-bit section of the 8-bit device, represent how the carry outs cascade into the carry in of the next device. This combination allows the addition of 2 8-bit numbers to produce a 9-bit output.

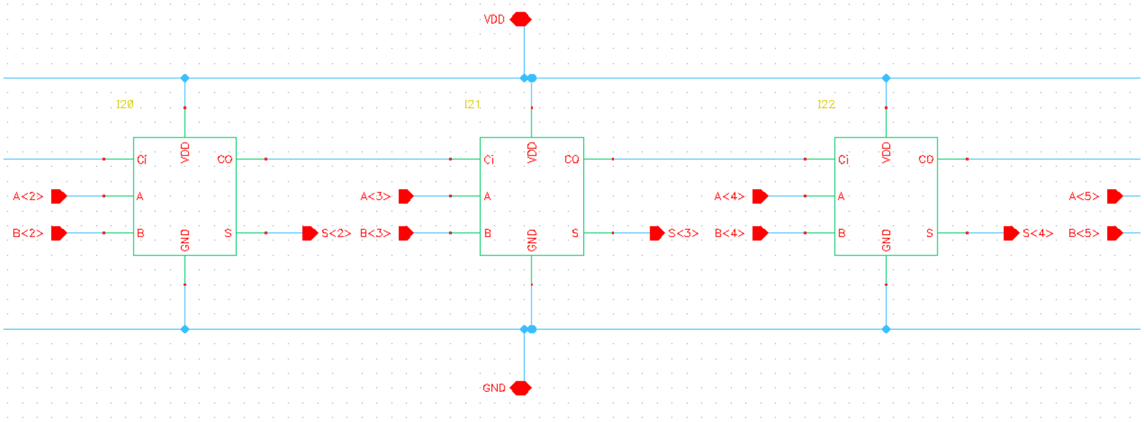


Figure 6: *Section of the 8-Bit Adder, Three Full adders Cascaded.*

To simulate the functionality of the 8-bit ripple carry adder two 8-bit number were choose to test the different combinations of input to different sections of the devices. The inputs used are seen in Table.2 and represent a transition from all 0.

	7	6	5	4	3	2	1	0
A	1	0	1	0	0	0	1	1
B	1	0	0	1	0	1	0	1
Cin	0	0	0	0	1	1	1	1

Table 2: *8-Bit test inputs*

With the expected outputs:

$$S = 00111001$$
$$Co = 1$$

This combination provides testing all 8 combinations in a the individual full adder sections as well as showing the cascading of bit in the 8 bit adder producing the desired 9-bit sum of the two 8-bit numbers. Only bit 0 of Cin is user controlled, with the other bits being a product of the previous adder. Fig.7 shows the 9 output bits from the specified inputs. This graph shows the expected output showing the complete functionality of the device.

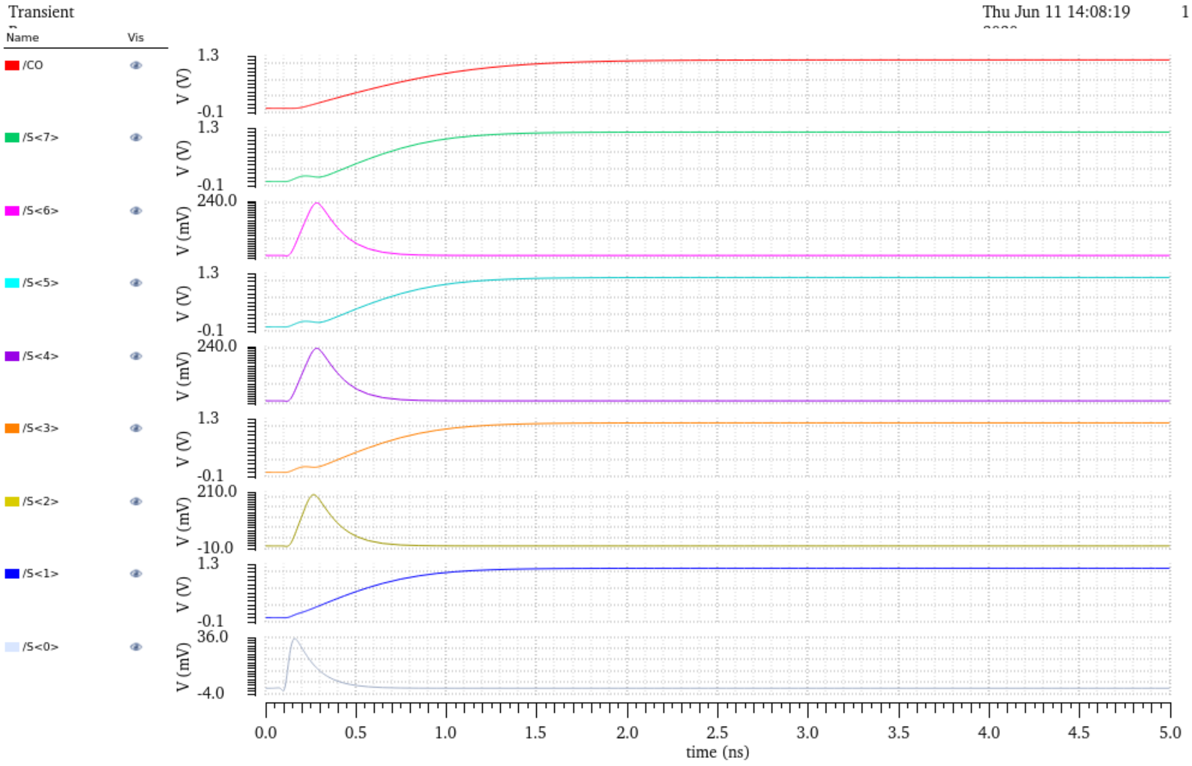


Figure 7: *Transient Simulation Testing All Inputs of the 8-Bit Ripple Carry Adder.*

iii. 8-Bit Adder Critical Path Analysis

The critical path of a circuit is the combination of inputs that will provide the slowest production of the output. In the case of a ripple carry adder this occurs when the final output relies on the carry signal cascading through all 8 bits of the adder. This stacks the delay of the carry signal 8 times. To produce a combination of inputs that would allow each bit to rely on the carry out from the previous bit the following inputs were used:

	7	6	5	4	3	2	1	0
A	1	1	1	1	1	1	1	1
B	0	0	0	0	0	0	0	1
Cin	1	1	1	1	1	1	1	0

Table 3: *8-Bit Critical Path Inputs*

With the output:

$$S = 00000000$$
$$Co = 1$$

This input also uses $A = 1$ and $B = 1$ in the intial bit to trigger the carry out, as this combination has a higher pull down resistance than $C\dot{(}A + B)$, due to the smaller transistors as seen in Fig.2.

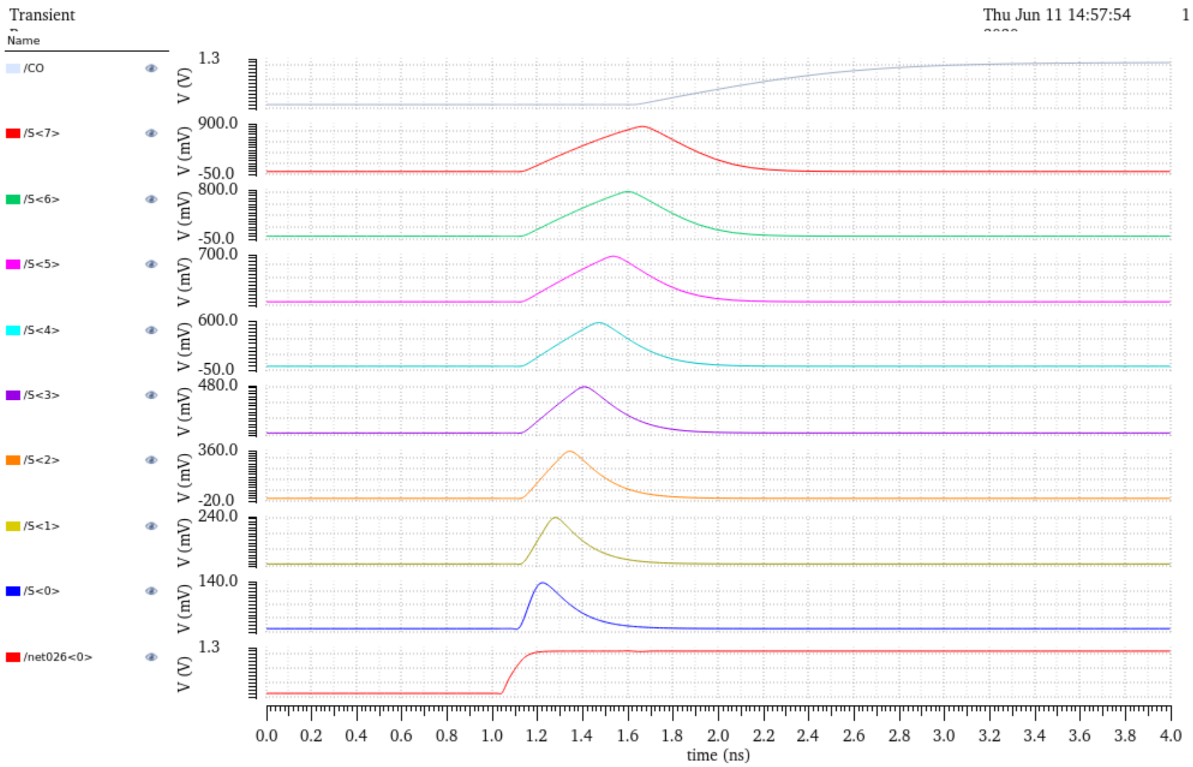


Figure 8: *Critical Path Simulation of the 8-Bit Ripple Carry Adder.*

	Delay (ps)	Rise (ps)
Carry Out	1052	764

Table 4: *Worst Case Delay and Rise Times*

Fig.8 shows the delay in CO based on the inputs specified in Table.3. It can be seen that output CO is heavily delayed in response to the input net026<0>. This is due to the carry signal having to propagating through all 8-bits before this output goes high. This propogation can be seen in $S < 0:7 >$, without the carry bit input all the S bit will equal to 1, as $1+0+0=01$. So they start to rise to the 1 logic level of 1.2V. Once the carry bit reaches that stage they fall back to 0, as $1+0+1=10$, and the carry bit goes high propagating to the next bit. only once it has been through all 8-bits will the final Cout go high. This is due to the Cout 0 → 1 transition having a long rise time of 764, as seen in Section IV.i.

iv. 8-Bit Adder Sizing (Optional)

From the analysis of critical path it is clear that the limiting factor in the 8-bit ripple carry adder is the propagation of the carry signal throughout the 8-bits. Therefore the area to look at to resize is the node X, as seen on Fig.9. The transition of this node from 1 → 0 is responsible for producing the Cout 0 → 1 transition that cascades through the 8-bit ripple carry adder. Therefore in order to increase the speed of this critical path the fall time of this node needs to be decreased. The formula for the transition time from high to low can be seen in Equation 1. This shows that the speed is dependant on the equivilant resistance of the pulldown network (R_{eqn}) and the capacitance at the node.

$$t_{pHL} = ln(2)R_{eqn}C_L \tag{1}$$

Three areas that will affect this can be seen in Fig.9, in three boxes. The first area, the blue box, this contains one large transistor. By decreasing the size of this transistor the capacitance at the node is decreased, due to the reduction of the transistors diffusion capacitance.

The second area is the area of the pull down network in the red box. In the critical path simulation the path through A and Ci is used as the pull down path. Therefore by increasing the size of these the pull down transistors, the resistance is decreased and therefore speed is increased. Inputs A and B need to have the same behaviours and as such the B transistor is also increase in size to match A.

The third area is the area of the pull down network containing two transistors in the green box. These transistor are not used in the middle of the cascade as they do not depend on the carry input but can be used to drive the carry out and start a cascade. They are used for this function in critical path simulation. As such increasing there sizes will not have as large of an effect as the transistors in the red box, but will still contribute and as such if speed is the only goal, they should be increased in size.

The proposed increase in size of the transistors can be seen in Fig.9. In order to demonstrate the effects mentioned where a transistor needs to be increased in size it is increased to the maximum used width of $840nm$ and where it is made smaller its decreased to the minimum used width of $280nm$. This proposed solution will decrease the delay time at the expense of the silicon area required.

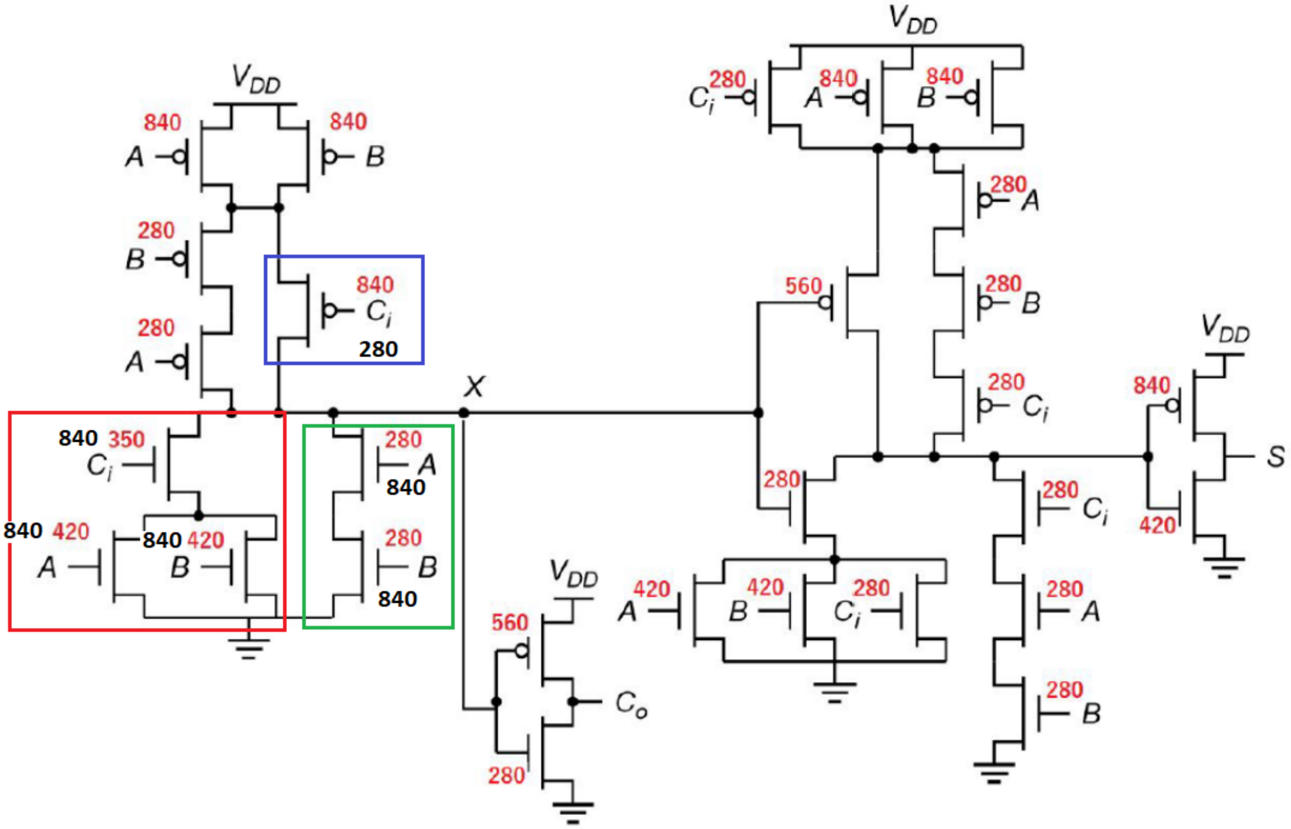


Figure 9: *Proposed Resizing Areas on Full Adder Schematic.*

Fig.10 uses the same simulation setup as Section IV.iii. With the input shown in Table.3. Prior to resizing the delay of the critical path was 1052*ps*, with the changes to the blue box the delay was 1010*ps*. With blue and red box 943*ps*. With all three areas 864*ps*. This represent a significant 18% decrease in the delay time.

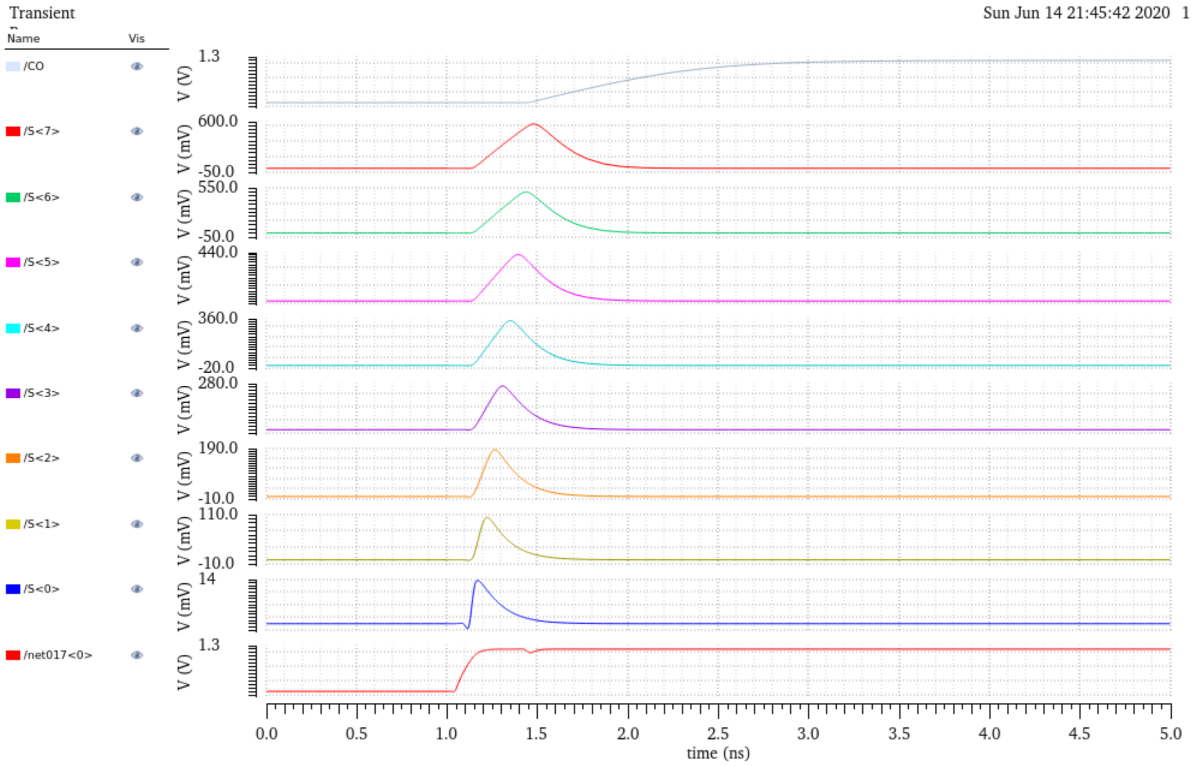


Figure 10: *Critical Path Simulation, With Transistor Resizing.*

V. Layout

i. Full Adder Layout Proposal

The aim when moving from a schematic to a layout is to produce an end result that achieves a combination of the following metrics and the quality of the layout will be assessed using these metrics.

- Achieve desired funcitonality
  The layout needs to contain the connections specified in the schematic to produce the desired outputs
- Minimise silicon area
  The cost of a silicon IC is proportional to the area that the design requires. Therefore it is extremely import to minimise the size of the design to reduce the cost of the final product.
- Minimise the parasitic capacitances and resistances in the circuit
  Extra wires and complex routing will increase the parasitic capacitances and resistance associated with the end result. Therefore to produce a chip that is able to operate at the highest speed careful attention needs to be taken to reduce the connecting metal wires routed in the design. This can be achieved by careful placement of the transistors to reduce routing lengths and the sharing of diffusions to eliminate the need to route connecting metal all together in some scenarios.

Fig.12 shows the preliminary proposal for the layout of the design, this is a result of several hand drawn stick diagrams trying to optimise the placement of transistors by sharing as much diffusion as possible. As can be seen the design is based around 3 main sections, where diffusions can be shared. The far left section represents the logic to produce signal X as seen in Fig.2, the middle section of logic using X as an input and the two output inverters are on the far right.

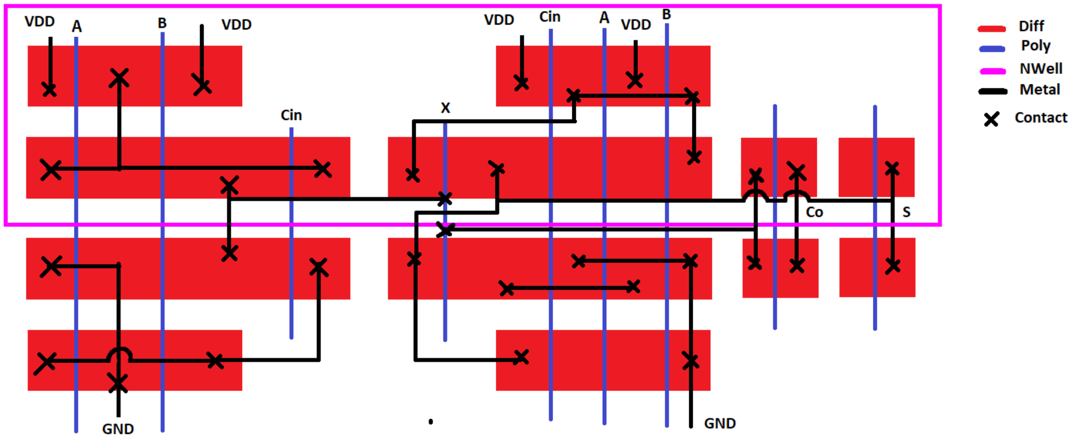


Figure 11: *Full Adder Layout First Plan.*

This layout takes advantage of sharing diffusion layers as best as possible. It re arranges one of the series string seen in Fig.2 in order to allow greater sharing of diffusion.

Careful thought has been put into the location of the inputs and outputs. A and B are accessible from the top and bottom of the design. Cin and Cout will be placed on opposite ends of the device allowing for easy cascading of the device. With the output coming out of the bottom of the device. This is with the intention to lay the 8-bit device out in a 1x8 configuration with the carry cascading from left to right.

ii. Full Adder Layout

The final full adder solution can be seen in Fig.12, this follows closesly the plan shown in Fig.11. The biggest difference is the position of the Cout inverter, after placing the transistors there was a large area in the middle of the design that allowed the the two halves to be split up and placed at the top and bottom in the middle, sharing the already existing X poly.

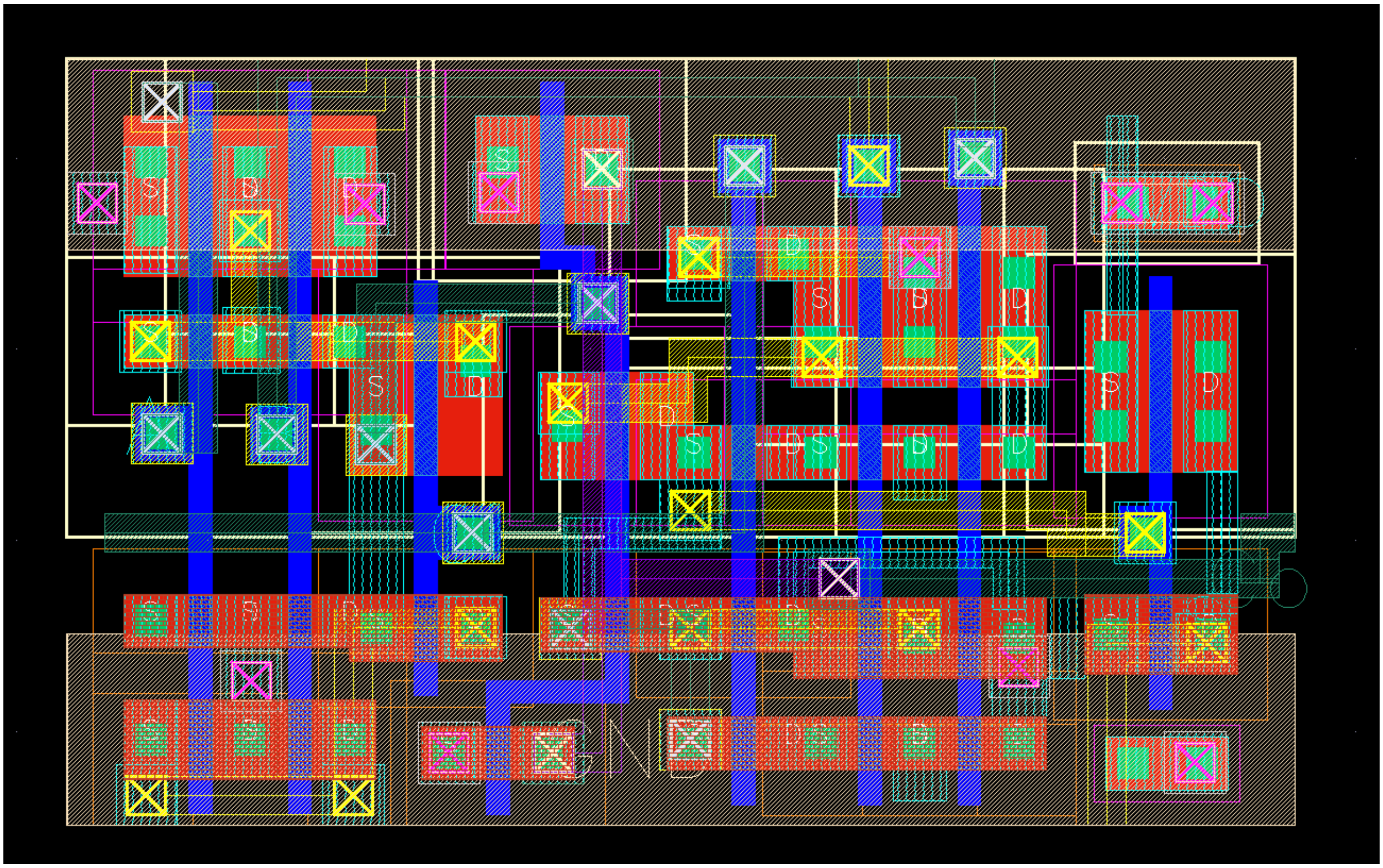


Figure 12: *Full Adder Layout.*

In this layout careful effort was made to reduce the use of higher layer metals and use the shortest possible routing paths, as higher level metals will introduce a large parasitic capacitance due to there proximity to the large metal 6 power planes. Only one use of metal 4 is required with all other traces using metal 3 or below.

The signals are distributed like planned, with A and B coming in from the top on metal 2 and metal 3 respectively. S is at the bottom the design using metal 2. Cin is on the left in a position to match up to Cout on the right side of a cascaded device, these use metal 3. There are two larger power planes for GND and VDD running the length of the device from left to right, these are the minimum required width for a power wire at $1\mu m$. This will allow direct connection to a device on either side, allowing ease of joining when 8 of these device are going to be put side to side.

Another issue that may effect performance of the device are the long A and B input wire under the top VDD Plane. These wire are about $2\mu m$ long and on metals 2 and 3. They will be introducing a significant capacitance between each other and the metal 6 VDD plane. Having the input signals at the top of the device is a worthwhile trade off for this as it allows a cleaner routing setup when cascaded into an 8-bit configuration.

The device has a small footprint with only a few little bits of wasted space, top right corner and middle left. This produced a final silicon footprint of $4.01\mu m \times 6.42\mu m$ for a total area of $25.74\mu m^2$.

iii. Full Adder Post Layout Simulation

A post layout simulation can be completed using Cadences QRC to generate an extracted view containing the extra parasitic capacitances and resistances associated with the physical layout of the device. Fig.13 shows the results of the simulation with the same inputs as Section IV.i. It shows the same function of the device across all input combinations.

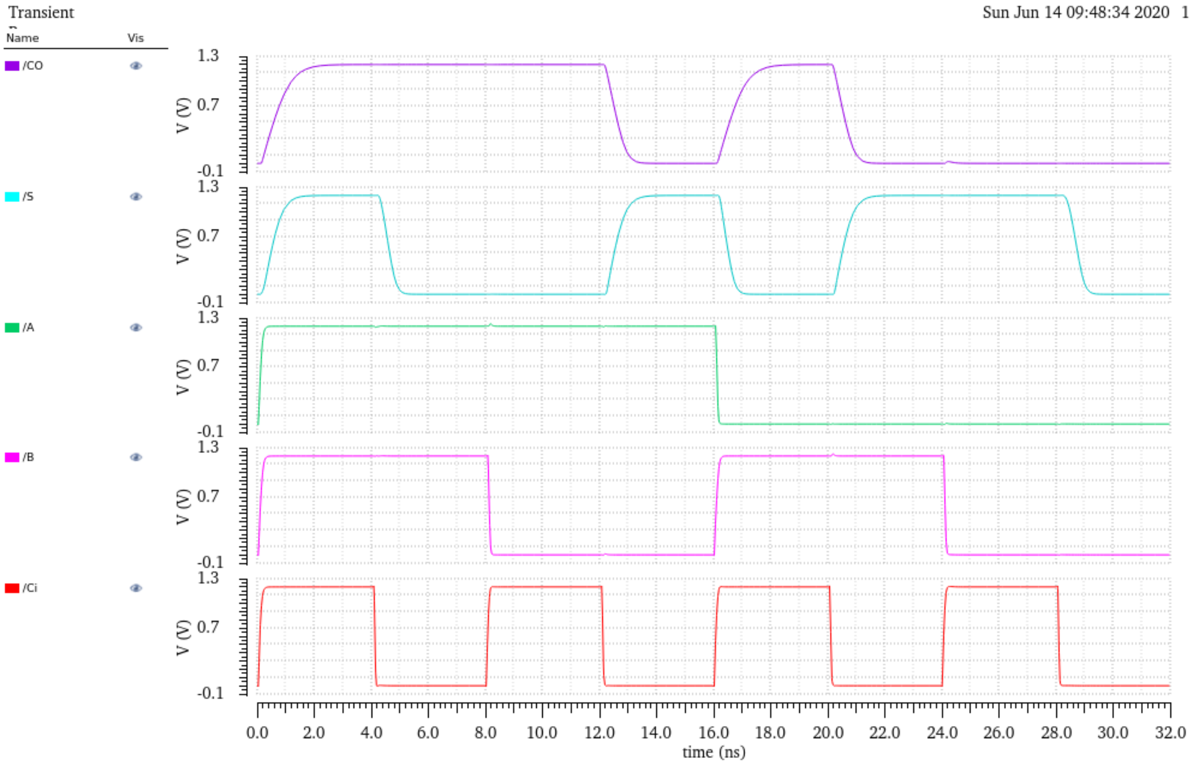


Figure 13: *Full Adder Post Layout Simulation.*

The inclusion of extra parasitic capacitances and resistances in the device lead to larger delays and rise/fall times. The extra capacitances require more energy to bring the voltage up to logic levels, therefore a larger time at the same current. The extra resistances limit the amount of current able to be supplied to the associated capacitances, increasing the time to raise the voltage to line levels. This relationship can be seen in Eq.2. The inclusion of the necessary routing and connecting wire will at both resistance and capacitance to the circuit, so to minimising this will result in better performance. This was done as outlines in Section V.i.

$$t_p = ln(2)CR \tag{2}$$

	Delay (ps)	Rise (ps)	Fall (ps)
Sum	615(24% ↑)	520(3% ↑)	387(7% ↑)
Carry Out	586(6% ↑)	790(3% ↑)	499(3% ↑)

Table 5: *Worst Observed Delay, Rise and Fall Times, Post Layout. With Percentage Increase from Pre-Layout Simulations.*

Table.5 shows the delay, rise and fall times of the outputs the same as section IV.i. As can be seen the biggest increase was seen in the delay of the Sum output. This delay represents a large amount of parasitics introduced in the path of S, leading to a 24% increase in the delay. One reason that this could be is the large metal two wire that takes S from the inverter on the far left to the bottom of the device to be used as an output. This wire run directly under the large metal 6 GND plane and will likely introduce a large capacitance between the two. This was a calculated trade off as having the output here will allow tighter packaging of the device when cascaded into the 8-Bit adder. The other values seen in Table.5 represent between $3\% - 7\%$ increase. This is an acceptable level that is to be expected when the connecting wire and routing have to be added.

iv. 8-Bit Adder Layout

The main goal with the 8-bit layout is to stack 8 of the full adders as close together in a rectangular shape with as little gaps in between, preferably a large portion of overlap between the units. As well as minimising the amount of additional wires needed to route signals and power, these would increase the parasitics associated with the final design. Due to planning put into the one bit adder there were no additional routing required. Cin lines up directly with Cout and the VDD and GND wires line up requiring no extra connections. The signal A, B, and S are all available from the top and bottom of the device as to not be interfered by stacking another device next to the device.

The entire device can be seen in Fig.14, but doe to the aspect ratio and size of the device the intricacy of the join between device is not clear. Fig.15 show a section of the device containing two devices and the transition between them. The limiting factor between the devices is the $20nm$ distance requirement between diffusion areas. As the limiting factor is the distance between the diffusions in transistors is means no space was wasted in combining the individual full adder circuits.

This produced a final silicon footprint of $4.01\mu m \times 48.56\mu m$ for a total area of $194.73\mu m^2$. This is equivalent to 7.57 times the area of the single full adder, showing a large overlap of 6.14% per device.

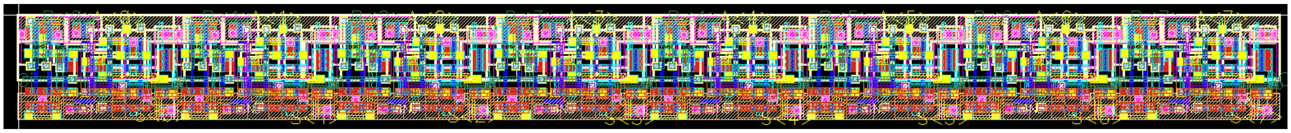


Figure 14: *8-Bit Ripple Carry Adder Full Layout*

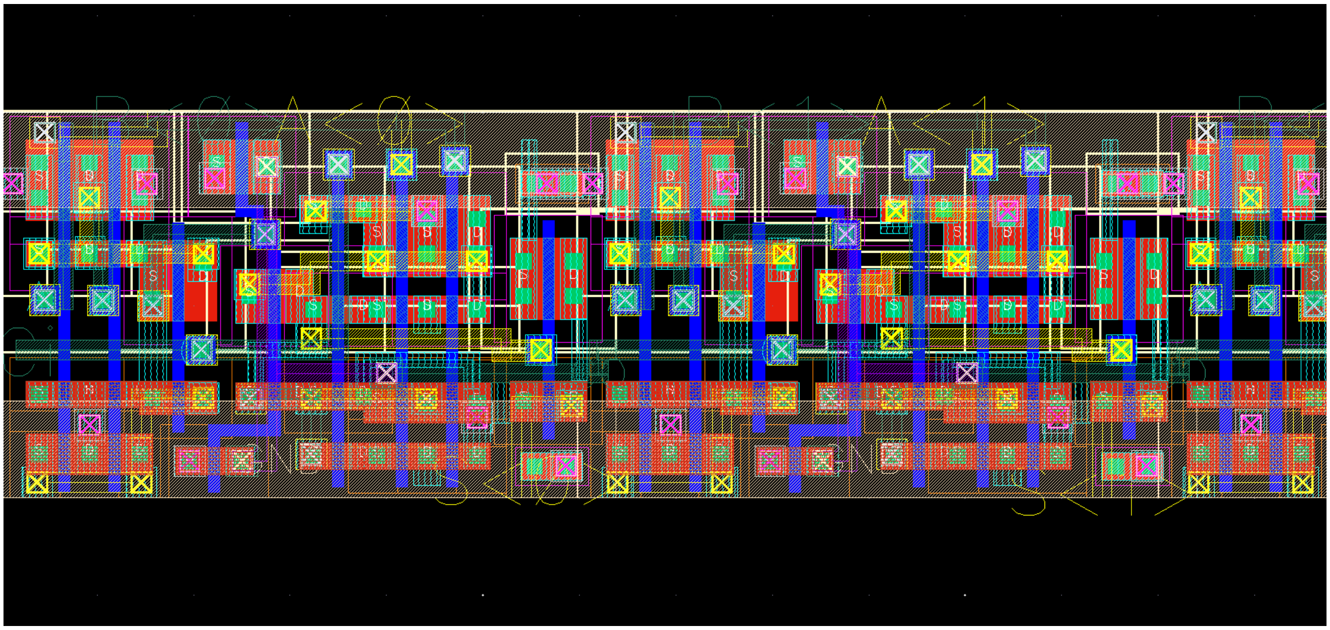


Figure 15: *Two Full Adder Blocks in the Final 8-Bit Layout.*

v. 8-Bit Adder Post Layout Simulation

Fig.16 shows the functionality of the 8-bit adders as outlined in Section.IV.ii. It shows the complete function of the 8-bit adder as specified in Table.2. The functionality is the same as expected but the rise times and delay of all signals have been increased, as is expected with the inclusion of extra parasitic capacitance and resistances that are included from the layout.

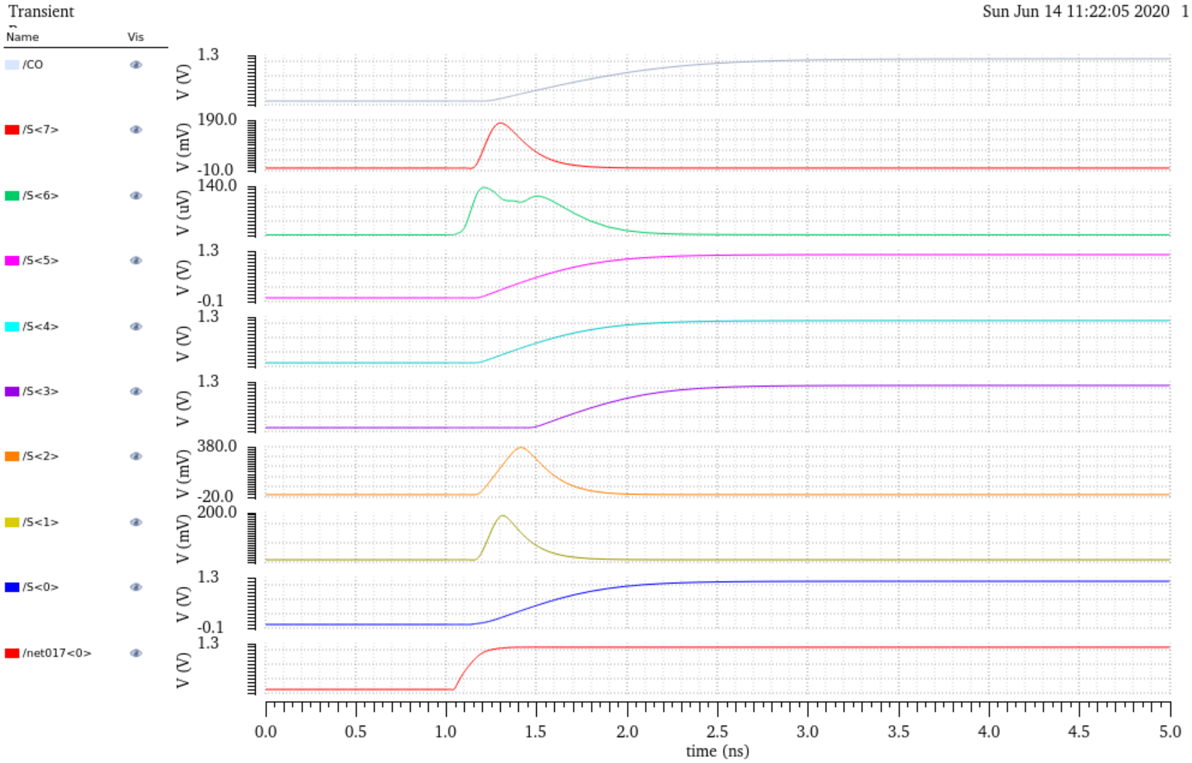


Figure 16: *Post Layout 8-Bit Testing.*

The critical path is the key behaviour and limiting factor behind a design, as such the delay experienced by this critical path is the most important aspect of the design. Fig.17 shows the outputs of the critical path as dictated in Section IV.iii.

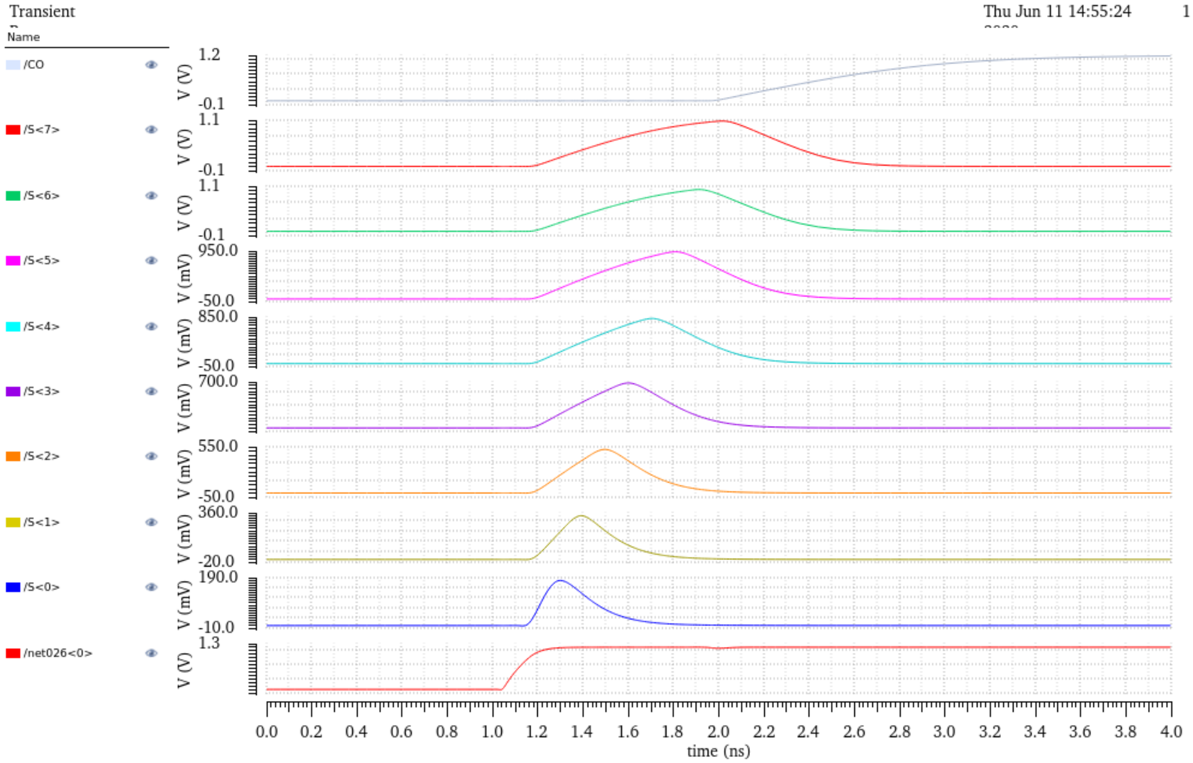


Figure 17: *Post Layout Critical Path Simulation.*

The change of performance from the post layout simulation can be seen in Table.6. This shows a significant 32% increase in delay on the final carry out bit, this is due to the parasitic capacitances and resistances associated with the routing on the design. Some other effects that can be seen in Fig.17 are the increase in peak voltage in S<7>, this output reaches 1.1V which can be taken as a logical high for a small time, potential introducing artefacts in the device that the output feeds into. This means that a ripple carry an larger the 8-bit will have a significant number of outputs at a logic high for a significant amount of time. This makes this implementation in its current sizing not ideal for a larger ripple carry adder, without careful effort to avoid using the outputs until this propagation has finished. The rise time of input signal net026<0> is also increased, this is to be expected as the signal inputs are routed underneath the power planes in an effort to reduce the silicon size of the device.

	Delay (ps)	Rise (ps)
Carry Out	1394(32% ↑)	772(2% ↑)

Table 6: *Worst Case Delay and Rise Times, Post Layout*

VI. Conclusion

An 8-bit ripple carry adder makes use of 8 full adder circuits with cascaded output to compute the sum of 2 8-bit numbers. As such a full adder can be used as a simple building block to combine into the larger and more useful 8-bit ripple carry adder. The report looks at a CMOS implementation of a full adder[1], its functionality, layout and finally using it as a building block for an 8-bit ripple carry adder.

The full adder used in this design makes uses of the CMOS topology shown in Fig.2 and using UMC 130nm technology a layout is proposed with a silicon area of $25.74\mu m^2$. This focus of this design was to minimise silicon area by having the maximum amount of shared diffusion and optimising the input and outputs to allow cascading of the full adder devices.

Some key features that allowed the small package and ease of cascading include:

- Large amount of shared diffusion, reducing connecting wiring and total amount of diffusion
- Splitting of the Cout inverter, drastically reduced length of device, made use of empty space and allowed sharing of the Z poly
- Positioning of input and output connections, allowing ease of cascading with multiple devices without sacrificing access to inputs and outputs.

The 8-Bit adder had a final area of $194.73\mu m^2$, the critical path of this devices had an increased delay of 32% due to the parasitic capacitances and resistances. This is an significant increase and to be expected.

There are a few small areas where the layout of the device can be improved, looking at the full adder there is empty space in the middle left, this leaves some potential to reduce the height of the device by moving some of the transistors in this area. There are a few areas of rouuting underneath the large power lines that would lead to a large parasitic capacitance associate with this, it may be possible to reduce this with changing the layout slightly.

The performance of this device can also be improved as stated in Section IV.iv. With potentially a decrease in delay of 18% this can solve the issue of the 8th bit reaching a high voltage level while waiting on the cascading carry bits. This performance comes at the expense of silicon space and the speed requirements would need to be known before it can be said if that is a worthwhile trade off.

References

[1] F. Boussaid, “Elec 5503 lab project description,” tech. rep., University of Western Australia, 2020.