Carry Select Adder

Purdue ECE 55900 MOS VLSI Design

1st Gustaf Swansen , 559mg104
gswansen@purdue.edu

I. Introduction

In this project a 16-bit wide 4-block Carry Select Adder, CSA, was created. The CSA, in comparison to a Ripple Carry Adder, RCA, shortens the critical path. This results in a faster circuit at the expense of complexity and more logic. This design is intended to drive a load capacitance of 2fF with an input rise/fall time of 50ps. The design must have a less than 2ns propagation delay and consume less energy than 850fJ while minimizing area.

II. Design

The following section will outline the design of the CSA. Optimizations were made primarily focused on optimizing propagation delay.

A. Topology

Complementary Metal Oxide Semiconductor, CMOS, topology was chosen for all gates in this design. CMOS was chosen because of its reliability and ease of implementation. Pass Through Logic, PTL, requires careful design to avoid leakage currents, and correct behavior of level restorers. Dynamic Cascode Voltage Switch Logic, DCVSL, has a reduction in logic gates. DCVSL would not benefit this design, because there were no points in the design where a signal and its inverse were both used.

B. Sizing Strategies

TABLE I
Critical Path Sizing

$NAND_0$ 1	**INV_0** 1.142	**$NAND_1$** 1.830	**$NAND_2$** 2.090	**$NAND_3$** 2.387
$NAND_4$ 2.726	**$NAND_5$** 3.113	**$NAND_6$** 3.555	**INV_1** 4.060	**MUX_0** 6.506
INV_2 5.213	**MUX_1** 8.354	**INV_3** 6.694	**MUX_2** 10.727	**INV_4** 8.595

Sizing started with first determining the ratio of mobility between NMOS and PMOS $\hat{\mu}$. $\hat{\mu}$ was found through parametric analysis to be 1.48. Using $\hat{\mu}$, each CMOS gate was then sized for equal rise and fall delays. Each gate's logical effort was also calculated using Python. 120nm was chosen the as scale S=1 for all gates. A reference inverter with width 120nm was simulated to determine its input capacitance, 0.145fF. Using the input capacitance and the load capacitance the overall fanout, F, was found to be 13.77.

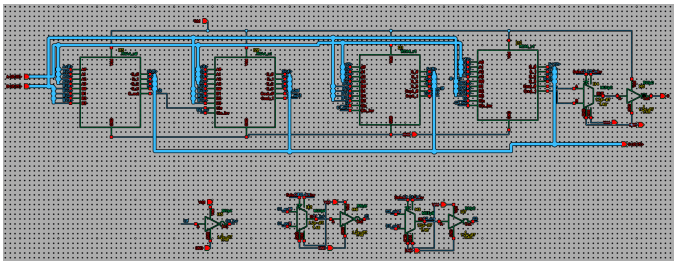

Fig. 1. This figure shows the schematic for the Carry Select Adder CSA

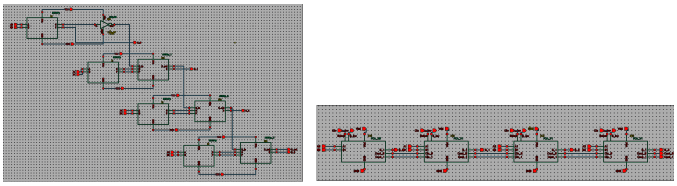

Fig. 2. This figure shows the schematic for the two versions of the Ripple Carry Adder. (A) RCA4_v0 (B) RCA4_v1

The schematic was then simulated to find the critical path. The critical path was determined to be the path from input B_0 to output C_{out}. This path was then coded into a Python script to determine the optimal scaling factor of each gate of the path. TABLE I contains the size of each gate in the critical path.

Using the scaled gates in TABLE I, the schematic was simulated again and the critical path remained to be $B_0 \rightarrow C_{out}$. As a result, no other paths were scaled since they would have no affect on the critical path. However, in the layout, the critical path is now the 11-15th bit of S.

C. Schematic

(Fig 1) Carry Select Adder. This schematic takes inputs A[15:0] and B[15:0] and outputs the Sum S[15:0] and Carry out C_{out}. This schematic of CSA utilizes scaled multiplexers and inverters to support optimal scaling. This schematic also uses two different version of the RCA to reduce unused logic.

(Fig 2) RCA4_v0 and RCA4_v1 : There are two versions of this block. Both versions function by taking in two 4-bit inputs

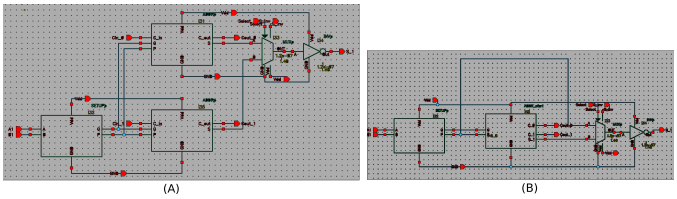

Fig. 3. This figure shows the schematic for two versions of the Full Carry Adder. (A) FCA_V1 (B) FCA_V0

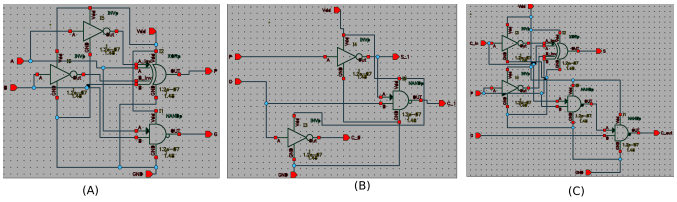

Fig. 4. This figure shows the schematic for 3 different blocks. (A) SETUPp (B) ADDR_start (C) ADDRp

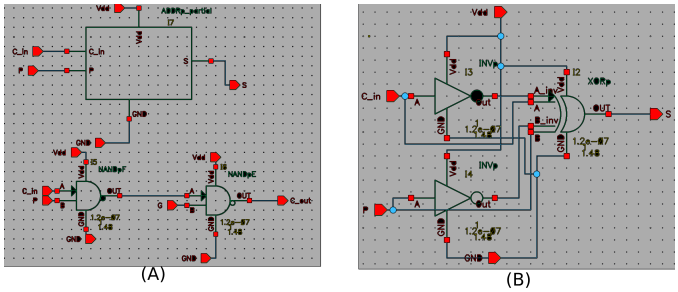

Fig. 5. This Figure shows the schematic for 2 different blocks. (A) ADDRp_# (B) ADDRp_partial

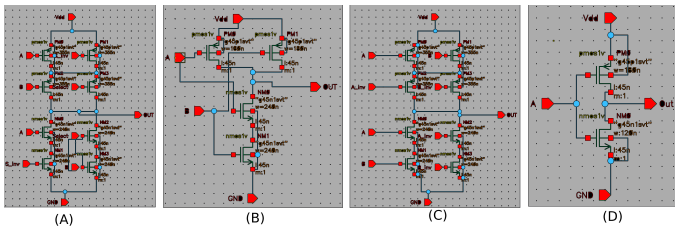

Fig. 6. This figure is of the schematic for the following gates (A) MUX2 (B) NAND2 (C) XOR2 (D) INV

A and B, and outputs the sum of them as S and Carry$_{out}$. Version 1 includes logic for when the carry bit input is 1 or 0. Version 0 reduces area usage by removing the logic for when the carry bit is 1. Version 0 also includes scaled versions of gates to support the optimal scaling factor for the critical path.

(Fig 3) FCA_v0 and FCA_v1 : There are two versions of this block. These blocks take A, B, C$_{in}$, and C$_{inv}$, and calculate a sum bit, and C$_{out}$ for when the C$_{in}$ is 0 or 1. Version 0 is chained together with M-1 Version 1s to form a M wide RCA. Version 1 includes logic for when the carry bit input is 1 or 0. Version 0 reduces area consolidating logic by using the P from SETUPp as the Sum bit for when C$_{in}$ is 0.

(Fig 4) SETUPp : This block outputs Propagate P and Generate G'. G' is output instead of G to reduce the number of gates in the system. The 2 AND gates and 1 OR gate that would generate C inside of ADDRp and ADDR_start can be replaced with 3 NAND gates through input-inversion.

(Fig 4) ADDRp and ADDR_start : These block outputs C$_o$$ut$ and S. ADDR_start reduce area by using consolidating logic. Instead of using an XOR gate, it generates the S bit for when the carry bit is 1 through an inverter.

(Fig 5) ADDRp_# : This block outputs is used to calculate the S and C$_{out}$ for the start of the CSA. There are 3 different scaled versions of this gate. ADDRp_1, ADDRp_2, ADDRp_3 are the names of the versions.

(Fig 5) ADDRp_partial : This block was created to support ADDRp_#. It includes all of the logic in ADDp apart from the two NAND gates. This allows for the NAND gates to be scaled without requiring to re-layout the rest of the logic in the ADDRp_#.

(Fig 6) MUX2 : Port "Select" is designated as the critical path pin because it incurs the least amount of internal capacitance. Euler's ordering was used to ensure the ordering of the PMOS and NMOS gates have no overlaps. There are 4 scaled versions of this gate. MUXp, MUXpA, MUXpB, MUXpC, are the names of the versions. All but MUXp implement multiple fins.

(Fig 6) NAND2 : Port "A" is designated as the critical path pin because it incurs the least amount of internal capacitance. There are 8 different scaled version of this gate. NANDp, NANDpA, NANDpB, NANDpC, NANDpD, NANDpE, NANDpF, NANDpG are the names of the versions.

(Fig 6) XOR2 : Port "A" is designated as the critical path pin because it incurs the least amount of internal capacitance. There is only one version of this gate since it does not appear in the critical path of the design.
(Fig 6) INV : There are 6 different scaled version of this gate. INVp, INVpA0, INVpA, INVpb, INVpC, INVpD are the names of the versions.

D. DRC/LVS Reports

E. Functionality

(Fig 8) This figure shows the waveform for the functionality test. At the bottom of the waveform, the signals have been converted from analog to digital for ease of verification. Note the waveforms are displaying in binary with the least significant bit, LSB, on the left and the most significant bit,

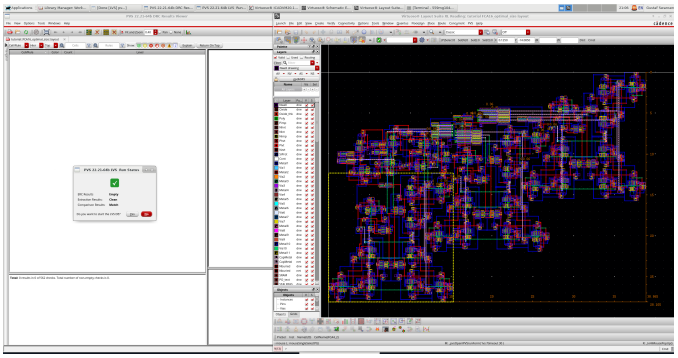

Fig. 7. This figure shows the clean LVS and DRC report for the layout of the CSA

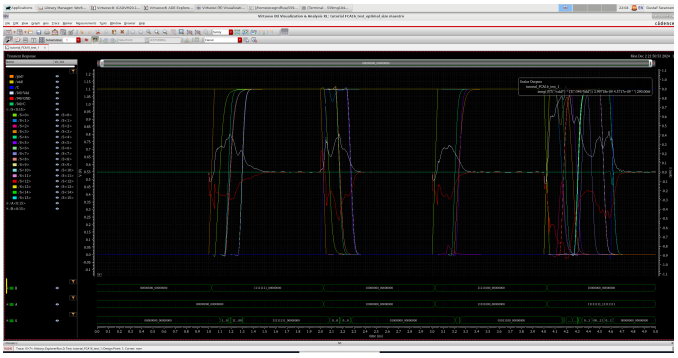

Fig. 8. This figure shows the waveform for the functionality test

MSB, on the right. Although non-intuitive to read, all of the operations outputs are correct. The test is for the following operations :
$0 + 0 = 00 + 255 = 255$
$1 + 1 = 2$
$27 + 31 = 58$
$16'hFFFF + 16'h0001 = 16'h0000, C = 1$

III. Energy-Latency-Area Analysis

Area is $1,092.3um^2$.

The worst path was tested by using the the following inputs: A = 16'hFFFF, B transition from 16'h0001 to 16'h0000. This was chosen because B is connected to the slower pin on the input NAND and this would create a transition from 0 to 1 for all carry bits. The resulting time was 0.4595ns.

The worst case energy was tested by using the following inputs: A transition from 16'hFFFF to 16'hAAAA, B transition from 16'hFFFF to 16'h5555. This would cause the most amount of logic transitions in the design, resulting in the most amount of energy consumption. The resulting energy was 336.2fJ.

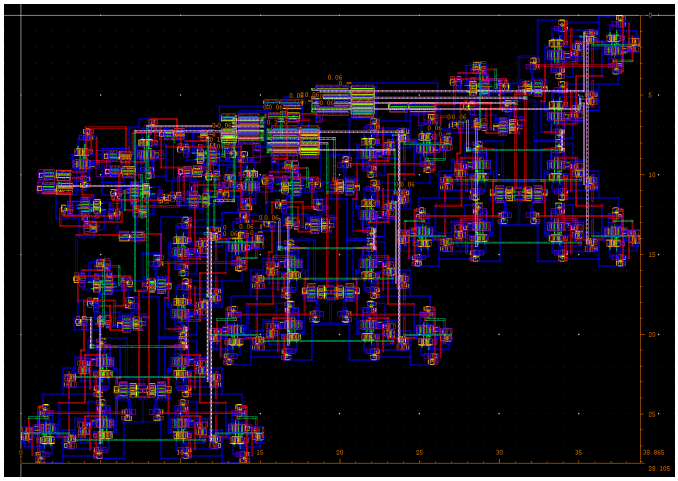

Fig. 9. This figure shows Carry Select Adder layout

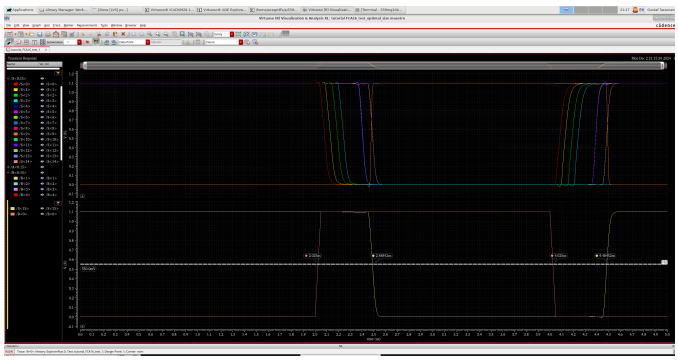

Fig. 10. This figure shows the measurement of the worst case timing waveform

IV. Bonus

A. Schematic

(Fig 12) is for a Square Root Carry Select Adder SRCSA. The SRCSA improves on the CSA by further reducing the critical path. This is done by modifying the block size from a linear 4. In the SRCSA starting at an M of 2, each subsequent RCA block will have 1 more full adder than the prior. This reduces the time that the carry bit leaves the first RCA block

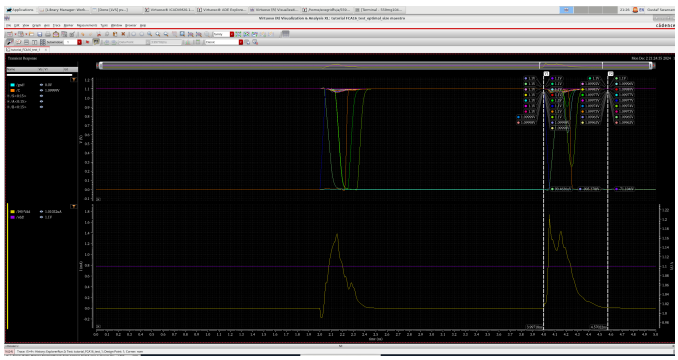

Fig. 11. This figure shows the measurement of the worst case power waveform

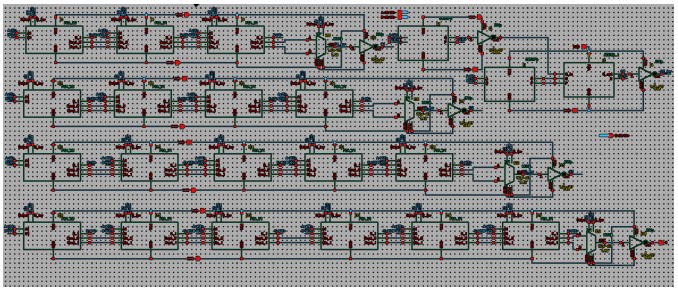

Fig. 12. This figure shows the Square Root Carry Select Adder schematic

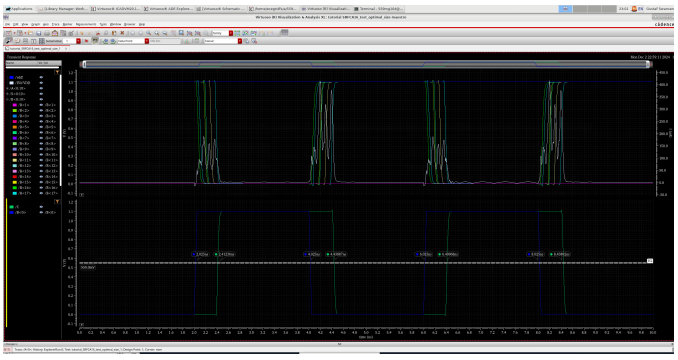

Fig. 13. This figure shows the Square Root Carry Select Adder worst case timing

and also makes the delay for each subsequent RCA block scale. Since the critical path is the carry bit, the RCA scaling should not affect performance.

B. Functionality

(Fig 13) This shows the worst case timing for the SRCSA. Note there was no scaling done on the gates used in this schematic and it is faster than the linear CSA. The resulting time was 0.41387ns.

(Fig 14) This figure shows the worst case power consumption. The power usage was 191.7fJ. I think there is a major difference between schematic simulation and layout simulation because the power the difference from the linear CSA and SRCSA was so large. I expected the SRCSA to consume more power since it has more logic (20b vs 16b).

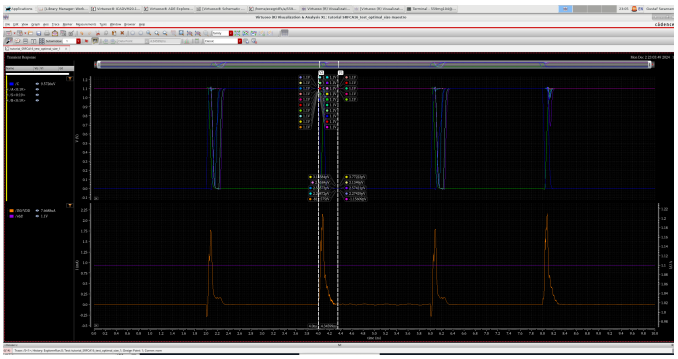

Fig. 14. This figure shows the Square Root Carry Select Adder worst case power

V. Conclusion

This design meets the requirements, but there is still room for improvements. Given more time, there are many aspects that I would like to redesign. The following subsections outline the aspects I would like to redesign.

A. Timing Redesign

In a redesign it would be best to not only scale the path for the C output, but also scale the output for the Sum bits 11-15. After simulation of the layout, the worst case path was no longer C, but Sum bits 11-15.

B. Topology Redesign

In a redesign it would be interesting to use PTL for MUX topology. This is because PTL for a MUX (including the level restored) would reduce the number of gates down from 8 to 4. This would likely reduce area.

C. Layout Redesign

The design meets the propagation delay and power consumption requirement by a large margin. The requirements specified minimizing area while meeting the delay and power requirements. There is an argument that the area should have been focused on more at the expense of these other two metrics. In a work environment, it would be best to get clarification on the design requirements and possibly redesign to reduce area.

Independent of requirements, one major issue of this layout is that the RCAs were not designed with the final layout in mind. This resulted in a diagonally skewed layout (see Fig. 1). This results in a large area of unused space. In a redesign, much of this area could be reclaimed by better sub-module layouts.

Another issue with the layout is that metal layers 3 and 4 do not always follow travel in one direction. This was due to rotating sub-modules when they were placed in higher level modules. This does not increase the area, but is bad practice and should be fixed in a redesign.