Design and Layout of High 9-bit Square Root Carry Select Adder

EE224: High-Speed CMOS Circuits with Dr. Hiu-Yung Wong

Johan Saltin & Abhishek Raol

Abstract

Herein we design a 9-bit square root carry select adder (SRCSA). The carry select adder is separated into 3 stages with the first stage containing 2bits, the second 3 bits, and the third 4 bits.

In a linear carry select adder, The mux of all bits are ready to generate a sum at the same time, and later bits must wait for the carry to propagate to generate the sum. In the square root carry select, the first stage computation is completed the earliest while the last stage computation takes the longest. However, the 1st stage can propagate the carry out while the later stages are finishing computation, so the carry-in for later stages arrives close to when the later stage mux is ready to generate the sum. The area requirement of the square root carry select is similar to a linear carry select, with significant improvements on delay.

Results

Total Delay (input->Cout) Schematic	**260.1 ps**
Total Delay QRC	**8ns**

Design Procedure

The first phase of the project was the design of a Full Adder as this had less delay and area than a single bit stage used in an SRCSA. However, it was obvious that although a sequential 9bit full adder, would save area, the delay would be very slow as each subsequent stage has to wait for the preceding to finish. We settled on the SRCSA.

We decided to use a Half Adder for the first bit because the first bit does not need to have a computation ready for two different carry-in values, and the Half Adder exhibits the least delay as well as least area.

All other bits are implemented with setup, carry propagate, mux and sum generation blocks. The starting bit schematic has no carry-out 0 for bit 0, instead of its already carried to bit 1 as G. The cause is due to PCi+G, where when Ci is 0, it is just G.

Using a Half Adder for the first bit saved us area, however there was an insignificant change in delay from using an HA for the first bit versus using the full 1-bit SRCS for the first bit, and this would make layout easier. We decided to make a last-minute change to use the same stage throughout to simplify the layout.

Design of Setup:

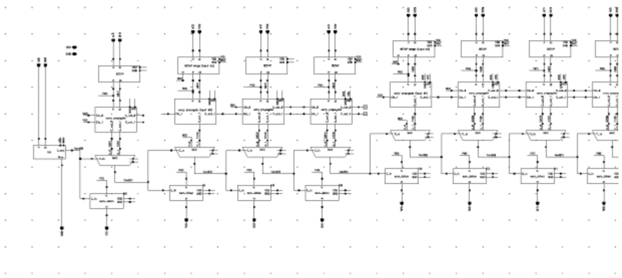
The setup is structured for A(+)B = P and A+B=G. Where G and P are carried to the carrying stages.

Design of MUX:

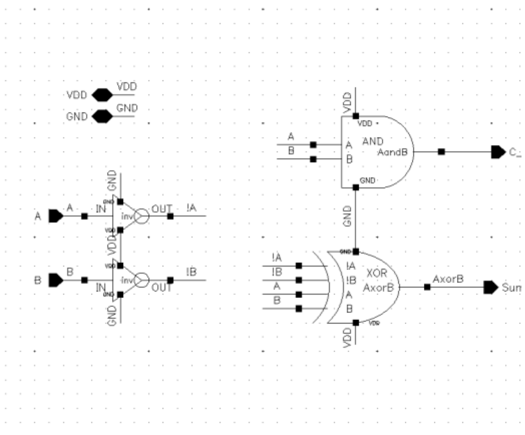
The Mux has an input of the C_out_1 and C_out_0 of the carry propagation step, for the current bit. C_in is used to select determining if 0 or 1. Thereafter C_out is carried to the next stage. The last mux determines what Carry in value will be given to the next bit block.

Components

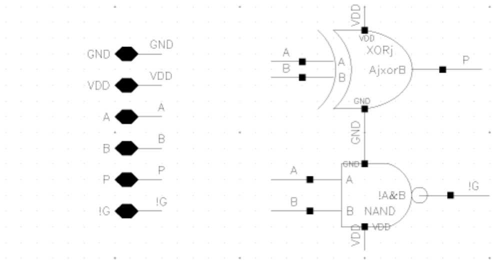
9-bit square root carry select adder schematic:



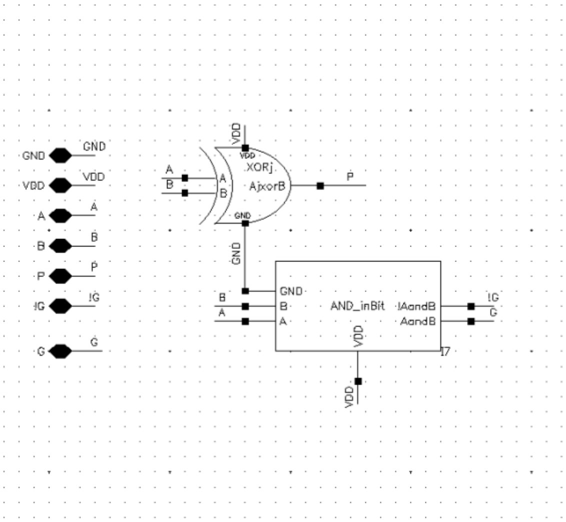
Half Adder:



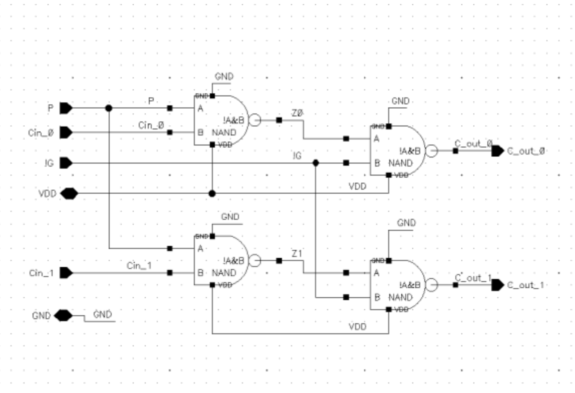
Setup:



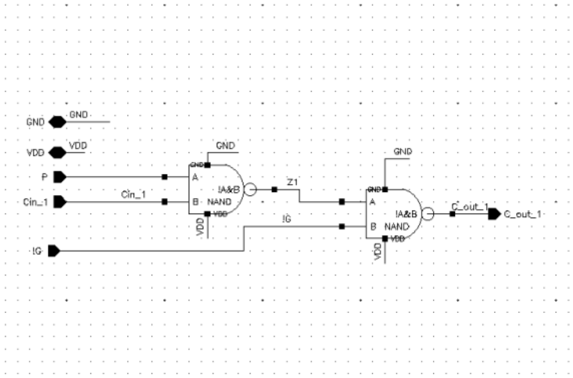
Setup input bit:



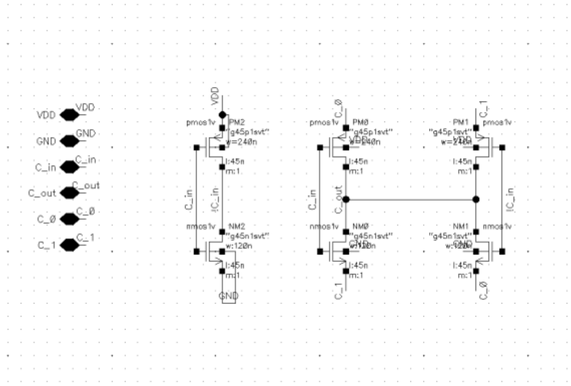
Carry Propagate:



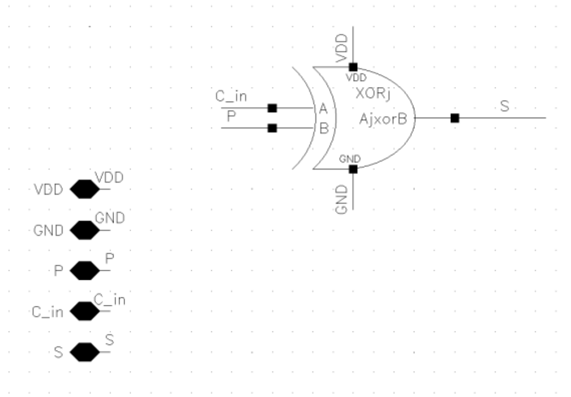
Carry Propagate Input Bit:



Mux:

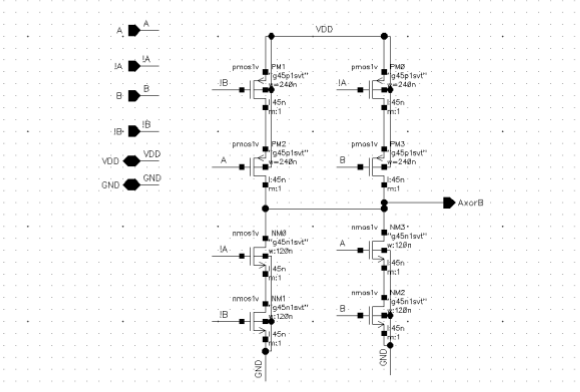


Sum:

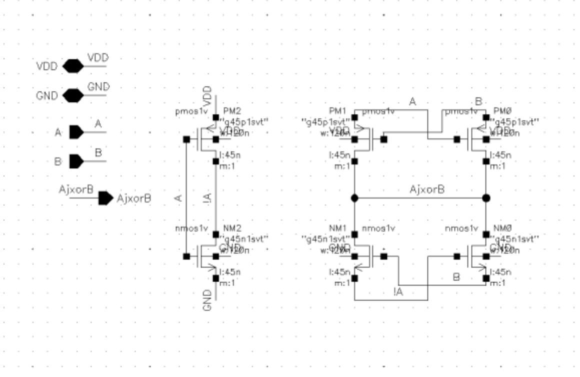


Logic Building Blocks:

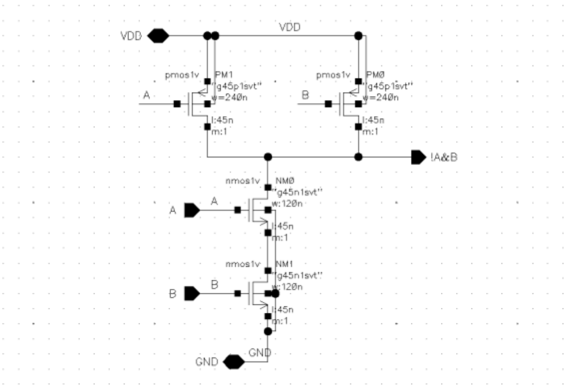
XOR (in Half Adder)



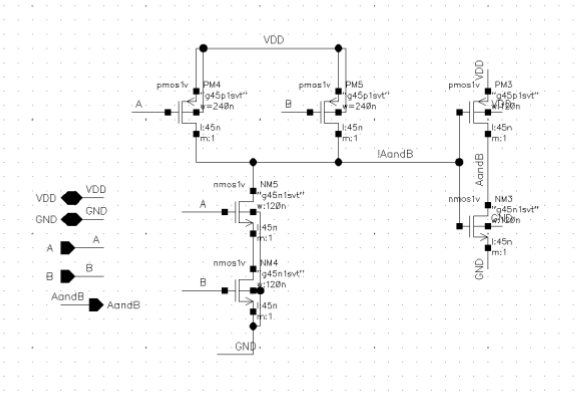
XOR (in Setup and Sum)



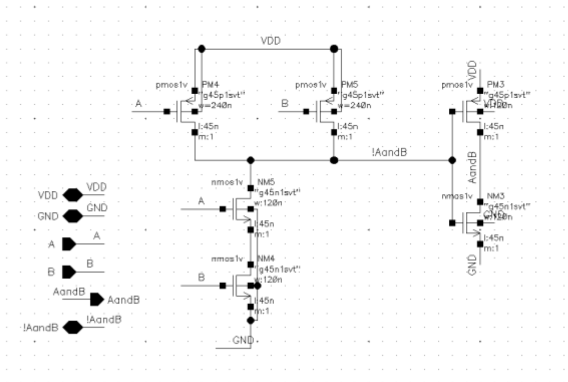
NAND:



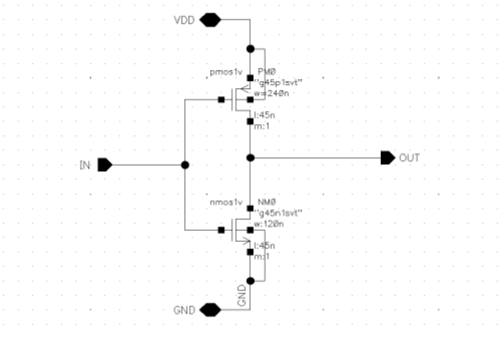
AND:



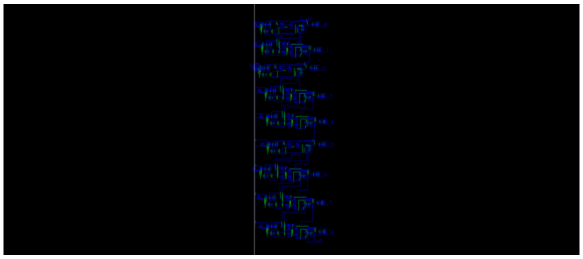
AND (in Setup input bit):



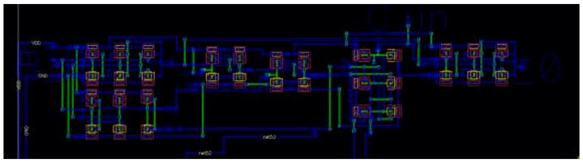
Inverter:



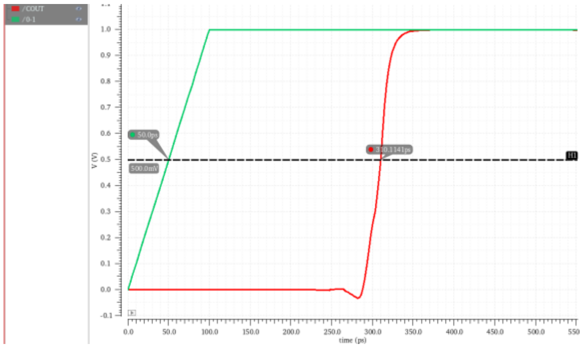
Layout:



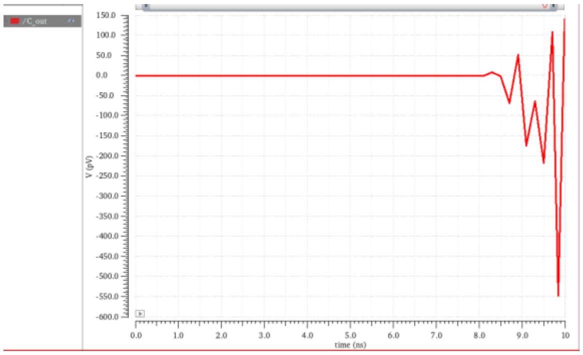
Layout, 1-bit:



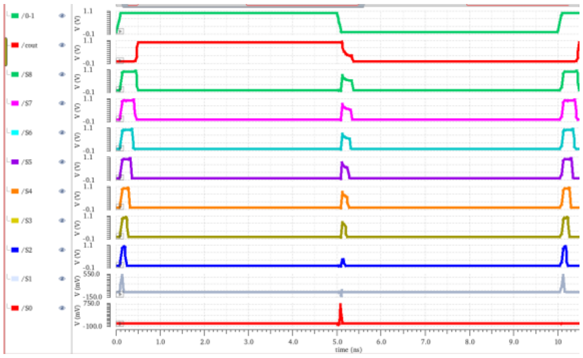
Simulation



Total Delay from ADE (input to Cout bit[8]): 260.1ps



QRC simulation (8ns)



Behavioral Simulation for Logic: (A,B)=(000000000,000000000) -> (111111111,000000001)

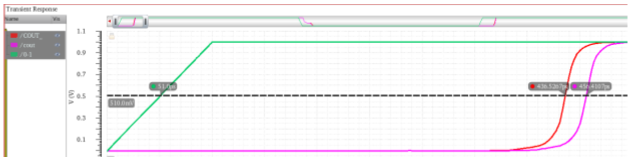
Contribution

Circuit Design (Schematic)	**Johan & Abhishek**
Logic Optimization (Schematic)	**Johan**
Area Optimization (Schematic/Layout)	**Abhishek**
Delay Optimization (Schematic/Layout)	**Johan & Abhishek**
Logic Blocks (Layout)	**Abhishek**
Routing & Interconnect (Layout)	**Johan**
Report	**Johan & Abhishek**

Improvement

We considered implementing a Full Adder (FA) for bit[1] instead of the SRCS 1 bit configuration because it would save on area. We thought that using the FA would save on area, but sacrifice some delay because the SRCS 1bit stage can start computing as soon as the input bits arrive, whereas the FA has to wait for the Half Adder (HA), to start the computation.

However, we simulated using the FA as bit[1] instead of the SRCS 1 bit stage and saw that the total circuit delay actually decreased 20ps.



This results in a better overall design because area and delay decrease with using the FA as bit[1]. However, rapid tape-out was necessary, so we used a more uniform design with only a HA as bit[0] and the SRCS configuration for all other bits.

When routing the logic blocks, we realized that we should have taken into consideration how we will connect our blocks while laying out the blocks so that the interconnects could have been shorter, and the critical path wouldn't run long and close to GND and VDD. This would have helped our delay. If we spent more time on layout and planned ahead, we could have formatted our layout more efficiently and reduced area and delay.

We also realized that we should have minimized the length of poly connections as a poly has more resistivity than metal1 and therefore contributes to more delay in our circuit. Furthermore, we only routed using metal1 and poly. It was too far into the layout until we realized we should have used other metals to route so that we route over other wires and save on area.

Conclusion

A 9-bit adder was created, using the root square carry select adder method. Overall the area, speed, and power were not fully optimized. However, the group designed schematics to optimize the structure, using full adder in the beginning. Due to time restrictions the full adder couldn't be optimized. But it was an idea to decrease transistor count and power. Overall, the project was rich in obstacles, which required patients and understanding of the technology computer-aided design software, cadence. For future projects, it would be better for us to start multiple weeks ahead and not work at multiple projects at once.