1 Introduction & Motivation

Adders are one of the most used cells in a digital system which is why their optimization is important. In this report, a CMOS 16-bit full adder will be designed and analyzed. First, the design will be chosen for the 1-bit full adder. Second, different architectures will be analyzed to find the fastest 16-bit static architecture. Lastly, simulations will be done to find worst case scenario for different corner simulations. The objective is to minimize adder delay to be below 400ps. Finally, the impact of power-supply voltage on power-delay product will be realized.

2 Adder Architecture Review

Many full adder static designs exist but the 1-bit mirror adder is prevalent due to its sizing with performance. It can be used as the basic building block for higher bit addition. One can daisy chain them into the ripple carry adder (RCA) but this would have a delay of O(n). Many options exist to improve this like carry-lookahead, carry-increment, carry-save, etc. These were simulated and the carry-select adder (CSLA) was found to be the fastest with carry-increment being second. This is because the uniform-sized CSLA divides the 16 bits into √(n) blocks where each block computes the addition using √(n) bits RCA. Each block computes the addition using both values of the carry bit which can be multiplexed at the end using the actual carry. The delay caused by this adder is O(√ (n)). The obvious disadvantage is the extra power consumption to compute both carry-in results with multiplexing and more than two times the area of other adders. If power and area are of concern, carry-select was found to be the worst.

3 Adder Architecture and Design

Figure 1 below is the 1-bit mirror full adder design and Figure 2 shows the block diagram for the 4-bit RCA that is built using the mirror adder. The 4-bit RCA will be used for the uniform-sized carry-select 16-bit adder shown in Figure 3. The last image (Figure 4) displays the test bench used for analysis in Section 4. It should be noted that that 2:1 multiplexors and buffers have been used from the gsclib045 library. The multiplexor chooses input A when Select is driven low and the buffer non-idealizes the signal.

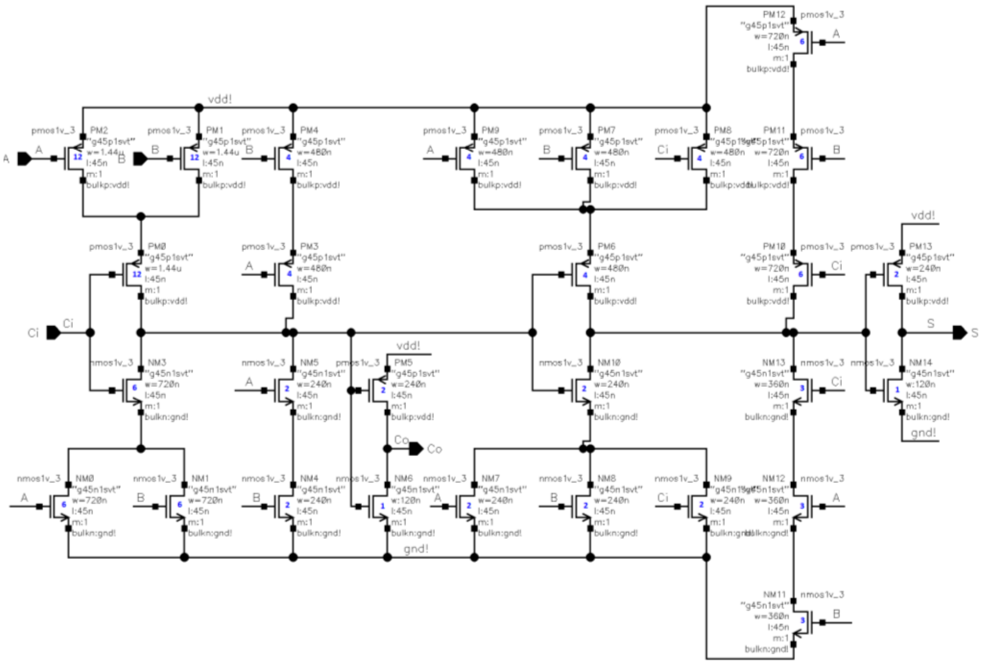

Figure 1. 1-Bit Mirror Full Adder

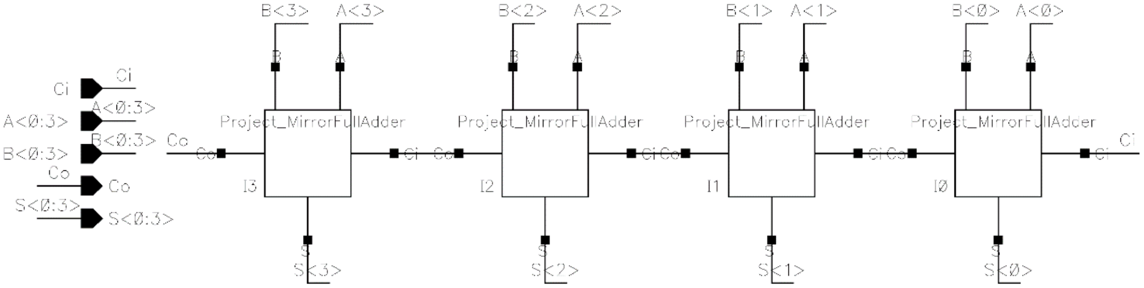

Figure 2. 4-Bit RCA

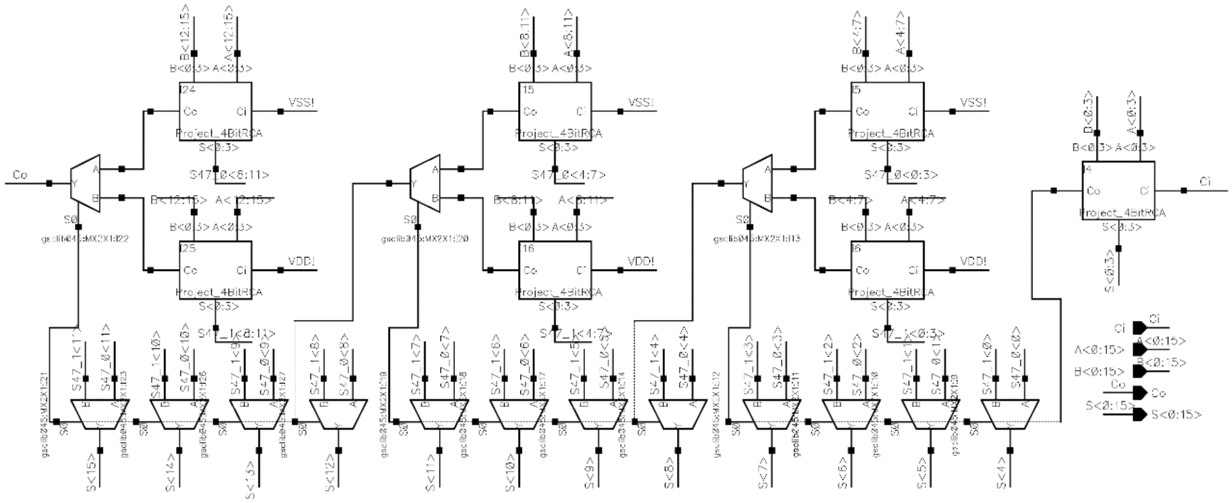

Figure 3. 16-Bit Carry-Select Adder

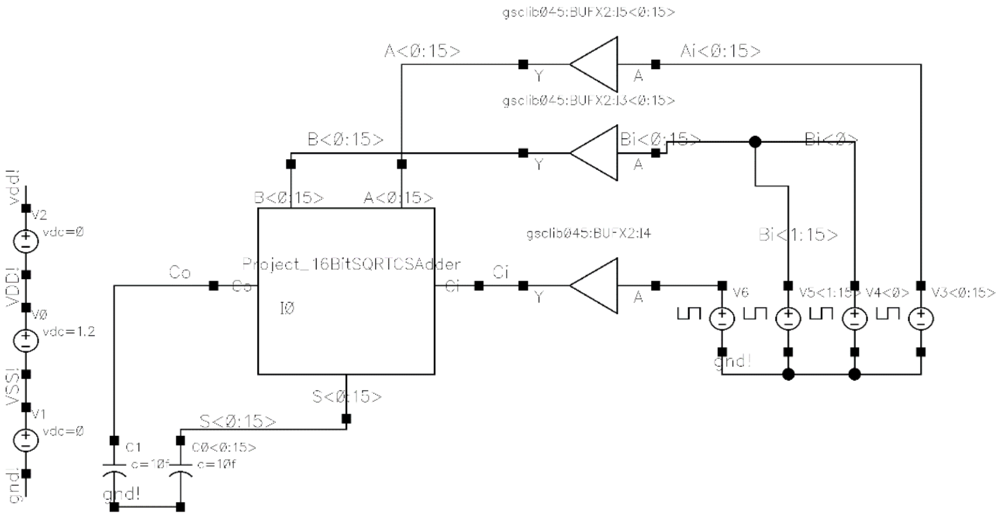

Figure 4. Vector A Testbench

4 Simulation Results and Analysis

The carry-select adder design was a success as seen in Figure 5 where three different vectors were simulated under TT 27°C corner. Vector A is 0xFFFF+0x0001, B is 0xFFFF+0xFFFF and C is 0xAAAA+0xAAAA. Then, Table 1 summarizes the delay and power consumption for these three vectors under three corners (TT 27°C, FF -25°C, SS 85°C) for a 1ns pulse width. The power consumption was measured for a 3ns duration for each source going from 0->Vector->0. It should be noted that the delay calculation would depend on the previous vector addition hence there should be correlation between addition instructions for best optimization. The worst-case delay was associated with vector A under SS 85°C as expected. It should be noted that the average delay is always less than 400ps even under the worst corner. Also, it was found that most power was consumed for vector B because of more blocks in the carry-select being used. Power consumption as at a maximum for faster corner simulations like FF -25°C but power delay product is at a minimum. Hence lower temperatures and FF mismatch are preferred for performance but not for power consumption.

For power delay product analysis with power supply variation, vector A was simulated under TT 27°C for 3ns. It was found that as VDD increases, the power delay product (PDP) decreases at lower voltages until velocity saturation and mobility degradation begins to occur at 1.2V. Then PDP continues to increase because of higher increase of power consumption compared to decrease of delay. This occurs because of many reasons such as more leakage occurs at higher voltages, more consumption by charging/discharging capacitors and smaller resistances due to higher electric fields. Using the same reasons, the delay should decrease as well until non-idealities begin to occur. When non-idealities such as velocity saturation and mobility degradation begin to occur, the delay is no longer decreasing as much with increasing VDD hence causing an increase in PDP.

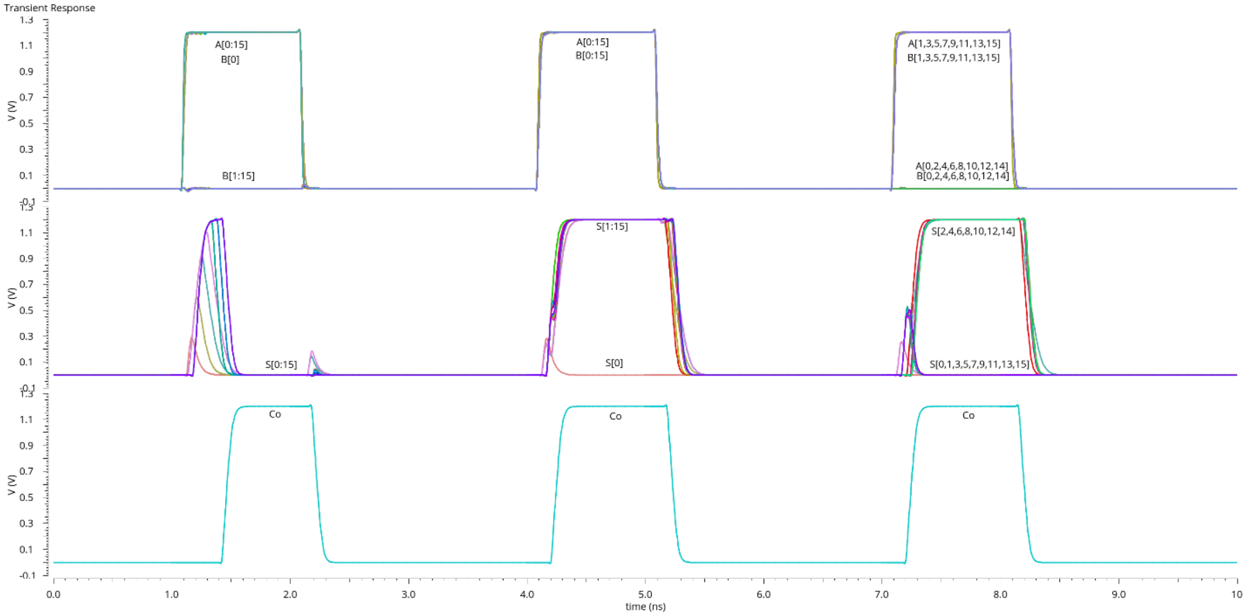

Figure 5. Functional Waveforms under TT 27°C

Table 1. Simulation Data

Test	A+B	S	Co	tpLH(ps)	tpHL(ps)	Average Delay (ps)	Average Power (uW)	Power Delay Product(J)
TT 27°C	0xFFFF+0x0001	0x0000	1	362.4	125.4	243.9	267.2	6.517E-14
TT 27°C	0xFFFF+0xFFFF	0xFFFE	1	146.1	128.6	137.4	402.3	5.526E-14
TT 27°C	0xAAAA+0xAAAA	0x5554	1	145.4	99.31	122.4	209.2	2.560E-14
FF -25°C	0xFFFF+0x0001	0x0000	1	256.6	88.7	172.6	278.7	4.811E-14
FF -25°C	0xFFFF+0xFFFF	0xFFFE	1	101.2	91.05	96.1	425.9	4.094E-14
FF -25°C	0xAAAA+0xAAAA	0x5554	1	101.2	69.63	85.4	216.9	1.853E-14
SS 85°C	0xFFFF+0x0001	0x0000	1	528.0	181.0	354.5	256.6	9.096E-14
SS 85°C	0xFFFF+0xFFFF	0xFFFE	1	218.6	185.3	202.0	380.9	7.692E-14
SS 85°C	0xAAAA+0xAAAA	0x5554	1	216.2	143.4	179.8	202.5	3.641E-14

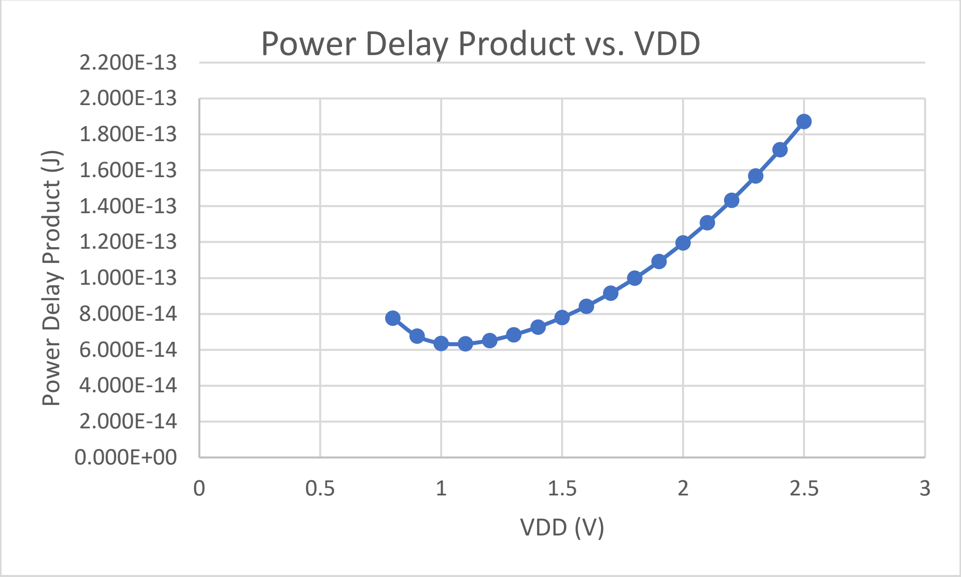

Figure 6. PDP vs. VDD Graph under TT 27°C

5 Conclusion

The design was a success, the CMOS carry-select 16-bit adder using mirror topology was fast and under 400ps for all corner simulations. The worst-case delay was for vector A under all corners and vector B consumed the most power. These delays were much better compared to other architectures like carry-lookahead. The disadvantage of carry-select is its power consumption, gate count and sizing. CMOS non-idealities were also verified by watching the power delay product under different supply voltages. At lower voltages, the delay is increasing faster compared to power consumption. At higher voltages, non-idealities such as velocity saturation and mobility degradation occur causing power consumption to increase faster compared to delay. All objectives were reached, and the tasks were completed. It can be improved by moving onto a dynamic design or switching libraries. If the fastest static architecture is required such as in military applications with disregard to power consumption, area, and gate count, then the carry-select is a great choice.