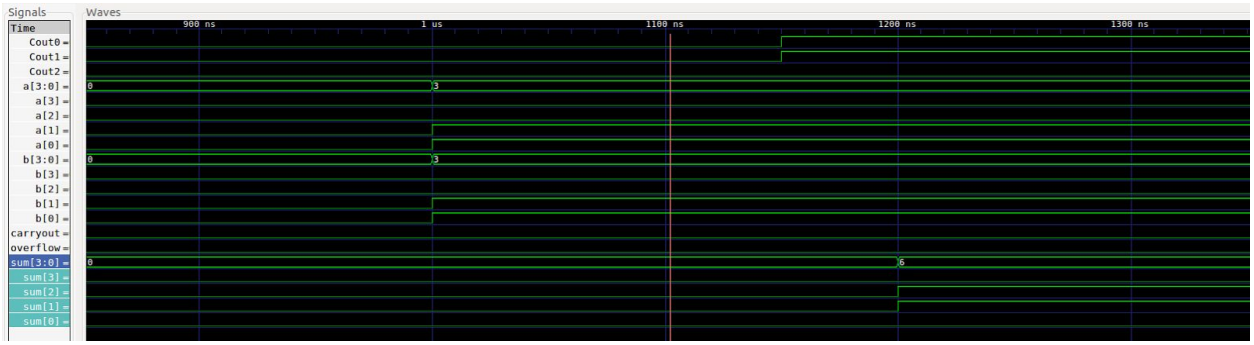
Jonah Spear, David Papp

Computer Architecture

Waveforms showing the full adder stabilizing after changing inputs. What is the worst case delay?

After testing several cases, it's evident from the graph that the delay is always 200ns. The graph above shows the addition of 0011 and 0011. We also tested the addition of 0011 + 0001 and 0000 + 1111 and obtained the same delay.



An explanation of your test case strategy. Why did you choose the tests you did?

Since we didn't want to write 256 test cases, we chose 16 tests that we thought would cover all the test cases. The goal was to test the sum, the carry out, and the overflow bit.

We wrote 12 test cases for testing non-overflow cases. We tested the addition of positive numbers, negative numbers, and one positive plus one negative number. We tested the addition of two opposite numbers to make sure their sum was 0. Everything worked as expected.

We also wrote 4 tests to test overflow. Two of these tests were for testing the addition of two negative numbers in which case the sum was below -8, and two were for adding positive numbers where the sum was greater than 7. For these cases, we only checked that the overflow bit was true and disregarded the nonsensical result for sum[3:0].

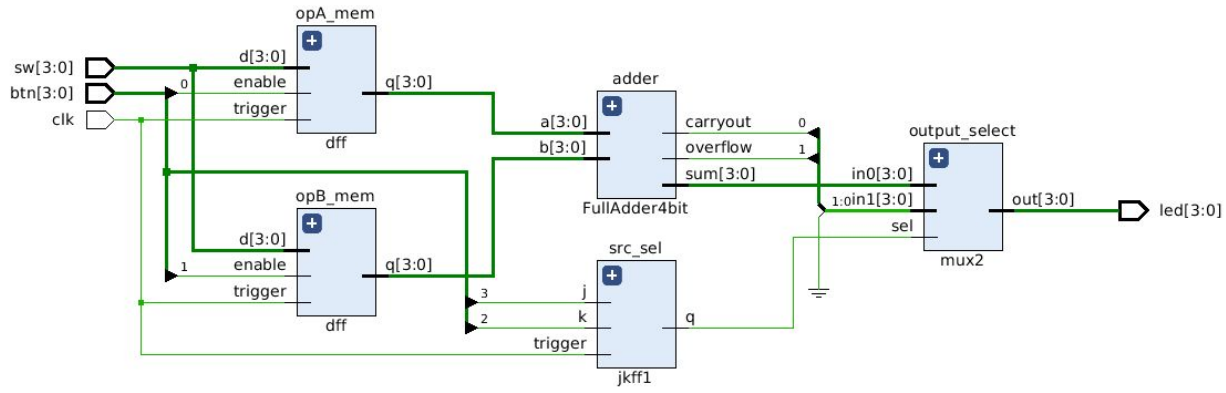
A list of test case failures and the changes to your design they inspired.
We had 3 failures on our test cases while creating our 4-bit adder:

1) Test Case:
a=0011;b=0000; #1000;
Expected result: Cout=0, sum = 0011
Result: a=1011 b=0000 | cout=0 sum=1011
It turns out we were referencing our inputs as a=0011 instead of 4'b0011.

2) Test case:

A B | Sum Cout Overflow
0000 0000 | 0000 0 1
Expected result: Sum: 0000, Overflow: 0
We were doing XNOR(sum[3], cout2) instead of XOR(carryout, cout2) to find the overflow.

A summary of testing performed on the FPGA board.



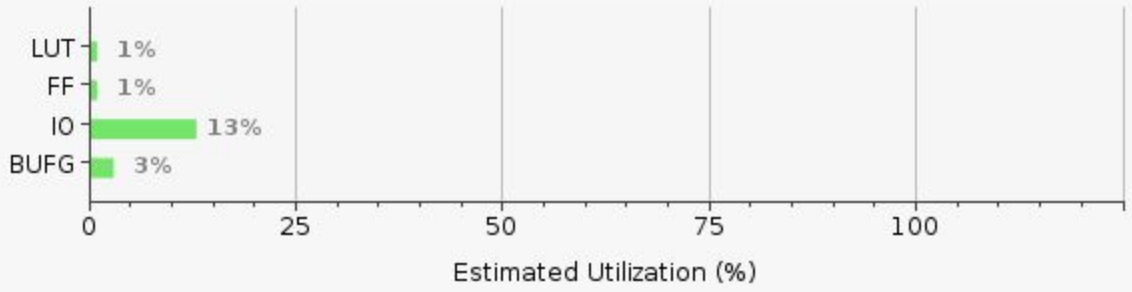
To test our FPGA board, we used the same 16 test cases as we did earlier in our test bench. We already knew the expected results, so this greatly streamlined the process. The FPGA board worked as expected, although it took us some time to get used to the process of interpreting the results. In the overflow cases, the bits lit up properly, as did the overflow bit. We attached a video of us testing the addition of 1101 and 1011, which is expected to output 1000 with carry out 1 and overflow 0 (fpga_testing).

```
A    B     | Sum  Cout Overflow
0000 0000  | 0000 0    0
0111 1111  | 0110 1    0
0011 0000  | 0011 0    0
0000 0011  | 0011 0    0
0101 0010  | 0111 0    0
0001 0011  | 0100 0    0
0010 1010  | 1100 0    0
1101 1011  | 1000 1    0
1101 0110  | 0011 1    0
1110 0010  | 0000 1    0
0001 0110  | 0111 0    0
1000 0010  | 1010 0    0
1000 1001  | 0001 1    1
1011 1100  | 0111 1    1
0101 0100  | 1001 0    1
0010 0111  | 1001 0    1
```

Summary statistics of your synthesized design from Vivado (Propagation Delay, Resources Used, etc)

Utilization **Post-Synthesis** | Post-Implementation

Graph | Table



Power **Summary** | On-Chip

Total On-Chip Power:	**7.326 W (Junction temp exceeded!)**
Junction Temperature:	**109.5 °C**
Thermal Margin:	-24.5 °C (-1.8 W)
Effective ϑJA:	11.5 °C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Implemented Power Report