James Johnston
Week 3 Module 3

Design Analysis:

1) List and explain the types of logic gates used in adder 16 and its sub-modules

16 Not gate- changes the bit to its opposite
16 mux 2-1 - selects between 2 inputs to connect to one output
-a control signal of 1 bit is used for selection
inside the mux there is a
Not gate - changes the bit the to its opposite
2 And gate - that performs the bit wise & operator on two bits
Or gate - that performs the bit wise or operator on two bits
16 Adder gate - bit wise adds the binary numbers and carries out Cout
inside the Adder
2 XOR gates - takes in two bits and performs the xor operation on them
2 And gates - performs the bit wise & operator on two bits
OR gate - performs the bitwise OR operator on two bits

2) You have been tasked to build a 64-bit adder/subtractor. Could you use adder 16 as a sub-module to build a 64-bit adder/subtractor? If yes, how? if not, why?

Yes, you could build a 64 bit adder by using the 16-bit adder 4 times to create a system that represents 64 bits. This can be done by wire where the count is carried over to the next adder module and then the bits are then carried.

3) What is the maximum frequency (fmax) at which slow adder will run reliably? Does it run at the same speed in all temperatures (0C and 85C)? If not, why?

The maximum frequency at which slow_adder will run reliably at 112.55MHz. It does not run at the same frequency in all temperatures. It runs faster at 0 degrees Celsius at 124.42MHz. This is because the silicon and resistors work better at a lower temperature and dissipate the heat created by the voltages.

4) What is “out reg” and its purpose? Give detailed description of this component.

The "out reg" is the register at the end of the logic gates and is the register just before the output. Its purpose is to make sure that the delays in the logic gates at the end sync with the clock. It makes sure that the input time is the same, at the D filp-flop.

Functional Analysis:

1) How does adder 16 perform subtraction?

It does subtraction by using 2 compliment for addition using negative numbers. The subtractor simply converts the subtracting number to negative by inverting it and adding 1 and then performs the addition.

2) List, explain and write function specification for each sub-module within adder 16.

The function of the multiplexer is to take in the user input S and if it is addition or subtraction change the control bit to 1 or 0 respectively thus allowing the 2 compliment function to be recognized in the addition processes.

The function of the adder is to add all three of the bits together and create a Cout if needed. In subtraction it utilizes the bits to preform 2- compliment and add the numbers together.

3) What is the use of “Cout”, an output of adder 16, and does it confirm to functional specification? If not, why? and how could you fix it?

Cout is the carry out of the end of the calculation that is bigger than 2 to the 16 in decimal numbers(65536) . So the number that is bigger than 16bits is being cut off and it does not meet functional specification. When you do subtraction the LED carry out turns on as does the LED carry out turns on when the answer is greater than 65536(decimal).

In order to fix this we need to add a "not" to the input s and "add" it to the Cout. This will create the output LED for the system.

The resulting table will be:

S	Cout	LED	
0	1	1	+
0	0	0	+
1	1	0	-
1	0	0	-

This will insure that LED will fire under the addition input where the Cout is 1.

Use fast adder project for the activities 3 & 4.

Timing Analysis:

1) By design, how many logic levels are there in adder16?

By design this board utilizes binary logic, where in there are two logic levels, low and high. These logic levels correspond to the binary system 0 and 1 respectively. When the voltage is high the binary is 1 when it is low the binary is 0.

2) How fast (MHz) is the clock running in fast adder?

The system clock is running at 50MHz. The output clock ratio is 6:1 and therefore the clock is running at 300MHz for fast_adder.

3) Does fast adder meet fmax requirements? If not, why?

No the fast adder doesn't meet fmax requirements. The expected 300MHz speed is actually 180MHz in a 6 to 1 ratio. In a 3 to 1 ratio expected 150MHz is actually 170MHz. My theory is that our DE0 Nano is a 32 bit processor and when we have a 6 to 1 ratio we are splitting the

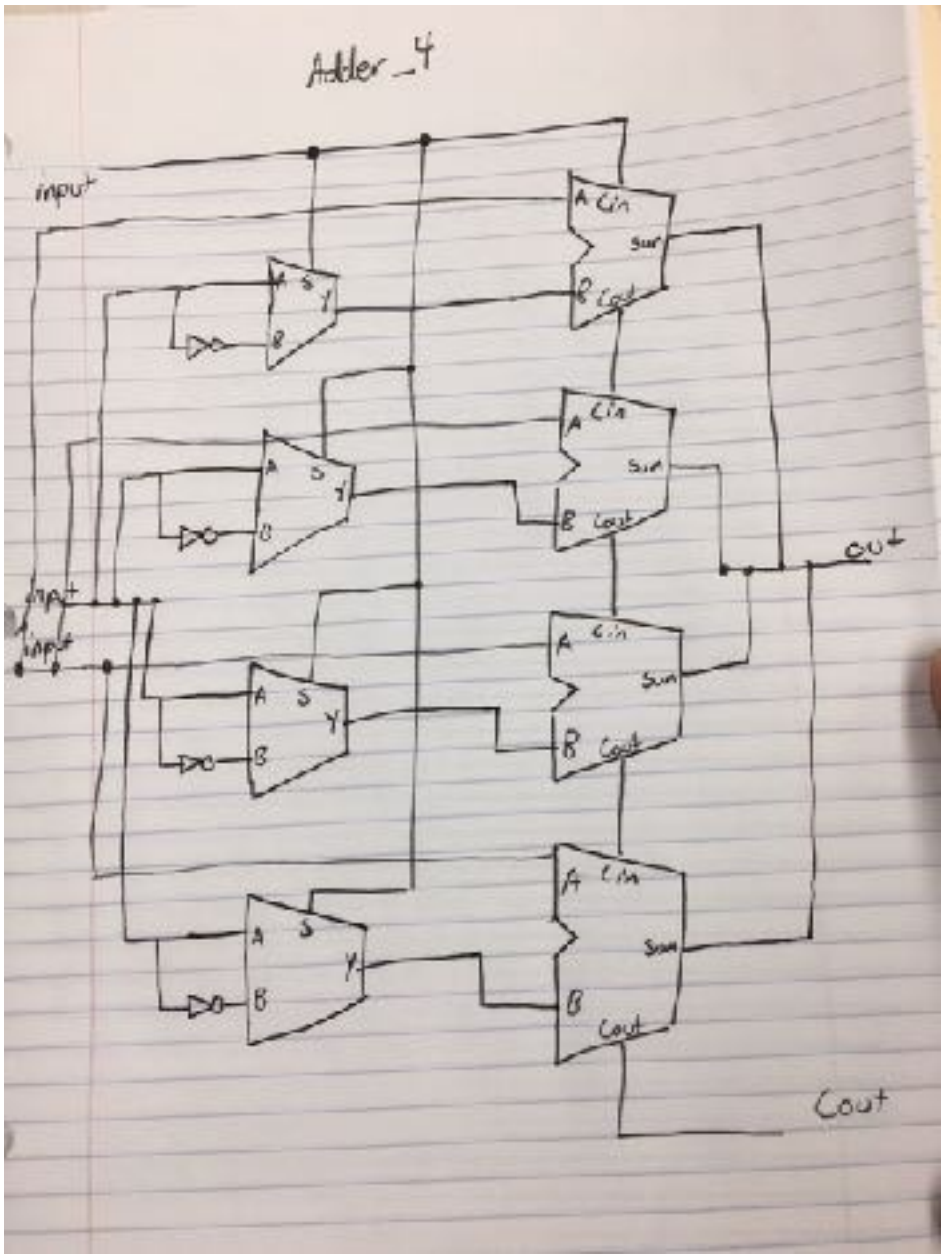
processing and thus it can only process 5 bits at a time and thus is not able to meet fmax requirements with the number of delays in adder_16.

Redesign:

1) Redesign fast adder, in paper, to meet timing requirement.

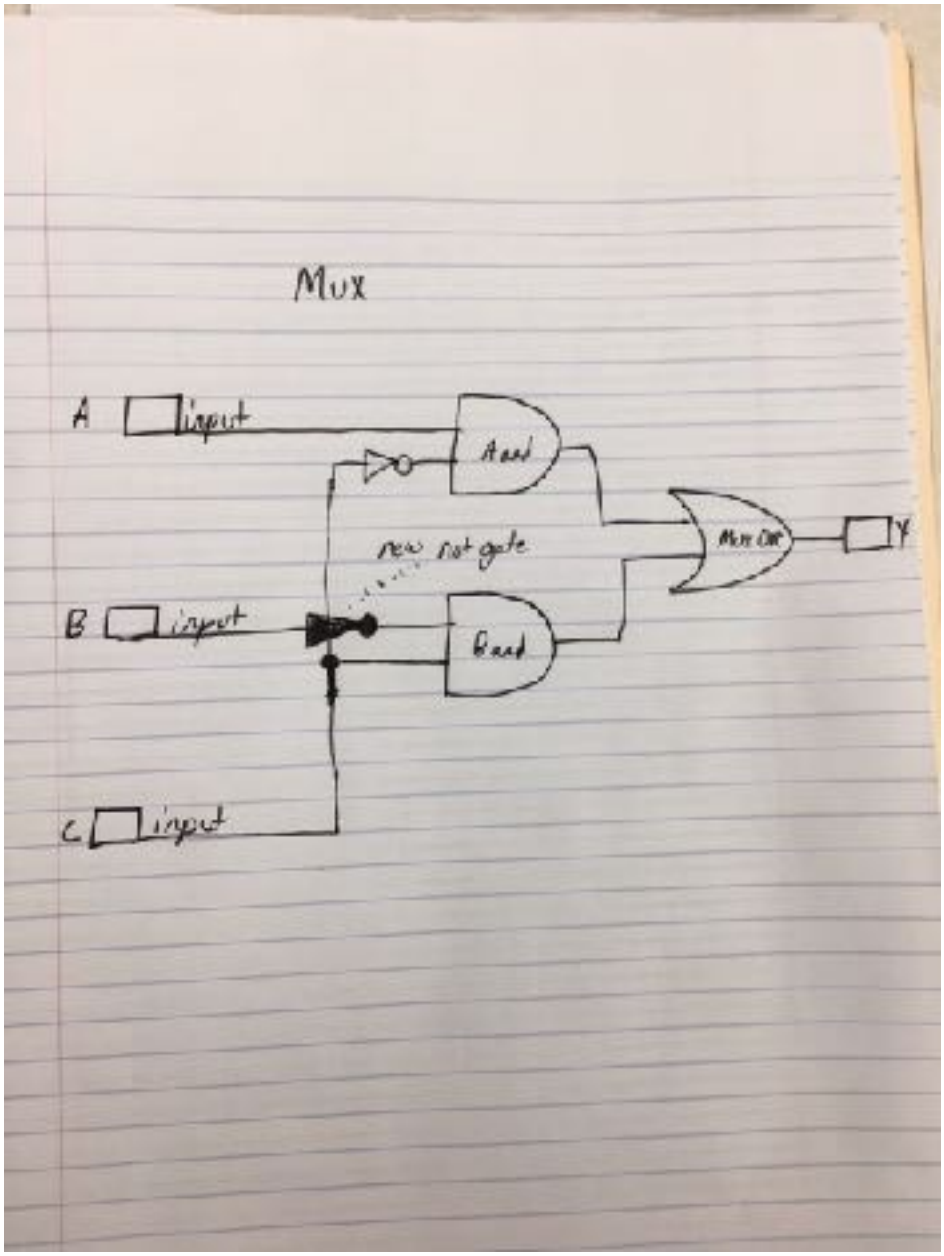
We discovered that the processor for the Cyclone IV is 32 bits. With the clock cycle multiplied by 6 then we have to figure out how many bit adders can be implemented to stay in the 32 bit processing area. So 6 x ? = 32 is our equation. In our adder we also allowed one bit for Cout and thus we were left with a 4 bit adder to complete our process at 6 to 1 clock ratio.

It looked something like this:

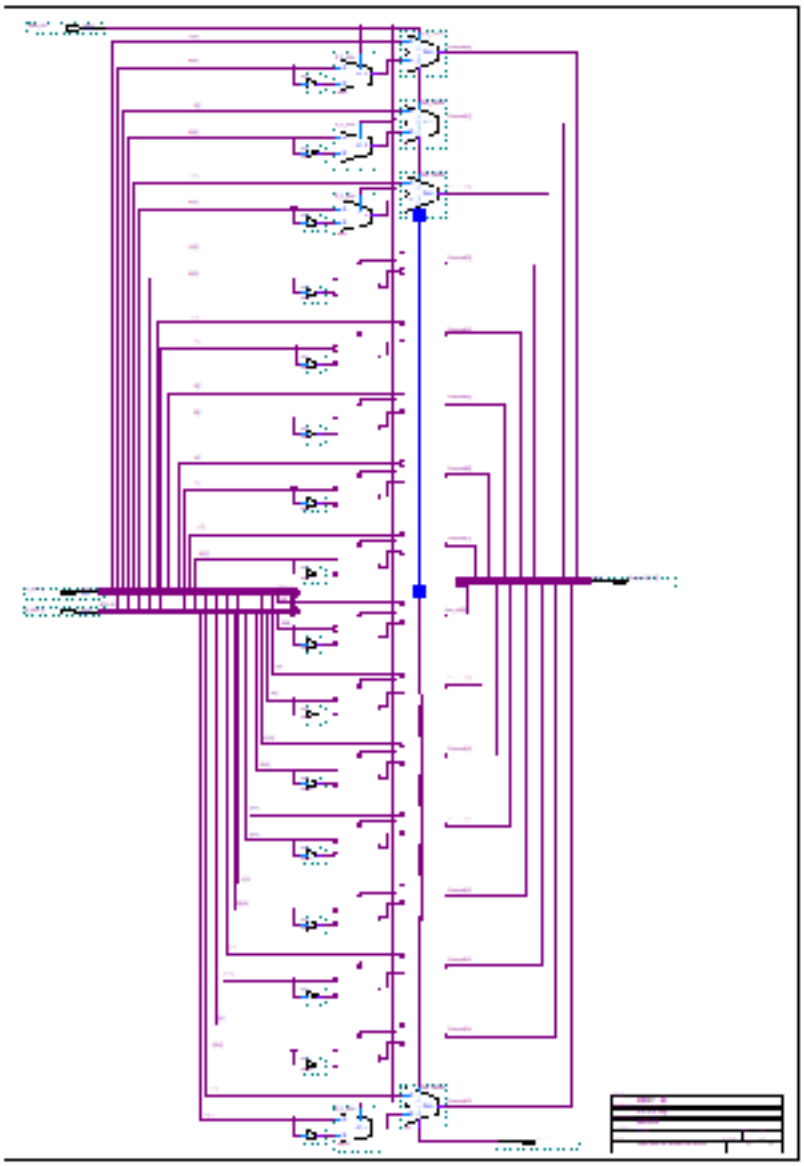


Another approach was to reduced the delays between gates in the system. Our timing analysis is a process of analyzing delays in the logic circuit to determine the conditions under which the circuit operates reliably. The condition we were most died on was the maximum clock frequency (fmax) for which the circuit will produce a correct output.

We reduced the delay by moving the initial not gate into the mux and thus eliminating a delay. it looked something like this:



2) Implement your modification using Quartus II software and
test it in DE0-Nano. You should make sure that your new design meets timing requirement before downloading it to DE0-Nano.



We discovered that reducing the delay was not enough to make the system reliable so we also reduced the adding nodes to allow for Fmax.