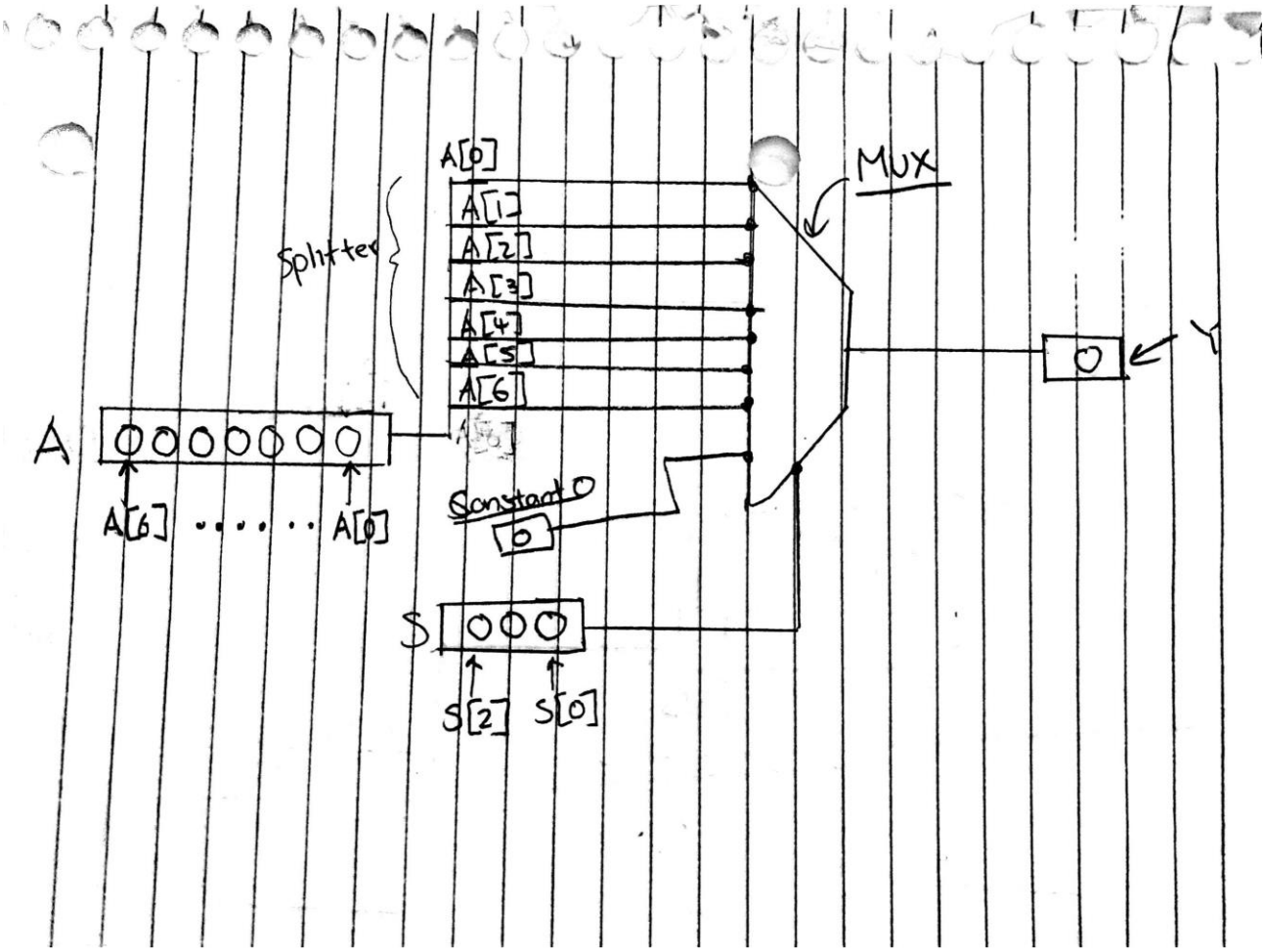
Tony Hong
1005109482
Oct. 7, 2020

CSC258 Prelab (Lab 3)

Part 1: Splitter + 7-to-1 MUX

1. Below is a schematic on how I designed the circuit that splits a multi-bit input to provide inputs to a 7to1 MUX.



b. The total size of the multi-bit input that would provide all the inputs to the 7-to-1 multiplexer would theoretically be 10 bits. 7 of the bits would feed the 7 non-select inputs of the multiplexer while the other 3 bits would feed the select input bits. In Logisim though, since a 3 select multiplexer can map 8 different inputs (2^3), therefore we would need another constant 0 input bit to fill in the last MUX input making the total number of input bits required 11.

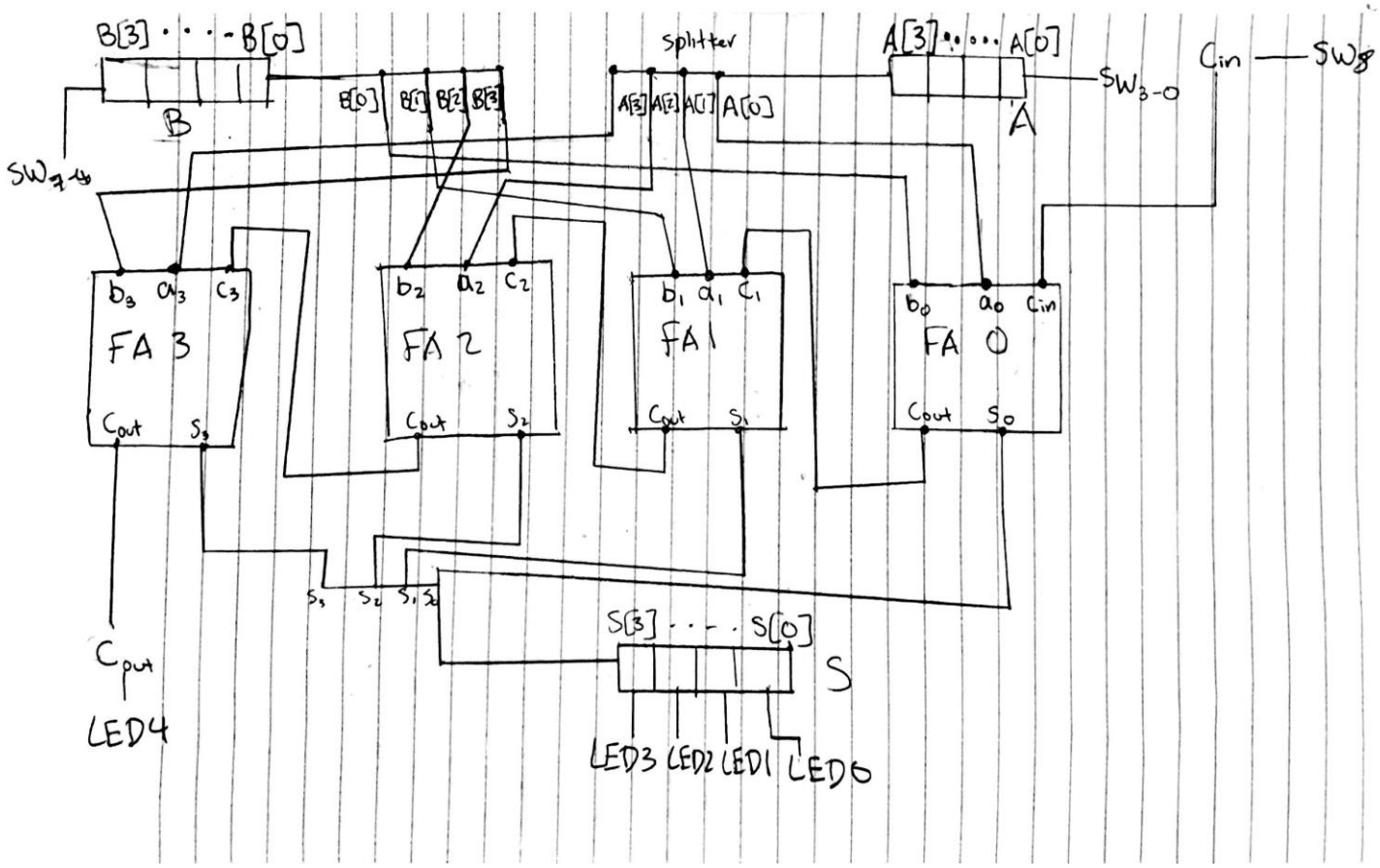
3. Below is a screenshot of a simulation result from my test vectors for this circuit

Passed: 15 Failed: 0

status	A	S	Y
pass	000 0000	000	0
pass	000 0001	000	1
pass	111 1110	000	0
pass	000 0010	001	1
pass	111 1101	001	0
pass	000 0100	010	1
pass	111 1011	010	0
pass	000 1000	011	1
pass	111 0111	011	0
pass	001 0000	100	1
pass	110 1111	100	0
pass	010 0000	101	1
pass	101 1111	101	0
pass	100 0000	110	1
pass	011 1111	110	0

Part 2: Ripple-Carry Adder

1. Below is a schematic of my ripple carry adder for two 4-bit inputs.



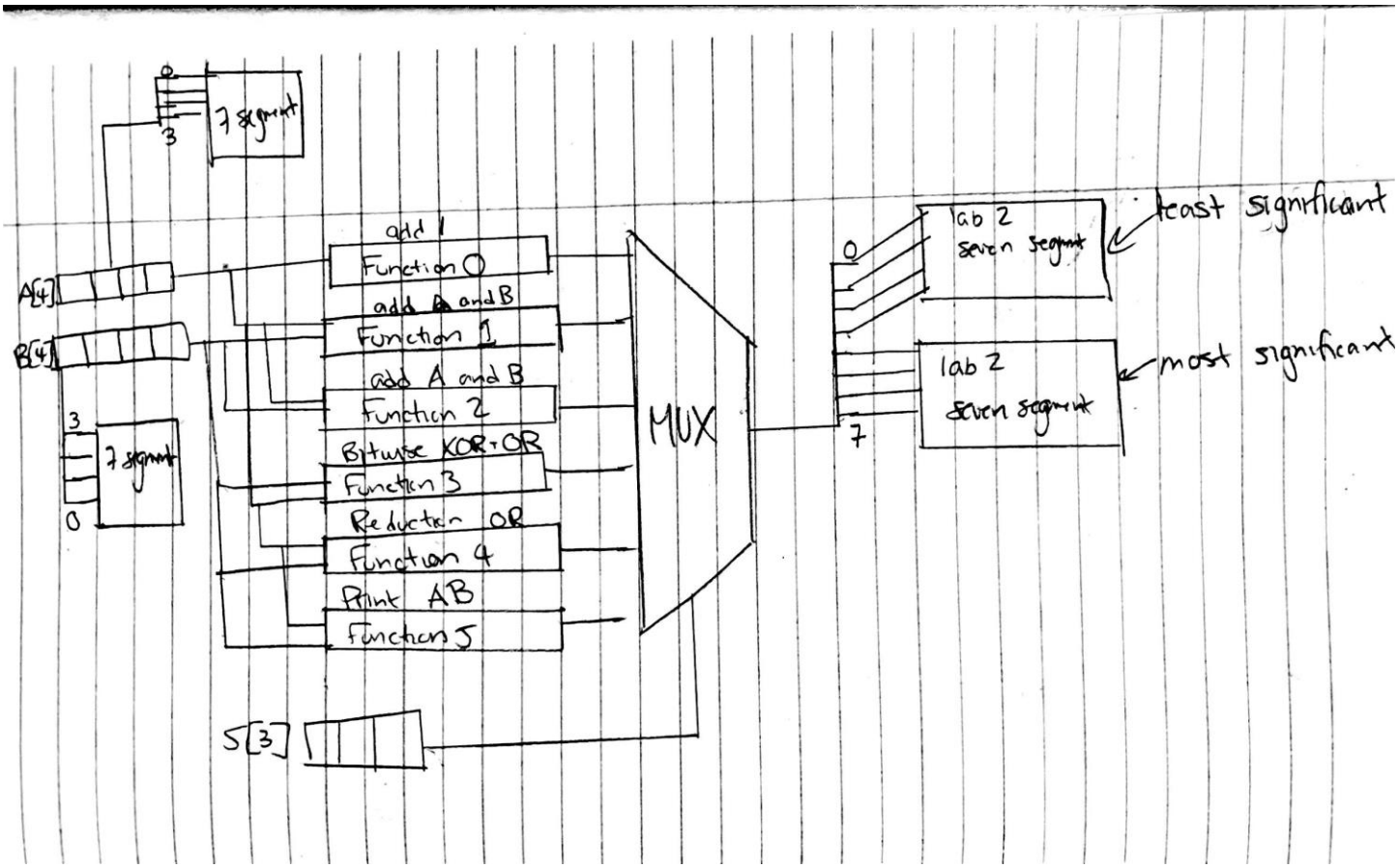
2. Below is a screenshot of a simulation result from my test vectors for this circuit

Passed: 14 Failed: 0

status	A	B	C_in	C_out	S
pass	0000	0000	1	0	0001
pass	0001	0000	1	0	0010
pass	0001	0001	1	0	0011
pass	0010	0000	0	0	0010
pass	0100	0011	0	0	0111
pass	0100	0011	1	0	1000
pass	1000	0111	0	0	1111
pass	1000	0111	1	1	0000
pass	1011	0100	1	1	0000
pass	1100	1100	0	1	1000
pass	1111	1111	0	1	1110
pass	1111	1111	1	1	1111
pass	0111	0111	0	0	1110
pass	0011	0011	0	0	0110

Part 3: Basic ALU

1. Below is a high-level schematic on how my ALU is designed with each individual function inside it's own lower level module.



Below are test vectors for each function the ALU supports.

Function 0: Adding 1 to A

Passed: 9 Failed: 0

status	A	S	Y
pass	0000	000	0000 0001
pass	0001	000	0000 0010
pass	0011	000	0000 0100
pass	0111	000	0000 1000
pass	1000	000	0000 1001
pass	1110	000	0000 1111
pass	1100	000	0000 1101
pass	1111	000	0001 0000
pass	0110	000	0000 0111

Function 1: Adding A and B with my ripple carry adder

Passed: 13 Failed: 0

status	A	B	S	Y
pass	0000	0000	001	0000 0000
pass	0001	0000	001	0000 0001
pass	0001	0001	001	0000 0010
pass	0011	0011	001	0000 0110
pass	0100	0011	001	0000 0111
pass	0111	0111	001	0000 1110
pass	1000	0111	001	0000 1111
pass	1110	0111	001	0001 0101
pass	1011	0100	001	0000 1111
pass	1100	1100	001	0001 1000
pass	1111	1111	001	0001 1110
pass	1111	0000	001	0000 1111
pass	0101	1010	001	0000 1111

Function 2: Adding A and B with built in adder

Passed: 13 Failed: 0

status	A	B	S	Y
pass	0000	0000	010	0000 0000
pass	0001	0000	010	0000 0001
pass	0001	0001	010	0000 0010
pass	0011	0011	010	0000 0110
pass	0100	0011	010	0000 0111
pass	0111	0111	010	0000 1110
pass	1000	0111	010	0000 1111
pass	1110	0111	010	0001 0101
pass	1011	0100	010	0000 1111
pass	1100	1100	010	0001 1000
pass	1111	1111	010	0001 1110
pass	1111	0000	010	0000 1111
pass	0101	1010	010	0000 1111

Function 3: Bitwise A XOR B and A OR B

Passed: 13 Failed: 0

status	A	B	S	Y
pass	0000	0000	011	0000 0000
pass	0001	0000	011	0001 0001
pass	0001	0001	011	0001 0000
pass	0011	0011	011	0011 0000
pass	0100	0011	011	0111 0111
pass	0111	0111	011	0111 0000
pass	1000	0111	011	1111 1111
pass	1110	0111	011	1111 1001
pass	1011	0100	011	1111 1111
pass	1100	1100	011	1100 0000
pass	1111	1111	011	1111 0000
pass	1111	0000	011	1111 1111
pass	0101	1010	011	1111 1111

Function 4: Reduction OR

Passed: 13 Failed: 0

status	A	B	S	Y
pass	0000	0000	100	0
pass	0001	0000	100	1
pass	0001	0001	100	1
pass	0011	0011	100	1
pass	0100	0011	100	1
pass	0111	0111	100	1
pass	1000	0111	100	1
pass	1110	0111	100	1
pass	1011	0100	100	1
pass	1100	1100	100	1
pass	1111	1111	100	1
pass	1111	0000	100	1
pass	0101	1010	100	1

Function 5: Display A with 4 most significant bits and display B with 4 least significant bits

Passed: 13 Failed: 0

status	A	B	S	Y
pass	0000	0000	101	0000 0000
pass	0001	0000	101	0001 0000
pass	0001	0001	101	0001 0001
pass	0011	0011	101	0011 0011
pass	0100	0011	101	0100 0011
pass	0111	0111	101	0111 0111
pass	1000	0111	101	1000 0111
pass	1110	0111	101	1110 0111
pass	1011	0100	101	1011 0100
pass	1100	1100	101	1100 1100
pass	1111	1111	101	1111 1111
pass	1111	0000	101	1111 0000
pass	0101	1010	101	0101 1010