ECE272 Lab 2
Spring 2018

Combinational Logic (Computer Arithmetic)

Isak Foshay

April 22nd, 2018

1. Introduction

This lab's purpose is to teach students about the digital logic of addition, as well as how to use the simulation wizard. This lab also teaches students how to create a symbol in lattice in order to use the same schematic multiple times (Like a function in higher level programming)

2. Design

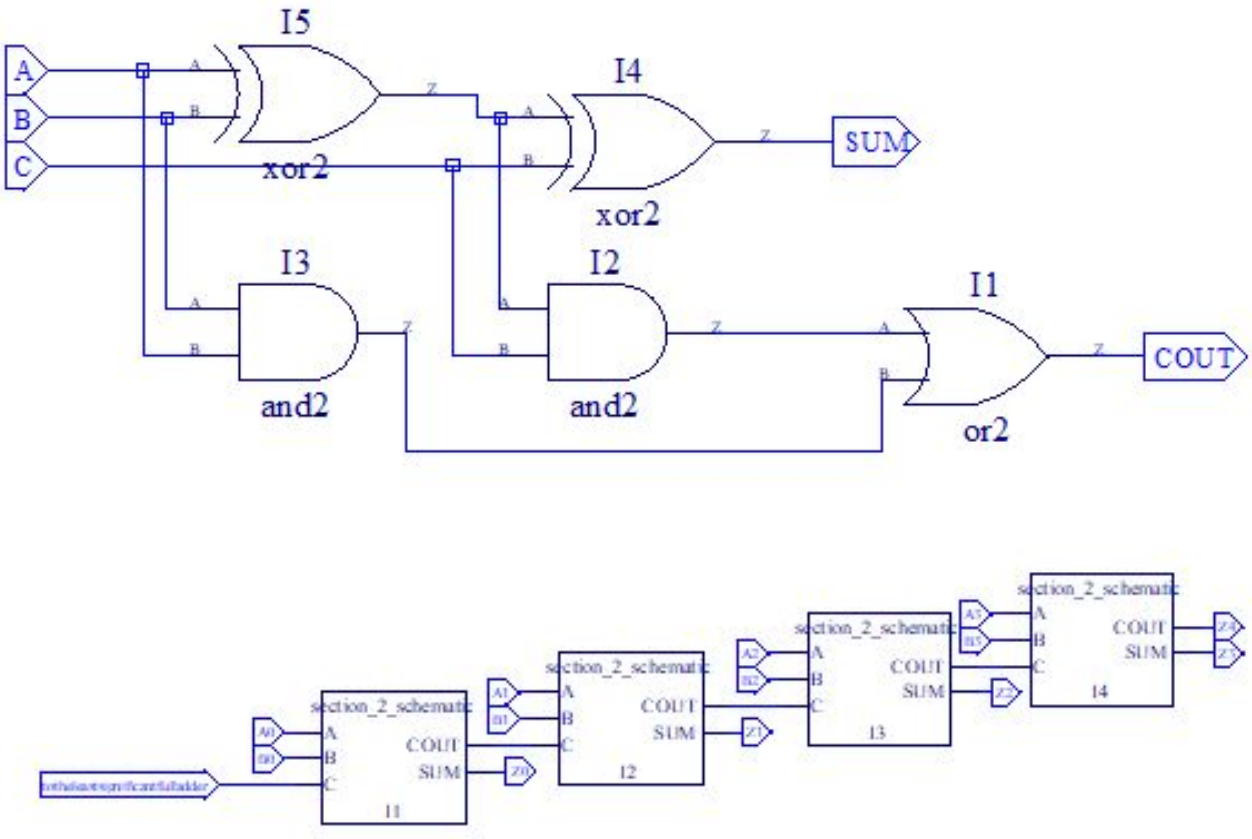
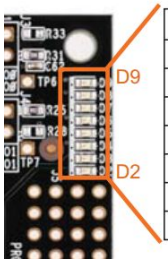


Figure 1: Schematic of logic block

	FPGA PIN	PULLMODE
INPUT A0	P9	UP
INPUT A1	M8	UP
INPUT A2	T9	UP
INPUT A3	N9	UP
INPUT B0	R9	UP
INPUT B1	R10	UP
INPUT B2	T10	UP
INPUT B3	P10	UP
CARRY	F5	UP
OUTPUT Z0	H11	DOWN
OUTPUT Z1	J13	DOWN
OUTPUT Z2	J11	DOWN
OUTPUT Z3	L12	DOWN
OUTPUT Z4	K11	DOWN

Table 1: Chosen Pins and Pull Modes

The pin choosing has become semi-arbitrary, mostly trying to keep the pins around the same location on the FPGA. For the LED pins I just went down the list to make analyzing the results easier to confirm at a glance.



LED	Net	MachXO3L Ball
D9	LED0	H11
D8	LED1	J13
D7	LED2	J11
D6	LED3	L12
D5	LED4	K11
D4	LED5	L13
D3	LED6	N15
D2	LED7	P16

Figure 2: LED pins

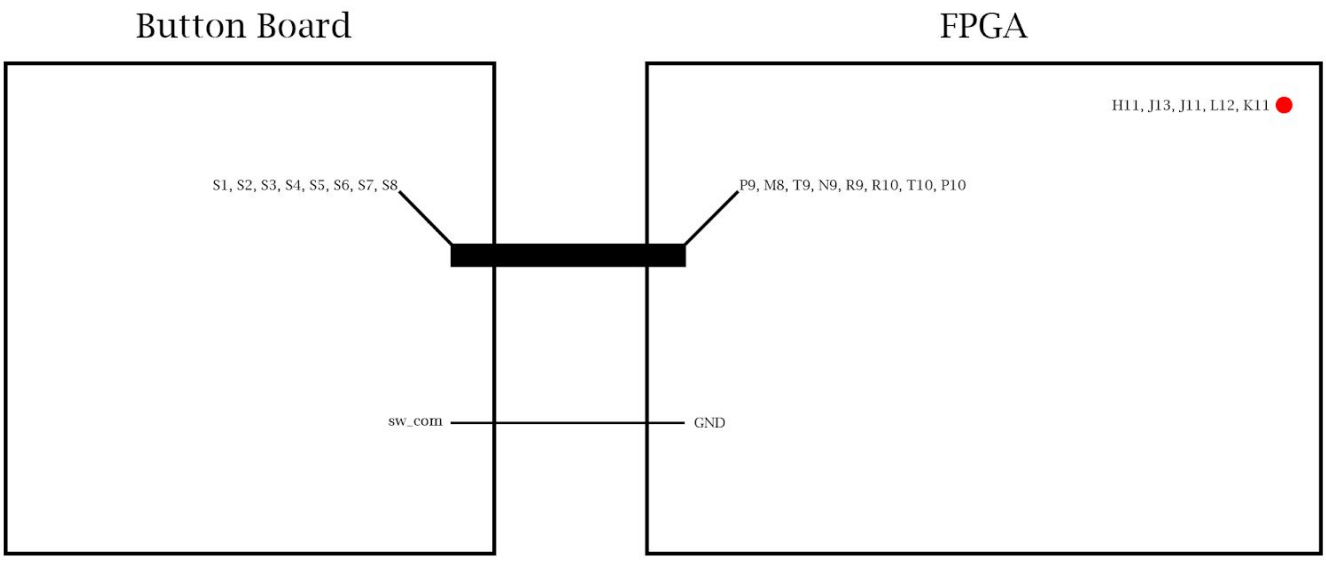


Figure 3: Block Diagram for Hardware

This diagram shows that the pins S1-S8 are connected to P9,M8T9,N9,R9,R10,T10, and P10 respectively. It also shows what needs to be connected to ground and what LEDs are being used.

3. Truth Tables

OPERAND 1	+	OPERAND 2	+	CARRY IN	=	VALUE (2-bit bin)	VALUE (uns-dec)
0b0	+	0b0	+	0b0	=	0b0	0
0b0	+	0b0	+	0b1	=	0b1	1
0b0	+	0b1	+	0b0	=	0b1	1
0b0	+	0b1	+	0b1	=	1b0	2
0b1	+	0b0	+	0b0	=	0b1	1
0b1	+	0b0	+	0b1	=	1b0	2
0b1	+	0b1	+	0b0	=	1b0	2
0b1	+	0b1	+	0b1	=	1b1	3

Table 2: Truth table for Full-Adder

OPERAND 1	+	OPERAND 2	=	VALUE (5-bit bin)	VALUE (uns-dec)	VALUE (sign-dec)
0b0000	+	0b0000	=	0b0000	0	0
0b0000	+	0b0001	=	0b0001	1	1
0b0000	+	0b0010	=	0b0010	2	2
0b0000	+	0b0011	=	0b0011	3	3
0b0000	+	0b0100	=	0b0100	4	4
0b0000	+	0b0101	=	0b0101	5	5
0b1000	+	0b0000	=	0b1000	8	8
0b1000	+	0b0001	=	0b1001	9	9
0b1000	+	0b0010	=	0b1010	10	10
0b1000	+	0b0011	=	0b1011	11	11
0b1111	+	0b0011	=	1b0010	18	-2
0b1111	+	0b1000	=	1b0111	23	-7
0b1111	+	0b1010	=	1b1001	25	-9
0b1111	+	0b1011	=	1b1010	26	-10

Table 3: Partial 4-bit Adder Truth Table

4. Simulation

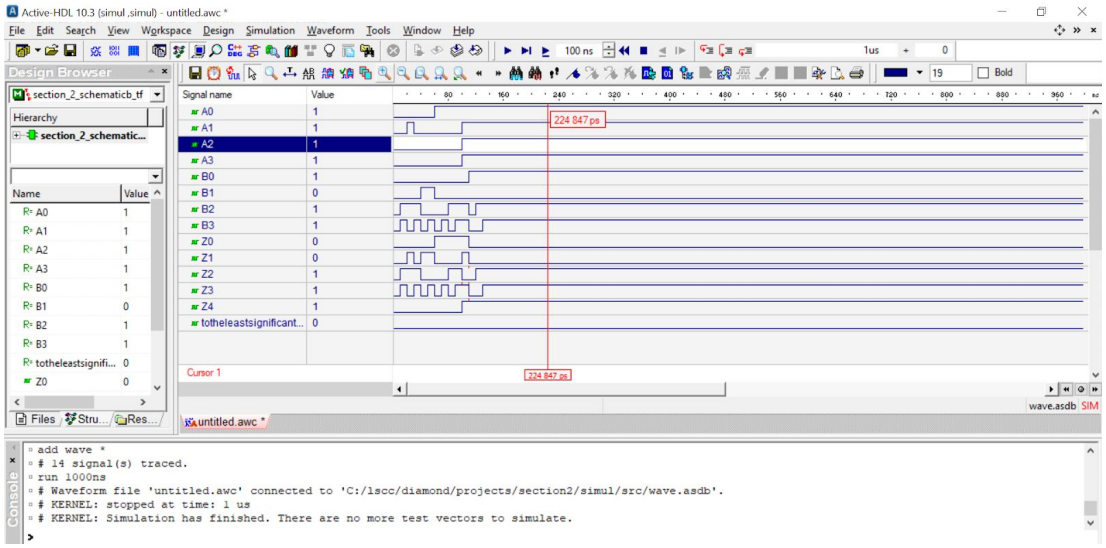


Figure 4: Lab 2 simulation

5. Results

The results were as expected, the buttons correctly adding together when pushed in combination. They also matched the (partial) simulation (Not a full simulation because the amount of possible combinations is too great)

6. Experiment Notes

This lab took longer than expected due to trouble using the lattice software, as well as filling out the tables took time.

Study Questions:

1. **Explain how you would convert my 4-bit adder to a 4-bit adder/subtractor.**

   In order to change an adder/subtractor I would flip the bits and add 1 for the number I would be subtracting (two's complement) then add the numbers together.

2. **Explain what pull resistors do in the FPGA, and why we use pull up resistors for our button inputs.**

   Pull resistors insure that a wire's signal is either high or low, no inbetween. It has the added benefit of not connecting our power directly to ground shorting out our system We use pull resistors with our button inputs so we don't short out our hardware.

3. **Explain your selection for the pullmode for the pin connected to the least-significant full-adder's carry-in**

   This should always be read as a 0 input carry because there is never going to be something to carry in.

Appendix

No code this time.