CDA4203L.001
Computer System Design Lab
Spring 2015

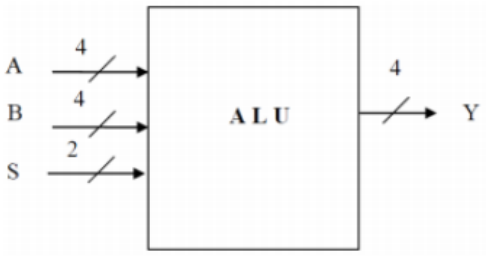
Chris Frazier
John Gangemi
Bassam Saed

# Lab 3 Report

## Introduction

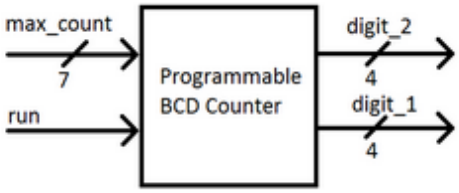
The Verilog hardware descriptive language can be used to synthesize a circuit design on an FPGA board. In this lab we will synthesize two Verilog designs from previous assignments onto our FPGA board. We will be modifying the following designs:

- 4-bit ALU




| Function | Select (S) S1 | S0 | Y |
|---|---|---|---|
| Invert | 0 | 0 | $\overline{A}$ |
| Add | 0 | 1 | A + B |
| Subtract | 1 | 0 | A - B |
| Double | 1 | 1 | 2*A |

- BCD Counter




| Port | Width | I/O | Function |
|---|---|---|---|
| max_count | 7 | I | Set the maximum counter value |
| run | 1 | I | Set run mode; 0=set, 1=run |
| digit_2 | 4 | O | Tens Digit BCD Output |
| digit_1 | 4 | O | Ones Digit BCD Output |

The FPGA board we are using has eight input dip switches and eight output GPIO LEDs. We will use these as our inputs and outputs to test the designs. Since our designs are already complete and fully tested, all we have to do is map the pins and make small modifications to accommodate the FPGA board capabilities.

## Design

In order to accommodate the FPGA board, the ALU must be reconfigured to use only eight total inputs. We will accomplish this by using 3-bit A and B inputs. This means we must modify the Verilog code accordingly. The following table shows how we will map the inputs and outputs to the pins on the board:

| Port | Width | I/O | Component | Mapping |
|---|---|---|---|---|
| In_A | 3 | I | Dip Switches | SW7 (MSB) - SW5 (LSB) |
| In_B | 3 | I | Dip Switches | SW4 (MSB) - SW2 (LSB) |
| SEL | 2 | I | Dip Switches | SW1 (MSB) - SW0 (LSB) |
| OUT | 4+ | O | GPIO LEDs | LED0 (LSB) - LED4+ (MSB) |

Since the BCD counter already uses eight inputs and outputs, we will not have to modify the Verilog code. All we have to do in this case is map the inputs and outputs to the pins on the board accordingly:

| Port | Width | I/O | Component | Mapping |
|---|---|---|---|---|
| run | 1 | I | Dip Switches | SW7 |
| max_count | 7 | I | Dip Switches | SW6 (MSB) - SW0 (LSB) |
| clk | 1 | I | Internal Clock | USER_CLK |
| Digit_1 | 4 | O | GPIO LEDs | LED3 (MSB) - LED0 (LSB) |
| Digit_2 | 4 | O | GPIO LEDs | LED7 (MSB) - LED4 (LSB) |

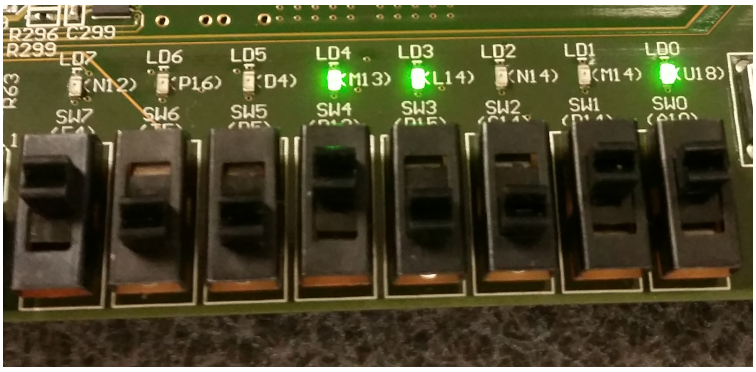
## Testing

We used two test vectors for each function including the BCD Counter, the ALU inverter, the ALU Adder, the ALU Subtractor, and the ALU Doubler. Generally we tried to include both a simple test that was easily readable and a more complex test that involved a negative number or an overflow. This allowed us to look at the first test and quickly determine if there was a flaw in the design, then perform a second more taxing test to completely analyze the designs. Below is the table of inputs and expected outputs we used.

| | BCD Counter | | ALU Invert | | ALU Add | | ALU Subtract | | ALU Double | |
|---|---|---|---|---|---|---|---|---|---|---|
| Test # | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| Run / Input A | 1 | 1 | 000 | 110 | 110 | 010 | 011 | 111 | 111 | 011 |
| Max_Count / Input B | 0010011 | 1111111 | 000 | 000 | 101 | 011 | 100 | 100 | 000 | 000 |
| Expected Output (LEDs) | 0001 1001 | 1001 1001 | 0001 1111 | 0001 1001 | 0000 1011 | 0000 0101 | 0001 1111 | 0000 0011 | 0000 1110 | 0000 0110 |

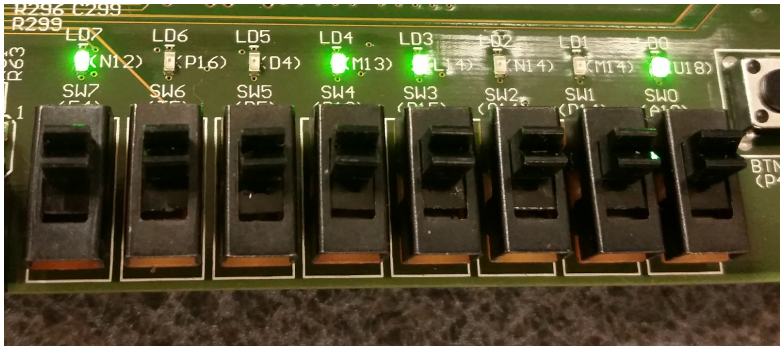
## Results

BCD Counter Test 1



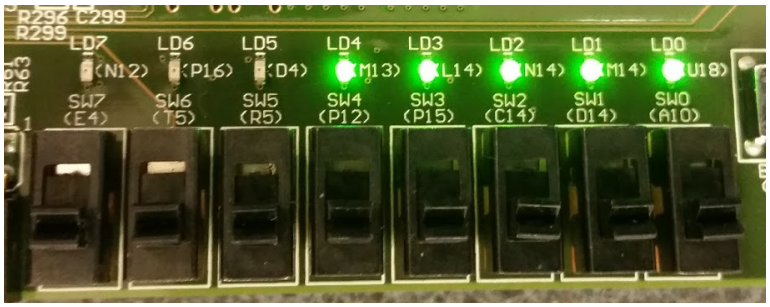

In this test, the max_count variable was set to 0010011 (19 in decimal) represented by the seven rightmost switches. As you can see the counter stopped at 0001 (1 in decimal) 1001 (9 in decimal) represented by the LEDs. This test was successful.

BCD Counter Test 2



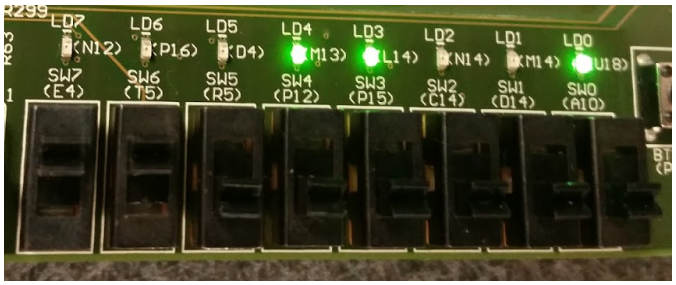

In this test, the max_count variable was set to 1111111 (128 in decimal) represented by the seven rightmost switches. As you can see the counter stopped at 1001 (9 in decimal) 1001 (9 in decimal) represented by the LEDs. This test was successful.

ALU Inverter Test 1



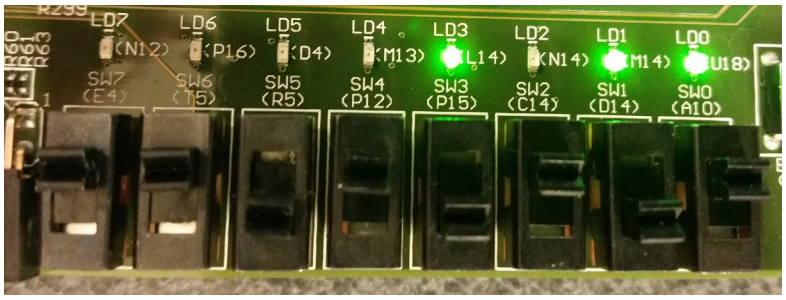

In this test, Input A was set to 000 represented by the three leftmost switches. The two rightmost switches represent the function select: 00 for the inverter in this case. As you can see the 5-bit output is 11111 represented by the LEDs. This test was successful.

ALU Inverter Test 2




In this test, Input A was set to 110 represented by the three leftmost switches. The two rightmost switches represent the function select: 00 for the inverter in this case. As you can see the 5-bit output is 11001 represented by the LEDs. This test was successful.

ALU Adder Test 1

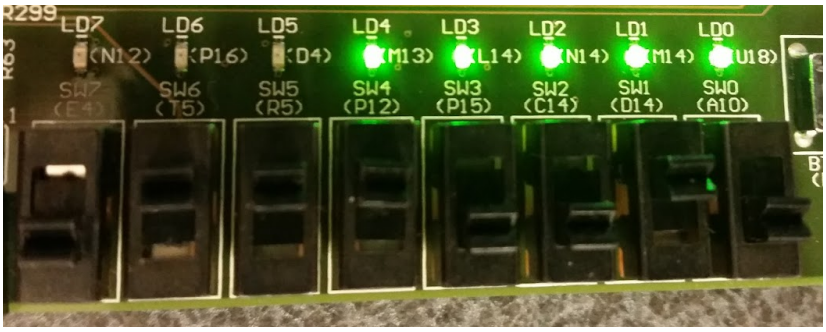



In this test, Input A was set to 110 (6 in decimal) represented by the three leftmost switches. Input B was set to 101 (5 in decimal) represented by the next three switches.The two rightmost switches represent the function select: 01 for the adder in this case. As you can see the 5-bit output is 01011 (11 in decimal) represented by the LEDs. This test was successful.

ALU Adder Test 2

In this test, Input A was set to 010 (2 in decimal) represented by the three leftmost switches. Input B was set to 011 (3 in decimal) represented by the next three switches.The two rightmost switches represent the function select: 01 for the adder in this case. As you can see the 5-bit output is 00101 (5 in decimal) represented by the LEDs. This test was successful.

ALU Subtractor Test 1

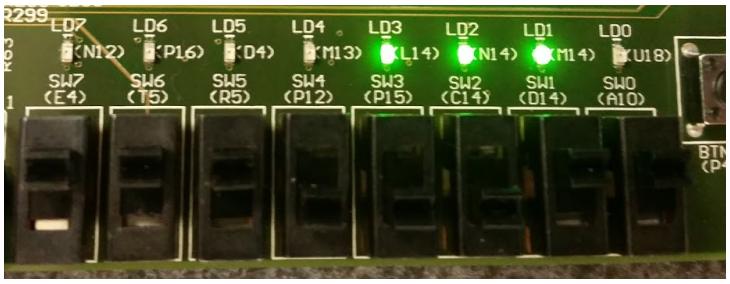



In this test, Input A was set to 011 (3 in decimal) represented by the three leftmost switches. Input B was set to 100 (4 in decimal) represented by the next three switches.The two rightmost switches represent the function select: 10 for the subtractor in this case. As you can see the 5-bit output is 11111 (-1 in twos complement decimal) represented by the LEDs. This test was successful.

ALU Subtractor Test 2

In this test, Input A was set to 111 (7 in decimal) represented by the three leftmost switches. Input B was set to 100 (4 in decimal) represented by the next three switches.The two rightmost switches represent the function select: 10 for the subtractor in this case. As you can see the 5-bit output is 00011 (3 in decimal) represented by the LEDs. This test was successful.

ALU Doubler Test 1




In this test, Input A was set to 111 (7 in decimal) represented by the three leftmost switches.The two rightmost switches represent the function select: 11 for the doubler in this case. As you can see the 5-bit output is 01110 (14 in decimal) represented by the LEDs. This test was successful.

ALU Doubler Test 2

In this test, Input A was set to 011 (3 in decimal) represented by the three leftmost switches.The two rightmost switches represent the function select: 11 for the doubler in this case. As you can see the 5-bit output is 00110 (6 in decimal) represented by the LEDs. This test was successful.

## Conclusion

Since we did not have to design any new logic functions, this lab was accomplished relatively quickly. We learned how to interface with the FPGA board and gained a basic understanding of the synthesis process, particuarly the generation of a “bit file” and how it is used to program the Spartan-6 chip.

The only task we struggled with was arranging the file structure in ISE to accommodate our pre-made verilog files and newly created constraint files (.ucf). At first the pin map files would automatically associate themselves with the wrong “top” module, however with persistent manipulation of the file structure for the project we were able to remedy this problem. In future labs we will be able to translate any functions we create into an FPGA compatible design.