SANTA CLARA UNIVERSITY	**ELEN 21** **Spring 2017**	**Dr. Sally Wood,** **Dr. Radhika Grover**

Laboratory #6: Mini-calculator with Small Arithmetic Logic Unit (ALU)

For lab sections May 16-19, 2017

I. OBJECTIVES

In this laboratory you will:

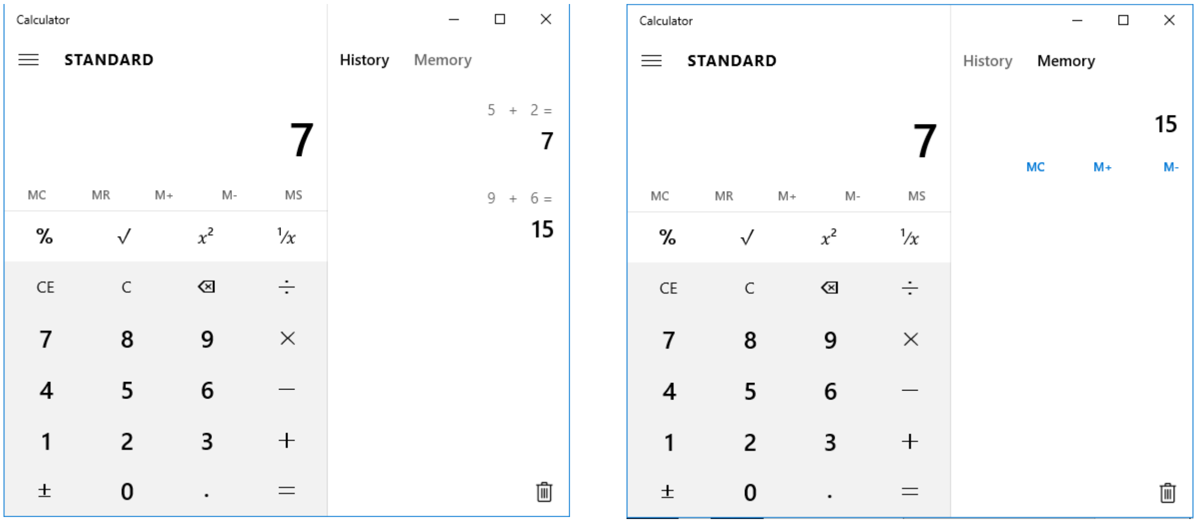
- Design and test a small adder/subtractor/logic unit (ALU).
- Use your ALU in a mini-calculator with one memory location.

PROBLEM STATEMENT

The mini-calculator will use a small Adder/Subtractor Logic Unit described below to perform arithmetic operations on 4-bit values set using switches. A pushbutton allows the current arithmetic result to be saved. An upgraded mini-calculator allows the saved value to be used as one of the operands.

- The small ALU that you will design will use a 4-bit adder to do several possible operations. The circuit will take two 4-bit input values from switches, **a3 a2 a1 a0** and **b3 b2 b1 b0,** to be the operands.
- Four switches, r1, r0, p1, and p0 will specify the arithmetic operation. These four bits can be interpreted as an **operation code**. The operations are specified in the table below. (Note that r1=0 for all table entries. A new feature for r1=1 will be added in the lab.)
- The output will be a 4-bit value **x3 x2 x1 x0** and a carry out from the adder.

Op-Select Input	**Output X**	**Notes**
r1 r0 p1 p0	**x3 x2 x1 x0**	**Operation**
0 0 0 0	X=A+B	Add
0 0 0 1	X=A+B+1	Add and Increment
0 0 1 0	X=A-B-1	Subtract and Decrement
0 0 1 1	X=A-B	Subtract
0 1 0 0	X=A	Transfer A
0 1 0 1	X=A+1	Increment A
0 1 1 0	X=A-1	Decrement A
0 1 1 1	X=A	Transfer A (duplicate – note that this can be done in 2 ways.



Windows 10 Calculator showing History and Memory

II. PRE-LAB

1. Design a four bit wide switch controlled by s. The two 4-bit inputs will be g3, g2, g1, g0 and h3, h2, h1, h0. The four outputs will be z3, z2, z1, and z0. The i^{th} output, z_i , will be g_i if s = 0 and it will be h_i if s = 1. Draw a schematic. In the lab you will make a symbol for this four bit wide switch and use it in your mini-calculator to add a feature.
2. Be sure you understand 2's complement representation of signed numbers. The 4-bit adder can be used for adding or subtracting since (-B) represented in 2's complement form can be added in the same way as (+B). Fill in the table below for your adder B inputs and carry in input for each of the four operations listed. The operations are specified by p1 and p0 when r1 = r0 = 0. The first entry has already been completed for you. Note that the B adder input will not always be the B switch values.

Op-Select Input	Adder 4-bit A inputs	Adder 4-bit B inputs	Adder Carry-in	Operation	Notes
r1 r0 p1 p0	**a3 a2 a1 a0**	**b3 b2 b1 b0**			
0 0 0 0	a3 a2 a1 a0	b3 b2 b1 b0	0	X=A+B	Add
0 0 0 1	a3 a2 a1 a0			X=A+B+1	Add and Increment
0 0 1 0	a3 a2 a1 a0			X=A-B-1	Subtract and Decrement
0 0 1 1	a3 a2 a1 a0			X=A-B	Subtract

3. Use a 4-bit adder and other logic components to design this first part of the arithmetic unit. Try to make a small efficient circuit. XOR gates are used quite often to implement efficient arithmetic circuits. Draw the schematic for your design.

4. Modify your design to add the other four operations shown in the problem statement table. Try to make a small efficient circuit that will implement 8 operations. Draw a schematic for your design.

5. Consider how you would efficiently and strategically start your test of your circuit after you implement it. Write the first four steps of your test plan. Indicate what values you would set

for the inputs and what you would expect to see at the outputs of each step. Indicate what you would have verified if you see the expected output. There are many possible testing strategies.

IV. PROCEDURE

Part 1 – Design and test the small ALU

ALU schematic:

1. Draw the schematic for your design of the small 8-operation ALU from the problem statement using the Altera libraries. Be sure that all inputs to your components are controlled and none are left floating.

Inputs/Outputs:

2. Connect the inputs of your mini-calculator circuit. Use bus connections as in the previous laboratory for the two 4-bit input values **a3 a2 a1 a0** and **b3 b2 b1 b0** for the switches on the board. Connect the operation select inputs of your circuit, **r1 r0 p1 p0,** to switches.
3. In order to observe the inputs of your circuit you will connect them to seven segment displays or LEDs.
    a. Connect **r1 r0 p1 p0** to LEDs.
    b. Connect the adder carry-in to an LED
    c. Connect switch inputs **a3 a2 a1 a0** to a 7 segment display. It will show the 4-bit binary value as a hexadecimal digit: 0,1,2,3,4,5,6,7,8,9,A,B.C,D,E, or F.
    d. Connect switch inputs **b3 b2 b1 b0** to a 7 segment display.
    e. Connect the four internal B inputs to your adder to a seven segment display**.** For some operations this will be the same as the B switches, but for other operations it will be different.
4. Connect the outputs to displays.
    a. Connect the adder carry out to an LED.
    b. Connect the adder output **x3 x2 x1 x0** to a 7 segment display.

5. Download your circuit to the FPGA.

Testing:

6. Check that your input switches are connected in the correct order. Set the input switches to values 1, 2, 4, and 8 and verify that the seven-segment displays show the correct values.
7. Check that the arithmetic operation is correct using a test plan. Try various inputs and ensure that all operations are being executed correctly in the ALU.

Example:
TEST 1: Set all data input switches and operation control switches to zero. Observe the output on the LEDs and seven segment display. What is the operation in this case?
EXPECTED RESULT: All LEDs should be off. Seven segment displays should show '0's.
PURPOSE: This is a very basic test of the input connections since it should be adding A+B with both inputs 0. Since the LEDs and the seven segment displays are our only indicators of how the circuit is functioning, we should test them first to see if they are working correctly.
TEST 2: Set the two data inputs to a value of 2 decimal. What do you expect to see at the output of the adder? If the output of the adder is 8 instead of the expected value, what might be the cause of the incorrect result?

8. Verify and demonstrate that the circuit functions correctly for all eight operations. Note that the adder logic is the same for addition of unsigned integers and signed integers in 2's complement representation, so the inputs and the 4-bit result may be interpreted either way. Also note that the results will not appear to be correct if the sum is not within the range that can be represented by 4 bits. This would be an overflow condition.
9. When your test plan has been successful, demonstrate the operations to the TA, who will test several input combinations.

Part 2: Add a memory save to the mini-calculator.

10. From the Quartus II library components, add one *8dff* component to make one memory location to save the results of an arithmetic operation. This part will store 8 bits, but we will use only four bits. Go to libraries→other→Max+Plus II and select *8dff.*
    a. Connect a pushbutton switch to the CLK input of the 8dff.
    b. Connect the CLRN and PRN inputs to a logic 1 level, Vcc. The clear and preset inputs will not be used in your circuit.
    c. Connect the adder output **x3 x2 x1 x0** to the inputs D1 to D4
    d. Make a bus for outputs Q1 to Q4 named **m3 m2 m1 m0** and connect the M outputs to another seven segment display so you can always see whatvalue has been saved.

11. Add the MUX circuit you designed in the prelab to switch between the B switch inputs and the M inputs. The data inputs to the MUXes should be the B switches and the M outputs from the memory. The MUX select will be r1.

12. Disconnect the input to your ALU circuit that came directly from the B switches and replace it with the D output of the Muxes, ie **m3 m2 m1 m0**. Now if r1 = 0, your ALU should operate as it did in part 1. If r1 = 1, the operations will use A and M rather than A and B.
13. Download your circuit to the FPGA.

Testing:

14. Verify that the ALU works as it did in Part 1 when r1 = 0.
15. Test the pushbutton operation to save the adder output. The display of M should not change as you change switches. It should only change when you press the pushbutton.
16. Test the operation of your mini-calculator. You should be able to do the operations from Part 1, save a result, and then use that saved result in another operation.
17. Show a sequence of operations using memory that will make your ALU output count by 3s. You should see the sequence 3, 6, 9, c=12, f=15 on the seven segment display.

V. REPORT

1. Introduction, procedure, results, conclusions and references.
2. Include schematic, test plan, and results for each operation.
3. Describe how negative results appear in the seven-segment display.
4. If the B input switches are zero, describe a sequence of operations using memory that you could use to get a minicalculator output of –A. How would you get an output equal to 3A? (Note: A means the value of the A switches at the first step, not hexadecinmal 10.)
5. How would you connect two 4-bit ALUs to make an 8-bit ALU?