**ELEC 204 Digital Design Preliminary Lab Report**

Preliminary Lab 3

Name: Mehmet Enes Erciyes

Date: date of submission on blackboard (MM/DD/YYYY)

\*Please delete the highlighted lines and write your own parts for the report.

\*\*Reminder: Your lab grade is a weighted average of your performance before, during and after the lab: **Total lab grade = Preliminary work\* (30%) + Lab interview and demo\* (40%) + Lab report (30%)**

**\*\*\*Please make sure the preliminary report does not exceed 2 A4 pages.**

For the preliminary work, please provide here:

* *Your answers to the questions in the tutorial lab*
* Circuit schematics (if applicable)
* State diagram (only applies for Labs 3, 4 and 5)

**Question 4.1**

7 = (0111)2 -7 = (1001)2 --- Take the complement and add one

3 = (0011)2 -3 = (1101)2

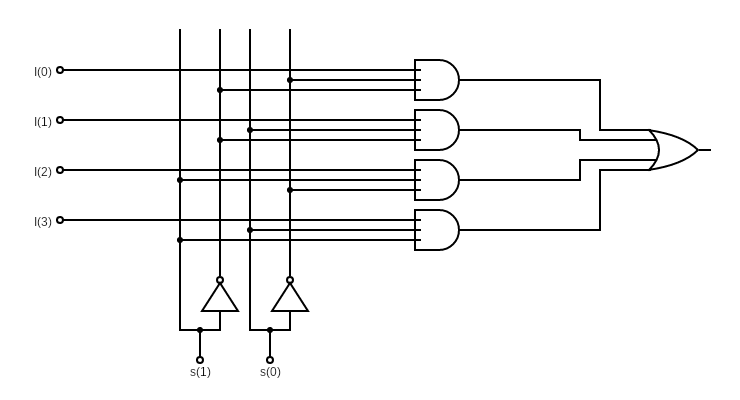
1 = (0001)2

1-7 = 1 + (-7) = (0001)2 + (1001)2 = (1010)2 = (-6)10 --- Since there is no overflow bit we see that this result is negative

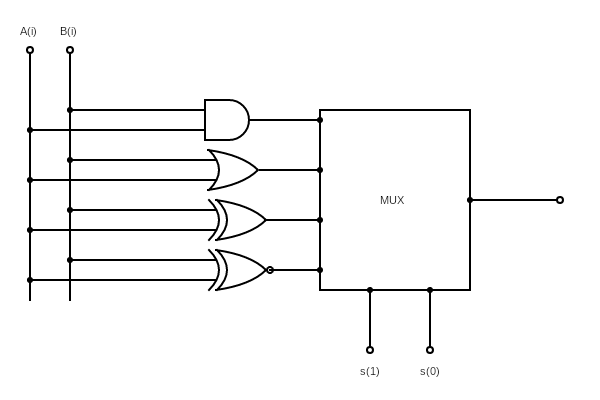
-3+7 = (1101)2 + (0111)2 = (10100)2 = (4)10 --- We discard the extra bit

**Question 4.2**

4-to-1 Multiplexer Design:

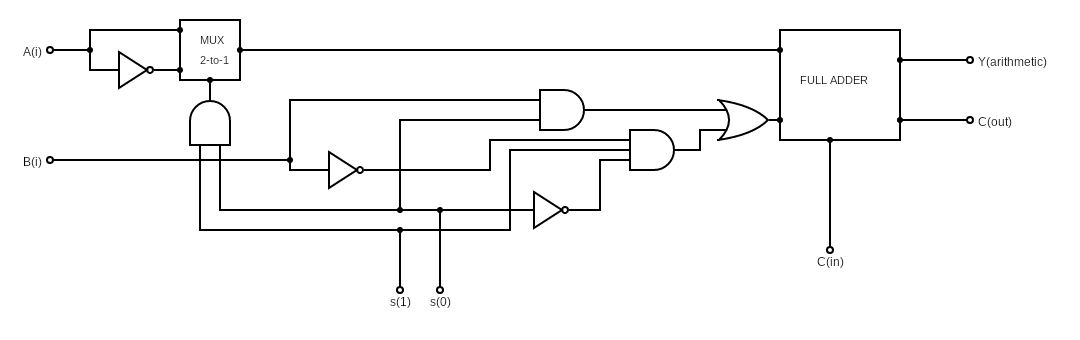


**Question 4.3**

According to the truth table, s(1) and s(0) acted as select bits that chooses which operation the logic unit performs. A 4-to-1 multiplexer can handle the selection of 4 different operations with two select bits. Multiplexer’s inputs are the results of functions applied to A(i) and B(i). Select bits choose which function to map to output according to given truth table.

In this circuit, the multiplexer whose design is given above is used.

**Question 4.4**



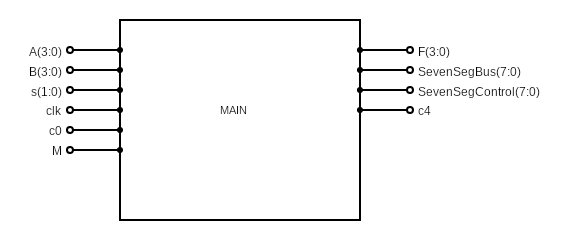
The arithmetic unit is a bit more complex in design.

To design this unit, we followed the following procedure:

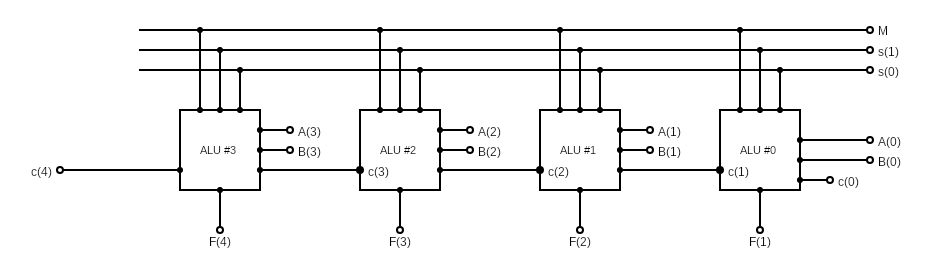
* Separated the function into three pieces : A , B and +1
* First of all, +1 only depends on C(0) – C(in) in the sketch. Therefore it is already the input to the full adder.
* Second, we looked at the A part. For only s(1)=1 and s(0) = 1, A(i) is complemented, for all the other times, it is A(i).
  + Therefore, A part is chosen with a 2-to-1 multiplexer which takes s(1)s(0) as the select bit. If s(1)s(0) = 0, A part is A(i) else it is A(i)’
* Last, we looked at the B part.
  + B part could be implemented using a 4-to-1 multiplexer. However, finding the minimum gate implementation using the truth table was more efficient.
  + B = s(0)B(i) + s(1)s(0)’B(i)’
* Arithmetic Unit gets 5 inputs. A(i),B(i),s(1),s(0) and C(0). First, it maps A(i) and B(i) to the correct input for full adder according to the select bits. Then full adder works and gives the output and carry bit.

**Question 4.6**

a) At the highest level, our main module will have the following RTL schematic:



And, the modular connection of one-bit ALUs will be like:



b) We are using the slower clock with the pin number P40. But I could not find at which frequency the clock works.

c) UCF File for the main module will be like:

net s(1) loc = BTN5;

net s(0) loc = BTN4;

net A(3) loc = P84;

net A(2) loc = P86;

net A(1) loc = P89;

net A(0) loc = P93;

net B(3) loc = P3;

net B(2) loc = P6;

net B(1) loc = P13;

net B(0) loc = P16;

net M loc = P77;

net c0 loc = P83;

NET "SevSegcontrol[0]" LOC = P50;

NET "SevSegcontrol[1]" LOC = P49;

NET "SevSegcontrol[2]" LOC = P52;

NET "SevSegcontrol[3]" LOC = P56;

NET "SevSegcontrol[4]" LOC = P59;

NET "SevSegcontrol[5]" LOC = P57;

NET "SevSegcontrol[6]" LOC = P60;

NET "SevSegcontrol[7]" LOC = P61;

NET "SevSegbus[7]" LOC = P71;

NET "SevSegbus[6]" LOC = P62;

NET "SevSegbus[5]" LOC = P65;

NET "SevSegbus[4]" LOC = P72;

NET "SevSegbus[3]" LOC = P73;

NET "SevSegbus[2]" LOC = P98;

NET "SevSegbus[1]" LOC = P64;

NET "SevSegbus[0]" LOC = P70;

NET "clk" LOC = P40;

**ELEC 204 Digital Design Lab Report**

Lab XX

Name: Student Name

Date: date of submission on blackboard (MM/DD/YYYY)

\*Please delete the highlighted lines and write your own parts for the report.

\*\*Reminder: Your lab grade is a weighted average of your performance before, during and after the lab: **Total lab grade = Preliminary work\* (30%) + Lab interview and demo\* (40%) + Lab report (30%)**

**\*\*\*Please make sure the lab report does not exceed 4 A4 pages.**

\*\*\*\*Please make sure you indicate the name of your lab collaborator (if there is any) who you worked together to solve the lab questions. Do not change your lab collaborator throughout the semester.

1. **Introduction and objectives**

Explain the objectives of the lab (refer to the lab instruction sheet),

Explain what your code has to do and describe how you did it.

1. **Methods**

Explain the inputs (how many bits, names of the inputs),

Explain the outputs (how many bits, names of the outputs),

Explain what the VHDL code must do

Explain how your code works

Provide the truth table

1. **Problems encountered, errors and warnings resolved**

Explain what problems you encountered while writing your code.

Explain what synthesis errors and warnings you observed.

Explain what problems you had to solve (or could not) on your board even if your code could be synthesized successfully.

1. **Conclusion**

Provide a 1 paragraph summary of the lab and explain what you learned from this lab.

References

1. Please cite any resource (web site, book, youtube video) you used for this lab.

**Appendix 1. Lab source code**

**Appendix 2. RTL schematics**

**Appendix 3. FPGA Board photos showing working code**

**Appendix 4. Screenshots from Xilinx for the errors and other board issues**