ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 222E
COMPUTER ORGANIZATION

PROJECT 2 REPORT

GROUP NO : 1

GROUP MEMBERS:

150170095 : Mehmet ALTUNER
150170026 : Berkay OLGUN
150170103 : Atahan ÖZER
150170076 : Ekrem UĞUR

SPRING 2020

1 Part 1

In the first part of the project, we are asked to design an ALU that takes two 8-bit data to operate on and 4-bit function selection data to select a function among 16 available functions.

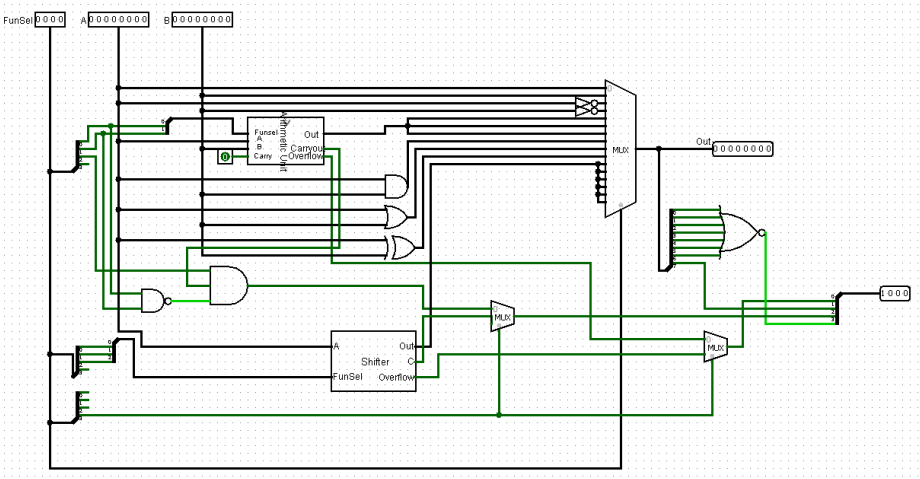

Figure 1: ALU circuit

We have used a MUX to choose the correct output. It has a 4-bit selection input, namely funsel, to choose from 16 different functions. Some functions are handled directly inside ALU circuit (e.g. the first function where the output is equal to the input 'A') while some functions are implemented in different units for the sake of simplicity of design and readability. These units are described below.

1.1 Arithmetic Unit

This unit handles the arithmetic functions.

- A + B
- A + B + carry
- A - B

This unit also takes a funsel data of 3-bit. This funsel is the same as the first 2-bit's of the main funsel that is given to the ALU. The reason is that, there are total of 3 arithmetic operations that we are interested in and we can select which arithmetic functions we use by the first 2 bit of the funsel.

Zero flag is determined by checking the NOR of the bits of the output.

Negative flag is determined by checking if the most significant bit is set or not.

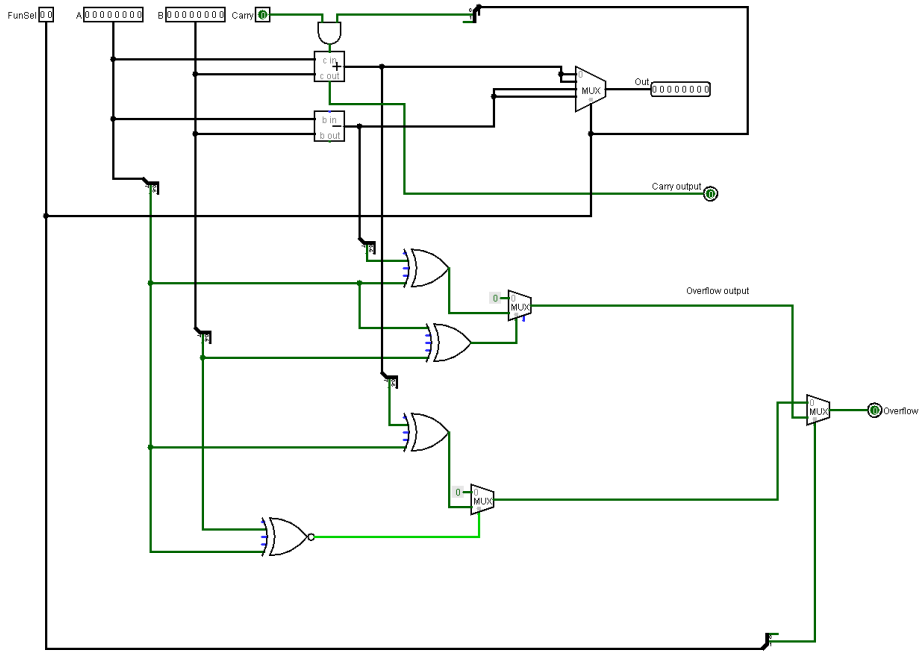

Figure 2: Arithmetic Unit Circuit

1.1.1 A+B

Sum operation is implemented by using a full adder.In order to check overflow we designed a simple sub circuit.MSB(A) and MSB(B) pinned to a XNOR gate so that it can be checked if signs are same so that overflow may occur,latter output of this operation connected to selector port of the multiplexer.In order to check overflow, output of the subtractor and MSB(A) pinned into an XOR gate.Truth table and subcircuit can be found below.

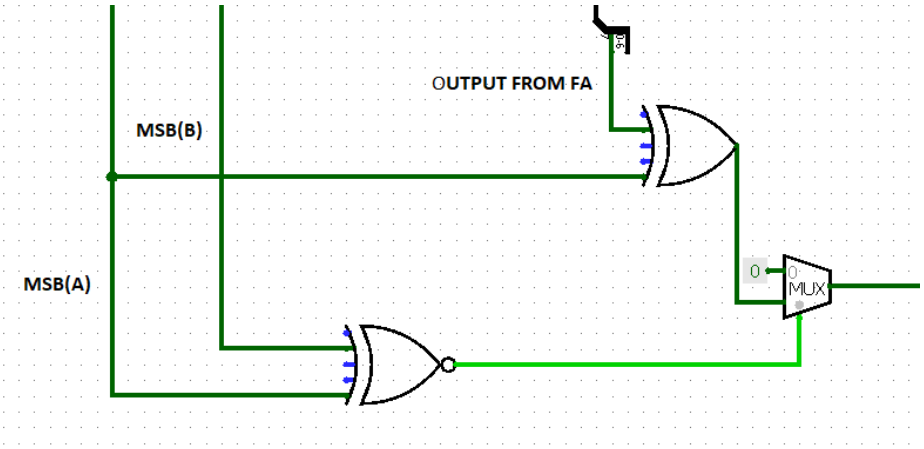

Figure 3: A+B overflow check

A	B	MSB(OUTPUT)	OVERFLOW
0	0	0	0
0	**0**	1	1
1	1	0	1
1	1	1	0

1.1.2 A-B

Subtraction operation is implemented by using a subtractor unit in logisim.In order to check overflow we designed a simple sub circuit.MSB(A) and MSB(B) pinned to a XOR gate so that it can be checked if signs are different so that overflow may occur,latter it is connected to selector port of the multiplexer. In order to check overflow, output of the subtractor and MSB(A) pinned into an XOR gate.Truth table and subcircuit can be found belove.

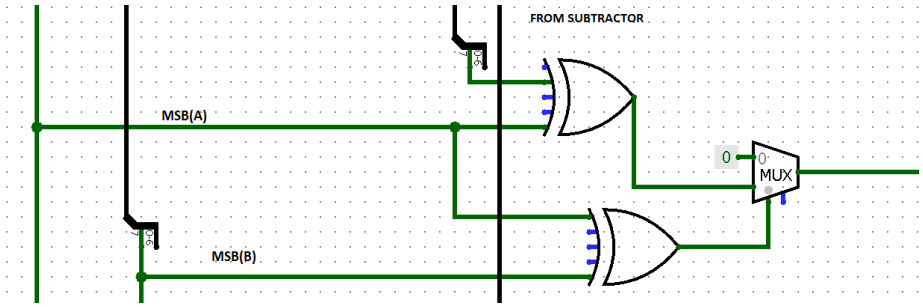

Figure 4: A-B overflow check

A	B	MSB(OUTPUT)	OVERFLOW
0	1	0	0
0	**1**	1	1
1	0	0	1
1	0	1	0

1.2 Shifter

This unit handles the shifting operations (ASL, ASR, LSL, LSR, CSL and CSR). The funsel for the Shifter Unit is of 3-bit because there are total of six shifting operations and we can select which shifting operation we use by the first 3 bit of the funsel.

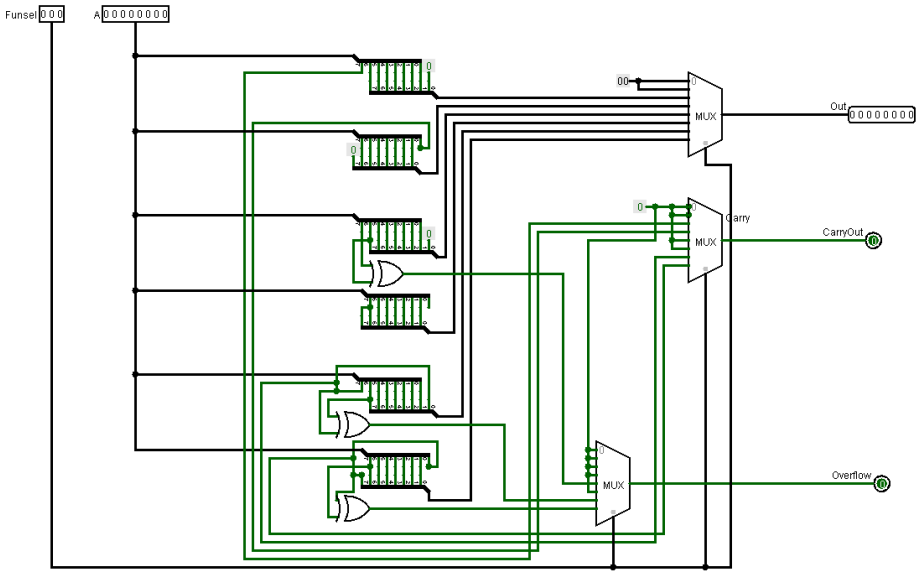

Figure 5: Shifter circuit

FunSel		Function
0	1010	LSL
1	1011	LSR
2	1100	ASL
3	1101	ASR
4	1110	CSL
5	1111	CSR

Table 1: Funsel inputs and their corresponding shifting functions

- LSL function shifts bits to the left by one (from LSB to MSB) and carry register will hold ex-MSB.
- LSR function shifts bits to the right by one (from MSB to LSB) and carry register will hold ex-LSB
- ASL function shifts bits to the left by one (from LSB to MSB). Note that overflow may occur.
- ASR function shifts bits to the right by one (from MSB to LSB). Note that overflow may not occur because MSB remains the same.
- CSL function shifts bits to the left by one in a circular manner (from MSB to LSB, MSB becomes LSB). Carry register will hold ex-MSB.

- CSR function shifts bits to the right by one in a circular manner (from LSB to MSB, LSB becomes MSB). Carry register will hold ex-LSB.

If a function that has a carry is selected, according carry data is given as output. For example for the CSL function the carry is set to the 7th bit of the data.

Again, there is an overflow detection for this unit and it is simply implemented by the output of the XOR values of the most significant bits and the bit it will become the MSB. For example for the ASL there is an overflow if the XOR of 7th and 6th bits are 1.

2 Part 2

In the second part of the project we combined all our previously implemented designs within an organization.

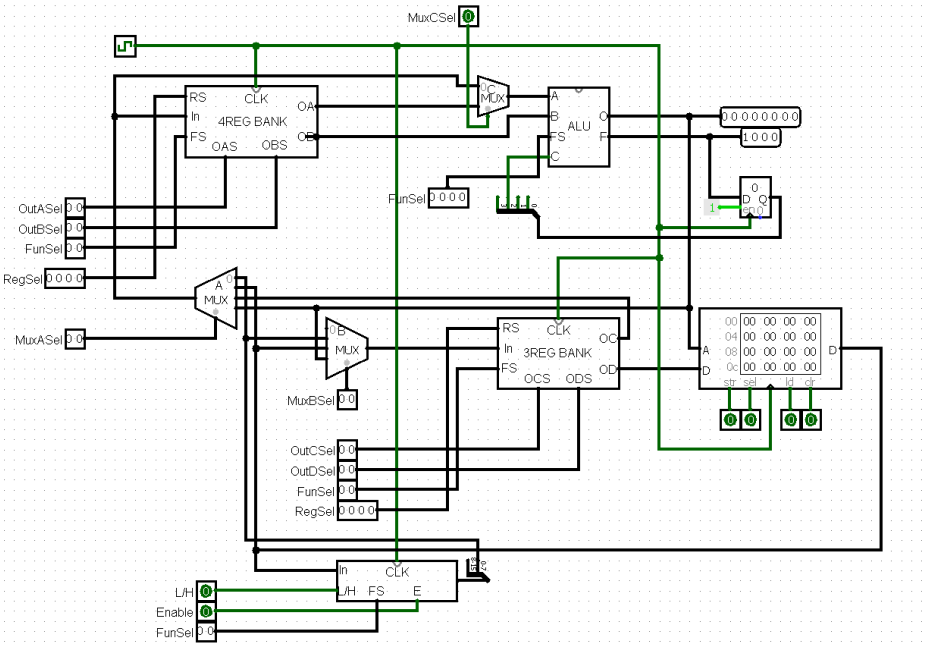

Figure 6: ALU circuit

In this part there are 3 multiplexers, 2 register files, a memory, a register and an ALU is connected to each other. All of the inputs are selectors for multiplexers and other files. Multiplexers characteristic tables are as follows.

MuxASel		MuxAOut
0	0	IROut(0-7)
0	1	Memory Out
1	0	Address Reg Out
1	1	OutALU

MuxBSel		MuxBOut
0	0	ϕ
0	1	IROut(0-7)
1	0	Memory Out
1	1	OutALU

MuxCSel	MuxCOut
0	MuxAOut
1	OutA

Table 2: Input-Output tables of Multiplexers

- ALU

  ALU is used to operate on inputs and give outputs to the memory and Multiplexer A. Inputs of A are coming from Multiplexer C and inputs for B are coming from our Register File's OutB section.

- Register File

  Register file is taking input from multiplexer A which possible outputs are shown above. The OutA selector selects what will go to MuxC and OutB selector selects what will go to the second output of ALU. RegSel activates registers, FunSel selects the functions which we will apply on the input coming from Multiplexer A.

- Address Register File

  Address register file is taking input from Multiplexer B. OutCSel selects what will appear on OutC and OutDSel selects what will appear on OutD, which will eventually go to the memory. Funsel selects the functions we will apply on the input from Multiplexer B. Regsel activates the registers we want to apply on.

- Memory

  Memory part in this homework is taking input from OutALU and Address Register's OutD. And storing the inputs if the memory is write enabled.

- Instructions Register

  Instructions register is taking input from Memory's output section. L/H flag selects the part we will work on, either bits 0-7 or 8-16. FunSel here selects the function to apply on the input here again. Enable is the general enable switch for the register, if it is closed no function will work.