Department of Computer Engineering

BLG 222E
Computer Organization

Project Report

Project : 2

Date : 12.04.2017

Group Members:

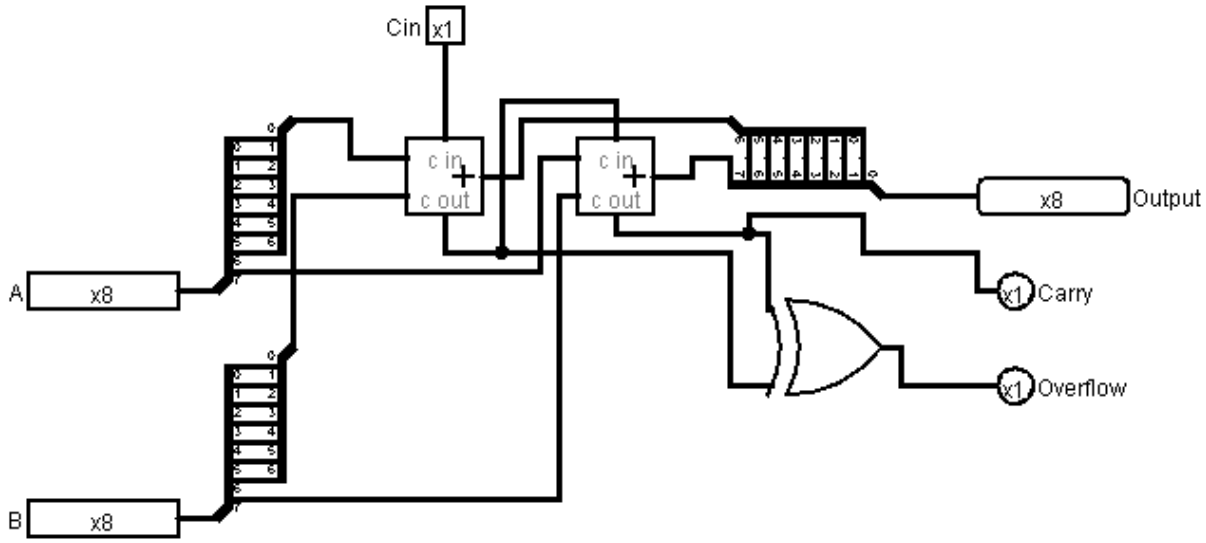
ID	Name	Surname
040140003	*Burak Mert*	*Gönültaş*
150140025	*Elif*	*Kılıç*
150150701	*Yunus*	*Güngör*
150130029	*Muhammed Emin*	*Küçük*

1. INTRODUCTION

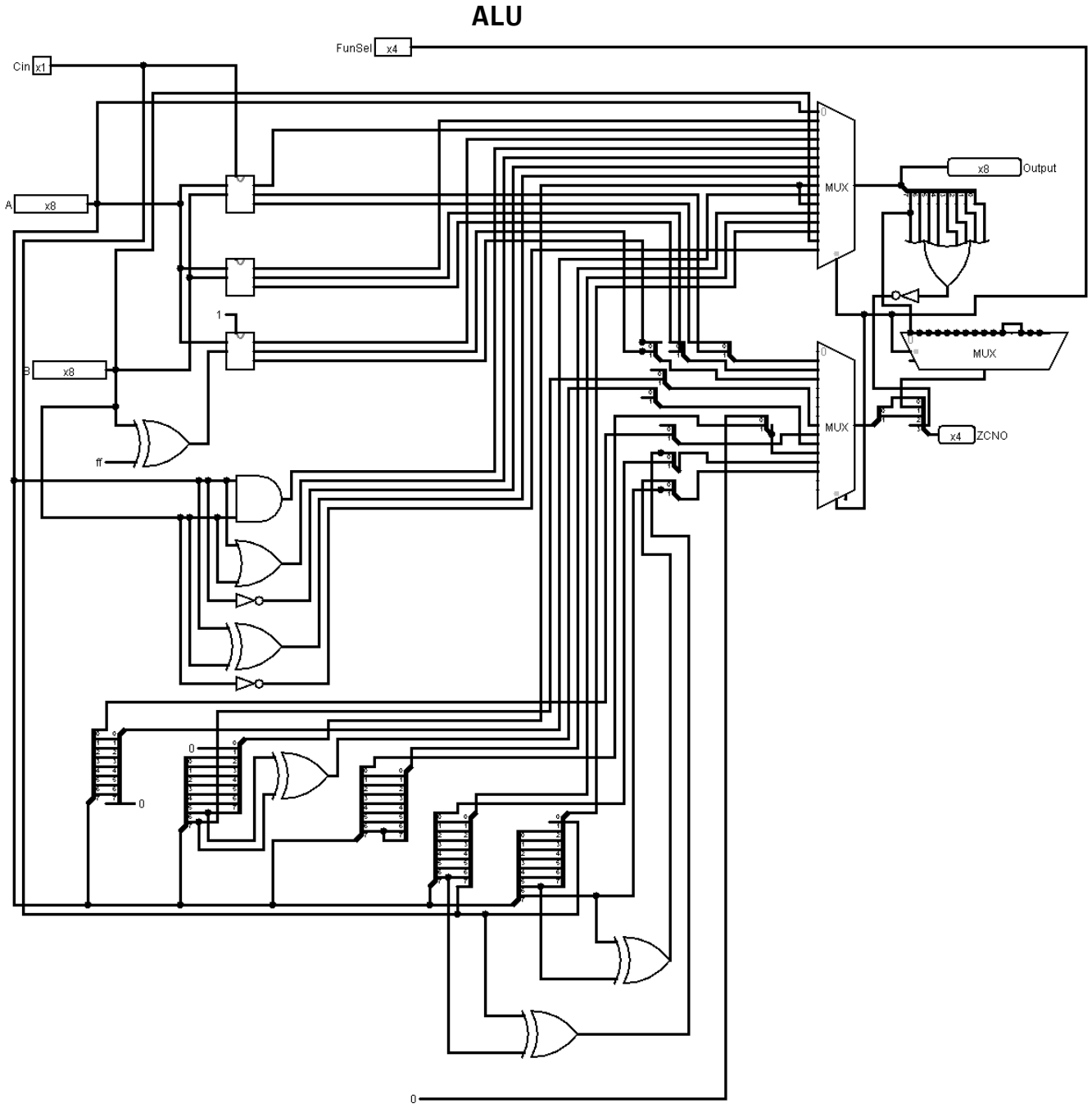
In this Project, we designed an 8-bit ALU that has 4 flags, “Zero”, “Carry”, “Negative”, “Overflow”. After that, we placed that ALU into a predefined structure containing several memory units and multiplexers.

2. REQUIREMENTS

ADDER WITH OVERFLOW



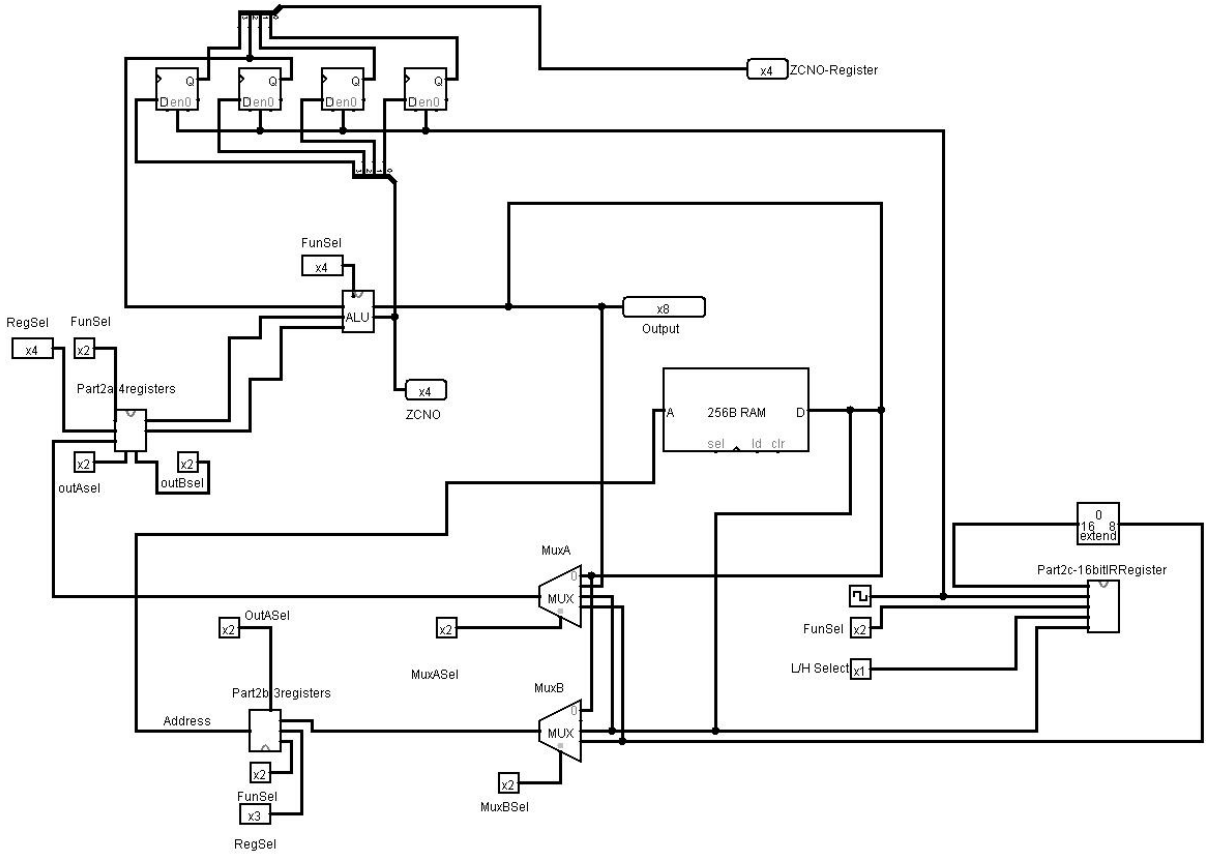
To achieve the desired ALU goals, we had to first design a special adder that contains discrete carry and overflow outputs. The overflow output is dependent on the overflow detection network.



After that, we designed the ALU with following capabilities, tied to FunSel selections.

FunSel	OutALU	Z	C	N	O
0000	A	√	–	√	–
0001	A + B	√	√	√	√
0010	A + B + Carry	√	√	√	√
0011	A - B	√	√	√	√
0100	A AND B	√	–	√	–
0101	A OR B	√	–	√	–
0110	NOT A	√	–	√	–
0111	A XOR B	√	–	√	–
1000	LSL A	√	√	√	–
1001	LSR A	√	–	√	–
1010	ASL A	√	–	√	√
1011	ASR A	√	–	–	√
1100	CSL A	√	√	√	√
1101	CSR A	√	√	√	√
1110	B	√	–	√	–
1111	NOT B	√	–	√	–

ORGANIZATION



MuxASel	MuxAOut
00	OutALU
01	Address
10	Memory Output
11	IROut (0-7)

MuxBSel	MuxBOut
00	OutALU
01	ϕ
10	Memory Output
11	IROut (0-7)

The ALU and the parts from the previous project are brought together in the desired form.

3. CONCLUSIONS

We implemented an ALU that is capable of doing ADD, SUBTRACT, AND, OR, NOT, XOR, LSR, LSL, ASR, ASL, CSR, CSL operations, depending on user defined FunSel bits. Prior to designing the ALU, we had to design a special adder with overflow detection and output. With this homework we improved our Logisim skills.