ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 222E

COMPUTER ORGANIZATION
PROJECT 3

EXPERIMENT NO : 3

EXPERIMENT DATE : 03.06.2020

GROUP NO : G18

GROUP MEMBERS:

150160141 : RESUL LEVENT KURU

150170043 : EBRAR ÖMER

150180064 : MELİS GÜNŞEBER

150180715 : GÖKSU GÜZ

SPRING 2020

Contents

1 INTRODUCTION

In this project we are asked to design a control unit which produce control signals.Computer has instructions to make operations which are fetch, decoded and executed.

2 ORGANIZATION

The control unit performs operations at three stages within specific times.

2.1 Instructions

2.1.1 Fetch and Decode

T0:IR(8-15) ← M[PC] At first, we select the PC at OutDsel and send it to memory to read what is in the PC.

OutDsel: 00;
Read: 1;
L/H':0;
Enable: 1;
FunselT:01

T1:PC ← PC + 1 Then, we increment the program counter to read the operation part of IR.

RegselP2B: 001;
FunselP2B: 10;

T2:IR(0-7) ← M[PC] We read the IR but not the address part and decide to operation.

OutDsel: 00;
Read: 1;
L/H':1;
Enable: 1;
FunselT:01

T3:PC ← PC + 1 The fetch operation ends after that time, the operation starts with the decoded opcode.

RegselP2B: 001;
FunselP2B: 10;

2.1.2 Execute

D0: Ri ←Value We write the value to the register that choosen.

IR(8)=1(Direct) D0T4:
MuxBSel: 01;
RegSelP2B: 010;
FunSelP2B: 01;

D0T5:
OutDSel: 01;
Read: 1;
MuxASel: 01;
RegSelP2A: IR(10,9));
FunselP2A: 01;
SC <− 0;

IR(8)=0(Immediate) D0T4:
MuxASel: 00;
RegSelP2A: IR(10,9));
FunSelP2A: 01;
SC <− 0;

D1: Value ← $\mathbf{R}_i$ With this operation we can store the value from a register.

D1T4:
MuxBSel: 01;
RegSelP2B: 010;
FunselP2B: 01;

D1T5:
OutDSel: 01;
OutASel: IR(10,9);
MuxCSel: 1;
FunSelALU: 0000;
Load: 1
SC <− 0;

D2: Destreg ← SrcReg1 This instruction moves data between the registers.

IR10'IR7': (Reg. file ←Reg. File)
D2T4:
OutASel: IR(6,5);
MuxCSel:1
FunSelALU: 0000;
MuxASel: 11;
RegSelP2B: IR(9,8);
FunSelP2B: 01;
SC <− 0;

IR10'IR7: (Reg. file ←Add. Reg. File)
D2T4:
OutCSel: IR(6,5);
MuxASel: 10;
RegSelP2B: IR(9,8);
FunSelX: 01;
SC <− 0;

IR10IR7': (Add. Reg. file ← Reg. File)
D2T4:
OutASel: IR(6,5);
MuxCSel: 1;
FunSelALU: 0000;
MuxBSel: 11;
RegSelP2A: IR(9,8);
FunSelY: 01;

SC <− 0;
IR10IR7: (Reg. file ←Add. Reg. File)

D2T4:
OutASel: IR(6,5);
MuxASel: 10;
RegSelP2A: IR(9,8);
FunSelY: 01;
SC <− 0;

D3: M[SP] ← SrcReg1, SP ← SP − 1 Data in register will be stored in stack register, and stack pointer will be decreased.

IR7: (data from Address Register File)
D3T4:
OutDSel: 01;
OutCSel: IR(6,5);
MuxASel: 10;
MuxCsel=0;
FunselAlu=0000;
Load: 1;

IR7': (data from Register File) D3T4:
OutASel: Ir(6,5);
MuxCsel=1;
FunSelALU: 0000;
Load: 1;

T5 Common
D3T5:
RegSelP2b: 100;
FunSelp2b: 10;
SC ← 0;

D4: SP ←SP + 1 , DestReg ← M[SP] Data will be pushed from stack register and written Source Register.

D4T4:

RegselP2B: 100;

FunSelP2B: 10;

IR10: (Source register is Address Register File)

D4T5:

OutDSel: 11;

Read : 1;

MuxBSel: 10;

RegSelP2A: IR(9,8);

FunSelP2A: 01;

SC ←0;

IR10': (Source register is Register File)

D4T5:

OutDSel: 11;

Read : 1;

MuxASel: 01;

RegSelP2B: IR(9,8);

FunSelY: 01;

SC ← 0;

D5: Destreg ← SrcReg1 + SrcReg2 Data from the source registers will be added and written in destination register.

IR10'IR7'IR4': (reg. file ← reg. file + reg. file)

D5T4:

OutASel: IR(6,5);

MuxCSel: 1;

OutBSel: IR(3,2);

MuxDSel: 0;

FunSelALU: 0100;

MuxASel: 11;

RegSelP2A: IR(9,8);

FunSelP2A: 01;
SC ←0;

IR10'IR7'IR4: (reg. file ← reg. file + address reg. file)
D5T4:
OutASel: IR(6,5);
MuxCSel: 1;
OutDSel: IR(3,2);
MuxDSel: 1;
FunSelALU: 0100;
MuxASel: 11;
RegSelP2A: IR(9,8);
FunSelP2A: 01;
SC ←0;

IR10'IR7 IR4': (reg. file ← address reg. file + reg. file)
D5T4:
OutCSel: IR(6,5);
MuxCSel: 1;
OutDSel: IR(3,2);
MuxDSel: 0;
FunSelALU: 0100;
MuxASel: 11;
RegSelP2A: IR(9,8);
FunSelP2A: 01;
SC ←0;

IR10'IR7 IR4: (reg. file ← address reg. file + address reg. file)
D5T4:
OutCSel: IR(6,5);
MuxCSel: 0;
OutBSel: IR(3,2);
MuxDSel: 0;
FunSelALU: 0100;
MuxASel: 11;
RegSelP2A: IR(9,8);
FunSelP2A: 01;

SC ←0;

IR10 IR7' IR4': (address reg. file ← reg. file + reg. file)
D5T4:
OutASel: IR(6,5);
MuxCSel: 1;
OutBSel: IR(3,2);
MuxDSel: 0;
FunSelALU: 0100;
MuxBSel: 11;
RegSelP2B: IR(9,8);
FunSelP2B: 01;
SC ←0;

IR10 IR7' IR4: (address reg. file ← reg. file + address reg. file)
D5T4:
OutASel: IR(6,5);
MuxCSel: 1;
OutDSel: IR(3,2);
MuxDSel: 1;
FunSelALU: 0100;
MuxBSel: 11;
RegSelP2B: IR(9,8);
FunSelP2B: 01;
SC ←0;

IR10 IR7 IR4': (address reg. file ← address reg. file + reg. file)
D5T4:
OutCSel: IR(6,5);
MuxCSel: 0;
OutBSel: IR(3,2);
MuxDSel: 0;
FunSelALU: 0100;
MuxBSel: 11;
RegSelP2B: IR(9,8);
FunSelP2B: 01;
SC ←0;

IR10 IR7 IR4: (address reg. file ← address reg. file + address reg. file)
D5T4:
OutCSel: IR(6,5);
MuxCSel: 0;
OutDSel: IR(3,2);
MuxDSel: 1;
FunSelALU: 0100;
MuxBSel: 11;
RegSelP2B: IR(9,8);
FunSelP2B: 01;
SC ←0;

D6: Destreg ← SrcReg1 - SrcReg2 Data will be subtracted between the source registers and written in destination register.

IR10'IR7'IR4': (reg. file ← reg. file - reg. file)
D6T4:
OutASel: IR(6,5);
MuxCSel: 1;
OutBSel: IR(3,2);
MuxDSel: 0;
FunSelALU: 0110;
MuxASel: 11;
RegSelP2A: IR(9,8);
FunSelP2A: 01;
SC ←0;

IR10'IR7'IR4: (reg. file ← reg. file - address reg. file)
D6T4:
OutASel: IR(6,5);
MuxCSel: 1;
OutDSel: IR(3,2);
MuxDSel: 1;
FunSelALU: 0110;

MuxASel: 11;
RegSelP2A: IR(9,8);
FunSelP2A: 01;
SC ←0;

IR10'IR7 IR4': (reg. file ← address reg. file - reg. file)
D6T4:
OutCSel: IR(6,5);
MuxCSel: 0;
OutBSel: IR(3,2);
MuxDSel: 0;
FunSelALU: 0110;
MuxASel: 11;
RegSelP2A: IR(9,8);
FunSelP2A: 01;
SC ←0;

IR10'IR7 IR4: (reg. file ← address reg. file - address reg. file)
D6T4:
OutCSel: IR(6,5);
MuxCSel: 0;
OutDSel: IR(3,2);
MuxDSel: 0;
FunSelALU: 0110;
MuxASel: 11;
RegSelP2A: IR(9,8);
FunSelP2A: 01;
SC ←0;

IR10 IR7' IR4': (address reg. file ← reg. file - reg. file)
D6T4:
OutASel: IR(6,5);
MuxCSel: 1;
OutBSel: IR(3,2);
MuxDSel: 0;
FunSelALU: 0110;
MuxBSel: 11;

RegSelP2B: IR(9,8);
FunSelP2B: 01;
SC ←0;

IR10 IR7' IR4: (address reg. file ← reg. file - address reg. file)
D6T4:
OutASel: IR(6,5);
MuxCSel: 1;
OutDSel: IR(3,2);
MuxDSel: 1;
FunSelALU: 0110;
MuxBSel: 11;
RegSelP2B: IR(9,8);
FunSelP2B: 01;
SC ←0;

IR10 IR7 IR4': (address reg. file leftarrow ← address reg. file - reg. file)
D6T4:
OutCSel: IR(6,5);
MuxCSel: 0;
OutBSel: IR(3,2);
MuxDSel: 0;
FunSelALU: 0110;
MuxBSel: 11;
RegSelP2B: IR(9,8);
FunSelP2B: 01;
SC ←0;

IR10 IR7 IR4: (address reg. file ← address reg. file - address reg. file)
D6T4:
OutCSel: IR(6,5);
MuxCSel: 0;
OutDSel: IR(3,2);
MuxDSel: 1;
FunSelALU: 0110;
MuxBSel: 11;
RegSelP2B: IR(9,8);

FunSelP2B: 01;
SC ←0;

D7: DestReg ← Destreg-1 Data in the destination register will be decremented by 1 and written again in destination register

IR10: (Source register is Address Register File)
D7T4:
RegSelP2B: IR(9.8);
FunSelP2B: 11;
SC ←0;

IR10': (Source register is Register File)
D7T4:
RegSelP2A: IR(9.8);
FunSelP2A: 11;
SC ←0;

D8: DestReg ← Destreg+1 Data in the destination register will be incremented by 1 and written again in destination register

IR10: (Source register is Address Register File)
D8T4:
RegSelP2B: IR(9.8);
FunSelP2B: 10;
SC ←0;

IR10': (Source register is Register File)
D8T4:
RegSelP2A: IR(9.8);
FunSelP2A: 10;
SC ←0;

D9: Destreg ← Srcreg1 and SrcReg2 IR10' IR7' IR4'(Register File←Register File) In this operation register sources will be compared by and operation and the result will be given to the destination register.

D9T4:

OutASel:IR(6,5); MuxCSel: 1;

OutBSel: IR(3,2);

MuxDSel: 0;

FunSelAlu: 0111

MuxASel: 11;

RegSelP2A: IR(9,8);

FunSelP2A: 01;

SC←0;

IR10' IR7' IR4(Register File←Register File and Address Register File)

D9T4:

OutDSel:IR(6,5); MuxCSel: 1;

OutCSel: IR(3,2);

MuxDSel: 1;

FunSelAlu: 0111

MuxASel: 11;

RegSelP2A: IR(9,8);

FunSelP2A: 01;

SC←0;

IR10' IR7 IR4'(Register File←Register File and Address Register File)

D9T4:

OutDSel:IR(6,5); MuxCSel: 1;

OutBSel: IR(3,2);

MuxDSel: 0;

FunSelAlu: 0111

MuxASel: 11;

RegSelP2A: IR(9,8);

FunSelP2A: 01;

SC←0;

IR10' IR7 IR4(Register File←Address Register File and Address Register File)

D9T4:

OutDSel:IR(6,5); MuxCSel: 1;

OutCSel: IR(3,2);

MuxDSel: 1;

FunSelAlu: 0111

MuxASel: 11;

RegSelP2A: IR(9,8);

FunSelP2A: 01;

SC←0;

IR10 IR7' IR4'(Address Register File←Register File and Register File)

D9T4:

OutASel:IR(6,5); MuxCSel: 1;

OutBSel: IR(3,2);

MuxDSel: 1;

FunSelAlu: 0111

MuxBSel: 11;

RegSelP2B: IR(9,8);

FunSelP2B: 01;

SC←0;

IR10 IR7' IR4(Address Register File←Register File and Address Register File)

D9T4:

OutASel:IR(6,5); MuxCSel: 1;

OutCSel: IR(3,2);

MuxDSel: 1;

FunSelAlu: 0111

MuxBSel: 11;

RegSelP2B: IR(9,8);

FunSelP2B: 01;

SC←0;

IR10 IR7 IR4'(Address Register File← Address Register File and Register File)

D9T4:
OutDSel:IR(6,5); MuxCSel: 1;
OutBSel: IR(3,2);
MuxDSel: 1;
FunSelAlu: 0111
MuxBSel: 11;
RegSelP2B: IR(9,8);
FunSelP2B: 01;
SC←0;

IR10 IR7 IR4(Address Register File← Address Register File and Address Register File)
D9T4:
OutDSel:IR(6,5); MuxCSel: 1;
OutCSel: IR(3,2);
MuxDSel: 1;
FunSelAlu: 0111
MuxBSel: 11;
RegSelP2B: IR(9,8);
FunSelP2B: 01;
SC←0;

D10: Destreg ←Srcreg1 OR Srcreg2 IR10' IR7' IR4'(Reg File←Not Register File)
In this operation, two source registers bit will be operated by or function then the result given to the destination register.
D10T4:
OutASel:IR(6,5); MuxCSel: 1;
OutBSel: IR(3,2);
MuxDSel: 0;
FunSelALU: 1000;
MuxASel: 11;
RegSelP2A: IR(9,8)
FunSelP2A: 01;
SC←0;

IR10' IR7' IR4(Register File← Register File or Address Register File)

D10T4:
OutASel:IR(6,5); MuxCSel: 1;
OutCSel: IR(3,2);
MuxDSel: 1;
FunSelALU: 1000;
MuxASel: 11;
RegSelP2A: IR(9,8)
FunSelP2A: 01;
SC←0;

IR10' IR7 IR4'(Register File← Address Register File or Register File)
D10T4:
OutDSel:IR(6,5); MuxASel: 10;
MuxCSel: 0;
OutBSel: IR(3,2);
MuxDSel: 0;
FunSelALU: 1000;
MuxASel: 11;
RegSelP2A: IR(9,8)
FunSelP2A: 01;
SC←0;

IR10' IR7 IR4(Register File← Address Register File or Address Register File)
D10T4:
OutDSel:IR(6,5); MuxASel: 10;
MuxCSel: 0;
OutCSel: IR(3,2);
MuxDSel: 1;
FunSelALU: 1000;
MuxASel: 11;
RegSelP2A: IR(9,8)
FunSelP2A: 01;
SC←0;

IR10 IR7' IR4'(Address Register File← Register File or Register File)
D10T4:
OutASel:IR(6,5); MuxCSel: 1;

OutBSel: IR(3,2);
MuxDSel: 0;
FunSelALU: 1000;
MuxBSel: 11;
RegSelP2B: IR(9,8)
FunSelP2B: 01;
SC←0;

IR10 IR7' IR4(Address Register File← Register File or Address Register File)
D10T4:
OutASel:IR(6,5); MuxCSel: 1;
OutCSel: IR(3,2);
MuxDSel: 1;
FunSelALU: 1000;
MuxBSel: 11;
RegSelP2B: IR(9,8)
FunSelP2B: 01;
SC←0;

IR10 IR7 IR4'(Address Register File←Address Register File or Register File)
D10T4:
OutDSel:IR(6,5); MuxASel: 10;
MuxCSel: 0;
OutCSel: IR(3,2);
MuxDSel: 1;
FunSelALU: 1000;
MuxBSel: 11;
RegSelP2B: IR(9,8)
FunSelP2B: 01;
SC←0;

IR10 IR7 IR4(Address Register File← Register File or Address Register File)
D10T4:
OutASel:IR(6,5); MuxASel: 10;
MuxCSel: 0;
OutBSel: IR(3,2);
MuxDSel: 1;

FunSelALU: 1000;
MuxBSel: 11;
RegSelP2B: IR(9,8)
FunSelP2B: 01;
SC←0;

D11: Destreg ← NOT Srcreg1 IR10' IR7'(Register File←Not Register File)
Source Register will be invert into its Not self then written to the destination register.
D11T4:
OutASel:IR(6,5); MuxCSel: 1;
FunSelALU: 0010;
MuxASel: 11;
RegSelP2A: IR(9,8)
FunSelP2A: 01;
SC←0;

IR10' IR7(Reg File←Not Address Register File)
D11T4:
OutDSel:IR(6,5); MuxASel: 10;
MuxCSel: 0;
FunSelAlu: 0010;
MuxASel: 11;
RegSelP2A: IR(9,8);
FunSelP2A: 01;
SC←0;

IR10 IR7'(Reg File←Not Address Register File)
D11T4:
OutDSel:IR(6,5); MuxCSel: 1;
FunSelAlu: 0010;
MuxASel: 11;
RegSelP2B: IR(9,8);
FunSelP2B: 01;
SC←0;

IR10 IR7(Reg File←Not Address Register File)

D11T4:
OutDSel:IR(6,5); MuxASel: 10;
MuxCSel: 0;
FunSelAlu: 0010;
MuxASel: 11;
RegSelP2B: IR(9,8);
FunSelP2B: 01;
SC←0;

D12: Destreg ← LSL Srcreg1 IR10' IR7'(Reg File←Not Register File)
Source Register 1 will be shifted left and written to the destination register.
D12T4:
OutASel:IR(6,5); MuxCSel: 1;
FunSelALU: 1010;
MuxASel: 11;
RegSelP2A: IR(9,8)
FunSelP2A: 01;
SC←0;

IR10' IR7(Reg File←Not Address Register File)
D12T4:
OutDSel:IR(6,5); MuxASel: 10;
MuxCSel: 0;
FunSelAlu: 1010;
MuxASel: 11;
RegSelP2A: IR(9,8);
SC←0;

IR10 IR7'(Address Register File←Not Register File)
D12T4:
OutASel:IR(6,5); MuxCSel: 1;
FunSelAlu: 1010;
MuxBSel: 11;
RegSelP2B: IR(9,8);
FunSelP2B: 01;
SC←0;

IR10 IR7(Address Register File←Not Register File)

D12T4:

OutASel:IR(6,5); MuxASel: 10;

FunSelAlu: 1010;

MuxBSel: 11;

RegSelP2B: IR(9,8);

FunSelP2B: 01;

SC←0;

D13: Destreg ← LSR Srcreg1

Source Register 1 will be shifted right and written to the destination register.

IR10' IR7'(Address Register File←Not Register File)

D13T4:

OutASel:IR(6,5); MuxCSel: 1;

FunSelALU: 1011;

MuxASel: 11;

RegSelP2A: IR(9,8)

FunSelP2A: 01;

SC←0;

IR10' IR7(Reg File←Not Address Register File)

D13T4:

OutDSel:IR(6,5); MuxASel: 10;

MuxCSel: 0;

FunSelAlu: 1011;

MuxASel: 11;

RegSelP2A: IR(9,8);

SC←0;

IR10 IR7'(Address Register File←Not Register File)

D13T4:

OutASel:IR(6,5); MuxCSel: 1;

FunSelAlu: 1011;

MuxBSel: 11;

RegSelP2B: IR(9,8);

FunSelP2B: 01;
SC←0;

IR10 IR7(Address Register File←Not Register File)
D13T4:
OutASel:IR(6,5); MuxASel: 10;
FunSelAlu: 1011;
MuxBSel: 11;
RegSelP2B: IR(9,8);
FunSelP2B: 01;
SC←0;

D14: PC ← Value We write the value to PC.

D14T4:
MuxBSel:01;
RegSelP2B: 001;
FunSelP2B: 01;
SC←0;

D15: If Z = 1 then PC ← Value We write the value to PC for condition Z = 1.
Z': SC←0;
Z: D15T4:
MuxBSel:01;
RegSelP2B: 001;
FunSelP2B: 01;
SC←0;

D16: If Z = 0 then PC ← Value We write the value to PC for condition Z = 0.

Z: SC←0;
Z': D16T4:
MuxBSel:01;
RegSelP2B: 001;
FunSelP2B: 01;

SC←0;

D17: M[SP] ← PC, SP ← SP - 1, PC ← Value

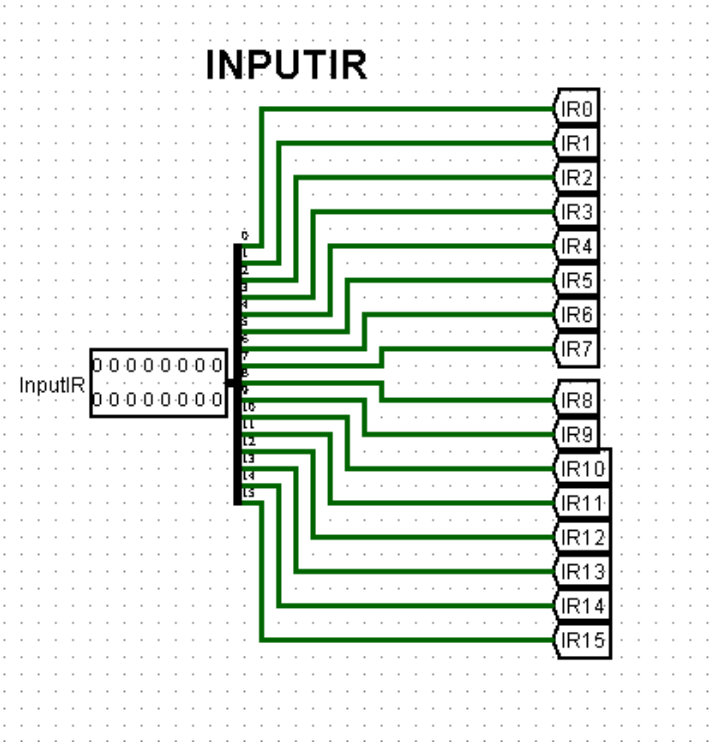
D17T4:
OutDSel:11;
OutCSel:00;
MuxASel:10;
MuxCSel:0;
FunSelAlu:0001;
Load: 1
D17T5:
RegSelP2B: 100;
FunSelP2B: 11;
D17T6:
RegSelP2B:001;
FunSelP2B: 01;
SC←0;

D18: SP ← SP + 1, PC ← M[SP]

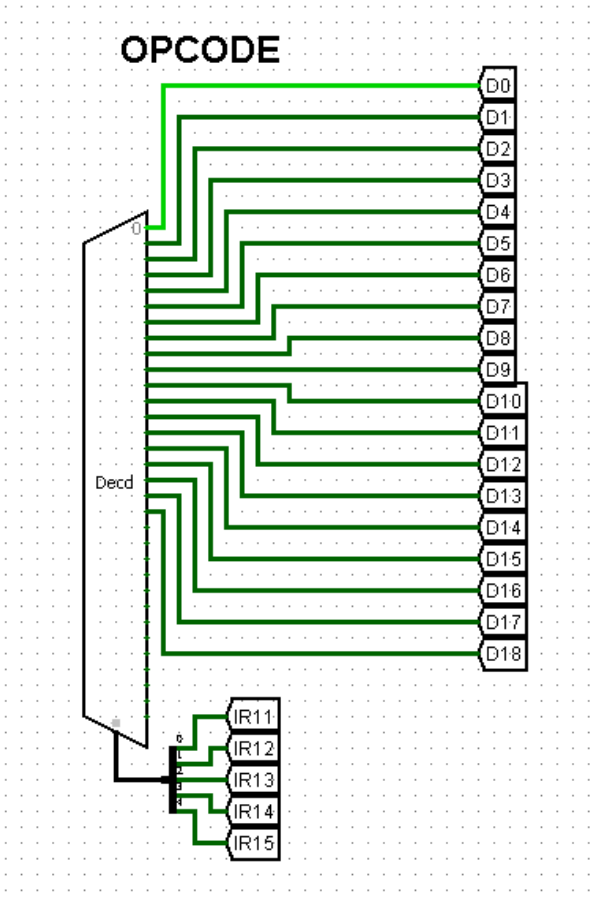
D18T4:
RegSelP2B: 100;
FunSelP2B: 10;
D18T5:
OutDSel:10;
Load: 1
MuxBSel:10;
RegSelP2B:001;
FunSelP2B: 01;
SC←0;

2.2 Implementations

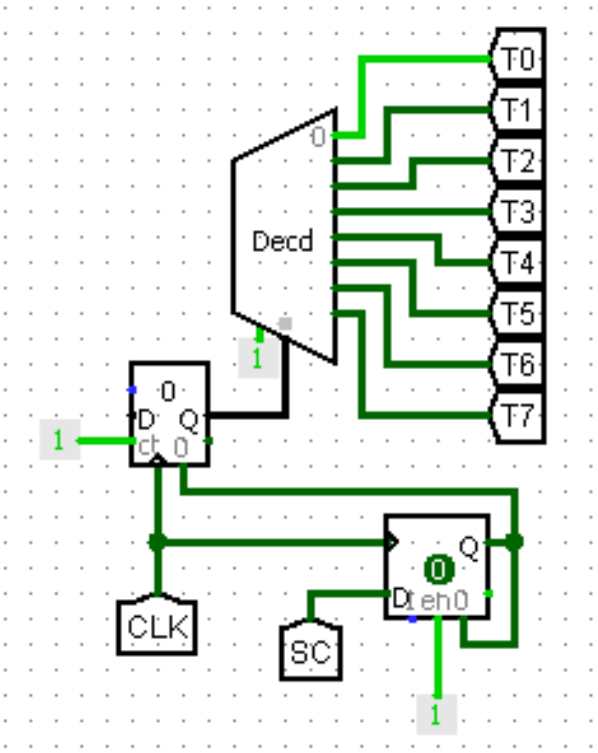
2.2.1 Instruction



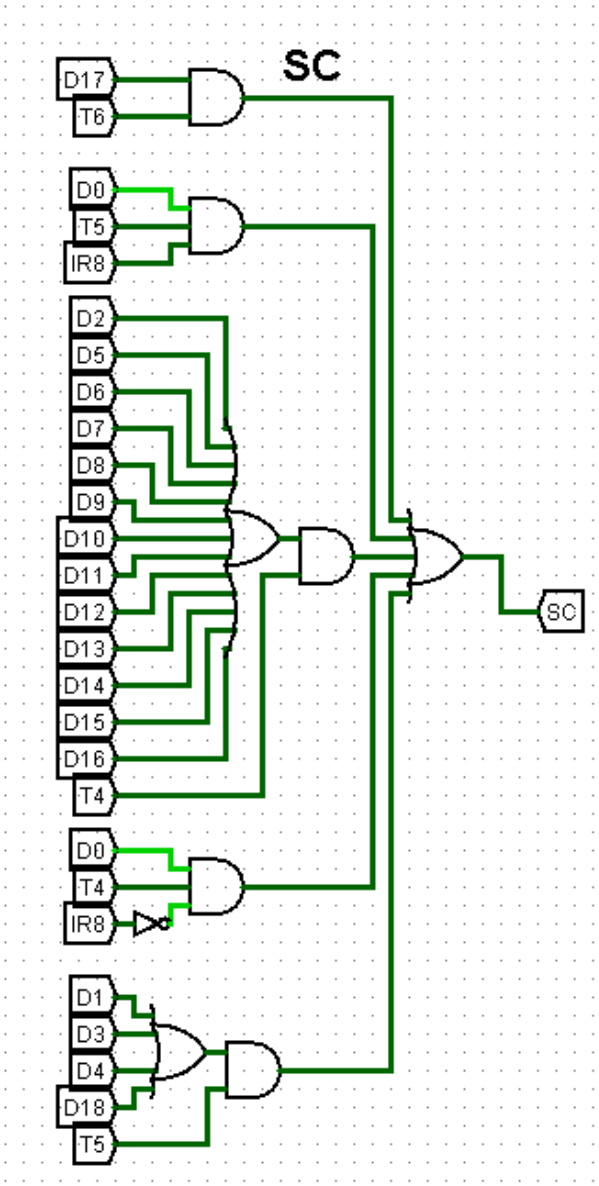
2.2.2 Decoding Opcode



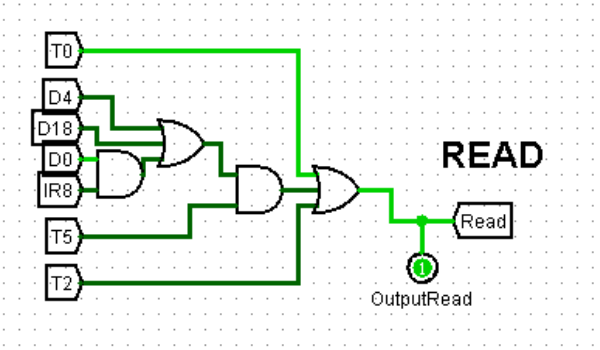
2.2.3 Time

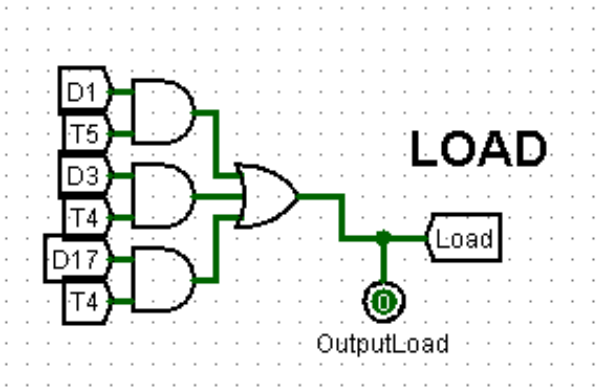


2.2.4 Sequence Counter

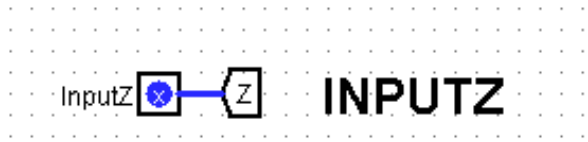


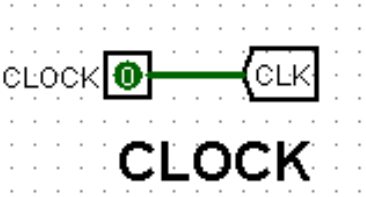
2.2.5 Read and load

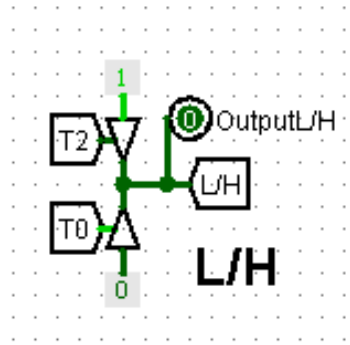


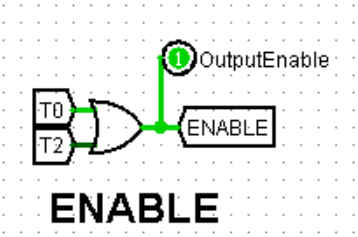


2.2.6 Clock L/H Input Z enable

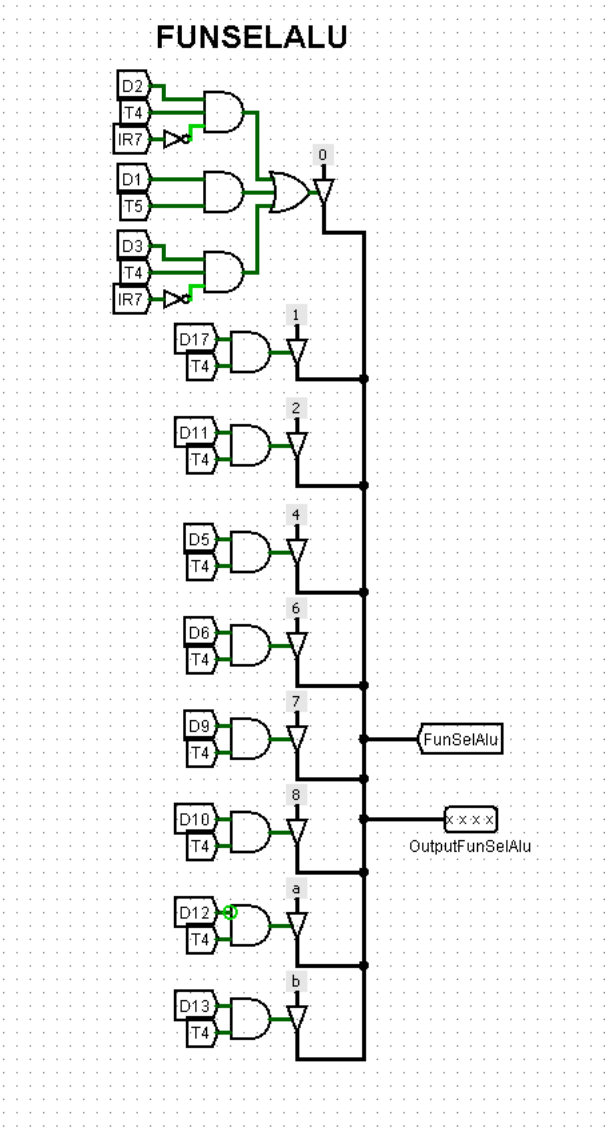


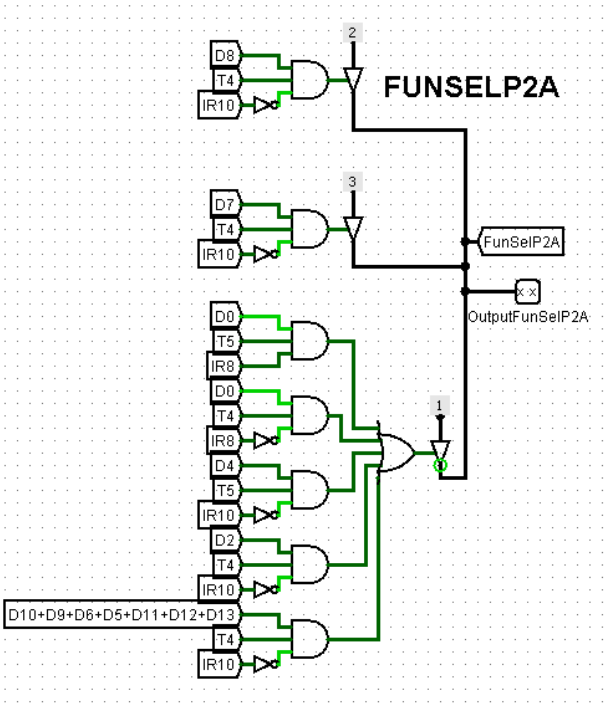


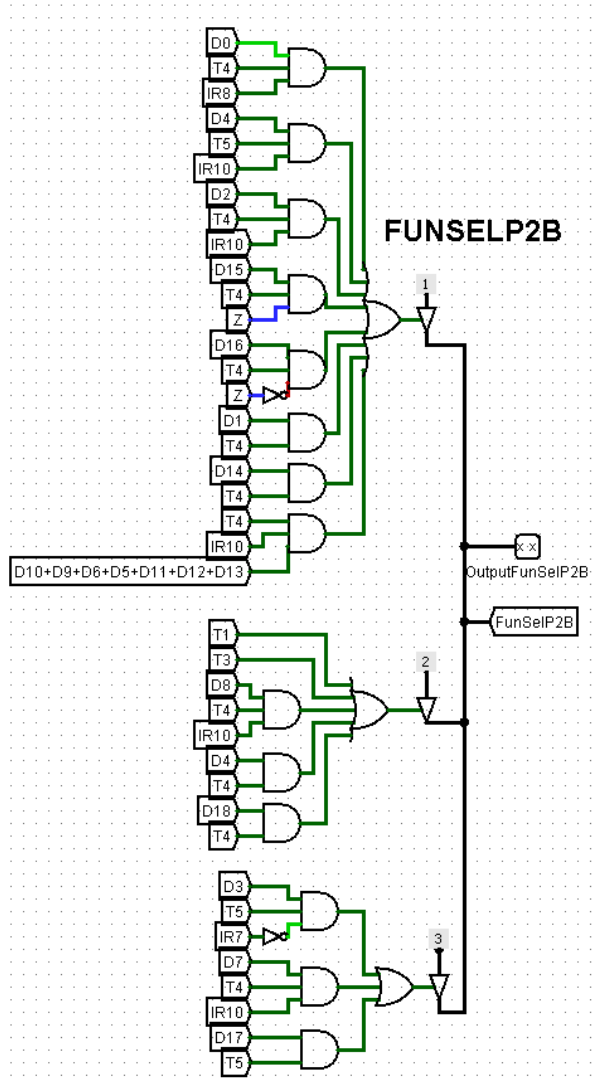


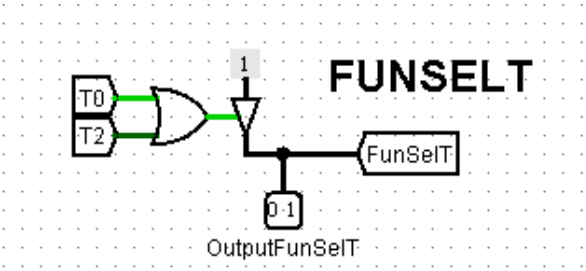


2.2.7 Funsels

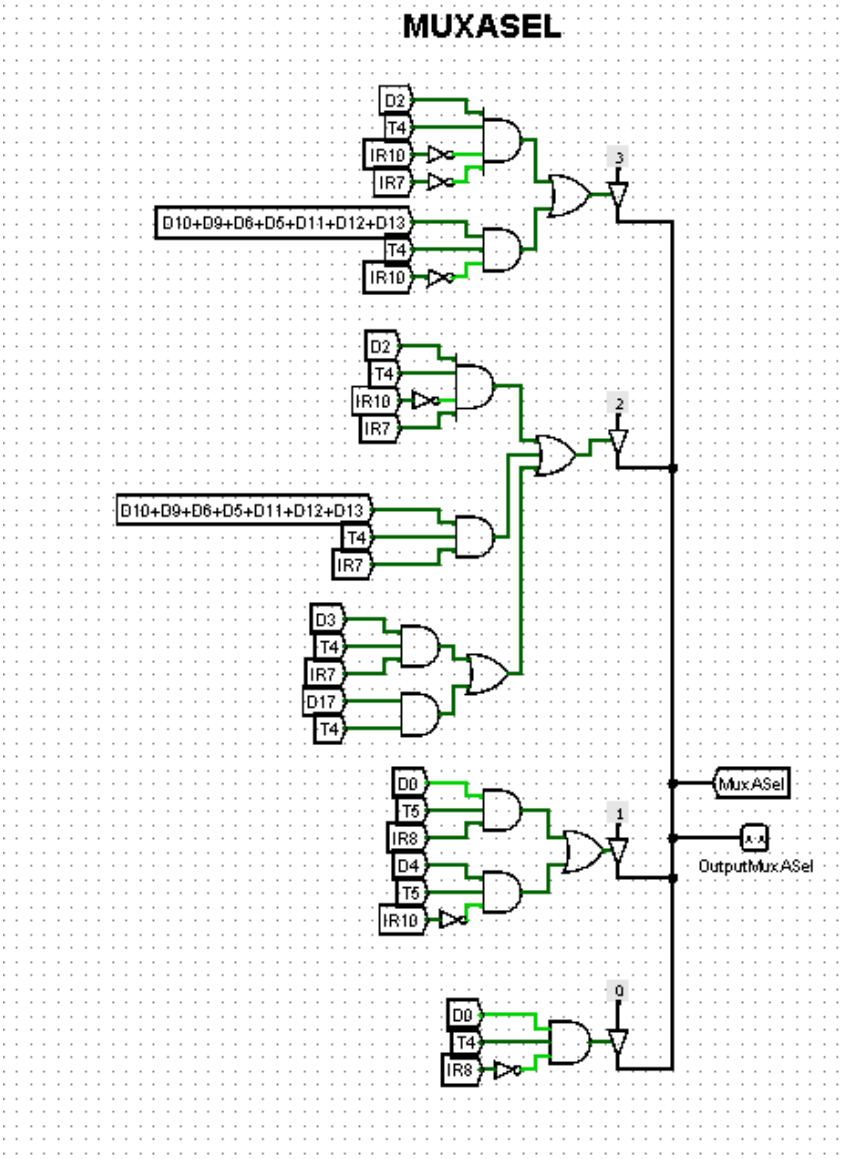

FUNSELALU
D2
T4
IR7
D1
T5
D3
T4
IR7
0
D17
T4
1
D11
T4
2
D5
T4
4
D6
T4
6
D9
T4
7
D10
T4
8
D12
T4
a
D13
T4
b
FunSelAlu
xxxx
OutputFunSelAlu

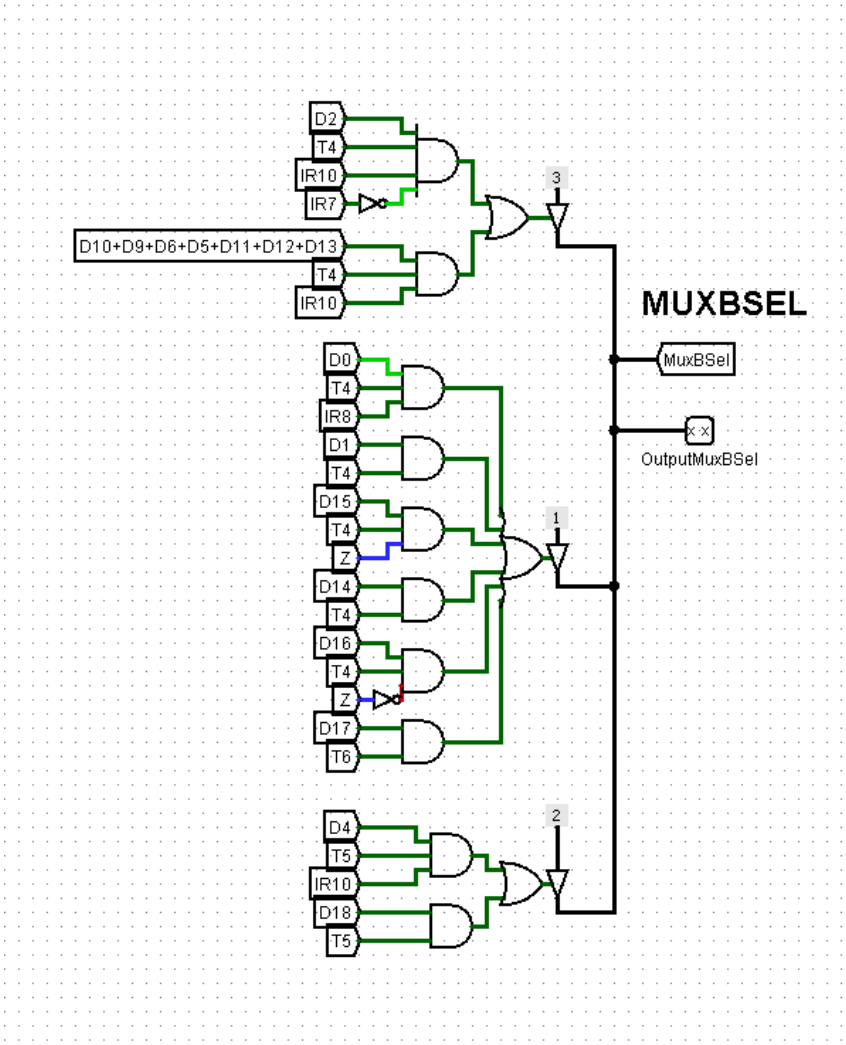

FUNSELP2A
D8
T4
IR10
2
D7
T4
IR10
3
FunSelP2A
OutputFunSelP2A
D0
T5
IR8
D0
T4
IR8
D4
T5
IR10
D2
T4
IR10
D10+D9+D6+D5+D11+D12+D13
T4
IR10
1

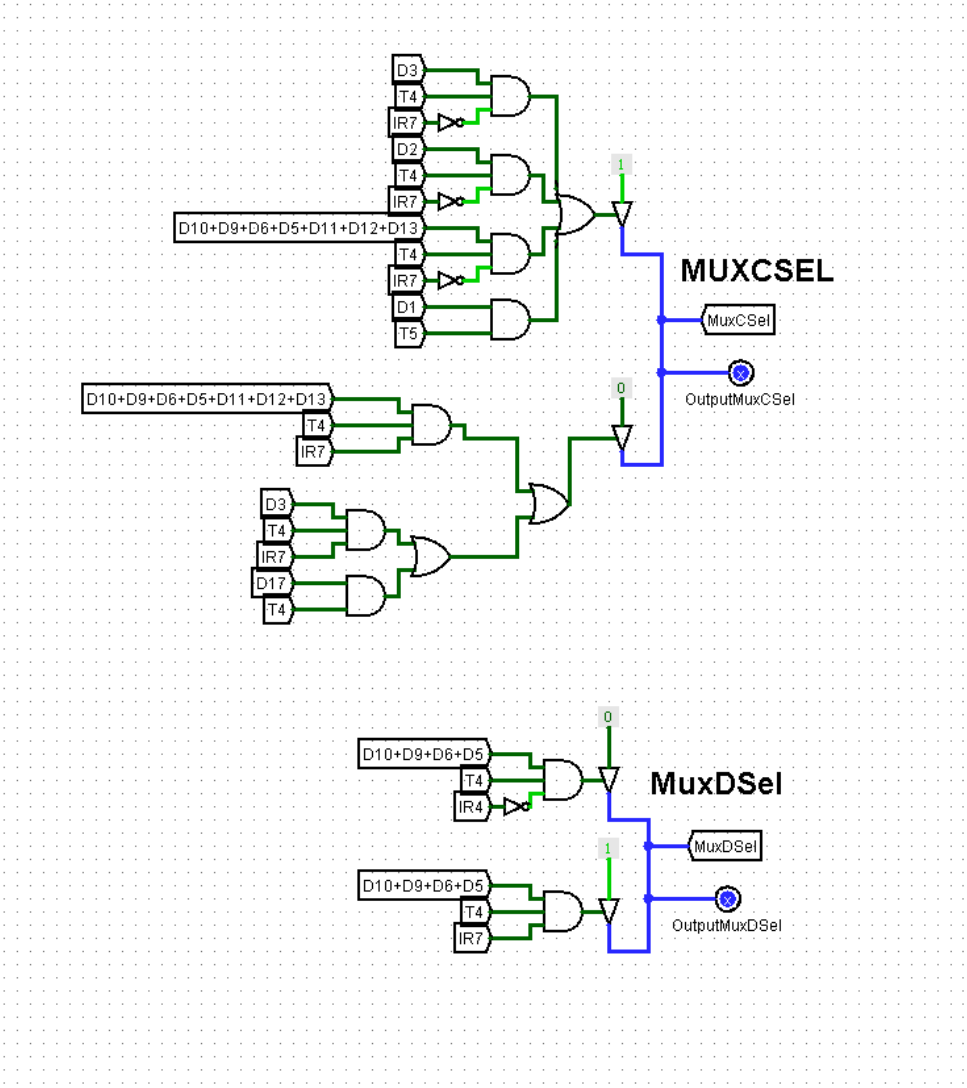

FUNSELP2B
D0
T4
IR8
D4
T5
IR10
D2
T4
IR10
D15
T4
Z
D16
T4
Z
D1
T4
D14
T4
T4
IR10
D10+D9+D6+D5+D11+D12+D13
1
OutputFunSelP2B
FunSelP2B
T1
T3
D8
T4
IR10
D4
T4
D18
T4
2
D3
T5
IR7
D7
T4
IR10
D17
T5
3



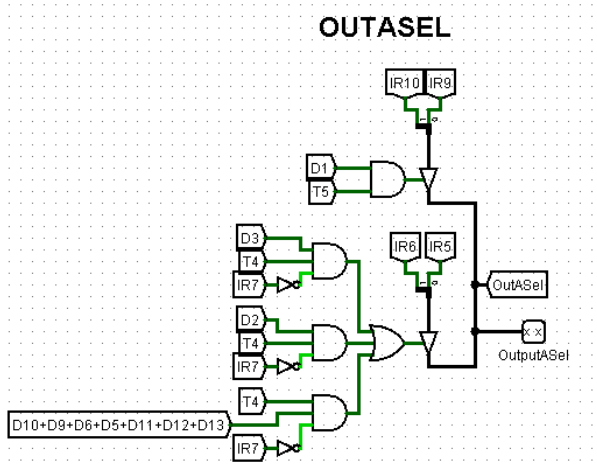
2.2.8 Mux Selections

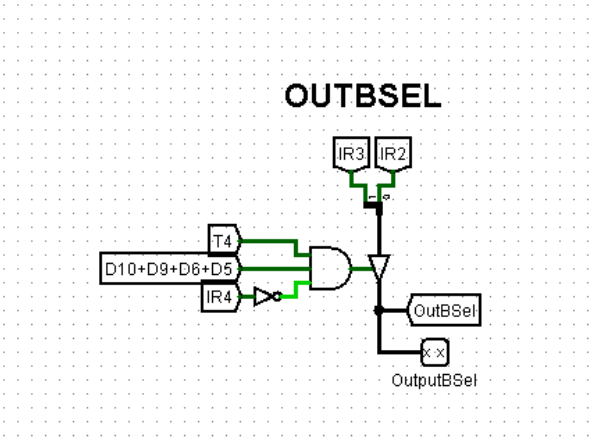


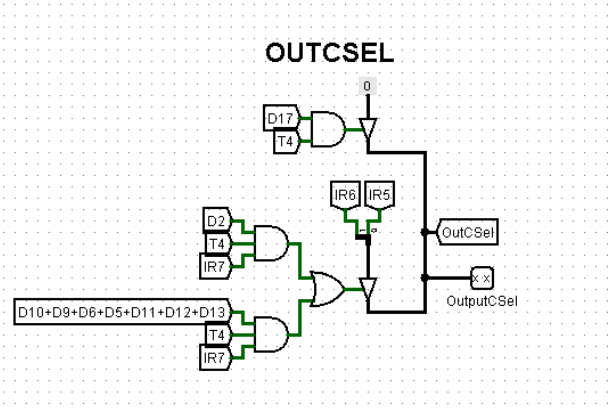

D2
T4
IR10
IR7
D10+D9+D6+D5+D11+D12+D13
T4
IR10
3
MUXBSEL
MuxBSel
x x
OutputMuxBSel
D0
T4
IR8
D1
T4
D15
T4
Z
D14
T4
D16
T4
Z
D17
T6
1
D4
T5
IR10
D18
T5
2

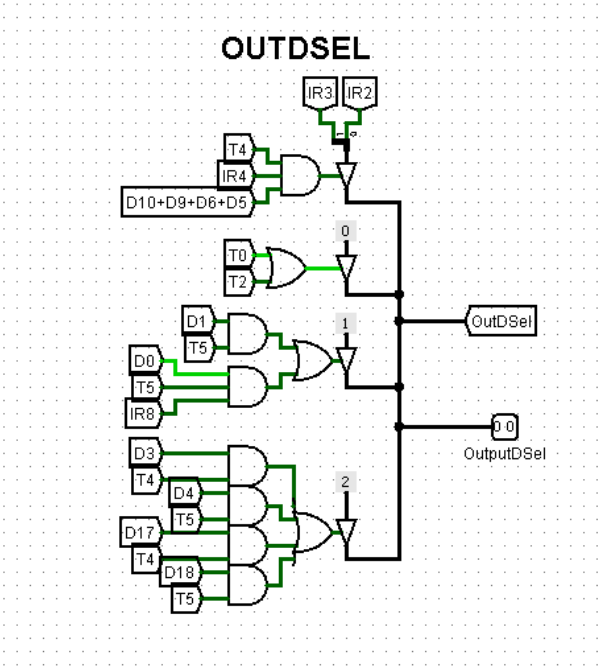


2.2.9 OutSelections

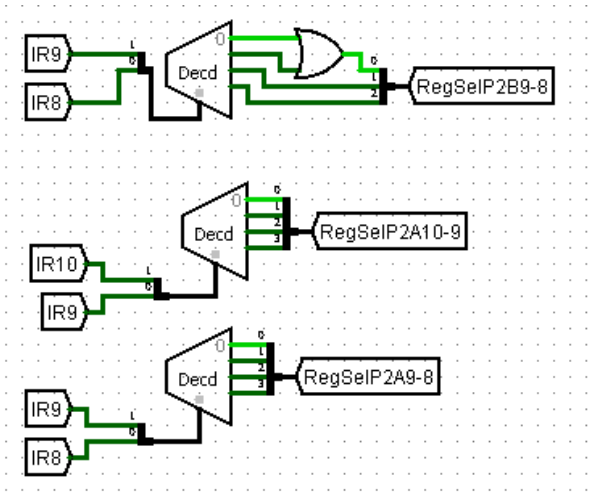


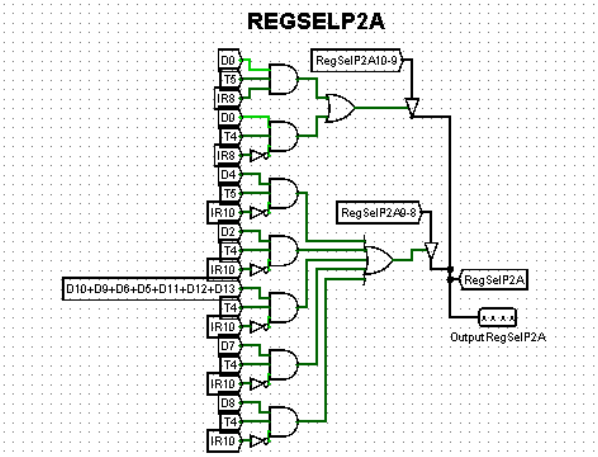

OUTBSEL
IR3
IR2
T4
D10+D9+D6+D5
IR4
OutBSel
OutputBSel

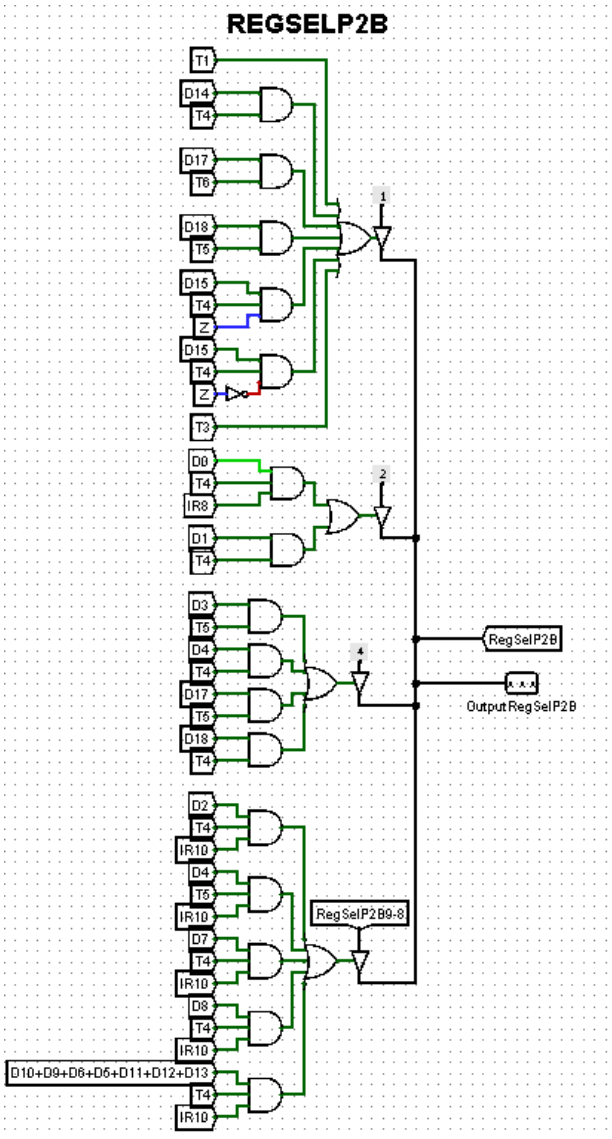

OUTCSEL
0
D17
T4
IR6
IR5
D2
T4
IR7
OutCSel
OutputCSel
D10+D9+D6+D5+D11+D12+D13
T4
IR7


OUTDSEL
IR3
IR2
T4
IR4
D10+D9+D6+D5
0
T0
T2
D1
T5
1
OutDSel
D0
T5
IR8
OutputDSel
D3
T4
D4
T5
2
D17
T4
D18
T5

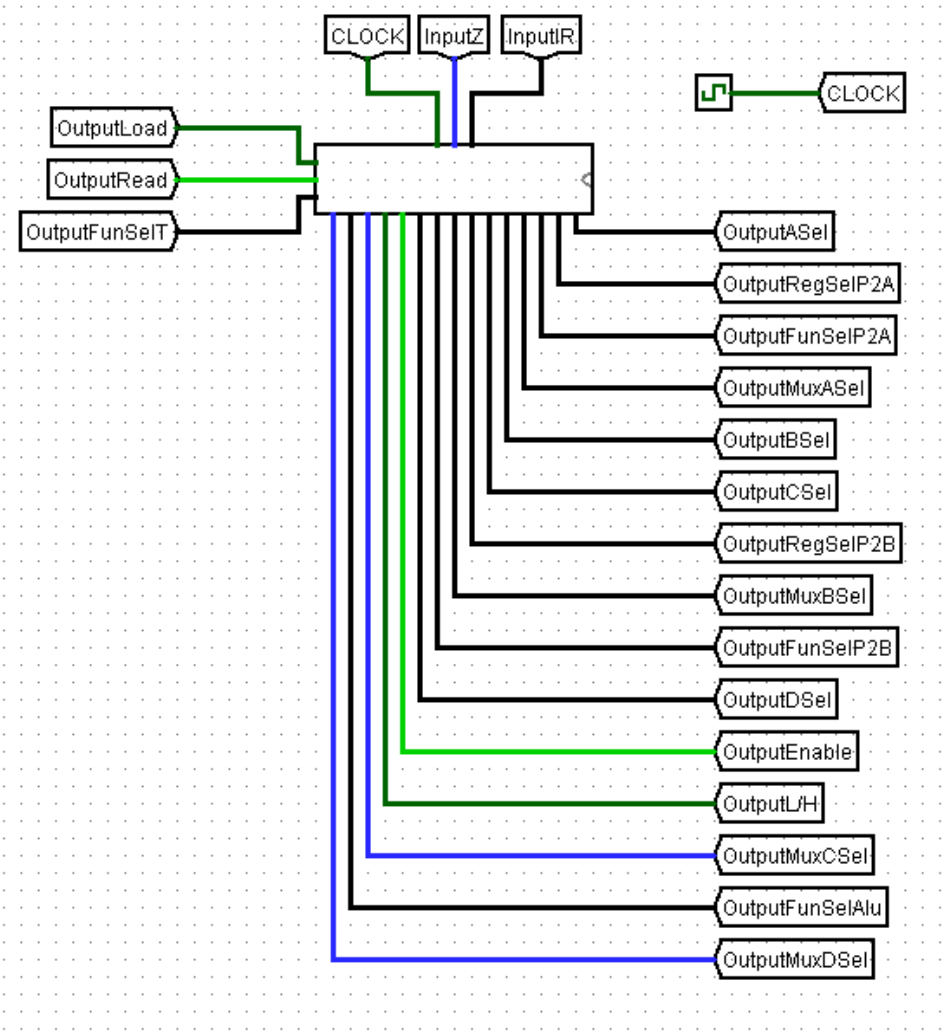
2.2.10 Register Selections






REGSELP2B
T1
D14
T4
D17
T6
D18
T5
D15
T4
Z
D15
T4
Z
T3
1
D0
T4
IR8
D1
T4
2
D3
T5
D4
T4
D17
T5
D18
T4
4
RegSelP2B
Output RegSelP2B
D2
T4
IR10
D4
T5
IR10
D7
T4
IR10
D8
T4
IR10
D10+D9+D6+D5+D11+D12+D13
T4
IR10
RegSelP2B9-8

2.2.11 Control unit



3 CONCLUSION

In this project we were asked to design a basic computer which has the functionalities that we were given in the project. Designing this project, we learned the basic functionalities and the structure of a computer. It was hard to understand those operations at first, but with the required researches we finally understood the mechanism, and implemented it.