ISTANBUL TECHNICAL UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT

BLG 222E
COMPUTER ORGANIZATION
PROJECT REPORT

PROJECT NO : 3
DUE DATE : 03.06.2020
GROUP NO : 36

GROUP MEMBERS:

150170087 : Sırrı Batuhan ÇOKSAK (Group Representative)

150170069 : Furkan Yusuf GÜRAY

150170039 : Fatih MURAT

150160142 : Ekin Zuhat YAŞAR

1 INTRODUCTION

In our last project, in order to control the structures we implemented in Project 1 and Project 2, we were expected to design a hardwire-control unit. Our design implemented and verified according to the tables shown in the figure-1, figure-2 and figure-3.

2 PROJECT

OPCODE (HEX)	SYMB	ADDRESSING MODE	DESCRIPTION
0x00	LD	IM, D	Rx ← Value (Value is described in Table 3)
0x01	ST	D	Value ← Rx
0x02	MOV	N/A	DESTREG ← SRCREG1
0x03	PSH	N/A	M[SP] ← Rx, SP ← SP - 1
0x04	PUL	N/A	SP ← SP + 1, Rx ← M[SP]
0x05	ADD	N/A	DESTREG ← SRCREG1 + SRCREG2
0x06	SUB	N/A	DESTREG ← SRCREG2 - SRCREG1
0x07	DEC	N/A	DESTREG ← SRCREG1 - 1
0x08	INC	N/A	DESTREG ← SRCREG1 + 1
0x09	AND	N/A	DESTREG ← SRCREG1 AND SRCREG2
0x0A	OR	N/A	DESTREG ← SRCREG1 OR SRCREG2
0x0B	NOT	N/A	DESTREG ← NOT SRCREG1
0x0C	LSL	N/A	DESTREG ← LSL SRCREG1
0xOD	LSR	N/A	DESTREG ← LSR SRCREG1
0x0E	BRA	IM	PC ← Value
0x0F	BEQ	IM	IF Z=1 THEN PC ← Value
0x10	BNE	IM	IF Z=0 THEN PC ← Value
0x11	CALL	IM	M[SP] ← PC, SP ← SP – 1, PC ← Value
0x12	RET	N/A	SP ← SP + 1, PC ← M[SP]

Figure-1

REGSEL	REGISTER
00	R0
01	R1
10	R2
11	R3

DESTREG/SRCREG1/SRCREG2	REGISTER
000	R0
001	R1
010	R2
011	R3
100	PC
101	PC
110	AR
111	SP

Figure-2

ADDRESSING MODE	MODE	SYMB	Value
0	Immediate	IM	ADDRESS Field
1	Direct	D	M[AR]

Figure-3

T0:

outDsel = pc (00,01), (Load pc as an adress to memory)

ram load = 1, (Enable ram load)

IR enable = 1, (Enable IR register)

IR Funsel = 11, (Set IR register to load)

IR L/H' = 0, (Set the input as high 8 bit)

PC_regsl = 001, (Select pc register)

PC_funsel = 10, (Select increment function)

T1:

outDsel = pc (00,01), (Load pc as an adress to memory)

ram load = 1, (Enable ram load)

IR enable = 1, (Enable IR register)

IR Funsel = 11, (Set IR register to load)

IR L/H' = 1, (Set the input as low 8 bit)

PC_regsl = 001, (Select pc register)

PC_funsel = 10, (Select increment function)

D0 = Rx<-Value

D0T2IR(8)':

MuxAsel = 00, (select Adress in MuxA)

R0/R1/R2/R3 regsel = F(IR(10),IR(9)), (select the corresponing register)

R0/R1/R2/R3 funsel = 01, (select load function)

---RESETTHECOUNTER

D0T2IR(8):

OutDsel = 10, (Select the address register as the address of the memory)

Ram load = 1, (enable ram load)

MuxAsel= 01, (Selecting the memory line from MuxA)

R0/R1/R2/R3 regsel = F(IR(10),IR(9)), (select the corresponing register)

R0/R1/R2/R3 funsel = 01, (select load function)

---RESETTHECOUNTER

D1 = Value<-Rx

D1T2IR(8):

OutBsel = IR(10,9), (Decode IR(10,9) and send the value to ALU using B input)

AluFunsel = 0001, (Moving B inside the alu)

OutDsel = 10, (Select the address register as the address of the memory)

Ram store = 1, (Enable ram store)

---RESETTHECOUNTER

D2 = DESTREG <- SRCREG1

IR(10)'IR(7)':

D2T2:

OutCsel = IR(6,5), (Outputing the source from OutC)

MuxAsel = 10, (Selecting the OutC from MuxA)

MuxCsel = 0, (Selecting the output of MuxA and outputing it from MuxC)

AluFun = 0000, (moving the input (output of MuxC) through ALU)

MuxBsel = 11, (selecting the line from ALU in MuxB)

PC/AR/SP Regsel = F(IR(9),IR(8)) (decode the destination and enable the corresponding register)

PC/AR/SP Funsel = 01, (Selecting the load function)

---RESETTHECOUNTER

IR(10)'IR(7):

D2T2:

OutBsel = IR(6,5), (Outputing the source from OutB)

AluFun = 0001, (moving the input (output of MuxB) through ALU)

MuxBsel = 11, (selecting the line from ALU in MuxB)

PC/AR/SP Regsel = F(IR(9),IR(8)) (decode the destination and enable the corresponding register)

PC/AR/SP Funsel = 01, (Selecting the load function)

---RESETTHECOUNTER

IR(10)IR(7)':

D2T2:

OutCsel = IR(6,5), (Outputing the source from OutC)

MuxAsel = 10, (Selecting the OutC from MuxA)

R0/R1/R2/R3 Regsel =F(IR(9),IR(8)) (decode the destination and enable the corresponding register)

R0/R1/R2/R3 Funsel = 01, (Selecting the load function)

---RESETTHECOUNTER

IR(10)IR(7):

D2T2:

OutBsel = IR(6,5), (Outputing the source from OutB)

AluFun = 0001, (moving the input (output of MuxB) through ALU)

MuxAsel = 11, (selecting the line from ALU in MuxA)

R0/R1/R2/R3 Regsel =F(IR(9),IR(8)) (decode the destination and enable the corresponding register)

R0/R1/R2/R3 Funsel = 01, (Selecting the load function)

---RESETTHECOUNTER

D3 = M[SP] <- Rx, SP <- SP - 1 (in instructions source1 = rx)

IR(7)':

D3T2:

OutBsel = IR(6,5), (Outputing the source from OutB)

AluFun = 0001, (moving the input (output of MuxB) through ALU)

Ram store = 1, (Enable ram store)

OutDsel = 11, (Select the SP as the address of the memory)

PC/AR/SP Regsel = 100, (enable the SP register)

PC/AR/SP Funsel = 11, (Selecting the decrement function)

---RESETTHECOUNTER

IR(7):

D3T2:

OutDsel = 11, (Select the SP as the address of the memory)

OutCsel = IR(6,5), (Outputing the source from OutC)

MuxAsel = 10, (selecting the line from OutC in MuxA)

MuxCsel = 0, (Selecting the output of MuxA and outputing it from MuxC)

AluFun = 0000, (moving the input (output of MuxC) through ALU)

---RESETTHECOUNTER

D4 = SP <- SP + 1, Rx <- M[SP] (in instructions destination = rx)

D4T2:

PC/AR/SP Regsel = 100 (enable the SP register)

PC/AR/SP Funsel = 10, (Selecting the decrement function)

D4T3:

IR(10)':

MuxAsel = 01, (Selecting the memory line from MuxA)

R0/R1/R2/R3 Regsel =F(IR(9),IR(8)) (decode the destination and enable the corresponding register)

R0/R1/R2/R3 Funsel = 01, (Selecting the load function)

---RESETTHECOUNTER

D4T3:

IR(10):

MuxBsel = 10, (selecting the line from memory in MuxB)

PC/AR/SP Regsel =F(IR(9),IR(8)) (decode the destination and enable the corresponding register)

PC/AR/SP Funsel = 01, (Selecting the load function)

---RESETTHECOUNTER

D5:

IR10’ IR7’ IR4’ T2

OutASel = IR(6,5), (Outputting SRCREG1 from OutA)

MuxCSel = 1, (Selecting OutA and outputting from MuxC)

OutBSel = IR(3,2), (Outputting SRCREG2 from OutB)

AluFunSel = 0100, (OutALU = A+B)

MuxASel = 11, (Selecting OutALU and sending from MuxA)

R0/R1/R2/R3 RegSel = F(IR(9,8)), (Selecting DESTREG to change it’s value)

R0/R1/R2/R3 FunSel = 01, (Selecting LOAD function)

---RESETTHECOUNTER

IR10’ IR7 IR4’ T2

(T2 part does DESTREG <- SRCREG1 so we can avoid changing MuxASel value twice in the same clock by doing DESTREG <- DESTREG + SRCREG2 which is almost the same

as the IR10'IR7'IR4'T2 above, only difference being new OutASel gets values from IR(9,8) for SRCREG1's new place.)

OutCSel = IR(6,5), (Outputting SRCREG1 from OutC)

MuxASel = 10 (Selecting OutC and outputting from MuxA)

R0/R1/R2/R3 RegSel = F(IR(9,8)), (Decoding destination and enabling dest reg)

R0/R1/R2/R3 FunSel = 01, (Selecting LOAD function)

IR10' IR7 IR4' T3

OutASel = IR(9,8), (Outputting SRCREG1 from OutA)

MuxCSel = 1, (Selecting OutA and outputting from MuxC)

OutBSel = IR(3,2), (Outputting SRCREG2 from OutB)

AluFunSel = 0100, (OutALU = A+B)

MuxASel = 11, (Selecting OutALU and sending from MuxA)

R0/R1/R2/R3 RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

R0/R1/R2/R3 FunSel = 01 (Selecting LOAD function)

---RESETTHECOUNTER

IR10 IR7' IR4' T2

OutASel = IR(6,5), (Outputting SRCREG1 from OutA)

MuxCSel = 1, (Selecting OutA and outputting from MuxC)

OutBSel = IR(3,2), (Outputting SRCREG2 from OutB)

AluFunSel = 0100, (OutALU = A+B)

MuxBSel = 11, (Selecting OutALU and sending from MuxB)

R0/R1/R2/R3 RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

R0/R1/R2/R3 FunSel = 01 (Selecting LOAD function)

---RESETTHECOUNTER

IR10 IR7 IR4' T2

OutCSel = IR(6,5), (Outputting SRCREG1 from OutC)

MuxASel = 10, (Selecting OutC and outputting from MuxA)

MuxCSel = 0, (Selecting MuxA and sending from MuxC)

OutBSel = IR(3,2), (Outputting SRCREG2 from OutB)

AluFunSel = 0100, (OutALU = A+B)

MuxBSel = 11, (Selecting OutALU and sending from MuxB)

R0/R1/R2/R3 RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

R0/R1/R2/R3 FunSel = 01 (Selecting LOAD function)

---RESETTHECOUNTER

D6:

IR10' IR7' IR4' T2

OutASel = IR(6,5), (Outputting SRCREG1 from OutA)

MuxCSel = 1, (Selecting OutA and outputting from MuxC)

OutBSel = IR(3,2), (Outputting SRCREG2 from OutB)

AluFunSel = 0100, (OutALU = A-B)

MuxASel = 11, (Selecting OutALU and sending from MuxA)

R0/R1/R2/R3 RegSel = F(IR(9,8)), (Selecting DESTREG to change it's value)

R0/R1/R2/R3 FunSel = 01, (Selecting LOAD function)

---RESETTHECOUNTER

IR10' IR7 IR4' T2

(T2 part does DESTREG <- SRCREG1 so we can avoid changing MuxASel value twice in the same clock by doing DESTREG <- DESTREG - SRCREG2 which is almost the same as the IR10'IR7'IR4'T2 above, only difference being new OutASel gets values from IR(9,8) for SRCREG1's new place.)

OutCSel = IR(6,5), (Outputting SRCREG1 from OutC)

MuxASel = 10 (Selecting OutC and outputting from MuxA)

R0/R1/R2/R3 RegSel = F(IR(9,8)), (Decoding destination and enabling dest reg)

R0/R1/R2/R3 FunSel = 01, (Selecting LOAD function)

IR10' IR7 IR4' T3

OutASel = IR(9,8), (Outputting SRCREG1 from OutA)

MuxCSel = 1, (Selecting OutA and outputting from MuxC)

OutBSel = IR(3,2), (Outputting SRCREG2 from OutB)

AluFunSel = 0100, (OutALU = A-B)

MuxASel = 11, (Selecting OutALU and sending from MuxA)

R0/R1/R2/R3 RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

R0/R1/R2/R3 FunSel = 01 (Selecting LOAD function)

---RESETTHECOUNTER

IR10 IR7' IR4' T2

OutASel = IR(6,5), (Outputting SRCREG1 from OutA)

MuxCSel = 1, (Selecting OutA and outputting from MuxC)

OutBSel = IR(3,2), (Outputting SRCREG2 from OutB)

AluFunSel = 0100, (OutALU = A-B)

MuxBSel = 11, (Selecting OutALU and sending from MuxB)

PC/AR/SP RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

PC/AR/SP FunSel = 01 (Selecting LOAD function)

---RESETTHECOUNTER

IR10 IR7 IR4' T2

OutCSel = IR(6,5), (Outputting SRCREG1 from OutC)

MuxASel = 10, (Selecting OutC and outputting from MuxA)

MuxCSel = 0, (Selecting MuxA and sending from MuxC)

OutBSel = IR(3,2), (Outputting SRCREG2 from OutB)

AluFunSel = 0100, (OutALU = A-B)

MuxBSel = 11, (Selecting OutALU and sending from MuxB)

PC/AR/SP RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

PC/AR/SP FunSel = 01 (Selecting LOAD function)

---RESETTHECOUNTER

D7:

IR10' IR7' T2

OutASel = IR(6,5), (Outputting SRCREG1 from OutA)

MuxCSel = 1, (Selecting OutA and outputting from MuxC)

AluFunSel = 0000, (Sending A from OutA)

MuxASel = 11, (Selecting OutA and sending from MuxA)

R0/R1/R2/R3 RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

R0/R1/R2/R3 FunSel = 01 (Selecting LOAD function)

IR10' IR7' T3

R0/R1/R2/R3 RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

R0/R1/R2/R3 FunSel = 11 (Selecting DECREMENT function)

---RESETTHECOUNTER

IR10' IR7 T2

OutCSel = IR(6,5), (Outputting SRCREG1 from OutC)

MuxASel = 10, (Selecting OutC and outputting from MuxA)

R0/R1/R2/R3 RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

R0/R1/R2/R3 FunSel = 01 (Selecting LOAD function)

IR10' IR7 T3

R0/R1/R2/R3 RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

R0/R1/R2/R3 FunSel = 11 (Selecting DECREMENT function)

---RESETTHECOUNTER

IR10 IR7' T2

OutASel = IR(6,5), (Outputting SRCREG1 from OutA)

MuxCSel = 1, (Selecting OutA and outputting from MuxC)

AluFunSel = 0000, (Sending A from OutA)

MuxBSel = 11, (Selecting OutA and sending from MuxA)

PC/AR/SP RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

PC/AR/SP FunSel = 01 (Selecting LOAD function)

IR10 IR7' T3

PC/AR/SP RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

PC/AR/SP FunSel = 11 (Selecting DECREMENT function)

---RESETTHECOUNTER

IR10 IR7 T2

OutCSel = IR(6,5), (Outputting SRCREG1 from OutC)

MuxASel = 1, (Selecting OutC and outputting from MuxA)

MuxCSel = 0, (Selecting MuxA and outputting from MuxC)

AluFunSel = 0000, (Sending A from OutA)

MuxBSel = 11, (Selecting OutA and sending from MuxA)

PC/AR/SP RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

PC/AR/SP FunSel = 01 (Selecting LOAD function)

IR10 IR7 T3

PC/AR/SP RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

PC/AR/SP FunSel = 11 (Selecting DECREMENT function)

---RESETTHECOUNTER

D8:

IR10' IR7' T2

OutASel = IR(6,5), (Outputting SRCREG1 from OutA)

MuxCSel = 1, (Selecting OutA and outputting from MuxC)

AluFunSel = 0000, (Sending A from OutA)

MuxASel = 11, (Selecting OutA and sending from MuxA)

R0/R1/R2/R3 RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

R0/R1/R2/R3 FunSel = 01 (Selecting LOAD function)

IR10' IR7' T3

R0/R1/R2/R3 RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

R0/R1/R2/R3 FunSel = 10 (Selecting INCREMENT function)

---RESETTHECOUNTER

IR10' IR7 T2

OutCSel = IR(6,5), (Outputting SRCREG1 from OutC)

MuxASel = 10, (Selecting OutC and outputting from MuxA)

R0/R1/R2/R3 RegSel = F(IR(9,8)) (Selecting DESTREG to change it's value)

R0/R1/R2/R3 FunSel = 01 (Selecting LOAD function)

IR10’ IR7 T3

R0/R1/R2/R3 RegSel = F(IR(9,8)) (Selecting DESTREG to change it’s value)

R0/R1/R2/R3 FunSel = 10 (Selecting INCREMENT function)

---RESETTHECOUNTER

IR10 IR7’ T2

OutASel = IR(6,5), (Outputting SRCREG1 from OutA)

MuxCSel = 1, (Selecting OutA and outputting from MuxC)

AluFunSel = 0000, (Sending A from OutA)

MuxBSel = 11, (Selecting OutA and sending from MuxA)

PC/AR/SP RegSel = F(IR(9,8)) (Selecting DESTREG to change it’s value)

PC/AR/SP FunSel = 01 (Selecting LOAD function)

IR10 IR7’ T3

PC/AR/SP RegSel = F(IR(9,8)) (Selecting DESTREG to change it’s value)

PC/AR/SP FunSel = 10 (Selecting INCREMENT function)

---RESETTHECOUNTER

IR10 IR7 T2

OutCSel = IR(6,5), (Outputting SRCREG1 from OutC)

MuxASel = 1, (Selecting OutC and outputting from MuxA)

MuxCSel = 0, (Selecting MuxA and outputting from MuxC)

AluFunSel = 0000, (Sending A from OutA)

MuxBSel = 11, (Selecting OutA and sending from MuxA)

PC/AR/SP RegSel = F(IR(9,8)) (Selecting DESTREG to change it’s value)

PC/AR/SP FunSel = 01 (Selecting LOAD function)

IR10 IR7 T3

PC/AR/SP RegSel = F(IR(9,8)) (Selecting DESTREG to change it’s value)

PC/AR/SP FunSel = 10 (Selecting INCREMENT function)

---RESETTHECOUNTER

D9:

IR10’ IR7’ IR4’

D9T2:

OutASel : (IR6,IR5) (Outputting SRCREG1 from OutA)

MuxCSel : 1 (Send OutA to ALU)

OutBSel : (IR3,IR2) (Outputting SRCREG2 from OutB)

ALUFunSel : 0111 (OutALU = A AND B)

MUXASel : 11 (Send A AND B to register file)

R0/R1/R2/R3 Register RegSel : (IR9,IR8) (Selecting DESTREG to change it’s value)

R0/R1/R2/R3 Register FunSel : 01 (Selecting LOAD function)

---RESETTHECOUNTER

IR10’ IR7 IR4’

(T2 part does DESTREG <- SRCREG1 so we can avoid changing MuxASel value twice in the same clock by doing DESTREG <- DESTREG AND SRCREG2 which is almost the same as the IR10’IR7’IR4’T2 above, only difference being new OutASel gets values from IR(9,8) for SRCREG1’s new place.)

D9T2:

OutCSel : (IR6,IR5) (Outputting SRCREG1 from OutC)

MuxASel : 10 (Send OutC to Register file)

R0/R1/R2/R3 Register Regsel : (IR9, IR8) (Decoding destination and enabling dest reg)

R0/R1/R2/R3 Register FunSel : 01 (Selecting LOAD function)

D9T3:

OutASel : (IR6,IR5) (Outputting SRCREG1 from OutA)

MuxCSel : 1 (Send OutA to ALU)

OutBSel : (IR3,IR2) (Outputting SRCREG2 from OutB)

ALU FunSel : 0111 (A AND B)

MuxASel : 11 (Send (A+B) to Register file)

R0/R1/R2/R3 Register RegSel : (IR9,IR8) (Selecting DESTREG to change it’s value)

R0/R1/R2/R3 Register FunSel : 01 (Load)

---RESETTHECOUNTER

IR10 IR7’ IR4’

D9T2:

OutASel : (IR6,IR5) (Outputting SRCREG1 from OutA)

MuxCSel : 1 (Send OutA to ALU)

OutBSel : (IR3,IR2) (Outputting SRCREG2 from OutB)

ALU FunSel : 0111 (OutALU = A AND B)

MuxBSel : 11 (A AND B to Addres Register)

PC/AR/SP Register RegSel : (IR9,IR8) (Selecting DESTREG to change it's value)

PC/AR/SP Register Funsel : 01 (Selecting LOAD function)

---RESETTHECOUNTER

IR10 IR7 IR4'

D9T2:

OutCSel : (IR6,IR5) (Outputting SRCREG1 from OutC)

MuxASel : 10 (Send OutC to MuxC)

MuxCSel : 0 (OutC to ALU A)

OutBSel : (IR3,IR2) (Outputting SRCREG2 from OutB)

ALU FunSel : 0111 (OutALU = A AND B)

MuxBSel : 11 (A AND B to Addres Register)

PC/AR/SP Register RegSel : (IR9,IR8) (Selecting DESTREG to change it's value)

PC/AR/SP Register FunSel : (Selecting LOAD function)

****

D10:

IR10' IR7' IR4'

D10T2:

OutASel : (IR6,IR5) (Outputting SRCREG1 from OutA)

MuxCSel : 1 (Send OutA to ALU)

OutBSel : (IR3,IR2) (OutC to ALU A)

ALUFunSel : 1000 (OutALU = A OR B)

MUXASel : 11 (Send A OR B to register file)

Register RegSel : (IR9,IR8) (Selecting DESTREG to change it's value)

Register FunSel : 01 (Selecting LOAD function)

---RESETTHECOUNTER

IR10’ IR7 IR4’

(T2 part does DESTREG <- SRCREG1 so we can avoid changing MuxASel value twice in the same clock by doing DESTREG <- DESTREG OR SRCREG2 which is almost the same as the IR10’IR7’IR4’T2 above, only difference being new OutASel gets values from IR(9,8) for SRCREG1’s new place.)

D10T2:

OutCSel : (IR6,IR5) (Outputting SRCREG1 from OutC)

MuxASel : 10 (Send OutC to Register file)

R0/R1/R2/R3 Register Regsel : (IR9, IR8) (Decoding destination and enabling dest reg)

R0/R1/R2/R3 Register FunSel : 01 (Selecting LOAD function)

D10T3:

OutASel : (IR6,IR5) (Outputting SRCREG1 from OutA)

MuxCSel : 1 (Send OutA to ALU)

OutBSel : (IR3,IR2) (Outputting SRCREG2 from OutB)

ALU FunSel : 1000 (OutALU = A OR B)

MuxASel : 11 (Send A OR B to register file)

R0/R1/R2/R3 Register RegSel : (IR9,IR8) (Selecting DESTREG to change it’s value)

R0/R1/R2/R3 Register FunSel : 01 (Selecting LOAD function)

---RESETTHECOUNTER

IR10 IR7’ IR4’

D9T2:

OutASel : (IR6,IR5) (Outputting SRCREG1 from OutA)

MuxCSel : 1 (Send OutA to ALU)

OutBSel : (IR3,IR2) (Outputting SRCREG2 from OutB)

ALU FunSel : 1000 (OutALU = A OR B)

MuxBSel : 11 (Send A OR B to register file)

PC/AR/SP Register RegSel : (IR9,IR8) (Selecting DESTREG to change it’s value)

PC/AR/SP Register Funsel : 01 (Selecting LOAD function)

---RESETTHECOUNTER

IR10 IR7 IR4’

D9T2:

OutCSel : (IR6,IR5) (Outputting SRCREG1 from OutC)

MuxASel : 10 (Send OutC to MuxC)

MuxCSel : 0 (OutC to ALU A)

OutBSel : (IR3,IR2) (Outputting SRCREG2 from OutB)

ALU FunSel : 1000 (OutALU = A OR B)

MuxBSel : 11 (Send A OR B to register file)

PC/AR/SP Register RegSel : (IR9,IR8) (Selecting DESTREG to change it’s value)

PC/AR/SP Register FunSel : (Selecting LOAD function)

****

D11:

IR10’ IR7’

D11T2:

OutASel : (IR6,IR5) (Outputting SRCREG1 from OutA)

MuxCSel : 1 (Select OutA)

ALU FunSel : 0010 (NOT A)

MuxASel : 11 (Select ALU Output)

R0/R1/R2/R3 Register RegSel : (IR9,IR8) (Selecting DESTREG to change it’s value)

R0/R1/R2/R3 Register FunSel : 01 (Selecting LOAD function)

---RESETTHECOUNTER

IR10’ IR7

(T2 part does DESTREG <- NOT SRCREG1 so we can avoid changing MuxASel value twice in the same clock by doing DESTREG <- NOT SRCREG1 which is almost the same as the IR10'IR4T2 above, only difference being new OutASel gets values from IR(9,8) for SRCREG1's new place.)

D11T2:

OutCSel : (IR6,IR5)	(Outputting SRCREG1 from OutC)
MuxASel : 10	(Select OutCSel)
R0/R1/R2/R3 Register RegSel : (IR9,IR8)	(Decoding destination and enabling dest reg)
R0/R1/R2/R3 Register FunSel : 01	(Selecting LOAD function)

D11T3:

OurASel : (IR9,IR8)	(Outputting SRCREG1 from OutA)
MuxCSel : 1	(Select OutA)
ALU FunSel : 0010	(NOT A)
MuxASel : 11	(Select ALU Output)
R0/R1/R2/R3 Register RegSel : (IR9,IR8)	(Selecting DESTREG to change it's value)
R0/R1/R2/R3 Register FunSel : 01	(Selecting LOAD function)

---RESETTHECOUNTER

IR10 IR7'

D11T2:

OutASel : (IR6,IR5)	(Outputting SRCREG1 from OutA)
MuxCSel : 1	(Select OutASel)
ALU FunSel : 0010	(NOT A)
MuxBSel : 11	(Select ALU Output)
PC/AR/SP Register RegSel : (IR9,IR8)	(Selecting DESTREG to change it's value)
PC/AR/SP Register FunSel : 01	(Selecting LOAD function)

---RESETTHECOUNTER

IR10 IR7

D1T2:

OutCSel : (IR6,IR5)	(Outputting SRCREG1 from OutC)
MuxASel : 10	(Select OutCSel)
ALU FunSel : 0010	(NOT A)
MuxBSel : 11	(Select ALU Output)
PC/AR/SP Register RegSel : (IR9,IR8)	(Selecting DESTREG to change it's value)
PC/AR/SP Register FunSel : 01	(Selecting LOAD function)

---RESETTHECOUNTER

****

D12:

IR10' IR7'

D11T2:

OutASel : (IR6,IR5)	(Outputting SRCREG1 from OutA)
MuxCSel : 1	(Select OutA)
ALU FunSel : 1010	(LSL A)
MuxASel : 11	(Select ALU Output)
R0/R1/R2/R3 Register RegSel : (IR9,IR8)	(Selecting DESTREG to change it's value)
R0/R1/R2/R3 Register FunSel : 01	(Selecting LOAD function)

---RESETTHECOUNTER

IR10' IR7

(T2 part does DESTREG <- LSL SRCREG1 so we can avoid changing MuxASel value twice in the same clock by doing DESTREG <- LSL SRCREG1 which is almost the same as the IR10'IR4T2 above, only difference being new OutASel gets values from IR(9,8) for SRCREG1's new place.)

D11T2:

OutCSel : (IR6,IR5)	(Outputting SRCREG1 from OutC)
MuxASel : 10	(Select OutCSel)
R0/R1/R2/R3 Register RegSel : (IR9,IR8)	(Decoding destination and enabling dest reg)
R0/R1/R2/R3 Register FunSel : 01	(Selecting LOAD function)

D11T3:

OurASel : (IR9,IR8)	(Outputting SRCREG1 from OutA)
MuxCSel : 1	(Select OutA)
ALU FunSel : 1010	(LSL A)
MuxASel : 11	(Select ALU Output)
R0/R1/R2/R3 Register RegSel : (IR9,IR8)	(Selecting DESTREG to change it's value)
R0/R1/R2/R3 Register FunSel : 01	(Selecting LOAD function)

---RESETTHECOUNTER

IR10 IR7'

D11T2:

OutASel : (IR6,IR5)	(Outputting SRCREG1 from OutA)
MuxCSel : 1	(Select OutASel)
ALU FunSel : 1010	(LSL A)
MuxBSel : 11	(Select ALU Output)
PC/AR/SP Register RegSel : (IR9,IR8)	(Selecting DESTREG to change it's value)
PC/AR/SP Register FunSel : 01	(Selecting LOAD function)

---RESETTHECOUNTER

IR10 IR7

D11T2:

OutCSel : (IR6,IR5)	(Outputting SRCREG1 from OutC)
MuxASel : 10	(Select OutCSel)
ALU FunSel : 1010	(LSL A)
MuxBSel : 11	(Select ALU Output)
PC/AR/SP Register RegSel : (IR9,IR8)	(Selecting DESTREG to change it's value)
PC/AR/SP Register FunSel : 01	(Selecting LOAD function)

****

D13:

IR10’ IR7’

D11T2:

OutASel : (IR6,IR5)	(Outputting SRCREG1 from OutA)
MuxCSel : 1	(Select OutA)
ALU FunSel : 1011	(LSR A)
MuxASel : 11	(Select ALU Output)
Register RegSel : (IR9,IR8)	(Selecting DESTREG to change it’s value)
Register FunSel : 01	(Selecting LOAD function)

---RESETTHECOUNTER

IR10’ IR7

(T2 part does DESTREG <- LSR SRCREG1 so we can avoid changing MuxASel value twice in the same clock by doing DESTREG <- LSR SRCREG1 which is almost the same as the IR10’IR4T2 above, only difference being new OutASel gets values from IR(9,8) for SRCREG1’s new place.)

D11T2:

OutCSel : (IR6,IR5)	(Outputting SRCREG1 from OutC)
MuxASel : 10	(Select OutCSel)
R0/R1/R2/R3 Register RegSel : (IR9,IR8)	(Decoding destination and enabling dest reg)
R0/R1/R2/R3 Register FunSel : 01	(Selecting LOAD function)

D11T3:

OurASel : (IR9,IR8)	(Outputting SRCREG1 from OutA)
MuxCSel : 1	(Select OutA)
ALU FunSel : 1011	(LSR A)
MuxASel : 11	(Select ALU Output)
R0/R1/R2/R3 Register RegSel : (IR9,IR8)	(Selecting DESTREG to change it’s value)
R0/R1/R2/R3 Register FunSel : 01	(Selecting LOAD function)

---RESETTHECOUNTER

IR10 IR7’

D11T2:

OutASel : (IR6,IR5) (Outputting SRCREG1 from OutA)

MuxCSel : 1 (Select OutASel)

ALU FunSel : 1011 (LSR A)

MuxBSel : 11 (Select ALU Output)

PC/AR/SP Register RegSel : (IR9,IR8) (Selecting DESTREG to change it's value)

PC/AR/SP Register FunSel : 01 (Selecting LOAD function)

---RESETTHECOUNTER

IR10 IR7

D1T2:

OutCSel : (IR6,IR5) (Outputting SRCREG1 from OutC)

MuxASel : 10 (Select OutCSel)

ALU FunSel : 1011 (LSR A)

MuxBSel : 11 (Select ALU Output)

PC/AR/SP Register RegSel : (IR9,IR8) (Selecting DESTREG to change it's value)

PC/AR/SP Register FunSel : 01 (Selecting LOAD function)

---RESETTHECOUNTER

D14(0x0E)(PC <- Value)

T2:

MuxBSel : 01 (Sends value to PC/PC/AR/SP Register)

PC/PC/AR/SP RegSel : 001 (PC)

PC/PC/AR/SP RegSel : 01 (Load)

TIME RESET

****

D15(0x0F)(IF Z=1 THEN PC <- Value)

T2(Z):

MuxBSel : 01 (Sends value to PC/PC/AR/SP Register)

PC/PC/AR/SP RegSel : 001 (PC)

PC/PC/AR/SP RegSel : 01 (Load)

TIME RESET

****

D16(0x10)(IF Z=0 THEN PC <- Value)

T2(Z'):

MuxBSel : 01 (Sends value to PC/PC/AR/SP Register)

PC/PC/AR/SP RegSel : 001 (PC)

PC/PC/AR/SP RegSel : 01 (Load)

TIME RESET

****

D17(0x11)(M[SP] <- PC, SP <- SP - 1, PC <- Value)

T2(M[SP] <- PC, SP <- SP - 1):

OutCSel : 00 (PC)

OutDSel : 11 (SP)

MuxASel : 10 (Sends OutCSel to R0/R1/R2/R3 Register)

MuxCSel : 0 (Sends output of MuxA to ALU)

ALU FunSel = 0000 (A) (Sends A input to RAM)

RAM Store : 1 (Loads PC to M[SP])

PC/PC/AR/SP FunSel : 11 (Decrement)

PC/PC/AR/SP Regsel : 100 (SP)

T3(PC <- Value):

MuxBSel : 01 (Sends value to PC/PC/AR/SP Register)

PC/PC/AR/SP RegSel : 001 (PC)

PC/PC/AR/SP RegSel : 01 (Load)

TIME RESET

****

D18(0x12)(SP <- SP + 1, PC <- M[SP])

T2(SP <- SP + 1):

PC/PC/AR/SP RegSel : 100 (SP)

PC/PC/AR/SP : 10 (Increment)

T3(PC <- M[SP]):

PC/PC/AR/SP OutSel : 11 (SP)

RAM Load : 1

MuxBSel : 10 (Sends M[SP] to PC/PC/AR/SP Register)

PC/PC/AR/SP RegSel : 00 (PC)

PC/PC/AR/SP FunSel : 01 (Load) (Loads M[SP] to PC)

TIME RESET

****

3 RESULT

During the project, when we combined the fetch decode execute parts we did above, the following circuit occurred on figure 4.

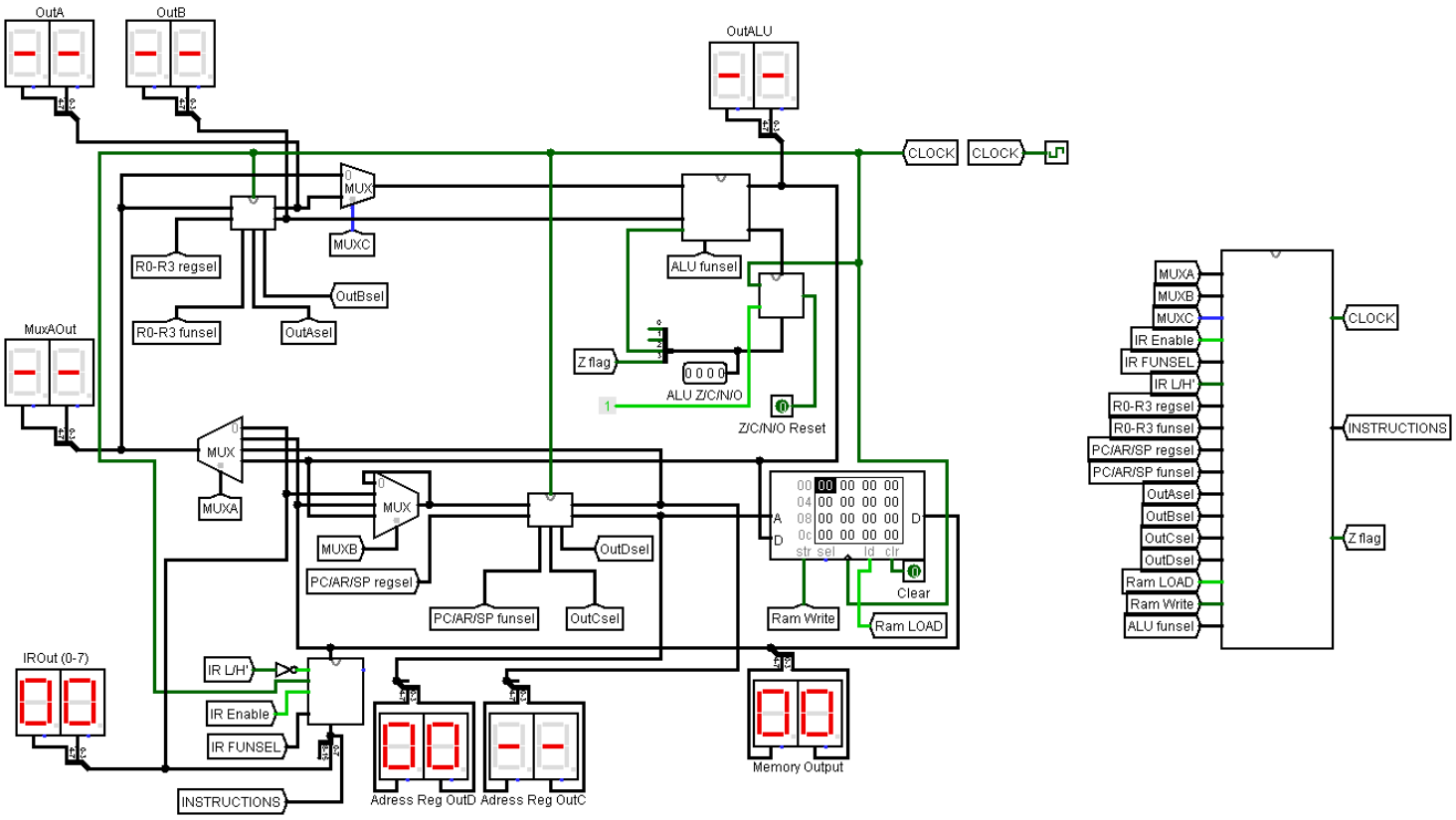

Figure-4 The Whole Of Project

4 DISCUSSION

In this final project, we were expected to implement hardwire-control unit using our project-1 and project-2 implementations that we did before according to given rules. There were two different type of instruction format. According to the incoming instructions, operations in the Figure-1 are served succesfully by our design. While we were implementing the project-2, we were not that aware of what is going on. However, with this final project, we completely understood the projects that we implemented before and learnt the logic behind the computer organization.

5 CONCLUSION

Understanding how complex hardware-control took a long time. We spent hours before being able to really get going in our design. Although we had 19 OPCODE operations some of them took only a couple changes from the others, which made us think we could simplify how large our hardware-control unit design was. Even after doing simplifications, what we had was enormous compared to projects 1 and 2, which made us appreciate even larger-scope projects.