CS224

Lab 04

Section No. 4

Maryam Shahid

21801344

Part 1.

b)

Location	Machine Instruction	Assembly Language Equivalent
00	0x20020005	addi $v0, $zero, 5
04	0x2003000c	addi $v1, $zero, 12
08	0x2067fff7	addi $a3, $v1, -9
0c	0x00e22025	or $a0, $a3, $v0
10	0x00642824	and $a1, $v1, $a0
14	0x00a42820	add $a1, $a1, $a0
18	0x10a7000a	beq $a1, $a3, 10
1c	0x0064202a	slt $a0, $v1, $a0
20	0x10800001	beq $a0, $zero, 1
24	0x20050000	addi $a0, $zero, 0
28	0x00e2202a	slt $a0, $a3, $v0
2c	0x00853820	add $a3, $a0, $a1
30	0x00e23822	sub $a3, $a3, $v0
34	0xac670044	sw $a3, 68 ($v1)
38	0x8c020050	lw $v0, 80 ($zero)
3c	0x08000011	j 0x0000011
40	0x20020001	addi $v0, $zero, 1
44	0xac020054	sw $v0, 84 ($zero)
48	0x8000012	j 0x0000012

c) RTL Expressions for:

ble:

```
IM[PC]
if(RF[rt] = RF[rt])
    PC  ← 4 + SignExt(immed) << 2
elseif(RF[rt] < RF[rs])
    PC  ← 4 + SignExt(immed) << 2
else
    PC  ← PC + 4
```

subi:

```
IM[PC]
RF[rt] ← RF[rs] – SignExt(immed)
PC  ← PC + 4
```

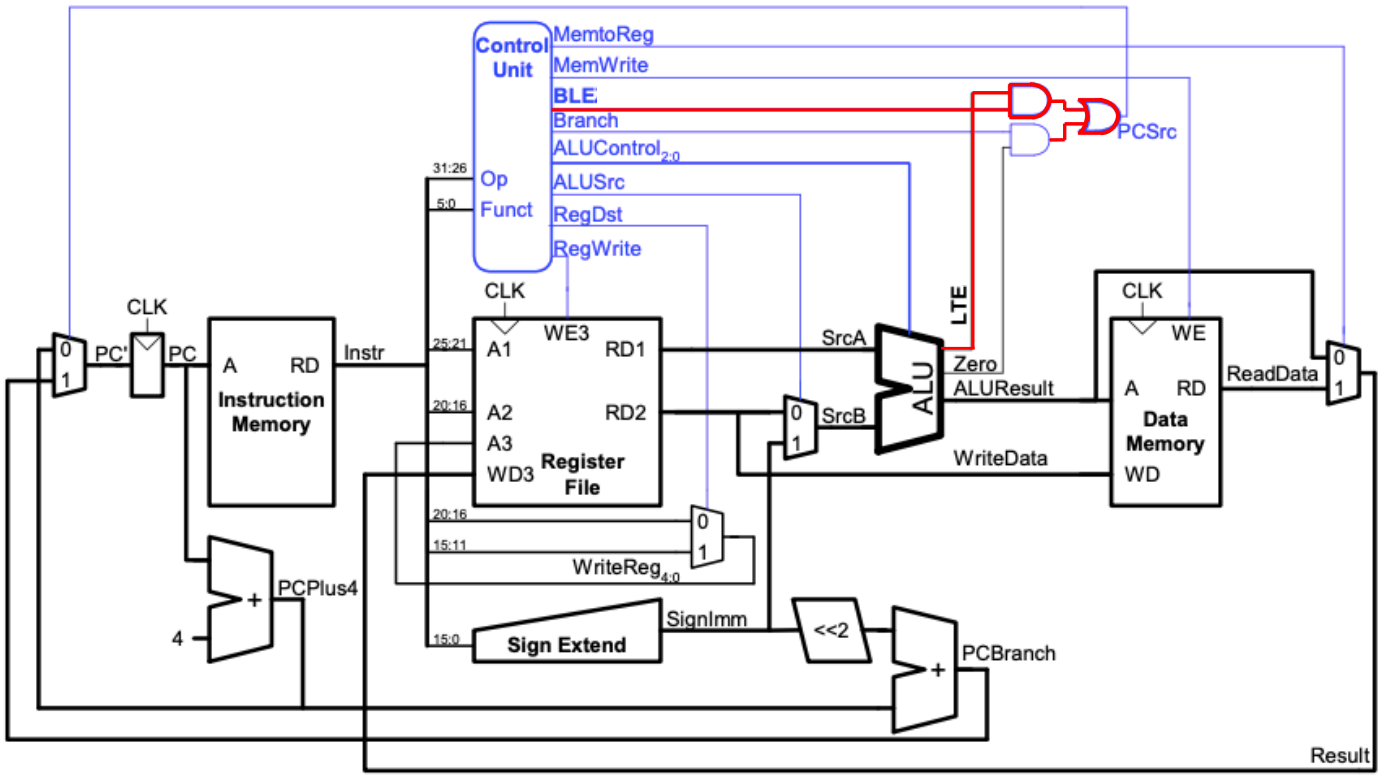
d) Datapath

ble:

First, ALU is modified to support less than and equal to (LTE). Then the datapath is modified as the following to support the result from ALU and take the new instruction to the control unit.

subi:

The datapath remains the same as it is the same as performing addi. However, to prevent changes made to register file and memory, control unit should perform a subtraction instead of addition. Thus, ALUControl should be set to 110.



e) Main Control Table

Instruction	$\mathbf{Op_{5:0}}$	RegWrite	RegDst	AluSrc	Branch	MemWrite	MemToReg	$\mathbf{ALUOp_{1:0}}$	Jump
R-type	000000	1	1	0	0	0	0	10	0
lw	100011	1	0	1	0	0	1	00	0
sw	101011	0	X	1	0	1	X	00	0
beq	000100	0	X	0	1	0	X	01	0
addi	001000	1	0	1	0	0	0	00	0
j	000010	0	X	X	X	0	X	XX	1
ble	000110	0	X	0	0	0	X	01	0
subi	001110	1	0	1	0	0	0	01	0

f) Test Program

```
# testing original10
# addi test
addi $v0, $zero, 3
# and test
and $a1, $v0, $v1
# or test
or $a0, $a1, $v1
# beq test
beq $a1, $a3, 0x0003
# slt test
slt $a2, $v0, $v1
# sw test
sw $a0, 68($v1)
# lw test
lw $a3, 68(v10)
# sub test
sub $a3, $a0, $a2

# testing new instructions
# subi test
subi $a3, $a0, 0x0010
# ble test
slt $a3, $a0, $a1
beq !a1, $v1, 0x0003

# j test
j 0x00400000
```