CS224

Lab 04

Section 01

Erdem Ege Eroğlu

21601636

b) The assembly language equivalents of the machine codes.

Address (8'hXX)	Instruction (32'hXXXXXXXX)	Assembly Language Equivalent
00	20020005	addi $v0, $0, 5
04	2003000c	addi $v1, $0, 12
08	2067fff7	addi $a3, $v1, -9
0c	00e22025	or $a0, $a3, $v0
10	00642824	and $a1, $v1, $a0
14	00a42820	add $a1, $a1, $a0
18	10a7000a	beq $a1, $a3, 10
1c	0064202a	slt $a0, $v1, $a0
20	10800001	beq $a0, $0, 1
24	20050000	addi $a1, $0, 0
28	00e2202a	slt $a0, $a3, $v0
2c	00853820	add $a3, $a0, $a1
30	00e23822	sub $a3, $a3, $v0
34	ac670044	sw $a3, 68($v1)
38	8c020050	lw $v0, 80($0)
3c	08000011	j 0x00000011
40	20020001	addi $v0, $0, 1
44	ac020054	sw $v0, 84($0)
48	08000012	j 0x00000012

c) RTL Expressions

Ble:

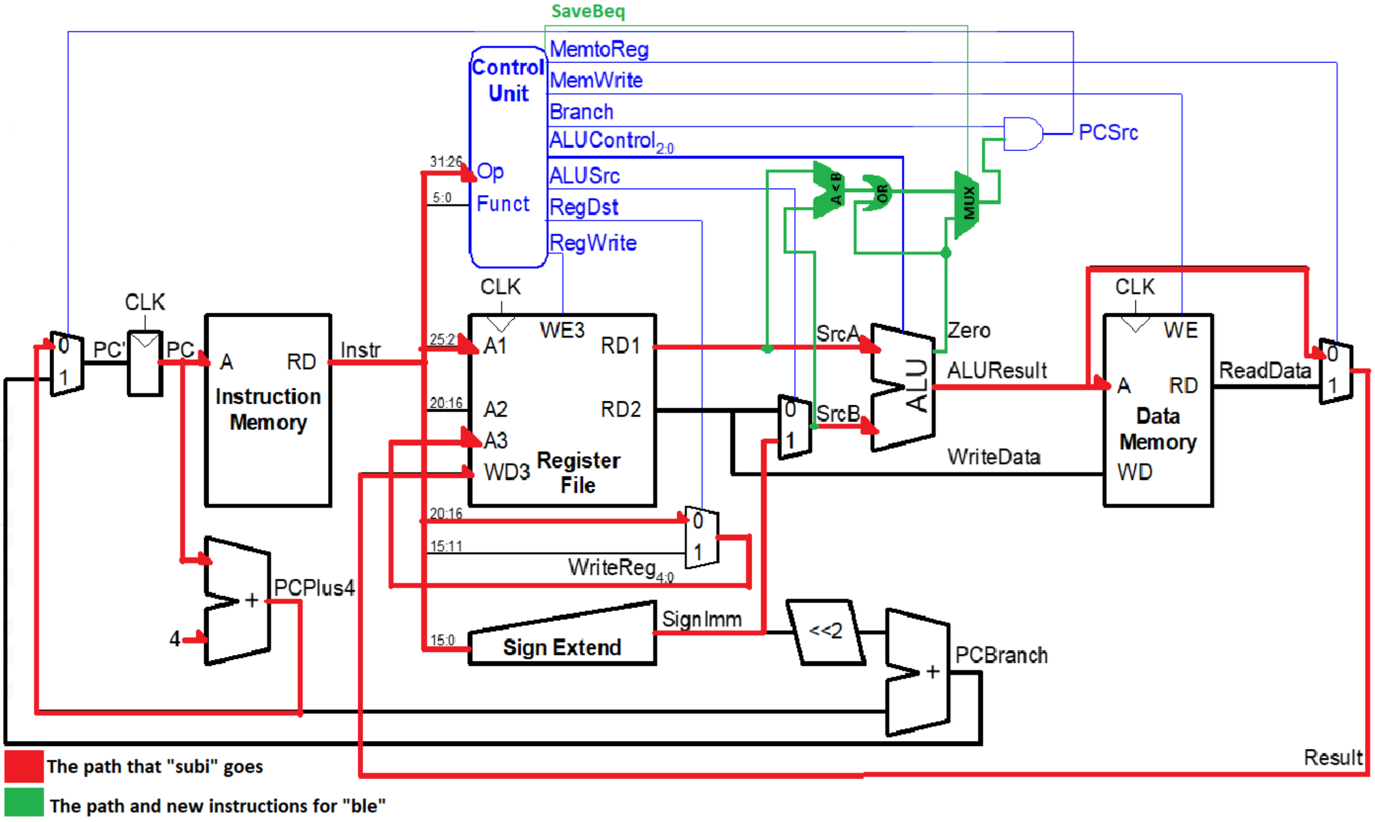
```
    Im [PC]
    if (RF [rs] == RF [rt])
        PC ← 4 + SignExt(immed) << 2
    else if (RF [rt] < RF [rs])
        PC ← 4 + SignExt(immed) << 2
    else
        PC ← PC + 4
```

Subi:

```
    Im [PC]
    RF [rt] ← RF [rs] - SignExt(immed)
    PC ← PC + 4
```

---

d) Datapath



e) Control Table with new operations (Ble, Subi)

Instructions	Opcode	RegWrite	Reg Dst	AluSrc	Branch	Mem Write	MemReg	AluOP	Jump	Save Beq
R-type	000000	1	1	0	0	0	0	10	0	X
Lw	100011	1	0	1	0	0	1	00	0	X
Sw	101011	0	X	1	0	1	X	00	0	X
Beq	000100	0	X	0	1	0	X	01	0	1
Addi	001000	1	0	1	0	0	0	00	0	X
J	000010	0	X	X	X	0	X	XX	1	X
Ble	010101	0	X	0	1	0	X	11	0	0
Subi	010000	1	0	1	0	0	0	01	0	X

ALUOp	Meaning
00	Add
01	Subtract
10	Look at “Funct” field
11	Set less than

f) Test Program

```
.text
        # TESTING NEW INSTRUCTIONS
        # if subi works, s0 becomes 1 or not s0 becomes -1
        # if equality check of ble works, s1 becomes 1 or not -1
        # if less check of ble works, s2 becomes 1 or not -1
        addi    $t0, $0, 5
        add     $t1, $0, $t0
        addi    $t2, $0, 7
        addi    $t0, $t0, -1
        subi    $t1, $t1, 1
        beq     $t0, $t1, subiWorks
        addi    $s0, $s0, -1
subiWorks:
        addi    $s0, $s0, 1
        ble     $t0, $t2, equalWorks
        addi    $s1, $s1, -1
equalWorks:
        addi    $s1, $s1, 1
        ble     $t0, $t2, lessWorks
        addi    $s2, $s2, -1
lessWorks:
        addi    $s2, $s2, 1
done:
        addi    $v0, $0, 10 # exit
        syscall
```