

BILKENT UNIVERSITY
COMPUTER ENGINEERING
CS224 COMPUTER ORGANIZATION

PRELIMINARY DESIGN REPORT

LAB 4 Section : 3

Boran Yıldırım

21401947

Monday, 03 April 2017

B)

<u>Address</u>	<u>Machine</u>	<u>Assembly</u>
0	20020005	addi \$2, \$0, 5
4	2003000C	addi \$3, \$0, 12
8	2067FFF7	addi \$7, \$3, -9
C	00E22025	or \$4, \$7, \$2
10	00642824	and \$5, \$3, \$4
14	00A42820	add \$5, \$5, \$4
18	10A7000A	beq \$5, \$7, end
1C	0064202A	slt \$4, \$3, \$4
20	10800001	beq \$5, \$7, around
24	20050000	addi \$5, \$0, 0
28	00E2202A	around: slt \$4, \$7, \$2
2C	00853820	add \$7, \$4, \$5
30	00E23822	sub \$7, \$7, \$2
34	AC670044	sw \$7, 68(\$3)
38	8C020050	lw \$2, 80(\$0)
3C	08000011	j end
40	20020001	addi \$2, \$0, 1
44	AC020054	end: sw \$2, 84(\$0)

C)

1) *bge*:

bge \$t2, \$t7, TopLoop

IM[PC]

$R[\$rd] \leftarrow R[\$rs] < R[\$rt]$

if($R[\$rd] \leq 0$)

$PC \leftarrow PC + 4 + \text{SignExt18b}(\{imm, 00\})$

else

$PC \leftarrow PC + 4$

2) *swapRM*:

swapRM \$v0, 1004(\$sp)

IM[PC]

$R[\$rt] \leftarrow \text{Mem4B}(R[\$rs] + \text{SignExt16b}(imm))$

$R[\$rd] \leftarrow R[\$rs] + R[\$rt]$

$\text{Mem4B}(R[\$rs] + \text{SignExt16b}(imm)) \leftarrow R[\$rt]$

$PC \leftarrow PC + 4$

E)

Instruction	OpCode	Reg Write	RegDst	ALUSrc	Branch	Mem Write	Mem toReg	ALUOp	Jump	Regwrite2
R-type	000000	1	01	0	0	0	00	10	00	0
lw	100011	1	00	1	0	0	01	00	00	0
sw	101011	0	xx	1	0	1	xx	00	00	0
beq	000100	0	xx	0	1	0	xx	01	00	0
j	000010	0	xx	x	x	0	xx	xx	01	0
bge	001001	1	01	1	1	0	10	xx	00	1
swapRM	010110	1	0	1	0	1	10	1	0	1

F)

.data

suc_BGE : " BGE succeeded."

suc_SwapRM: "SwapRM succeeded."

fatalError: "Fail!"

.text

addi \$a0, \$0, 1910 ## test

li \$v0, 1 ## print the test

syscall

add \$a0, \$a0, \$a0 ## test

li \$v0, 1 ## print test

syscall

lw \$a0, 12(\$zero) ## test

li \$v0, 1 ## print test

syscall

testing for beq

beq \$0, \$0, branch

li \$v0, 4

la \$a0, fatalError ## error bqe failed

syscall

branch:

```
j middle          ## testing for jump
li $v0, 4
la $a0, fatalError    # error j failed
syscall
```

middle:

```
add $t0, $a0, $0
bge $a0, $s0        ## testing for bge
li $v0, 4
la $a0, suc_BGE
syscall
```

quit :

```
li $v0, 10
syscall
```