15

BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Quiz
Duration: 35 minutes

Semester :Summer 2022
Full Marks: 15

CSE 340: Computer Architecture

Name: Kazi Md. Al-Wakil ID: 19301051 Section: 7

1. **(CO2)** Consider that the Program counter has the value (in hex) 0x00C06715. If the offset value (in decimal) is 228, then **calculate** the conditional (branch) address and unconditional (jump) target address. **[6 Marks]**

$PC = (00C06715)_{16} = (12609301)_{10}$

$offset = (228)_{10}$

conditional branch:

Now,

$228 \times 4 = (912)_{10}$ [2 bit left Shift, because each ~~slo~~ instr. holds 4 slots]

Now,

$(12609301 + 4) = (12609305)_{10}$. [value of PC]

Now,

$12609305 + 912 = (12610217)_{10} = (00C06AA9)_{16}$

Ans: Jump address conditional = 0x00C06AA9

6

Unconditional branch:

$(228 \times 4) = 912$ (2 bit left shift, because each ~~slot~~ instr. has 4 slots)

This $(912)_{10}$ holds 28 bits. (26 bit offset + 2 bit [for Shift]

Now, The msb ~~ø~~ 4 bit of $(00C06715)_{16}$ will be the msb of the offset.

MSB 4 bit of $(00C06715)_{16} = (0000)_2 \Rightarrow (0)_{16}$

$(912)_{10} = \underbrace{(0000390)_{16}}_{28\ bit} \Rightarrow \underbrace{(00000390)_{16}}_{32\ bit}$

Ans: Jump address: $(00000390)_{16}$

2. **(CO2) Convert** the following C code to MIPS code: **where the base address of C is in $s5 and x, y are in $s6 and $s7. [9 Marks]**

```
for (int x = 4; C[x] >= 5; x++){
    if( x == y ) {
        C[x+1] = C[x];
    } else {
        C[x+3] = C[x+1];
    }
}
```

c → $s5
x → $s6
y → $s7

$s6 → 4
$t0 → 4x
$t1 → C[x] address
$t2 → Value of C[x]

```
        add   $s6, $zero, 4
Loop :  sll   $t0, $s6, 2
        add   $t1, $s5, $t0
        lw    $t2, 0($t1)
        ~~Slti $t2, 5~~
        Slti  $t4, $t2, 5
        bne   $t4, $zero, exit

        ~~be~~
        bne   $s6, $s7, else
        sw    $t2, 4($t1)
        addi  $s6, $s6, 1
        J     loop

else:
        lw    $t5, 4($t1)
        sw    $t5, 12($t1)
~~exit~~:
        addi  $s6, $s6, 1
        J     loop

exit:
```

```
        add   $s6, $zero, 4
loop:   sll   $t0, $s6, 2
        add   $t1, $s5, $t0
        lw    $t2, 0 ($t1)
        Slti  $t4, $t2, 5
        bne   $t4, $zero, exit
        bne   $s6, $s7, else
        sw    $t2, 4 ($t1)
        addi  $s6, $s6, 1
        J     loop

else:
        lw    $t5, 4 ($t1)
        sw    $ t5, 12($t1)
        addi  $s6, $s6, 1
        J     loop

exit:
```

out of scope

8+1