CS224
Fall 2020
Quiz No. 2
Date Quiz Given: Oct. 21, '20
Date of Solutions: Nov. 3, '20

Q1.

text segment

No. of mach. inst

Location (hex)

```
        label1: add $t0, $t1, $t2      1   <- 0x00 40 00 CC
2 inst. -> label2: la $t2, sum         2   <- 0x00 40 00 D0
will be         lw $t3, 4($t2)         1
generated       ble $t3, $t2, label1   2
                j  label2
```

$$\begin{array}{r} CC_{16} \\ +\ 4_{16} \\ \hline D0_{16} \end{array}$$

data segment

Location (hex)

```
                                      0x 00 10 01 0C
12 byte -> array1: .word 10, 20, 30 -> 0x 00 10 01 18
           array2: .space 16        -> 0x 00 10 01 28
           sum:    .space 4         ->  ^ location of sum
```

24 = 18

$$\begin{array}{r} 0C \\ +\ 0C \\ \hline 18_{16} \end{array}$$

$t0: $8
$t1: $9
$t2: $10
$t3: $11

① add $t0, $t1, $t2
add $8, $9, $10
add rd, rs, rt

size in bits

```
        6      5   5   5   5      6
R inst: opcode/rs/rt/rd/shamt/funct
          0     9  10  8   0      20 hex
        00 0000 01001 01010 01000 00000 10 0000
         0    1    2   A    4    0    2    0
```

0x 01 2A 40 20

② la $t2, sum (sum is defined at 0x 00 10 01 28)

lui $at, 0x0010
ori $t2, $at, 0x0128 → R[rt]: R[rs] | ZeroExtImm

lui: from Greencard
R[rt] = {imm, 16'b0}
rs is not specified: use $0

```
opcode/rs/rt/imm
 f hex  0  1  0x0010
00 1111 00000 00001  0x0010
 3   C    0     1
```

0x 3C 01 00 10 for lui

```
opcode/rs/rt/imm
 d hex  1  10  0x0128
00 1101 00001 01010 0x0128
  3   4    2    A
```

0x 34 2A 01 28 for ori

Q1 cnt.

③ lw $t3, 4($t2)

lw $11, 4($10)

lw rt, 4(rs)

lw : opcode / rs / rt / imm

23hex 10 11 0x0004

10 0011 01010 01011 0x0004

8 D 4 B 0x0004

0x 8D 4B 00 04 for lw

⑤ j label2

label2 : 0x 00 40 00 D0

0000

will not be stored in inst

j label2

opcode / add

skip

2 0x00 40 00 D0

00 0010 0000 0100 0000 0000 0000 1101 00

0 8 1 0 0 0 3 4

0x 08 10 00 34

④ ble $t3, $t2, label1

($t3 <= $t2) ≡ ($t3 not > $t2)

not ($t3 > $t2)

not ($t2 < $t3)

not ($10 < $11)

slt $at, $10, $11

beq $at, $zero, label1

slt $at, $10, $11

slt $1, $10, $11

slt rd, rs, rt

opcode / rs / rt / rd / shamt / funct

0 10 11 1 0 2A

00 0000 01010 01011 00001 00000 10 1010

0 1 4 B 0 8 2 A

0x 01 4B 08 2A for slt

beq $at, $zero, label1 ← 6 inst back

beq $1, $0, −6

beq rs, rt, −6 −6 ⇒ 0110

1001

+ 1

1010

opcode/rs/rt/imm

4 1 0 0xFFFA

00 0100 00001 00000 0xFFFA

1 0 2 0

0x10 20 FFFA

Q2 Pseudo instruction implementation

① **move $t0, $t1**

```
add $t0, $t1, $zero
```

given as an example

② **invert $t1**

```
addi $at, $zero, -1   # $at contains FF--FF
xor $t1, $t1, $at
```

xor --> inverting bits
10 --> 1
01 --> 1
00 --> 0
11 --> 0

③ **sw 4, 0($t0)**

store 4 to memory loca. 0($t0)

```
addi $at, $zero, 4
sw   $at, 0($t0)
```

④ **lwinc $t1, 0($t0)**

```
lwinc $rt, 0($rs)
```

not required provide for explanation

RTL description
IM[PC]
RF[rt] ← DM[RF[rs] + SignExtimm] + 1
PC ← PC + 4

```
lw   $at, 0($t0)
addi $at, $at, 1
add  $t1, $zero, $at
```

⑤ **swap $t1, $t2**

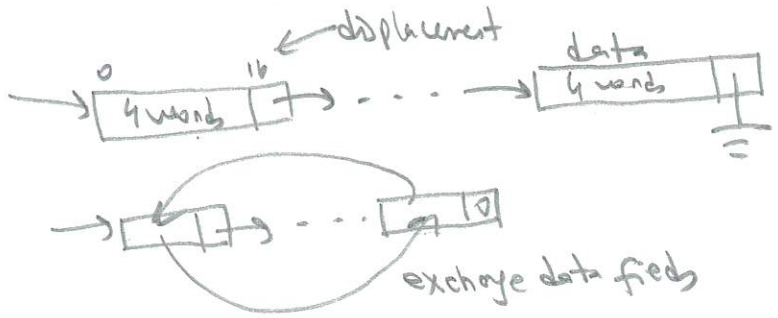
```
add $at, $zero, $t0
add $t0, $t1, $zero
add $t1, $at, $zero
```

⑤ **bgezero $t1, label**

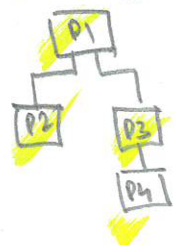
not($t1 < $0)

```
slt $at, $t1, $zero
beq $at, $zero, label
```

Q3. Linked list: exchange data fields of head and tail nodes.



Program Hierarchy



Subprogram	Function
P1: exchangeHeadTail	What we want to do $a0: points to the linked list
P2: findTail	Return address of tail node in $v0, head node passed in $a0
P3: swapDataFields	Swap data fields of nodes pointed by $a0 and $a1
P4: swapItem:	Swap the contents of two words pointed by $a0 and $a1

P1 exchangeHeadTail:

```
    add $sp, $sp, -12      save
    sw $s0,8($sp)          $s0, $s1
    sw $s1,4($sp)          $a0 to
    sw $ra,0($sp)          stack
     beq $a0,$zero,done1
     move $s0,$a0
     jal findTail
# $v0: points to tail.
      move $s1,$v0
 # $s0: head, $s1: tail pointer
 # if only one node: done
        beq $s0,$s1,done1
        add $a0,$zero,$s0
        add $a1,$zero,$s1
        jal swapDataFields
done1: lw $ra,0($sp)        restore
       lw $s1,4($sp)        values of
       lw $s0,8($sp)        $a0,$s1,
       addi $sp,$sp,12      $s0 from
       jr $ra               stack
```

P2 findTail:

```
    addi $sp, $sp, -8
    sw $s0,4($sp)
    sw $s1,0($sp)
    move $s0,$a0
    lw $s1,$16($s0)
# $s0: current pointer
# $s1: next pointer
seekNext:
    beq $s1,$zero,done2
    add $s0,$zero,$s1
    lw $s1,16($s1)
    j seekNext
done2:
    add $v0,$zero,$s0
    lw $s1,0($sp)
    lw $s0,4($sp)
    addi $sp,$sp,8
    jr $ra
```

16 byte

P3 swapDataFields:

```
  addi $sp,$sp,-4
  sw $ra,0($sp)
  jal swapItem
  addi $a0,$a0,4
  addi $a1,$a1,4
  jal swapItem
  addi $a0,$a0,4
  addi $a1,$a1,4
  jal swapItem
  addi $a0,$a0,4
  addi $a1,$a1,4
  jal swapItem
  lw $ra,0($sp)
  addi $sp,$sp,4
  jr $ra
```

P4 swapItem:

```
    addi $sp,$sp,-8
    sw $s0,4($sp)
    sw $s1,0($sp)
exchange      lw $s0,0($a0)
2 mem.        lw $s1,0($a1)
words pointed sw $s0,0($a1)
by $a0 and    sw $s1,0($a0)
$a1
```

swapItem (cont)

```
    lw $s1,0($sp)
    lw $s0,4($sp)
    addi $sp,$sp,8
    jr $ra
```

First Part

Q4. Floating point number representation.

-1024.5 → SP ① / → DP ②

428.44_{16} → SP ③ / → DP ④

SP = Single Precision

DP = Double Precision

① -1024.5 in IEEE 754 SP format

$\frac{1024}{16} = 64$ remainder = 0

$\frac{64}{16} = 4$ remainder = 0

$\frac{4}{16} = 0$ remainder = 4

↑ =0 stop

.5
16
——
8.0

↑ zero stop

$1024.5 = 400.8_{16}$

$\underbrace{100\ 0000\ 0000}.1000$

$1.00\ 0000\ 0000\ 1 \times 2^{10}$

↑ not stored

IEEE SP

1 8 23 bits
S exp fraction

exp = bias / excess
$127 = 7F_{16}$

$7F + A = 89_{16}$

$F + A = 25_{10} = 16 + 9$

$\underbrace{1\ 100}_{C}\underbrace{0\ 100}_{4}\underbrace{1\ 000}_{8}\underbrace{0\ 000}_{0}\underbrace{0\ 001}_{1}\underbrace{00..}_{0..}$

-1024.5 SP → $\boxed{0X\ C4\ 80\ 10\ 00}$

② -1024.5 in IEEE 754 DP format

1 11 52 bits
s exp fract

exp = bias/excess
1023

$3FF + A = 409_{16}$ ← exponent

→ fraction

$\underbrace{1\ 100}_{C}\underbrace{0000}_{0}\underbrace{1001}_{9}\underbrace{0000}_{0}\underbrace{0000}_{0}\underbrace{0010}_{2}\ 0...0$

-1024.5 DP → $\boxed{0X\ C0\ 90\ 02\ 00\ 00\ 00\ 00\ 00}$

③ $42B.44_{16}$ → $\underbrace{100\ 0010\ 1011}.0100\ 0100$ → fraction

exp = 10

$7F + A = 89$

→ fraction

$\underbrace{0\ 100}_{4}\underbrace{0\ 100}_{4}\underbrace{1\ 000}_{8}\underbrace{0\ 101}_{5}\underbrace{1\ 011}_{6}\underbrace{0\ 100}_{8}\underbrace{0\ 100}_{8}...$

$\boxed{0X\ 44\ 85\ 68\ 80}$?

Quiz 2 / 5 of 6

④ 42B.44 in double precision

$$\begin{array}{r} 3FF \\ +\ \ A \\ \hline 409 \end{array}$$

s exp fraction

exp.

$0\ \underbrace{100}_{4}\ \underbrace{0000}_{0}\ \underbrace{1001}_{9}\ \underbrace{00\ 00}_{0}\underbrace{10\ 10}_{A}\underbrace{11\ 01}_{D}\underbrace{00\ 01}_{1}00$

$$\boxed{0x40\ 90\ AD\ 10\ 00\ 00\ 00\ 00}$$

Q4. Second part

SP. → 0X42 B4 40 00 → Dec? ①

DB → 0X405F 70 00 00 00 00 00 → Dec. ②

① 42 B4 40 00

$\underbrace{0100\ 0010\ 1}_{}$ ... $\underbrace{0100}_{8}\ \underbrace{0010\ 1}_{5}011$ 0100 0100 ...

1011 → fraction

$$\begin{array}{r} 85 \\ -\ 7F \\ \hline 06_{16} \end{array}$$ excess value

$\frac{2}{16} = \frac{1}{8} = 0.125$

$1.011\ 0100\ 010 \cdots \times 2^6$

$\underbrace{1\ 01}_{5}\underbrace{1\ 010}_{A}.\underbrace{0010}_{2}$

$5A.2_{16} \Rightarrow 5\times16^1 + 10\times16^0 + 2\times16^{-1}$

$$\boxed{90.125}$$

1 11 52 bits

s exp fract

② 0X 40 5F 70 00 ...

0100 0000 0101 1111 0111

→ fraction

$$\begin{array}{rl} 405_{16} & \text{exp} \\ -\ 3FF & \text{bias} \\ \hline 006 & \text{Exponent} \end{array}$$

$1.1111\ 0111 \times 2^6$

$\underbrace{1\ 11}\underbrace{11\ 01}.11_2$

1100 --> C

$7D.C_{16}$ $7\times16^1 + 13\times16^0 + 12\times16^{-1}$

$$\boxed{125.75}$$

Quiz 2 / 6 of 6