SEAN CONNOR

Assignment 5a
CORRECTIONS

2.20)

```
srl  $t2, $t0, 11
sll  $t2, $t2, 26
sll  $t1, $t1, 6
srl  $t1, $t1, 6
or   $t1, $t0, $t2
```

ANS

SEAN CONNOR

2.26) b)

```
B = 0;
i = 10;
while (i > 0) {
    i = i - 1;
    B = B + 2;
}
```

ANS

SEAN CONNOR

abs $s4, $s7

```
sra $t0, $s7, 31
xor $t1, $s7, $t0
sub $s4, $t1, $t0
```

ANS

SEAN CONNOR

rol $t4, $t3, 12

```
sll $t0, $t3, 12
srl $t1, $t3, 20
or  $t4, $t1, $t0
```

ANS

SEAN CONNOR
3 July 2018
Comp. Org. 5a

2.20) 1. srl $t0, $t0, 11

1111 1111 1111 111x xxxx x111 1111 1111

↳

0000 0000 0001 1111 1111 1111 11xx xxxx

2. sll $t0, $t0, 26

0000 0000 0001 1111 1111 1111 11xx xxxx

↳

xxxx xx00 0000 0000 0000 0000 0000 0000

3. sll $t1, $t1, 6

4. srl $t1, $t1, 6 — give

0000 0011 1111 1111 1111 1111 1111 1111

5. add $t1, $t0, $t1

```
  xxxx xx00 0000 0000 0000 0000 0000 0000
+ 0000 0011 1111 1111 1111 1111 1111 1111
-------------------------------------------
  xxxx xx11 1111 1111 1111 1111 1111 1111
```

```
1  srl $t0, $t0, 11
2  sll $t0, $t0, 26
3  sll $t1, $t1, 6
4  srl $t1, $t1, 6
5  add $t1, $t0, $t1
```

ANS

2.25) a) I-type is most appropriate because I-type includes 16 bits for address, which is required for handling the loop address.

b)
```
loop:  blez  $t2 , next
       addi  $t2 , $t2 , -1
       j     loop
next:
```

2.26) a) 1 - $t1 = 10 $t2=1 $t1=9 $s2=2
2 - $t1= 9 $t2=1 $t1=8 $s2=4
3 -
4 -
5 -
6 -
7 -
8 -
9 -
10 - $t1=1 $t2=1 $t1=0 $s2=20
11 - $t1=0 $t2=0 → DONE

$s2=20
ANS

b)
```
while (i>0) {
    i=i-1
    B=B+2
}
```
ANS

c) The final iteration only executes 2 instructions, while all others execute 5. Thus the answer is 5N+2

5N+2
ANS

abs $s4, $s7

1. sra $t0, $s7, 31

Sign bit is shifted in, so $t0 will be 0x0 if positive and 0xFFFFFFFF if negative.

2. xor $s7, $s7, $t0

ex1. 5 = 00000101 $t0 = 00000000 $s7 = 00000101 ⇒ 00000101

ex2 -5 = 11111011 $t0 = 11111111 $s7 = 11111011 ⇒ 00000100

3. sub $s4, $s7, $t0

ex1. 00000101 - 00000000 = 00000101 = 5

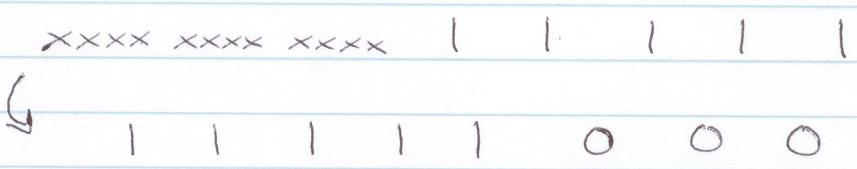
ex2 00000100 - 11111111
= 00000100 + 00000001 = 00000101 = 5

```
sra $t0, $s7, 31
xor $s7, $s7, $t0
sub $s4, $s7, $t0
```

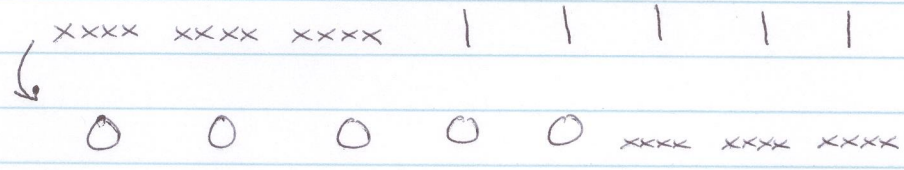
ANS

rol $t4, $t3, 12

1. sll $t0, $t3, 12



2. srl $t3, $t3, 20



3. $t0 = 1 1 1 1 1 0 0 0
   $t3 = 0 0 0 0 0 xxxx xxxx xxxx

or $t3, $t3, $t0

$t3 = 1 1 1 1 1 xxxx xxxx xxxx

```
sll $t0, $t3, 12
srl $t3, $t3, 20
 or $t3, $t3, $t0
```

ANS

se

ld $t6, 0($s4)

```
lw $t6, 0($s4)
lw $t7, 4($s4)
```

ANS