COMP 2421 - Homework 1

Name: Guanlin Jiang (21093962D)

Q1

1.1 Answer

X = 124 base 6

This number base 7 is 103.

1.2 Answer

(1)

The binary form of decimal value -19 is 111101101.

(2)

The decimal value of the binary number 110101001 is -87.

1.3 Answer

(1)

0111 + 1111 = 10110 (not overflow)

(2)

1110 + 1000 = 10110 (overflow, because two negative number add, but get positive)

1.4 Answer

Memory Address B can be instuction, because A is in the range of 0x00000000 to 0x00400000, which is reserved, so address A can't be instuction.

1.5 Answer

X	Y	X'	Y'	F	X+Y	X'+Y'	X'+Y	X+Y'
1	1	0	0	1	1	0	1	1
0	1	1	0	1	1	1	0	1
1	0	0	1	0	1	1	1	0
0	0	1	1	1	0	1	1	1

f = x' y' + xy + x' y ==> = x' + y

Q2

2.1 Answer

The $8 value for Code Block A is 0, becasue for signed values, $6 > $7.

The $8 value for Code Block B is 1, becasue for unsigned values, $6 < $7, $7 will be a large postive value if unsigned.

2.2 Answer

(1)

```
lui $t2, 0x1234
ori $t3, $t3, 0xabcd
addu $t1, $t2, $t3
```

(2)

```
slt $t3, $t2, $t1
bne $t3, $0, L1
```

Q3

(1)

```
addu $t3, $t1, $t2 // add the $t1 and $t2, store to $t3
lw $t4, 12($t6) // adding the value in $t6 and the immediate offset of 12, and stores in $t4
addu $t0, $t3, $t4 // add the $t3 and $t4, store to $t0
```

(2)

```
lw $t3, 12($t6) // adding the value in $t6 and the immediate offset of 12, and stores in $t3
sll $t3, $t3, 2 // $t3 shift left logical 2 bits, store to $t3
addu $t3, $t3, $t5 // add the $t3 and $t5, store to $t3
lw $t3, 0($t3) // adding the value in $t3 and the immediate offset of 0, and stores in $t3
addu $t0, $t1, $t3 // add the $t1 and $t3, store to $t0
```

Q4

(1)

ble $t1, $v1 implement the if less then or equal to flow control statement

(2)

$t0 --> a pointer which point the element in array

$t1 --> store the current element in array

$v0 --> store the address of the smallest element

$v1--> store the value of the smallest element

(3)

move $v0, $t0 --> copy the memory address of the current smallest element to $v0 (update the address of smallest element to $v0)

move $v1, $t1 --> copy the memory address of the current smallest element to $v1 (update the address of smallest element to $v1)

(4)

bne $t0, $a1, loop --> loop and stop at the value of $t0 is equal to the value of $a1, which use to when the all the elements in array S are evaluated it will be stop.

(5)

This function is to find the smallest value address ($v0) and value ($v1) in array S.

(6)

$v0 --> 0x20060004

$v1 --> -29