

Bill Davis

605.421 Assignment 2

1. (a) In order to negate a 64-bit value we need to use unsigned instructions to avoid an overflow exception.

```
subu $s9, $zero, $s9
subu $s8, $zero, $s8
addi $s8, -1
```

- (b) In order to subtract a 64 bit value from another we can subtract the low order part to determine if we need a carry

```
subu $v0, $s3, $s9
sltu $v0, $s3, $v0 #determine carry
subu $s2, $s2, $s8
subu $s2, $s2, $v0 #subtract carry if set
```

2. (a) $835623 = 0000\ 000\ 0000\ 1100\ 1100\ 0000\ 0010\ 0111$
 $000000\ 00000\ 11001\ 10000\ 00001\ 00111$

```
sra $s0, $zero, $t9
```

- (b) In excess 2^{16} 835623 would be interpreted as 770087.

- (c) The range of values in this system is -2^{16} to $2^{16} - 1$.

3. $1281977701 = 4C696D65 = \text{LIME}$

4. `addi $s4, $at, -12593`

`addi = 8 $s4 = 20 $at = 1`

`-12593 = 1100111011001111`

So the immediate instruction is:

`=001000 00001 10100 1100111011001111`

`=0010 0000 0011 0100 1100 1110 1100 1111`

`=0x2034CECF`

5. Decode `0x3408CECF`

`0011 0100 0000 1000 1100 1110 1100 1111`

Since `001101 = ori` then we can break up the instruction as follows

`001101 00000 01000 1100111011001111`

`ori $zero $t0 immediate`

This would therefore place the value `0xFFFF3130` into the register `$t0`