Kyler Little
CPT_S 260

Homework #2

1. Represent the following signed numbers in 32-bit two's complement:

(1). +12

It's easy to convert by using a placeholder's table.

sign	2^{30}	2^{29}	...	2^3	2^2	2^1	2^0
0	0	0	...	1	1	0	0

Thus, +12 is 0000_0000_0000_0000_0000_0000_0000_1100.

(2). -12.

We can use the results of part (1) to compute this. First, we flip each bit to negate the number. Now, since there is one more negative number than positive numbers (due to the number zero), we must add 1 to obtain our final result.

After flipping, we have 1111_1111_1111_1111_1111_1111_1111_0011.

After adding one, we have 1111_1111_1111_1111_1111_1111_1111_0100, our final result.

2. Download Mars , the MIPS simulator, and write a MIPS program that will print the string Hello on the terminal. Make sure you run the program on the simulator and it actually works.

```
        .text
        .globl main
main:
        la $a0, helloStr
        li $v0, 4
        syscall

        li $v0, 10
        syscall


        .data
helloStr: .asciiz "Hello\n"
```

3. Every register stores 32-bits. Suppose now that the register s1 stores the following 32 bits: 111 · · · 1111 where each bit is simply 1.

(1). When I run the instruction add s2, s1, s3, what's the value of the number stored in s1?

Because of two's complement, all ones represents the number:

$$-1$$

(2). When I run the instruction addu s2, s1, s3 instead, what's the value of the number stored in s1?

Now, the first bit is no longer a sign bit, so we get:

$$2^{32} - 1 = 4{,}294{,}967{,}295$$