# Exercise 1

## Binary to MIPS Instruction

1100 0101 0111 0000 0000 0000 0000 0000

110001 01 0111 0000 0000 0000 0000 0000

Opcode 0d110001 -> 0x31 -> load FP single [lwc1]

Lwc1 is type I

I type:

Opcode rs rt immediate

110001 01011 10000 0000000000000000

0d11 0d16

Lwc1 $t3 $s0 0d0

lwcl1 $s0, 0($t3)

## Binary Single precision IEEE 754 to decimal

1100 0101 0111 0000 0000 0000 0000 0000

IEEE 754 floating point looks like this (MIPS Greensheet)

Sign exponent fraction

1 bit 8 bits 23 bits

Sign Exponent Fraction

1 10001010 11100000000000000000000

0d1 0d138 ½ + ¼ + 1/8 = 7/8 = 0,875

Bias = 127 for single precision (MIPS Greensheet)

So with hidden bit:

(-1)^S x (1+Fraction) x 2^(Exponent-Bias) = (-1)^1 x (1 + 0,875) x 2^(138 – 127) = -3840

Without hidden bit:

(-1)^S x Fraction x 2^(Exponent-Bias) = (-1)^1 x 0,875 x 2^(138 – 127) = -1792

## Binary Signed integer to hex sign magnitude

1100 0101 0111 0000 0000 0000 0000 0000

First bit is 1, so it is negative. In order to turn it positive: flip the bits and add 1.

1100 0101 0111 0000 0000 0000 0000 0000

0011 1010 1000 1111 1111 1111 1111 1111 flip bits

0011 1010 1001 0000 0000 0000 0000 0000 add 1

2 A 9 0 0 0 0 0

So it’s magnitude in hex is 2A900000

## Binary unsigned integer to hex

1100 0101 0111 0000 0000 0000 0000 0000

Converting each 4 bits to hex gives:

C 9 7 0 0 0 0 0

So it becomes C970000 in hex

## Binary to ASCII

1100 0101 0111 0000 0000 0000 0000 0000

ASCII is each 7 bits. So we drop the last 4 bits.

1100010 1011100 0000000 0000000

0d98 0d92 0d0 0d0

b \ NUL NUL

So the ASCII is b\ NUL NUL

# Exercise 2

# Exercise 3

Only if both operands are positive, there can be overflow. Here I assume overflow is when the positive

The overflow occurs when the sign of the result is negative, while the operands are positive.

So: Overflow = (a(msb-1) nand b(msb-1)) and result(msb)

# Exercise 4

4