COMP 2421 Computer Organization

Assignment 1

Name: Zhu Jin Shun

ID:22101071d

Part 1：Questions with Short Answers

Q1

Hexadecimal number: FD2

Into binary two’s complement: 1111 1101 0010.

To find the decimal value of a negative number in 2's complement, we flip the bits and add 1.

So, flipping the bits of 1111 1101 0010 gives 0000 0010 1101.

Adding 1 gives 0000 0010 1110, in decimal equals 2+4+8+32=46

Therefore, the decimal value of the hexadecimal number FD2 in 12-bit 2's complement representation is -46.

Q2

1 1000 1001 111 1100 1101 0000 0000 0000

Sign bit: 1 means negative number

8 bits for biased exp: 1000 1001

k=8 bias=2^8-1-1=127

E=1+8+128=137

E-bias=10

23 bits for significand:

1. 11111001101000000000000=1.975098

Final decimal value=-1*2^10*1.975098=-2022.5

Q3

1)Result is 0111+1111=10110, there is no overflow because 1111 is -1 and 0111 is 7. So if positive + negative = positive 1(ignore) 0110=6 is possible, so there is no overflow

2)1110+1000=10110, there is overflow because 1110 is -2 and 1000 is -8, but negative + negative =positive 1(ignore) 0110=6 which is not possible so there is overflow.

Q4

The third instruction should be sltu $t3,$0,1

Q5

addi $sp, $sp, -8

sw $s1, 4($sp)

sw $s2, 4($sp)

Part 2：MIPS: Translate Pseudo-instructions

Q1

```
srl $t2, $t1, 7

sll $t3, $t1, 25

or $t1, $t2, $t3
```

Q2

```
sll $t1,$t2,4

sub $1,$1,$2
```

Q3

```
lui $t2, 0x0001

ori $t2, $t2, 0x0002

add $t3, $t1, $t2

lw $t4, 0($t3)
```

Part 3： MIPS: Translate MIPS program into C program

$s0=x, $s1=y, $s2=z, $s3=w

The transversed code is

```
 if (z<=w || (x>=y&&x==z))

{ if (z==w)

{
```

```
x=y+z;
}else{
z=y+(-2);
}
}
```

Part 4：MIPS: Understand MIPS Code

Q1

The ble statement implements a while loop. It means that it will continue to execute until $t1<=$v1 and $t0 !=$ a1

Q2

Usage:

$t0: This register is used similar as the i-value used in a for loop in programming and to add up the addresses under a loop and finally add up the address. It stores the address of the elements.

$t1: This register is used for copying value to register $t1. It specially stores the value of the elements who has address stored in $t0.

$v0: This register is used for storing the address of the maximum found in each execution and stores the address in $v0

$v1:This register is used to accept the maximum value. It specially stores the maximum value found in each execution and has address stored in $v0

Q3

Move instruction is used to copy from register to register. So, move $v0, $t0; move $v1, $t1. The first move instruction means to copy register $t0 and move its address into $v0. The second move instruction means to copy register $t1 and move its value into $v1. In this situation, these two instructions are used to update the address and the value of the maximum found in each execution and store it in $v0 and $v1.

Q4

The usage of bne instruction is to test if registers are not equal, so for this bne instruction bne $t0, $a1, loop. It means that if $t0 is not equal to $a1, then go to loop. In this case, it is used to identify if the current element is the last elements in the array and compare with the current maximum, if it isn't the last elements, the loop will continue until the final elements is reached and compared to find the maximum.

Q5

This program finds the maximum value in an array of integers and the address of that maximum value.

The base address of the array is loaded into $a0 and the address of the last element of the array is calculated and stored in $a1.

The program then enters a loop where it compares each element of the array with the current maximum value (initially the first element of the array). If the current element is greater than the current maximum, it updates the maximum value and the address of the maximum value. The loop continues until it has processed all elements of the array. Finally, the maximum value of the array is stored in $v1 and its address is stored in $v0 and returned as output.

Q6

$v0 will contain the address of the maximum value in the array.

In this question, the maximum value is 106, which is the seventh element of the array. As one integer value conducts 4 bytes

So, $v0 will be 0x20060000 + 6*4 = 0x20060018.

$v1 will contain the maximum value in the array, which in this question is 106.