Computer Science 605.611

Problem Set 2

1. (5) Every MIPS assembly language instruction that corresponds to a builtin true-op instruction can be translated into a single 32-bit machine instruction. Use 8 hex digits to show the 32-bit machine instruction that corresponds to the following MIPS assembly language instruction: sll $0,$0,0

R-type
sll op: 0
rs: 0
rt: 0
rd: 0
sll $0,$0,0 -> 00000000

2. (5) Register $8 contains the 32-bit pattern corresponding to 0x8D6A0000. To what value (in decimal format ±d.dd * 10^{E}) does the pattern correspond if interpreted as an IEEE-754 floating point number? Express your answer to two decimal places in the format ±d.dd*10^E.

0x8D6A0000 -> 1000 1101 0110 1010 0000 0000 0000 0000
Sign bit: 1
Exponent field: 00011010; 26-127 = -101
mantissa: 1.11010100000000000000000 = 1.8281250
Decimal: -7.210682e-31

3. (5) Recall that the excess-B representation of a signed integer N is given by N+B, where B is the bias. Use 8 hex digits to show the excess-2147483648 representation of the negative decimal integer -1305464520 . Where 2147483648 is the bias.

-1305464520 + 2147483648 = 842019128 -> 0x32303138

4. (15) If register $8 (also referred to as $t0) contains the 32-bit pattern 0x34789602, indicate (yes or no) whether each of the following interpretations is a valid interpretation of this pattern.

0x34789602 -> 0011 0100 0111 1000 1001 0110 0000 0010

a) a one’s complement integer (if yes, write the integer in decimal)

one’s compliment: 11001011100001110110100111111101 = 880317954

b) a two’s complement integer (if yes, write the integer in decimal)

two’s compliment: 11001011100001110110100111111110 = 880317954

c) a 32-bit IEEE 754 floating point number (if yes, write the number in decimal)

Sign bit: 0
Exponent field: 01101000; 104-127 = -23
mantissa: 1.11110001001011000-00010 = 1.9420779
Decimal: 2.3151372e-7

d) a MIPS machine instruction (if yes, show the assembly language instruction)

invalid

e) a sign & magnitude integer (if yes, write the integer in decimal)

sign: 0 -> +
magnitude = 01101000111100010010110000000010 -> 880317954
ans: +880317954

5. (5) a) Register $4 contains the 32-bit pattern 0x3E000002, which in binary is 00111110000000000000000000000010. What does the CP1 (coprocessor 1) register $f4 contain after the instruction mtc1 $4,$f4 is executed? Express your answer in hex. $f4 = _________

mtc1 simply copies the data from a register, therefore the destination register would contain the same hex data, 0x3E000002

(5) b) CP1 register $f6 contains the IEEE 754 representation of the negative decimal value -2.5. Use 8 hex digits to show the result in register $f8 after executing the instruction cvt.w.s $f8,$f6 .

-2.5 = 11000000001000000000000000000000 = 0xc0200000
cvt.w.s converts single precision floating point to integer
Assuming you round down, the integer value should be stored as -3
2’s compliment representation of -3: 11111111111111111111111111111101
hex representation of -3: 0xFFFFFFFD

6. (5) a) The jump instruction j loop transfers control to the instruction to which the label “loop” is attached. If the machine code for this jump instruction is located at memory address 0x40CE88C0. What is the highest address to which the label “loop” can correspond if this jump instruction is to work properly? Express you answer in hex.

Assuming the machine is using byte addressing, the loop label can be no closer than 4 addresses above the j loop instruction. Therefore, the label loop can be no higher than 0x40CE88C0 – 0x00000004 = 0x40CE88BC

(5) b) The conditional branch instruction beq $9,$8,exit transfers control to the instruction to which the label "exit" is attached if registers $9 and $8 contain equal values. If the machine code for this beq instruction is located at memory address 0x40CE88C0. What is the lowest address to which the label "exit" can correspond if the instruction is to work properly? Express you answer in hex.

Assuming the machine is using byte addressing, the exit label can be no fewer than 4 addresses below the beq instruction. Therefore, the label exit can be no lower than 0x40CE88C0 + 0x00000004 = 0x40CE88C4

7. Appendix A contains the description of each of the instructions mentioned below. Use those descriptions to answer the following instructions:
(5) a) What 32-bit pattern (expressed using 8 hex digits) is placed into register $5 by the instruction lui $5,0x9AE3 ? Register $5 initially contains the two's complement representation of -2.

2's compliment representation of -2 = 11111111111111111111111111111110 = 0xFFFFFFFE
$5 = 0x9AE30000

(5) b) What 32-bit pattern (expressed using 8 hex digits) is placed into register $5 by the instruction addi $5,$5,0x9AE3 ? Register $5 initially contains the two's complement representation of -2.

0xFFFFFFFE + 0x00009AE3 -> $5 = 0x00009AE1

(5) b) What 32-bit pattern (expressed using 8 hex digits) is placed into register $5 by the instruction addiu $5,$5,0x9AE3 ? Register $5 initially contains the two's complement representation of -2.

0xFFFFFFFE + 0x00009AE3 -> $5 = 0x00009AE1

(5) c) What 32-bit pattern (expressed using 8 hex digits) is placed into register $5 by the instruction srl $5,$5,3 ? Register $5 initially contains the two's complement representation of -2.

Shifting to the right by 3 and backfilling with 0's produces:
000111111111111111111111111111
= 0x1FFFFFFF

(5) c) What 32-bit pattern (expressed using 8 hex digits) is placed into register $5 by the instruction sra $5,$5,3 ? Register $5 initially contains the two's

complement representation of -2.

0xFFFFFFFF

8. (9) The variable name X corresponds to a 32-bit memory word that contains some integer in two's complement form. The address of the memory word is 0x100400CC. Write down a series of instructions (containing only MIPS true-ops) that adds the contents of the variable X to the contents of register $5 and places the sum back into register $5. Recall that each built-in true-op instruction (unlike pseudo-instructions) corresponds to a single machine instruction.

```
lw $t0,4(100400CC)          # get the value of the memory word
sll $t1,$t0,16              # $t1 is the most significant 16 bits of X
srl $t2,$t0,16
sll $t2,$t2,16              # $t2 is the least significant 16 bits of X
sll $t3,$5,16               # $t3 is the most significant 16 bits of $t5
srl $t4,$5,16
sll $t4,$t4,16              # $t4 is the least significant 16 bits of $t5
addu $t6,$t2,$t4            # $t6 is the addition of the least significant bits
sltu $t7,$t6,$t4            # get the "carry" of the least significant bits
addu $t8,$t1,$t3            # $t8 is the addition of the most significant bits
addu $t9,$t8,$t7            # add the addition of the carry to $t8
sll $t9,$t9,16              # shift $t9 over 16 bits as most significant bits
or $5,$t6,$t9               # $t5 will have the addition of X and $5
```

9. (6) Show the true-op assembly language instructions to which the MIPS pseudo-instruction la $10,Y corresponds. This pseudo-instruction places the address of Y into register $10. Assume that the address of Y is 0x87659324.

```
lui $at,4097              #0x1001 -> upper 16 bits of $at
ori $10,$at,0x00000000    # assuming 0 data bit is at 0x00000000
```

b) (5) Why is the instruction lw $4,X not a valid MIPS true-op instruction?

X is a variable in this case, and lw requests the address of the value. The value of X may be outside the range of memory and throw an error.

10. (5) What 32-bit pattern (expressed using 8 hex digits) is placed into register $5 by the instruction xori $5,$5,0x9AE3? Register $5 initially contains the two's complement representation of -2.

0xFFFFFFFE – 0x00009AE3 = 0xFFFF651c