Computer Science 605.611

Problem Set 2 Answers

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
Looking up this instruction in appendix A reveals that it is an R-type shift left logical instruction with opcode=0. The rs and rt registers are both 0 and the shift amount is 0. The unused fields within the machine instruction are 0. Hence the 32-bit machine instruction in hex is 0x00000000.

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.
The corresponding binary pattern is 1 00011010 11010100000000000000000
So the sign is negative, the characteristic is decimal 26 which corresponds to an exponent of 26-127 = -101. Therefore the pattern represents – 1.110101 * 2^-101 = -1.D4 * 2^-101 = -1D4 *2^-8 * 2^-101 = - 468 * 2^-109 = - 7.21 * 10^-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.
a) a one's complement integer (if yes, write the integer in decimal) Yes. Since the MSB is 0 this represents a positive value +880317954. If the MSB had been 1, we would complement the pattern to determine what it represents.

b) a two's complement integer (if yes, write the integer in decimal) Yes. Since the MSB is 0 this represents a positive value +880317954. If the MSB had been 1, we would complement the pattern to determine what it represents. Note that when the MSB is 0 the value has the same representation in both two's and one's complement.

c) a 32-bit IEEE 754 floating point number (if yes, write the number in decimal) Yes. The MSB is 0 so the sign is positive. The next 8 bits (01101000) give the excess-127 exponent = 104 – 127 = -23. The next 23 bits (11110001001011000000010) give the fraction. So the value represented is + 1.F12C04 * 2^-23
= 1F12C04 * 2^-24 * 2^-23 = 1F12C04 * 2^-47 = 32582660 * 2^-47 = 2.315 * 10^-7

d) a MIPS machine instruction (if yes, show the assembly language instruction)
The 6-bit opcode = 001101 = 0xD so the instruction operator mnemonic is ori (from appendix A). This is an I-type instruction. The rs field contains 00011. The rt field contain 11000. The remaining 16 bits give the immediate operand. So the corresponding assembly language instruction is ori $24,$3,0x9602. So the answer is yes.

e) a sign & magnitude integer (if yes, write the integer in decimal)
Yes. The sign bit is 0 so this represents a positive integer = +880317954.
Note that positive integers have the same 32-bit representation in one's complement, two's complement and sign & magnitude systems. The representations only differ for negative integers.

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 = 0x3E000002
This CPU instruction simply copies the bit pattern without change from CPU register $4 to CP1 register $f4.

(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 .
This instruction uses truncation on $f6 to produce a whole number and places the two's complement representation of the integer into $f8. So the pattern produced in $f8 represents the negative value -2. So $f8 contains 0xFFFFFFFE.

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.
The jump instruction shifts the rightmost 26 bits in the machine instruction left 2 bits (i.e., multiplies it by 4) and prepends the upper 4 bits of the PC to the resulting 28 bits to generate the 32-bit jump address. Hence the highest address to which the label loop can correspond is 0x4FFFFFFC.

(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.
Conditional branch instructions such as beq use the rightmost 16 bits in the machine instruction as a signed number that indicates the number of instructions to branch forward (if positive) to higher addresses or backwards (if negative) to lower addresses. The most negative 16-bit number is -32768, so the instruction can branch backwards up to 32768 instructions. At runtime, this 16 bit displacement is shifted left 2 bits (i.e., multiplied by 4) and sign extended to 32 bits. The 32-bit signed integer is then added to the already incremented PC to generate the branch destination address (also called the branch target address). The most negative displacement, when added to the PC, generates the lowest target address. Hence the lowest target address is 0x40CE88C4 + (- 32768*4) = 0x40CE88C4 – 0x20000 = 0x40CC88C4.
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.
This instruction places 0x9AE30000 into $5. The low 16 bits are zeroed out.

(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.
This instruction adds the sign extended immediate operand to $5. Hence $5 will contain 0xFFFFFFFE + 0xFFFF9AE3 = 0xFFFF9AE1.
(-2 + -25885) = -25887

(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.
The add immediate unsigned still adds the sign extended immediate to register $5. Hence register $5 will contain 0xFFFF9AE1 in this case as well.

(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.
This is a logical right shift so $5 will contain 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.
This is an arithmetic right shift, so the sign is preserved by bringing in copies of the original sign bit on the left. Register $5 will contain 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.

```
lui     $1,0x1004
lw      $1,0xCC($1)
add     $5,$5,$1
```

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     $10,0x8765
ori     $10,$10,0x9324
```

b) (5) Why is the instruction lw $4,X not a valid MIPS true-op instruction?
Instructions that reference memory operands must use a base register and displacement to generate the operand address. There is no direct addressing for memory operands since the 32-bit address will not fit into the rightmost 16 bits of the instruction.

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.
The logical instructions (unlike the arithmetic) integer instructions zero extend the 16-bit immediate operand to a 32-bit pattern. Hence the result produced in register $5 is 0xFFFFFFFE XOR 0x00009AE3 = 0xFFFF651D .