Lebanese American University
School of Arts and Sciences
Department of Computer Science and Mathematics

CSC 320 – Computer Organization

# Problem Set 3: Instructions: Language of the Computer

**Exercise 1**

The following problems explore translating from C to MIPS. Assume that the variables f, g, h, and i are given and could be considered 32-bit integers as declared in a C program.

| | |
|---|---|
| a. | f = g - h; |
| b. | f = g + ( h - 5 ); |

**1.1** For the C statements above, what is the corresponding MIPS assembly code? Use a minimal number of MIPS assembly instructions.

Solution:

| | |
|---|---|
| a. | sub f, g, h |
| b. | addi f, h, -5 (note, no subi)<br>add f, f, g |

**1.2** For the C statements above, how many MIPS assembly instructions are needed to perform the C statement?

Solution:

| | |
|---|---|
| a. | 1 |
| b. | 2 |

**1.3** If the variables `f`, `g`, `h`, and i have values 1, 2, 3, and 4, respectively, what is the end value of `f`?

Solution:

| | |
|---|---|
| a. | -1 |
| b. | 0 |

The following problems deal with translating from MIPS to C. Assume that the variables g, h, i , and j are given and could be considered 32-bit integers as declared in a C program.

| | |
|---|---|
| a. | addi f, f, 4 |
| b. | add f, g, h<br>add f, i, f |

**1.4** For the MIPS assembly instructions above, what is a corresponding C statement?

Solution:

| | |
|---|---|
| a. | f = f + 4 |
| b. | f = g + h + i |

**1.5** If the variables f, g, h, and i have values 1, 2, 3, and 4, respectively, what is the end value of f?

Solution:

| | |
|---|---|
| a. | 5 |
| b. | 9 |

**Exercise 2**

The following problems deal with translating from C to MIPS. Assume that the variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively.

| | |
|---|---|
| a. | f = - g - A[4]; |
| b. | B[8] = A [ i - j ]; |

**2.1** For the C statements above, what is the corresponding MIPS assembly code?

Solution:

| | |
|---|---|
| a. | lw $s0, 16($s6)<br>sub $s0, $0, $s0<br>sub $s0, $s0, $s1 |
| b. | sub $t0, $s3, $s4<br>sll $t0, $t0, 2 |

| | |
|---|---|
| | add $t0, $s6, $t0<br>lw $t1, 0($t0)<br>sw $t1, 32($s7) |

**2.2** For the C statements above, how many MIPS assembly instructions are needed to perform the C statement?

Solution:

| | |
|---|---|
| a. | 3 |
| b. | 5 |

**2.3** For the C statements above, how many different registers are needed to carry out the C statement?

Solution:

| | |
|---|---|
| a. | 3 |
| b. | 6 |

**Exercise 3:**

The following problems deal with translating from MIPS to C. Assume that the variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively.

| | |
|---|---|
| a. | sll $s2, $s4, 1<br>add $s0, $s2, $s3<br>add $s0, $s0, $s1 |
| b. | sll $t0, $s0, 2<br>add $t0, $t0, $s6<br>lw $t1, 0($t0)<br>lw $t0, 4($t0)<br>add $t0, $t0, $t1<br>sll $t1, $s1, 2<br>add $t1, $t1, $s7<br>sw $t0, 0($t1) |

**3.1** For the MIPS assembly instructions above, what is the corresponding C statement?

Solution:

| | |
|---|---|
| a. | f = 2j + i + g; |
| b. | B[g] = A[f] + A[f + 1]; |

**3.2** How many registers are needed to carry out the MIPS assembly as written above? If you could rewrite the code above, what is the minimal number of registers needed?

Solution:

| | |
|---|---|
| a. | 5 |
| b. | 6 |

**Exercise 4**

The following problems explore number conversions from signed and unsigned binary numbers to decimal numbers.

| | |
|---|---|
| a. | 0100 1001 |
| b. | 1111 1011 |

**4.1** For the patterns above, what base 10 number does the binary number represent, assuming that it is a two's complement integer?

Solution:

| | |
|---|---|
| a. | 73 |
| b. | -5 |

**4.2** For the patterns above, what base 10 number does the binary number represent, assuming that it is an unsigned integer?

Solution:

| | |
|---|---|
| a. | 73 |
| b. | 251 |

**4.3** For the patterns above, what hexadecimal number does it represent?

Solution:

| | |
|---|---|
| a. | 49 |
| b. | FB |

The following problems explore number conversions from decimal to signed and unsigned binary numbers.

| | |
|---|---|
| a. | -1 |
| b. | 1024 |

**4.4** For the base ten numbers above, convert to 2's complement 16-bit binary system.

Solution:

| | |
|---|---|
| a. | 1111 1111 1111 1111 |
| b. | 0000 0100 0000 0000 |

**4.5** For the base ten numbers above, convert to 2's complement hexadecimal.

Solution:

| | |
|---|---|
| a. | FFFF |
| b. | 0400 |

**4.6** For the base ten numbers above, determine the 2’s complement hexadecimal representation of the negated values in the table.

Solution:

| | |
|---|---|
| a. | 0001 |
| b. | FC00 |

**Exercise 5**

In the following problems, the data table contains bits that represent the opcode of an instruction. You will be asked to translate the entries into assembly code and determine what format of MIPS instruction the bits represent.

| | |
|---|---|
| a. | 0x01084020 |
| b. | 0x02538822 |

**5.1** What binary number does the above hexadecimal number represent?

Solution:

| | |
|---|---|
| a. | 0000 0001 0000 1000 0100 0000 0010 0000 |
| b. | 0000 0010 0101 0011 1000 1000 0010 0010 |

**5.2** What decimal number does the above hexadecimal number represent?

Solution:

| | |
|---|---|
| a. | 17317920 |
| b. | 39028770 |

**5.3** What instruction does the above hexadecimal number represent?

| | |
|---|---|
| a. | opcode = 0, rs = 8, rt = 8, rd = 8, shamt = 0, funct = 32<br>⇨ add $t0, $t0, $t0 |
| b. | opcode = 0, rs = 18, rt = 19, rd = 17, shamt = 0, funct = 34<br>⇨ sub $s1, $s2, $s3 |

In the following problems, the data table contains the values of various fields of MIPS instructions. You will be asked to determine what the instruction is, and find the MIPS format for the instruction.

| | |
|---|---|
| a. | op= 0, rs=3, rt=2, rd=3, shamt=0, funct=0x22 |
| b. | op= 0x23, rs=1, rt=2, const=0x4 |

**5.4** What type (I-type, R-type) instruction do the instructions above represent?

Solution:

| | |
|---|---|
| a. | R-type |
| b. | I-type |

**5.5** What is the MIPS assembly instruction described above?

Solution:

| | |
|---|---|
| a. | sub $v1, $v1, $v0 |
| b. | lw $v0, 4($at) |

**5.6** What is the binary representation of the instructions above?

Solution:

| | |
|---|---|
| a. | 0x00621822 |
| b. | 0x8C220004 |

**Exercise 6**

For these problems, the table holds some logical operations that are not included in the MIPS instruction set. How can these instructions be implemented?

| | |
|---|---|
| a. | not $t1, $t2 //bit-wise invert |
| b. | orn $t1, $t2, $t3 //bit-wise OR of $t2, !$t3 |

**6.1** The logical instructions above are not included in the MIPS instruction set, but are described above. If the value of $t2 = 0x00FFA5A5 and the value of $t3 = 0xFFFF003C, what is the result in $t1?

Solution:

| | |
|---|---|
| a. | 0xff005a5a |
| b. | 0x00ffffe7 |

**6.2** The logical instructions above are not included in the MIPS instruction set, but can be synthesized using one or more MIPS assembly instructions. Provide a minimal set of MIPS instructions that may be used in place of the instructions in the table above.

Solution:

| | |
|---|---|
| a. | nor $t1, $t2, $t2 |
| b. | nor $t1, $t3, $t3<br>or $t1, $t2, $t1 |

**6.3** For your sequence of instructions in **6.2**, show the bit-level representation of each instruction.

Solution:

| | | |
|---|---|---|
| a. | nor $t1, $t2, $t2 | 000000 01010 01010 01001 00000 100111 |
| b. | nor $t1, $t3, $t3 | 000000 01011 01011 01001 00000 100111 |
| | or $t1, $t2, $t1 | 000000 01010 01001 01001 00000 100101 |

Various C-level logical statements are shown in the table below. In this exercise, you will be asked to evaluate the statements and implement these C statements using MIPS assembly instructions.

| | |
|---|---|
| a. | A = B \| !A; |
| b. | A = C[0] << 4; |

**6.4** The table above shows different C statements that use logical operators. If the memory location at C[0] contains the integer values 0x00001234, and the initial integer values of A and B are 0x00000000 and 0x00002222, what is the result value of A?

Solution:

| | |
|---|---|
| a. | 0xFFFFFFFF |
| b. | 0x00012340 |

**6.5** For the C statements in the table above, write a minimal sequence of MIPS assembly instructions that does the identical operation. Assume $t1 = A, $t2 = B, and $s1 is the base address of C.

Solution: Assuming $t1 = A, $t2 = B, $s1 = base of Array C

| | |
|---|---|
| a. | nor $t3, $t1, $t1<br>or $t1, $t2, $t3 |
| b. | lw $t3, 0($s1)<br>sll $t1, $t3, 4 |

**6.6** For your sequence of instructions in 6.5, show the bit-level representation of each instruction.

Solution:

| | | |
|---|---|---|
| a. | nor $t3, $t1, $t1<br>or $t1, $t2, $t3 | 000000 01001 01001 01011 00000 100111<br>000000 01010 01011 01001 00000 100101 |
| b. | lw $t3, 0($s1)<br>sll $t1, $t3, 4 | 100011 10001 01011 0000000000000000<br>000000 00000 01011 01001 00100 000000 |