CPE 431/531 Instructions – Language of the Computer Fall 2022

**The University of Alabama in Huntsville**

**ECE Department**

**CPE 431 01, CPE 531 01/01R**

**Instructions – Language of the Computer**

**Fall 2022**

**Due September 7, 2022**

***You must show your work to get full credit. The number In parentheses is the point value of the problem. The numbers in <> indicate the sections of the book that discuss this topic.***

1. **0(15), 2.0(5) 3.0(10), 4.0(5), 5.0 (10), 6.0(5), 7.0(15), 8.0.1(5), 8.0.2(5), 9.0(5), 10.0(5)**

**1.0** For the following C statement, what is the corresponding MIPS assembly code? 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. Also, assume that **A** and **B** are arrays of words.

**B[f] = B[i-j] + A[g]**

Sub $t0, $s3, $s4 //i-j

Sll $t0, $t0, 2 //t0 = 4\*temp

Add $t0, $t0, $s7 // temp = &B[i-j]

Sll $t1, $s1, 2 //temp1 = 4\*g

Add $t1, $t1, $s6 //temp1 = &A[g]

Add $t0, $t0, $t1 //temp = &(B[i-j] + A[g]

Sll $t2, $s0, 2 //temp2 = 4\*f

Sw $s0, $t2($s7) //stores temp to s7 at index 4\*f

**2.0** Translate 0xF806\_4020 from signed 2s complement representation to decimal.

1111\_0100\_0000\_0110 0100\_0000\_0010\_0000

**Answer : 4,161,159,200**

**3.0 <2.2, 2.3>** For the MIPS assembly instructions below, what is the corresponding C statement? 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** c **$s7**, respectively.

**sll $t0, $s0, 2** //temp = f\*4

**add $t0, $s6, $t0** //temp = &A[f]

**sll $t1, $s1, 2** //temp1 = g \*4

**add $t1, $s7, $t1** //temp1 = &B[g]

**lw $t2, 0($t0)** //temp2 = &A[f]

**addi $t0, $t0, 4** //temp = &A[f+1] bc 4\*1 = 4

**lw $t0, 0($t0)** //load temp with temp : t0 = &A[f+1]

**add $t0, $t0, $t2** //temp = &A[f+1] + &A[f]

**sw $t0, 0($t1)** //M[temp1] = temp : B[g] = A[f+1] + A[f]

**Answer: B[g] = A[f+1] + A[f]**

**4.0 <2.2, 2.3>** Translate the following MIPS code 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 and that **A** and **B** are arrays of words.

**addi $t0, $s6, 4** //temp = &A[1]

**lw $t0, 0($t0)** //temp = A[1]

**add $t0, $t0, $t0** //temp = A[1] + A[1]

**sw $t0, 0($s6)** //A[1] = A[1] + A[1]

**Answer: A[1] = A[1] + A[1] or A[1]\*2**

**5.0 <2.4>** Assume that **$s0** holds the value -1450ten.

X + (-1450) > 2^31 -1

X + (-1450) < -2^31

**?5.0.1 <2.4>** For the instruction **add $t0, $s0, $s1**, what is the range(s) of values for $s1 that would result in overflow?

-2­31 to 231 – 1 : -2,147,483,648 to 2,147,483,647

-1450 + (-2,147,483,648) = -2,147,485,098 overflowed caused by the lower limit

-1450 + 2,147,483,647 = 2,147,482,197 is below the positive threshold

-2,147,485,098 – (-2,147,483,648) = 1450 (its overflowed by this amount)

-2,147,483,648 + 1450 = -2,147,482,198 (+1 will overflow us)

-1450 + (-2,147,482,198) = -2,147,483,648 (checking that hit the lower limit)

**Answer:** **[-2,147,483,648 to** **- 2,147,482,198]**

**?5.0.2 <2.4>** For the instruction **sub $t0, $s0, $s1**, what is the range(s) of values for $s1 that would result in overflow?

-2­31 to 231 – 1 : -2,147,483,648 to 2,147,483,647

-1450 – 2,147,483,647 = -2,147,485,097 (upper limit causes an overflow) by

-2,147,485,097 -2,147,483,648 = 1449.

2,147,483,647 – 1449 = 2,147,482,198 +1 makes it overflow at 2,147,482,199

Therefore, 2,147,482,198 is the upper limit before an overflow

-1450 – (-2,147,483,648) = 2,147,482,198 which isn’t overflow on the positive end but the max negative number we can represent

**Answer: [2,147,482,198 to 2,147,483,647]**

**6.0 <2.2, 2.5>** Provide the type and assembly language instruction for the following binary value:

**0000 0010 0001 0000 1000 0000 0010 0000two**

**000000\_10000\_10000\_10000\_00000\_100000two**

0\_16\_16\_16\_0\_32

**Answer: add $s0, $s0, $s0 R-type**

**7.0 <2.6>** Assume the following register contents: **$t0 = 0x1357\_9BDE**, **$t1 = 0x8697\_51CA**

1357\_9BDE: 0001-0011-0101-0111\_1001-1001-1101-1110

Shift 4 left: 0011-0101-0111-1001\_1001-1101-1110-0000

Shift 3 right: 0000-0010-0110-1010\_1111-0011-0011-1011

FFEF: 0000-0000-0000-0000\_1111-1111-1110-1111

-1: 1111-1111-1111-1111\_1111-1111-1111-1111

8697\_51CA: 1000-0110-1001-0111\_0101-0001-1100-1010

OR: 1011-0111-1111-1111\_1101-1101-1110-1010

Andi: 0011-0101-0111-1001\_1001-1101-1110-0000

Andi 7.3: 0000-0000-0000-0000\_1111-0011-0010-1011

**7.0.1** **<2.6>** For the register values given, what is the value of **$t2** for the following sequence of instructions?

**sll $t2, $t0, 4**  //t2 = 357B\_DE00

**or $t2, $t2, $t1** //B7FF\_DDEA

**Answer: 0xB7FF\_DDEA**

**7.0.2** **<2.6>** For the register values given, what is the value of **$t2** for the following sequence of instructions?

**sll $t2, $t0, 4**  //t2 = 357B\_DE00

**andi $t2, $t2, -1** //3579\_9DE0

**Answer: 0x3579\_9DE0**

**7.0.3** **<2.6>** For the register values given, what is the value of **$t2** for the following sequence of instructions?

**srl $t2, $t0, 3** //t2 = 026A\_F33B

**andi $t2, $t2, 0xFFEF** //F32B

**Answer: 0xF32B**

**8.0 <2.7>** Consider the following MIPS loop:

**LOOP: slt $t2, $0, $t1** //set t2 if 0 <= t1

**beq $t2, $0, DONE** //branch if t2 = 0

**subi $t1, $t1, 1** //t1 = t1-1

**addi $s2, $s2, 2** //s2 = s2 + 2

**j LOOP** //loop

**DONE:**

**8.0.1 <2.7>** Assume that the register **$t1** is initialized to the value 25. What is the value in register **$s2** assuming **$s2** is initially 1000?

The addi will run 25 times. 25\*2 = 50 + s2(initial) = 1050

**Answer: 1050**

**8.0.2 <2.7>** For the loop written in MIPS assembly above, assume that the register **$t1** is initialized to the value N. How many MIPS instructions are executed?

5 instructions. First two run once we get to zero.

**Answer: 5N + 2 instructions are executed**

**?9.0 <2.6, 2.10>** If the current value of the PC is 0x57F0\_3280, can you use a single jump instruction to get to the PC address 0x5700\_3291? If so, give the address value for the jump instruction.

1475359360 jump to 1459630737. Need to jump by -15,728,623 = FF10\_0011

26 bits left to specify a jump technically 28 (26 bits << 2 = 28 bits)

Jump address \* 4 should give you location of where you’re going

57F0\_3280: 0101-0111-1111-0000\_0011-0010-1000-0000

5700\_3291: 0101-0111-0000-0000\_0011-0010-1001-0001

FF10\_0011: 1111-1111-0001-0000\_0000-0000-0001-0001

Right Shift: 0011-1111-1100-0100\_0000-0000-0000-0100 (no remove first)

3FC4\_0004

**Answer: No, you could not use a single jump instruction because the address is more than 26 bits**

**?10.0 <2.6, 2.10>** If the current value of the PC is 0x1FFF F000, can you use a single branch instruction to get to the PC address 0xFFFE\_52AC? If so, give the offset value for the branch instruction.

16 bits to specify a branch

536,866,816 to 4,294,857,388. Branch by -3,757,990,572

PC + address = word

1FFF\_F000

FFFE\_52AC

FFFF\_FFFF\_2001\_9D54

**Answer: No you would have to load the first 16 bits and last 16 bits to a register. Then add those to get the full address then branch to that.**