Department of Electrical and Computer Engineering

*University of Wisconsin – Madison*

ECE 552 Introductions to Computer Architecture

Homework #2 (Due Tue. 2/6)

1. *Translating C statement to MIPS assembly code*

In the following sub-problems, assume variables **f,** **g, h, i,** and **j** are stored in registers $s0, $s1, $s2, $s3, and $s4 respectively. Assume that the base address of arrays **A** and **B** are in registers $s6, and $s7 respectively. For the C statement in each sub-problem, translate it into corresponding MIPS code using minimum number of assembly instructions

1. f = g + (-f -5)
2. B[8] = A[i-j] (i ≥ j)
3. *Translating MIPS assembly code to C statements*

In the following sub-problems, assume variables **f,** **g, h, i,** and **j** are stored in registers $s0, $s1, $s2, $s3, and $s4 respectively and their values are multiples of 4. Assume that **&A**, the base address of arrays **A,** and **&B**, the base address of array **B** are in registers $s6, and $s7 respectively. For the MIPS assembly codes in each sub-problem, write down the corresponding C statements.



**add $t0, $s6, $s0**

**add $t1, $s7, $s1**

**lw $s0, 0($t0)**

**addi $t2, $t0, 4**

**lw $t0, 0($t2)**

**add $t0, $t0, $s0**

**sw $t0, 0($t1)**

**addi $t0, $s6, 4**

**add $t1, $s6, $0**

**sw $t1, 0($t0)**

**lw $t0, 0($t0)**

**add $s0, $t1, $t0**

1. *Simulating MIPS assembly programs*
2. Assume that the contents of registers $t0 and $t1 are [$t0] = 0xF00DD00D, and [$t1] = 0x11111111 respectively. What is the value $t2 after execution of the following sequence of MIPS instructions?

**sll $t2, $t0, 4**

**addi $t2, $t2, -1**

1. Suppose that [$t0] = 0x5FBE4000, and [$t1] = 0x3FF80000. What is the value of $t2 after execution of the following instructions?

**slt $t2, $0, $t0**

**bne $t2, $0, ELSE**

**j DONE**

**ELSE: addi $t2, $t2, 2**

**DONE:**

1. *Pseudo-instructions*

Consider a hypothetical MIPS pseudo-instruction

**orn rd, rs, rt**

that will perform bit-wise or of rs and ~(rt), the bit-wise complement of rt, and put result into rd. Using minimum number of MIPS instructions to synthesize this pseudo-instruction.

1. *Branch instructions*
2. Consider the following sequence of MIPS instructions

**addi $t1, $0, $0**

**LOOP: lw $s1, 0($s0)**

**add $s2, $s2, $s1**

**addi $s0, $s0, 4**

**addi $t1, $t1, 1**

**slti $t2, $t1, 100**

**bne $t2, $0, LOOP**

Translate these instructions to a C program segment assuming integer I is in register $t1, and $s2 holds the integer result, and $s0 holds the base address of the integer MemArray.

1. Rewrite the loop to reduce the MIPS instruction executed.