CSE 315: Computer Organization

Sheet 3

1. For each of the following multiple choice questions, choose the one best answer and briefly explain why you chose it

I) A character x can be stored at location _____ in the memory of a MIPS processor.

a) 1007_{10}	b) $FAFC_{16}$	c) 111100_2	**d) all the previous**

II) An instruction for a MIPS processor can be stored at location _____ in the memory.

a) 4_{10}	b) $FAFCFFFF00_{16}$	c) 111101_2	d) all the previous

III) The destination in a MIPS arithmetic instruction is _____.

a) a memory location	**b) a register**	c) the stack	d) the instruction

IV) The destination in a MIPS sw instruction is _____.

a) a memory location	b) a register	c) the stack	d) the instruction

V) Of the following, _____ is an immediate instruction.

a) add	**b) lui**	c) lw	d) sub

VI) Of the following, _____ is a data transfer instruction.

a) add	b) lui	**c) lw**	d) sub

VII) Of the following, _____ has no immediate version.

a) add	b) and	c) or	**d) sub**

VIII) R-type instructions can use up to _____ register(s).

a) one	b) two	**c) three**	d) four

IX) The MIPS instruction lui can be equivalent to an addi instruction followed by an _____ instruction.

a) sll	b) srl	c) sra	d) all the previous

X) The minimum number of MIPS assembly instructions needed to write `A[100] = A[0] + 100`, where A is an integer array with base saved in $s0, is _____ instruction(s).

a) one	b) two	**c) three**	d) five

XI) The minimum number of MIPS assembly instructions needed to write `A[i] = A[i] + 100`, where A is an integer array with base saved in $s0, is _____ instruction(s).

a) one	b) two	c) three	**d) five**

XII) The minimum number of MIPS assembly instructions needed to write `f = a + b + c - d`, where a, b, c, d, and f are all in registers, is _____ instruction(s).

a) one	b) two	**c) three**	d) five

XIII) The minimum number of MIPS assembly instructions needed to write `if (x<y) then z = 1; else z = 0`, where x, y, and z are all in registers, is _____ instruction(s).

a) one	b) two	c) three	d) five

XIV) The largest immediate operand for a MIPS addi instruction is bounded by _____.

a) 2^{12}	**b) 2^{15}**	c) 2^{16}	d) 2^{32}

XV) The largest immediate operand for a MIPS andi instruction is bounded by _____.

a) 2^{12}	b) 2^{15}	**c) 2^{16}**	d) 2^{32}

XVI) If we need to set the least significant 2 bits of register $s0, the mask should be _____.

a) 3_{10}	b) $FFFFFFFC_{16}$	c) 11_8	d) none of the previous

XVII) If we need to set the least significant 2 bits of register $s0, we should use a MIPS _____ instruction.

a) and	b) or	c) ori	**d) Answer (b) or Answer (c)**

XVIII) If we need to set the most significant 2 bits of register $s0, we should use a MIPS _____ instruction.

a) and	**b) or**	c) ori	d) Answer (b) or Answer (c)

2. For the following C-Level logical statements, if the memory location at $C[0]$ contains the integer value $0x00001234$, and the initial integer value of A and B are $0x00000000$ and $0x00002222$, what is the result value of A in each case? Write a minimal sequence of MIPS assembly instructions that does the identical operation. Show the bit-level representation for the instructions you wrote.

a. $A = B \,\&\, C[0];$

b. $A = A\ ?\ B : C[0];$

Solution:

The result value of A:

a. $0x00000220$
b. $0x00001234$

The equivalent MIPS instructions:
Assuming $\$t1 = A$, $\$t2 = B$, and $\$s1 = base\ of\ Array\ C$,

a. $lw\ \$t3,0(\$s1)$
   $and\ \$t1,\$t2,\$t3$

b.
```
        beq $t1,$0,ELSE
        add $t1,$t2,$0
        j  END
ELSE: lw $t1,0($s1)
END:
```

The bit-level representation of the instructions:

a. $lw\ \$t3,0(\$s1)$	100011 10001 01011 0000000000000000
$and\ \$t1,\$t2,\$t3$	000000 01010 01011 01001 00000 100100
b. $beq\ \$t1,\$0,ELSE$	000100 01001 00000 0000000000000010
$add\ \$t1,\$t2,\$0$	000000 01010 00000 01001 00000 100000
$j\ END$	000010 00000000000000000000000001
$ELSE: lw\ \$t1,0(\$s1)$	100011 10001 01001 0000000000000000
$END:$	

3. Assume that A is a 10-integer array whose starting address is in register $s0. The values 1 to 10 are stored in A in ascending order. Assume that B is another 10-integer array whose starting address is in register $s1. $s0 and $s1 contain the values 1000 and 2000 respectively. Consider the following MIPS code:

```
      add   $t0, $s0, $zero
      addi  $t1, $zero, 9
      sll   $t1, $t1, 2
      add   $t1, $t1, $s1
L1:   lw    $t2, 0($t0)
      sw    $t2, 0($t1)
      addi  $t0, $t0, 4
      addi  $t1, $t1, -4
      slt   $t3, $t1, $s1
      beq   $t3, $zero, L1
```

a. What are the contents of $t0, $t1, $t2, $t3 at the end of the program above?
b. What are the contents of arrays A and B at the end of the program?

c. How many instructions were executed in total?
d. Does the above code correspond to accessing arrays using indices or using pointers?

Solution:

a. $t0 = 1040, $t1 = 1996, $t2 = 10, $t3 = 1
b. Array A contains the values {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
   Array B contains the values {10, 9, 8, 7, 6, 5, 4, 3, 2, 1}
c. The number of instructions executed = 4 + 6 * 10 = 64 instructions
d. It corresponds to using pointers

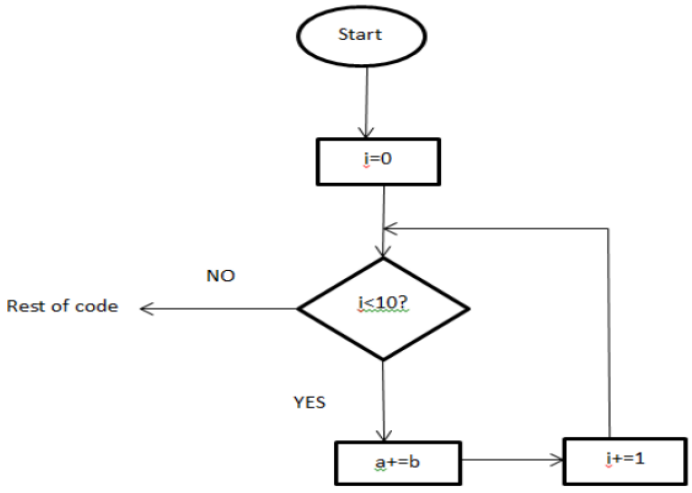
4. For the each of the following C code statements, draw a control-flow graph of the C code. Translate the C code to MIPS assemble code, using the minimum number of instructions, Assume that the value a, b, i are in registers $\$s0, \$s1, \$t0$, respectively. Also, assume that register $\$s2$ holds the base address of the array D. How many MIPS instructions does it take to implement the C code? If the variables a and b are initialized to 10 and 1 and all the elements of D are initially 0, what is the total number of MIPS instructions that is executed to complete the loop?

a. $for\ (i = 0; i < 10; i++)\ a += b;$
b. $while\ (a < 10)\{$
   $D[a] = b + a;\ a += 1;$
   $\}$

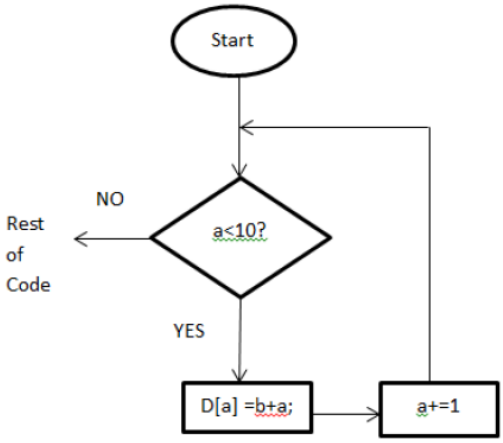
Solution:

The Control-Flow graph:

a.



b.



The equivalent MIPS instructions:

a.
$$addi\ \$t0, \$0, 0$$
$$j\ TEST$$
$$LOOP: add\ \$s0, \$s0, \$s1$$
$$addi\ \$t0, \$t0, 1$$
$$TEST: slti\ \$t2, \$t0, 10$$
$$bne\ \$t2, \$0, LOOP$$

OR

$$addi\ \$t0, \$0, 0$$
$$LOOP: slti\ \$t2, \$t0, 10$$
$$beq\ \$t2, \$0, END$$
$$add\ \$s0, \$s0, \$s1$$
$$addi\ \$t0, \$t0, 1$$
$$j\ LOOP$$
$$END:$$

b.
$$LOOP: slti\ \$t2, \$s0, 10$$
$$beq\ \$t2, \$0, DONE$$
$$add\ \$t3, \$s1, \$s0$$
$$sll\ \$t2, \$s0, 2$$
$$add\ \$t2, \$s2, \$t2$$
$$sw\ \$t3, 0(\$t2)$$
$$addi\ \$s0, \$s0, 1$$
$$j\ LOOP$$
$$DONE:$$

The number of MIPS instructions:

a. 6 instructions to implement and 44 instructions executed (Or 53 instructions in case of the second solution)
b. 8 instructions to implement and 2 instructions executed

5. For the following MIPS assembly code fragments, what is the total number of MIPS instructions executed? Translate the loops below to C code. Assume that i is held in register $\$t1, \$s2$ holds $result$, and $\$s0$ holds the base address of the integer array $MemArray$. Rewrite the MIPS code to reduce the number of MIPS instructions executed.

a.
$$addi\ \$t1, \$0, 100$$
$$LOOP: lw\ \$s1, 0(\$s0)$$
$$add\ \$s2, \$s2, \$s1$$
$$addi\ \$s0, \$s0, 4$$
$$subi\ \$t1, \$t1, 1$$
$$bne\ \$t1, \$0, LOOP$$

b.
$$addi\ \$t1, \$s0, 400$$
$$LOOP: lw\ \$s1, 0(\$s0)$$
$$add\ \$s2, \$s2, \$s1$$
$$lw\ \$s1, 4(\$s0)$$
$$add\ \$s2, \$s2, \$s1$$
$$addi\ \$s0, \$s0, 8$$
$$bne\ \$t1, \$s0, LOOP$$

Solution:

The total number of MIPS instructions:

a. 501
b. 301

The equivalent C code:

a.
```
for(i=100; i>0; i--){
    result+=*MemArray;
    MemArray++; }
OR
a=0;
for(i=100; i>0; i--){
    result+=MemArray[a];
    a++; }
```

b. *for(i=0; i<100; i+=2){*
result+=MemArray[i];
result+=MemArray[i+1]; }

The reduce MIPS code:

a. $addi\ \$t1, \$s0, 400$
$LOOP: lw\ \$s1, 0(\$s0)$
$add\ \$s2, \$s2, \$s1$
$addi\ \$s0, \$s0, 4$
$bne\ \$s0, \$t1, LOOP$

b. Already reduced to minimum instructions

6. We want to implement the following C-language code segment in MIPS assembly using a JumpTable. Assume a is in $s0, b is in $s1, and c is in $s2.

```
switch (a) {
    case 0:     b = b + a; break;
    case 2:     b = b - a;
    case 3:     c = b - a; break;
    case 5:
    case 6:     c = b + a; break;
    default:    b = c + a;
}
```

a. Show the contents of the Jump Table (Hint: Show only the labels corresponding to the addresses stored in the table.
b. What is the compiled MIPS assembly code for the above code segment? Assume that the base of the JumpTable is in register $t4.

Solution:

a.

L0
Ld
L2
L3
Ld
L6
L6

b. Assume register $t2 contains number 7.

```
        slt     $t3,    $s0, $zero      # test if a < 0
        bne     $t3,    $zero, Ld       # exit if a < 0
        slt     $t3,    $s0, $t2        # go to default if a < 7
        beq     $t3,    $zero, Ld       # go to default if a ≥ 7
        sll     $t1,    $s0, 2          # $t1 = 4*a
        add     $t1,    $t1, $t4        # JumpTable[a] address
        lw      $t0,    0($t1)          # $t0 = JumpTable[a]
        jr      $t0
L0:     add     $s1,    $s1, $s0        # b = b + a
        j       Exit                    # break
L2:     sub     $s1,    $s1, $s0        # b = b - a
L3:     sub     $s2,    $s1, $s0        # c = b - a
        j       Exit                    # break
L6:     add     $s2,    $s1, $s0        # c = b + a
        j       Exit                    # break
Ld:     add     $s1,    $s2, $s0        # b = c + a
Exit:
```