CMSC 411.05 Homework Assignment 2 Due: Oct 3, 2022, 2:29 pm

**Question 1** (10 points)

Convert the following C statements to equivalent MIPS assembly language. Assume that the variables f, g, h and j are assigned to registers $s0, $s1, $s2 and $s3 respectively. Assume that the base address of the array A and B are in registers $s6 and $s7 respectively.

a) f=g+h+B[5]

b) f=g–A[B[3]]

**Question 2 (10 points)**

Translate the following machine code to MIPS:

1010 11 10 000 0 1011 0000 0000 0000 0100

**Question 3 (10 points)**

Convert the following C fragment to equivalent MIPS assembly language. Assume that the variables a and b are assigned to registers $s0 and $s1 respectively. Assume that the base address of the array D is in register $s2.

while(a < 10){

D[a] = b + a;

a += 1;

}

**Question 4 (20 points, i.e. 2 points for each sub-task)**

Assume we are using the Big-Endian notation. Register $t0 contains 0x10000000 and $s0 contains 0x01234567. Suppose a portion of memory contains the following data

Address Data

0x10000000 0x12345678

0x10000004 0x9ABCDEF0

1. Complete the DATA column of following table

| Address | DATA |
| --- | --- |
| 0x10000000 |  |
| 0x10000001 |  |
| 0x10000002 |  |
| 0x10000003 |  |
| 0x10000004 |  |
| 0x10000005 |  |
| 0x10000006 |  |
| 0x10000007 |  |

Show the effects on memory and registers of the following instructions. **Assume each of the following instructions is executed independently of the others, starting with the values given above.**

1. lw $t1, 0($t0)
2. lw $t2, 4($t0)
3. lb $t3, 0($t0)
4. lb $t4, 4($t0)
5. lb $t5, 3($t0)
6. lh $t6, 4($t0)
7. sw $s0, 0($t0)

| Address | DATA |
| --- | --- |
| 0x10000000 |  |
| 0x10000001 |  |
| 0x10000002 |  |
| 0x10000003 |  |
| 0x10000004 |  |
| 0x10000005 |  |
| 0x10000006 |  |
| 0x10000007 |  |

1. sb $s0, 4($t0)

| Address | DATA |
| --- | --- |
| 0x10000000 |  |
| 0x10000001 |  |
| 0x10000002 |  |
| 0x10000003 |  |
| 0x10000004 |  |
| 0x10000005 |  |
| 0x10000006 |  |
| 0x10000007 |  |

1. sb $s0, 7($t0)

| Address | DATA |
| --- | --- |
| 0x10000000 |  |
| 0x10000001 |  |
| 0x10000002 |  |
| 0x10000003 |  |
| 0x10000004 |  |
| 0x10000005 |  |
| 0x10000006 |  |
| 0x10000007 |  |

**Question 5 (20 points)**

Convert the following program into machine. Note that the first column lists where these instructions are stored in the memory.

| 0xFC00000C  0xFC000010  0xFC000014  0xFC000018  0xFC00001C | start:  loop: addi $t0,$t0,-1  sw $t0, 4($t2)  bne $t0, $t3, loop  j start |
| --- | --- |