113 計算機組織 Homework 2_2 Due date:2024/10/21

- You must write down / type the calculations, or you won't get the scores.
- 請用手寫於 A4 紙上,並掃描上傳至北科 I 學園 plus
- 請勿抄襲，抄襲者與被抄襲者單一分數取平均

一、(15%)Translate the following C code 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. Assume that the elements of the arrays A and B are 4-byte words:

A[i+3] = A[4] - B[j+5];

二、(10%)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.

```
addi $t0, $s6, 16
lw      $t0,0($t0)
sll     $t0,$t0,3
sll     $t1,$s1,2
add     $t1,$t1,$s7
lw      $t1,0($t1)
sll      $t1,$t1,1
add     $t0,$t0,$t1
add     $t0,$t0,$s2
addi    $s4,$t0,-4
```

三、(10%)Provide the type and hexadecimal representation of following instruction:

(1)(5%) lw $t1, 20($s1)

(2)(5%) add $t0, $s3, $t5

四、(15%)Provide the type(5%), assembly language instruction(5%), and binary representation(5%) of instruction described by the following MIPS fields:

op=35, rs=11, rt=20, imm= 5000

五、(10%)Provide the type(5%) and assembly language instruction(5%) for the following hexadecimal value: $0xAD6D0022_{hex}$

六、(20%)Assume that we would like to expand the MIPS register file to **256 registers** and expand the instruction set to contain **two times** as many instructions.

How this would this affect the size of each of the bit fields in the **R-format(10%)** and **I-format(10%)** instructions?

七、(10%)For the following C statement, write a minimal sequence of MIPS assembly instructions that does the identical operation. Assume that **$s0 = a**, **$s1 = b**, and **$s2** is the base address of the array **C**, and the elements of the arrays C is 4-byte words.

b = C[a+8] >> 2;

八、(10%)For the following MIPS Code, If we assume we place the loop starting at location 80000 in memory, what is the MIPS machine code for this loop ? (Please fill in table by decimal)

```
Loop:  sll $t1, $s3, 2
       add $t1, $t1, $s6
       lw $t0, 0($t1)
       bne $t0, $s5, End
       addi $s3, $s3, 1
       j Loop
End:
```

0			9		0
0				0	
		8	0		
5					
8			1		