Computer Architecture

**Fall, 2022**

**Week 3**

**2021.09.26**

**組別：＿＿＿＿＿　簽名：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿**

[group1]

For the following two C statements, what are the corresponding MIPS assembly codes? Assume that the variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4, respectively.

1. f = g – h;
2. f = h + (g - 3);

j = f;



[group7]

1. Why registers are easier for compiler to use?
2. Why is the memory address represented by pointer and offset in the data transfer instructions?
3. What are the definition of “Spilling” and “Alignment”?



[group4]

Based on MIPS instruction set architecture, which of the following statements are true?

1. There are 32 general purpose registers and 32 floating point registers.



1. If we want to access sequential data in memory by word address, 872 and 954 would be accessed.



1. If our program frequently uses a certain constant x, we can save x into the $zero register to speed up the whole program.



1. The desired memory address is the sum of an offset (in byte) and a register containing a pointer to memory.



[group12]

In this week’s online video, we have learned three design principles.Please answer principle’s names and give an example for each one.



[group10]

True or False and explain if it is false.

1. There are 32 MIPS registers and each is 32 bits wide.



1. The content in $s1 is 6, and the content in $0 after the following instruction will be 10.



addi $0,$s1,4



1. To implement the following code($s1 : b, $s2 : c, $s3 : base on address of A) in C language:



c = b + A[4]

First, we need to transfer memory to register first by

lw $t0, 4($s3) #$t0 receives A[4]



Next, add it to b and replace c



add $s1, $s2, $t0

1. Registers stored in the processor can improve code density since compared to memory location, registers are named with fewer bits.



1. We can keep hardware simple via regularity, e.g. keeping syntax rigid. The benefit is that the implementation will be simpler.



1. lw(load word) means to transfer word from register to memory; sw(store word) means to transfer word from memory to register.



1. Assembly language uses variables to operate just like high-level language.



[group13]

Please select the correct options.

1. It’s more difficult to access memory than register. We use base register and offset to access memory efficiently.



使用memory比register 更困難，因此我們用base register和offset 會使memory更有效率

1. If assembly operands have been in memory, it's NOT necessary to load them to registers.



不需要load assembly operands 到register

1. Even if we define a fixed and finite number of variables in high level language, the compiler may NOT use the same number of variable registers.



即使我們在hign level language中定義了固定且有限數量的變數，但是在compiler中可能不會使用相同數量的變數在 registers。

1. Memory alignment helps access memory more efficiently in most case.



Memory alignment 幫助我們更有效率的使用memory

1. To reduce the usage of hardware resources, MIPS instructions may have different length.



為了減少hardware resources，MIPS instruction 會有不同的長度

1. lw, sw, ..., are data transfer instructions. When we talk about load and store instructions, the direction of data flow is viewed from the aspect of register.



Lw,sw 是data轉換的instruction 我們所說的load和store 是根據register的角度來看

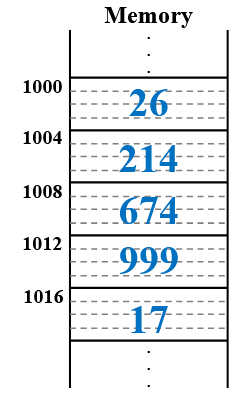
1. If registers are NOT full used, the least used variable can still stay in registers.



如果register沒有全部使用，最常使用的variable 會留在register



[group14]

Assume $s0 = 1000. Given the memory table M and following instructions, what is the value of M[1016]?

lw $t0, 0($s0)



lw $t1, 4($s0)



add $t2, $t0, $t1



sw $t2, 16($s0)

