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;

1. Ans: sub $s0, $s1, $s2

2. Ans: addi $t0, $s1, -3
   add $s0, $s2, $t0
   add $s4, $s0, $zero

[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”?

1. 比 memory 更快速，有 temporary storage
   can improve code density

2. 可以更快速執行 而不須用 32 bits 指向 memory address

3. spilling：所用變數超過 registers 數目
   Alignment：記憶體對齊，資料從 4 的倍數開始擺

[group4]

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

√ a. There are 32 general purpose registers and 32 floating point registers.

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

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

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

b. 需為 4 的倍數

c. $zero 的值必為零

[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.

1. Simplicity favors regularity: 簡單有益於規則，可提升速度並降低成本，例如原本找到 memory 要 32 bits，利用 registers 可降為 5 bits

2. smaller is faster: 小就是快，有 32 bits wide，若使 registers 增加為 64 個則需用 6 bits，這樣可能速度會下降

3. Make the common case fast: 設定新的 instruction 專門處理常發生的 case

[group10]

True or False and explain if it is false.

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

F 2. The content in $s1 is 6, and the content in $0 after the following instruction will be 10.

addi $0,$s1,4 $0 always =0

F 3. 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]
```

4x4=16 bytes

Next, add it to b and replace c

```
add   $s1, $s2, $t0
```

T 4. Registers stored in the processor can improve code density since compared to memory location, registers are named with fewer bits.

T 5. We can keep hardware simple via regularity, e.g. keeping syntax rigid. The benefit is that the implementation will be simpler.

F 6. lw(load word) means to transfer word from register to memory; sw(store word) means to transfer word from memory to register.

F 7. Assembly language uses variables to operate just like high-level language.

需要從 Memory load 出來用 $s0, $s1... 等 operate

[group13]

Please select the correct options.

T 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

F 2. If assembly operands have been in memory, it's NOT necessary to load them to registers.
不需要 load assembly operands 到 register

T 3. 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。

T 4. Memory alignment helps access memory more efficiently in most case.
Memory alignment 幫助我們更有效率的使用 memory

F 5. 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

T 6. 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

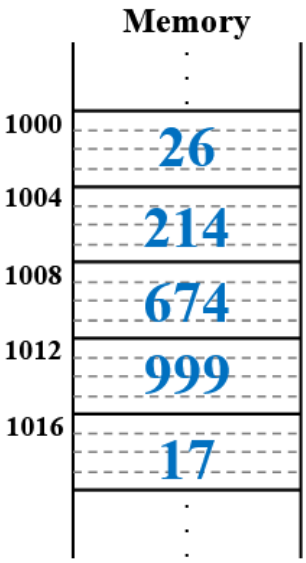
T 7. If registers are NOT full used, the least used variable can still stay in registers.
如果 register 沒有全部使用，最常使用的 variable 會留在 register

2、assembly operands are registers

5、Each instructions is 32 bits

[group14]

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




```
lw   $t0, 0($s0)   → 26
lw   $t1, 4($s0)   → 214
add  $t2, $t0, $t1 → 240
sw   $t2, 16($s0)
```

Ans: M[1016] = 240