Computer Architecture

Fall, 2024
Week 3
2024.09.16

組別：_____ **簽名：**_____

[group 1]

1. 問答題

   a) Why are MIPS instructions fixed at 32 bits in length? What is the advantage of this design?

   b) Why does MIPS have a special register, $zero, which is always set to 0? What is its use? What is its restriction?

Ans:

a) MIPS instructions are fixed at 32 bits to keep the architecture simple and regular. This consistency simplifies hardware design, allowing for faster execution and reducing cost. It follows the design principle that simplicity favors regularity.

b) The $zero register is used because the value zero is frequently needed in instructions. Having a dedicated register that always holds zero simplifies operations like moving data between registers or initializing values, improving the overall efficiency of the instruction set. The value of $zero is hardwired to 0 and cannot be changed. Any attempt to write to $zero will be ignored, meaning that instructions like add $zero, $t1, $t2 will have no effect. The value of $zero will always remain 0, regardless of any operations attempted on it.

[group 3]

2. 設計 ISA 的原則有哪三點？原因為何？下列 MIPS 的設計分別遵循哪個原則？

   a) MIPS 只使用 32 個暫存器
   b) 每個指令的大小皆為 32 bits
   c) 透過 I type 指令使用常數
   d) MIPS 只有三種類型的指令
   e) 採用 load/store 暫存器設計，算術 (arithmetic) 的運算元皆為暫存器而不是記憶體

Ans:

1. simplicity favors regularity: 實作容易，能夠以較低的成本取得好的性能
2. smaller is faster: 邏輯設計的特性
3. make the common case fast: 根據 Amdahl's law，受影響的比例越大，優化效果越顯著

a) MIPS 只使用 32 個暫存器：2
b) 每個指令的大小皆為 32 bits：1
c) 透過 I type 指令使用常數：3

d) MIPS 只有三種類型的指令：1

e) 採用 load/store 暫存器設計，算術 (arithmetic) 的運算元皆為暫存器而不是記憶體：1

[group 6]

3. 是非題（錯誤請說明原因）

a) Registers are faster and can reduce memory traffic.

b) Registers $s1and $t1 are used for long-term data storage.

c) Add $s0,$0,$t0 means that we are moving data between two registers.

d) Just like C programming, assembly uses variables.

e) Using registers requires more bits to specify operands than accessing memory locations.

Ans:

a) True.

b) False, $t1 is used for temporary data storage.

c) True.

d) False, assembly directly manipulates the CPU's register and memory address.

e) False, using registers requires less bits (5bits) to specify operands than accessing memory locations (32bits).

[group 10]

4. 是非題 (錯誤請說明原因)

a) Registers in MIPS are unlimited in number and can be dynamically created during program execution.

b) In MIPS, it is possible to load a word from memory and store it directly into another memory location without using any registers.

c) MIPS provides a special register $zero, which always holds the value 0 and cannot be changed.

d) In MIPS, memory addresses are typically calculated using only the value in a base register.

Ans:

a) False, MIPS has a fixed number of 32 registers, and they cannot be dynamically created.

b) False, MIPS requires that data be loaded into a register first using lw before it can be manipulated or stored back into memory using sw.

c) True, $zero is a dedicated MIPS register that always holds the value 0, and any attempt to write to it will be ignored

d) False, MIPS uses a base register plus an offset to calculate memory addresses

[group 11]

5. What is x if the maximum number of memory words you can use in a 32-bit MIPS machine in a single program is expressed as 2^x ? (Note: MIPS uses a byte addressing scheme.)

Ans:

x = 30. Since it is 4 bytes per word, the memory of MIPS machine has $(2^{32} / 4) = 2^{30}$ words.

[group 12]

6. In R-format instruction, what is the meaning of all-null value for the 'shamt'? What is the result of the following operation if the value in **$s1** is **10100** and **$s2** is **00100**?

000000 10001 10010 01000 00000 100010

Ans:

a) 00000 means non-shift operation so no shifting operation will be done

b) The opcode (first 000000) means the instruction is R-instruction, and the last 6 digits are 100010 means it's a subtraction function. if s1 is 10100 (20) and s2 is 00100 (4), the result will be 10000 (16)

[group 13]

7. Why is using registers to store elements better than using memories? List at least three reasons of it.

Ans:

1) Registers are faster than memories in hardware
2) Registers are easier for a compiler to use
3) Registers can hold variables to reduce memory traffic
4) Registers improve code density since register named with fewer bits than memory location

[group 7]

8. Write the following sequence of code into MIPS assembler: x=x-y+z-q,Assume that x, y, z, q are stored in registers $s1-$s4.

Ans:

```
sub $s1, $s1, $s2
add $s1, $s1, $s3
sub $s1, $s1, $s4
```

[group 2]

9. Translate the following MIPS code to C. Assume that the variables a,b,c are assigned to registers $s0, $s1, $s2, respectively. Furthermore, array A has a base address in register $s4.

```
lw $s2, 8($s4)
addi $s0, $s0, 10
```

add $s1, $s0, $s2
sw $s1, 12($s4)

Ans:

c = A[2];
a = a+10;
b = a+c;
A[3] = b;

[group 4]

10. The following problem deal with translating from C to MIPS.

f = g + h + A[4];

Assume that the variables f, g,and h are assigned to registers $s0, $s1, and $s2, respectively. Assume that the base address of the arrays A is in register $s3. You may use temporary register $t0, $t1. What is the corresponding MIPS assembly code?

Ans:

lw $t0, 16($s3)
add $t1, $s1, $s2
add $s0, $t1, $t0

[group 5]

11. 問答題

a) If we want to perform a subtract immediate instruction, what should we do?
b) How to use the add operator to move $s2 to $s1? Write the assembly code.
c) Explain under what conditions does spilling occur?
d) What is alignment, and what are its advantages?

Ans:

a) 要從暫存器中減去一個 constant value，可以使用 addi 指令搭配負數
b) add $s1, $s2, $zero
c) 當程序在執行過程中需要操作的變數超過了可用暫存器的數量時，會發生 spilling。暫存器數量有限，且暫存器是用來快速存取數據的。如果程序使用的變數超過了可用的暫存器，一些變數就必須被 spilling 到記憶體中。由於存取記憶體的速度比暫存器慢，因此溢出會對性能產生負面影響。
d) alignment 是指數據儲存在與其大小倍數對應的記憶體地址上（例如 4 字節的數據存放在可被 4 整除的地址）。可以確保 CPU 能夠更快速、有效地存取數據。未 alignment 的數據可能需要額外的記憶體操作，從而降低性能。