

Computer Organization 2016

HOMEWORK I

Due day: 2016 年 3 月 31 日 23:59

本作業主要目標是讓你熟悉 MIPS 指令集架構，包含:認識指令集架構中的 FORMAT、如何撰寫組語、如何操作 MIPS 模擬器。作業內容是要求你使用助教列出之指令實作 Fibonacci 數列運算。

繳交作業須知

- 本作業需要個人獨立完成，**嚴禁抄襲**，違者本學期作業分數不予計算，**包含抄襲者與被抄襲者皆會處罰**。
- 請壓縮所有**要求檔案為 ZIP 檔**，並且上傳至 Moodle 作業平台。**不符規定者將予以扣分**。
- **提醒!**請盡早至 Moodle 平台上傳檔案，一切以 **Moodle 繳交狀態為主**，**拒絕接受任何理由**。且**不接受任何額外補交方式**。

評分標準

- 完成**本作業要求**，並且完成 Project Report，。
- 完成本作業 Project Report，請使用課程網頁上提供 HW1 Project Report template 完成報告。
課程網頁: [計算機組織 2016 課程網頁](#)
- 繳交作業目錄請遵循下列配置，請參考 Fig.1

F740XXXXX(你的學號)(此為資料夾)

SRC(資料夾 內容存放你的程式碼)

F740XXXXX.docx(你的 Project Report)

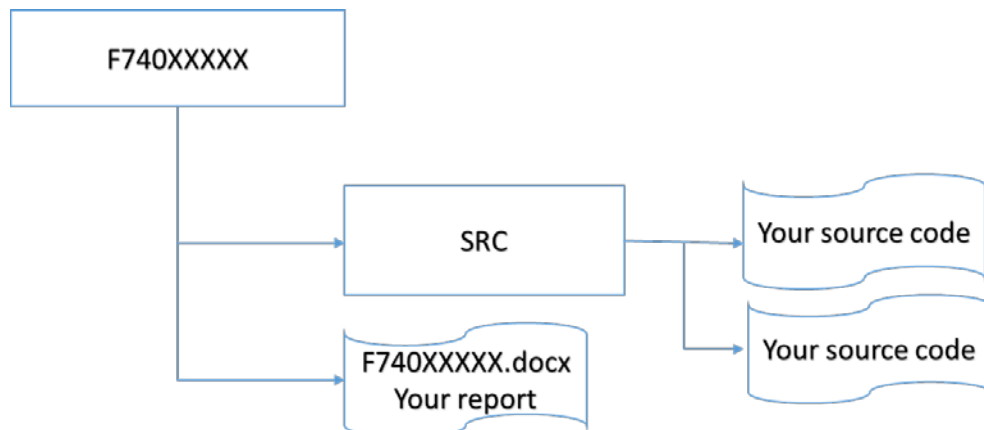


Fig.1 上傳檔案的檔案配置圖

作業內容

本作業挑選數個指令集提供你使用，請勿使用本作業規定以外的 MIPS 指令，(因為未來你將會使用你實作的 CPU 驗證本作業，如果使用其他未在規範內的指令，你將會在後面的 HW 無法正確的驗證你的程式與 CPU)。

使用下列提供的 MIPS 指令集撰寫 Fibonacci 運算。以下亦會介紹 Fibonacci 數列運算。

MIPS 指令集

R Type

Assembler Syntax

instruction	rd	rs	rt
-------------	----	----	----

Machine code Format

opcode		rs	rt	rd	shamt	funct					
31	26	25	21	20	16	15	11	10	6	5	0
opcode	Mnemonics	SRC1	SRC2	DST	funct	Description					
000000	nop	00000	00000	00000	000000	No operation					
000000	add	\$Rs	\$Rt	\$Rd	100000	Rd = Rs + Rt					
000000	sub	\$Rs	\$Rt	\$Rd	100010	Rd = Rs – Rt					
000000	and	\$Rs	\$Rt	\$Rd	100100	Rd = Rs & Rt					
000000	or	\$Rs	\$Rt	\$Rd	100101	Rd = Rs Rt					
000000	xor	\$Rs	\$Rt	\$Rd	100110	Rd = Rs ^ Rt					
000000	nor	\$Rs	\$Rt	\$Rd	100111	Rd = ~(Rs Rt)					
000000	slt	\$Rs	\$Rt	\$Rd	101010	Rd = (Rs < Rt)?1:0					
000000	sll		\$Rt	\$Rd	000000	Rd = Rt << shamt					
000000	srl		\$Rt	\$Rd	000010	Rd = Rt >> shamt					
000000	jr	\$Rs			001000	PC=Rs					

I Type

Assembler Syntax

instruction	rt	rs	imm
-------------	----	----	-----

Machine code Format

opcode								rs		rt		immediate							
31	26	25	21	20	16	15	0												

opcode	Mnemonics	SRC1	DST	SRC2	Description
001000	addi	\$Rs	\$Rt	imm	Rt = Rs + imm
001100	andi	\$Rs	\$Rt	imm	Rt = Rs & imm
001010	slti	\$Rs	\$Rt	imm	Rt = (Rs < imm) ? 1 : 0
000100	beq	\$Rs	\$Rt	imm	If(Rs == Rt) PC=PC+4+imm
000101	bne	\$Rs	\$Rt	imm	If(Rs != Rt) PC=PC+4+imm
100011	lw	\$Rs	\$Rt	imm	Rt = Mem[Rs + imm]
101011	sw	\$Rs	\$Rt	imm	Mem[Rs + imm] = Rt

J Type

Assembler Syntax

instruction	Target(label)
-------------	---------------

Machine code Format

opcode	address
31	26 25 0

opcode	Mnemonics	Address	Description
000010	j	jumpAddr	PC = jumpAddr
000011	jal	jumpAddr	R[31] = PC + 8 ; PC = jumpAddr

Fibonacci 數列

費波那契數列（義大利語：Successione di Fibonacci），又譯費波拿契數、斐波那契數列、費氏數列、黃金分割數列。

在數學上，費波那契數列是以遞迴的方法來定義：

$$F_0=0$$

$$F_1=1$$

$$F_n = F_{n-1} + F_{n-2} \quad (n \geq 2)$$

用文字來說，就是費波那契數列由 0 和 1 開始，之後的費波那契系數就由之前的兩數相加。首幾個費波那契系數是以下：

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233.....

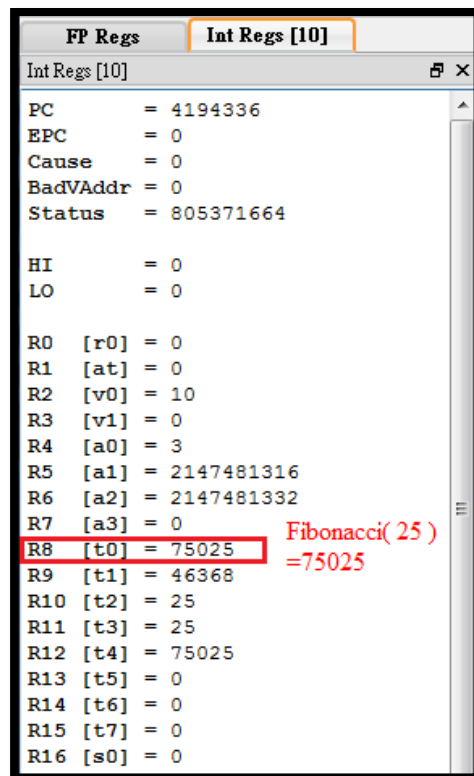
Fibonacci Pseudo code

```
function fibonacci(n)
    integer a = 1
    integer b = 0
    integer t

    for i from 1 to n
        t = a + b
        b = a
        a = t
    return a
```

作業要求

1. 請使用本作業提供的 MIPS 指令完成 function fibonacci(n)。
2. 請使用 MIPS 模擬器運行組合語言運算出 **fibonacci(25)** 之結果並存在 **register \$t0**。
3. 完成 Project Report。
4. 將完成結果截圖並附在 Report 當中。範例如 Fig2:



The screenshot shows the MIPS simulator's register window. The 'Int Regs [10]' tab is selected. The register list includes PC, EPC, Cause, BadVAddr, Status, HI, LO, and R0 through R16. Register R8, which is \$t0, is highlighted with a red box and contains the value 75025. To the right of the register list, the text 'Fibonacci(25) = 75025' is displayed in red.

Register	Value
PC	4194336
EPC	0
Cause	0
BadVAddr	0
Status	805371664
HI	0
LO	0
R0 [r0]	0
R1 [at]	0
R2 [v0]	10
R3 [v1]	0
R4 [a0]	3
R5 [a1]	2147481316
R6 [a2]	2147481332
R7 [a3]	0
R8 [t0]	75025
R9 [t1]	46368
R10 [t2]	25
R11 [t3]	25
R12 [t4]	75025
R13 [t5]	0
R14 [t6]	0
R15 [t7]	0
R16 [s0]	0

Fig2. 完成答案結果

提醒

上傳作業時，請確定上傳內容有符合本作業要求，並且符合評分標準之規定，包含:目錄結構、須繳交檔案、以及報告格式，如不符規定者將扣分。

如有其他問題請聯繫助教。