

1. 考慮兩種實現相同指令集架構(ISA) P1, P2 的設計，其中共有四種指令，分別為 A, B, C, D，下圖為其時脈速度與每個指令所需之時脈(CPI)，

	Clock rate	CPI class A	CPI class B	CPI class C	CPI class D
P1	2GHz	1	2	2	5
P2	4GHz	2	2	3	4

若有一程式具有 30% 的 A 指令，40% 的 B 指令，20% 的 C 指令，10% 的 D 指令，

- (a) 兩種設計之平均 CPI 各為多少？
 (b) 兩種設計的 MIPS(每秒百萬指令) 各為多少？
 (c) 請問哪種設計執行程式較快？
 (d) 若在 Compiler 作指令最佳化改進，P1 及 P2 之指令 A 減少 20%，指令 B 減少 15%，指令 C 減少 10%，指令 D 減少 5%，請問 P1, P2 執行效能各提高多少？哪種設計執行程式較快？
2. 以下為原始機器(Mbase 時脈為 2500MHz)及改進機器(Mopt 時脈為 4000MHz)的測試數據。

Instruction class	CPI	Frequency
A	2	40%
B	3	25%
C	3	25%
D	5	10%

Instruction class	CPI	Frequency
A	2	40%
B	2	25%
C	3	25%
D	4	10%

- (a) 試問兩機器的 CPI 為何？
 (b) Mbase 與 Mopt 原始的 MIPS 值為多少？
 (c) Mopt 比 Mbase 在效能上快多少？
 (d) 若在 Compiler 作指令最佳化改進(Mcomp)可使 Mbase 之指令 A 減少 40%，指令 B 減少 30%，指令 C 減少 20%，指令 D 減少 10%，試求 Mcomp 的 CPI 值
 (e) 若將 Mopt 的技術結合 Mcomp 的技術(Mboth)所造出來的機器比 Mbase 快多少？
 (若沒有計算機，需列出計算步驟及算式)
3. 請繪圖說明乘法硬體第一版與最佳版本的動作原理，並以 4 位元數值計算 $3*5$ (0011*0101)
 [包含初始值、步驟、乘數、被乘數、及乘積的所有過程]

4. 利用所附之 MIPS 組合語言指令表及機器碼(圖一)，將下列 MIPS 組合語言程式轉換成機器碼

```

LOOP: slt  $t3, $0, $t1
      beq  $t1, $0, DONE
      sub  $t1, $t2, $t2
      add  $s2, $s4, $t1
      lw   $t1, 0($s2)
      j    LOOP
  
```

(假設 LOOP 的起始位置=60000H)

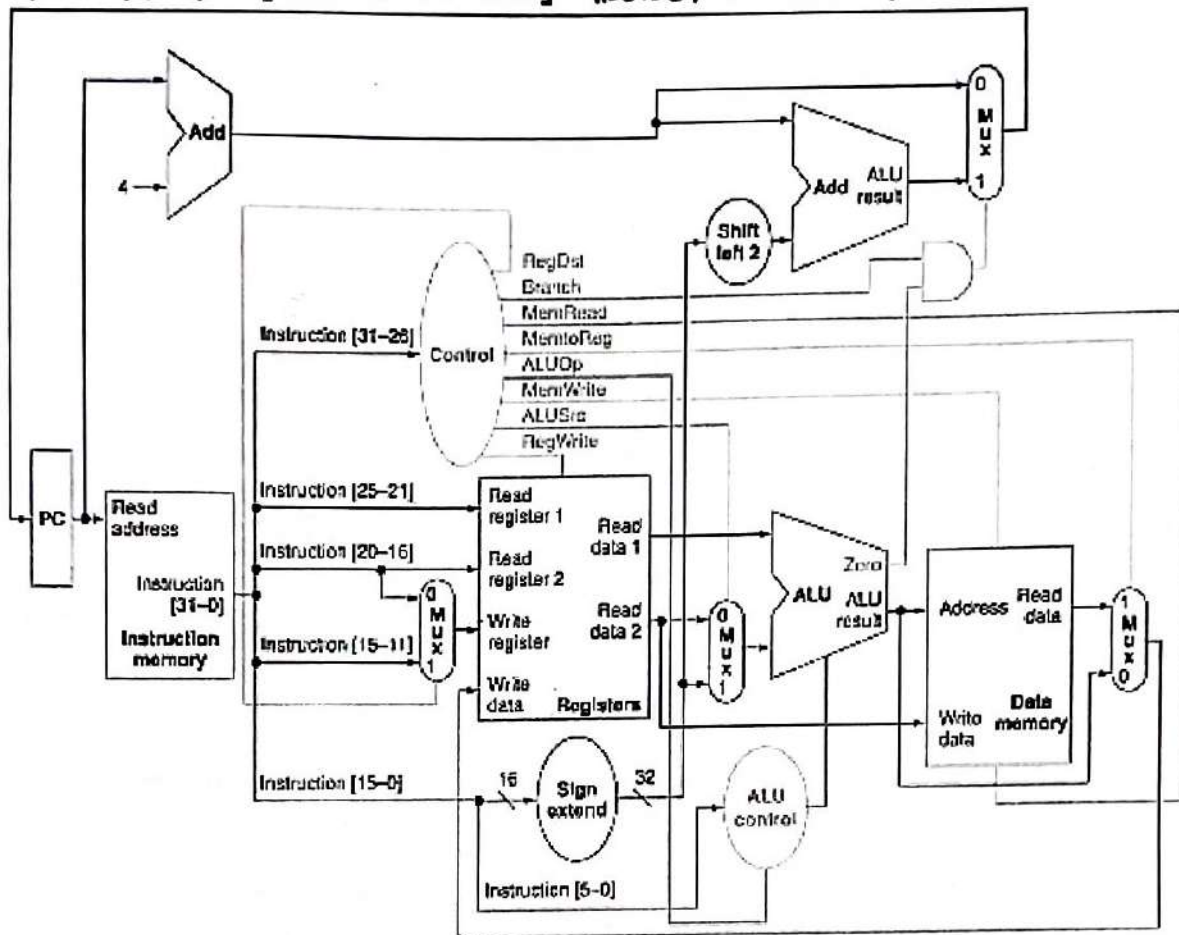
(hint: \$t0=\$8, \$s0=\$16)

DONE:

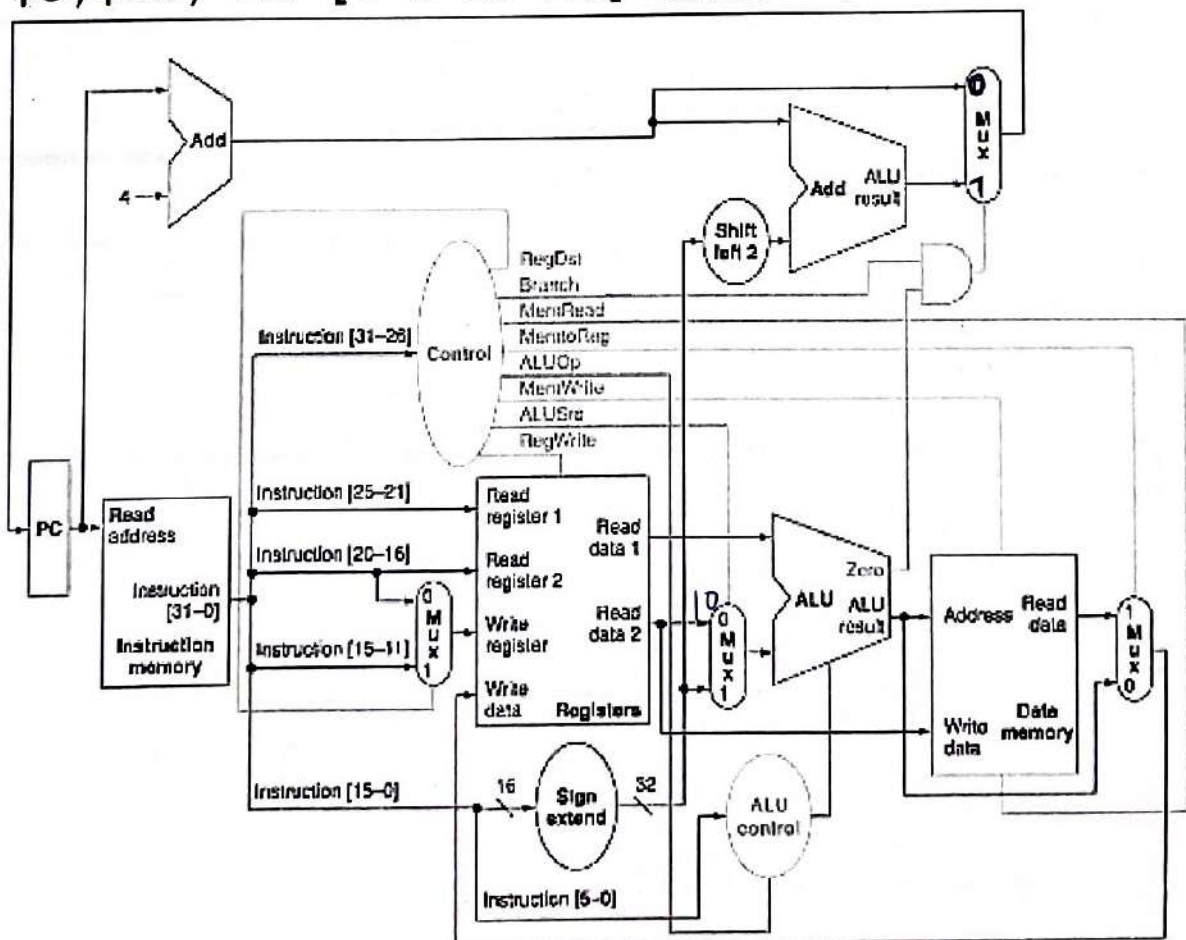
2. 請說明 ALU 的設計(圖二)中

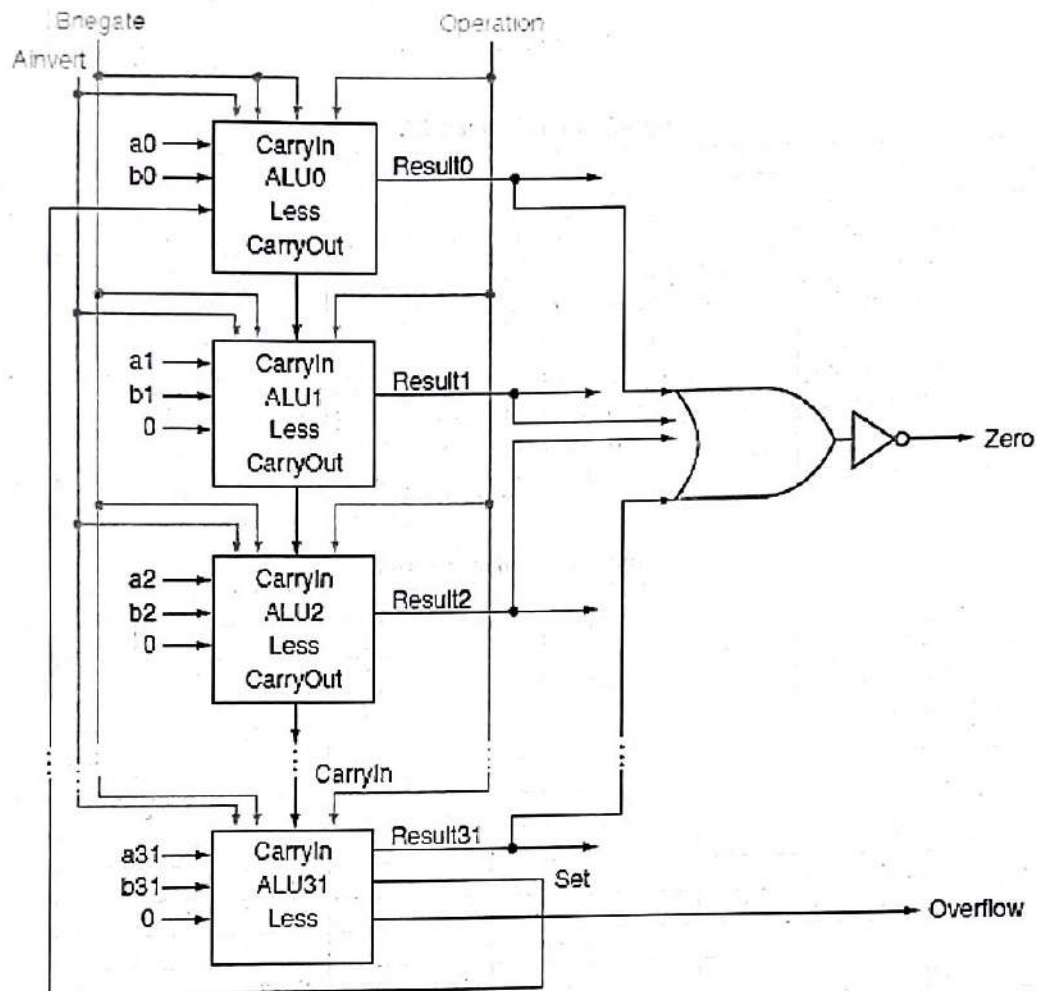
- (a) slt 指令是如何被設計的, ALU operation 等於多少？
 (b) ALU operation = 0110 是作甚麼的動作？為什麼？
 (c) 這個電路可以執行 NAND 指令嗎？如果可以，ALU operation 等於多少？
 (d) 請說明 SLT 指令是如何產生結果的

sw \$1,20(\$5) [35 5 1 20] 假設\$1=200 \$5=500 PC=1000



beq \$5,\$10, 20 [4 5 10 20] 假設\$5=\$10=500 PC=1000





MIPS assembly language

Category	Instruction	Example	Meaning	Comments
Arithmetic	add	add \$1,\$2,\$3	$\$1 = \$2 + \$3$	3 operands: data in registers
	subtract	sub \$1,\$2,\$3	$\$1 = \$2 - \$3$	3 operands: data in registers
Data Transfer	load word	lw \$1,100(\$2)	$\$1 = \text{Memory}[\$2+100]$	Data from memory to register
	store word	sw \$1,100(\$2)	$\text{Memory}[\$2+100] = \1	Data from register to memory
Conditional Branch	branch on equal	beq \$1,\$2,L	if $(\$1 == \$2)$ go to L	Equal test and branch
	branch on not eq.	bne \$1,\$2,L	if $(\$1 != \$2)$ go to L	Not equal test and branch
	set on less than	slt \$1,\$2,\$3	if $(\$2 < \$3)$ $\$1=1$; else $\$1=0$	Compare less than; for beq,bne
Unconditional Jump	jump	j 10000	go to 10000	Jump to target address
	jump register	jr \$31	go to \$31	For switch, procedure return
	jump and link	jal 10000	$\$31 = \text{PC} + 4$; go to 10000	For procedure call

MIPS machine language

Name	Format	Example						Comments
add	R	0	2	3	1	0	32	add \$1,\$2,\$3
sub	R	0	2	3	1	0	34	sub \$1,\$2,\$3
lw	I	35	2	1	100			lw \$1,100(\$2)
sw	I	43	2	1	100			sw \$1,100(\$2)
beq	I	4	1	2	100			beq \$1,\$2,100
bne	I	5	1	2	100			bne \$1,\$2,100
slt	R	0	2	3	1	0	42	slt \$1,\$2,\$3
j	J	2	10000					j 10000 (see section 3.7)
jr	R	0	31	0	0	0	8	jr \$1
jal	J	3	10000					jal 10000 (see section 3.7)
Field size		6 bits	5 bits	5 bits	5 bits	5 bits	6 bits	All MIPS instructions 32 bits
Format R	R	op	rs	rt	rd	shamt	funct	Arithmetic instruction format
Format I	I	op	rs	rt	address			Data transfer, branch format