

以下題目均按學號規則推算出指定的題數來做，學號為奇數者做奇數題，偶數者做偶數題，做錯題目則該題不計分，共 99 分，1 分奉送！

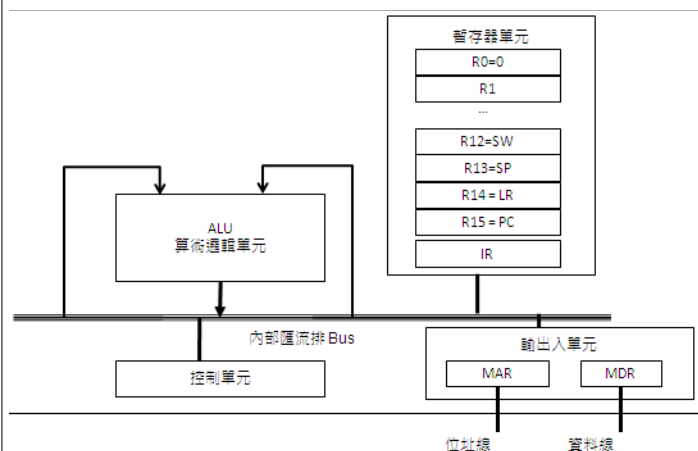
名詞解釋：每題 3 分，做指定的 6 題，共 18 分。

1. 組譯器 (Assembler)：用來將組合語言轉換成機器碼（目的檔）的工具。
2. 連結器 (Linker)：用來將很多目的檔連結成一個執行檔或函式庫的工具。
3. 動態連結 (Dynamic Linking) 在執行時才進行連結動作的一種運作機制，這讓沒執行到的函式庫可以不需要被載入，而且很多個程式可以共用一個函式庫，在記憶體內只需要存在一份，不需重複。
4. Objdump：目的檔傾印工具，用來印出目的檔的表頭與內容資訊，有時還可以進行反組譯。
5. 假指令 (Pseudo Instruction)：組合語言中，不是真正對應機器指令的命令，稱為假指令，通常是用來提供資訊給組譯器，作為組譯指引用的。
6. 機器碼 (Machine Code)：電腦處理器所能直接執行的二進位碼，通常和組合語言指令有直接對應關係。
7. make：make 是 GNU 的專案建置工具，可用來解譯 Makefile 檔案並進行大型專案的自動建置工作。
8. 暫存器 (register)：CPU 當中用來儲存可直接進行運算資料的地方，有些暫存器會有特殊用途，例如程式計數器或連結暫存器等。
9. gcc：開源組織 GNU 所推出的 c 語言編譯器，包含一整套編譯連結工具。
10. 程式計數器 (PC, Program Counter)：用來儲存擷取指令位址的暫存器，通常也是用來作相對定址的原點。
11. 狀態暫存器 (Status Word)：用來儲存處理器狀態的暫存器，儲存內容包含比較結果旗標，中斷控制位元，模式位元等等。
12. 連結暫存器 (LR, Link Register) 在副程式呼叫時，可以用來儲存返回位址的暫存器，再呼叫返回指令 RET 時可以回到副程式後，讓程式繼續正常執行的暫存器。

簡答題：每題 5 分，做指定的 4 題，共 20 分。

1. 請畫出一顆 CPU 的架構示意圖，並說明其中各個元件的功用？

以 CPU0 為例，R0 到 R15 為暫存器單元，另有輸出入暫存器與 ALU，以及控制線路所形成的控制單元等。



2. 請說明 CPU0 有哪些特殊用途暫存器，各是做什麼用途的？

R0：永遠都是 0

R12：SW 狀態暫存器，儲存比較結果等旗標。

R13：SP 堆疊暫存器，儲存堆疊指標，用在 PUSH, POP 等堆疊指令上。

R14：LR 連結暫存器，儲存呼叫返回位址，CALL 指令執行時會將下一個指令的位址放入 LR，等到 RET 指令時會取回 LR 放回 PC 中，如此才能回到上層程式繼續執行。

R15：PC 程式計數器，用來儲存指令位址，並在擷取指令完成後進到下一個位址。

3. 請說明組譯器第一階段的功能？

第一階段：計算指令位址，並且記住所有符號的位址，建立符號表

4. 請說明組譯器第二階段的功能？

第二階段：將組合語言指令轉換成機器碼，並進行資料編碼的動作。

5. 請寫出 LDI R1, 100 的機器碼 (16 進位寫法), 並說明其編碼的原理?

LDI R1, 100

08 1 00064

LDI 之 OP 為 08,  
R1 編為 1  
100 轉為 16 進位即為 64,

於是得到 08 1 00064

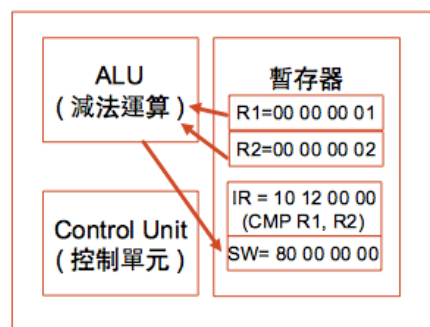
6. 請寫出 SHR R5, R6, 4 的機器碼 (16 進位寫法), 並說明其編碼的原理?

SHR R5, R6, 4

1F 5 6 0004

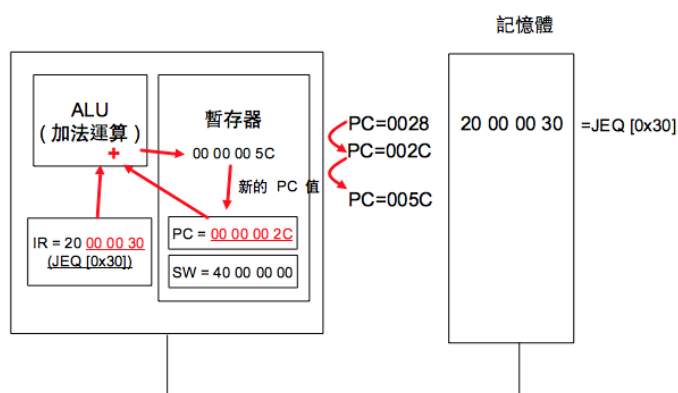
SHR 之 OP 為 1F  
R5 編為 5  
R6 編為 6  
4 轉為 16 進位還是 4

7. 請舉例說明 CPU0 中比較指令 CMP 的運作原理。



上例為 CMP R1, R2 的結果

8. 請舉例說明 CPU0 中條件跳躍指令 JEQ 的運作原理。



上例為 JEQ 0030 的結果。

考生姓名:

學號:

程式題 (請以 CPU0 的組合語言撰寫): 每題 10 分, 做指定的 4 題, 共 40 分。

1. 請寫出一個組合語言程式, 可以將變數 X, Y 的內容交換。

```
LD R1, X
LD R2, Y
ST R1, Y
ST R2, X
```

2. 請寫出一個組合語言程式, 可以將變數 X 的內容設定為 Y+10。

```
LDI R1, 10
LD R2, Y
ADD R3, R2, R1
ST R3, X
```

5. 請寫出一個「組合語言副程式」, 可以設定 R1 為  $R1 * R1 * R1$ 。
6. 請寫出一個組合語言程式, 說明如何透過暫存器傳遞參數。

下列程式透過暫存器 R1 將參數傳給 double, 並且也透過 R1 取得傳回的兩倍值。

```
LD R1, X
CALL double
ST R1, Y

double:
ADD R1, R1, R1
RET

X: WORD 3
Y: WORD 0
```

3. 請寫出一個組合語言程式, 可以取得變數 X, Y 中較小的值放入變數 max 中。

```
LD R1, X
LD R2, Y
CMP R1, R2
JGT ELSE
ST R2, min
JMP EXIT
ELSE: ST R1, min
EXIT: RET

X: WORD 3
Y: WORD 5
min: WORD 0
```

4. 請寫出一個組合語言程式, 可以取得變數 X, Y 中較大的值放入變數 min 中。

```
LD R1, X
LD R2, Y
CMP R1, R2
JLT ELSE
ST R2, min
JMP EXIT
ELSE: ST R1, min
EXIT: RET

X: WORD 3
Y: WORD 5
min: WORD 0
```

7. 請寫出一個組合語言程式, 說明如何透過堆疊傳遞參數。
8. 請寫出一個組合語言程式, 說明如何透過堆疊傳遞參數。

下列程式先設定好堆疊後, 可以透過堆疊傳入 x, 然後在 f 函數裏取出候在加上五, 最後放在 R1 傳回。

```
LD R13, StackEndPtr
LD R2, x
PUSH R2
CALL f
ST R1, y
RET

x: WORD 1
y: RESW 1
Stack: RESW 100
StackEnd:
StackEndPtr: StackEnd
f:

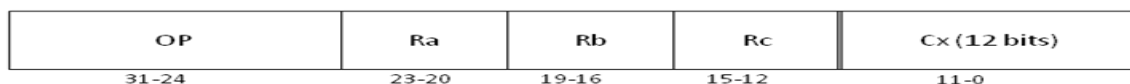
POP R2
LDI R4, 5
ADD R1, R2, R4
```

	<b>ST R1, r</b> <b>RET</b> <b>r: RESW 1</b>
--	---

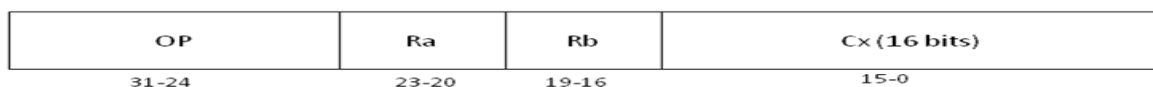
請寫出下列程式的位址欄與機器碼 (每列 3 分, 位址欄 1 分, 機器碼 2 分, 做指定的 7 列, 共 21 分)

題號	位址欄	組合語言	機器碼
1.	0000	LDI R9, 1	08900001
2.	0004	LDI R1, 0	08100000
3.	0008	LD R2, ptr	002F002E
4.	000C	WHILE: LDI R8, 99	08800063
5.	0010	LDB R3, [R2]	02320000
6.	0014	CMP R3, R8	10380000
7.	0018	JEQ EXIT	2000000C
8.	001C	ADD R1, R1, R3	13113000
9.	0020	ADD R2, R2, R9	13229000
10.	0024	JMP WHILE	26FFFFE4
11.	0028	EXIT: RET	2C000000
12.	002C	id: BYTE 1,0,9,9,1,0,6,7,9, 99	01000909010006070963
13.	0036	sum: WORD 0	00000000
14.	003A	ptr: WORD id	0000002C

A 型



L 型



J 型



類型	格式	指令	OP	說明	語法	語意
載入儲存	L	LD	00	載入 word	LD Ra, [Rb+Cx]	$Ra \leftarrow [Rb + Cx]$
	L	ST	01	儲存 word	ST Ra, [Rb+ Cx]	$Ra \rightarrow [Rb + Cx]$
	L	LDB	02	載入 byte	LDB Ra, [Rb+ Cx]	$Ra \leftarrow (\text{byte})[Rb + Cx]$
	L	STB	03	儲存 byte	STB Ra, [Rb+ Cx]	$Ra \rightarrow (\text{byte})[Rb + Cx]$
	A	LDR	04	LD 的暫存器版	LDR Ra, [Rb+Rc]	$Ra \rightarrow (\text{byte})[Rb + Rc]$
	A	STR	05	ST 的暫存器版	STR Ra, [Rb+Rc]	$Ra \rightarrow [Rb + Rc]$
	A	LBR	06	LDB 的暫存器版	LBR Ra, [Rb+Rc]	$Ra \leftarrow (\text{byte})[Rb + Rc]$
	A	SBR	07	STB 的暫存器版	SBR Ra, [Rb+Rc]	$Ra \rightarrow (\text{byte})[Rb + Rc]$
	L	LDI	08	立即載入	LDI Ra, Rb+Cx	$Ra \leftarrow Rb + Cx$
運算指令	A	CMP	10	比較	CMP Ra, Rb	$SW \leftarrow Ra \geq Rc$
	A	MOV	12	移動	MOV Ra, Rb	$Ra \leftarrow Rb$
	A	ADD	13	加法	ADD Ra, Rb, Rc	$Ra \leftarrow Rb + Rc$
	A	SUB	14	減法	SUB Ra, Rb, Rc	$Ra \leftarrow Rb - Rc$
	A	MUL	15	乘法	MUL Ra, Rb, Rc	$Ra \leftarrow Rb * Rc$
	A	DIV	16	除法	DIV Ra, Rb, Rc	$Ra \leftarrow Rb / Rc$
	A	AND	18	邏輯 AND	AND Ra, Rb, Rc	$Ra \leftarrow Rb \text{ and } Rc$

	A	OR	19	邏輯 OR	OR Ra, Rb, Rc	$Ra \leftarrow Rb \text{ or } Rc$
	A	XOR	1A	邏輯 XOR	XOR Ra, Rb, Rc	$Ra \leftarrow Rb \text{ xor } Rc$
	A	ROL	1C	向左旋轉	ROL Ra, Rb, Cx	$Ra \leftarrow Rb \text{ rol } Cx$
	A	ROR	1D	向右旋轉	ROR Ra, Rb, Cx	$Ra \leftarrow Rb \text{ ror } Cx$
	A	SHL	1E	向左移位	SHL Ra, Rb, Cx	$Ra \leftarrow Rb \ll Cx$
	A	SHR	1F	向右移位	SHR Ra, Rb, Cx	$Ra \leftarrow Rb \gg Cx$
跳躍指令	J	JEQ	20	跳躍 (相等)	JEQ Cx	if SW(=) $PC \leftarrow PC + Cx$
	J	JNE	21	跳躍 (不相等)	JNE Cx	if SW(!=) $PC \leftarrow PC + Cx$
	J	JLT	22	跳躍 (<)	JLT Cx	if SW(<) $PC \leftarrow PC + Cx$
	J	JGT	23	跳躍 (>)	JGT Cx	If SW(>) $PC \leftarrow PC + Cx$
	J	JLE	24	跳躍 (<=)	JLE Cx	if SW(<=) $PC \leftarrow PC + Cx$
	J	JGE	25	跳躍 (>=)	JGE Cx	If SW(>=) $PC \leftarrow PC + Cx$
	J	JMP	26	跳躍 (無條件)	JMP Cx	$PC \leftarrow PC + Cx$
	J	SWI	2A	軟體中斷	SWI Cx	$LR \leftarrow PC; PC \leftarrow Cx$
	J	CALL	2B	跳到副程式	CALL Cx	$LR \leftarrow PC; PC \leftarrow PC + Cx$
	J	RET	2C	返回	RET	$PC \leftarrow LR$
堆疊指令	A	PUSH	30	推入 word	PUSH Ra	$SP -= 4; [SP] = Ra;$
	A	POP	31	彈出 word	POP Ra	$Ra = [SP]; SP += 4;$
	A	PUSHB	32	推入 byte	PUSHB Ra	$SP--; [SP] = Ra; \text{(byte)}$
	A	POPB	33	彈出 byte	POPB Ra	$Ra = [SP]; SP++; \text{(byte)}$