金門大學 資訊工程系　102 學年度下學期　系統程式 期中考　出題者：陳鍾誠

考生姓名：　學號：　得分：

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

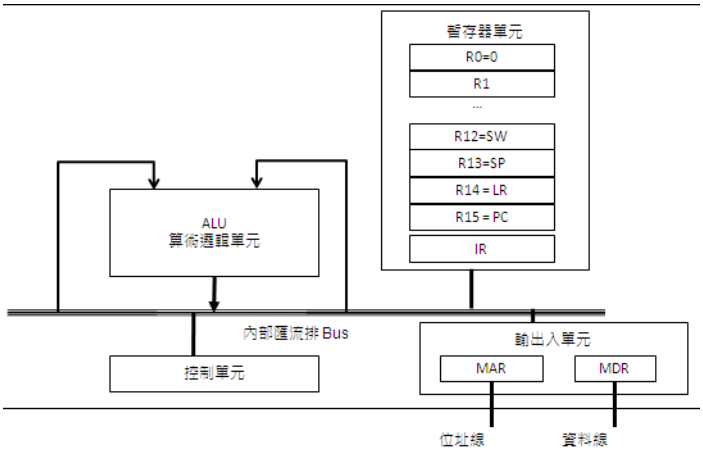
名詞解釋：每題 3 分，做指定的 6 題，共 18 分。

1. 目的檔 (object file)：編譯或組譯後輸出的檔案，通常包含機器碼與修改描述資訊，像是函式庫、執行檔等都是以目的檔形式存在的。
2. objdump：GNU 的目的檔傾印工具，可以用來列出目的檔資訊並進行反組譯動作。
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. 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。
5. 組合語言 (Assembly Language)：　一種直接對應到機器碼的文字型語言，寫完後可透過組譯器轉爲目的檔、執行檔或機器碼。
6. 假指令 (Pseudo Instruction)：組合語言中，不是真正對應機器指令的命令，稱爲假指令，通常是用來提供資訊給組譯器，作爲組譯指引用的。
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. 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。
9. 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。
10. 算術邏輯單元 (ALU)：CPU 當中用來進行運算的核心單元，通常可以用來進行加減乘除移位或邏輯等等運算。
11. 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。
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 爲例，R0 到 R15 爲暫存器單元，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. 請畫出電腦的馮紐曼架構，並說明其中各個元件的功用？

記憶體 Memory：儲存程式和資料，以供 CPU 讀取或寫入。
CPU：電腦指令運作的核心，可以解讀指令並執行對應

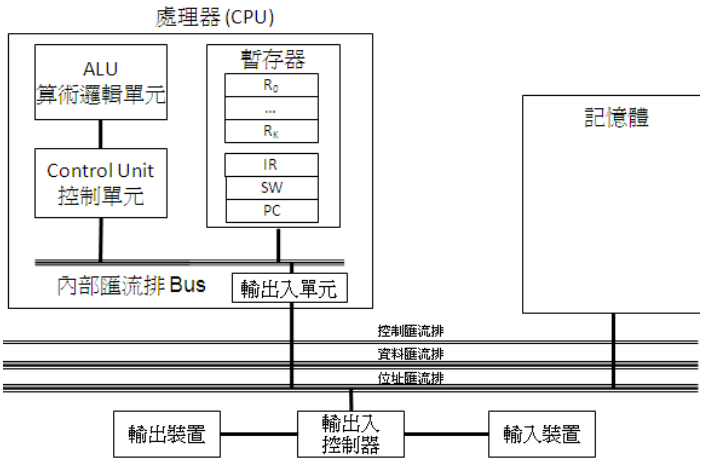
3. 請說明組譯器第一階段的功能？

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。

的動作。
輸出入：包含輸出裝置 (螢幕、硬碟、...)、輸入裝置(鍵盤、滑鼠、硬碟...) 與輸出入控制器 (介面卡, ...)，讓 CPU 可以達成輸出入動作。
匯流排 BUS：連接 CPU, Memory, IO Device 的線路，可細分爲「位址、資料與控制」等三種線路。



5. 請寫出 XOR R1, R2, R3 的機器碼 (16 進位寫法)，並說明其編碼的原理？

查表可知 XOR 爲 A 格式，且 OP 代碼爲 1A，因此編碼如下：

```
OP   Ra,  Rb,  Rc,     Cx
XOR  R1,  R2,  R3
1A    1    2    3      000
```

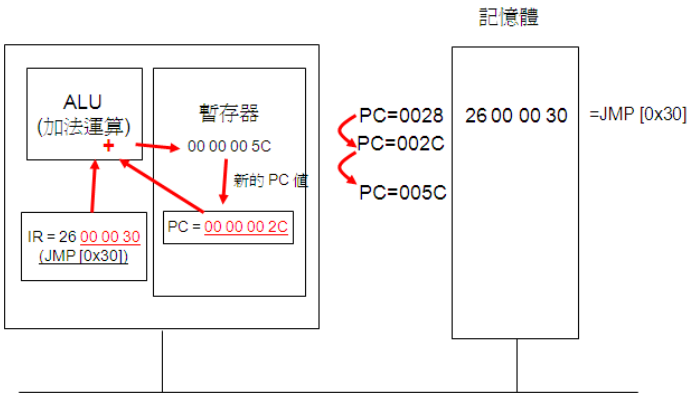
6. 請寫出 SHL R1, R2, 4 的機器碼 (16 進位寫法)，並說明其編碼的原理？

查表可知 XOR 爲 A 格式，且 OP 代碼爲 1E，因此編碼如下：

```
OP   Ra,  Rb,  Rc,     Cx
SHL  R1,  R2,          4
1A    1    2    0      004
```

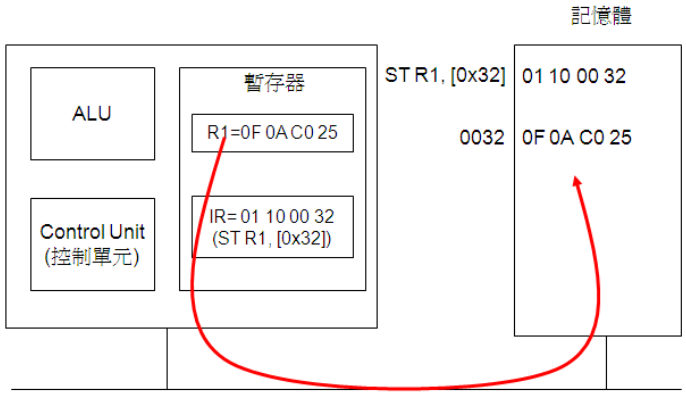
7. 請舉例說明 CPU0 中跳耀指令 JMP 的運作原理。

圖 2.11 跳躍指令JMP [0x30] 的執行過程



8. 請舉例說明 CPU0 中儲存指令 ST 的運作原理。

圖 2.14 儲存指令ST R1, [0x32] 的執行過程



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 的相加後放到變數 Z 中。

```
    LD R1, X
    LD R2, Y
    ADD R3, R1, R2
    ST R3, Z
    RET
X:  WORD 3
Y:  WORD 5
Z:  WORD 0
```

2. 請寫出一個組合語言程式，可以將變數 X 的內容加 10 後放回 X 中。

```
    LD R1, X
    LDI R2, 10
    ADD R3, R1, R2
    ST R3, X
    RET
X:  WORD 3
```

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 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
```

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 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
```

5. 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
```

6. 請寫出一個「組合語言副程式」，可以設定 R1 爲 3^{R2} 。

```
LD  R13, SPTR
```

7. 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
```

```
        PUSH R14
        LDI R2, 3
        CALL Power3
        POP R14
        RET
STACK:  WORD 0
SPTR:   WORD STACK

Power3: LDI R4, 1       ; R4 初始□□ 1
        LDI R1, 1       ; R1 初始□□ 1
        LDI R3, 3       ; R3 是常數 3
        LDI R5, 1       ; R5 是常數 1
LOOP:   CMP R4, R2      ; (R4 < R2) ?
        MUL R1, R1, R3  ; R1=R1*R3
        ADD R4, R4, R5  ; R4=R4+1
        JLT LOOP        ; goto LOOP
EXIT:   RET
```

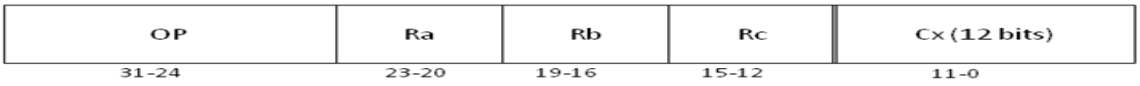
```
        ADD R1, R2, R4
        ST R1, r
        RET
r:      RESW 1
```

8. 請寫出一個組合語言程式，說明如何透過堆疊傳遞參數。

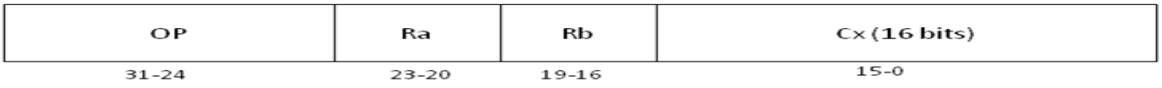
請寫出下列程式的位址欄與機器碼 (每列 3 分，位址欄 1 分，機器碼 2 分，做指定的 7 列，共 21 分)

題號	位址欄	組合語言	機器碼
1.	0000	LDI R1, 1	08100001
2.	0004	LD R2, aptr	002F002F
3.	0008	LDI R4, 1	08400001
4.	000C	FOR: LDB R3, [R2]	02320000
5.	0010	CMP R3, R0	10300000
6.	0014	JEQ EXIT	2000000C
7.	0018	MUL R4, R4, R3	15443000
8.	001C	ADD R2, R2, R1	13221000
9.	0020	JMP FOR	26FFFFE8
10.	0024	EXIT: ST R4, f	014F000B
11.	0028	RET	2C000000
12.	002C	a: BYTE 3, 2, 7, 6, 8, 5, 0	03020706080500
13.	0033	f: WORD 0	00000000
14.	0037	aptr: WORD a	0000002C

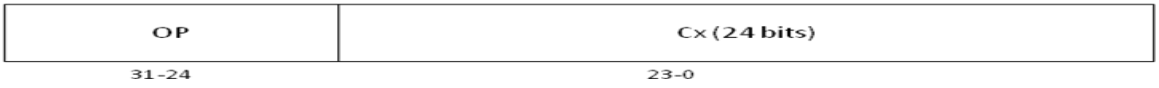
A 型



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←[Rb+ Cx]
	L	ST	01	儲存 word	ST Ra, [Rb+ Cx]	Ra→[Rb+ Cx]
	L	LDB	02	載入 byte	LDB Ra, [Rb+ Cx]	Ra←(byte)[Rb+ Cx]
	L	STB	03	儲存 byte	STB Ra, [Rb+ Cx]	Ra→(byte)[Rb+ Cx]
	A	LDR	04	LD 的暫存器版	LDR Ra, [Rb+Rc]	Ra→(byte)[Rb+ Rc]
	A	STR	05	ST 的暫存器版	STR Ra, [Rb+Rc]	Ra→[Rb+ Rc]
	A	LBR	06	LDB 的暫存器版	LBR Ra, [Rb+Rc]	Ra←(byte)[Rb+ Rc]
	A	SBR	07	STB 的暫存器版	SBR Ra, [Rb+Rc]	Ra→(byte)[Rb+ Rc]
	L	LDI	08	立即載入	LDI Ra, Rb+Cx	Ra← Rb + Cx
運算指令	A	CMP	10	比較	CMP Ra, Rb	SW ← Ra >=< Rb
	A	MOV	12	移動	MOV Ra, Rb	Ra ← Rb
	A	ADD	13	加法	ADD Ra, Rb, Rc	Ra ← Rb+Rc
	A	SUB	14	減法	SUB Ra, Rb, Rc	Ra ← Rb-Rc
	A	MUL	15	乘法	MUL Ra, Rb, Rc	Ra ← Rb*Rc
	A	DIV	16	除法	DIV Ra, Rb, Rc	Ra ← Rb/Rc
	A	AND	18	邏輯 AND	AND Ra, Rb, Rc	Ra ← Rb and Rc
	A	OR	19	邏輯 OR	OR Ra, Rb, Rc	Ra ← Rb or Rc
	A	XOR	1A	邏輯 XOR	XOR Ra, Rb, Rc	Ra ← Rb xor Rc
	A	ROL	1C	向左旋轉	ROL Ra, Rb, Cx	Ra ← Rb rol Cx
	A	ROR	1D	向右旋轉	ROR Ra, Rb, Cx	Ra ← Rb ror Cx
	A	SHL	1E	向左移位	SHL Ra, Rb, Cx	Ra ← Rb << Cx
	A	SHR	1F	向右移位	SHR Ra, Rb, Cx	Ra ← Rb >> Cx
跳躍指令	J	JEQ	20	跳躍 (相等)	JEQ Cx	if SW(=) PC ← PC+Cx
	J	JNE	21	跳躍 (不相等)	JNE Cx	if SW(!=) PC ← PC+Cx
	J	JLT	22	跳躍 (<)	JLT Cx	if SW(<) PC ← PC+Cx
	J	JGT	23	跳躍 (>)	JGT Cx	If SW(>) PC ← PC+Cx
	J	JLE	24	跳躍 (<=)	JLE Cx	if SW(<=) PC ← PC+Cx
	J	JGE	25	跳躍 (>=)	JGE Cx	If SW(>=) PC ← PC+Cx
	J	JMP	26	跳躍 (無條件)	JMP Cx	PC ← PC+Cx
	J	SWI	2A	軟體中斷	SWI Cx	LR ← PC; PC ← Cx
	J	CALL	2B	跳到副程式	CALL Cx	LR← PC; PC ← PC+Cx
	J	RET	2C	返回	RET	PC ← 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; (byte)
	A	POPB	33	彈出 byte	POPB Ra	Ra = [SP]; SP++; (byte)