

□『系統程式』─課程作業(1)

∄ File Name: E236600-H-AY111A-01.doc

### 一、注意事項:

- (1) 本作業繳交方式:以上傳電子檔方式繳交,並應配合下列之說明。
  - (A) 請使用本作業電子檔之作答頁依題序完成本作業,於首頁填入個人識別 資料後,以[學號]-HW01.doc 之格式命名存檔並繳交。
  - (B) 將完成本作業所有相關檔案(含本作業檔案)收集於一資料夾內(資料夾之 命名方式:[學號]-HW1),並附資料夾內容說明檔;將該資料夾壓縮後, 上傳至成功大學數位學習平台(Moodle)之指定位置。壓縮檔名之格式為: [學號]-HW01.(rar | zip | 7z),壓縮格式可為 rar, zip 或 7z。
- (2) 切勿更改本作業原始版面設定,並依規定保存所有相關資料與程式等檔案。
- (3) 獨立完成作業,如有抄襲或剽竊情事,除酌情議處並扣減學期成績。
- (4) 評分基準:各題配分×85%,優增劣減。遲交酌減得分,扣減方式另行公佈。

### 二、<u>基本資料</u>:

繳交期限: 2022/10/27 繳交日期: 2022/10/21

學號: F64096114 姓名: 郭家佑 班級: 電機113

- 三、作業內容:(Main Theme Getting familiar and practicing with SIC and its assembler)
  - (1) 試比較 Object program 與 Executable program 之異同。 (10 points)
  - (2) Do the exercise 5 in page 40 of the course textbook. (10 points)
  - (3) Do the exercise 4 in page 112 of the course textbook. (25 points)
  - (4) 請使用 SIC 或 SIC/XE 組合語言設計如下規格之程式片段:
    - (4-1) 定義一名為 MATRIX 之 8×8 且未初始化整數矩陣。 (5 points)
    - (4-2) 儘可能有效率地將(4-1)定義之 MATRIX 初始化為單位矩陣。(20 points)
    - [註] ① 僅需依題意完成原始程式碼片段,無須繳交完整原始程式碼。
      - ② 建議善用程式碼註解,以助益原始程式碼說明
  - (5) 請使用 SIC 組合語言設計如下規格之程式:

(30 points)

- 輸入資料: 整數 A 與 B,且 0 < A, B < 1000。
- 輸出資料: 整數 A 與 B 之最大公因數,GCD(A, B)。格式為

GCD(A, B) = n

[例] 設輸入之整數為 625 與 75,則輸出為 GCD(625,75) = 25

Course Textbook: "System Software – An Introduction to Systems Programming", 3rd Ed.





Leland L. Beck / 汪 柏 導讀, (Pearson/高立圖書有限公司) © 2016.





✍ 課程作業作答頁

### ≤ 請使用本頁起之空間完成作業。(注意事項:請標明題號並依題序完成作業。) (1)

Object program 是把高階語言經過 compiler 和 assemble 加工後所得到的 code, 其 program 內容是經由 assembly program 轉換成對應的 object code, 在經由 整個 object code 對照各個 record 的 format 編排出一個由 record(+program name) + HEX 碼組合而成的 Object program。

Executable program 則是由 Object program,透過 linker 和 loader 把所需要 用到的像是 function 1brary 通通加進程式內,最後製造出只有 binary 表示的 code •

而雨者相同的地方就在於都是用數字來表示(一個 HEX 一個 BIN),均是人們沒辦 法直接看 code 就能了解其在做什麼。

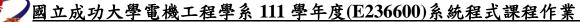
而相異的地方除了一有用 record 來標示整個 program 格式外,還有可讀性的差 別。像是 Object program 還可以利用對照 record 還原 object code,再查表還 原 assembly code, 就大致能了解 program 各步驟在做甚麼;而 Executable program 就比較難還原成可以用指令形式來解讀了。

(2)

5. Write a sequence of instructions for SIC/XE to divide BETA by GAMMA, setting ALPHA to the value of the quotient, rounded to the nearest integer. Use register-to-register instructions to make the calculation as efficient as possible.

LDF	BETA	
DIVF	GAMMA	
FIX		
STA	ALPHA	
ALPHA	RESW	1
ВЕТА	RESW	1
GAMMA	RESW	1





🗷 課程作業作答頁

(3)

4. Suppose that you are to write a "disassembler"—that is, a system program that takes an ordinary object program as input and produces a listing of the source version of the program. What tables and data structures would be required, and how would they be used? How many passes would be needed? What problems would arise in recreating the source program?

### Required and used:

我們必須要有如下圖的表格,將 input 進來的 object program 依照 record type 對應的 format 轉換成一個一個 object code。

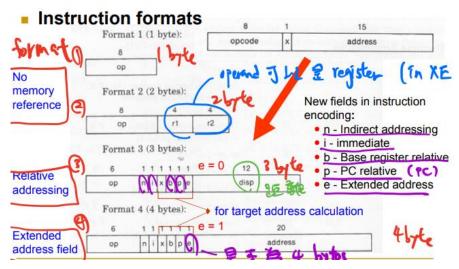
## **SIC/XE Object Program Format**

Record Type	Field	Content/Description	Format
Record Type			
	Col. 1	H	Char
Header	Col. 2 ~ 7	Program name	Char String
пеацег	Col. 8 ~ 13	Starting address of the object program	Hex(6)
	Col. 14 ~ 19	Length of the object program in bytes	Hex(6)
	Col. 1	Т	Char
Tout	Col. 2 ~ 7	Starting address of the object code	Hex(6)
Text	Col. 8 ~ 9	Length of the object code in this record	Hex(2)
	Col. 10 ~ 69	Object code (2 col. per byte)	Hex(60)
Modification	Col. 1	М	Char
	Col. 2 ~ 7	Starting location to be modified	Hex(6)
(Relocation)	Col. 8 ~ 9	Length of address field to be modified	Hex(2)
End	Col. 1	E	Char
End	Col. 2 ~ 7	Transfer address of the object program	Hex(6)

E2-166 (E236600) 系統程式

U02- Assemblers

有了 object code 後,我們還要再利用下圖 instruction formats 將 object code 解碼出他的 opcode, n, i, x, b, p, e 和 disp, 就能轉成我們可解讀的 assembly code 了。







☎ 課程作業作答頁

而 structure 的部份, 我們需要 1. Opcode table 2. Symbol table 3. Location counter。

Opcode table 可以把前 8 bits 翻譯出來,可以順便知道他的 n 和 i 是多少,就能知道的的 TA 是不是 15 bits 了(若 n=i=0),或沒有 TA(i=1),而再加上 extended bit 就能判斷 TA 是標準得 12 bits 或是 20 bits。

Symbol table 則是利用 M record 將一些會跳去絕對位置的 label 搭配 Location counter 給紀錄下來。

#### Passes:

至少要有兩個 pass。第一個 pass 可以將 object program 的 T record 轉成一行一行的 object code, 並順便記錄下 M record 可能記載的 label。第二個就是將一行一行 object code 轉為 assembly code 了。

#### Problem:

雖然網路上有找到會有 opcode 和 n, i 混在一起無法分辨的情況發生,但我覺得這個可以輕鬆克服,因為我們一般查的 opcode 都是以 n=i=0 來算的 8bits,而若是這個 8bits+3 的範圍內都是同樣的 opcode,只差在 n, i 不同而已。

另外一個網路上說會有問題的就是 TA 是 12 bits 還是 20bits,這點也是可以用 e 來判斷的。

而我認為會有問題的只有 label 名子不知道要還原成甚麼而已,其他認真分析下都可以解決。





✍ 課程作業作答頁

(4)

4-1

```
LDA #8
STA ROW
STA COLUMN

ROW RESW 1
COLUMN RESW 1
MATRIX RESW 64
```

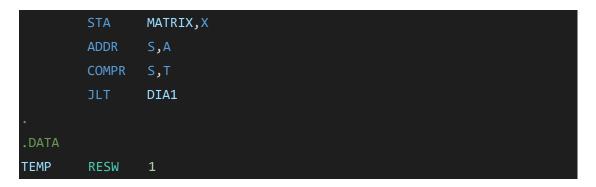
#### 4-2

- <b>-</b>			
• • •			
	LDA	#8	
	STA	ROW	
	STA	COLUMN	
•			
	LDX	#0	
	LDS	#3	WORDBYTESIZR
	LDT	#192	
•			
INI0	LDA	#0	
	STA	MATRIX,X	
	ADDR	S,X	
	COMP	X,T	
	JLT	INI0	
•			
DIA1	LDS	#0	MATRIXINDEX
	STS	TEMP	ZERO TO SEVEN
	LDX	TEMP	CALCULATE INDEX
	LDT	#8	MAXVALUE and MULTIPLICAND
	LDA	#3	WORDSIZE
	MULR	X,S	
	ADDR	X,S	
	MULR	X,A	
	LDA	#1	UNIT MATRIX ELEMENT



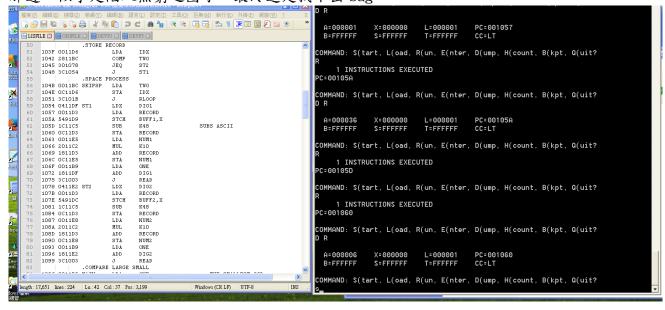


赵 課程作業作答頁



(5)

最後一題我有成功產生 ERECORD,但是再跑 SIMULATION 時,最後會卡在 OutDevice 那邊,似乎是陷入無窮迴圈了,最終還是找不出 bug







赵 課程作業作答頁

```
L=000001
T=FFFFF
LISFILE X GOBIFILE X GDEVES X GDEVFS X
   103F 0011D6
1042 2811BC
1045 301078
1048 3C1054
                                                                         COMMAND: S(tart, L(oad, R(un, E(nter, D(ump, H(count, B(kpt, Q(uit?
                                                                          1 INSTRUCTIONS EXECUTED
C=00105A
                           COMMAND: S(tart, L(oad, R(un, E(nter, D(ump, H(count, B(kpt, Q(uit?
                                                                          A=000036 X=000000 L=000001 PC=00105A
B=FFFFFF S=FFFFFF T=FFFFFF CC=LT
                                           SUBS ASCII
                                                                        {\tt COMMAND: S(tart, L(oad, R(un, E(nter, D(ump, H(count, B(kpt, Q(uit?}
                                                                         1 INSTRUCTIONS EXECUTED
PC=00105D
                                                                         COMMAND: S(tart, L(oad, R(un, E(nter, D(ump, H(count, B(kpt, Q(uit?
                                                                          OMMAND: S(tart, L(oad, R(un, E(nter, D(ump, H(count, B(kpt, Q(uit?
                                                                                                 L=000001 PC=001060
T=FFFFFF CC=LT
                                                                          OMMAND: $\tart, L(oad, R(un, E(nter, D(ump, H(count, B(kpt, Q(uit?
                                          Windows (CR LF) UTF-8
```

▲讀檔確認是對的,但最後指令大於 5000 都會在寫檔那邊跳來跳去

