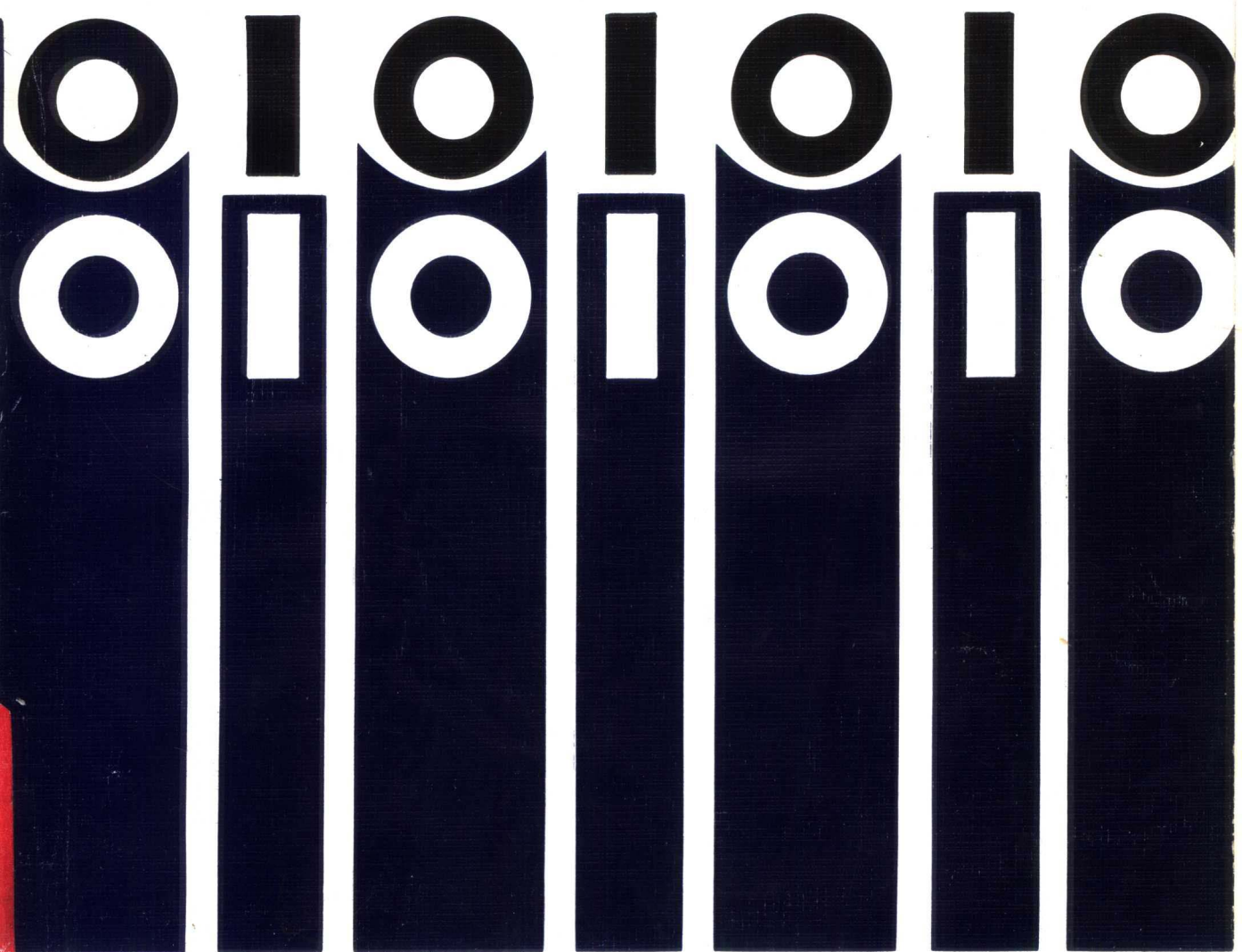


V 數位系統電路實作叢書之四

自己製作微電腦

蔡明介 編譯

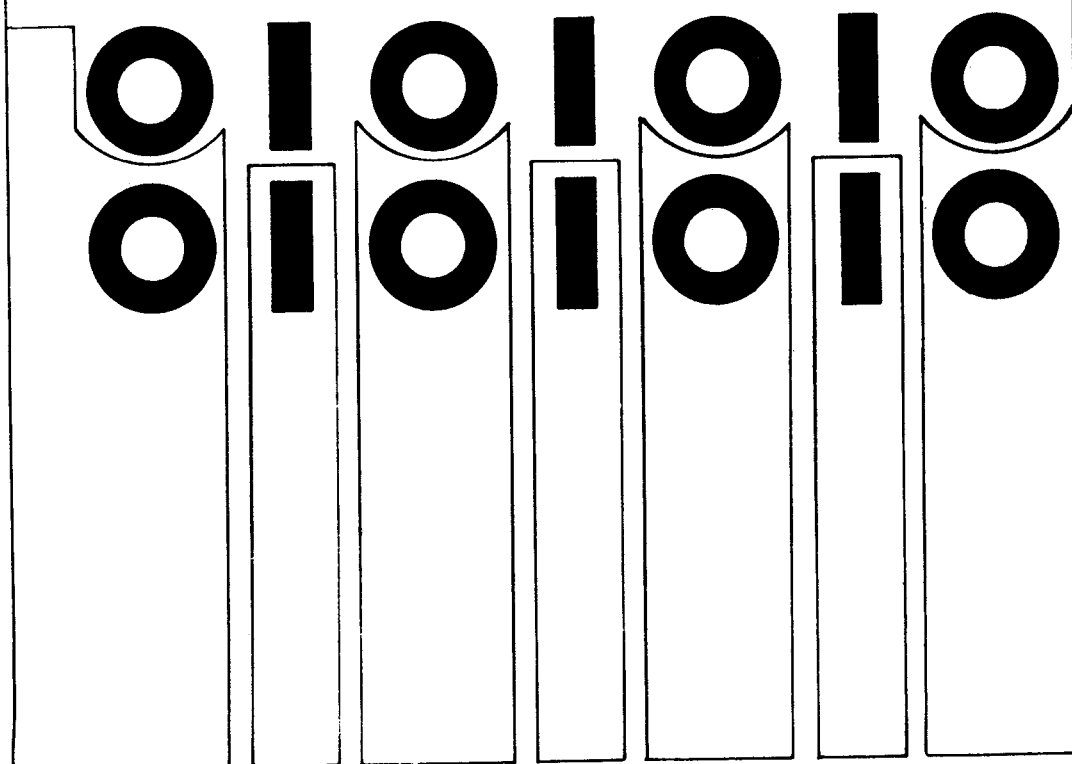


全華科技圖書公司印行

數位系統電路實作叢書之四

自己製作微電腦

蔡明介 編譯



全華科技圖書公司印行



全華圖書 版權所有 翻印必究

局版台業字第0223號 法律顧問：陳培豪律師

自己製作微電腦

蔡明介 編譯

出版者 全華科技圖書股份有限公司
北市龍江路76巷20-2號
電話：581-1300・541-5342
581-1362・581-1347
郵撥帳號：100836

發行人 陳 本 源
印刷者 華一彩色印刷廠
定價 新臺幣 150 元
再版 中華民國72年7月

數位系統電路實作叢書

總序

數位 IC 的製作成功在電子儀器、控制電路或其他電路設計技術上引起了一次「數位化革命」。由於數位技術的突飛猛進，對於從事電子電路設計者必須不斷吸收新知識，才能趕上時代的潮流。在工業進步快速的時代中，時間就是金錢，選擇一本良好的參考書才能使您以最少的時間和精力獲得最新的科技知識。

全華爲了推展此一科技知識，特地從日本引進這一套實用的「數位系統電路實作叢書」。本書最大的特色在於提供此行業的技術人才進修之用，同時也提供給正在學習的朋友們一個很好的機會。本叢書已出版四冊：第一冊爲「數位儀器之製作」，列舉實驗室、家庭中、汽車上與業餘方面各種實用電路，供讀者參考與實作。第二冊爲「數位電路之設計與研究」，分基礎篇與實務篇，由淺入深引發讀者思考、分析與設計能力。第三冊爲「計算機之製作」，分別介紹軟體與硬體技術，並以實例說明如何利用便宜的單晶片微處理器 IC 來製作計算機，讀者可經由實作過程而對硬體與軟體都有進一步的瞭解。第四冊爲「自己製作微電腦」，介紹 Intel-8080 與 F-8 兩種系統，並介紹數種套件，著重實際微電腦系統的製作，程式方面以實用有趣爲主，可增進讀者對微電腦的興趣與瞭解。

本叢書出版以來，深獲讀者好評，爲了迎合數位技術的不斷更新，我們將繼續爲您推出此一系列的新技術和新知識，讓我們共同爲科技發展而努力，使我們的電子界趕上先進國家。

自己製作微電腦

序 文

本書具有一般日文科技書籍的特色——實用，是欲熟習程式應用技巧者一本很好的參考書，而且書中之程式除了實用之外更兼具趣味性。

本書以Intel 8080 為主，書中尚有許多有用的硬體與界面電路，亦是對一位平常工作上須用到硬體與軟體技術者很有用的材料。另外書中並介紹了與 8080 結構不同的F-8系統，亦可增加讀者對微電腦之認識及使用技巧。

目 錄

第一章 FAMCOM-80之製作與尋找情人機器..... 1

8080 微電腦之概要 / 8080 CPU 之構造 / 指令存數器 (Instruction Resister) 及指令控制器 / 資料匯流排及緩衝器 / 8080 CPU 之指令 / 8080 CPU 之界面 / 8224 定時產生器 / 雙向匯流排驅動器與系統控制邏輯 / 位址緩衝器之設計 / 記憶器及其界面 / I/O 之界面 / 家用電腦 FAMON-80 之製作 / FAMCOM-80 之 CPU / CPU 模組 / 記憶模組 / 記憶寫入 / 讀出模組 / 指令位址顯示 / 中斷控制 / I/O 匯流排緩衝器 / FAMCOM-80 CPU 之電源 / FAMCOM-80 CPU 之製作 / FAMCOM-80 之 I/O / I/O 位址之分配 / 指令碼 / I/O 現狀 / I/O 中斷之分配 / I/O 指令時序 / 紙帶讀字機之界面 / 紙帶打孔機之界面 / 打字機界面 / FAMCOM-80 之軟體 / FAMCOM-80 之 IPL / 情人尋找機 (LOVER FINDER) / LOVER FINDER 之說明 / 次常式 (SUBROUTINE) / 公用程式 (UTILITY PROGRAM)

第二章 摩斯密碼解碼實驗..... 75

無線電信 / 8080 軟體心基礎 / 以 8080 做成之摩斯碼解讀軟體 / 摩斯符號 / 程式用來接收信號 / 英文與日文之摩斯碼

第三章 用8080之視像遊戲.....109

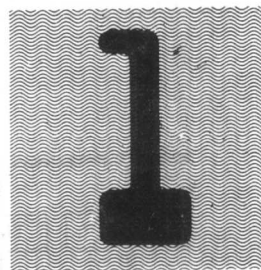
必要之硬體 / 軟體部份 / 次常式 / 流星遊戲 / 如何實行此遊戲 / 軟體構成 / 幾個問題

第四章 TK-80之製作與應用..... 125

以套件式製作電腦 / TK-80 之製作 / 硬體部份 / CPU 部份 / 記憶部份 / 顯示部 / PPI 與鍵盤 / 軟體部——監視程式 / 鍵指令 / 實際之製作程式 / 程式之實行 / 階段分離動作 / 次常式 / 卡式界面電路 / 界面電路 / 資料之送出與接收 / 界面電路 / 資料之送出與接收 / 記憶器之擴張 (2K RAM 之製作方法) / TK-80 之應用程式 / 數位定時器 / 電子琴 / 圖形發生器

第五章 F-8微電腦之製作..... 141

Fairchild F-8 之概要 / 匯流排之構成 / 程式記憶器 / 資料記憶器 / 輸出、輸入 / DMA / 信號分配進入 / F-8 之指令 / ACCUMULATOR GROOP 指令 / BRNCH 指令 / 記憶參考指令 / 位址存數器 GROUP (指示) / 草稿型紙型存數器組指示 / 機器指令 / F-8 之軟體除錯器之操作 / 硬體部份 / 面版控制程序 / START / EXAM , EXAMN / DEP , DEPN / STOP / STEP / START / JMP STOP



FAMCOM-80之製作與尋找情人機器

近來微處理機之價格十分便宜，我們於數月中謹慎的使用幾種型式的機器，發覺由此來入門似乎不錯。

在幾種微電腦中選擇 8080，其理由不是很明顯。「8080 是現在最流行的微處理機」是一種理由，事實上根據雜誌調查，在日本以微電腦作成之應用成品有 24.8 % 使用 8080。在此前數年這 8080 之姐妹品為 8008 系統，其 MCS-8 的使用手冊正可由此入手。

此章中，是使用 8080CPU 作成辦公用電腦及家用電腦，其構造為附 2K 數元組記憶之 8080CPU，I/O 是用讀紙帶機，打紙機，及 IBM 打字機接連而成，FAMCOM-80 即 Family Computer 之名稱而來。

●8080微電腦之概要

為了以 8080 作微電腦，先要說明其動作原理及構造，其 CPU 之動作了解之後，即可知道

只有 8080CPU 是不能做任何事之原理了。

8080 為微電腦之頭腦，而其週邊要有定時波產生器，現狀資料定鎖器，記憶器，以及輸出埠，這些均可以 74 系列產品之 TTL 來構成，但為了操作方便，用 Intel 公司之 8224（定時波產生器），8228（系統控制器），8212（I/O 輸出入埠）等均可。本文要先說明這些方便之 IC，而請在研讀此說明時，同時參考 Intel 公司出版之 8080 微電腦之系統使用者技術手冊。

●8080CPU之構造

首先表 1 為 8080CPU 之性能的大略表示。

圖 1 是 8080CPU 之方塊圖，8080CPU 由以下之基本單位構成

1. 存數器列與位址邏輯電路
2. 算術與邏輯演算部 (ALU)
3. 指令存數器及指令控制部

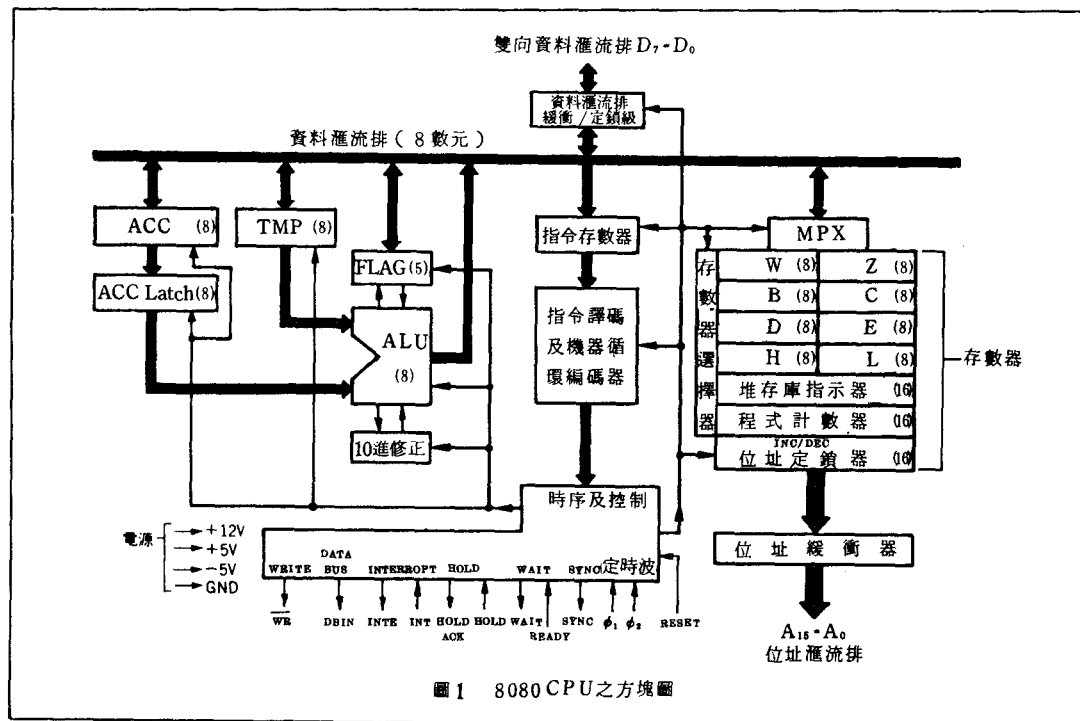


圖 1 8080 CPU 之方塊圖

表1 8080 CPU之大概

語 命 令	長 8 數元
命 令 語	8 , 16 , 24 數元
命 令 數	78 數元組
記 憶 器	64 K 可直接取位
輸 入 口 數	輸 入 共 256
取 位 方 式	直接 存數器 經由存數器間接 緊接
存 數 器	ACC (8 數元) 通用存數器 , 6 個 SP (16 數元) PC (16 數元) 信號存數器 (5 數元)
中 斷	8 層
流 動 排	資料流動排 (8 數元) —— 雙方向性 3 狀態 流動排 (16 數元) —— 3 狀態
定 時 波	0.48 μ S (2.08 MHz) $\sim$ 2 μ S (500 KHz)
電 源	+12V (70mA) , +5V (80mA) , -5V (1mA)
包 裝	40 腳 DIP
其 他	DMA 機能 , READY 機能

4. 雙向之三態滙流排緩衝級

以下說明這些功能

■ 存數器列為 16 數元之 6 個靜態 RAM 構成

，此存數器具具有下列之分別

1. 程式計數器 (PC)
2. 堆存庫指示器 (SP)
3. 6 個 8 數元之通用存數器 (B, C, D, E, H, L 存數器), 2 個爲一對
4. 暫時存數器 (W, Z 存數器)

PC 是在執行指令時，保持指令位址之存數器，指令之取得（fetch）時，（由記憶器讀出此等指令），此程式計數器之內容即要自動的改變。

SP 是保存存數器及程式計數器之內容與記憶器之位址的一種存數器，在此可由程式設定其位址。

6 個通用存數器可作為單一存數器或存數器對，因其為通用存數器，故可輸入資料或作位址輸入時參照記憶器之用。不用說 8080 CPU 之記憶位址為 16 數元，此時即為存數器對之使用場合，暫時存數器 W 與 Z 是 CPU 執行指令時之工作存數器，因此為草稿紙型記憶器（Scratch - Pad Memory），不能由程式直接取出或寫入資

料。

內部流動排及前面之存數器列中之資料傳輸，則由存數器選擇器及多選一器置於其中來完成，在此存數器列及位址定鎖器或增加／減少部份間的 16 數元資料之傳輸亦是可能的。這些定鎖器與增／減器則由用來驅動 16 數元之緩衝級，因增／減器是由定鎖器中取得資料，這些亦可送到存數器列。

■ 算術邏輯演算部 (ALU)

ALU 之意義即加，減等算術運算與邏輯和，邏輯積等邏輯運算之執行部份，此外資料之移位與轉位運算亦在此執行。8080 之 CPU 的 ALU 可處理 8 bit 之資料。此種 ALU 內有以下之存數器，即：

- 累積存數器 (accumulator register, 8 數元), ACC
- 暫時累積器 (Temporary register, 8 數元), ACT
- 信號存數器 (flag register, 5 數元), FREG
- 暫時存數器 (Temporary register, 8 數元), TMP

此 ALU 將 TMP 與 ACT, FREG 內的進位正反器之內資料輸入，在 ALU 內運算，把輸出

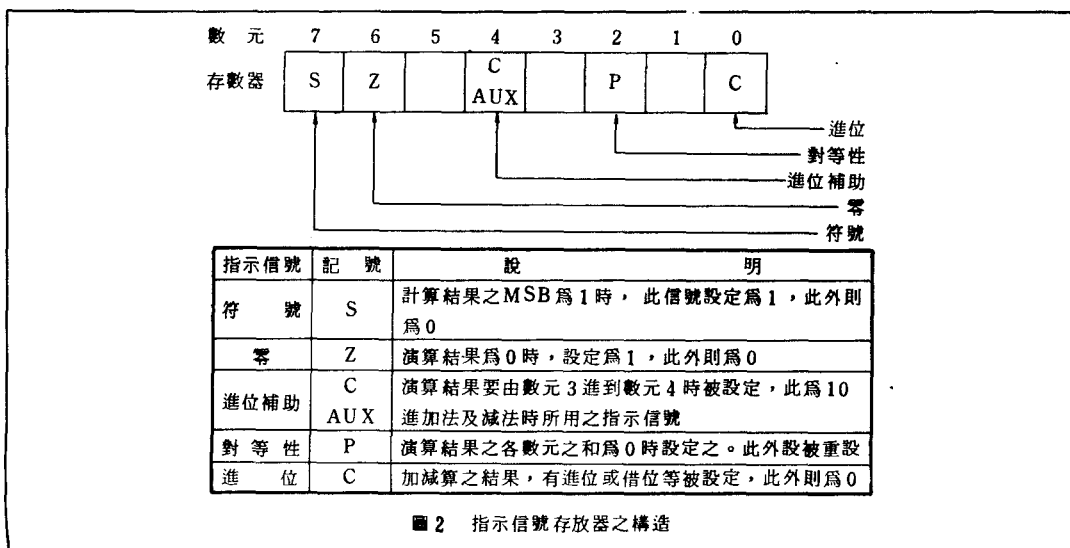
由內部信號匯流排與 ACC 來傳送資料。

TMP 是把資料由匯流部取出，此資料之全部或一部份由 FREG，經由內部匯流排送出，在 ACC 內由 ALU 及內部資料匯流排把資料饋入，而且此 ACC 可把資料送到 TMP 及內部之匯流排。

FREG 是把 ALU 內運算結果表示出來的地方，其中有 5 個正反器這些信號因條件跳出 (Conditional jump) 之指令或條件叫出之不同而改變。圖 2 為此 FREG 之構成及各信號之意義。

● 指令存數器 (Instruction Resister) 及指令控制器

8080 CPU 在取出指令時，由記憶器中取得資料 (指令之 OP code)，在此 IR 中儲存，其次再經指令譯碼器找出指令內容，此譯碼器之輸出即為各種時變信號之組成，而為存數器，ALU，及資料緩衝器之各種控制信號。此外，此輸出信號及外部控制信號再與狀態控制部配合，而成為狀態時間信號及循環時間 (Cycle timing) 信號。



●資料匯流排及緩衝器

8080 CPU 用 8 數元雙向，三狀態之緩衝器組成，可以把 CPU 之內部與外部資料分離，當 CPU 為輸出態時，內部資料即在資料定鎖器（latch）中，此定鎖器可驅動資料緩衝器，此資料輸出緩衝器在 CPU 為輸入態時及資料不傳送時為 OFF 狀態。CPU 為輸入態時，由外部匯流排通過資料緩衝器送入內部資料，上述之控制信號由必需的存數器取得。

●8080 CPU 之動作

首先討論 8080 CPU 動作之循環方法；1 個指令循環之意為將指令由記憶中取出而加以執行所需之時間。指令取出時，由 PC 指定的位址之記憶中把指令讀出，再把此存在 CPU 內之指令存數器中，其次此指令再經譯碼，前者為取出循環，後者為執行循環。

8080 CPU 之指令循環，因指令而有 1 ~ 5 個機器循環（machine cycle）。此 CPU 在出入記憶器或 I/O 出入口時要一個機器循環。指令中之取出循環，在指令取出 1 數元組時亦要一機器循環。8080 之 CPU 指令為 1 ~ 3 數元組，因而指令取出之時間需 1 ~ 3 機器循環，指令循

環中之執行循環隨指令之不同而要不同之機器循環，有些指令在取出循環時亦可作執行。有些指令之記憶器與 I/O 之出入時即要 1 ~ 2 個機器循環。

DAD 指令（存數對之並排運算）為一例外，在此種指令中，記憶與 I/O 不需每次參照，而指令之執行需二機器循環，此 8080 CPU 之 ALU 為 8 數元，在兩倍字長之加法時，CPU 可顯示出其可做運算之能力。

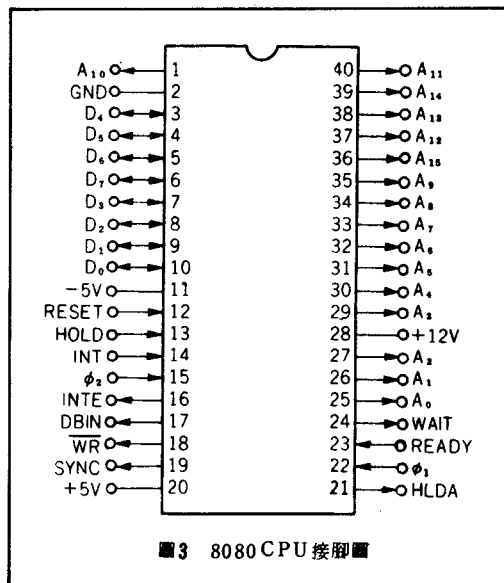


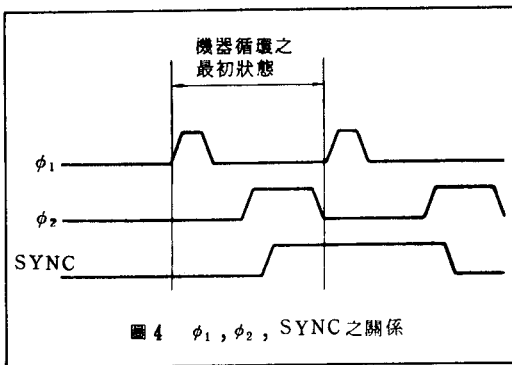
表 2 8080 CPU

接腳名稱	I/O	說明
A ₁₅ —A ₀	O	位址流動排，位址及 I/O 出入口之選擇用
D ₇ —D ₀	I/O	資料流動排，CPU 與記憶和 I/O 間之資料傳送用
SYNC	O	同步信號：CPU 之各機器循環之開始
DBIN	O	資料流動排輸入時之表示信號
READY	I	由記憶來或 I/O 來的資料已在資料流動排上時向 CPU 傳達之信號
WAIT	O	CPU 在等待狀態
WR	O	由 I/O 之資料寫入之信號
HOLD	I	CPU 成為 HOLD 狀態之必要的信號

接腳名稱	I/O	說明
HLDA	O	CPU 已在 HOLD 之狀態之信號
INTE	O	中斷使動信號，此表示 CPU 內之中斷正反器之狀態
INT	I	中斷要求信號
RESET	I	此信號為 H 時，CPU 內之 PC，INTE，HLDA 等正反器均被清除
V _{SS}		GND
V _{DD}		+12V±5%
V _{CC}		+5V±5%
V _{BB}		-5V±5%

以下將圖 3 中 8080 之各接腳在表 2 中各腳之功能做一說明。前述之各個機器循環中有 3 到 5 個狀態，此狀態為 8080 CPU 之處理資料的最小單位，而由定時信號 ϕ_1 之周期而定。

8080 CPU 之定時信號為由二不重合之定時信號 ϕ_1 及 ϕ_2 來產生。這兩個定時信號由外部電路供給，此 CPU 之控制邏輯電路之时序，即用此二定時信號，各機器循環之開始由同步信號 SYNC 之發生來表示出，此 SYNC 信號在 ϕ_2 信號之上升時觸發而產生，圖 4 為 ϕ_1 ， ϕ_2 ，及 SYNC 之時間相關圖。



狀態之時間關係有三個例外 (WAIT, HLDA 及 HALT 狀態)，這三個狀態與 CPU 外部之條件有關，其持續時間不能以一定的方式決定。然而，這些狀態亦是定時信號來同步控制的，因而其持續時間為定時信號週期之整數倍，以上所得之狀態由定時信號之週期決定，此狀態每 3~5 個即為一個機器循環，而每 1~5 個機器循環即為一指令循環，欲執行一指令，即需 4~18 個狀態。

■ 機器循環之分別

前述之 DAD 指令為例外之外，機器循環有一定的規則可行。這就是指令之取得及執行時，記憶與 I/O 出入埠共參考之次數為幾次而言。與其他許多 CPU 相同的，在 8080 的一個機器循環中，由一個位址中不能輸出資料，因而在指令之取得與執行時，必需與記憶有數次的互相參考，此指令之執行即因此要有數個機器循環。

以存數器加法運算為例 (ADD r)，此指令之取得與執行時，不必用到 1 個機器循環，此指令之工作為，把 6 個通用存數器中之一個的內容，加上累積存數器中之內容。此種指令中，CPU 必須做的事是把 1 數元組記下來，而與相對的記憶參照，所以開始之 3 個狀態為命令之取得，其次之狀態即為指令之執行，於是此指令即為一由 4 個狀態組成之機器循環完成。

其次看看記憶加法運算指令 (ADD M)，此指令為 H 及 L 存數器所指定之記憶位址的內容加到累積存數器中。此指令亦為 1 數元組長，其運算與前例十分相似，欲加之資料由記憶中讀出，其指令之執行亦要有一機器循環以上。

此指令之運算如下，最初之機器循環中之 3 個狀態為指令取得用，次一機器循環中之 3 個狀態則把 H, L 存數器中之記憶內容讀出，同一機器循環之剩下的一狀態則用來做加法運算，因而此指令要用到 2 個機器循環，共有 7 個狀態。

在以要用到較多機器循環的 SHLD 指令為例。此指令為 3 個數元組長。其動作為把 H, L 存數對之內容，在同指令第二數元組與第三數元組中之內容的位址之記憶中存入。最初的機器循環之 3 個狀態是讀出指令中第一數元組之內容，次一狀態則予以譯碼，次 3 個狀態構成之 2 個機器循環則讀出指令第二與第三數元組中之內容。這些資料再存入 W、Z 存數器。H, L 存數器之內容則為記憶之位址。再次之 3 個狀態所成之機器循環則把 H, L 存數器內容寫入記憶中。因而此 SHLD 指令之實行要 5 個機器循環，而共用了 16 個狀態。

以上所述為 8080 CPU 之指令實行時需用 1~5 個機器循環。各個機器循環之動作則分成以下十類。

1. 指令取得 (M_1)
2. 記憶讀出
3. 記憶寫入
4. 堆存庫讀出 (stack read)

6 自己製作微電腦

5. 堆存庫寫入 (stack write)

6. 輸入

7. 輸出

8. 中斷 (interrupt)

9. HALT (CPU停止)

10. HALT之中斷

各種指令之第一機器循環為取出，其他循環則因指令之不同而不同，8080 CPU之各機器循環的第一狀態，則把8數元組成之現狀計數 (Status Count) 送到外部回路，此現狀與 SYNC 信號同步時，亦送到 CPU 之資料流動排。外部回路就把此現狀定住，然後發生各種控制信號。表3所示為此現狀之意義。

此外，此現狀與8080 CPU之接腳上之出來信號同為正值邏輯。圖5所示為此現狀之定住方法。原來 CPU 之現狀資料是在資料流動排上，

8080 CPU受 IC 四十支腳的限制，因而其現狀之分別法用現狀計數的方法。

■ CPU狀態之遷移

一個指令由3~5個狀態組成，已如前述。這些狀態即為 $T_1 \sim T_5$ ，以及 T_w 等名稱，圖6為一機器循環內之狀態變遷的形狀。此圖中之 READY，HOLD，及 INT 信號由 CPU 來取樣，如此可了解其中狀態變化發生之情形。

此處，首先由基本狀態遷移及 READY 信號之關係着手，HOLD 與 INTERRUPT 之時序在以後再予說明。然後，表4所列為對各狀態時之 CPU 動作加以說明。

8080 CPU將這些狀態有關的資料由 IC 之接腳送出 (INTE, HLDA, DBIN, WR, WAIT)，如此，這 CPU 之資料定住循環的過程可以追蹤回去。圖7為此循環中之時間圖。

表3 8080 CPU之現狀

資料流動排之數元		現 狀										← 現狀字型態
		指令之取得	記憶讀出	記憶寫入	堆存庫讀出	堆存庫寫入	I/O讀出	I/O寫入	中斷確認	HALT 確認	HALT之中斷確認	
		①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	
D ₀	INTA	0	0	0	0	0	0	0	1	0	1	← 現狀字型態
D ₁	WO	1	1	0	1	0	1	0	1	1	1	
D ₂	STACK	0	0	0	1	1	0	0	0	0	0	
D ₃	HLTA	0	0	0	0	0	0	0	0	1	1	
D ₄	OUT	0	0	0	0	0	0	1	0	0	0	
D ₅	M ₁	1	0	0	0	0	0	0	1	0	1	
D ₆	INP	0	0	0	0	0	1	0	0	0	0	
D ₇	MEMR	1	1	0	1	0	0	0	0	1	0	

(出現RST信號)

現 狀	說 明
INTA *	對應於中斷之要求信號，CPU加以確認，中斷控制回路，對應此在DB
W ₀	"L"時表示為I/O或記憶寫入，"H"時為讀出
STACK	CPU之AB上(A ₀ ~A ₁₅)之資料為SP之複製
HLTA	HALT指令之實行
OUT *	位址流動排示輸出位址，DB上WR為L時，資料則在流動排上
M ₁	CPU取得指令之第一數元組
INP *	AB示出入口之位址，DBIN為"H"時資料流動排上為輸入資料
MEMR *	位址及資料流動排為記憶讀出時之用

※付此種星號之信號為資料流動排上之資料流動方向控制用

表4 8080 CPU 之狀態定義

狀態	動作定義
T_1	記憶位址或 I/O 出入口之數目在 AB 上，現狀則在 DB 上，I/O 口之數目則與 $A_0 \sim A_7$ 和 $A_8 \sim A_{15}$ 同值
T_2	READY 與 HOLD 輸入之狀態的取樣，並核對有否 HALT 指令
T_w	T_2 時 READY 為“L”時，且 HALT 指令在實行，CPU 進入 WAIT 狀態
T_3	指令數元組（取出循環），資料數元組（讀、記憶、堆存庫之循環）或中斷指令（中斷循環）由資料流動排取出。而記憶，堆存庫寫入與輸出循環時，流動排上為資料。
T_4	T_4 、 T_5 對指令之實行為必要時則存在，不要時則跳過，
T_5	T_4 、 T_5 與 CPU 內部之動作有關。

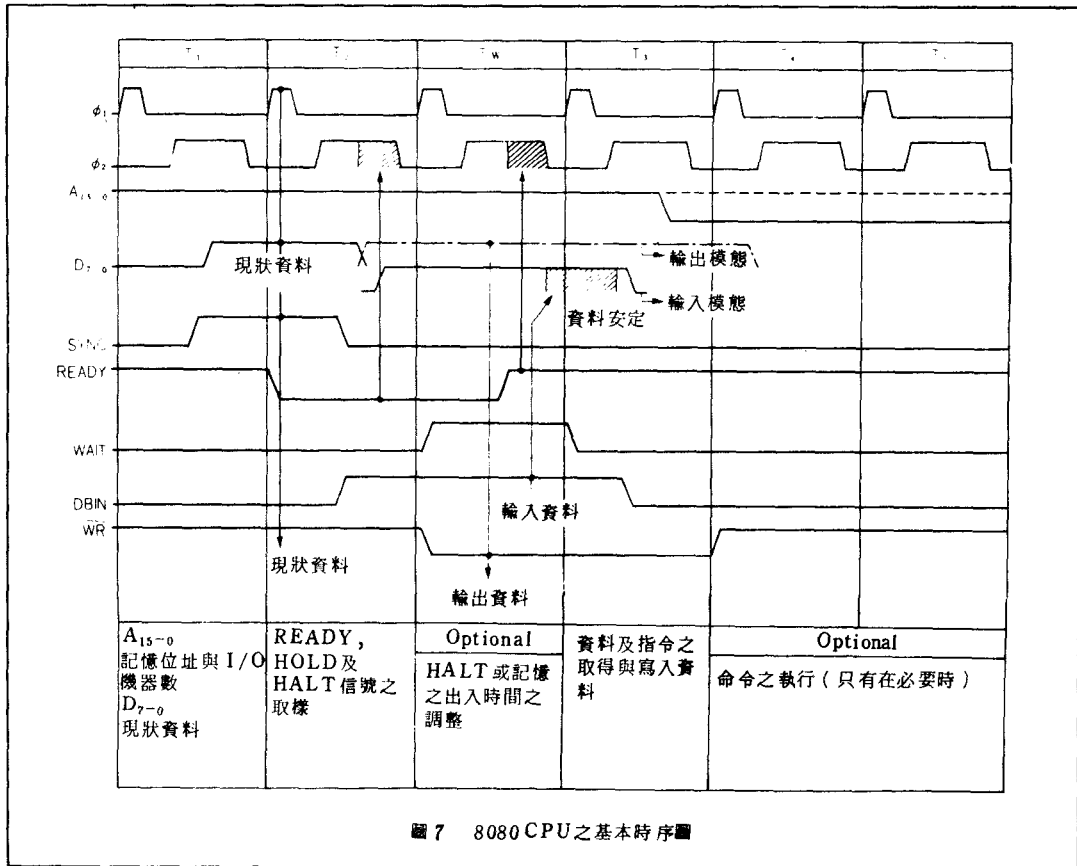


圖7 8080 CPU之基本時序圖

CPU 在此狀態 T_1 中 ϕ_2 之上升緣的延遲時間內 ($t_{DA} = 200\text{ns}$)，其位址即進入 $A_0 \sim A_{15}$ ，此位址在 T_3 狀態之 ϕ_2 的週期中為有效。

CPU 在記憶的相對位址上送資料時，記憶器之 READY 信號為 L，因而 CPU 要求之 WAIT 信號可以發生，此 READY 信號（第 23 支腳）在 T_2 與 T_w 狀態之 ϕ_2 的波幅為 H 之前 120ns 必需為 L，此時間稱為 READY SET

UP 時間 (t_{rs})，此 READY 信號變為“L”之後，CPU 之處理則暫時中止，而進入 IDLE 狀態。亦即 CPU 在 T_2 狀態終了之後，不進入 T_3 狀態，而進入 T_w 狀態，CPU 依 WAIT 信號之成為 H，而從外部電路得知進入 T_w 狀態。此 WAIT 信號在 ϕ_1 之上升邊緣時觸發，CPU 在 T_w 時間大約少於一個延遲時間 ($t_c = 120\text{ns}$)。

此 T_w 狀態持續到 READY 再次變成 H 時，

READY 信號為 H 之時間，比 ϕ_2 下降邊緣要前移 t_{rs} ，如此 CPU 在次一 ϕ_1 之上升邊緣時，再回到其處理之工作（進入 T_3 狀態）， T_w 之狀態進入時為 ϕ_1 時間做基準，而由 T_w 狀態再出來時亦由 ϕ_1 而出，因而 T_w 之持續時間為定時信號之週期的整數倍。

READY 信號通常是在“H”，因此不是 T_w 之狀態，記憶器之出入時間比定時信號之週期必須要更短，例如 2MHz 之定時信號使用時，記憶器之出入時間則必須在 500ns 以下。通常使

用之記憶器為 RWM 是 2102，PROM 為 1702 A，此二者均滿足此一條件。

因為定時信號延遲時間之考慮，此為 8080 之特有的優點之一，此使用之記憶器若不能與 CPU 之速度相匹配時， T_w 狀態即是為了使二者同步而產生。

在 READY 信號使用時，CPU 之單層步驟之運算更易實行。當 CPU 在取得指令內容時，READY 信號為“L”，由開關來之 READY 信號如為 H 時，則 CPU 可進一步至下一指令。

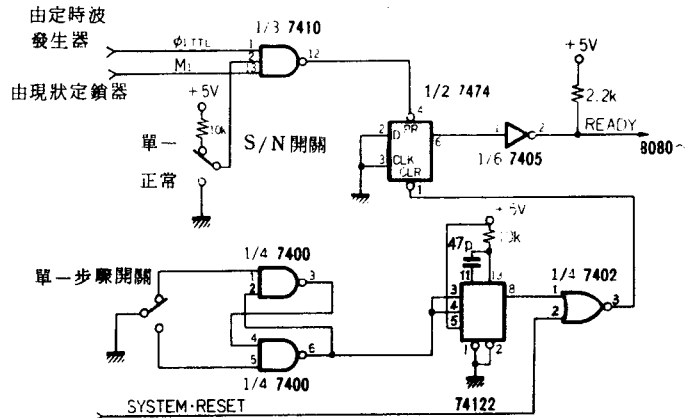


圖 8 READY 信號之單步驟控制

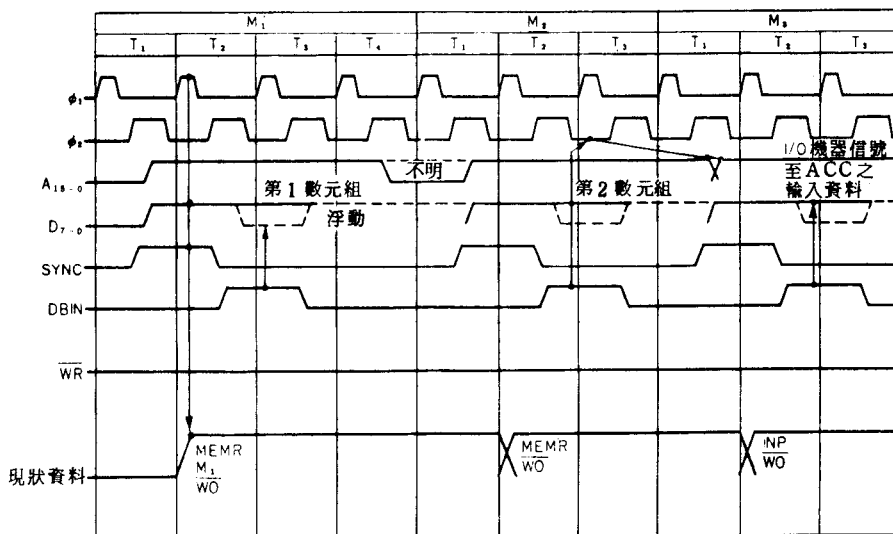


圖 9 輸入指令之時序圖

圖8 為用 READY 信號作之單層步驟控制器。

T_3 狀態之 CPU 的處理工作，隨機器循環之種類而異，取得循環時，資料流動排上之資料，因指令而不同，在記憶讀出或堆積存數器讀出循環時則與資料有關。在記憶寫入與堆積存數器之寫入循環時，CPU 將欲寫入記憶器之資料由資料流動排上輸送。在 I/O 指令時，資料之給予不在記憶上做，I/O 出入埠間則以同樣方式進行。

圖9 為由 I/O 之資料輸入之時間圖，此圖中，位址匯流排上之資料成為 I/O 位址，這位址如有記憶取位時，亦可用在記憶之讀出。CPU 則在 T_2 狀態之 ϕ_2 的上升邊緣，由資料匯流排取出資料，而接受輸入資料。

由 I/O 來之資料在 T_3 狀態之 ϕ_1 的下降前 30nS，而在同一狀態之 ϕ_2 的上升前 150nS 時即以安定下來，前者稱 ϕ_1 資料建立時間間隔 (t_{DS1})。

再者，此資料必須由 ϕ_1 上升邊緣起 t_{DB} 時間內 (100nS) 一直持續著，此時間則稱為資料保持時間間隔 (t_{DB})，I/O 或記憶之資料匯流排上之資料，在 T_3 狀態內被取樣到，而送到 CPU

之內。

在輸入工作時，CPU 送出 DBIN (第 17 支脚) 信號，記憶與 I/O 則用此一信號，把資料送至流動排上，此 DBIN 信號使用時之機器循環，其內容如下：

- (1) 取得循環
- (2) 記憶讀出循環
- (3) I/O 讀出循環
- (4) 堆積存數器讀出循環
- (5) 中斷循環

DBIN 信號在 T_2 狀態之 ϕ_2 上升時送出，而在 T_3 狀態之同一時間取出， T_2 狀態與 T_3 狀態間有 T_w 在的場合時，只要把 DBIN 之長度依其延伸即可。

圖 10 為向 I/O 輸出時之時間圖，此圖中，位址匯流排亦以 I/O 位址表示出，此在記憶取位址時，亦用來做記憶寫入指令用。CPU 在 T_2 之 ϕ_2 的上升時，資料匯流排上之現狀資料被清除，累積存數器之內容在資料匯流排上被設定，此動作要比 ϕ_2 上升時遲了 t_{DD} ($\approx 200nS$)，此時間被稱為資料輸出延遲時間間隔。此外，這些資料一直持續到此機器循環終了時為止。

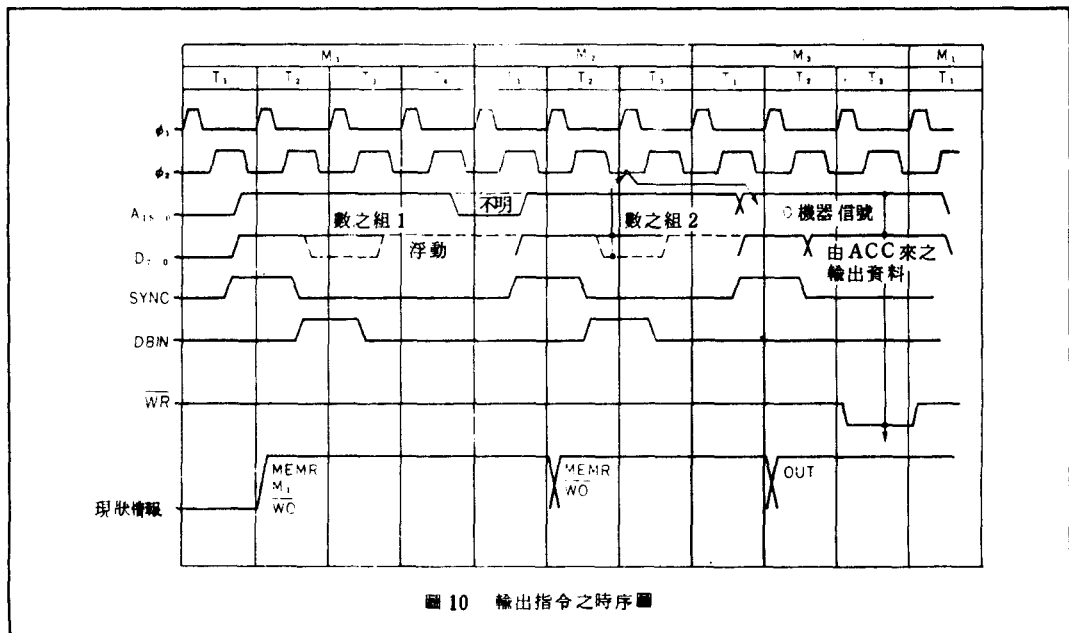


圖 10 輸出指令之時序圖

8080 CPU輸出時， $\overline{WR}$ 信號（第18腳）為輸出，I/O與記憶將為同步，此 $\overline{WR}$ 信號使用時之機器循環如下：

1. 記憶寫入循環
2. 資料堆積處寫入循環
3. I/O寫入循環

$\overline{WR}$ 信號下降與 T_2 狀態之後的 ϕ_1 之上升邊緣同步，其間只有 t_{DC} 之延遲，而此信號持續到 T_3 狀態之 ϕ_1 的上升邊緣為止。

與前述DBIN信號一樣，此信號也是在 T_2 與 T_3 狀態間有 T_w 狀態存在時，只要依 T_w 部份延長其時間。

CPU中的機器循環，到目前為止所述一定有 T_1 ， T_2 ， T_3 狀態。因此CPU要送資料，而要等待由記憶及I/O來的反應的情況下，會有幾個 T_w 狀態在其中。CPU之資料輸送時，在 T_1 與 T_3 狀態實行。

T_3 狀態終止後之CPU動作則與其情況之不同而異。各種指令實行之場合，有 T_4 及 T_5 狀態接續在 T_3 狀態之後。但是，並不是要一定有

T_4 及 T_5 狀態。此因各種指令及機器循環而變。CPU在有些機器循環之執行時有用到全部狀態，此點為機器循環之終點。因而，機器循環終了時有 T_3 ， T_4 及 T_5 等狀態。

■ 程序中斷次序

8080 CPU可由外部來的中斷指令使其接受中斷。I/O與計時器等在CPU之中斷要求信號（第14腳）為“H”時，使CPU接受中斷指令。

圖11為中斷次序之時間圖，由圖可知，

INT信號為H，而且INTE（使中斷信號，第16腳）亦為“H”時，內部之中斷正反器，由 ϕ_2 而觸發。此例之前後INTE不為“H”，則不接受中斷信號，此INTE實即為CPU內部中斷正反器狀態之表現。

此正反器由ENABLE INTERRUPT(EI)指令來設定。而由DISABLE INTERRUPT指令(DI)來歸零。此外，CPU在接受INTERRUPT時，此正反器被歸零。CPU內部之中斷正反器被設定時，即在指令循環之最後狀態。事實上中斷指令之接受是由目前執行中之

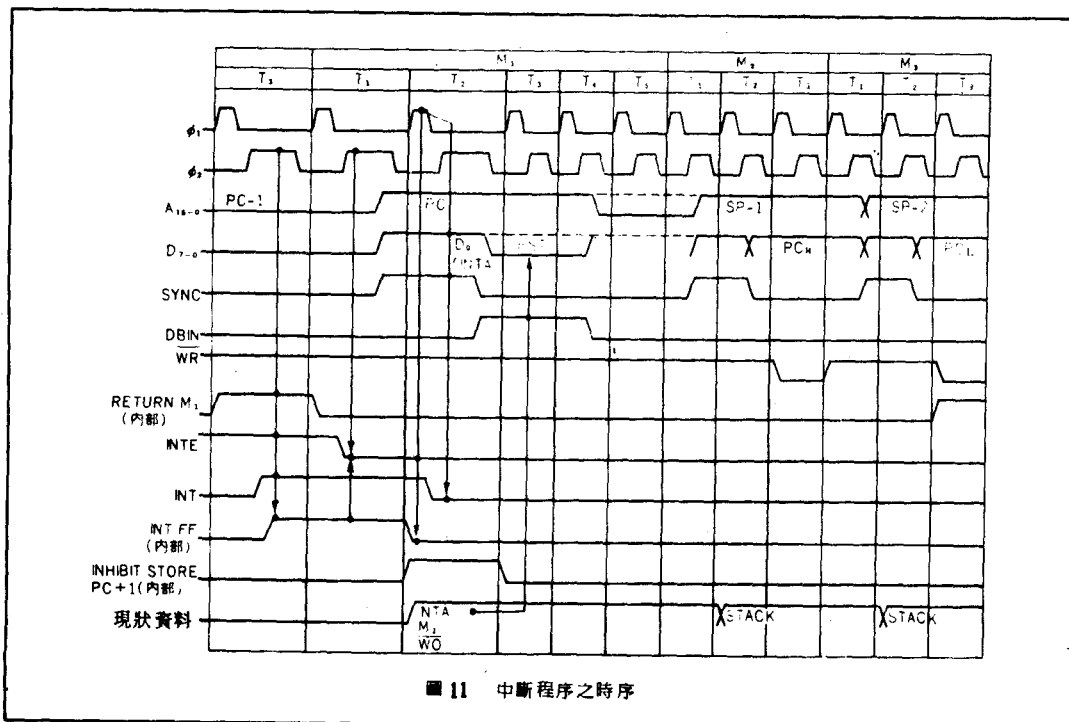


圖 11 中斷程序之時序