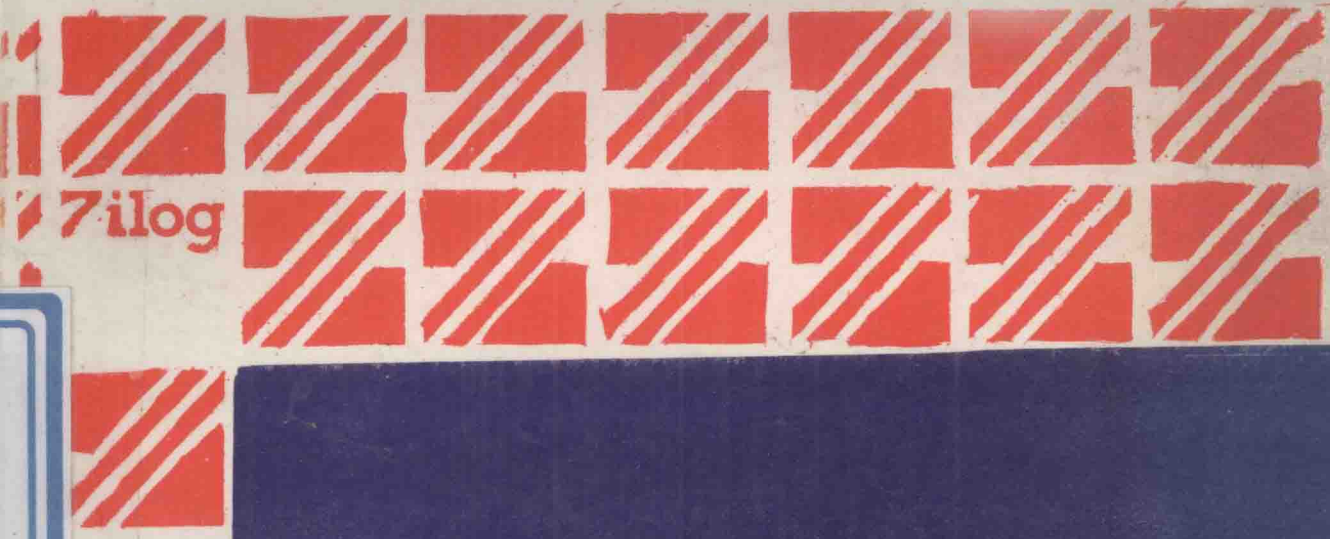
Zilog

Z-80
微算機結構

Microprocessor Handbook

蔣春木 編



Z－80 微型電腦是目前普遍爲學校及企業界採用的微電腦之一。但多年來一直沒有一本較好的教本做爲訓練之教材。

本社從事電子科技圖書出版工作多年共出版微電腦中、英文叢書60餘本，有感于實際之需要特編成本書以供學校教學之用。

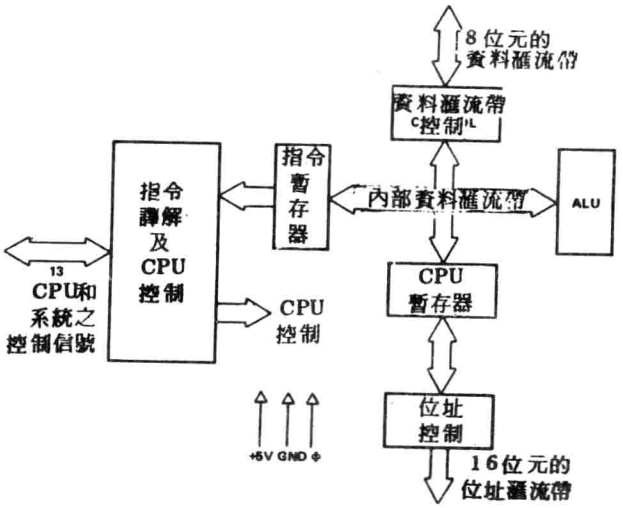
前鋒出版社　敬上

69. 9. 29.

目　　錄

第一章　Z-80中央處理單元

Z－80CPU之內部結構如圖A－1。在本圖中我們可以看出CPU內部的幾個重要方塊。其與外界之溝通，全靠8個位元的資料匯流帶，16位元的位址匯流帶以及13條CPU和系統之控制信號綫。



圖A－1 Z－80CPU的內部結構

在圖中其實只有二個是主角，亦即CPU內的暫存器以及算術／邏輯單元（ALU），其餘均是協助此二者的設施，現將此二者詳加解釋。

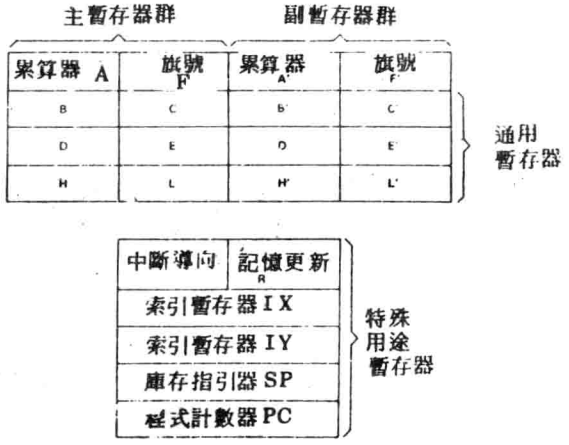
在Z－80內提供了208個位元之R／W記憶體，供使用者直接使用。這些記憶體又如同圖A－2般再予劃分成二套六個通用暫存器群，每個暫存器爲8個位元，二套累算器及旗號暫存器亦各爲8位元，累算器內保存算術及邏輯運算之後的結果，而旗號則記錄運算時發生之特殊現象，以及六個特殊用途的暫存器。後者之說明如下：

(1)中斷導向暫存器（I），又稱爲中斷頁位址暫存器。在中斷後，CPU需要一個16位元之位址值，以處理中斷之申請。I暫存器即爲此目的而設，它可提供首要八位元之高位址值，其餘的末要低位址值，則係由中斷之機件所提供的。

(2)記憶更新暫存器（R），此暫存器係提供做爲記憶位址之更新而用的，因此使得動態記憶體與靜態記憶般易於使用。

(3)索引暫存器（IX，IY），此兩個暫存器均爲16個位元，且各自獨立，但均可做爲索引定址法（參看下面文章）之基底之用。

(4)庫存指引器（SP），庫存指引器永遠指向RAM位址內作爲庫存使用的RAM之頂點。庫存係採用後入先出（CIFO，Last in First out）之結構，每做一次PUSH或POP指令均將改變其值，其值由16個位元組成。



(5)程式計數器（PC），程式計數器之內容係一個十六位元之位址，它永遠指向欲提取之記憶位址，而一旦將其位址值送出位址匯流帶後，其值將自動加1。

算術／邏輯單元，顧名思義，所有CPU內的8位元算術及邏輯運算均在此單元內完成，其職司相當多，包括：

加（Add）	旋轉及平移（Rotate，Shift）
減（Subtract）	增加（Increment）
及運算（AND）	減少（Decrement）
或運算（OR）	置定位元（Set）
排外型或運算（XOR）	復置位元（Reset）
比較（Compare）	測試位元（Test）

指令動作白對象，亦需要標示出來，指令動作的對象有暫存器，記憶位址，或是資料等。將執行之對象標示，尋找出來就稱爲定址，定址之模式在Z－80中計分成十種，我們將在下面詳加說明，在此，我們先研究指令的格式，指令應具有下列的格式：

操作碼	運算元1	運算元2

首先由操作碼（Op code）定出動作之性質，而後定出二個運算元，有時其中一個運算元就是累算器本身而且若是算術或邏輯運算，其結果亦放在累算器內。指令有時僅需一個位元組就可表示出來，有時則需要2，3，4個位元組。茲各舉一例以說明之。

單位元組；指令：ADD C

動作：A←A＋C暫存器

實體碼：10000001 或81H

雙位元組；指令：INC IX

動作：IX← IX＋1

實體碼：11011101 DDH

或

00100011 23H

參位元組；指令：LD A,(nn)

動作：A←(nn)；將nn位址內的值傳送入A中。

實體碼：00111010　　3AH

←— n —→　或　n

←— n —→　　n

肆位元組：指令：BIT b,(IX+d)

動作：Z←$\overline{(IX+d)_b}$　將位址(IX+d)內第b個位元的值取補數後，放入Z旗號正反器內。

實體碼：11011101　　DDH

11001011　　CBH

←— d —→　或　d

01←b→110　　X

從上面這些例子中我們亦可體會出定址法的重要，現將十個定址詳列出來。

立即定址法：操作碼之後的記憶位址上就存放著所要運算的資料值，其格式如

操作碼	←——1或2個位元組
運算元	←——1個位元組

例　LDH，36H，將譯解成

位址A	26	←— 操作碼
A+1	36	←— 運算元

延伸立即定址法：立即定址法，只定出一個位元組，此法則定出二個位元組，其餘均相同。

如

位址A	操作碼	←——1或2個位元組
位址A+1	運算元	←——低位元組 } 16位元之資料
位址A+2	運算元	←——高位元組

例　LD BC,1234H將編譯成

01
34
12

零頁定址法：此爲相當特殊的召用指令，其召用之位址爲頁0之記憶體內的八個固定位址。其操作碼格式如

11×××111

×××可爲任意值，而其指定之位址爲零頁上的00×××000位址上。

相對定址法：相對位址法出現在跳躍指令中，其內的數值代表下一個指令位址是由目前的指令往

前或往後幾個記憶位址。此數值即稱之爲補償值或位移值。格式如下：

位址A	操作碼	← 單位元組
A＋1	補償值	← 8位元，以2的補數爲表示負值 d
A＋2		

下一個指令所在的位址即是

A＋2＋d；d的大小爲＋127至－128之間。

例　　JR　$＋4

若目前JR之指令爲A，則將來之位址爲A＋4故得A＋4＝A＋2＋d

d＝2　故

18
02

完整定址法：將二個位元組的位址包含在指令之內，指令可據此跳躍至該處繼續執行，從該處取得資料或將資料置入該處。其格式如下，請注意位址如何安放。

操作碼	← 1或2個位元組
低位址	} 先放低位址，再放高位址。
高位址	

例　　CALL　LEBEL

若LABEL的位址爲1234H，則

位址A	CD
A＋1	34
A＋2	12

索引定址法：與相對定址法很相近，仍然要加上補償值，不過在此法中，位址A係存放在索引暫存器（IX或IY）內，而計算新位址時，亦不必再加2了。

例　ADD　A，（IX＋d）則經編譯後

位址A	CD
A＋1	86
A＋2	d

暫存器定址法：此法　在Z－80中，係定出指令所需要之暫存器的代名。

例　ADD　A，r其操作碼格式爲

10000×××

000＝B

001＝C

010＝D

011＝E

100＝H

101＝L

111＝A

內隱定址法：依字面意義可知，指令本身即已暗示出，指令係對那一個暫存器加以執行。

例　　ADD （HL）

此指令業已指出動作爲

A←A＋（HL），其操作碼爲

06H

暫存器間接定址法：利用一個16位元的CPU暫存器偶對內的位址來找出所需的記憶位址（故稱之爲間接），通常係利用HL暫存器偶對。索引暫存器亦是，不過得加上補償值。上例中之ADD （HL）即是將HL內的值看成位址，故加上括號。

位元定址法：Z－80中有許多的位元指令，可將個別置定，復置或測試某記憶位址內的某一位元。其法係在操作碼中保留三個位元以供定義出位元之位置。

例　　BIT　b，（HL）

位址A	CB
A＋1	01bbb110

其中

bbb之值	位元位置
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

指令執行之時，若要表示執行期間所發生的狀況，則需要利用F暫存器保留下來，存個底，以供將來測試之用，這些狀況分別用一個正反器來儲存，此正反器總稱爲狀態旗號，若右邊之條件滿足，則正反器之值爲1，否則爲0，茲表示如下：

進位旗號(C)：表示從累算器的首要位元處有進位（Carry）。

零值旗號(Z)：表示累算器內的值爲 0 。

符號旗號(S)：表示累算器之值爲負值，故 S 與累算器位元 7 之值相同。

比價 / 溢位旗號（P/V）：此旗號有雙重用途。邏輯運算時代表結果爲偶數比價，算術運算時則代表運算之結果超出累算器之容量，故得到錯誤的值。

半進位旗號(H)：代表加、減運算時，末要的 4 個位元有進位或借位之動作。

減法旗號(N)：代表進行之指令，爲減法運算。

這六個旗號前四個可供測試，後二者則不可測試，專供DAA十進制調節之用。

Z－80中共有158個不同的指令，這些指令係組合前面所有的ALU動作，定址法，旗號等而成的，且可以細分成8大分項：

- 置放和交換（ Load and Exchange ）
- 集體轉移和尋找（ Block transfer and Search ）
- 算術和邏輯運算（ Arithmetic and Logic ）
- 旋轉和平移（ Rotate and Shift ）
- 位元操縱指令（ Bit Manipulation ）
- 跳躍，召用和回歸（ Jump , Call and Return ）
- 輸入 / 輸出指令（ Input/Output ）
- 基本CPU控制（ Basic CPU Control ）

在下面數頁中，我們即是按此分類將每一指令之助憶代號，動作，實體碼，旗號，所需之位元組數，至於其時間（M週期數與T週期數，可參看附錄C）分別予以列出，希望能提供讀者諸君方便。

下列表中所用的符號說明如下：

A，F，B，C，D，E，H，L	都是8位元的暫存器，A是累算器而F是旗號暫存器。
AF′，BE′，CD′，HL′	是另一套暫存器偶對。
ADDR	代表16 位元的記憶器位址。
X(B)	是8-位元 的暫存器或記憶器中位址X的第B號位元。
COND	程式跳動的條件。這些條件是：
	NZ — 不是零 （Z＝0）
	Z — 是零 （Z＝1）
	NC — 沒有進位 （C＝0）
	C — 有進位 （C＝1）
	PO — 奇數個“1” （P＝0）
	PE — 偶數個“1” （P＝1）
	P — 正 號 （S＝0）
	M — 負 號 （S＝1）
DATA	代表一個8 位元二進制的資料。
DATA16	代表一個16 位元二進制的資料。
DISP	代表一個8 位元 帶正負號的補償值
XX(HI)	是16 位元數值XX的高位8 位元。
IV	中斷導向暫存器（8 位元 ）。
IX，IY	是各16 位元的索引暫存器。
LABEL	是16 位元的指令位址。
XX(LO)	是16 位元數值XX的低位8 位元 。
PC	程式計數器（Program counter ）
PORT	是I／O接頭的位址（8 位元 ）。
PR	代表暫存器偶對：
	BE，DE，HL， 或AF。
R	是更新暫存器（8 位元）。
REG	代表單個的暫存器。
	A，B，C，D，E，H，或L。
RP	代表下列任何一個暫存器偶對：
	BC，DE，HL，或SP 。
SP	庫存指引器（16位元）。

STATUS	C — 進位狀態
	Z — 零狀態
	S — 正負號狀態
	P/V — 比價／溢位狀態
	H — 輔助進位狀態
	N — 減算狀態
	下列符號使用於表的狀態欄中。
	X — 旗號受運算的影響。
	(空白)— 旗號不受運算影響。
	1 — 旗號被設定爲 1 。
	0 — 旗號被復置爲 0 。
	? — 旗號在運算後變成不確定。
	P — 旗號代表 比價之 狀態。
	O — 旗號代表溢位發生。
	I — 旗號顯示可否中斷的狀態。
〔 〕	代表括號中位址之內容。如果括號中是暫存器的名稱，它就代表暫存器的內容。如果 I／O 接頭的位址放在括號中，就代表該 I／O 接頭的資料。如果記憶器的位址括在其中，那就代表該位址處的記憶體內容。
〔〔 〕〕	間接的記憶器選址方式；雙括號中是暫存器的名稱，用該暫存器的內容做位址，再憑址取出記憶體中的資料。
∧	代表邏輯運算 AND。
∨	代表邏輯運算 OR。
∀	代表邏輯運算 XOR。
:	比較的運算。X : Y 會影響狀態旗號但不改變 X 或 Y 。
←	資料按箭頭方向傳送。
← →	在此箭頭兩方的資料相互交換位置。

(一) 資料的置入與交換

指令名稱	運算元	指令長度	狀態 C Z S P/V H N	動作說明
LD	DST,SRC	1		〔DST〕←〔SRC〕 把SRC 暫存器的內容移至DST 暫存器中。SRC 和DST 可能爲A,B,C,D,E,H或L
LD	A,IV	2	XX I 0 0	〔A〕←〔IV〕 把中斷暫存器內容移至累算器中。
LD	A,R	2	XX I 0 0	〔A〕←〔R〕 把更新暫存器內容移至累算器中。
LD	IV,A	2		〔IV〕←〔A〕 把累算器的內容移至中斷暫存器中。
LD	R,A	2		〔R〕←〔A〕 把累算器的內容移至更新暫存器中。
LD	SP,HL	1		〔SP〕←〔HL〕 把HL的內容移至SP 中。
LD	SP,IX SP,IY	2		〔SP〕←〔IX〕 或 〔SP〕←〔IY〕 把索引記憶器的內容移至SP 中。
LD	A,(ADDR)	3		〔A〕←〔ADDR〕 從指定的記憶體的位址移動資料到累算器。
LD	HL,(ADDR)	3		〔H〕←〔ADDR+1〕,〔L〕←〔ADDR〕 從指定的記憶體位址移動資料到HL。
LD	RP,(ADDR) IX,(ADDR) IY,(ADDR)	4		〔〔RP(HI)〕〕←〔ADDR+1〕,〔RP(LO)〕←〔ADDR〕 或 〔IX(HI)〕←〔ADDR+1〕,〔IX(LO)〕←〔ADDR〕 或 〔IY(HI)〕←〔ADDR+1〕,〔IY(LO)〕←〔ADDR〕 從指定的記憶器位址移動資料到暫存器偶對或是索引暫存器。
LD	(ADDR),A	3		〔ADDR〕←〔A〕 把累算器中內容存到指定的記憶器位址。
LD	(ADDR),HL	3		〔ADDR+1〕←〔H〕,〔ADDR〕←〔L〕 把HL 的內容存到指定的記憶器位址。

指令名稱	運算元	指令長度	狀態 C Z S P/V H N	動作說明
LD	(ADDR),RP (ADDR),IX (ADDR),IY	4		(ADDR+1)←(RP(HI)),(ADDR)←(RP(LO)) 或 (ADDR+1)←(IX(HI)),(ADDR)←(IX(LO)) 或 (ADDR+1)←(IY(HI)),(ADDR)←(IY(LO)) 把暫存器偶對或是索引暫存器的內容存入指定的記憶器位址。
LD	A,(BC) A,(DE)	1		(A)←((BC))或(A)←((DE)) 從暫存器偶對內容所指示的記憶體位址取出資料存入累算器。
LD	REG,(HL)	1		(REG)←((HL)) 從HL 內容所指定的記憶體位址移送資料到暫存器。
LD	(BC),A (DE),A	1		((BC))←(A)或((DE))←(A) 把累算器的內容儲存到BC 或DE 所指示的記憶體位址。
LD	(HL),REG	1		((HL))←(REG) 把暫存器的內容儲存到HL 所指示的記憶體位址。
LD	REG,(IX+DISP) REG,(IY+DISP)	3		(REG)←((IX)+DISP) 或(REG)←((IY)+DISP) 使用基底相對定位算出記憶體位址，將該處之資料移送至暫存器。
LD	(IX+DISP),REG (IY+DISP),REG	3		((IX)+DISP)←(REG) 或((IY)+DISP)←(REG) 把暫存器的內容儲存至記憶體中，位址用基底相對定址算出。
LD	REG,DATA	2		(REG)←DATA 將指令所含的數據放入暫存器。
LD	RP,DATA16	3		(RP)← DATA16 將指令所含的 16 位元 數據放入暫存器偶對
LD	IX,DATA16 IY,DATA16	4		(IX)←DATA16 或 (IY)←DATA16 將指令所含的 16 位元 數據放入索引記憶器

指令名稱	運算元	指令長度	狀態 C Z S P/V H N	動作說明
LD	(HL),DATA	2		〔〔HL〕〕←DATA 將指令所含的數據存入HL 所指示的記憶體位址中。
LD	(IX+DISP),DATA (IY+DISP),DATA	4		〔〔IX〕+DISP〕←DATA或 〔〔IY〕+DISP〕←DATA 將指令所含的數據存入相對基底定址法算出的記憶體位址。
PUSH	PR	1		〔〔SP〕−1〕←〔PR(HI)〕 〔〔SP〕−2〕←〔PR(LO)〕 〔SP〕←〔SP〕−2 把成對的暫存器內容放在 庫存的 頂上，庫存指引器 減 2 。
PUSH	IX IY	2		〔〔SP〕−1〕←〔IX(HI)〕 或 〔〔SP〕−1〕←〔IY(HI)〕 〔〔SP〕−2〕←〔IX(LO)〕 或 〔〔SP〕−2〕←〔IY(LO)〕 〔SP〕←〔SP〕−2 把索引暫存器內容放在 庫存 頂上，庫存指引器值減 2 。
POP	PR	1		〔PR(LO)〕←〔〔SP〕〕 〔PR(HI)〕←〔〔SP〕+1〕 〔SP〕←〔SP〕+2 把庫存的頂上之內容放入暫存器偶對中，庫存指引器值加 2 。
POP	IX IY	2		〔IX(LO)〕←〔〔SP〕〕或 〔IY(LO)〕←〔〔SP〕〕 〔IX(HI)〕←〔〔SP〕+1〕 或 〔IY(HI)〕←〔〔SP〕+1〕 〔SP〕←〔SP〕+2 把庫存的 頂上之內容放入索引暫存器中， 庫存指引器 加 2 。

指令名稱	運算元	指令長度	狀態 C Z S P V H N	動作說明
EX	DE,HL	1		(DE)← →(HL) 交換DE和HL的內容。
EX	AF,AF	1		(AF)← →(AF′) 交換兩套程式狀態。
EX	(SP),HL	1		(H)← →((SP)+1) (L)← →((SP)) 把HL之內容和庫存的頂上之內容交換。
EX	(SP),IX (SP),IY	2		(IX(HI))← →((SP)+1)或 (IY(HI))← →((SP)+1) (IX(LO))← →((SP))或 (IY(LO))← →((SP)) 把索引暫存器之內容和庫存的頂上之內容交換。
EXX		1		{(BC) (DE) (HL)} ← → {(BC′) (DE′) (HL′)} 交換兩套成對的暫存器內容。

(二) 集體轉移和尋找

指令名稱	運算元	指令長度	狀態 C Z S P/V H N	動作說明
LDIR		2	0 0 0	重覆動作直到〔BC〕=0； 〔〔DE〕〕←〔〔HL〕〕 〔DE〕←〔DE〕+1 〔HL〕←〔HL〕+1 〔BC〕←〔BC〕−1 把記憶器中一整塊資料搬移至記憶器中另一區域。原來資料的起始位址在HL中，目的地的起始位址在DE中，位址由低逐次增高。BC中內容代表要傳送的位元組數。
LDDR		2	0 0 0	重覆動作直到〔BC〕=0； 〔〔DE〕〕←〔〔HL〕〕 〔DE〕←〔DE〕−1 〔HL〕←〔HL〕−1 〔BC〕←〔BC〕−1 把記憶器中一塊資料搬移至記憶器中另一區域。原來資料的起始位址在HL中，目的地的起始位址在DE中，位址由高逐次降低。BC中內容代表要傳送的位元組數。
LDI		2	X 0 0	〔〔DE〕〕←〔〔HL〕〕 〔DE〕←〔DE〕+1 〔HL〕←〔HL〕+1 〔BC〕←〔BC〕−1 從HL所示的記憶器位址傳送一個位元組的資料至DE所指示的記憶器位址，把HL和DE的內容各加1，BC的內容減1。
LDD		2	X 0 0	〔〔DE〕〕←〔〔HL〕〕 〔DE〕←〔DE〕−1 〔HL〕←〔HL〕−1 〔BC〕←〔BC〕−1 從HL所指示的記憶器位址傳送一個位元組的資料至DE所指示的記憶器位址，再把HL和DE和BC的內容均減1
CPIR		2	X X X X 1	重覆動作，除非〔A〕=〔〔HL〕〕或〔BC〕=0； 〔A〕:〔〔HL〕〕 〔HL〕←〔HL〕+1 〔BC〕←〔BC〕−1 用累算器的內容和HL所指示記憶器位址的內容相比較，HL的位址逐次增大。內容相等或是BC內容減爲零時停止