IBM PC BIOS
程式剖析

林書華 著

IBM PC BIOS
程式剖析

林書華　著

儒林圖書公司　印行

IBM PC BIOS 程式剖析

著　者：林書華
發行人：楊鏡秋
出版者：儒林圖書有限公司
地　址：台北市重慶南路一段111號
電　話：3812302　3110883　3140111
郵政劃撥：0106792-1號
吉豐印刷廠有限公司承印
板橋市三民路二段正隆巷46弄7號
行政院新聞局局版台業字第1492號

中華民國七十四年七月初版
定價新台幣　270　元正

前　言

此書是針對IBM PC的BIOS 程式做細部的解析。BIOS程式佔有 8K位元組記憶體，位於實際位址FE000H到FFFFFH。BIOS程式內包括了自行測試（power on self test）和BOOT程式。這二個部分佔了大於 2K 位元組位址。其它的 6K 位元組位址放著一些有用的副程式，如RS-232C、diskette、MONITOR、printer、keyboard 等副程式，這些副程式可以由 DOS 的裝置驅動程式或使用者自己所寫的程式來使用。

本書將BIOS內的程式分段畫成類似流程圖的步驟圖，以便說明解釋。在有必要介紹 LSI 和舉例時，本書都有詳盡的解說。

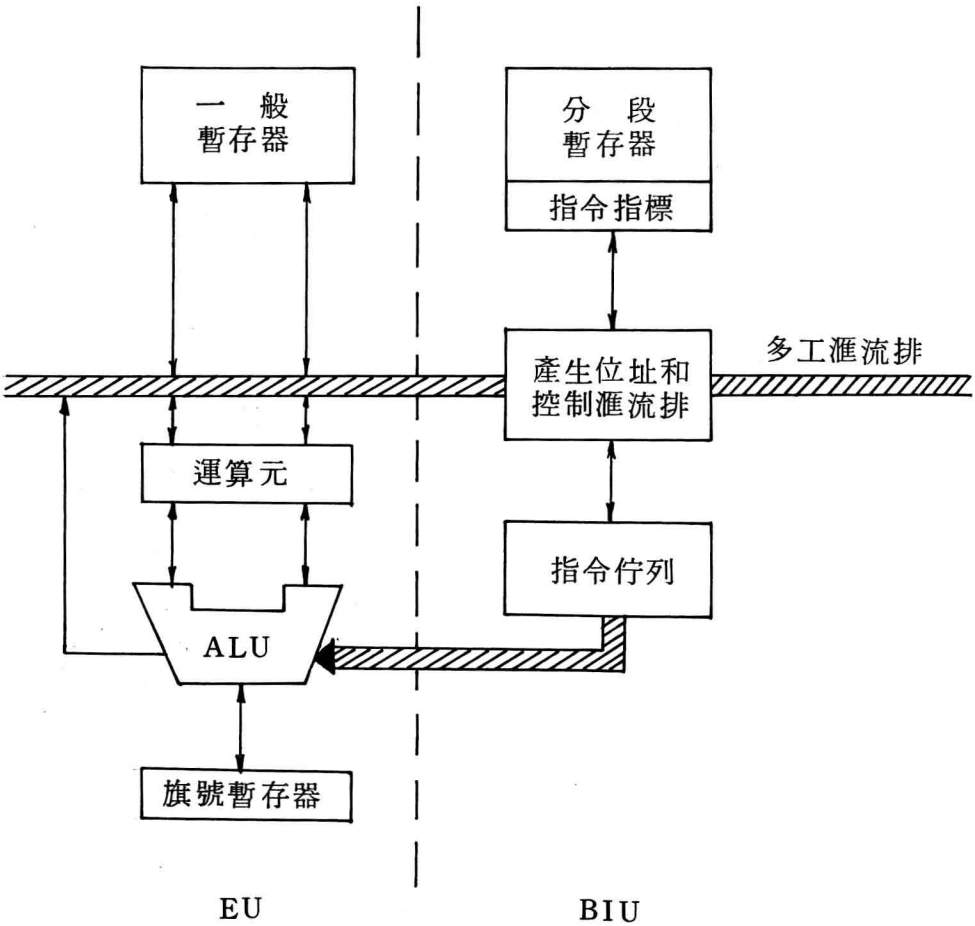
本書是解說IBM PC的BIOS程式，而非 IBM PC/XT 的BIOS 程式。PC和PC/XT的差異在於 PC/XT 可以加上硬式磁碟機，而 PC不能。所以 PC/XT的BIOS程式比PC的BIOS程式，多一段測試硬式磁碟機界面卡上的 ROM 是否存在的小程式，此小段程式只有 53 個位元組而已，而且相當簡單。若讀者將本書看完後，再來看此段程式時會覺得相當簡單。PC和 PC/XT的BIOS程式大部分是相同的，尤其是佔了 6K 位元組的副程式，幾乎一字未改。

目　　錄

第一章

8088組合語言介紹

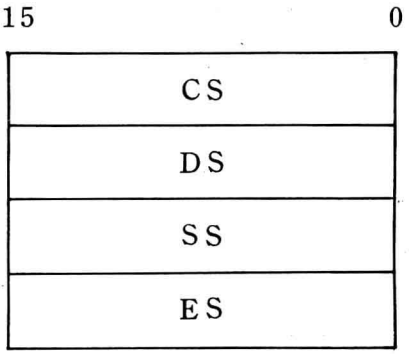
雖然8088的資料滙流排（data bus）只有8條，但其內的資料滙流排有16條，我們還是叫它16位元（bit）微處理機。因爲內在的資料滙流排有16條，所以8088的暫存器都是16位元。雖然如此，8088依然可以處理8位元的資料。我們知道8088的位址滙流排（address bus）有20條，所以它的記憶體最大容量可達一百萬個位元組（1M byte）。8088的特點是它的內部結構分爲二個單元。其一爲執行單元（Execution Unit, EU），它和外界是隔離的，它只執行指令，指令由滙流排界面單元（Bus Interface Unit, BIU）供給。另一單元爲滙流排界面單元（Bus Interface Unit），它負責外界的溝通工作，放出位址線執行取指令碼或資料的存取。它有一個可暫時存放4個指令碼的指令佇列（instruction queue），8088的滙流排界面單元（BIU）隨時將指令佇列填滿，以便供給執行單元（EU）執行，所以8088的滙流排是很“忙”的。



8088的暫存器可以分爲：分段暫存器（Segment Registers）、一般暫存器（General Registers）、指令指標（Instructions Pointer）和旗號（Flags）暫存器。分別敍述如下：

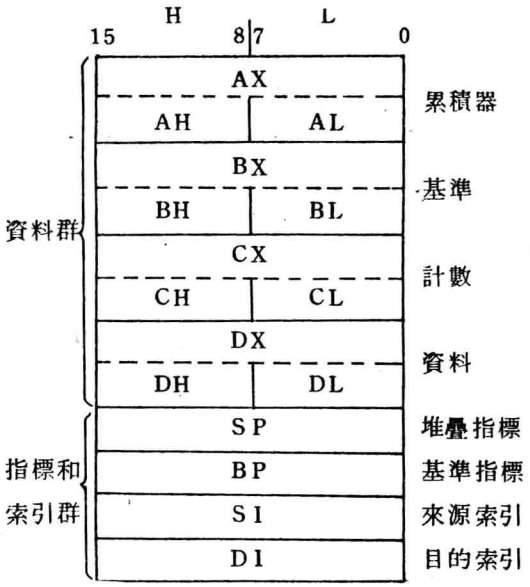
分段暫存器

分段暫存器有4個，它們都是16位元。



8088將一百萬位元組記憶體（1M byte memory）分割成16個64K位元組的段落。8088 可以同時進入其中4個段落。每個段落的基準位址（base address）存放在分段暫存器內。程式分段（code segment，CS）暫存器指著程式部分，指令碼由 CS 取來。堆疊分段（stack segment，SS）暫存器指著堆疊部分，有關堆疊的指令由 SS 處理。資料分段（data segment，DS）暫存器指著資料部分，它通常存放著程式的變數和資料。額外分段（extra segment，ES）暫存器指著額外部分，它通常用做資料的儲存。

一般暫存器



在資料群（data group）（AX，BX，CX和DX）中，每個16位元暫存器可以分爲二個8位元暫存器（AL，AH，BL，BH，CL，CH，DL和DH）。所以8088可以處理8位元的資料。指標和索引群（pointer and index group）（SP，BP，SI和DI）不能分成二個8位元暫存器，它們都是16位元暫存器。

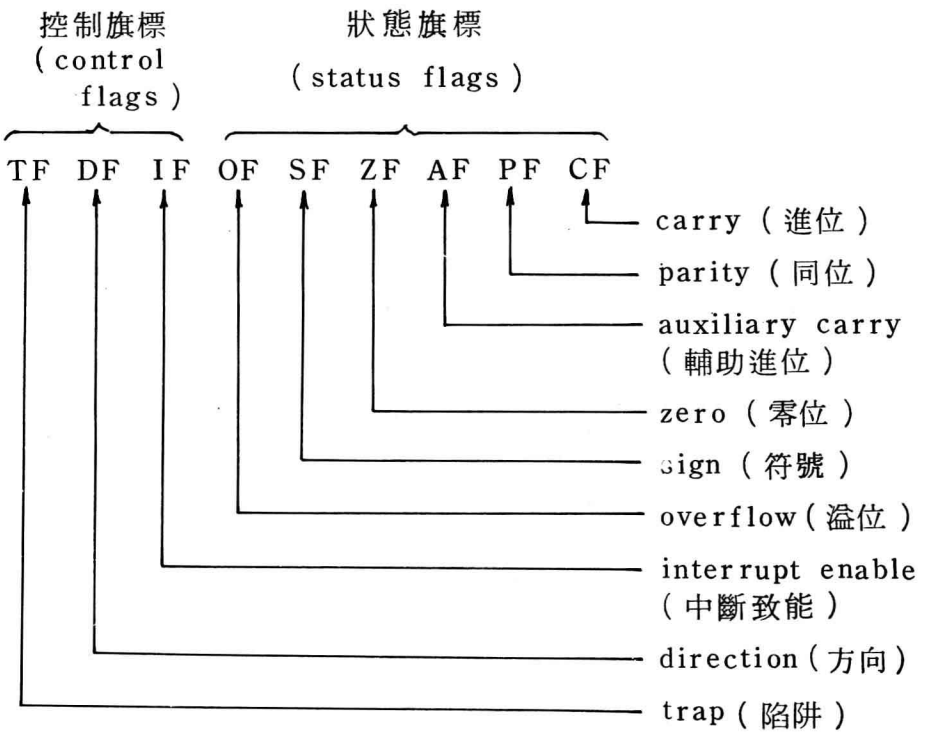
分段暫存器和一般暫存器在產生實際位址（physical address generation）、定址模式（addressing mode）和組合語言（assembly language）中將有詳細的說明。

指令指標

它是一個16位元暫存器，其中放著相對於CS的間距（offset），指著下一個要取的指令碼。組合語言可以改變指令指標的

值，例如JMP；也可以將指令指標存放到堆疊上，例如CALL；也可以從堆疊上取回指令指標，例如RET。

旗號暫存器

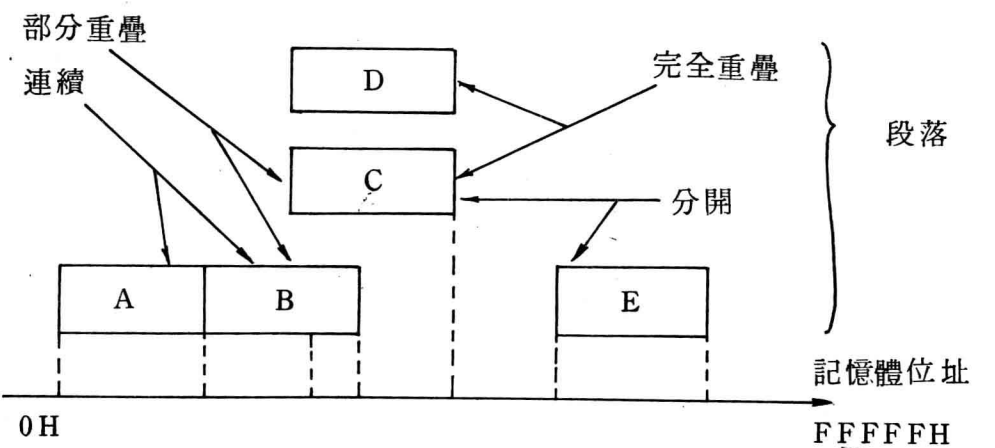


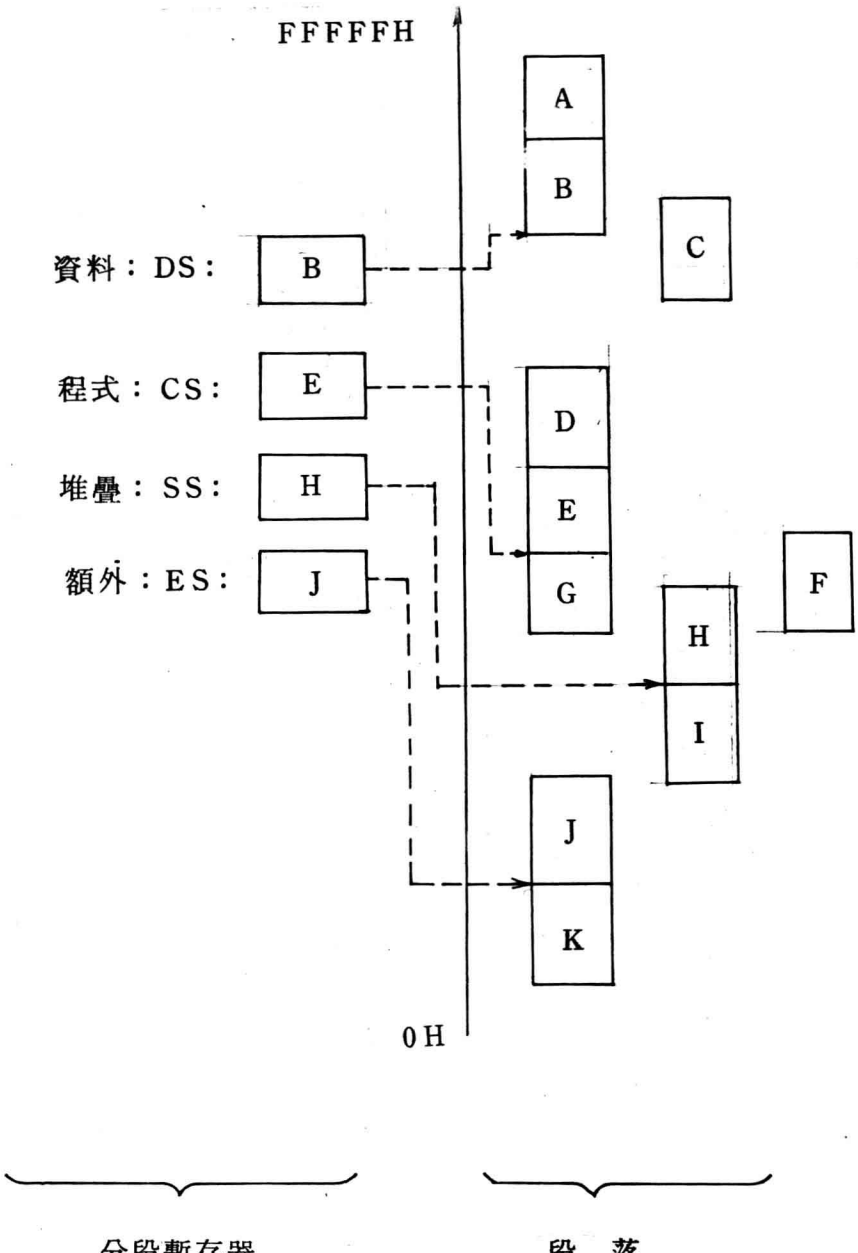
1. 如果AF爲1，就表示從低半位元組（low nibble）（位元0到位元3）有1個進位到高半位元組（high nibble）（位元4到位元7），或者是從高半位元組借位到低半位元組。AF旗號通常用在十進位制的算術中。
2. 如果CF爲1，就表示有一進位或一借位進入結果的最高位元（位元7或位元15），CF旗號通常用在加和減的算術中；旋轉指令也用到CF旗號。
3. 如果OF爲1，就表示在算術中產生溢位（overflow）現

象，這時會產生溢位中斷信號。

4. 如果SF爲1，結果的最高位元（位元7或位元15）爲1。SF旗號代表著結果的正負號（0表示正，1表示負）。
5. 如果ZF爲1，結果爲零。
6. 如果DF爲1，在字串指令（string instruction）中，索引指標（DI和SI）將會自動減少。如果DF爲0，在字串指令中，索引指標將會自動增加。
7. 如果IF爲1，就允許8088可以處理外來的可以遮罩的中斷信號。
8. 如果IF爲1，則8088就會進入單一步驟模式，方便於除錯工作。

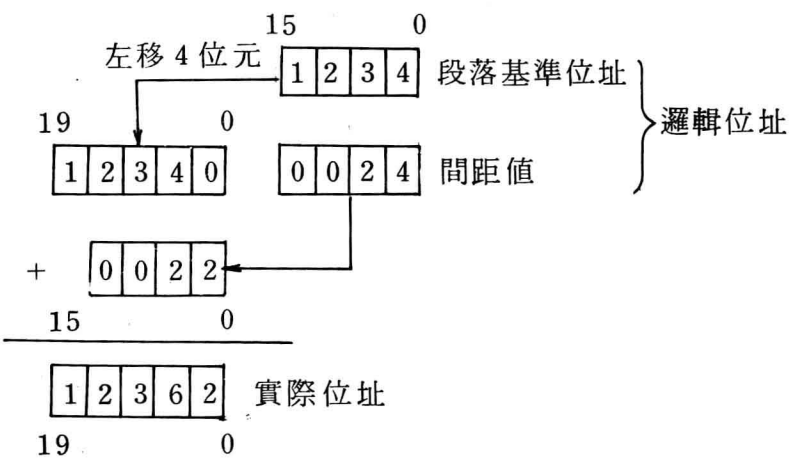
8088將一百萬位元組（1M byte）的記憶體分割成數個段落（segment），每一個段落有64K位元組容量。段落的基準位址保存在分段暫存器中（CS，ES，SS，DS），所以8088可以同時進入4個段落。段落可以相互重疊、連續或者分開，這個決定於分段暫存器內的起始位址。



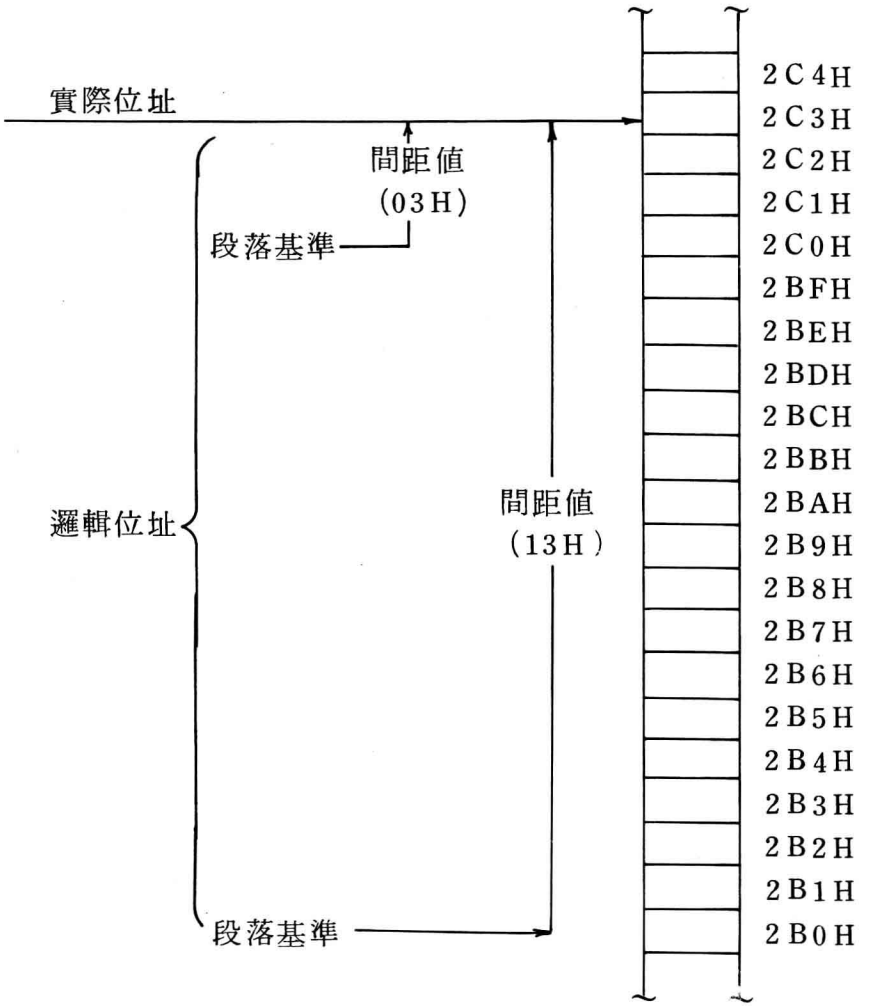

FFFFFH
A
B
C
資料：DS：
B
程式：CS：
E
D
E
堆疊：SS：
H
額外：ES：
J
G
F
H
I
J
K
0H
分段暫存器
段　落

實際位址產生（Physical Address Generation）

這裡所謂的實際位址，是爲了區分邏輯位址（logical address）之故。實際位址從零到FFFFFH，也就是8088 所能讀寫的記憶體的位址。在設計硬體時，我們注意的是實際位址，在寫組合語言時，我們就得注意邏輯位址了。邏輯位址由分段暫存器內的基準位址和間距所組成。分段暫存器內的基準位址（base address）和間距都是 16 位元。請看下面的例子，實際位址和邏輯位址的關係。



不同的邏輯位址可能指向相同的實際位址，請看下例。



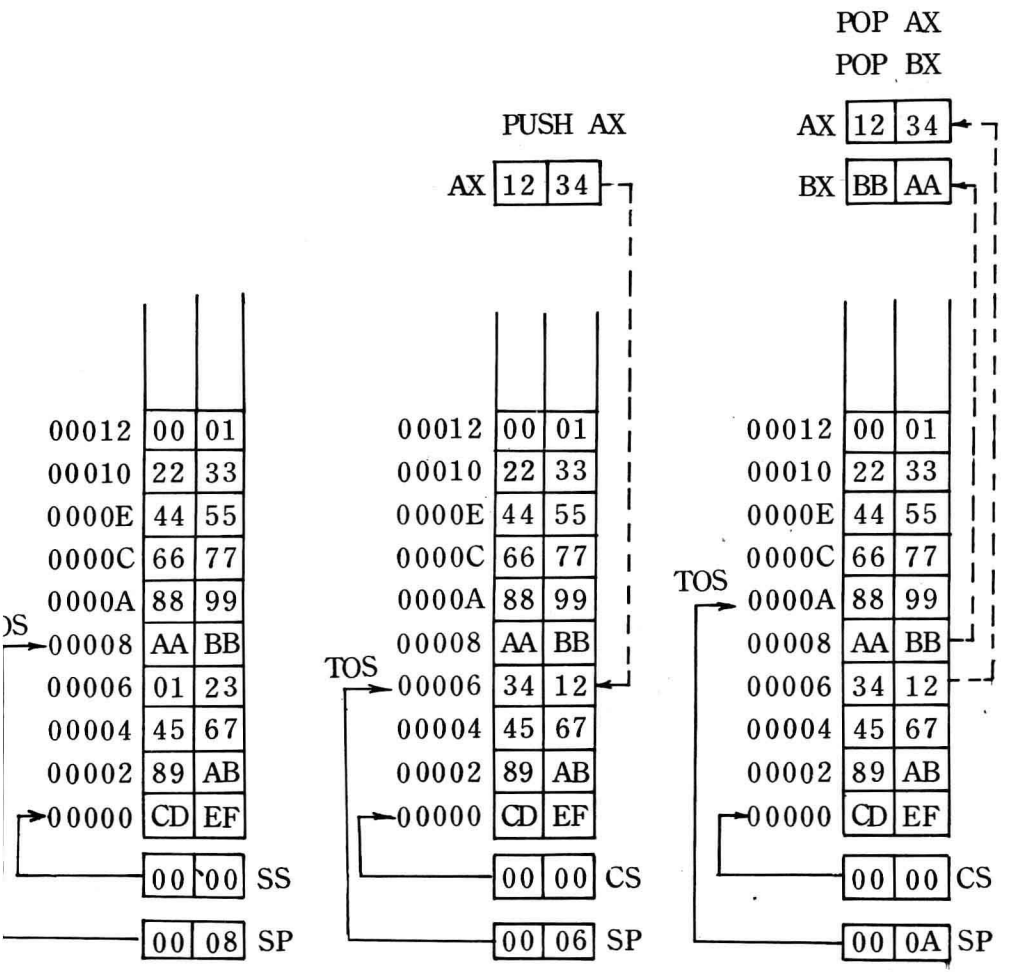
既然邏輯位址是由分段暫存器內的基準位址和一個間距值組成，那麼我們可以指令的功能來列出一個表格。這個表格很明白地告訴我們邏輯位址的組合和指令功能的關係。

指令功能	CPU 預設的基準點分段暫存器	可代替的基準點分段暫存器	間距
取指令(instruction fetch)	CS	無	IP
堆疊(stack)	SS	無	SP
變數(variable)	DS	CS,ES,SS	EA*
字串源地(string source)	DS	CS,ES,SS	SI
字串目的地(string destination)	ES	無	DI
BP 被使用做基準暫存器	SS	CS,ES,DS	EA*

* EA(Effective Address):有效位址。在定址模式中有詳細的說明。

上表中所謂的CPU預設的基準點分段暫存器(default segment base register),意思爲若程式設計者不告訴CPU 8088 需要使用其他的基準點分段,那麼CPU 8088 就以其預設的分段暫存器內的位址爲基準位址。上表中,可代替的基準點分段暫存器裡列出三個分段暫存器,表示程式設計者可以使用三個的其中一個。上表中,間距值也就是IP,SP,SI,DI 各個暫存器內的值。EA 有效位址表示一個經過定址模式演算之後得到的位址間距。

下面的例子相當重要,它使我們了解堆疊的動作。如果堆疊分段暫存器(stack segment register)放著0000H,堆疊指標放著0008H,那麼堆疊指標(TOS,TOP OF STACK)指著00008H的位址,請注意每一個PUSH或POP後堆疊指標和堆疊指標的變化(SS不會改變)。注意:PUSH和POP 都是以二個位元組爲單位。



由上面的例子，我們可以看出做 PUSH 之前，先將堆疊指標減 2，然後將暫存器 AX 內的低位元組放入低位址（ 00006H ）內，高位元組放入高位址（ 00007H ）內。 POP 指令先將低位址內的資料放入暫存器的低位元組內，高位址內的資料放入暫存器的高位元組內，然後將堆疊指標加 2 。所以堆疊指標（TOS）是指著已被佔用的位址。 PUSH 和 POP 指令也可以用在分段暫存器和記憶體位址上。這裡只是爲了解釋堆疊的動作，並不表示 PUSH 和 POP 指令只能用在一般暫存器上。 BP 暫存器也可以當做堆疊的間距。