

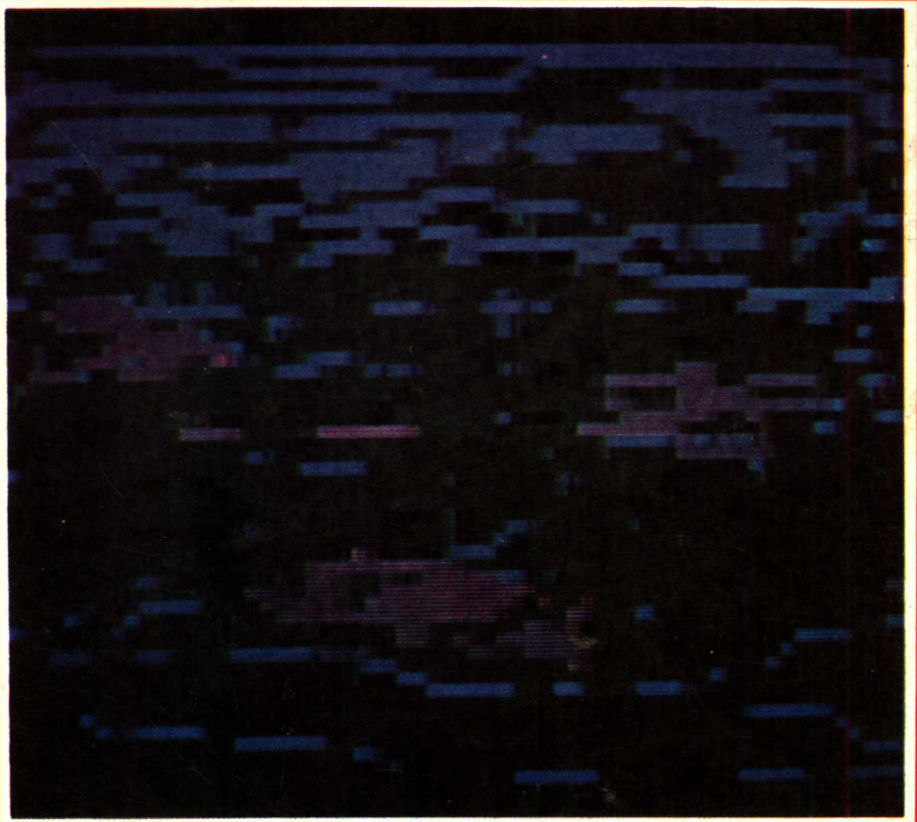
電腦叢書

電腦結構

COMPUTER ORGANIZATION

陳逢時 廖嘉成 查貴祥 合譯

HAMACHER VRANESIC ZAKY 原著



電腦語言中心出版

電 腦 結 構

COMPUTER ORGANIZATION

陳逢時
廖嘉成 合譯
查貴祥

電腦語言中心出版

電 腦 結 構

編 者：廖嘉成、陳達時、查貴祥

出版者：電 腦 語 言 中 心

發行者：

九龍彩虹道400號六樓

印刷者：合興隆印刷公司

香港仔宏利工業大廈七樓

定價港幣・\$ 22.00

目 錄

第一章 計算機的基本結構	1
1-1 計算機組成單位.....	1
1-2 計算機作業基本概念.....	9
1-3 元件間的通路結構.....	11
1-4 結 論.....	12
第二章 位址形成方式與程式執行流程	13
2-1 記憶位置，位址與資訊編碼法.....	13
2-2 記憶的作業.....	15
2-3 指令的格式與執行流程.....	16
2-3-1 指令的直線執行流程.....	17
2-3-2 躍離指令.....	19
2-4 位址形成方式.....	21
2-5 PDP-11 的指令與位址形成方式.....	23
2-5-1 位址形成方式.....	24
2-5-2 PDP-11 的指令與程式範例.....	27
2-5-3 PDP-11 的組合語言.....	29
2-5-4 程式位置記錄器所扮演的角色.....	31
2-5-5 躍離指令與狀況碼.....	33
2-5-6 運算元與邏輯指令.....	35
2-6 輸入／輸出控制程式.....	40
2-7 疊 表.....	43
2-8 副 程 式.....	46
2-8-1 副程式的銜接與參數的傳遞.....	47
2-8-2 副程式的巢狀呼叫實例.....	51
2-9 結 論.....	53

2-10 問題	54
---------	----

第三章 指令的設計 57

3-1 PDP-11 的指令組	57
3-2 大型計算機指令組	61
3-3 疊表式的計算機	64
3-3-1 HP 3000 的疊表構造	65
3-3-2 HP 3000 的指令	68
3-3-3 疊表上的記錄器	71
3-4 微算機的指令組	72
3-5 結 論	73
3-6 問 題	74

第四章 中央處理單位 75

4-1 基本觀念	75
4-1-1 記憶的讀取	77
4-1-2 記憶的回存	78
4-1-3 記憶間的傳輸	79
4-1-4 算術與邏輯的運算	80
4-1-5 記憶器的閘極控制與資料傳輸的時間控制	80
4-1-6 多通道結構	82
4-2 指令的執行	86
4-3 控制訊號的先後次序	88
4-3-1 硬體接線控制	88
4-3-2 微程式控制	93
4-4 結 論	96
4-5 問 題	96

第五章 微程式控制 99

5-1 微 指 令	99
5-2 控制信號的組合	100

5-3 微程式執行順序	104
5-3-1 跳接位址的改變	108
5-3-2 多跳接位址	108
5-3-3 詳細的例子	109
5-4 具下一個位址區域的微指令	111
5-5 微指令的先取	116
5-6 相 媲	117
5-7 結 論	118
5-8 問 題	118
第六章 輸入——輸出組織	121
6-1 I/O裝置的位址	122
6-2 資料傳送	122
6-3 同 步	126
6-3-1 點名或狀況測試	126
6-3-2 截 斷	127
6-4 截斷的處理	128
6-4-1 裝置的識別	129
6-4-2 向量化截斷	129
6-4-3 能和不能截斷	130
6-4-4 優 先 權	131
6-4-5 截 斷 罩	131
6-4-6 截斷程式的例子	133
6-5 I/O通道	134
6-5-1 通道程式	138
6-5-2 通道操作的啟動和控制	142
6-6 結 論	144
6-7 問 題	144
第七章 算術運算	149
7-7 數的表示	149

7- 2	正數的加法	150
7- 3	快速加法器的邏輯設計	153
7- 4	正數與負數的加與減法	156
7- 5	整數算數的溢位	160
7- 6	正數的乘法	160
7- 7	符號運算元的乘法	164
7- 8	快速的乘法	169
7- 9	整數的除法	174
7-10	浮點數與其運算	177
7-10-1	浮點數的算術運算	180
7-10-2	守衛數元與四捨五入	181
7-10-3	浮點數運算的完成	182
7-11	結 論	185
7-12	問 題	185

第八章 主 記 憶.....193

8-1	基本觀念	193
8-2	半導體RAM記憶	196
8-3	半導體ROM記憶	200
8-4	磁環RAM單位	201
8-5	多模組記憶與交插存取	207
8-6	貯藏記憶	208
8-6-1	映成函數	210
8-6-1	替代演算法	213
8-6-2	例 子	213
8-6-3	記憶	214
8-8	結 論	216
8-9	問 題	216

第九章 計算機週邊機和界面體.....221

9-1	I/O界面體	221
-----	--------	-----

9-1-1	資料傳輸的控制	222
9-1-2	界面體設計的例子	226
9-1-3	通路定序	230
9-1-4	標準界面體	231
9-2	I/O裝置	235
9-2-1	電傳打字機	235
9-2-2	CRT終端機	236
9-2-3	圖形的輸入裝置	241
9-2-4	讀卡機和打卡機	242
9-2-5	列表機	243
9-3	連線儲存體	244
9-3-1	磁碟系統	244
9-3-2	磁鼓系統	249
9-3-3	磁帶系統	250
9-4	結 論	252
9-5	問 題	252

第十章 軟體 256

10-1	語言和翻譯程式	256
10-2	載入程式	257
10-3	連接程式	260
10-4	作業系統	264
10-4-1	Spooling	267
10-4-2	複 程 式	268
10-4-3	複程式的作業系統控制	269
10-4-4	交談計算的作業系統	272
10-5	結 論	274
10-6	問 題	274

第十一章 微處理機 277

11-1	單境處理元件	277
------	--------	-----

11-2	微處理機的系列	278
11-3	微處理機的指令指	282
11-3-1	M 6800 指令集	282
11-3-2	Intel 8080 指令集	288
11-4	數元—薄片積體電路的處理機	295
11-5	微處理機的應用	298
11-6	討 論	299
11-7	問 題	299
第十二章 計算機通信		301
12-1	單一遠方終端機的通信	302
12-1-1	數位資料的傳送	302
12-1-2	同步和不同步傳送	304
12-1-3	雙向和單向的連接線	305
12-1-4	標準通信界面體	306
12-2	錯誤控制	309
12-2-1	傳送錯誤的偵測	309
12-2-2	傳送錯誤復原的程序	310
12-3	多終端機的結構	313
12-3-1	星狀結構	313
12-3-2	一線多點	313
12-3-3	迴路	317
12-3-4	樹狀結構	318
12-3-5	控制的層次性	322
12-4	線路轉換和消息轉換	322
12-4-1	設計網路的考慮要點	323
12-4-2	網路控制	324
12-5	結 論	325
12-6	問 題	326

附錄A	PDP-11 計算機指令組	327
A.1	前 言	327
A.2	二位運算元指令	328
A.3	一位運算元指令	329
A.4	躍離指令	331
A.5	副程式指令與躍至指令	331
A.6	中斷執行指令	332
A.7	狀況指令	333
A.8	控制指令	333
附錄B	文字編碼與數值轉換	335
B.1	文字編碼	335
B.2	十進制與二進制的數值轉換	338

第一章 計算機的基本結構

前 言

本章主要是介紹一些與計算機有關的基本知識與專門術語，僅做一些簡單的介紹，至於詳細的內容，讀者可由下面的各章節中知道其結構與應用。

首先讓我們先了解什麼叫做“數位計算機”(digital computer)或“計算機”(computer)，這常爲人所誤用，誤會的“電腦”。事實上計算機就是一個可做高速運算的電子機器，它能接受數位化的輸入資料，而以存放於它內部記憶(memory)中的程式(program)的已定義方式來處理、運算，最後產生結果，即所謂的輸出資料(output)。

1-1 計算機組成單位

“計算機”這一個名詞，事實上可分成許多種型式來看，它們可依計算機的速度、大小、能力與價格來分類，爲了詳細區分這些機種我們用下列名詞來區別它們。

“迷你計算機”(minicomputer)或稱“小型計算機”，即是較小型，造價較低，運算能力稍慢，有部分限制的計算機。到了七十年代的早期，由於大型積體線路(LSI)的迅速發展，“微型電腦”或“微算機”，“微處理機”(microcomputer, microprocessor)因硬體價格的大幅滑落，相繼以數個大型積體線路即可組成一個微算機，微處理機的特性誕生，這使得計算機又進入另一紀元。

當然大型計算機在容量、運算能力、價格，及設計的複雜度上要超出小型計算機，微算機許多，但它們基本的觀念，方法還是一樣的，以下我們就這一套有系統的設計觀念，方法按步就班的介紹給各位；我們的重點則是以一般的數位計算機(general-purpose computer)爲主，其它一些特殊用途的計算機(special-purpose computer)則不在討論範圍之內。

一個計算機的基本組成單位分成五個重要的部分；輸入(input)，記憶(memory)，運算與邏輯處理單位(arithmetic and logic unit)，輸出(output)。

2 電腦結構

output)，和控制單位 (control unit)，就如同圖 1-1 所顯示的五個相互獨立又相互傳輸資料的單位。

首先介紹輸入單位，輸入單位可經由人工或電子機械裝置把外界的資料，依已編訂好的方式，讀入計算機中，然後把這些資料放於記憶單位上。而接著運算與邏輯單位就根據已存放於記憶上的程式，一步一步地照著程式定義的方式處理這些資料，最後把結果經由輸出單位送到外界去；而以上這些動作均需要由控制單位來控制、協調。在圖 1-1 中並沒會顯示出各個單位間的連接情形，當然是有連接的必要，但因計算機架構的不同即有不同的連接方法，這些會在以後慢慢介紹給各位。

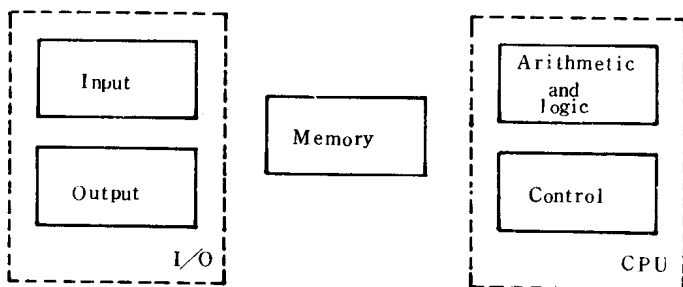


圖 1-1 計算機的基本功能單位

在習慣上我們把運算邏輯單位與控制單位合併稱為“中央處理機”或“中央處理單位”(CPU: central processing unit)，而輸入、輸出亦合併稱為輸入/輸出單位(I/O)。許多標準化的裝備均配備有輸入、輸出的雙重功能，以電傳打字終端機為例，對使用者而言就是個可供輸入輸出的終端機，但計算機在作業時仍把輸入、輸出的功能分開執行處理。

由圖 1-2 我們可看出，一部大型計算機可由許多個功能處理單位(functional unit)組成，如 IBM S 370/158，而小型計算機如同圖 1-3 所顯示的只有一張桌面的大小，如 PDP 8/M 與 PDP 11/05，為了加強小型計算機的能力，我們再加入一些裝置，甚至加入網路來連接，其大小也僅如圖 1-4 所顯示的而已，然而其功能業已可充分滿足一般的需要了。

以上我們常提到一個名詞“資訊”(有時用“輸入/輸出資料”)(information)，事實上資訊該區分成兩種：“指令”(instruction)與“資料”(data)。

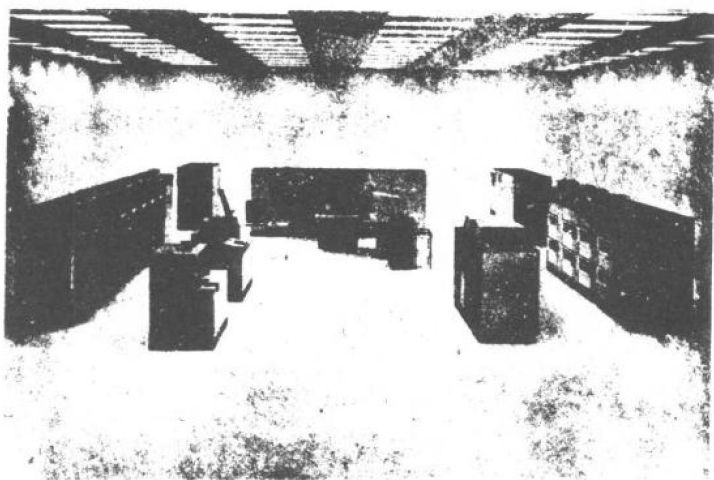


圖 1-2 大型計算機—IBM S 370/158

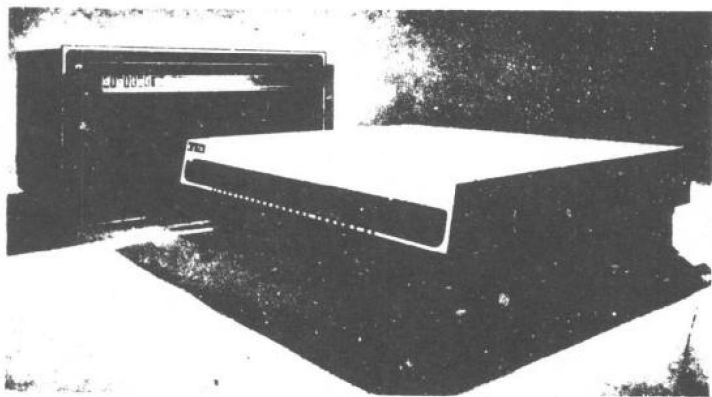


圖 1-3 迷你計算機—PDP 8與PDP 11/05

所謂的指令就是：

- 一些控制計算機資訊的傳輸情形命令，例如控制輸入輸出裝置與計算機執行的訊息通信。
- 指明運算邏輯單位該執行的步驟、情形。

由一組指令組成一個“程式”(program)，通常把這些程式存放於記憶單位上。由CPU到記憶上讀取指令，由指令既定的義意、步驟，來執行所有的運算、傳輸。通常指令的執行是一個接一個的順序執行，必要時可經由“躍離”(branch，

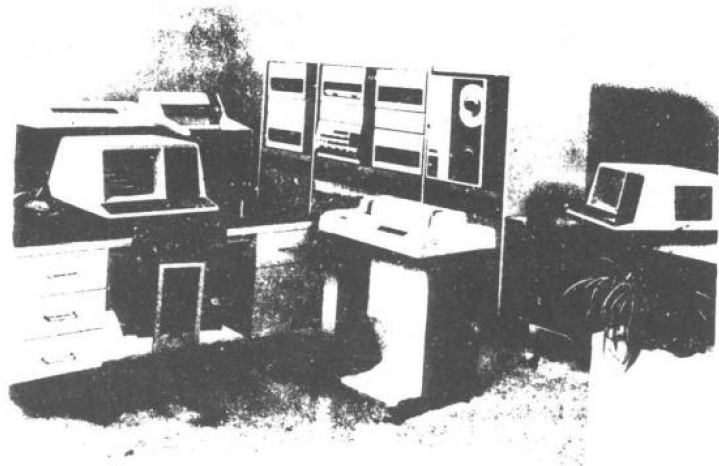


圖 1-4 迷你計算機系統 (DEC)

jump) 指令, 不按依次順序, 而直接跳到標明的新位置, 再依次執行。所以說計算機是由存放的程式來控制它的運作, 如果要中斷它的執行則需發出“中斷執行”指令 (interrupt), 這使得操作者可利用人工或由其它機器, 裝置中斷計算機的執行, 處理些必需, 較緊急的事務。

所謂的“資料”(Data) 就是一些數字 (number), 編碼完成的文字 (encoded character)。這些數字、字碼用來當作指令的運算元, 每次指令均標明它的動作是針對那些數字、字碼做運作, 而字碼又有象徵的意義, 它可能是個資料, 也可能是件裝置這都由使用者事件定義好來。我們的程式要轉換成內部的可執行型式 (internal form) 則要經過“編譯”(compilation) 的工作, 把我們使用的“高階語言”(high level language) 轉換成計算機可識別的基本指令, 這時程式事實上就可視為一筆資料, 而“編譯程式”(compiler) 就是一組做轉換工作的指令, 也就是一個程式 (program) 了。

既然計算機只能接受編碼好, 固定型式的資訊, 所以硬體線路所能表達的型式就影響計算機的作業情形了。今日一般的數位電子元件可表示兩種情形“開與關”(ON & OFF), 於是用二進位表示法來表示資訊。也就是說數字、字碼, 指令均用一串二進位的數元 (bit) 來表示, 每一數元僅有二個值, 0 或 1, 數字的表示法可參見第七章, 有詳細的介紹, 最常見的表示法就是“十進位數字二進制表示法”(BCD: binary coded decimal), 我們可把 0 到 9 用 4 位數元表示出來。

另外文字表示仍是用二進位的數元表示之，編碼的方式有許多種，如美國資訊標準交換碼(ASCII)是用7位數元，而十進位二進制交換碼(EBCDIC)則用了8位數元來表示一個文字。接著我們就五個組成單位介紹一下：

輸入單位

計算機經由輸入單位“讀入”(reading)資料，例電傳打字機把資料送入計算機中。每當打字機鍵盤上的按鍵被敲擊時，經由連好的線路就會送出相對的字碼，於是使用者的意願被轉換成一串字碼送入計算機的記憶中或CPU上，如此才可與計算機交談。

例如ASR 33電傳打字機除了打字機的功能之外，尚且加入了一個紙帶機(paper tape reader)，可做紙帶的讀入與輸出，輸入。然而電傳打字機的速度過慢，且在處理大量資料時則顯得笨拙而困難。於是我們打資料打在卡片上，每張卡片有80行，每一行代表一個文字，則經由高速的“讀卡機”(card reader)把資料，程式存入計算機中，通常一般的讀卡機每分鐘可讀1000張卡片，如圖1-5。其它的輸入裝置還有陰極射線管(CRT)配備的各種終端機(terminal)，容於第九章介紹。

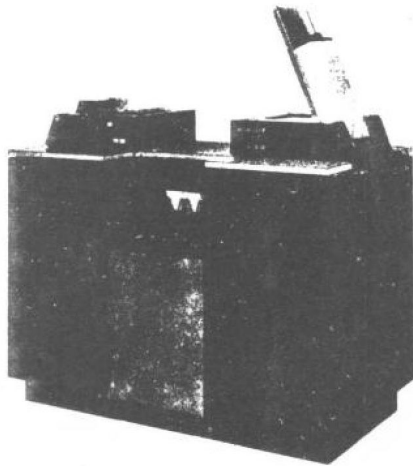


圖 1-5 讀 卡 機 (IBM)

記憶單位

記憶的功能就是用來儲存資料與程式，而記憶單位依它的容量。大小，存取速度又可分成“主記憶”(primary memory)與“輔助記憶(secondary memory)兩種。

主記憶(primary memory)可以依電子行進的速度對存於上面的資

6 電腦結構

料，程式做讀取的工作，可由磁圈（cores）或半導體（semiconductor）組成。主記憶事實上就是由許多小記憶體組成，每一個記憶體在只有一位數元，0或1，通常我們處理一組小記憶體；即是一組數元，稱為數元組（byte），或字組（word），主記憶即是由許多組數元組，或一個字組為單位，組成的記憶裝置。

而為了可明確的指出成們欲讀取記憶上的任一個記憶數元組或字組，於是我們用一些編號來標示每一記憶字數位置，這些編號稱為“位址”（address）。我們只要指出位址來，則可正確地控制、處理在相關位置上的數元組或字組資料了。而每個字組的長度（即由多少位數元組成）稱為此計算機“字組長度”（word length）。通常大型計算機字組長為32或更多位數元組成，小型計算機則在12到24之間，現在幾乎統一成16位，而微算機則為4或8位。計算機的大小與記憶容量的大小很有關係，小型機器可能只有數千個字組容量而已（ $2^{10} = 4096$ 為極小），而大型機器可到數百萬個字數。資料處理的單位通常為字組，也可能是數個字組，一個典型的記憶存取動作就是到記憶上讀取資料或重新寫入資料到記憶單位上去。

如同上面所提的，一個程式必須資料放在資記憶中。在執行過程中由CPU讀取指令與資料，控制傳輸與作業，把結果輸出的外界或寫回記憶上去，所以說希望在存取記憶的過程上要愈快愈好。而標示記憶的位址即可存取該記憶單位的記憶我們稱之“隨機存取記憶”（RAM: random access memory），存取一個單位的時間我們稱為一個週期時間（memory cycle time），通常在0.3～1百萬分之一秒之間。

然常主記憶的價格甚為昂貴，於是有了“輔助記憶”（secondary memory）來提高記憶容量，雖然速度慢了一點，但價格卻低很多，於是可把些不常用的資料放在這兒。例如：磁碟（disk），磁鼓（drums），磁帶（tapes）均可用來當做輔助記憶。第八章，第九章就主記憶，輔助記憶有更詳盡的介紹。

運算與邏輯單位（ALU: arithmetic and logic unit）

計算機的大部分作業均是在運算與邏輯單位（ALU）上完成，例做加法，我們要把加數與被加數由記憶上取出，交由ALU，由它上面的運算單位進行實際的“加”動作，而結果再存回記憶上去。同理；其它的算術或邏輯運算仍依此過程經由ALU處理。

這裏要指出的是，並不是所有的運算元都要在記憶上才可，CPU上有幾個更高速的記錄器（register）可用來暫存運算元用，一個記錄器通常可存放一個字組長的資料，但由於記錄器數目有限，故記憶上的存取時間仍是執行速度上的主要因素，通常記錄器的存取速度為主記憶的5至10倍。

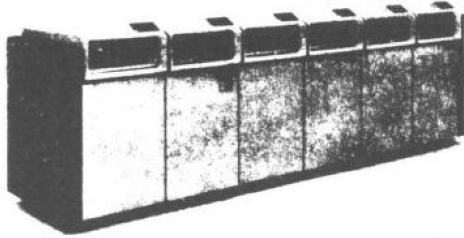


圖 1-6 磁 碟 機 (I B M)

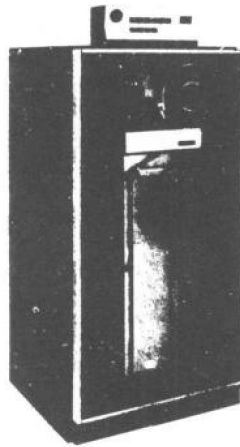


圖 1-7 磁 帶 機 (I B M)

通常ALU的速度要較連接於計算機的裝置快很多倍，於是我們可設計一個較複雜，多外界裝置的系統，只配備一個CPU，由它來配置執行的過程，以達最大效用。這些外在裝置可為磁碟、磁帶、感應裝置、機械裝置、控制器等。

輸出單位

輸出單位與輸入單位相同，它的作用在於把結果送回外界。有許多裝置把輸入輸出合併起來，如電傳打字機，終端機等。但也有單用來做輸出的裝置，如印表機（