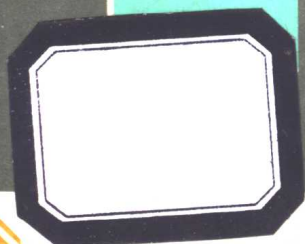
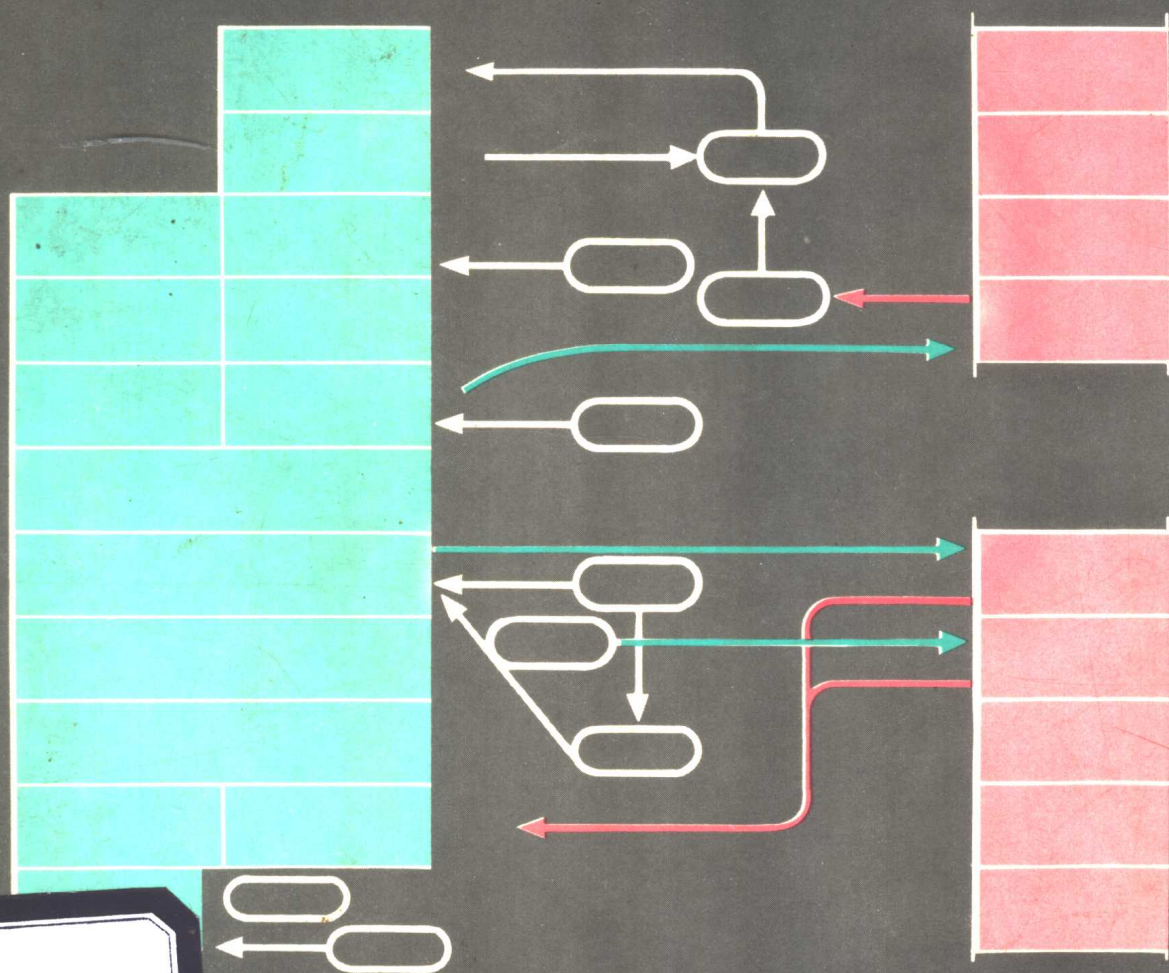


Z-80指令圖解手冊

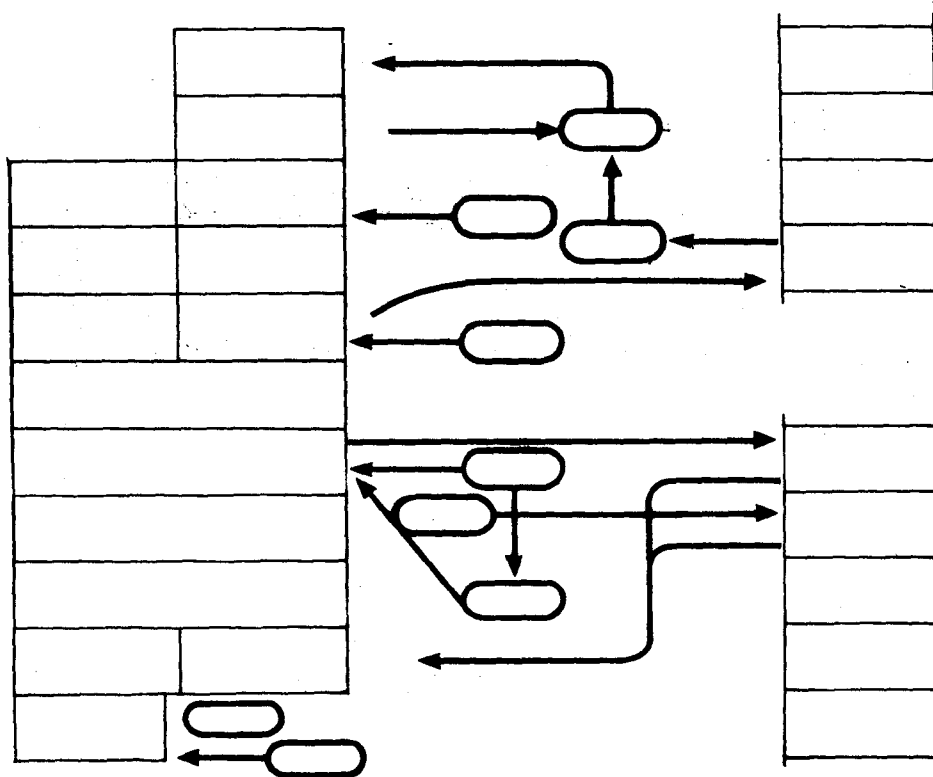
張建亨 編著



全華科技圖書股份有限公司印行

Z-80指令圖解手冊

張建亨 編著



全華科技圖書股份有限公司印行



全華圖書 版權所有 翻印必究

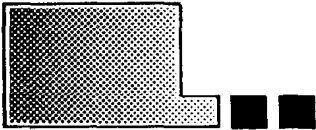
局版台業字第0223號 法律顧問：陳培豪律師

Z-80 指令圖解手冊

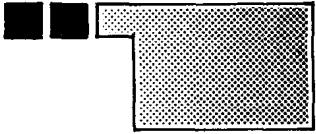
張建亨 編著

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 太一彩色印刷廠
定價 新臺幣 210 元
初版 中華民國72年9月



序 言



從 Z-80 指令表去瞭解每一個指令，只能得到既籠統又抽象的概念。坊間書籍中也只列舉了幾個代表性的指令作象徵性的說明，或列舉了記憶器讀、寫及指令擷取時序圖來概括所有指令，給人一種矇矓的印象，無法作深入的瞭解。筆者有鑑於此，遂產生了把 Z-80 指令執行過程中，CPU 與外部記憶器或 I/O 元件間運作情形以圖解方式作詳細說明的構想，因而於 71 年 3 月着手 Z-80 指令之實驗，務求每一個指令在每一個機器週期中每一個時序週期作何動作作清楚的交代，並於實際教學中作多次的修正以最簡明的圖形呈現在您的面前。

攤開本書，呈現在您眼皮子底下的右頁是 Z-80 指令圖解部份，左頁是說明部份，圖解部份包括：

- (1) 指令表：提供該指令之所有數據。
- (2) 方塊圖：繪出該指令執行過程中之來龍去脈。其中深藍色表示位址 BUS，淺藍色表示資料 BUS，灰色表示 CPU 內部邏輯。
- (3) 時序圖：執行該指令時 CPU 各有關接腳在各時序週期中之狀態。

本書各指令之說明具有獨立性，讀者大可不必從頭看起，使用本書就如同使用字典一般，把組合語言當生字，用本書去查該生字的解釋，您將會對該指令頓有所悟。

最後謹以此書獻給我那口子，她就像時序 ϕ 是使 CPU 動作的原動力。

張 建 亨 謹識

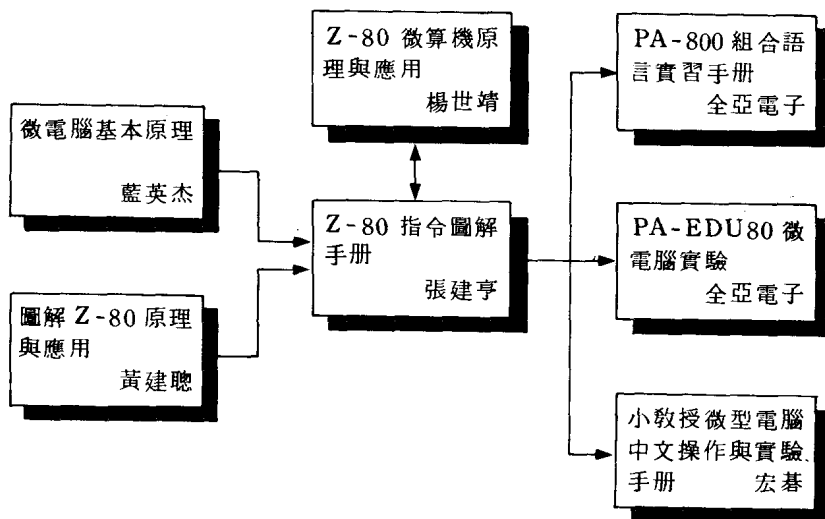
中華民國七十一年十月

編輯部序

「系統編輯」是我們的編輯方針，我們所將提供給您的，絕不只是一本書，而是關於這門學問的所有知識，它們由淺入深，且循序漸進。

現在，我們將這本「Z-80 指令圖解手冊」呈獻給您。微處理機之所以在工商業間引起革命性的震撼，祇要是它能執行各式各樣的應用功能，此功能的執行就要靠軟體程式了；其中指令便是構成軟體程式最主要的要素，因此要寫漂亮的程式及對程式變化能夠靈活的應用，就必需對指令的特性能徹底的瞭解，這些就是本書編著的最大旨意。本書的編排左頁為指令說明部份，右頁為指令圖解部份，同時以彩色圖解繪出該指令執行過程的來龍去脈及時序週期的動作，清晰、易懂、明瞭內容涵蓋 Z-80 所有指令的解說，是坊間解說指令最具特色、資料最齊全的一本書。

此外，為了使您對這門學問有更完整的了解，我們以流程圖方式，列出各有關圖書之閱讀次序，以減少您研習此門學問時的摸索時間，以及對這門學問有完整的知識。若您在這方面有任何問題，歡迎來函連繫，我們將竭誠為您服務。



指令索引

A

ADC A, n	78
ADC A, r	76
ADC A, (HL)	80
ADC A, (IX+d)	82
ADC A, (IY+d)	82
ADC HL, ss	110
ADD A, n	78
ADD A, r	76
ADD A, (HL)	80
ADD A, (IX+d)	82
ADD A, (IY+d)	82
ADD HL, ss	108
ADD IX, pp	112
ADD IY, rr	112
AND n	78
AND r	76
AND (HL)	80
AND (IX+d)	82
AND (IY+d)	82

B

BIT b, r	130
BIT b, (HL)	132
BIT b, (IX+d)	134
BIT b, (IY+d)	134

C

CALL nn	156
---------	-----

CALL NZ, nn	158
CALL Z, nn	158
CALL NC, nn	158
CALL C, nn	158
CALL PO, nn	158
CALL PE, nn	158
CALL P, nn	158
CALL M, nn	158
CCF	96
CPL	92
CP n	78
CP r	76
CP (HL)	80
CP (IX+d)	82
CP (IY+d)	82
CPD	74
CPD R	74
CPI	72
CPIR	72

D

DAA	90
DEC r	84
DEC (HL)	86
DEC (IX+d)	88
DEC (IY+d)	88
DEC IX	116
DEC IY	116
DEC ss	114
DI	104

DJNZ e154

E

EI104

EX AF, AF' 60

EX DE, HL 58

EX (SP), HL 64

EX (SP), IX 66

EX (SP), IY 66

EXX 62

H

HALT102

I

IM 0106

IM 1106

IM 2106

INC r 84

INC (HL) 86

INC (IX+d) 88

INC (IY+d) 88

INC ss114

INC IX116

INC IY116

IN A, (n)168

IN r, (c)170

IND172

INDR172

INI172

INIR172

J

JP nn142

JP NZ, nn144

JP Z, nn144

JP NC, nn144

JP C, nn144

JP PO, nn144

JP PE, nn144

JP P, nn144

JP M, nn144

JP (HL)150

JP (IX)152

JP (IY)152

JR e146

JR C, e148

JR NC, e148

JR Z, e148

JR NZ, e148

L

LDD 70

LDDR 70

LDI 68

LDIR 68

LD A, (BC) 18

LD A, (DE) 18

LD A, (nn) 20

LD A, I 26

LD A, R 26

LD I, A 28

LD R, A 28

LD (BC), A 22

LD (DE), A 22

LD (nn), A 24

LD r, r' 2

LD r, n 4

LD r, (HL)	6
LD r, (IX+d)	8
LD r, (IY+d)	8
LD (HL), r	10
LD (IX+d), r	12
LD (IY+d), r	12
LD (HL), n	14
LD (IX+d), n	16
LD (IY+d), n	16
LD dd, nn	30
LD IX, nn	32
LD IY, nn	32
LD HL, (nn)	34
LD dd, (nn)	36
LD IX, (nn)	38
LD IY, (nn)	38
LD (nn), HL	40
LD (nn), dd	42
LD (nn), IX	44
LD (nn), IY	44
LD SP, HL	46
LD SP, IX	48
LD SP, IY	48

N

NEG	94
NOP	100

O

OR n	78
OR r	76
OR (HL)	80
OR (IX+d)	82
OR (IY+d)	82

OTDR	178
OTIR	178
OUTD	178
OUTI	178
OUT (c), r	176
OUT (n), A	174

P

POP qq	54
POP IX	56
POP IY	56
PUSH qq	50
PUSH IX	52
PUSH IY	52

R

RES b, r	136
RES b, (HL)	138
RES b, (IX+d)	140
RES b, (IY+d)	140
RET	160
RET NZ	162
RET Z	162
RET NC	162
RET C	162
RET PO	162
RET PE	162
RET P	162
RET M	162
RETI	164
RETN	164
RLA	118
RLCA	118
RLC r	120

RLC (HL)	122
RLC (IX+d)	124
RLC (IY+d)	124
RLD	126
RL r	120
RL (HL)	122
RL (IX+d)	124
RL (IY+d)	124
RRA	118
RRCA	118
RRC r	120
RRC (HL)	122
RRC (IX+d)	124
RRC (IY+d)	124
RRD	128
RR r	120
RR (HL)	122
RR (IX+d)	124
RR (IY+d)	124
RST P	166

S

SBC A, n	78
SBC A, r	76
SBC A, (HL)	80
SBC A, (IX+d)	82
SBC A, (IY+d)	82
SBC HL, ss'	110
SCF	98
SET b, r	136
SET b, (HL)	138
SET b, (IX+d)	140
SET b, (IY+d)	140

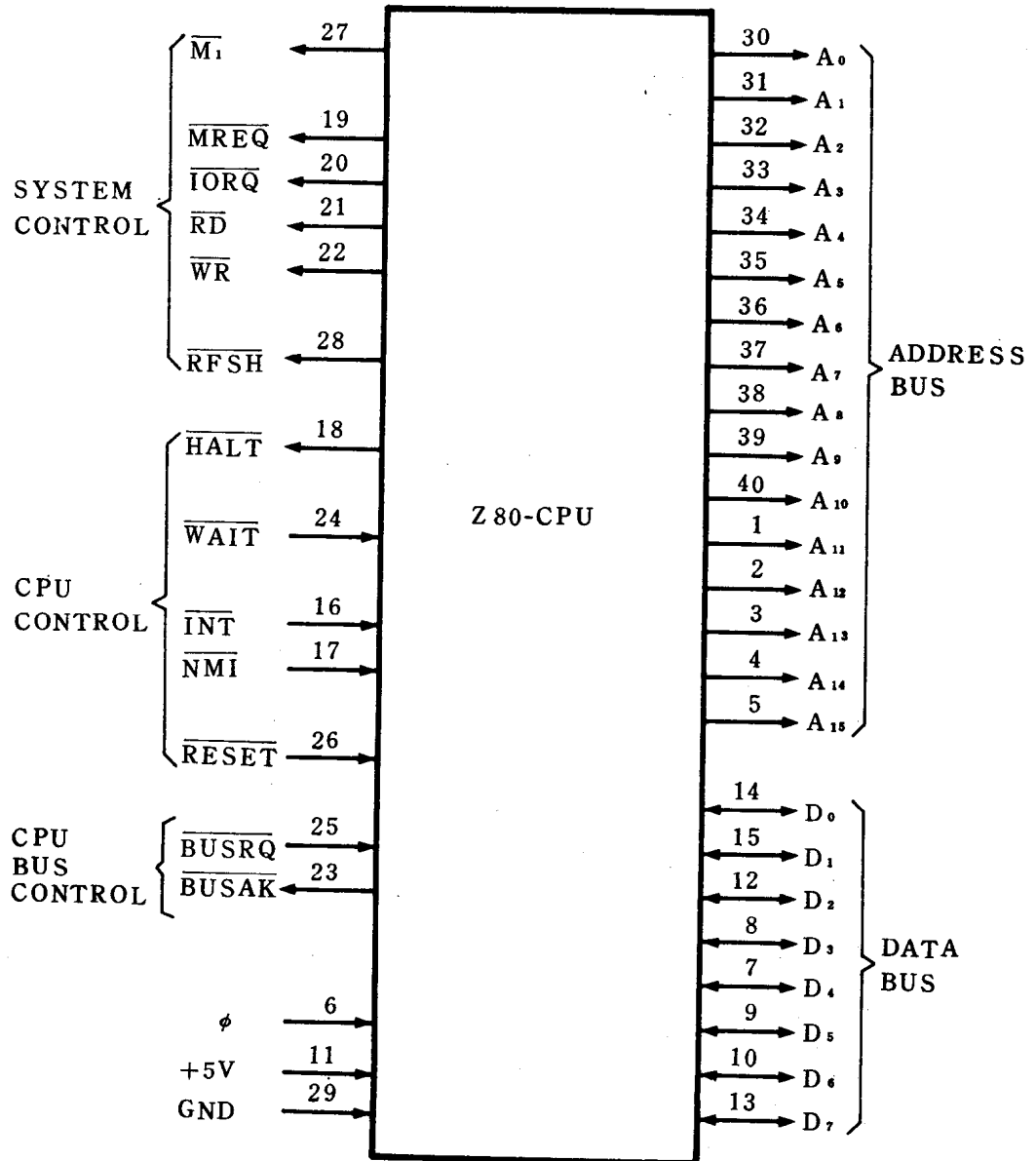
SLA r	120
SLA (HL)	122
SLA (IX+d)	124
SLA (IY+d)	124
SRA r	120
SRA (HL)	122
SRA (IX+d)	124
SRA (IY+d)	124
SRL r	120
SRL (HL)	122
SRL (IX+d)	124
SRL (IY+d)	124
SUB n	78
SUB r	76
SUB A, (HL)	80
SUB (IX+d)	82
SUB (IY+d)	82

X

XOR n	78
XOR r	76
XOR (HL)	80
XOR (IX+d)	82
XOR (IY+d)	82

附 錄

一、搜址方式	180
二、Z-80 指令 (依操作碼大小排列)	181
三、Z-80 指令 (依英文字母順序排 列)	188
四、Z-80 指令對旗號影響總表	194
五、Z-80 指令總表	196





本指令為 8 數元存放 (load) 指令，採暫存器搜址的方式。動作符號 $r \leftarrow r'$ ，表示將暫存器 r' 的內容存放到暫存器 r 中， r 與 r' 為 $A = 111$ ， $B = 000$ ， $C = 001$ ， $D = 010$ ， $E = 011$ ， $H = 100$ ， $L = 101$ ，七個暫存器中的任一個，當然 r 與 r' 為同一暫存器亦無不可。指令執行前、後狀態旗號不改變。欲將本指令改寫成機械語言，只要將操作碼中來源暫存器 r' 的三數元代號放入 0，1，2 三位元中，將目標暫存器 r 的三數元代號放入 3，4，5 三位元中即成本指令之機械語言。如 $LD A, B$ 來源暫存器 B 之代號為 000。目標暫存器 A 之代號為 111，如 $LD A, B$ 操作碼為 01111000，對十六進制而言即 78H。本指令長度為一個數元組 (8 bit)，佔程式記憶體一個位置。一個機械週期，四個時序週期，如系統使用時序頻率為 2 MHz，則時序週期為 0.5 μs ，所以本指令執行完畢須費四個時序週期，即 2 μs 時間。

對照時序圖和方塊圖，將每個機械週期中每個時序週期作何動作分別討論如下：

T_1 ： ϕ_H ：PC 內容送至位址 BUS 後 PC 內容自動加 1。 $\overline{M}_1$ 降下表示將取運算碼入指令暫存器 IR 中。

ϕ_L ： $\overline{MREQ}$ 、 $\overline{RD}$ 降下，將位址 BUS 指定之記憶體內容送至資料 BUS。

T_2 ： ϕ_H 降下時檢查 CPU 外部是否有等待要求，結果 $WAIT = 1$ 表示無等待要求。 ϕ_L 升起時將資料 BUS 內容讀入指令暫存器 IR 中。

T_3 ： ϕ_H ： $\overline{M}_1$ 、 $\overline{MREQ}$ 、 $\overline{RD}$ 升起完成讀運算碼之動作。同時將暫存器 R 內 7 數元內容送至位址 BUS 中之 $A_0 - A_6$ 後 R 內容自動加 1，且 $\overline{RFSH}$ 降下準備刷新動態記憶體內容。

ϕ_L ： $\overline{MREQ}$ 降下開始刷新位址 BUS 指定之動態記憶體內容。

在此同時 CPU 外部作動態記憶刷新，內部則將指令暫存器 IR 內容解碼以作必要之控制。

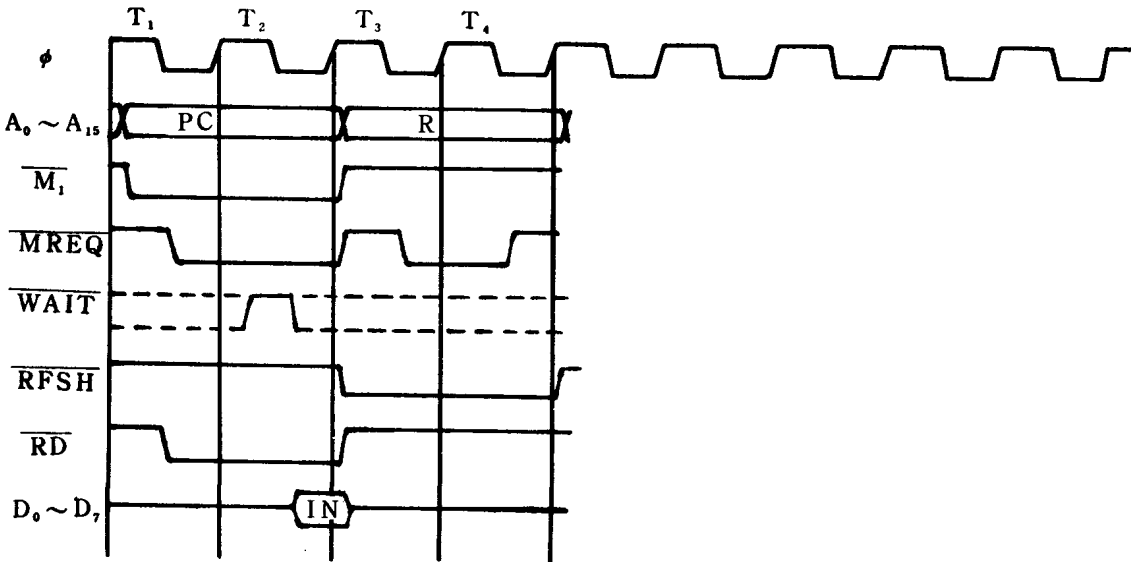
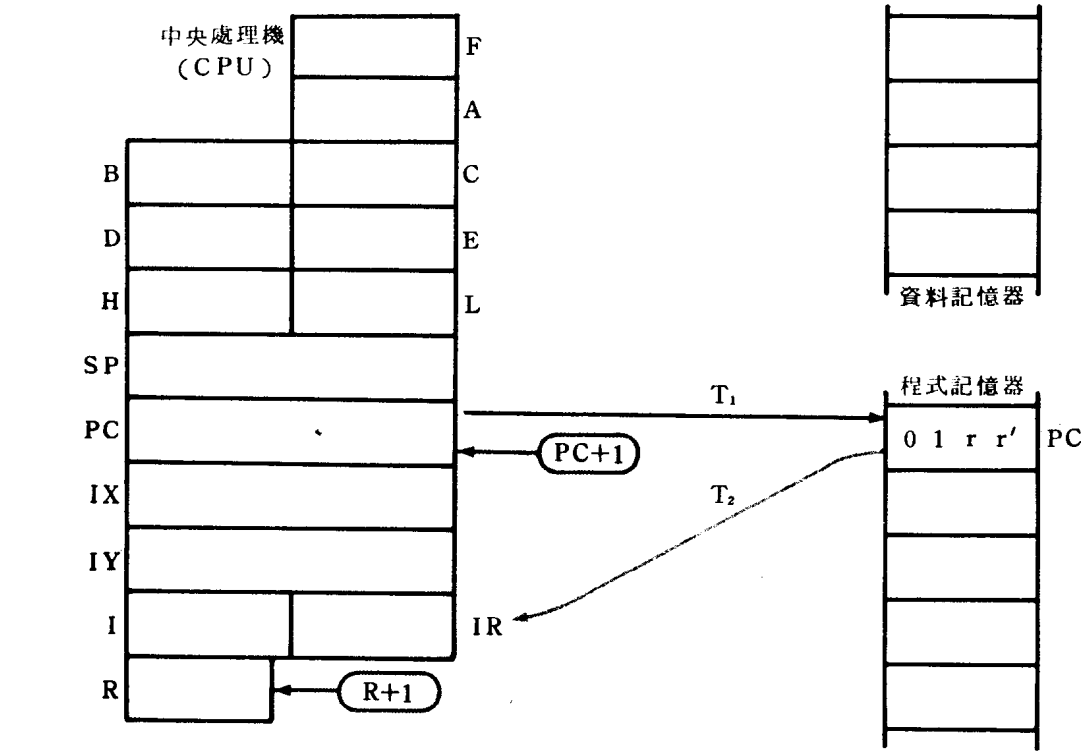
T_4 ： ϕ_L ： $\overline{MREQ}$ 升起完成刷新工作。

例： $LD H, L$

執行前 $PC = 1800H$ ($1800H$) = 65H $IR = D3H$ $R = 13H$ $H = 08H$
 $L = 66H$

執行後 $PC = 1801H$ ($1800H$) = 65H $IR = 65H$ $R = 14H$ $H = 66H$
 $L = 66H$

組合語言	動作符號	狀態旗號						操作碼								數元組 總數	機器週 期總數	時序週 期總數
		C	Z	$\frac{P}{V}$	S	N	H	7	6	5	4	3	2	1	0			
LD r, r'	$r \leftarrow r'$	.	.	.	.	.	.	0	1	r			r'			1	1	4





本指令為 8 數元存放指令，採即刻尋址的方式，動作符號 $r \leftarrow n$ ，表示將 8 數元之 n 存放入暫存器 r 中， r 為 $A = 111$ ， $B = 000$ ， $C = 001$ ， $D = 010$ ， $E = 011$ ， $H = 100$ ， $L = 101$ ，七個暫存器中的任一個。指令執行前、後狀態旗號不會改變。欲將此指令改寫成機械語言，只要將目標暫存器 r 的三數元代號放入操作碼中第一數元組的 3、4、5 位元，將 8 數元之 n 放入第二數元組即成機械語言。如 $LD\ A, AF_H$ 操作碼為 00111110、10101111，對十六進制而言即 $3E$ 、 AF 。本指令長度為二個數元組，第一個數元組為運算碼，第二個數元組為運算元，佔程式記憶體二個位置。二個機械週期、七個時序週期，如系統使用時序頻率為 2MHz ，則時序週期為 $0.5\mu\text{s}$ ，本指令執行完畢須時 $0.5\mu\text{s} \times 7 = 3.5\mu\text{s}$ 。

對照時序圖和方塊圖將每個機械週期中每個時序作何動作，分別討論如下：

M_1T_1 ： ϕ_H ：將 PC 內容送至位址 BUS 後 PC 內容自動加 1，且 $\overline{M_1}$ 降下表示將取運算碼入指令暫存器 IR 中。

ϕ_L ： $\overline{MREQ}$ 、 $\overline{RD}$ 降下，將位址 BUS 指定之記憶體內容送至資料 BUS。

T_2 ： ϕ_H ：降下時檢查 CPU 外部是否有等待要求，結果 $\overline{WAIT} = 1$ 表示無等待要求。

ϕ_L ：升起時將資料 BUS 內容讀入指令暫存器 IR 中。

T_3 ： ϕ_H ： $\overline{M_1}$ 、 $\overline{MREQ}$ 、 $\overline{RD}$ 升起完成取運算碼之動作。同時將暫存器 R 之七數元內容送至位址 BUS 之 A_0-A_6 後 R 內容自動加 1，且 $\overline{RFSH}$ 降下以控制動態記憶體。

ϕ_L ： $\overline{MREQ}$ 降下開始刷新位址 BUS 指定之動態記憶體內容。

在此同時 CPU 外部作動態記憶刷新，內部則將 IR 內容解碼以作必要之控制。

T_4 ： ϕ_L ： $\overline{MREQ}$ 升起完成刷新工作。

M_2T_1 ： ϕ_H ：PC 內容送至位址 BUS 後，PC 內容自動加 1。

ϕ_L ： $\overline{MREQ}$ 、 $\overline{RD}$ 降下，將位址 BUS 指定之記憶體內容 n 送至資料 BUS。

T_2 ： ϕ_H ：降下時檢查 CPU 外部是否有等待要求，結果 $\overline{WAIT} = 1$ 表示無等待要求。

T_3 ： ϕ_H ：降下時將資料 BUS 內容讀入暫存器 r 中。

ϕ_L ： $\overline{MREQ}$ 、 $\overline{RD}$ 升起完成 $r \leftarrow n$ 之動作。

例： $LD\ H, 35_H$

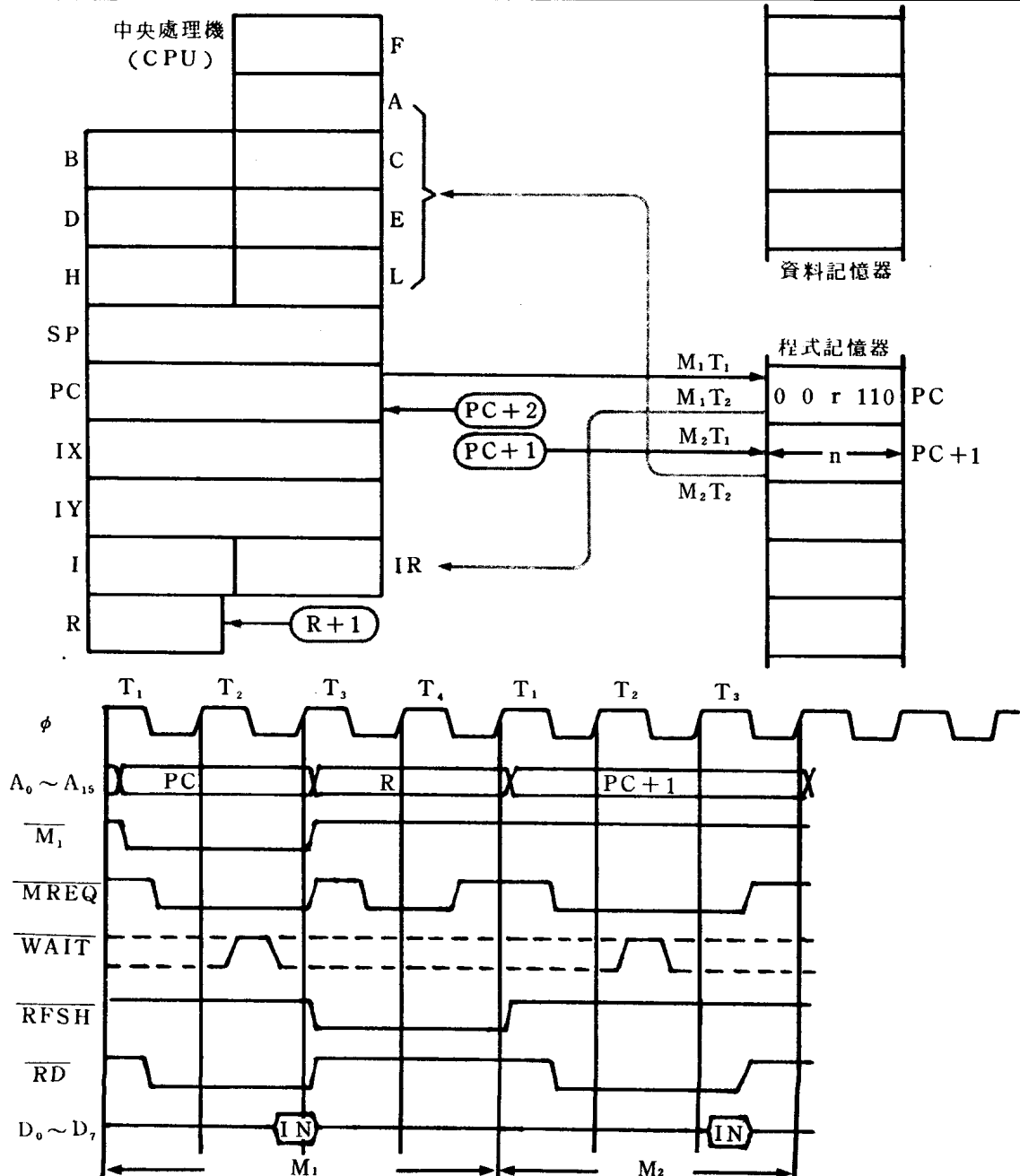
執行前 PC = 1001_H (1001_H) = 26_H (1002_H) = 35_H IR = 65_H

R = 14_H H = 66_H

執行後 PC = 1003_H (1001_H) = 26_H (1002_H) = 35_H IR = 26_H

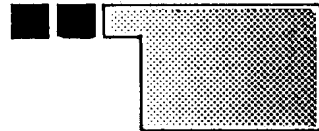
R = 15_H H = 35_H

組合語言	動作符號	狀態旗號						操作碼								數元組 總數	機器週 期總數	時序週 期總數
		C	Z	P/V	S	N	H	7	6	5	4	3	2	1	0			
LD r, n	$r \leftarrow n$	.	.	.	.	.	.	0	0	1	1	1	0			2	2	7





LD r,(HL)



本指令為 8 數元存放指令，採暫存器間接搜址的方式，動作符號 $r \leftarrow (HL)$ 表示將暫存器對 (register pair) HL 內容當位址，將此位址內之資料存放入暫存器 r 中，r 為 A = 111, B = 000, C = 001, D = 010, E = 011, H = 100, L = 101，七個暫存器中之任一個。指令執行前、後狀態旗號不會改變。欲將本指令編譯成機械語言，只須將目標暫存器 r 的三數元代號放入操作碼中的第 3, 4, 5 位元即可，如 LD B, (HL) 操作碼為 01000110，即 46H。本指令長度為一個數元組，佔程式記憶體一個位置。二個機械週期。其中第一個機器週期乃將本指令之運算碼讀入指令暫存器 IR 中，第二個機器週期乃執行 $r \leftarrow (HL)$ 之動作。七個時序週期，如系統使用時序頻率為 2MHz，則時序週期為 $0.5 \mu s$ ，所以欲完成本指令須時 $0.5 \mu s \times 7 = 3.5 \mu s$ 。

對照時序圖和方塊圖，將各機器週期中每個時序週期之動作說明如下：

M₁T₁ : ϕ_H : 將 PC 內容送至位址 BUS 後 PC 內容自動加 1，且 $\overline{M_1}$ 降下表示將取運算碼入指令暫存器 IR 中。

ϕ_L : $\overline{MREQ}$ 、 $\overline{RD}$ 降下，將位址 BUS 指定之記憶體內容送至資料 BUS。

T₂ : ϕ_H : 降下時檢查 CPU 外部是否有等待要求，結果 $\overline{WAIT} = 1$ 表示無等待要求。

ϕ_L : 升起時將資料 BUS 內容讀入指令暫存器 IR 中。

T₃ : ϕ_H : $\overline{M_1}$ 、 $\overline{MREQ}$ 、 $\overline{RD}$ 升起完成取運算碼之動作。同時將暫存器 R 之七數元內容送至位址 BUS 中之 A₀ - A₆ 後 R 內容自動加 1，且 $\overline{RFSH}$ 降下以控制動態記憶體。

ϕ_L : $\overline{MREQ}$ 降下開始刷新位址 BUS 指定之動態記憶體內容。

在此同時 CPU 外部作動態記憶刷新，內部則將 IR 內容解碼以作必要之控制。

T₄ : ϕ_L : $\overline{MREQ}$ 升起結束刷新工作。

M₂T₁ : ϕ_H : HL 內容送至位址 BUS。

ϕ_L : $\overline{MREQ}$ 、 $\overline{RD}$ 降下，將位址 BUS 指定之記憶體內容送至資料 BUS。

T₂ : ϕ_H : 降下時檢查 CPU 外部是否有等待要求，結果 $\overline{WAIT} = 1$ 無等待要求。

T₃ : ϕ_H : 降下時將資料 BUS 內容讀入暫存器 r 中。

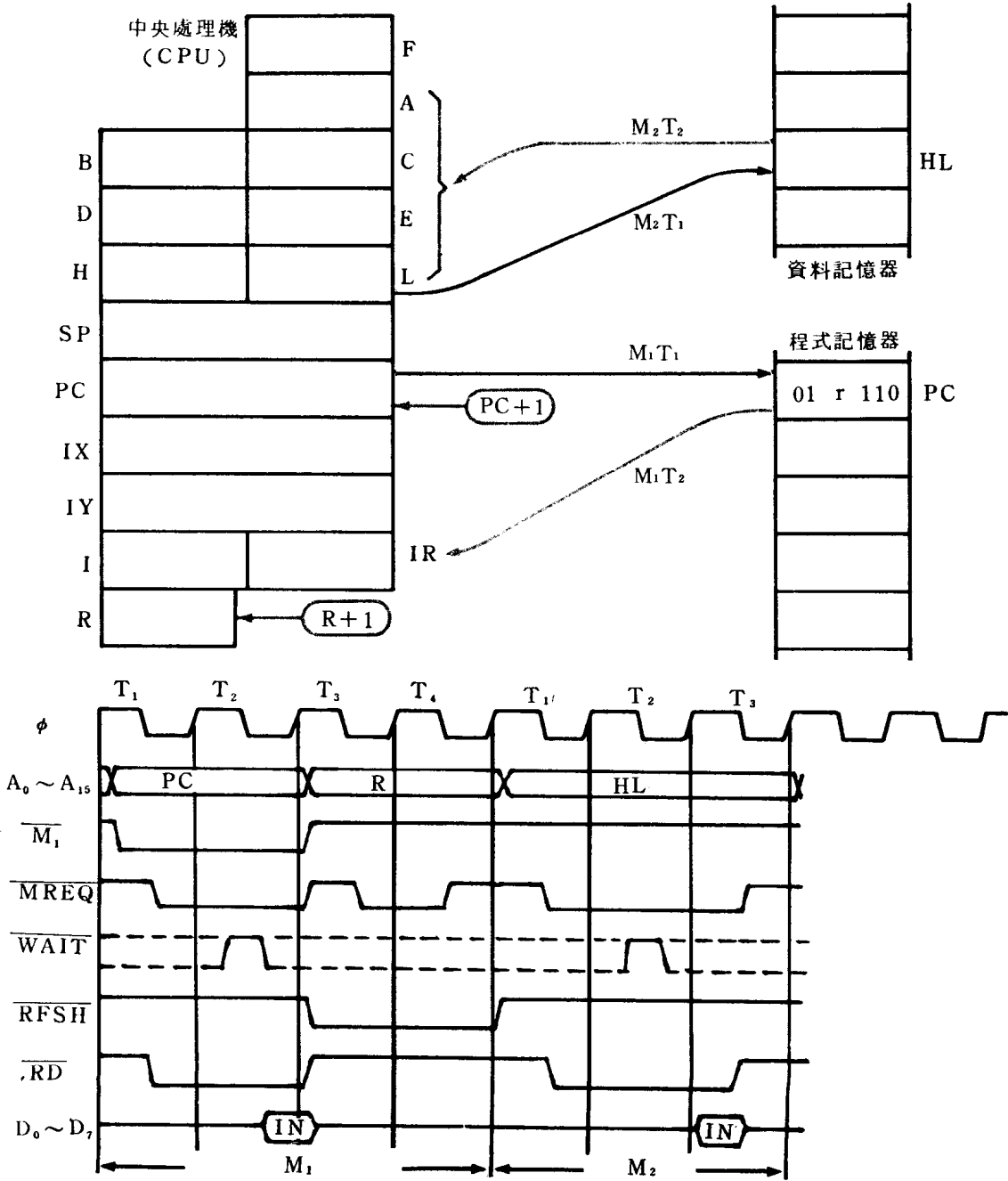
ϕ_L : $\overline{MREQ}$ 、 $\overline{RD}$ 升起完成 $r \leftarrow (HL)$ 之動作。

例：LD A, (HL)

執行前 PC = 1003H (1003H) = 7EH IR = 26H HL = 3566H

R = 15_H (3566_H) = 49_H A = 23_H
執行後 PC = 1004_H (1003_H) = 7E_H IR = 7E_H HL = 3566_H
 (3566_H) = 49_H A = 49_H

組合語言	動作符號	狀態旗號						操作碼								數元組 總數	機器週 期總數	時序週 期總數
		C	Z	$\overline{P}$ V	S	N	H	7	6	5	4	3	2	1	0			
LD r,(HL)	r ← (HL)	.	.	.	.	.	.	0	1	r			1	1	0	1	2	7



LD $r, (IX+d)$ 或 LD $r, (IY+d)$

本指令為 8 數元存放指令，採指標搜址的方式。動作符號 $r \leftarrow (IX+d)$ 或 $r \leftarrow (IY+d)$ 表示以指標暫存器 IX 或 IY 內容與位移 d 相加後的和當位址，再將該位址之內容存放至暫存器 r 中，r 為 A = 111, B = 000, C = 001, D = 010, E = 011, H = 100, L = 101 中之任一個。指令執行前後狀態旗號不會改變。欲將本指令改寫成機器語言，只要將目標暫存器 r 的三數元代號放入第二數元組中第 3, 4, 5 位元，將 8 數元之位移 d 放入第三數元組即可。其中如 d 為負，其絕對值之 2 的補數便是負值 d 二進制表示法。因 d 可正亦可負，所以 8 數元之 d 便介於 +127 (01111111) ~ -128 (10000000) 之間！本指令長度為三個數元組，佔程式記憶體三個位置。五個機器週期中 M₁ 讀取運算碼 DD_H 或 FD_H，M₂ 讀取第二運算碼，M₃ 讀取運算元 d，M₄ 在 CPU 內部作 IX+d 或 IY+d 的動作，M₅ 執行 $r \leftarrow (IX+d)$ 或 $r \leftarrow (IY+d)$ 之動作。19 個時序週期，如系統使用時序週期為 0.5 μ s，則完成本指令須時 0.5 μ s $\times$ 19 = 9.5 μ s。

對照時序圖和方塊圖，將各機器週期中每個時序週期之動作說明如下：

M₁T₁ : ϕ_H : 將 PC 內容送至位址 BUS 後，PC 內容自動加 1。 $\overline{M}_1$ 降下表示欲取運算碼入指令暫存器 IR 中。

ϕ_L : $\overline{RD}$ 、 $\overline{MREQ}$ 降下，將位址 BUS 指定之記憶體內容送至資料 BUS。

T₂ : ϕ_H : 降下時檢查 CPU 外部是否有等待要求，結果 $\overline{WAIT} = 1$ 無等待要求。

ϕ_L : 升起時將資料 BUS 內容讀入指令暫存器 IR 中。

T₃ : ϕ_H : $\overline{M}_1$ 、 $\overline{MREQ}$ 、 $\overline{RD}$ 升起結束取運算碼之動作。將暫存器 R 之七數元內容送至位址 BUS 中之 A₀-A₆ 後 R 內容自動加 1，且 $\overline{RFSH}$ 降下以控制動態記憶體。

ϕ_L : $\overline{MREQ}$ 降下開始刷新位址 BUS 指定之動態記憶體內容。

在此同時 CPU 外部作動態記憶刷新，內部則將 IR 內容解碼以作必要之控制。

T₄ : ϕ_L : $\overline{MREQ}$ 升起結束刷新工作。

M₂T₁ ~ T₄ : 與 M₁T₁ ~ T₄ 作完全相同之動作以讀取第二運算碼入指令暫存器。

M₃T₁ : ϕ_H : 將 PC 內容送至位址 BUS 後 PC 內容自動加 1。

ϕ_L : $\overline{RD}$ 、 $\overline{MREQ}$ 降下，將位址 BUS 指定之記憶體內容 d 送至資料 BUS。

T₂ : ϕ_H : 降下時檢查 CPU 外部是否有等待要求，結果 $\overline{WAIT} = 1$ 無等待要求。

T₃ : ϕ_H : 降下時將位址 BUS 指定之記憶體內容 d 讀入 CPU 內部。