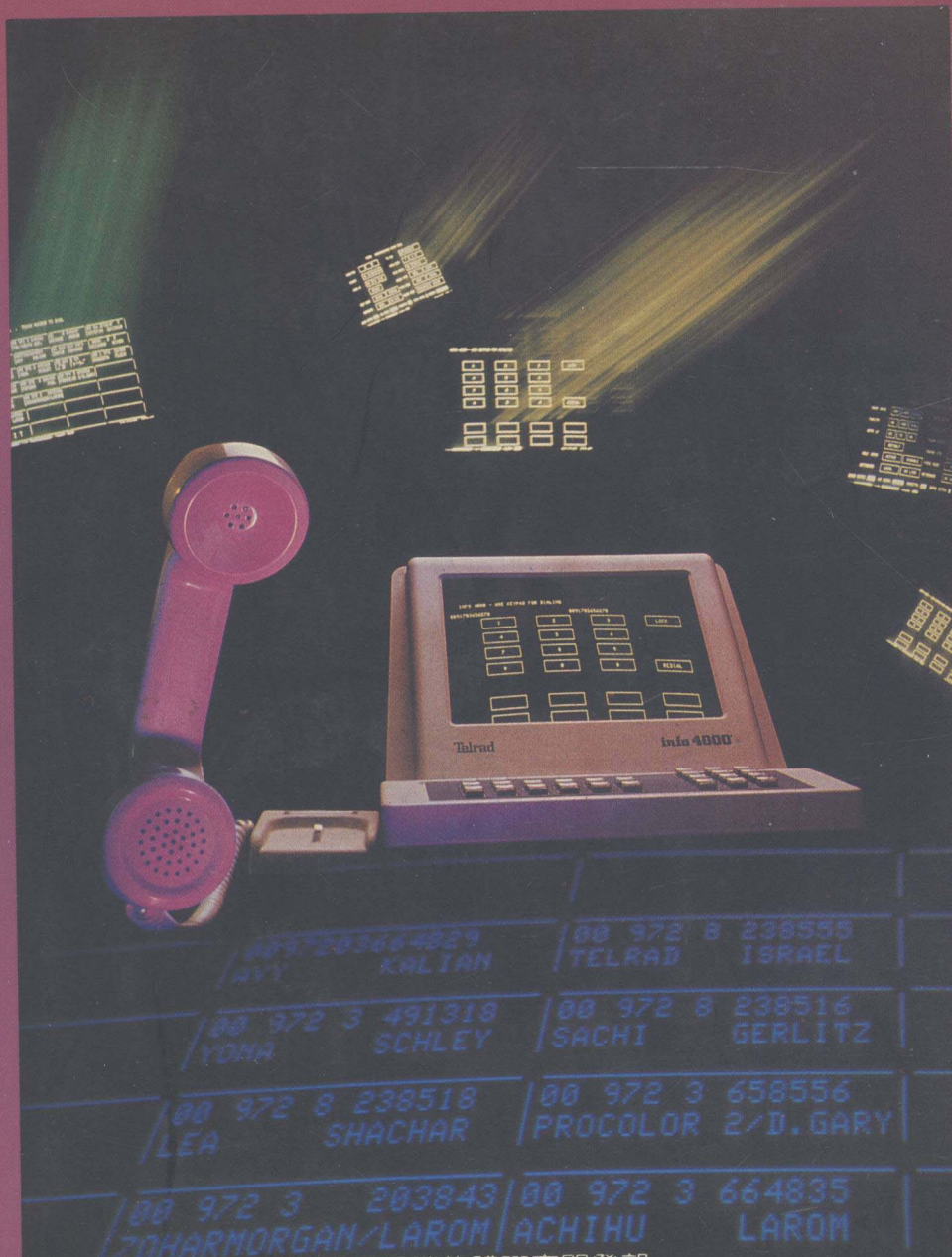
SoftTip 組合語言字典

(1990夏)

(附贈範例原始碼及PE3 & Assembler軟體)



螢圃電腦軟體研究開發部

SoftTip 尖端電腦

BIC 63060
ISBN 957-24-0078-9

製　作：瑩圃電腦出版社
發行者：周魏筱玉
發行所：中華民國台北市和平西路一段 84 號10樓（瑩圃大廈）
信件請寄：台北郵政 44-89號信箱
郵政劃撥帳號：07923480　戶名：魏筱玉
FAX：(02)705-4472

中華民國七十九年七月出版

出版登記證：局版臺業字第四二八九號

●本書之參考書籍●

ASSEMBLY LANGUAGE FOR IBM PC
ASSEMBLY LANGUAGE PRIMER FOR IBM PC & XT
BLUEBOOK OF ASSEMBLY ROUTINES FOR THE IBM PC & XT
MASM QUICK REFERENCE
TURBO ASSEMBLER REFERENCE
TURBO ASSEMBLER USER'S GUIDE

＜書號＞　63060/ ISBN 957-24-0078-9

開 數： 16 開　　頁數： 712頁

適合讀者：對 Assembler有興趣之讀者

所需軟體：Assembler 範例原始碼 <隨書附贈 PE3 & Assembler軟體>

＜本書簡介＞

任何電腦語言之精練，在於電腦指令函數之了解與應用，就像 "說寫" 英文，對單字了解與應用之重要。

這本字典不論是初學者或資深工程師均必須擁有，隨時查閱。

內容特點：

1. 指令功能分類：易於依功能別找尋所需指令。
2. 指令集：將全部指令製成一摘要說明，并提供了期型之敘述範例、定址法、執行時間。
3. ROM/ BIOS 中斷服務摘要。
4. 每一指令均含：A:功能摘要
   B:描述--詳細說明用法，特性、規定，技巧，注意事項，務使讀者充分了解。
   C:參數描述
   D:呼叫範例/ 執行時間
   E:範例--針對每一指令，至少含一個完整之範例，以配合描述說明，更易於了解應用。
5. 內含 10 個完整可執行之範例程式及對此原始程式詳細分析。
6. 本書含 PE3 & ASSEMBLER 學習版及完整可執行範例之原始碼，讀者在學習中可收到事半功倍之效果。
7. 本書若依正常字整印刷（不將程式及字體縮小），將高達 2000 頁，故讀者花 700頁之費用，購得 2000 頁之知識。

在使用書附磁片之前，請您先閱讀磁片中之 "README" 相關名稱之檔案。

＜相關書籍＞

組合語言在硬體上之應用（入門）

組合語言根基

ASSEMBLER 速查手冊

PE3 & ASSEMBLER

指令功能分類

BCD 調整

AAA AAD AAM AAS

組譯器指示元

ALIGN CONST DUP ARG DATASEG DW ARPL DB DWORD ASSUME DD ELSE CATSTR DF ELSEIF BYTE DISPLAY END CODESEG DOSSEG ENDIF COMM DP ENDM COMMENT DT ENDP ENDS EQ EQU ERR ERRIFIDN ERRIFIDNI ERRIFNB ERRIFNDEF ERRNB ERRNDEF EVEN EVENDATA EXITM EXTRN FAR FARDATA FWORD GE GLOBAL GROUP GT HIGH IDEAL IF IF1 IF2 LARGE PWORD PROC P186 P286 OR NOT NOEMUL NAME MASK LT LOW LE LENGTH MOD NE NEAR NOJUMPS NOLOCALS NOMASM51 NOMULTERRS NOWARN OFFSET P287 P386 P387 P8086 P8087 PAGE PN087 PTR PUBLIC PURGE QUIRKS QWORD RADIX XLIST WARN TBYTE SUBSTR STRUC SIZE SHORT SHL／SHR EG RE PT SEGMENT SET condition SMALL SUBTTL THIS TITLE TYPE UDATASEG UFARDATA UNION UNKNOWN USES WIDTH WORD

邏輯運算

AND XOR NOT OR

CPU 指令

BOUND WAIT STC REP condition NOP MOV LOCK ENTER DAA CMC CLI DAS ESC HLT SHL SHR STI ROL／ROR JCXZ／JECXZ LES／LFS／LGS SHLD／SHRD LEAVE Jcondition BT／BTC／BTR／BTS LGDT／LIDT／LLDT SGDT／SIDT／SLDT

副程式呼叫

CALL RET／RETN／RETF

除法

CBW CDQ IDIV DIV CWD CWDE

乘法

CLC MUL IMUL

加法

ADC ADD INC

減法

DEC SUB SBB NEG

設定方向旗標

CLD STD

比較測試

CMP TEST

字串處理

CMPS／CMPSB／CMPSW／CMPSD　SCAS／SCASB／SCASW／SCASD　REP
MOVS／MOVSB／MOVSW／MOVSD　LODS／LODSB／LODSW／　LODSD　STOS／STOSB／STOSW／STOSD

輸出入

IN　INS／INSB／INSW／INSD　OUT　OUTS／OUTSB／OUTSW／　OUTSD

中斷呼叫

INT　INTO　IRET／IRETD

跳越

JMP

旗標載入

LAHF　SAHF

位址載入

LDS／LES／LFS／LGS／LSS　LEA

迴圈

LOOP　LOOP condition

資料搬移

MOV　MOVSX　MOVZX　XCHG

堆疊存取

POP　POPA／POPAD　POPF／POPFD　PUSH　PUSHA／PUSHAD/　PUSHF／PUSHFD

移位旋轉

RCL／RCR／ROL／ROR　SAL／SAR／SHL／SHR

表格存取

XLAT／XLATB

其它

BSF／BSR　CLTS　EMUL　LAR　LMSW　LSL　LSS　LTR　MULTERRS　SECOND　SMSW
STR　SYMTYPE　VERR／VERW

指令集

本附錄是由一系列的表格所構成，這些表格將 8088 的指令集作一摘要說明，每個所列出的指令都提供了期型的組合語言之敘述範例及定址法，每個指令的執行時間，則由表中所規定的時脈數目指示，假如該行中出現 "+EA"的話，表示要多花時間來計算在記憶體中運算元的有效位址，這個時間的長短端視視用以存取此運算元的定址方式 (Addressing Mode)而定，它可在表 A-1中得到。

E A 的 成 分		時序脈衝數 *
Displacement Only		6
Base or Index Only	(BX, BP, SI, DI)	5
Displacement +		9
Base or Index	(BX, BP, SI, DI)	
Base +	BP + DI,BX + SI	7
Index	BP + SI, BX + DI	
Displacement +	BP + DI + DISP BX + SI + DISP	8
Base +	BP + SI + DISP	11
Index	BX + DI + DISP	12

*每個分節凌越 (segment override) 要多加 2個時序脈衝

代 碼	意 義
1	無條件式設置
0	無條件式清除
X	改變數值以反應運算結果
U	未定 (被遮罩掉的)
R	由記憶體所取代 (例如，SAHF)
b	(空白) 表不受影響

一.8088指令集表

AAA	AAA (no operands) ASCII adjust for addition			ODITSZAPC Flag U UUXUX
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	4	-	1	AAA

AAD	AAD (no operands) ASCII adjust for division.			ODITSZAPC Flag U XXUXU
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	60	-	2	AAD

AAM	AAM (no operands) ASCII adjust for multiply			ODITSZAPC Flag U XXUXU
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	83	-	1	AAM

AAS	AAS (no operands) ASCII adjust for subtraction			ODITSZAPC Flag U UUXUX
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	4	-	1	AAS

* 對 8086 而言，在每一個奇數位址的 16 位元之字的轉移，多加 4個時序脈

衝。對 8088 而言，每一個 16 位元的字之轉移均要多加 4個時序脈衝。

A D C	ADC destination,source ADD with carry			ODITSZAPC Flag X XXXXX
Operands	Clocks	Transfers*	Bytes	Coding Example
register,register	3	-	2	ADC AX, SI
register,memory	9+EA	1	2.4	ADC DX, BETA [SI]
memory,register	16+EA	2	2.4	ADC ALPHA [BX] [SI]. DI
register,immediate	4	-	3.4	ADC BX, 256
memory,immediate	17+EA	2	3.6	ADC GAMMA, 30H
accumulator,immediate	4	-	2.3	ADC AL, 5

A D D	ADD destination,source Addition			ODITSZAPC Flag X XXXXX
Operands	Clocks	Transfers*	Bytes	Coding Example
register,register	3	-	2	ADD CX, DX
register,memory	9+EA	1	2.4	ADD DI, [BX], ALPHA
memory.register	16+EA	2	2.4	ADD TEMP. CL
register,immediate	4	-	3.4	ADD CL, 2
memory,immediate	17+EA	2	3.6	ADD ALPHA, 2
accumulator,immediate	4	-	2.3	ADD AX, 200

A N D	AND destination,source Logical AND			ODITSZAPC Flag 0 XXUX0
Operands	Clocks	Transfers*	Bytes	Coding Example
register,register	3	-	2	AND AL, BL
register,memory	9+EA	1	2.4	AND CX, FLAG_WORD
memory,register	16+EA	2	2.4	AND ASCII [DI].AL
register,immediate	4	-	3.4	AND CX0, F0H
memory,immediate	17+EA	2	3.6	AND BETA, 01H
accumulator,immediate	4	-	2.3	AND AX, 01010000B

CALL	CALL target Call a procedure			Flag NOne
Operands	Clocks	Transfers*	Bytes	Coding Example
near.proc	19	1	3	CALL NEAR_PROC
far.proc	28	2	5	CALL FAR_PROC
memptr 16	21+EA	2	2.4	CALL PROC_TABLE [SI]
regptr 16	16	1	2	CALL AX
memptr 32	37+EA	4	2.3	CALL MEM_DWORD

* For the 8086, add four clocks for each 16-bit word transfer with an odd address.

For the 8088 four clocks for each 16-bit word transfer.

CBW	CBW (no operands) Convert byte to word			ODITSZAPC Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	2	-	1	CBW

CLC	CLC (no operands) Clear carry flag			ODITSZAPC Flag 0
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	2	-	1	CLC

CLD	CLD (no operands) Clear direction flag			ODITSZAPC Flag 0
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	2	-	1	CLD

CLI	CLI (no operands) Clear interrupt flag			ODITSZAPC Flag 0
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	2	-	1	CLI

CMC	CMC (no operands) Complement carry flag			ODITSZAPC Flag X
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	2	-	1	CMC

CMP	CMPdestination,source Compare destination to source			ODITSZAPC Flag X XXXXX
Operands	Clocks	Transfers*	Bytes	Coding Example
register,register	3	-	2	CMP BX, CX
register,memory	9+EA	1	2.4	CMP DH, ALPHA
memory.register	9+EA	1	2.4	CMP [BP+2],SI
register,immediate	4	-	3.4	CMP BL, 02H
memory,immediate	10+EA	1	3.6	CMP [BX,RADAR [DI], 3420H
accumulator,immediate	4	-	2.3	CMP AL, 00010000B

* For the 8086, add four clocks for each 16-bit word transfer with an odd address.
For the 8088 four clocks for each 16-bit word transfer.

CMPS	CMPS dest-string,source-string Compare string			ODITSZAPC Flag X XXXXX
Operands	Clocks	Transfers*	Bytes	Coding Example
dest-string, source-string (repeat) dest-string, source-string	22 9+22/rep	2 2/rep	1 1	CMPS BUFF1, BUFF2 REPE CMPS ID, KEY

CWD	CWD (no operands) Convert word to doubleword			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	5	-	1	CWD

DAA	DAA (no operands) Decimal adjust for addition			ODITSZAPC Flag X XXXXX
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	4	-	1	DAA

DAS	DAS (no operands) Decimal adjust for subtraction			ODITSZAPC Flag U XXXXX
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	4	-	1	DAS

DEC	DEC destination Decrment by 1			ODITSZAPC Flag X XXXXX
Operands	Clocks	Transfers*	Bytes	Coding Example
reg16	2	-	1	DEC AX
reg8	3	-	2	DEC AL
memory	15+EA	2	2.4	DEC ARRAY [SI]

DIV	DIV source Division,unsigned			ODITSZAPC Flag U UUUUU
Operands	Clocks	Transfers*	Bytes	Coding Example
reg8	80-90	-	2	DIV CL
reg16	144-162	-	2	DIV BX
mem8	(86-96) +EA	1	2.4	DIV ALPHA
mem16	(150-168)+EA	1	2.4	DIV TABLE [SI]

* For the 8086, add four clocks for each 16-bit word transfer with an odd address.

For the 8088 four clocks for each 16-bit word transfer.

ESC	ESC external-opcode,source Escape			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
immediate, memory immediate, register	8+EA 2	1 -	2.4 2	ESC 6,ARRAY [SI] ESC 20, AL

HLT	HLT(no operands) Halt			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	2	-	1	HLT

IDIV	IDIV source Integer division			ODITSZAPC Flag U UUUUU
Operands	Clocks	Transfers*	Bytes	Coding Example
reg8	101-112	-	2	IDIV BL
reg16	165-184	-	2	IDIV CX
mem8	(107-118)+EA	1	2.4	IDIV DIVISOR BYTE [SI]
mem16	(171-190)+EA	1	2.4	IDIV [BX]. DIVI-SOR WORD

IMUL	IMUL source Integer multiplication			ODITSZAPC Flag X UUUUX
Operands	Clocks	Transfers*	Bytes	Coding Example
reg8	80-98	-	2	IMUL CL
reg16	128-154	-	2	IMUL BX
mem8	(86-104) +EA	1	2.4	IMUL RATE_BYTE
mem16	(134-160)+EA	1	2.4	IMUL RATE_WORD[BP] [DI]

IN	IN accumulator,port Input byte or word			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
accumulator,immed8	10	1	2	IN AL, OFFEAH
accumulator, DX	8	1	1	IN AX, DX

* For the 8086, add four clocks for each 16-bit word transfer with an odd address.

For the 8088 four clocks for each 16-bit word transfer.

INC	INC destination Increment by 1			ODITSZAPC Flag X XXXXX
Operands	Clocks	Transfers*	Bytes	Coding Example
reg8	2	-	1	INC CL
reg16	3	-	2	INC BX
memory	15+EA	2	2.4	INC ALPHA [DI] [BX]

INT	INT interrupt-type interrupt			ODITSZAPC Flag 00
Operands	Clocks	Transfers*	Bytes	Coding Example
immed8 (type = 3)	52	5	1	INT 3
immed8 (type <> 3)	51	5	2	INT 67

INTR	INTR (external maskable interrupt) Interrupt if INTR and IF = 1			ODITSZAPC Flag 00
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	61	7	1	INTR

INTO	INTO (no operands) Interrupt if overflow			ODITSZAPC Flag 00
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	53 or 4	5	1	INTO

IRET	IRET (no operands) Interrupt Return			ODITSZAPC Flag RRRRRRRRR
Operands	Clocks	Transfers*	Bytes	Coding Example
(no operands)	24	3	1	IRET

JA／JNBE	JA/JNBE short-label Jump if above/Jump if not below nor equal			ODITSZAPC Flag None
Operands	Clocks	Transfers*	Bytes	Coding Example
short-label	16 or 4	-	2	JA ABOVE

* For the 8086, add four clocks for each 16-bit word transfer with an odd address.

For the 8088 four clocks for each 16-bit word transfer.

JAE／JNB	JAE/JNB short-label Jump if above or equal/Jump if not below			ODITSZAPC Flag None
Operands	Clocks	Transfers*	Bytes	Coding Example
short-label	16 or 4	-	2	JAE ABOVE_EQUAL

JB／JNAE	JB/JNAE short-label Jump if below/Jump if not above nor equal			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
short-label	16 or 4	-	2	JB BELOW

JBE／JNA	JBE/JNA short-label Jump if below or equal/Jump below nor equal			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
short-label	16 or 4	-	2	JNA NOT ABOVE

JC	JC short-label Jump if carry			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
short-label	16 or 4	-	2	JC CARRY_SET

JCXZ	JCXZ short-label Jump ifCX is zero			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
short-label	18 or 6	-	2	JCXZ COUNT_DONE

JE／JZ	JE/JZ short-label Jump if equal/Jump if zero			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
short-label	16 or 4	-	2	JZ ZERO

* For the 8086, add four clocks for each 16-bit word transfer with an odd address.

For the 8088 four clocks for each 16-bit word transfer.

JG／JNLE	JG/JNLE short-label Jump if greater/Jump if not less nor equal			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
short-label	16 or 4	-	2	JG GREATER

JGE／JNL	JGE/JNL shor-label Jump if greater of equal/ jump if not less			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
short-label	16 or 4	-	2	JGE GREATER_EQUAL

JLE／JNG	JLE/JNG short-label Jump if less or equal/Jump if not greater			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
short-label	16 or 4	-	2	JNG NOT_GREATER

JMP	JMP target Jump			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
short-label	15	-	2	JMP SHORT
near-label	15	-	3	JMP WITHIN_SEGMENT
far-label	15	-	5	JMP FAR_LABEL
memptr16	18+EA	1	2.4	JMP [BX],TARGET
regptr16	11	-	2	JMP CX
memptr32	24+EA	2	2.4	JMP OTHER_SEG [SI]

JNC	JNC short-label Jump if not carry			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
short-label	16 or 4	-	2	JNC NOT_CARRY

* For the 8086, add four clocks for each 16-bit word transfer with an odd address.

For the 8088 four clocks for each 16-bit word transfer.

JNE／JNZ	JNE/JNZ short-label Jump if equal/Jump if not zero			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
short-label	16 or 4	-	2	JNE NOT_EQUAL

JNO	JNO short-label Jump if not overflow			Flag : None
Operands	Clocks	Transfers*	Bytes	Coding Example
short-label	16 or 4	-	2	JNO NO_OVERFLOW