

APPLE

(6502CPU)型微机

常用指令速查卡

微型机的使用者在编程序和使用微机时，可能会对记忆微型机的各种指令感到困难。在大量的资料中查找这些指令，则要花费很多宝贵时间。本卡将APPLE微型机的各种指令、命令(BASIC语句，DOS命令，键盘命令，屏幕编辑，图形显示等)分类，编成一个简明的折叠卡。使用者可以迅速从卡中查出所需要的指令，从而可以充分利用编制程序的技巧，节省编程时间和查找指令的时间。

本卡是参考《APPLE II使用手册》和《DOS磁盘操作系统手册》编制的。使用者可参考这两本书来加深理解。

本卡的指令中[]的内容表示可用可不用，| |的内容表示可分别使用。

董健康 编

人民邮电出版社

目 录

算符(1)逻辑符(1)字符函数(1)数值函数(2)磁盘操作系统 *DOS* 的装入(2)系统转换图(3)磁盘命令(3)文本文件使用的 *DOS* 命令(5)盒式磁带输入输出命令(7)*DOS* 命令的参数范围(7)*DOS* 命令适用的文件类型(8)*BASIC* 命令和语句(9)高分辨率图形机器语言造型表(14)屏幕控制(15)屏幕显示的键盘编辑功能(16)程序纠错命令(17)错误信息表(17)游戏控制(19)命令使用的限制(20)整数与浮点 *BASIC* 程序的代换(20)保留字(21)

算 符

运算符

+	加号
-	减号
*	乘号
/	除号
^	乘方号
-	负号
MOD	模的余数

比较符

=	等于
<	小于
<=	小于等于
<>或><	不等于
>	大于
>=	大于等于

逻辑符

NOT 逻辑非

NOT 真 = 假 NOT 假 = 真

AND 逻辑与

真 AND 真 = 真 假 AND 假 = 假

真 AND 假 = 假 假 AND 真 = 假

OR 逻辑或

真 OR 真 = 真 假 OR 假 = 假

真 OR 假 = 真 假 OR 真 = 真

字符函数

$Y = ASC(X\$)$

字串首字转换为 ASCII 码

$X\$ = CHR\(Y)

ASCII 码转换为字符

$X = LEN(X\$)$

计字串字数

$Y\$ = LEFT\$($

取左端指定字数的字符

$(X\$, 字数)$

$Y\$ = MID\$($

取指定位置和字数的字符

$(X\$, 位置, 字数)$

$Y\$ = RIGHT\$($

取右端指定字数的字符

$(X\$, 字数)$

$X\$ = STR\(X)

数值转化为字串

$X = VAL(X\$)$ 字串转化为数值

式中: X —数值 $X\$$ —字串或字符变量

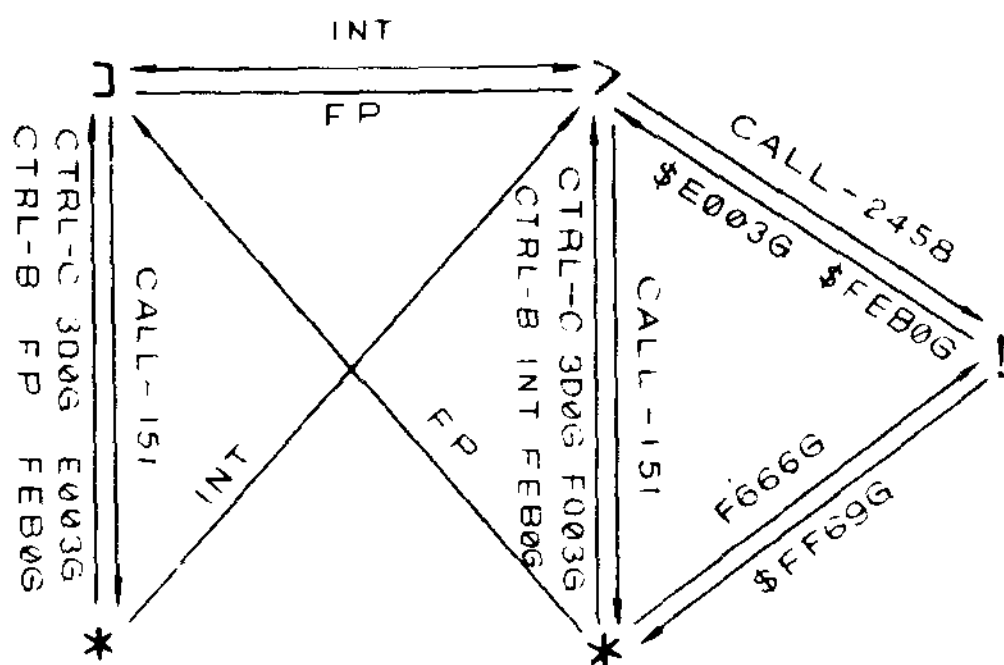
Y —ASCII 码 $Y\$$ —截取字串的字符变量

数值函数

$Y = \text{ABS}(X)$	求 X 的绝对值
$Y = \text{INT}(X)$	取 X 的整数
$Y = \text{RND}(X)$	随机函数, 随 X 及 $>$ 、 $]$ 而定
$Y = \text{SGN}(X)$	符号函数
$X < 0 \quad Y = -1$	
$X = 0 \quad Y = 0$	
$X > 0 \quad Y = 1$	
$Y = \text{SQR}(X)$	X 的平方根
$Y = \text{EXP}(X)$	e 的 X 次方
$Y = \text{LOG}(X)$	自然对数
$Y = \text{SIN}(X)$	正弦函数
$Y = \text{COS}(X)$	余弦函数
$Y = \text{TAN}(X)$	正切函数
$Y = \text{ATN}(X)$	反正切函数
X — 数值或数值表达式或数值变量	

磁盘操作系统 DOS 的装入

- > 整数 $BASIC$ 语言系统提示符
-] 浮点 $BASIC$ 语言系统提示符
- $PR \#$ 槽号 (接口槽号)
- $IN \#$ 槽号 (接口槽号)
- $PR \# 6 \quad IN \# 6$
- * 监控系统提示符
- C 槽号 $00G \quad C 600G$
- 槽号 $CTRL-P \quad 6CTRL-P$
- ! 汇编语言系统提示符



系统转换图

磁盘命令

CATALOG [*S_n*][, *D_n*]

读磁盘目录

CATALOG CATALOG *S* 6, *D* 2

磁盘文件的类型标识

A 浮点*BASIC*语言文件

B 机器语言文件

I 整数*BASIC*语言文件

T 文本文件

R 浮动文件

S *DOS*刷新文件

INIT 文件名[, *S_n*][, *D_n*][, *V_n*]

格式化磁盘

INIT HELLO

INIT FUWU, *S* 6, *D* 2, *V* 30

SAVE 文件名[, *S_n*][, *D_n*][, *V_n*]

写磁盘文件

SAVE JIHUA

SAVE GONGZI, *S* 6, *D* 1, *V* 15

LOAD 文件名[, *S_n*][, *D_n*][, *V_n*]

读磁盘文件

LOAD JILU

LOAD LUOJI, *S* 6, *D* 2

RUN 文件名[, *S_n*][*D_n*][, *V_n*]

读取并运行磁盘文件

RUN JISUAN

RUN GOIDAO, S 6, D 2

BSAVE 文件名, A_n , L_n [, S_n] [, D_n] [, V_n]

写机器语言磁盘文件

BSAVE GOIDAO, A 4096, L 256

BLOAD 文件名 [, A_n] [, S_n] [, D_n] [, V_n]

读机器语言磁盘文件

BLOAD GQUXIAN, A 4096, S 6, D 2

BRUN 文件名 [, A_n] [, S_n] [, D_n] [, V_n]

读取并运行机器语言磁盘文件

BRUN GQUXIAN

BRUN LUOJI, A 4096, S 6, D 2

LOCK 文件名 [, S_n] [, D_n] [, V_n]

加锁保护磁盘文件

LOCK GONGZI, S 6, D 2

UNLOCK 文件名 [, S_n] [, D_n] [, V_n]

开锁取消保护磁盘文件

UNLOCK GONGZI, S 6, D 2

RENAME 旧文件名, 新文件名 [, S_n] [, D_n] [, V_n]

磁盘文件换名

RENAME JILU, JIHUA

RENAME HELLO, FUWU, S 6, D 2

DELETE 文件名 [, S_n] [, D_n] [, V_n]

删除磁盘文件

DELETE JILU

DELETE QITA, S 6, D 2

VERIFY 文件名 [, S_n] [, D_n] [, V_n]

检查磁盘文件

VERIFY DATA FILE

FP [, S_n] [, D_n] [, V_n] 读取浮点 *BASIC* 语言系统

APPLE II PLUS 不用读入

FP FP, S 6, D 2

INT 读取整数 *BASIC* 语言系统

INT

CHAIN 文件名 [, S_n] [, D_n] [, V_n]

读取整数 *BASIC* 磁盘文件

CHAIN HUITU

CHAIN LUOJI, D 2

存储空间管理

HIMEM: 地址 设定程序存储区的上限

LOMEM: 地址 设定程序存储区的下限

HIMEM: 32000

LOMEM: 2000

MAXFILES 数值 预留文件缓冲区(1~6个)

MAXFILES 6

RUN[行号] 运行或从给定行号开始运行内存程序

RUN *RUN* 100

LIST[起始行号, 终止行号]

内存程序列表

LIST *LIST* 50, 150

LIST, 300 *LIST* 200,

DEL 行号[, 终止行号]

行删除

(删除单行可直接按入行号)

DEL 100, 500

NEW 清除内存程序 *NEW*

文本文件使用的DOS命令

DOS命令只能按照特定格式使用在程序中

> *D\$* = " " (引号内按入CTRL-D不显示)

PRINT *D\$*; "CATALOG"

] *D\$* = *CHR\$* (4)

PRINT *D\$*; "CATALOG"

下面的DOS命令只能用在程序里

CLOSE[文件名] 关闭顺序文件

不给出文件名则将全部在盘文本文件关闭。

> *D\$* = " "] *D\$* = *CHR\$* (4)

PRINT *D\$*; "CLOSE"

PRINT *D\$*; "CLOSE

HUITU"

APPEND 文件名[, *S_n*][, *D_n*][, *V_n*]

打开顺序文件, 并把文件指针指向末尾, 可在文件内继续写入。

> *D\$* = " "] *D\$* = *CHR\$* (4)

```

PRINT D$; " APPEND HUITU, D 2"
POSITION 文件名, Rn
文件指针前移
>D$ = " " ] D$ = CHR$(4)
PRINT D$; " POSITION JILU, R 3"

```

顺序存取的文本文件

```

OPEN 文件名[, Sn][, Dn][, Vn]
打开顺序文件
>D$ = " " ] D$ = CHR$(4)
PRINT D$; " OPEN DATA, S 6,
D 2"

WRITE 文件名[, Bn]
写入顺序文件
>D$ = " " ] D$ = CHR$(4)
PRINT D$; " WRITE LIST"

READ 文件名[, Bn]
读取顺序文件,
运行这一指令后程序内的 INPUT,
或 GET 指令就由磁盘上的顺序文件内读取信息。
>D$ = " " ] D$ = CHR$(4)
PRINT D$; " READ JILU"

```

随机存取的文本文件

```

OPEN 文件名, Ln[, Sn][, Dn][, Vn]
打开随机文件
>D$ = " " ] D$ = CHR$(4)
PRINT D$; " DPEN JIHUA, L 100,
S 6, D 2"

WRITE 文件名[, Rn][, Bn]
写入随机文件
>D$ = " " ] D$ = CHR$(4)
PRINT D$; " WRITE DATA, R 3"

READ 文件名[, Rn][Bn]
读取随机文件
>D$ = " " ] D$ = CHR$(4)
PRINT D$; " READ JILU, R 3, B 20"

```

执行型文本文件

```

EXEC 文件名[, Rn][, Sn][, Dn][, Vn]
运行执行型文本文件
EXEC BUBBA

```


盒式磁带输入输出命令

SAVE 程序写入磁带

SAVE

LOAD 读磁带中的程序

LOAD

STORE 数组写入磁带

STORE

RECALL 读磁带中的数组

RECALL

* 0.1*W* 造型表首地址·造型表末地址*W*

造型表写入磁带

写入前先用*CALL-151*进入监控, 计算造型表字节数, 并用16进制数存入内存*00:*LLHH*, 然后写入磁带。

* 00 : 0700

* 0.1*W* 6000·6006*W*

SHLOAD 读磁带中的造型表

SHLOAD

DOS 命令的参数范围 (方括号内的符号和 *n* 的取值)

	代码	参 数 内 容	最小值	最大值
全部文件 可用的参数	<i>S</i>	驱动器接口板槽号	1	7
	<i>D</i>	接口板上连接的驱动器号	1	2
	<i>V</i>	软磁盘册卷号	0	254
顺序存取 文本文件	<i>B</i>	字 节	0	32767
	<i>R</i>	相对字段数	0	32767
	<i>R</i>	绝对字段数	0	32767
随机存取 文本文件	<i>L</i>	记录长度	1	32767
	<i>R</i>	记录编号	0	32767
机器语言 文 件	<i>A</i>	起始地址	0	65535
	<i>L</i>	长度 (字节数)	1	32767

DOS 命令适用的文件类型

	浮点 BASIC 文件 (A)	机器语言 文件 (B)	整数 BASIC 文件 (I)	顺序文本 文件 (T)	随机文本 文件 (T)
LOCK	*	*	*	*	*
UNLOCK	*	*	*	*	*
DELETE	*	*	*	*	*
RENAME	*	*	*	*	*
VERIFY	*	*	*	*	*
INIT	*		*		
SAVE	*		*		
LOAD	*		*		
RUN	*		*		
CHAIN			*		
BSAVE		*			
BLOAD		*			
BRUN		*			
OPEN				*	*
WRITE				*	*
READ				*	*
CLOSE				*	*
EXEC				*	
APPEND				*	
POSITION				*	

BASIC 命令和语句

<i>IN</i> #槽号	由指定接口输入信息 <i>IN</i> # 6
<i>PR</i> #槽号	向指定接口输出信息 <i>PR</i> # 1 <i>PR</i> # 0
<i>AUTO</i> 起始行号 [, 增量]	整数 <i>BASIC</i> 自动编定行号 <i>AUTO</i> 100, 10
<i>MAN</i>	停止自动编定行号 <i>MAN</i>
<i>FRE</i> (<i>x</i>)	计算内存剩余字节 <i>PRINT FRE</i> (0)
<i>FRE</i> (0)	清除内存无用内容 <i>FRE</i> (0) <i>PRINT FRE</i> (0)
<i>CALL</i> 十进制地址	转机器语言子程序 <i>CALL</i> -936
<i>PEEK</i> (十进制地址)	读地址单元的内容 <i>PEEK</i> (36)
<i>POKE</i> 地址, 数值	将数值写入地址单元 <i>POKE</i> 800, 5
<i>WAIT</i> 地址, 数值 1 [数值 2]	等待输入比较 <i>WAIT</i> - 16286, 128
<i>USR</i> (数值)	转入机器语言子程序并读累加器的值。 <i>PRINT USR</i> (8)
<i>STOP</i>	结束程序运行, 回到命令级。 <i>STOP</i>
<i>CONT</i>	停止中断恢复运行 <i>CONT</i>
变量	数字或字符变量, 以字母开头有规定的字符串。] >
实数变量	名只认 2 字 只能存整数 可存 $\pm 1.7 \times 10^{+6}$
整数变量	名以 % 结尾 名可达 100 字

	可存 ± 32767	可存 ± 32767
数组变量	名后有下标 一维到多维	只能用变量名 或一维
字符变量	名 ≤ 238 字 可有下标 均以 \$ 结尾,	名 ≤ 100 字 无下标 均可存 < 256 字

[LET] 变量 = 表达式 赋值语句 = 赋值号
 $LET \quad X = X1 + X2$
 $X = X1 + X2 \quad T\$ = "ABC"$

REM 文字 注释语句
 程序中的说明, 不执行。
 $REM \quad APPLESOFT \quad STORED$

INPUT [" 提示" ;] 变量 ; 变量 ……
 输入语句
 $> INPUT \quad "X = ", A$
 $] INPUT \quad "X = "; A$

PRINT [变量或表达式] [;] 变量或表达式 ……
 显示语句
 分号为紧凑格式; 逗号为自动格式显示。
 $PRINT \quad 3 * 6 + 5, \quad 3 + 5 * 6$
 $PRINT \quad B\$; CT\$$

GET 变量 接受键盘输入, 不显示。
 $GET \quad SB\$$

IF 逻辑表达式 THEN
 条件语句
 $IF \quad X > 0 \quad THEN \quad X = X + 1 : Y = Y + 1$
 $IF \quad B = 0 \quad THEN \quad GOTO \quad 300$
 $IF \quad E < 3 \quad THEN \quad 500$

FOR 循环变量 = 起始值 TO 终止值 STEP 步长值
 循环体 **NEXT 循环变量**
 循环语句
 $FOR \quad I = 1 \quad TO \quad 5 \quad PRINT \quad A\$ \quad NEXT \quad I$

DIM 数组变量说明
 $> DIM \quad 变量(n_1), 变量(n_2) \dots$
 说明变量的字符个数为 $n_i + 1$ 个字
 $DIM \quad S1\$ (5), S2\$ (10), NB (12)$
 $] DIM \quad 变量(数值i[, 数值j] \dots)$
 说明数组的维数和各维的变量个数
 $DIM \quad R\$ (10, 20), R\% (5, 10, 10)$

CLR
CLEAR } 全体变量清零

>CLR *CLEAR*

DEF FN 自定义的函数名 = 表达式

自定义函数

DEF FN CS(X)=1-SIN(X)

FN 自定义函数名 使用自定义函数

FN CS(4) *FN CS(10)*

数值[, 数值……]

DATA [" 字符" [, 字符……]

置字语句

DATA 5, 7, 3.8, 3.1416, 20, 24

DATA "4BC", "DEFG", "IJKL"

READ 变量[, 变量……]

读字语句

READ A, C1, V *READ* B\$, T\$, C\$

RESTORE 恢复置字起始顺序

RESTORE

GOTO { 行号表达式 } 程序转移运行
 行号

>GOTO X*100

GOTO 500

GOSUB 行号表达式 转子程序运行
 行号

>GOSUB Y*200

GOSUB 2000

RETURN 子程序结束返回

RETURN

POP 子程序不返回

POP

ON 变量 *GOTO* 行号 1, 行号 2 ……

变量选择转移

ON E4% *GOTO* 100, 200, 150, 600

ON 变量 *GOSUB* 行号 1, 行号 2 ……

变量选择转子程序

ON BX *GOSUB* 1000, 2000, 3500

END 程序结束语句

END

文本显示指令

<i>HOME</i>	清除文本屏幕 与 <i>CALL-936</i> 作用相同 <i>HOME</i>
<i>INVERSE</i>	反相显示 <i>INVERSE</i>
<i>FLASH</i>	闪动显示 <i>FLASH</i>
<i>NORMAL</i>	正常显示 <i>NORMAL</i>
<i>SPEED=数值</i>	控制显示速度 <i>SPEED=0 SPEED=255</i>
<i>PRINT</i> 表达式 ₁ ; 表达式 ₂	， 为留空格式——两表达式首字间>为8字，]为16字 ： 为连接格式——两表达式首尾相接 <i>PRINT 5, 10 PRINT 5; 10</i>
<i>SPC</i> (数值)	当前光标位移 <i>PRINT SPC(15); "ABCDE"</i>
<i>TAB</i> (数值) <i>TAB</i> 数值	相对当前行首的显示位置 <i>]PRINT TAB(10); A\$;</i> <i>TAB(20); 4</i> <i>>TAB 10:PRINT A\$;</i> <i>TAB 20:PRINT A</i>
<i>HTAB</i> 列标	显示的水平位置 <i>HTAB 30 HTAB 5</i>
<i>VTAB</i> 行标	显示的垂直位置 <i>VTAB 20 VT 4B 3</i>
<i>POS</i> (0)	当前光标所在的列 与 <i>PEEK(36)</i> 作用相同 <i>POS(0)</i>
<i>PEEK</i> (37)	当前光标所在的行 <i>PEEK(37)</i>
<i>TEXT</i>	图形转入全文本显示 <i>TEXT</i>

图形显示命令

GR 低分辨率图形文本混合显示

图形40*40点文字4行

GR

COLOR=色码 指定显示点的颜色

COLOR=9 **COLOR=1**

COLOR=14

色码表

0 黑	4 深绿	8 棕	12 绿
1 紫红	5 灰	9 桔红	13 黄
2 深蓝	6 蓝	10 灰	14 水色
3 紫	7 浅蓝	11 桃红	15 白

PLOT 横坐标 纵坐标

绘出一个点

PLOT 20, 16 **POLOT** Y/2+10, X-3

HLIN 起点横坐标, 终点横坐标 **AT** 行标

在指定行上划水平线

HLIN 0, 35 **AT** 6

VLIN 起点纵坐标, 终点纵坐标 **AT** 列标

在指定列上划垂直线

VLIN 10, 30 **AT** 32

HGR 高分辨率图形文本混合显示

图形280*160点文字4行

HGR

HGR2 高分辨率图形第二页

全屏幕图形280*192点

HGR2

HCOLOR=色码 指定高分辨率图形显示点颜色

HCOLOR=3 **HCOLOR=5**

色码表

0 黑	2 紫	4 黑	6 蓝
1 绿	3 白	5 桔红	7 白

HPLT 横坐标, 纵坐标

描绘一点

HPLT 0, 0 **HPLT** 3, 15

HPLT TO 横坐标, 纵坐标

由当前坐标划线到指定坐标

HPLT TO 35, 57

HPLOT 横坐标 1, 纵坐标 1 TO 横坐标 2, 纵坐标 2
 TO 横坐标 3, 纵坐标 3 TO.....

描绘一组线

HPLOT 0, 0 TO 279, 0 TO
 279, 191...

高分辨率图形机器语言造型表

造型向量定义表

动作	↑	→	↓	←	↑	→	↓	←
二进制	000	001	010	011	100	101	110	111
十进制	0	1	2	3	4	5	6	7

向量字节字位表 (举例)

	向 量	字 位	16进码
字节 1	不动 ↑ →	0 0 1 0 0 1 0 1	25
字节 2	不动 ↓ ←	0 0 1 1 0 1 1 1	37
⋮	⋮	⋮	⋮
字节 i	造型表结束	0 0 0 0 0 0 0 0	00

造型表索引 (举例)

字节	0	1	2	3	4	5	6	7	
值	01	00	04	00	25	37	00		
说明	造型数	空字节	绝对 低位	距离 离位	造型定义	结束			

造型表的存入

可放入 768~975 存储单元之间 (举例)

```
POKE 768, 01 ] POKE 769, 00
POKE 770, 04 ] POKE 771, 00
POKE 772, 25 ] POKE 773, 37
POKE 774, 00
```


使用图形造型指令

SCALE = 点数 指定每个向量绘制点数
SCALE = 1 *SCALE* = 5
DRAW 造型号数 [*AT* 横坐标, 纵坐标]
将指定的造型显示在指定的起始位置
或当前光标位置
DRAW 1 *AT* 140, 96 *DRAW* 10

VDRAW 造型号数 *AT* 横坐标, 纵坐标
将指定起始位置的指定造型消去
VDRAW 1 *AT* 140, 96

ROT = 数值 旋转造型坐标系
数值为 0 ~ 64, 根据 *SCALE* 设置点数确定。

SCALE = 数值 圆图形分割 *ROT* = 16 *ROT* = 48
SCALE = 1 2 3 4 5
圆分割次数 4 8 16 32 64
取值域 $2^1 \times n$ $2^3 \times n$ $2^2 \times n$ $2 \times n$ $1 \times n$

屏幕控制

文本屏幕的设置

POKE 32, 0 ~ 39 设置屏幕显示左极限
POKE 33, 0 ~ 40 设置屏幕显示宽度
POKE 34, 0 ~ 24 设置屏幕显示上限
POKE 35, 0 ~ 24 设置屏幕显示下限
POKE 36, 0 ~ 39 设置光标水平位置
POKE 37, 0 ~ 23 设置光标垂直位置
CALL -868 清除光标所在行 (同 *SEC-E*)
CALL -912 屏幕上卷
CALL -922 光标换行 (同 *CTRL-J*)
CALL -936 清除屏幕 (同 *HOME* 指令)
CALL -958 清除光标后屏幕 (同 *SEC-F*)