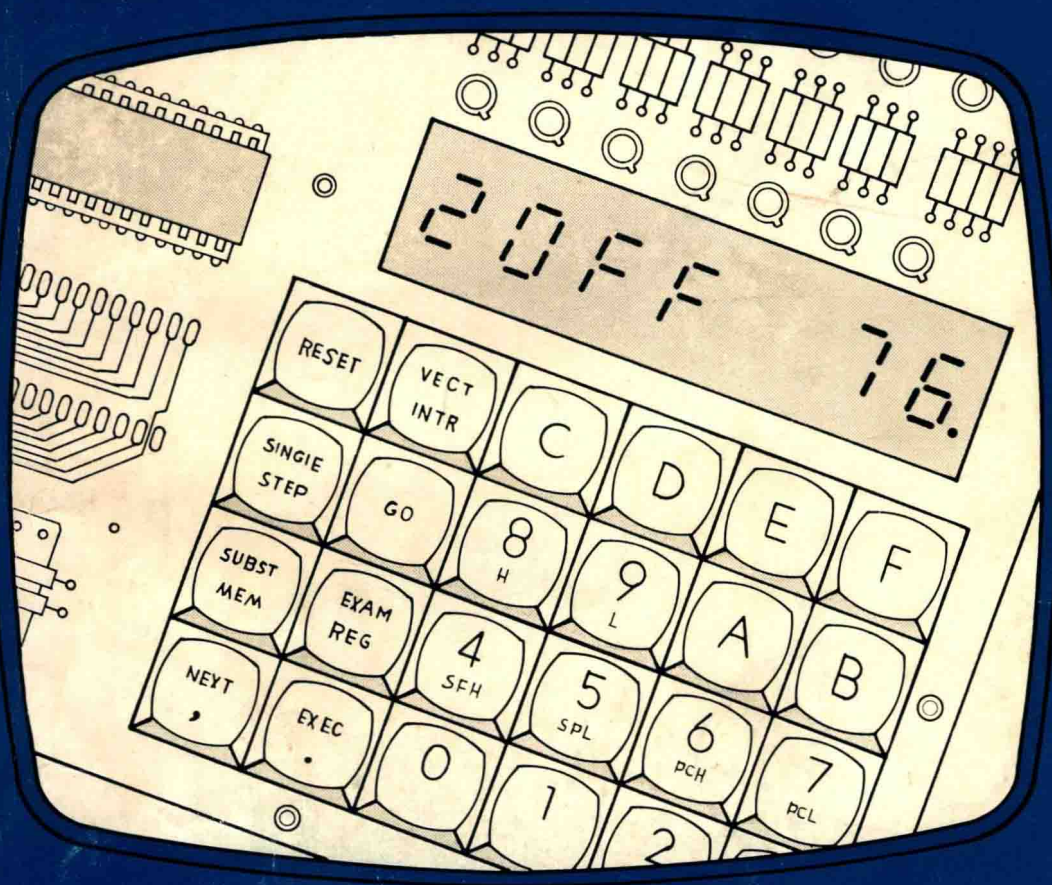


# 微型计算机

microcomputer



14

上海交通大学

技术情报资料室  
微型计算机研究室

# DJS-053 通用微型计算机 系统总体设计

杜毅仁 赵正校 徐子亮 朱煜清

江苏工业学院图书馆  
藏书章

# 目 录

1. DJS - 053 通用微型计算机系统总体设计 ..... 1—1 ~ 1—25
2. DJS - 053 微型计算机监控程序 ..... 2—1 ~ 2—63
3. DJS - 053 微型计算机 MDOS - 53 操作系统  
..... 3—1 ~ 3—115
4. DJS - 053 64K 动态存贮器 ..... 4—1 ~ 4—6
5. DJS - 053 64K 静态存贮器..... 5—1 ~ 5—10
6. DJS - 053 微型机 CRT 接口电路 ..... 6—1 ~ 6—12
7. DJS - 053 行式打印机接口 ..... 7—1 ~ 7—4
8. DJS - 053 微型计算机与盒式  
磁带机高速接口的研制 ..... 8—1 ~ 8—12
9. DJS - 053 软盘控制器及其与主机的连接  
..... 9—1 ~ 9—14
10. DJS - 053 EPROM 的写入装置 ..... 10—1 ~ 10—4
11. DJS - 053 微型计算机主机电源 ..... 11—1 ~ 11—10
12. DJS - 053 微型计算机的机械结构设计 ..... 12—1 ~ 12—6

## DJS-053 通用微型计算机系统总体设计

### 一、 引 言

DJS-053 是一台以 8085A 为中央处理部件的通用微型计算机系统；其半导体内存为 64KB（B 表示字节， $K=1024$ ），一对单面双密度软盘驱动器；每张软盘的存储容量为 500KB，两张软盘的容量则为 1 MB；本系统最多可带两对这样的双驱动器，这时总容量可达 2 MB；除标准的 CRT 显示器，输入键盘和小型打印机外，还增加了 1200 波特率的音频盒式录音机及 EPROM 程控写入器等外设。软件方面除 DJS-053 的 4 KB 监控程序 MONITOR-53；软盘操作系统 MDOS-53；汇编语言 ASM-53；编辑程序 EDIT-53 等基本软件外，还配有 PL/M，BASIC 和 FORTRAN 等高级语言。以及若干应用程序。从而使 DJS-053 既有一般高档通用微型计算机系统的特点，还兼有某些微处理机开发系统的功能。整个系统装在一个单人写字台式的机箱内，如图 1 之照片所示，使 DJS-053 显得完整、协调。

实际运行表明，DJS-053 的性能达到了 1977~1978 年间国际上名牌产品的水平，为我国提供了一套自行设计、自行研制、先进、可靠的高档通用微型计算机系统。值得指出的是，由于 DJS-053 是按国际通用标准设计和研制的，而且采用 Intel 公司 806 线的多重系统总线体系，因而与国际上同类系统在硬件和软件上均是充分兼容的。这不仅意味着国内外用户可以方便地使用这个系统，并在软件和硬件上获得强大的支持，更重要的是具有进一步扩充的可能性，新出现的软件可以引入本系统，新开发的硬件亦可通过标准的单板插件纳入本系统，使 DJS-053 具有很强的适应能力。

DJS-053 可以广泛用于生产控制，专用微处理机研制，辅助设计，智能终端，生产管理，文件处理，医学应用，教学研究，销售管理，数据处理，会计管理等方面。由于 DJS-053 的系统配置，可以根据用户和应用对象的要求，合理安排，可大可小，从而进一步扩展了 DJS-053 微型机的应用范围。

## 二、系统构成

图2是总机框图，8085A是DJS-053的CPU，一般情况下，均由它指挥一切，唯一例外是当软盘工作时，总线控制权暂时让给软盘控制器，因而8085A的基本特性，在很大程度上决定了系统的性能。

### 1. 8085A的指令系统

8085A的操作码为一个字节，故8085A的机器指令共有256种，其中有254种与8080A完全一样，只有RIM和SIM两条指令是新增加的，（这两条指令用来控制转接口和为中断屏蔽编制程序的）为了纳入我国050系列，在总体设计中规定暂时不使用这两条指令，同时把8085A相对8080A所增加的功能引脚亦不使用，一句话，就是把8085A当成（8080A+8224+8228）使用，从而使DJS-053符合我国050系列的各项技术规范。

8085A的指令有三种形式：

① 单字节指令：这是主要的指令形式，单字节既是操作码，亦隐含了操作数的地址，即一般称为隐含寻址的方式。

② 双字节指令，第一字节仍为操作码，第二字节表示立即数或操作数或I/O转接口地址，这意味着立即寻址方式。

③ 三字节指令，第一字节仍为操作码，第二字节表示立即数或操作数地址之低八位，第三字节表示立即数或操作数地址的高八位，这意味着16位立即数寻址方式或直接存贮器寻址方式。

指令执行时间与寻址方式和指令类型有关，DJS-053系统的基本时钟为3.072MHz（8085A片内时钟发生器需要外接一个晶体或RC网络，其谐振频率为6.144MHz，是基本频率的两倍）一般以时钟周期为计时单位，8085A的指令执行时间最短者需要4个时钟周期，即 $1.3\mu\text{s}$ ，最长为18个时钟周期约 $5.9\mu\text{s}$ （见附录I）

8085A的指令系统，按功能可分成五大类，数据传送指令组，逻辑操作指令组；算术操作指令组；转移指令组，以及堆栈，I/O和机器控制指令组，各种指令的助记符，注解及执行时间（按时钟周期数计算，每个时钟周期值为 $0.326\mu\text{s}$ ）均列于附录1中。

### 2. 十六位地址总线



由图 2 可知, 8085A 的 16 根地址线中, 高八位由  $A_8 \sim A_{15}$  引脚直接输出。低八位则与数据线共用, 由  $AD_0 \sim AD_7$  分时进出, 在每次机器周期的开始, 由  $\overline{ALE}$  之后沿选通, 使低八位地址锁存于 8283 (8 位带锁存的反相驱动器) 两只 8283 的 16 位地址的反码沿系统地址总线,  $\overline{AB}$  输出, 其上之地址经反相缓冲以后, 既可对 64KB 的 RAM 寻址, 亦可访问主机板内的 EPROM, 其中存放着 DIS-053 之监控程序 (4KB), 尚有 12KB 的空位留给其他程序固化, 此外,  $\overline{AB}$  线上的地址还经地址译码器和随机逻辑电路以后, 向各 I/O 接口片提供片选信号。

### 3. 八位双相数据总线:

八位数据经双相的总线驱动器 8287 进出系统 数据总线  $\overline{DB}$ , 在主机板内, 再经 8287 反相, 形成主机板内的数据总线 DB, 以便与外围片子进行通信, 为什么在系统总线中要采用反码呢? 这是因为在较大的系统中 8085A 不能直接驱动众多的外围片子, 势必通过总线驱动器, 反相的总线驱动器的延时小, 经较长的线路把信号送到需要部件 (列如 RAM), 再反相以后即可驱动比较重的负载, 这样安排主要是为了提高系统的可靠性, 同时亦可减小传输延迟。

### 4. CRT 显示器及通用 ASCII 键盘:

DIS-053 主机板上装有三片可程序通信接口组件 (8251) 其中一片与 CRT 接口联接, 沟通了 CPU 与输入键盘及 CRT 显示器之间的串行通道, 目前是按 1200 波特率与 CRT 进行通信, 如用户需要可用程序改变。

### 5. 盒式录音机及其接口

另一片 8251 与盒式音频录音机接口电路相联, 沟通了 CPU 与录音机之间的录/放通道, 按 1200 波特率对盒式磁带进行写入和读出, 一盘 60 分钟的磁带, 单面可存 80KB, 双面的容量为 160 160KB, 磁带是由监控程序直接控制进行读、写的, 在写入时, 可先打入文件名, 然后按首址将内存里的程序写入磁带, CRT 显示其写入内容, 预定的内容写完则停止写入, CRT 亦停止跟踪, 读出时, 先由键盘打入需要文件名和存放该文件的内存首址等内容, 起动以后, 自动寻找文件名, 这时 CRT 不跟踪, 一旦找到, 则 CRT 跟踪显示

文件名，并将其内容读出 存储在规定的内存区，CRT此时亦跟踪显示这些内容，文件结束时，CRT亦停止。并自动显示出错地址和错误信息，这种工作方式有点系软盘的工作过程，但它给用户提供一个价格极其低廉的外存储器，其主要用途可用来取代光电带，光电输入机和穿孔机，但有时亦可暂时顶替软盘的某些任务。

6. 第三片8251 留给用户使用（主要是留给远程通信用的）

7. 打印机及其接口：

DJS-053 中装有两片可程序外围接口片8255，其中一片的PA转接口，通过8286缓冲以后，与打印机接口相联，PC转接口的一半，即PC<sub>4</sub>~PC<sub>7</sub>直接与打印机接口相联，从而沟通了小型打印机与CPU之间的输出通道，在CPU控制下，可以按各种打印机要求的速度打印字符，本系统采用的打印机具有小型、省电、标准、价廉等特点，其打印速度也比较适中，可打大、中、小三种字型，对中字来说，每行80个字符，其速度为28 LPM。大字每行50字符，小字每行132字符。

8. EPROM 写入器及其接口

另一片8255与EPROM写入接口相联，沟通了EPROM输入器与CPU之间的通道，在监控程序的控制下，可以对2716，2708，8755等EPROM进行写入和读出。由于这个接口是外接的，故用户可以把它排作他用。

9. 可程序时间脉冲发生器

DJS-053 中装有一片可程序时间脉冲发生器8253，它可提供三种不同频率的时间脉冲，其中一个送给两片8251，使其按1200波特率工作，另一个提供给第三片8251，尚余一个留给用户使用。

10. 中断处理

DJS-053 中，采用8259来处理八级优先级中断，总体设计决定8259是按完全嵌套方式工作的，故八个中断请求中，以IR<sub>0</sub>的优先权最高（称为0级中断），IR<sub>7</sub>最低，当中断被8085A响应时，优先级最高的首先被确定，向8085A发出中断请求，8085A处理完当前程序，允许中断时，则向8259发出中断应答信号（INTA），8259即向8085A提供三字节无条件调指令（CALL），当8085A

完全收到该指令后，即将主程序之PC值（16位）压入堆栈保存，并将CALL指令后面两个字节的内容输入程序计数器（PC），从而开始执行中断程序，当中断程序结束，则从堆栈中弹出主程序的PC值输入PC，使机器又按主程序规定的轨道执行下去。

### 11. 64KB 内存

DJS-053 提供两种RAM板，一种由4K位的静态RAM片构成，每块印制板的容量为32KB，读写周期为310ns，单5V电源，耗电0.7A（包括所有外用电路）共用两块这样的RAM插件。

另一种是由16K位的动态RAM片构成，只用一块印制板其容量就可达64KB，采用三种电源，-5V、+5V和+12V，其中+5V的电流为0.8A，采用动态RAM板不单只用一块，印制板插件，而且功率下降80%，价格降低3倍，但是需要三个电源。

### 12. 软盘控制器及其接口

由两块印制板构成的软盘控制器及接口电路，是国际市场上典型产品，它可以控制两对双驱动器，从而使系统最多可以带上四张单面双密度的软盘，每张存储容量为500KB，总容量可达2MB，一般情况下，DJS-053只配置一对双驱动器，即提供两张软盘在系统中使用。

DJS-053 配有强有力的软盘操作系统 MDDS-53，

当DJS-053的CPU要调用磁盘时，先向磁盘控制器和接口电路发送访盘参数块之首址；磁盘控制器收到后，就向CPU发出请求8085A让出总线；8085A一旦响应其请求，发出HLDA应答信号给磁盘控制器。同时8085A处于保持状态，并让出系统总线；软盘控制器占用总线后，从而成为系统的主控器，它通过系统总线对RAM进行读或写，每次读或写发出之后，要求RAM在其数据出入总线之后，应给软盘控制器回送一个信号，软盘收到它后，马上对此数据进行读或写；然后软盘控制器把总线还给8085A；8085A控制总线后，马上向软盘接口发出询问，测试其操作是否完成，如来完成，又向8085A发出使用总线的请求，8085A答应并让出总线后，再度使自己处于保持状态，重复上述操作如此继续下去，直到整个数据交换的过程结束为止，当整个过程结束时，如果数据交换得正确，则磁盘



控制器将输出正常的标志，如果有差错，软盘控制器将输出表示不同类型错误的标志码。实际运行表明，这种软盘控制器及其接口电路的工作是很令人满意的。

### 13. 电源

DJS-053 共使用四种电源，即+5V、-5V、+12V、-12V。写 EPROM 所用之+25伏电源则由外部供给，四种电源中+5V是主要电源，最大电流值为8A，主机功耗不超过80W。

### 14. 多重系统总线结构：

DJS-053 的基本系统共用4块310mm×175mm（或5块，如果采用静态RAM板），印制板。为了与国际上同类系统兼容，故总体设计决定采用Intel公司的多重系统总线体系，印制板的横向尺寸亦同，六只86芯的插座装在一块总线印制板上，构成一个六块插件的主体机箱，其中实用4块（或五块）余下的空位留给用户作为进一步扩充时使用。多重系统总线各线的含义及符号均附在文末之附录中，供读者参考。

采用这种总线结构的优点是显而易见的，除了使主机系统规整，装配简易，调试方便互换性强等优点外，更重要的是便于扩充，任何扩充性的单板均能方便地纳入本系统，从而提高了DJS-053的适应能力。

## 三、系统基本工作原理

系统的每一个操作过程，都可以看成是8085A单独或在外部组件的配合下，执行某一条8085A指令的过程。现以STA指令的执行过程为例，来说明部分工作原理，STA是把8085A内累加器中的内容存入第二字节和第三字节所指定的内存单元。执行这条指令共需四个机器周期，而每一个机器周期又由3~6个时钟周期构成，在8085A中共七种性质不同机器周期，这些周期主要与控制总线上三个信号状态有关，如表1所示，图3是STA指令的时间波形图，由图3可知。

### 1. 取操作码机器周期( $M_1$ )

表 1 8085A 的机器周期表

| 机器周期名称        | 状 态               |       |       |
|---------------|-------------------|-------|-------|
|               | $IO/\overline{M}$ | $S_1$ | $S_0$ |
| 取操作码周期 (OF)   | 0                 | 1     | 1     |
| 存储器读周期 (MR)   | 0                 | 1     | 0     |
| 存储器写周期 (MW)   | 0                 | 0     | 1     |
| I/O 读周期 (IOR) | 1                 | 1     | 0     |
| I/O 写周期 (IOW) | 1                 | 0     | 1     |
| 中断响应周期 (INA)  | 1                 | 1     | 1     |
| 总线空闲周期 (BI)   | 0                 | 1     | 0     |

8085A 在每一个机器周期开始时,均要在  $IO/\overline{M}$ 、 $S_0$ 、 $S_1$  三个输出端上送出控制讯号,以确定是那一类机器周期,在  $M_1$  的  $T_1$  时,由于  $IO/\overline{M}=0$ ,  $S_0=1$ ,  $S_1=1$  故表明是取操作数周期, (FO) 因而8085A 把其PC之内容馈入16位地址总线,高八位地址沿ABH送出是容易理解的,PC的低八位则是在ALE后沿的选通下锁存到ABL的8283中,这16根地址总线沿系统总线馈入RAM板的地址总线,选定RAM中的某一单元,在  $M_1$  的  $T_2$  时期,8085A 送出读出控制信号RD,在  $IO/\overline{M}=0$  的配合下,读出上述RAM单元中的内容(就是STA的操作码32H)沿DB总线馈入8085A,在  $T_3$  后期RD回到高电平,结束数据沿AD总线送入8085A的过程,这意味着STA操作码已存入8085A中的指令寄存器,并在  $M_1$  的  $T_4$  周期中对操作码进行识别,并决定执行该条指令所需的操作,因为是STA,故要求CPU必须接着执行两个存储器读周期和一个存储器写机器周期,才能完成STA指令所要求的操作。顺便说明一下,有的指令要求  $M_1$  持续6个时钟周期,有的则只要求4个时钟周期,这可以从附录1中所列举的时钟数目中分析出来。

## 2. 存储读周期:

因为是 STA 指令, 故  $(PC + 1)$ , 并在  $M_2$  周期的  $T_1$  时使  $IO/\overline{M} = 0$   $S_1 = 1$ 、 $S_0 = 0$ , 指出  $M_2$  是存储读周期与前面过程一样, 在  $M_2$  之  $T_1$  时期, PCH 沿 ABH, PCL 在 ALE 后沿的选通下锁存于 ABL 的 8287 中, 然后沿系统总线送到 RAM, 选中 RAM 下一个地址,  $M_2$  的  $T_2$  时期 8085A 发出  $\overline{RD}$  信号, 控制读出选中单元的内容, 沿 8 位  $\overline{DB}$  总线输入 8085A, 当  $T_3$  结束以前, 操作数低 8 位地址已输入 8085A, 值得注意的是, 在 8085A 中规定除取操作数周期  $M_1$  外, 其余的周期均由三个时钟周期组成。因而  $T_3$  以后, 接着就进入  $M_3$  周期。

因为是 STA, 故  $M_3$  仍为存储器读周期, 即 PC 把再度加 1 后的 16 位地址, 沿  $\overline{AB}$  总线送给 RAM, 选中第三个指令字节, 在  $\overline{RD}$  和  $IO/\overline{M} = 0$  的控制下, 读出其内容, 沿  $\overline{DB}$  总线馈入 8085A 中, 到  $M_3$  的  $T_3$  时, 8085A 已经全部收到 STA 的三个指令字节, 现在它应该执行这条指令了, 即把累加器中的内容存入指定的内存单元, 这些操作将在  $M_4$  周期完成。

### 3. 存储器写操作周期:

因为是 STA 指令, 故  $M_4$  是存储器写周期, 在  $M_4$  之  $T_1$  时期,  $IO/\overline{M} = 0$ ,  $S_1 = 0$ ,  $S_0 = 1$ , 8085A 把刚才从内存中取得的两个字节数据馈入 16 位地址总线, 在  $M_4$  之  $T_2$  时期, 8085A 发出写控制信号  $\overline{WR}$  在  $T_2 T_3$  时期, 8085A 将 A 中内容沿  $\overline{DB}$  总线送入 RAM 从而在  $\overline{WR}$  和  $IO/\overline{M} = 0$  的控制下, 写入内存指定单元, 完成 STA 的指令操作。接着转入  $M_1$ , 即进入取新指令之操作码的周期, 并且继续工作下去。

### 4. I/O 读和 I/O 写周期

上面以 STA 为例, 描述系统访问 RAM 的读、写过程显然对于系统访问 I/O 接口的过程亦大体相同, 只不过  $IO/\overline{M} = 1$  而已。

值得指出的是, 不管是访问 RAM, 还是访问 I/O 接口, 只要在  $T_2$  时间 READY 端为低电平, 均可在  $T_2$  和  $T_3$  之间插入等待周期 (T<sub>WAIT</sub>) 如图中所示, 以此来拉长访问存储器或 I/O 接口的时间, 从而使慢速的存储器和 I/O 接口能够与 8085A 配合使用。

## 5. 中断响应周期

当外部中断被 8085A 响应，而发出 INTA 以后，外部中断逻辑应在  $M_1$  时期向 8085A 提供指令操作码，一般 RESET 和 CALL 指令是最合理的选择对象，因为这两种指令迫使 8085A 在转移到中断程序之前，先把 PC 的内容压入堆栈，以便在中断结束后，恢复主程序的秩序。本系统采用 8259 来处理中断，故在  $M_1$  时期，8259 可以把 CALL 指令的操作码馈入 8085A 中，在这种情况下，8085A 即进行译码，同时确定 CALL 指令所需要的另外两个指令字节，如图 4 所，8085A 执行第二个中断周期  $M_2$ ，以便从 8259 中取出第二字节，接着在  $M_3$  中又从 8259 中取出 CALL 指令的第三字节，到此为止，8085A 已经收到用于响应中断的完整指令 CALL，值得注意的是，在 8085A 从 8259 中取得三字节 CALL 指令的三个中断周期时期，8085A 禁止程序计数器递增，这样，才能在中断的  $M_4$  和  $M_5$  中，把正确的 16 位 PC 值压入堆栈，同时把 CALL 指令的第二和第三字节分别置入 PC，从而把程序转移到 CALL 指令所指部位，从这里开始中断程序的执行。详细的时间波形如图 4 所示。

## 6. 总线空闲周期

从上述可知，8085A 中的大多数机器周期均与读或写有关，但有两例外。

第一个例外发生在 DAD 指令期间，因为 8085A 需要六个时钟周期时间去执行 DAD 所要求的双倍字长加法操作。可是又不希望  $M_1$  具有 10 个时钟周期，解决办法是在 DAD 指令时，8085A 特别发生两个额外的机器周期，每个周期仍持续三个时钟周期，由于在这两个机器周期中不读也不写，故称为总线空闲周期，这时除  $\overline{RD}$  维持高电平及 ALE 不发生之外，总线空闲周期与存储器读周期将是一样的。

另一个例外发生在 RST5.5、RST6.5、RST7.5 或 TRAP 中断内部操作码发生期间，由于这些中断是 8085A 所增加的，总体设计规定暂不使用，故不赘述。

#### 四、软 件

DJS-053 配有丰富有力的软件；主要有：

1. 4KB 监控程序 (MONITOR-53)

2. DJS-053 软盘操作系统, MDOS-53,

3. PL/M-53, BASIC, FORTRAN-53 等高级语言, 因为后面有专文论述, 这里就不多加讨论, 只把能在 DJS-053 上进行的主要软件目录打印于后, 这些软件均保存在软盘上, 该软件在软盘上占用的段数及对应的字节数已印在该文件名之后, 读者可看出

DJS-053 的软件是很丰富的。

#### DIRECTORY OF : F0 : MODS.V1

| 文件名    | 扩展部  | 段数   | 字节数    | 属性   |
|--------|------|------|--------|------|
| NAME   | .EXT | BLKS | LENGTH | ATTR |
| MDOS   | .DIR | 26   | 3200   | IF   |
| MDOS   | .MAP | 5    | 512    | IF   |
| MDOS   | .TO  | 24   | 2944   | IF   |
| MDOS   | .LAB | 54   | 6784   | IF   |
| MDOS   | .BIN | 94   | 11740  | SIF  |
| MDOS   | .CLI | 20   | 2402   | SIF  |
| ASM53  |      | 110  | 13728  | WSI  |
| ASM53  | .OV0 | 17   | 1970   | WSI  |
| ASM53  | .OV1 | 17   | 1959   | WSI  |
| ASM53  | .OV2 | 18   | 2110   | WSI  |
| ASM53  | .OV3 | 9    | 988    | WSI  |
| ASM53  | .OV4 | 188  | 23639  | WSI  |
| ASXREF |      | 35   | 4294   | WSI  |
| ATTRIB |      | 40   | 4879   | WSI  |
| BINOBJ |      | 27   | 3246   | WSI  |
| COPY   |      | 68   | 8442   | WSI  |
| DELETE |      | 39   | 4794   | WSI  |
| DIR    |      | 55   | 6810   | WSI  |

DIRECTORY OF : F0 : MODS.V1

| 文件名        | 扩展部  | 段数   | 字节数    | 属性   |
|------------|------|------|--------|------|
| NAME       | .EXT | BLKS | LENGTH | ATTR |
| EDIT       | 58   | 58   | 7235   | WSI  |
| FIXMAP     |      | 52   | 6493   | SI   |
| FORMAT     |      | 62   | 7789   | SI   |
| HDCOPY     |      | 48   | 5989   | WSI  |
| HEXOBJ     |      | 34   | 4128   | WSI  |
| IDISK      |      | 63   | 7890   | WSI  |
| LIB        |      | 82   | 10227  | WSI  |
| LINK       |      | 105  | 13074  | WSI  |
| LINK       | .OYL | 37   | 4578   | WSI  |
| LOCATE     |      | 120  | 15021  | WSI  |
| OBJHEX     |      | 28   | 3332   | WSI  |
| RENAME     |      | 20   | 2341   | WSI  |
| SUBMIT     |      | 39   | 4816   | WSI  |
| SYSTEM.LIB |      | 24   | 2846   | WS   |
| FPAL .LIB  |      | 74   | 9125   | SI   |
| INSERT.ERR |      | 66   | 8192   | W I  |
| BASIC      |      | 184  | 23063  | WSI  |
| FORT53     |      | 36   | 4374   | WS   |
| FPA .LIB   |      | 72   | 8838   | WS   |
| F53RUN.LIB |      | 135  | 16831  | WS   |
| FPEF .LIB  |      | 138  | 17203  | WS   |
| FLINK .CSD |      | 2    | 80     | WS   |
| F53NIO.LIB |      | 4    | 331    | WS   |
| F53ISS.LIB |      | 552  | 69439  | WS   |
| FORT53.OV0 |      | 259  | 32385  | WSI  |
| FORT53.OV1 |      | 54   | 6723   | WSI  |
| FORT53.OV2 |      | 242  | 30339  | WSI  |
| FORT53.OV3 |      | 153  | 19171  | WSI  |



| 文件名    | 扩展部  | 段数   | 字节数    | 属性   |
|--------|------|------|--------|------|
| NAME   | .EXT | BLKS | LENGTH | ATTR |
| FORT53 | .OV4 | 155  | 19441  | WSI  |
| PLM53  | .LIB | 45   | 5615   |      |
| PLM53  |      | 172  | 21605  |      |
| PLM53  | .OV0 | 150  | 18731  |      |
| PLM53  | .OV1 | 232  | 29122  |      |
| PLM53  | .OV2 | 66   | 8156   |      |
| PLM53  | .OV3 | 189  | 23706  |      |
| PLM53  | .OV4 | 72   | 8932   |      |
| BASIC  |      | 184  | 23063  | WSI  |

注：上表中 FORT53 即 FORTRAN - 53

## 五、系统的开启和软件之间的转移

当开启电源或按下 RESET 键以后, DJS-053 就自动转到引导程序, 在其控制下, 首先安排系统的初始状态, 顺序进行下列操作:

1. 设置中断处理片 8259 之控制字, 状态字和屏蔽字。

2. 测试 RAM 的大小, 用 RAM 顶之地址值来确定用户堆栈和监控程序堆栈指示器之初始值, 就是说, 只要 RAM 的容量不要太小, 监控程序均能使用。

3. 设置各外围接口片 (三片 8251, 两片 8255, 一片 8253) 的控制字和工作方式, 即对这些片子进行初始化。

4. 判别软盘驱动器是否准备好, 如果是则将 DJS-053 软盘操作系统 MDOS-53 的常驻部分送到内存指定驻留区 (0000H~3000H 区间), 并使 DJS-053 置于 MDOS-53 的控制下进行工作, 这时 CRT 显示器出现提示符为:

MDOS-53 , V1.1

—

如果软盘系统没有准备好, 则 DJS-053 由监控程序控制, 并在 CRT 上出现下列提示符:

DJS-053 MONITOR

MONITOR READY

\*

从此, 用户即可使用 DJS-053 系统所提供的多种软件进行工作, 各软件之间的转换亦是很方便的, 这可用图 5 来说明。

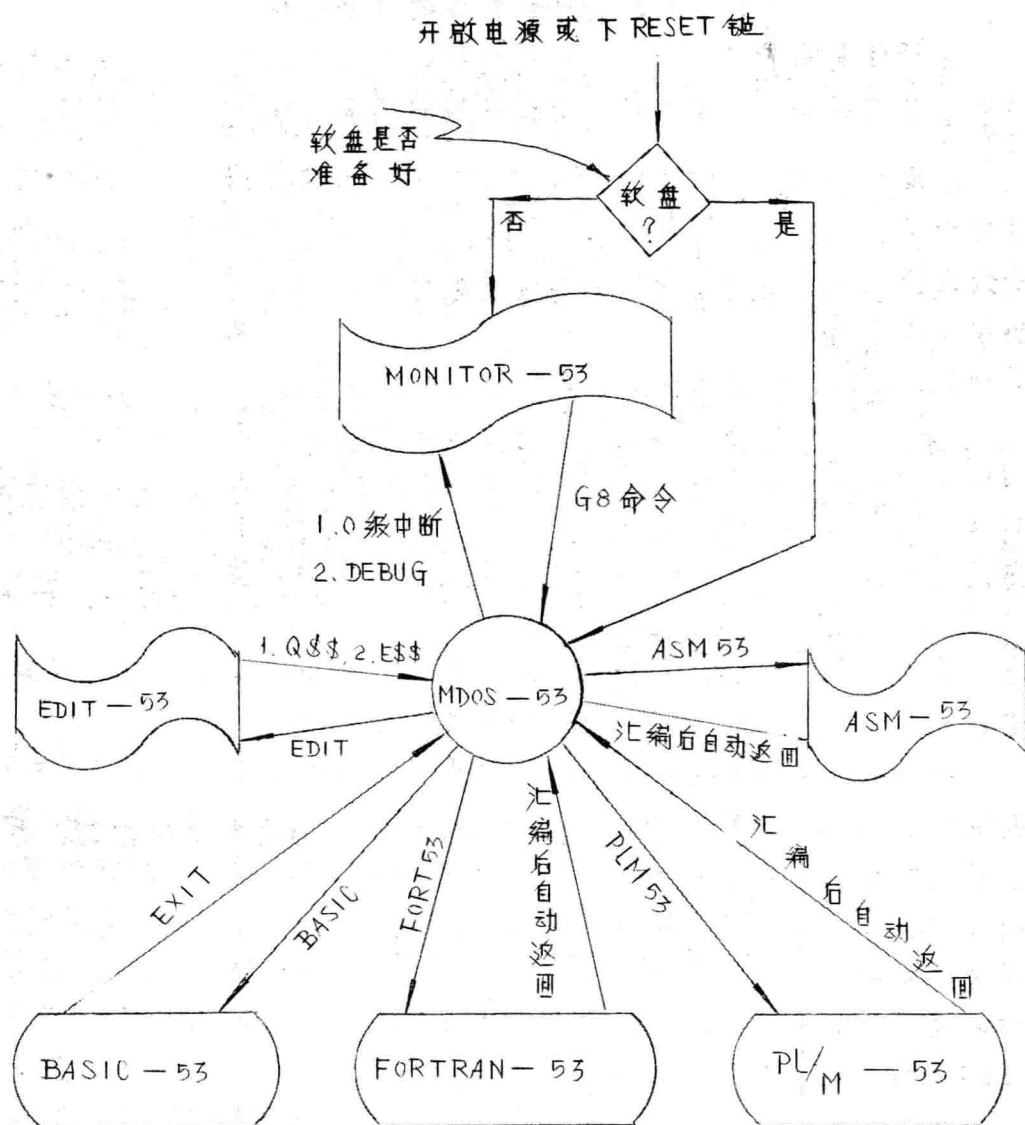


图 5 软件之间的转换