

DJS—100系列电子数字计算机

软件

第二册

单用户BASIC解释程序框面

单用户BASIC解释程序清单

221

DJS—100系列电子数字计算机软件联合设计组

一九七六年一月

北 京

单用户BASIC解释程序框面

· 主程序部份 ·

(开封市电子计算机研制组整理)

DJS—100系列电子数字计算机软件
(第二册)

单用户Basic程序

DJS—100系列电子数字计算机软件联合设计组

*

韶关无线电厂承印
广东省始兴印刷厂印刷
广东省粤北印刷厂制版
1976年5月印刷
印数00001—15000

目 录

开工程序.....	(1)
(00320) 开工.....	(2)
输入和输出.....	(5)
(一) 输入输出的任务和方式.....	(5)
(二) 对输入的管理.....	(5)
(三) 对输出的管理.....	(7)
(四) 输入输出缓冲区.....	(9)
(672) 输入 头.....	(10)
(7614) PTR启动程序.....	(11)
(7652) PTP 启动 程序.....	(11)
(2) 重新 启动.....	(11)
(506) 中断 处理总 控程序.....	(12)
(7766) P T R 处 理.....	(13)
(7762) T T I 处 理.....	(13)
(7771) T T O 处 理.....	(13)
(7776) P T P 处 理.....	(13)
(535) 掉电 处 理.....	(14)
(552) 输出 处 理.....	(15)

(633) 输入处理.....	(16)
换码.....	(19)
(一) 为什么要换码?	(19)
(二) 换码的步骤.....	(19)
(三) 换码的方法.....	(19)
(四) 换码后语句的形式.....	(20)
(五) 机内代码表.....	(22)
(1302) 换 码.....	(23)
编 辑.....	(33)
(一) 编辑的任务.....	(33)
(二) 编辑的方法.....	(34)
(2007) 编辑控制程序.....	(37)
(2231) 编 辑.....	(38)
解释执行.....	(45)
(3246) RUN 执行.....	(46)
(3362) 取执行语句.....	(47)
(3414) 分析为何种语句.....	(48)
(3666) END语句的执行.....	(49)
(3673) STOP语句的执行.....	(50)
(1077) $\square @ nnn \times \text{NULL} \rightarrow \text{I 区}$	(50)
(3477) GOTO GOSUB 语句的执行.....	(51)
(3653) DIM语句的执行.....	(52)

DIM语句的执行和数组.....	(52)
(3540) DEF语句的执行.....	(53)
(3714) LET语句的执行.....	(54)
(3324) READ语句的执行.....	(55)
(3274) RESTOR语句的执行.....	(56)
(3526) RETURN语句的执行.....	(56)
转子返回栈.....	(56)
(3571) IF语句的执行.....	(57)
(4140) FOR语句的执行.....	(59)
(4314) NEXT语句的执行.....	(63)
循环的执行和循环参栈.....	(64)
1、循环和循环的层.....	(64)
2、循环变址.....	(64)
3、循环语句的执行.....	(65)
(4071) INPUT语句的执行.....	(66)
(3735) PRINT语句的执行.....	(67)
(3207) NEW 的 执 行.....	(70)
(2644) LIST 的 执 行.....	(71)
附录 1、输入处理与状态关系表.....	(78)
附录 2、键盘操作表.....	(79)
附录 3、BASIC工作状态.....	(80)
附录 4、错误标志表.....	(61)

开 工 程 序

开工程序是为解释系统的正式运行作准备的。它做了如下几点准备工作：

1、清两个在输入输出管理中有重要作用的工作单元。

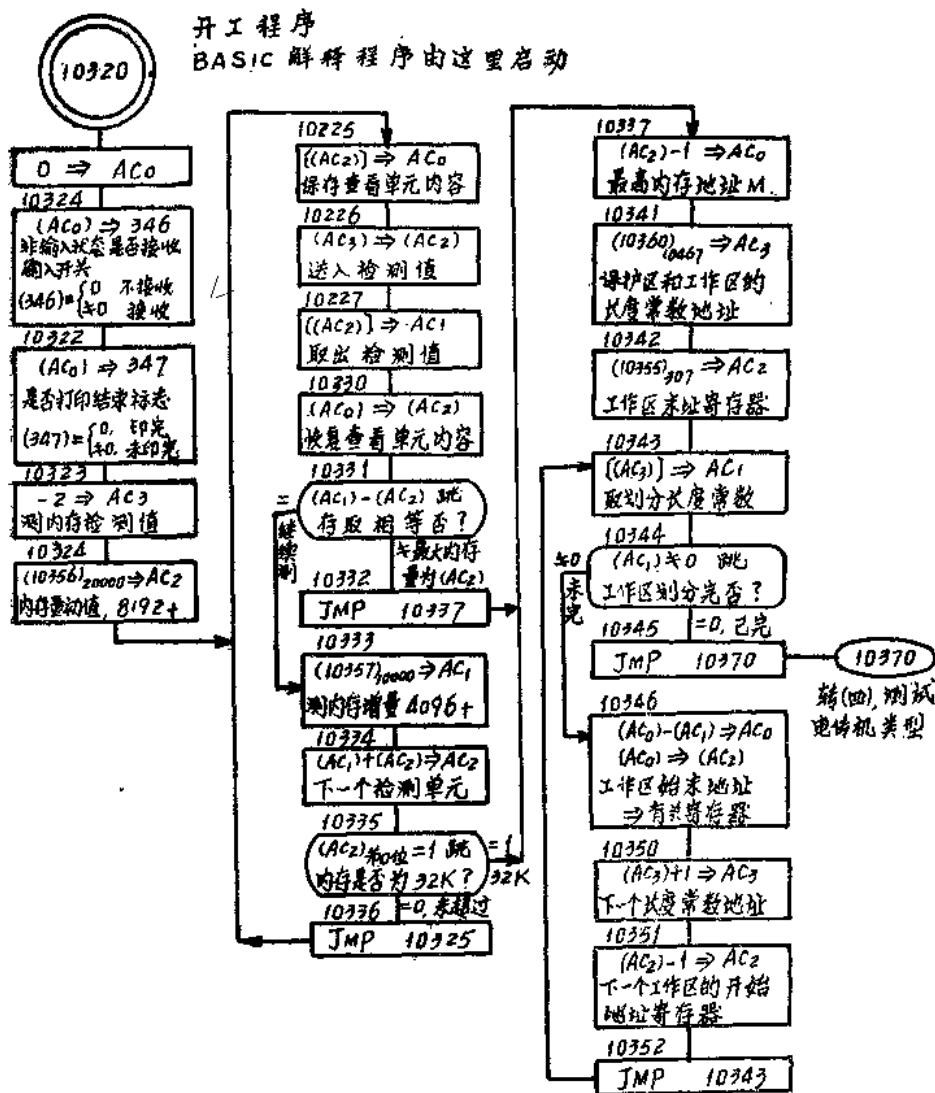
2、测定内存容量（系统要求最少8K内存），并根据测定值划定位于内存高部的几个工作区。

3、测定段配置的电传机的类型（五单位的或是八单位的）。操作人员应该做的配合是，系统运行后，按一个可以标志电传机类别的键符（例如八单位电传时的字母键，五单位电传时任何键都可）。

4、通过TTO向操作人员提问标准输出时一段的打印格数，操作人员见到提问后，应按下电传机的Y键（14格）或N键（15格）进行回答（五单位电传时应先按档符），系统还根据回答和电传机的类型确定一行的总格数。

5、对系统将要使用的工作区一律清0（输入输出缓冲区除外），这时，实现以上1—4点的开工程序部分也被清除掉了，这一部分内存也作为工作区的一部分。

这之后，系统便进入到输入状态，开始接受源程序的输入。



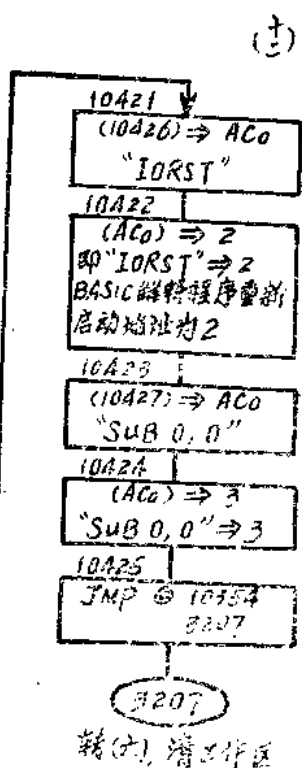
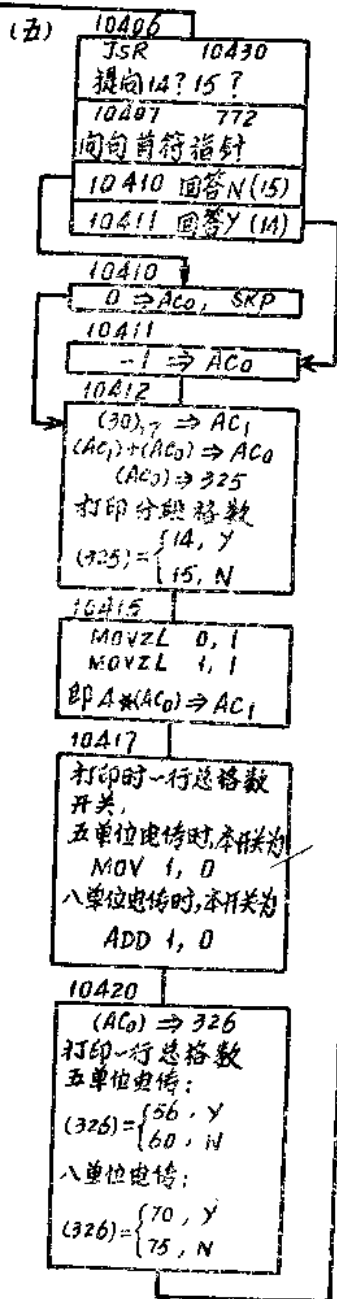
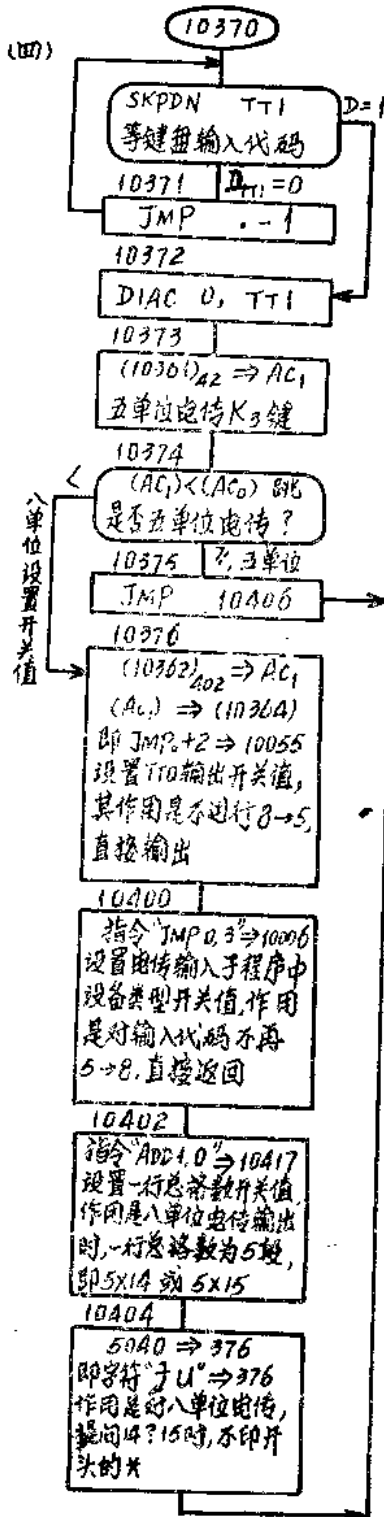
(一) 置特征单元初值

(二) 测内存

(三) 划分工作区, 结果→有关工作地址, 列表如下:

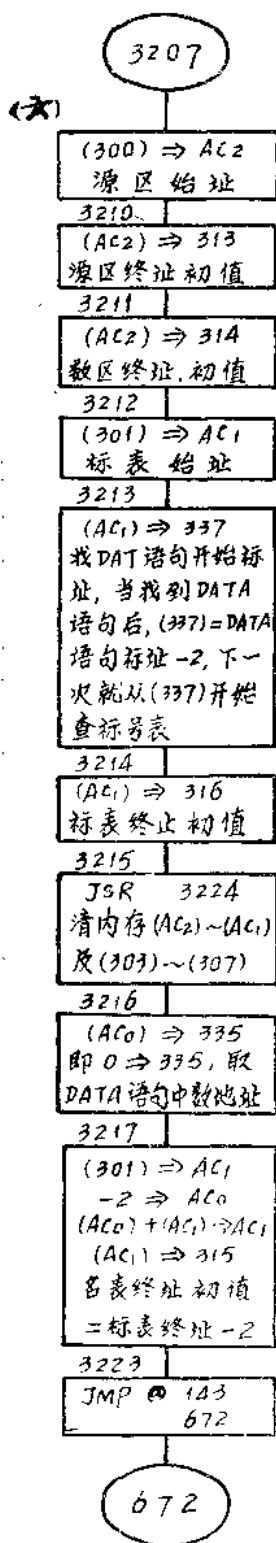
301	M - 420	标号表始址
302	M - 417	第一缓冲区(I区)始址
303	M - 351	自定义函数入口表始址
304	M - 317	自定义函数工作栈始址
305	M - 267	转子返回栈始址
306	M - 257	调变参数栈始址
307	M - 200	调变参数栈末址

M为最大内存地址



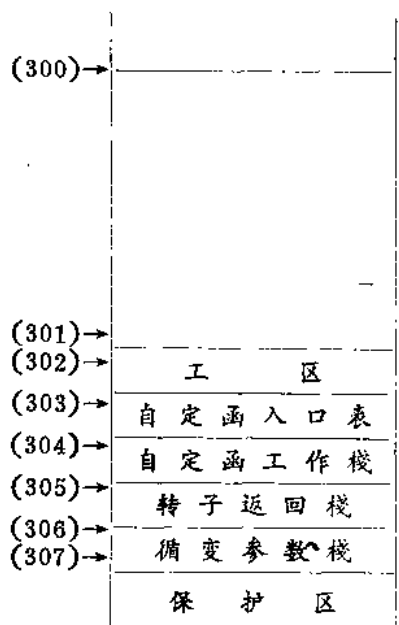
(四)测试电传机类型, 八单位电传时并置诸开
关值

(五)提问14, 15?
设置重新启动入口



清工作区, 给有关工作单元送初值;

1. 清工作区:



2. 送初值

313	(300)	源区 终址
314	(300)	数区 终址
315	(301) - 2	名字表 终址
316	(301)	标号表 终址
335	0	取数地址
377	(301)	找 DATA 开始标址
341	(301)	语句执行标号计数器
342	(306)	循环参数计数器
343	(305)	转子返回栈计数器

输入和输出

一、BASIC系统输入输出的任务和方式

	任 务	方 式
输 入	1. 有标号的源程序语句	纸带或键盘
	2. 键盘命令和键盘运输语句	键 盘
	3. INPUT即时输入	
输 出	1. 有标号的 PRINT 语句输出 (字符串或结果数据)	凿孔或打印
	2. LIST列程序清单输出	打 印
	3. PRINT键盘运算语句输出	
	4. 错误输出	
	5. 对键盘输入信息的显示性打印	

从上面的表不难看出, BASIC系统对外部设备的适应性是很强的。目前, 多数DJS-130机配备的电传机是55型的, 这种电传机只有打印输出和键盘输入两种功能, 无论输出或输入, 代码均是五单位的, 而且还有上下档符之分。就光电输入机和凿孔机而言, 输入和输出的纸带类型, 用户根据自己的具体情况, 又可能有五、七、八单位之分。BASIC系统考虑到了这些情况, 它不仅能适应八单位ASCII代码的电传和纸带的输入输出, 而且也能适应字符为五单位国际代码的电传或纸带的输入输出。这当然要求系统对输入输出要有正确的管理。如果管理不当, 系统就不能有正确的控制, 不能正确的进行输入和输出, 当然更不能对源程序进行正确的解释执行。

BASIC语言具有会话性的特点。会话性, 就是在答题过程中进行简单的人—机“对话”。BASIC系统允许进行键盘操作来查错, 修改和控制运算的进行, 也可以对某些语句进行即时输入, 解释执行和查看结果。键盘操作 (见附录二) 和BASIC系统的其它功能有机配合, 使解题方便灵活。BASIC语言会话性的这个特点, 使得电传机作为操作人员 and 系统之间传递信息的桥梁, 在系统整个运行中有特殊作用。体现系统的会话性, 是系统对输入输出正确管理的一个重要方面。

二、对输入的管理

1、系统对解释程序的运行划分成五种工作状态, 分别用工作单元332的五个号码——0、

1、2、3、4来标志（见附录三）。

为了保证系统的运行协调地进行，系统要求除去INPUT即时输入这种情况之外，只有当 $(332) = 1$ ，即输入状态，方接受来自纸带或键盘的信息。就是说，只有在输入状态中，源程序语句、键盘运算语句和键盘命令才可以被系统接受。但是BASIC语言会话性的特点，允许操作人员随时可以对系统进行控制。为此，系统设置了一个例外，即ESC（在五单位电传机上，电K1键或X键代替）。这个键，只要机器处于“开中断”状态，系统随时可以接受。

2、系统将输入完全纳入到中断处理程序之中，亦即任何信息都是通过程序中中断输入到机内的。为此，系统设计了一个在整个运行中占有十分重要地位的程序段，即自672开始的“输入头”程序段。当系统开始运行，在开工程序之后，便进入了这个程序段。正是在这个程序段中，系统使 $(332) = 1$ ，进入了输入状态，同时使机器“开中断”，并且在TTO机X之后，输入设备（例如PTR）被启动。这样输入设备通过程序中断，把信息输入到机内。同样，一旦系统完成了当前的任务，控制即转向这个程序段，自动地转换为输入状态，开始对新的输入信息的接收。

把输入完全纳入到中断处理程序之中，使输入信息的必要条件是机器“开中断”例如在换码时，由于换码程序实际上是中断处理程序的一部分，是在“关中断”的状态中进行的，所以除非有错误输出（这时开中断），包括ESC键在内，任何信息都不能输入。不难理解，这对BASIC语言会话性的特点，并没有实质性的影响。

3、为了适应会话性的要求，系统在设置ESC键的同时，还设置了一个类似“中断”的开关，即331单元。操作人员如果需要对系统进行控制，应当先按下ESC键进行“预告”。这时，只要是“开中断”，331被置为一1，否则总保持 $(331) = 0$ 。BASIC系统每当正在做的事情告一段落（例如一个语句已执行完毕，或者列清单时一个语句已输出完毕），总要查看一下331单元的状态，如果 $(331) = -1$ ，控制就被转至“输入头”672程序段，从而自动地转换为输入状态，并且通过电传机发出某种信息（打印某种符号或者X），表示已经做好了接收键盘命令的准备。这时操作人员就可以通过键盘对系统进行有效地控制。而当恢复正常的工作以后，331单元的内容就被恢复为0了。

4、为了清楚地表明，对输入的键盘信息是否被接收，系统有显示性打印的功能（纸带输入时无此功能），每当接收一个字符，系统就控制电传机立即打印这个字符，没有打印就是说没有被接收。当然这也提供了检查输入信息的方便条件。

5、INPUT语句，是即时输入语句。当解释执行INPUT语句时，电传机自动打印“？”，表示系统在主动向操作人员索取数据。这时，虽然系统正处于解释执行状态 $(332) = 3$ ，照样接收并加工处理自键盘输入的信息。那末，在非输入状态中，对于ESC键之外的信息，系统是如何地做出决定，是接收信息并加工处理，还是拒绝接收不预理睬呢？这就是开关346单元。通常，总是 $(346) = 0$ ，而当INPUT语句解释执行时，便置 $(346) \neq 0$ ，如果不是输入状态，中断处理程序每当接到输入信息之后，总要查看346单元的内容，如果 $(346) = 0$ ，则立即退中，即对收到的这个信息不预理睬，继续做正在做的事，如果 $(346) \neq 0$ ，则打印出收到的信息（表示还被接收），直至接到X信息（这表示输入的数据已完）时，恢复恢复 $(346) = 0$ ，进而加工处理刚刚输入的这个数据。

6、对于输入设备，系统既可以适应TTI，也可以适应PTR。但是系统在开工程序之后转到672“输入头”等待输入时，或者由其它状态经过672程序段转为输入状态时（换码状态例外），

只是对TTI的输入做好了初始准备。例如在开工程序之后，当TTO的打字头自提问 14, 15 的问句末尾经过 μ 到新的一行的开始位置时，TTI的信息就能够顺利的被系统接收。倘在这时，操作人员不是使用TTI，而是使用PTR输入，即使已经把纸带放在PTR上，PTR并不能被启动。那末是采取什么措施以适应PTR的输入呢？这就是“PTR启动程序7614”。

在“PTR启动程序”中，系统把中断处理中涉及到输入设备的几条指令取代为适应PTR的指令，并且通过TTO打印“*TAPE?”，向操作人员询问即将输入的纸带的类型（五单位、七单位还是八单位），根据操作人员的回答（按电传机的5，7或8键，五单位电传时要先按档符），系统将纸带类型开关10027单元置入相应的开关值，以便对不同类型的纸带作不同处理（例如对五单位纸带，要5→8；八单元纸带，要检查校验位等）。在这时，系统进入了输入状态，PTR被启动。

但是系统并不能自动地把控制转移到7614，它要求操作人员做一点配合。当系统经过“输入头”程序段等待输入时（这在开工程序后以TTO μ 为标志；在一个源程序已经解释完毕，以TTO印出*READY为标记；查LIST列清单完毕时，以列完了段要求的清单并且TTO μ 为标志等等），若要用PTR输入一个源程序或者补充一段源程序，应当先按面板上的停机键，人工自7614启动“PTR启动程序”，在回答了“*TAPE?”之后，PTR上的源程序就顺利的被系统接受，进入了系统的正常运行。

应当指出，系统对TTI的适应是自动复原的。这是指，一旦系统接受了操作人员的键盘信息（例如在源程序输入后，下达命令RUN μ ），系统就把有关开关值完全地恢复成适应TTI的情况。在这之后，倘若又要使用PTR输入源程序，还必须重新经过上述的步骤。

7、系统对源程序的格式是有限制的。如果是PTR输入，纸带中每一语句的后面，应该尾随 μ 和 α ，以表示一个语句的结束，源程序结束时，在最后一个语句后字符 μ 和 α 的后面，还要再尾随一个 μ 字符。系统在遇到源程序结束标志，即最后一个 α 字符时，就不再启动输入设备，从而停止了PTR输入。但是对于键盘输入时，则只要求每一语句的后面跟随一个 μ 字符，由于对键盘输入，系统有显示性打印功能，操作人员完全清楚在什么时刻应该结束源程序的输入而转至下达键盘命令，所以不再要求以某一个键符作为源程序的标志。

但是，已如上述，倘使电传机是五单位的，ESC键以K₁键或者 α 键代替。事实上，系统严格地区分了纸带上的K₁字符和五单位电传的K₁键，把它们当作完全不同的两个字符，所以对于五单位电传机的K₁键应该正确地使用。例如正从键盘输入一个源程序语句时，倘若接下去按下K₁键，结果就使得系统删去了当前输入的这个语句。当然，对于八单位电传来说，K₁就不再起ESC键的作用。

三、对输出的管理

1、除去对TTI输入符的显示性打印以外，解释系统将其它任何输出（即PRINT输出，LIST输出，错误输出等）都纳入字符串输出的形式之中。字符串的结束标志是“NULL”，就是说，输出处理程序辨认字符串是否已经输出完的标志，是遇到了没有字符“NULL”。所以系统使任何输出的末尾，都要尾随一个字符“NULL”。

2、字符串的输出，主要是使用程序中断来实现的，但是必须首先直接输出字符串的头，即第一个字符，以创造可以使用程序中断的条件（例如使输出设备请求中断等），这就是“头字符

输出子程序611”。换句话说，611子程序和中断输出程序的联合，实现了任何字符串的输出。当然，在调用611子程序时，或者在调用611子程序后，机器必须是“开中断”状态（若为“关中断”状态，应该打开之），这样当输出设备完成第一个字符的输出时，其中断请求便能得到响应，从而对字符串的其它字符进行输出。而一旦遇到 NULL 字符，中断处理程序便保持输出设备的状态为“闲”，使它不再请求中断而退中。

3、输出程序使设备有自动回车，换行的能力。就是说在一个字符串的输出过程中，若已够一行（由多少个字符定义为一行，在开工程序中由“问14、15”子程序根据回答和电传机的类型已被确定），则系统使设备输出一个■和×，之后再继续输出字符串中的字符。这时，如果输出设备是TTO，则我们会看到TTO自动地回车和换行，然后继续打印。

这个自动回车、换行的能力，主要是依靠“列计数器334”进行控制的。事实上，每当输出一个字符，只要不是对设备起控制作用的代码，列计数器334的内容就加1，而每在一个字符的输出之前，系统要先检查一下是否已满一行，满行时，处理程序就先自身使设备输出■和×，然后再继续输出，而在换行的同时，列计数器334就被恢复为0，开始在新的一行的计数。这个性质也使得当字符■和×不在字符串中出现时，当前输出的字符串就相当于上一次输出的字符串的继续。换句话说，要使字符串的输出在新的一行开始，必须以■和×作为字符串的头两个字符，或者先输出一个“■×NULL”。顺便指出，PRINT语句的不同书写形式（即语句最后■字符前有没有逗号“，”或分号“；”），能够确定下一个PRINT语句输出是否自新的一行开始，就是由输出管理的这个功能保证的。

4、为了使输出既能适应TTO，也能适应PTP，系统在输出处理中设计了三个重要的标志单元，它们的功能列表如下：

标志单元	作用	内 容	含 义
7611	指出用户指定的输出设备	0	设备为 TTO
		≠ 0	设备为 PTP
7612	指出当前字符的输出方式	0	使用611子程序直接输出
		1	通过 TTO 中断处理程序输出
		- 1	通过 PTP 中断处理程序输出
7613	指出直接输出时使用的设备	0	使用 TTO
		≠ 0	使用 PTP

标志单元7612和7613，是在输出字符串的头字符的子程序611和输出字符串中其它字符的子程序620中，具体地发挥其开关作用的。实际上，在每次输出字符串之前，总是(7612) = 0，子程序611据此进一步去询问单元7613的状态，并根据这个单元所指出的设备去具体地输出字符串的第一个字符。当第一个字符输出完毕，将发生程序中断，一旦进入中断，和出设备相应的处理程序便改变7612单元的内容，使之和当前输出的方式一致，这样，在中断处理程序具体地调用620

子程序输出字符时，只需根据7612的内容，使用相应的设备将字符输出出去，这个字符输出完毕后，设备又一次地请求中断，直至遇到字符“NULL”使设备为“空闲”状态而退中，同时恢复 $(7612) = 0$ ，为下一个字符串的输出做好了初始准备。

由上可知，标志单元7612是控制字符串的输出过程的，而在这个过程中所使用的设备实际上取决于7613。这个单元的初值是0，在611子程序中对它访问之后，又把它重新恢复为0，所以在一个字符串的输出之前，倘若对它不加限制，则输出的方式就是TTO打印（例如错误输出），对于需要按照用户指定的设备输出的任务，则在输出前先使 $(7611) \rightarrow 7613$ （例如PRINT输出和LIST输出）。

5、和输入的情况类似，系统为使用TTO输出做好了初始准备，就是说当源程序输入结束后，如果操作人员自键盘下达了命令RUN $\blacksquare$ 、GOTOn $\blacksquare$ 或符LIST $\blacksquare$ （nLIST $\blacksquare$ ），则系统对PRINT语句的输出或者清单输出，都是通过TTO完成的。这是因为系统在接收键盘命令之前，在“输入头”程序段672中，已先使上述的三个标志单元一律地被置为0。操作人员若要用PTP输出，就应象使用PTR进行输入一样，在系统等待操作人员的命令时，应先强迫使机器停机，然后人工启动系统中设置的“PTP启动程序”。这个程序将通过电传打印“*TAPE?”，向用户询问凿孔输出时纸带的类型，在接到操作者的回答时，系统将置入控制纸带输出类型的开关值，同时使 $(7611) = 1$ ，以表示用户指定的输出设备为PTP，之后，系统就等待操作人员输键盘命令。这时如果下达命令RUN $\blacksquare$ 、GOTOn $\blacksquare$ 或者LIST $\blacksquare$ （nLIST $\blacksquare$ ），则PRINT的输出或者LIST输出就由PTP完成了。

应该指出，当系统完成了键盘命令时，又一次地为使用TTO输出做好了初始准备。

四、输入输出缓冲区

为了节省内存，输入输出公用一个缓冲区（即I区）。这就要求，在新的输入和新的输出开始前，必须先“等输出完”（子程序1162）。所谓“等输出完”，就是等缓冲区的字符串输出完毕。为此系统设置了开关347单元。这个单元当引用子程序611输出字符串的第一字符时被置为 $(347) \neq 0$ ，只有当输出遇到“NULL”，即遇到字符串的结束符时，才被恢复为 $(347) = 0$ 。或者说“等输出完”就是“等 $(343) = 0$ ”。有了这个开关，使得对缓冲区的使用状况的了解，脱离开具体输出的字符串，从而更加有效地利用程序中断而完成输出任务，在输入输出公用一个缓冲区的条件下提高了效率。

顺便指出，由于只有一个缓冲区，使得PRINT语句作为键盘运算语句时，最多只能包含一个表达式（或字符串），需要打印几个表达式值（或字符串），就需输入几个PRINT语句。

