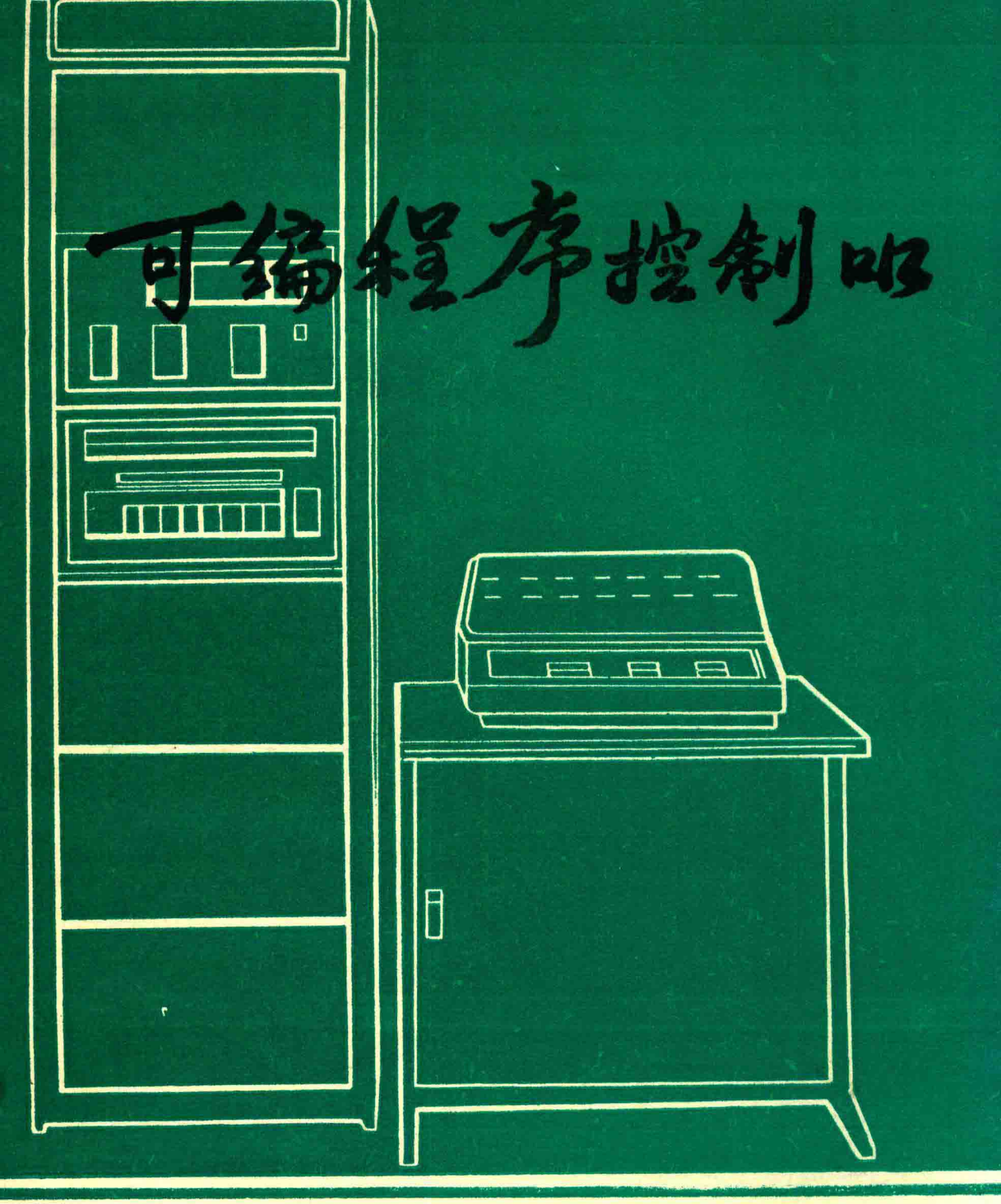
可编程序控制器



第一机械工业部机械研究院机电研究所

一九七六年十月

YODIC-S 可编程序控制器

軟　　件

参考资料之三

毛 主 席 语 录

思想上政治上的路线正确与否是决定一切的。

什么“三项指示为纲”，安定团结不是不要阶级斗争，阶级斗争是纲，其余都是目。

……一切外国的东西，如同我们对于食物一样，必须经过自己的口腔咀嚼和胃肠运动，送进唾液胃液肠液，把它分解为精华和糟粕两部分，然后排泄其糟粕，吸收其精华，才能对我们的身体有益，决不能生吞活剥地毫无批判地吸收。

中国人民有志气，有能力，一定要在不远的将来，赶上和超过世界先进水平。

目　录

第一章 概 述

顺序控制是以继电器控制电路为基础发展起来的。但是，由于继电器电路是一种固定程序系统（由布线实现程序，又称硬接线逻辑系统），所以要修改和增加程序将给设计和制造人员带来许多麻烦。何况不仅设计和制造上的错误，而且控制对象以及产品的变动都会引起程序的更改。为此，本公司设计制造了一种可自由改变程序的可编程序控制器，型号为YODIC—S。

在本章中，首先以继电器为中心，介绍一下程序设计的概念，并就YODIC—S的使用情况作一说明。再将YODIC—S的编程序次序，从决定顺序控制的方法到顺序控制器的完成，以程序设计为中心作一说明，并将涉及到YODIC—S的程序存储方法和执行方法。

还要说明的一点是：本资料旨在阐述编程序的实际业务，因此，本章着重从程序设计的角度进行概述。至于总的设计思想和方案，可参考其他部分（见附录五—1）。

1.1 程序设计

图1.1是两台电动机起动和停止的继电器展开图。现就该例研究一下用什么样的表现方法来编制顺序控制程序。

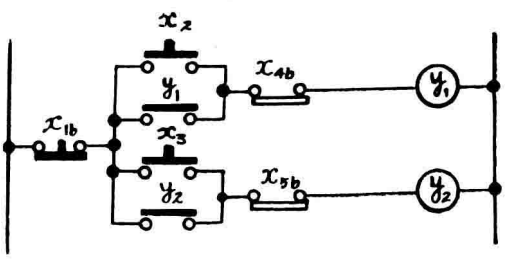


图1.1 继电器展开图

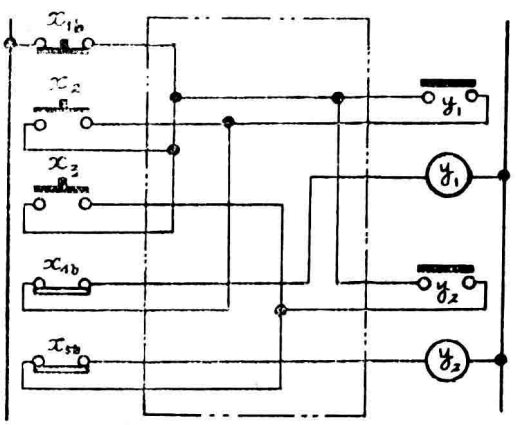


图1.2 变形后的继电器展开图

图1.2是由左图变化而来的继电器展开图。方框内只表示接线部分。图中的接线关系就是继电器控制电路的顺序控制程序。在编制这种控制电路的程序时还要把接点的种类列入程序设计中，以确定究竟使用a接点，还是b接点。在图1.2中，只要少了一条线，或者用了不同的交叉点作为连接点（·），或者改变了接点的种类，都会变成另一种程序。

再来研究一下，当图1.1有更动时，程序将发生什么样的变化。

在图1.1中，x_{1b}是停止按钮，x_2和x_3是起动按钮，x_{4b}和x_{5b}分别是两台电动机自动停止用的限位开关。此外，以各电动机的输出继电器y_1和y_2的辅助接点组成一个自保电路，从

图上可以看出，两台电动机是独立工作的。如果两台电动机联合工作，相互之间就得设置封锁电路，例如控制电梯的升降等等，就需要有这种电路。

假如现在对两台电动机附加不能同时工作的条件，图1.1则变更为图1.3，可先别管这种简单的变化实际上是否会发生，仅研究其程序的变化过程。

图1.4是由图1.3变化而来的继电器展开图。如图所示，（1）～（8）的程序有所改变，（1）、（2）是追加的继电器接点，（3）、（4）是去掉的接线，（5）～（8）是追加的线路。

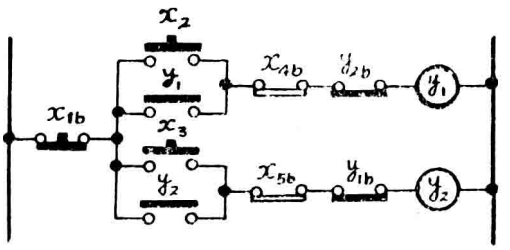


图1.3　工作条件改变后的继电器展开图

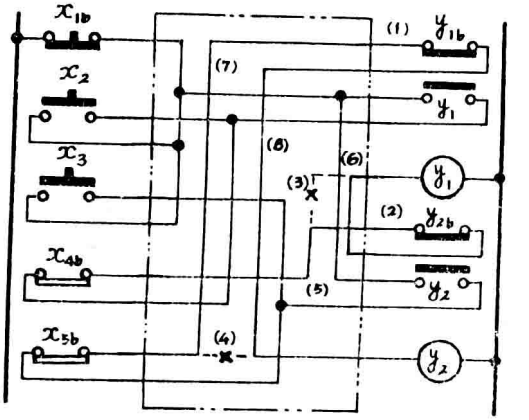


图1.4　工作条件改变后继电器展开图的变形

如果只把图1.1与图1.3加以比较，不禁就会感到只不过是追加了两个接点而已，但如从程序设计的角度来看，则会发现在程序上发生了很大变化。在继电器控制电路中，改变程序是依靠改变布线来实现的，这种固定程序系统的最大缺点是程序变化时缺乏灵活性。

而采用YODIC—S时，则是把图1.2和图1.4的方框内的线路程序，改换成另一种形式的程序。这种程序因为是无形的线路，所以不管工作条件发生多少次变化，都能灵活地更改。这种系统又称为可编程序系统（软接线逻辑系统），即使在控制顺序变化不太多的情况下，使用起来也非常方便，例如可编制系统校验程序，以监视检测端、操作端的动作情况。

根据图1.1和图1.3的继电器展开图编制YODIC—S的程序是极为方便的，差不多只要机械地改写一下就行了。图1.5是根据图1.1编制的程序。图1.6是根据图1.3编制的YODIC—S

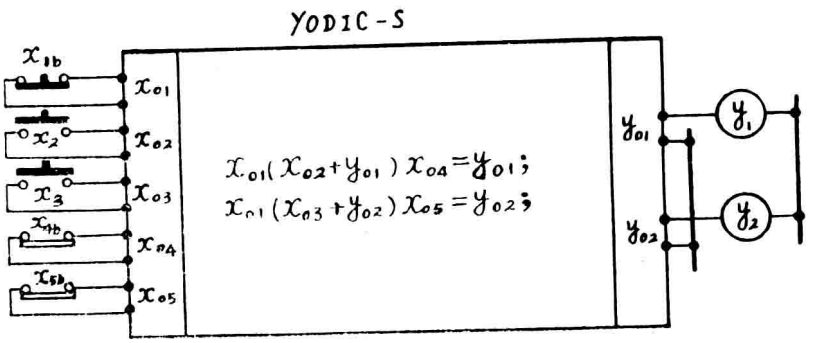


图1.5　YODIC—S的程序（图1.1之例）

程序（输入接点、输出继电器上都附有YODIC—S的输入输出号）。

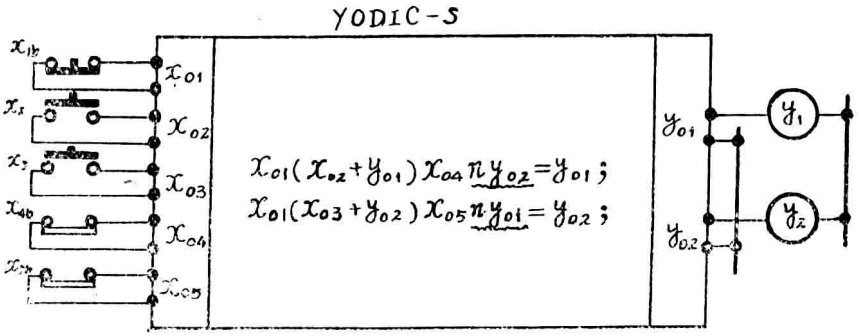


图1.6　YODIC—S的程序（图1.3之例）

特别是从图1.6可明显地看出，要改变程序，仅仅更改方框内的程序即可。另外，在YODIC—S中，输出继电器的状态如何，YODIC—S本身可以记住，所以在编程序时也就不必像图1.2和图1.4的继电器控制电路那样使用输出继电器接点了。

这里值得注意的是：在图1.5和图1.6中采用a、b两个输入接点，这仅仅是为了与继电器控制电路的图1.2和图1.4进行比较，在YODIC—S中不必采用。图1.7是只用a接点编制图1.6的程序之例。可见，在程序设计中可以随意用b接点来代替a接点。

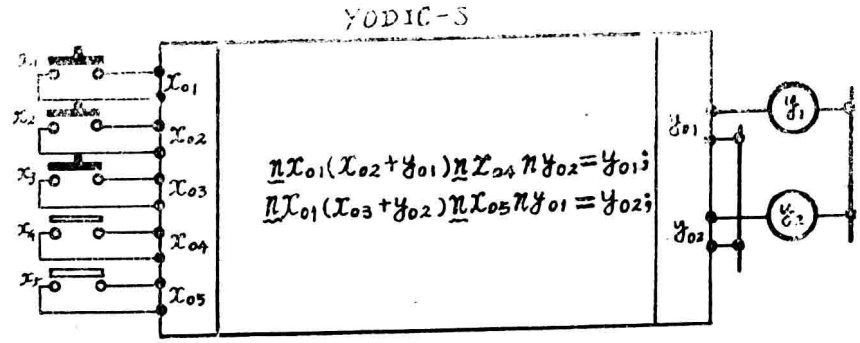


图1.7　工作条件变更后的YODIC—S程序（仅使用a接点时）

以上说明了以继电器控制电路为中心的程序设计概念。还指出了在使用YODIC—S可编程序控制器时，修改程序是极为方便的，而且根据继电器展开图，也容易编程序。

1.2　顺序控制器的制作次序与程序設計的关系

图1.8是继电器控制电路和YODIC—S的制作次序的示意图。

在继电器控制电路的情况下，设计说明书决定后，整理出顺序图表，然后编制继电器展开图和接线图，最后制造继电器控制电路。在这个过程中，编制继电器展开图是程序设计的主要环节，但直到板内接线完成，编程序才算最后结束。

编制继电器展开图需要相当的时间和丰富的经验。因为一旦出了差错，那就要做大量的更正而进行艰苦的修改工作，而且为了按照板内部件的配置，绘制出一目了然的接线图，也是要下不少功夫的。

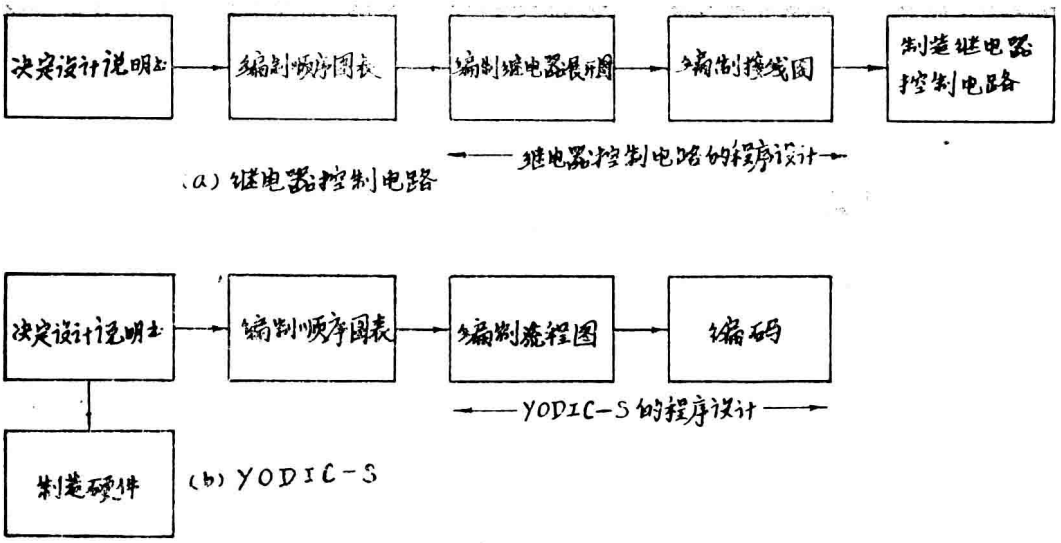


图1.8　顺序控制器的制作次序

在继电器控制电路的制作次序上，程序设计和设备制造是不可分割的，往往程序设计有了变动，设备就要跟着变动。另外，尽管顺序图表能很好地反映出顺序控制中相当重要的时间关系和顺序，但因继电器展开图只能反映出空间接线关系。所以，好容易制成的顺序图表仍然不能充分有效地利用。

而对YODIC—S而言，如图1.8(b)所示，设计说明书一经决定，马上就可以制造硬件。程序设计可以不和硬件的制造发生关系，只要把编好的程序存入YODIC—S的存储器里即可，而且顺序图表一作好就能很容易地完成流程图。按照流程图在规定的格式纸上机械地抄写就完成了编码工作。这种手工作业，是任何人都能完成的。不过，YODIC—S的程序设计是以顺序图表为基础的。所以，在确定设计说明书的阶段，就要重视正确编制顺序图表（关于顺序图表可参阅第七章）。

在比较简单的顺序控制中，甚至无需编制顺序图表，直接编制流程图即可。这时，流程图可用作表示顺序控制内容的图表。如果继电器展开图已作好，并改换成YODIC—S的程序时，那就连流程图也不需要，直接编码就可以了。

1.3　YODIC—S的程序存储

现在简单地说明YODIC—S是用什么方式来存储程序的，但正如第三章所述，由于YODIC—S备有方便的软件系统，在实际的程序设计中，这方面的知识并不是非有不可的。为了说明程序的存储方法，绘制了部分装置的结构图（见图1.9）。程序就存储在磁芯存储器里，磁芯存储器是以4K字（1K字为1024字）为一单元，最大为16K字。将程序存到磁芯存储器中的装置叫做程序输入输出装置。这里有两种装置：一种是程序设定台；另一种是盒式磁带机。

在程序设定台上装有键盘、打字机和文法校验电路。当使用程序设定台输入程序时，需像图1.9的(1)那样，接上电缆（程序总线），然后按编码表上的程序，从头至尾依次按压相对应的键符（如图1.5～图1.7所示的符号或数字），程序就被打字机逐字打印出来。与此同时，每个字符又通过程序总线和运控器存放到磁芯存储器中。运控器在此种场合，起整顿

交通秩序的作用，使每个字符都正确地存到所指定的地址中以及进行语句号的登记工作（详见第三章）。

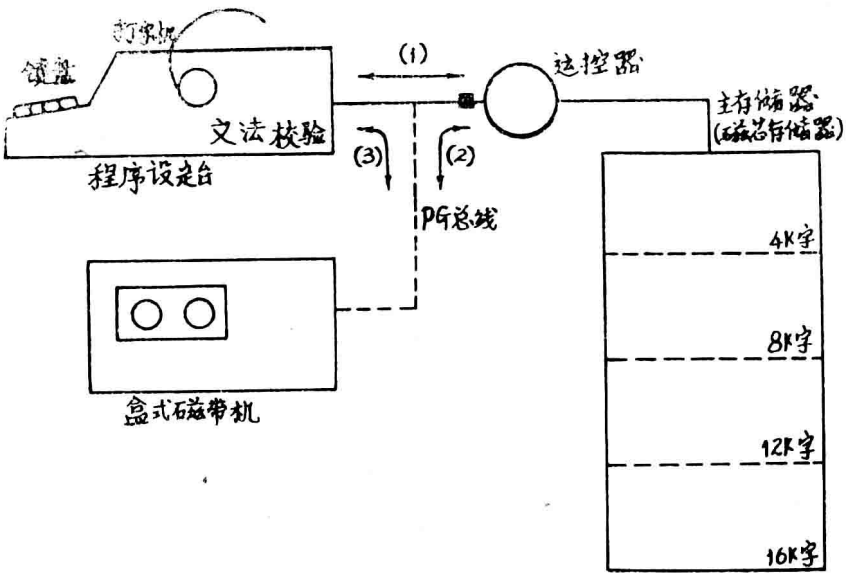


图1.9　程序存储示意图

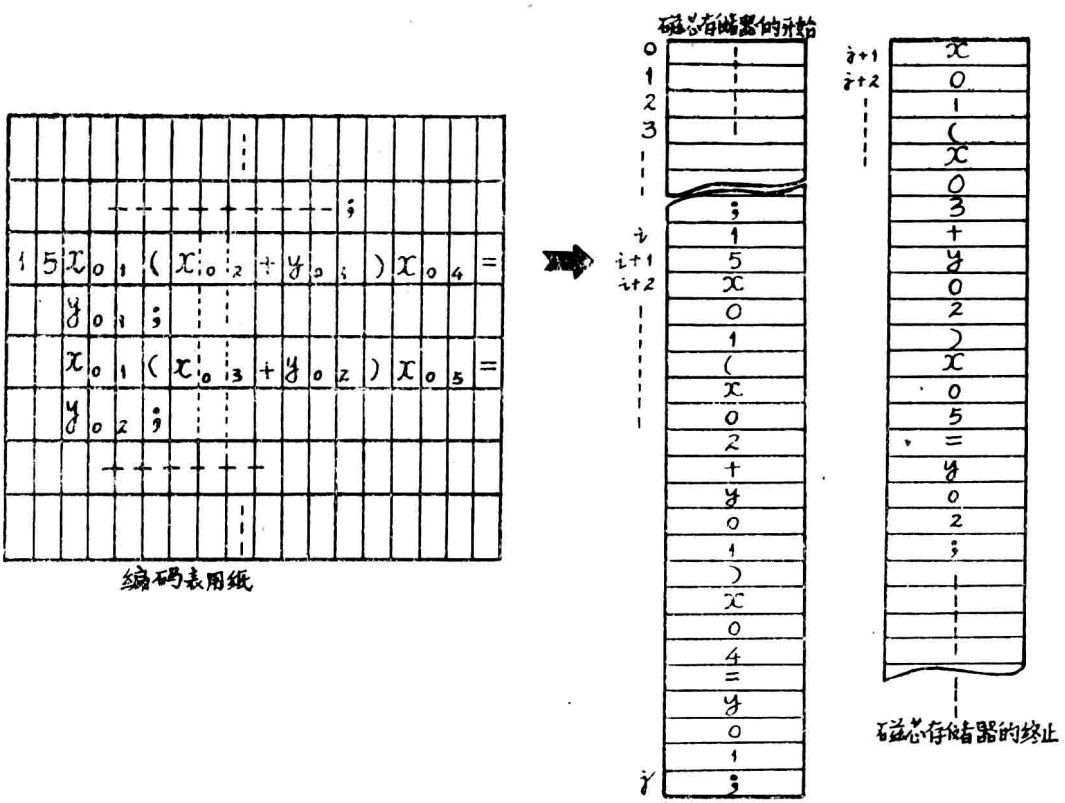


图1.10　编码表用纸和磁芯存储器的存储状态示意图

图1.10是已存有程序的磁芯存储器的存储状态示意图。程序是按编码表的顺序存入磁芯存储器中的，一连串的字符排列成长带状，每个字符依次编上地址号，例如1号地址，2号地址，……。因此，编码表反而成为磁芯存储器的抄本。

如果用盒式磁带机输入程序，就要预先在磁带上存入程序。

程序存入磁带的步骤正如图1.9的(3)那样，把程序设定台的程序总线连到盒式磁带机上，然后，与存入磁芯存储器的情况一样，靠按压键盘上的按钮，把程序逐字录制到磁带上。这时盒式磁带机起整顿交通秩序的作用，使磁带走动，把每个字符存到正确的位置上。此时与运控器和磁芯存储器无关，所以称为离线使用方式。

图1.11是程序被转储在磁带上的情况，可以看出，与磁芯存储器的情况相同。

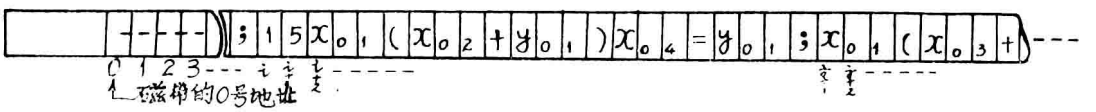


图1.11　磁带的存储

在用键盘输入时，一旦输入文法上有错误的程序，程序设定台的文法校验电路立刻能发现错误，同时能阻止文法上有差错的字被存储下来（参阅第四章）。

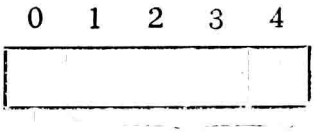
如果要把磁带上存储的程序转储到磁芯存储器中，需像图1.9的(2)那样连接程序总线。操纵盒式磁带机，就可以让磁芯存储器把磁带上的内容存储下来，由于磁带的移动速度很快，所以，能够在很短的时间内完成输入工作。

程序输入输出装置，除上述程序输入工作外，在完成程序的检查和抄写方面，使用起来也很方便。例如需要检查磁芯存储器中的程序时，只要把程序设定台连接起来，操纵键盘，就能通过运控器读出磁芯存储器的内容，并由打字机打印出来。为了录下程序的抄本，只要与盒式磁带机连接，操纵磁带机，就能读出磁芯存储器的内容，并记录在磁带中。这样，在读出时，磁芯存储器的内容不会破坏，在磁芯存储器中仍然保留着程序。也就是说，一次存储下来的程序，只要没有别的程序再一次写进来，就能永远保存，即使切断电源也不会遗失（盒式磁带机也相同）。从这个意义上讲，可以把磁芯存储器的程序看作是同继电器控制电路的接线是一样的。

利用程序输入输出装置还可自由地对程序进行部分的校正和补充（参阅第八章、第九章），详细的操作方法见附录五—4。

最后谈一谈磁芯存储器中字的组成。YODIC—S磁芯存储器的字长为5位。因此，一个字能记忆$2^5=32$种信息，即十六种符号和十六种数字。

0位用来区分符号或数字，1～4位用来记忆$2^4=16$种符号或数字。各个符号和数字的对应代码参见第二章。



符号或数字编码16种

——用于区别符号和数字，“1”表示符号，“0”表示数字。

图1.12　字的组成

1.4 YODIC—S的结构和程序的执行

前一节叙述了编好的程序如何存储在磁芯存储器中。本节介绍记忆好的程序如何使用和执行。

图1.13说明了YODIC—S的信号传送关系。因为程序已被写入，所以图中省略了不需表示的程序输入输出装置。接通电源并按压运控器上的“起动”（START）按钮后，顺序控制就开始执行。

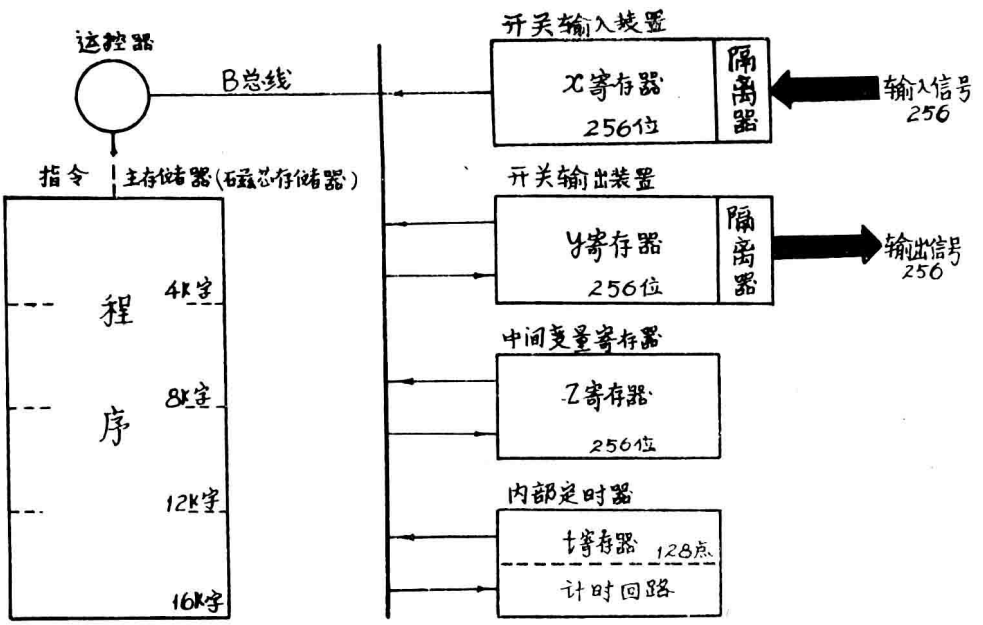


图1.13 YODIC—S的结构及信号传送关系

在说明执行情况之前，先讲一下YODIC—S的结构。从运控器引出B总线（开关输入输出总线），与B总线连接的有开关输入装置、开关输出装置、中间变量寄存器以及内部定时器。

开关输入装置是将被控系统的检测端、控制板等出来的信息，作为输入信号送进YODIC—S中去，大部分输入信号由输入端子引入。

开关输出装置，恰与开关输入装置相反，它把YODIC—S的信号输出，从输出端子出来的多数输出信号都通到操作面板，其中，一部分信号通过操作面板的处理就形成驱动被控对象操作端的操作信号，而另一部分信号则作为显示之用。

中间变量寄存器是在YODIC—S内部记忆控制信号用的，它能够记忆程序的操作步骤等等。

内部定时器是半固定式的定时器。它与简单的钟表或延时继电器等相当，是同装在操作面板上的非固定式定时器有区别的。操作面板上的定时器叫做外部定时器，而YODIC—S的定时器叫做内部定时器。后一种定时器是用程序控制的。所以，没有输入输出端子，但只要接上B总线就能使用（参阅第六章）。

上述各装置都有记忆信息的功能，中间变量寄存器本来就是一种记忆装置，又称为z寄存器。在开关输入装置中也有一个记忆最新输入信号的输入寄存器(参阅第二章监控程序)，

称为x寄存器。在开关输出装置中同样也有一个记忆和保持输出信号的输出寄存器，称为y寄存器，在内部定时器中也有记忆定时器正在动作或复位的寄存器，称为t寄存器。y寄存器与z寄存器是完全相同的，不过，z寄存器可以认为是没有输出电路（隔离器、输出端子等）的开关输出装置。

执行顺序控制的核心是运控器，它可通过B总线与各装置连接，并按各装置记忆的控制信息进行顺序控制。进行这种控制的最终目的是为了把输出信号非常正确地按照顺序发出，并正确地、有次序地改写y寄存器的内容。在这里所以要把运控器与各装置间的信息通路叫做B总线（Bit总线），是因为各种装置都能记忆用“0”或“1”表示的1位二进制信息并能取送1位二进制信息。

图1.13中，磁芯存储器记忆的程序就是指示运控器应该进行何种控制的指令。因此，在进行顺序控制时，运控器不记录指令的内容(即程序)，而是读出程序，并按指令进行控制。

现以简单的电动机起动和停止的自保程序为例来说明程序的执行情况。如用继电器展开图表现程序，将如图1.14，但在YODIC—S中若只用a接点输入时，程序可写为：

$nx_{02}\ y_{01}+x_{01}=y_{01}$;

另外，在程序执行前输入输出装置所存内容如为：

x寄存器　x_{01}为“1”（按压“起动”按钮x_1）

　　x_{02}为“0”（按压“停止”按钮x_{2b}）

y寄存器　y_{01}为“0”（电动机y_1正在停止状态）

则运控器，将按下列顺序，从头开始依次读出程序并进行顺序控制。

（1）读出nx_{02}（n意味着信息取反码）。

（2）从x寄存器读出x_{02}的值(“0”)，并取其反码，然后把所得结果“1”存入运控器本身的寄存器（假定为A寄存器）中，则A=“1”。

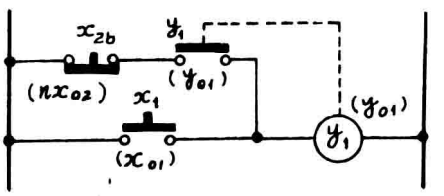


图1.14　自保电路

（3）读出程序y_{01}。

（4）从y寄存器读出y_{01}的值(“0”)，同A=“1”进行运算（在这种运算中，只是等号左、右两边都是“1”时，运算结果才等于“1”），然后把得出的结果“0”记忆在A寄存器里，则A=“0”。

（5）读出“$+x_{01}$”。

（6）从x寄存器读出x_{01}的值(“1”)，与A=“0”进行运算（+运算意味着不论哪一边是“1”，其结果都为“1”），然后把得出的结果“1”，记忆在A寄存器里，则A=“1”。

（7）读出“$=y_{01}$;”。

（8）把A寄存器的结果“1”，写入y寄存器的y_{01}中。

按以上程序执行后（执行时间很短，见附录四），输出y_{01}则从“0”变成“1”。因此，原处于停止状态的电动机y_1就开始动作起来了。形成这种情况是由于按压了“起动”按钮x（即x_{01}为“1”）的缘故。

运控器在执行完一次上述程序后，又执行其他程序，然后再次执行上述程序。这样做是因为程序是按循环处理原则编制的（见第二章2.2节）。

执行到第二次以后，即使不按压“起动”按钮x_1(x_{01})，已经起动起来的电动机y_1将自

动地继续工作下去。这是因为：

x寄存器　x_{01}为“0”（不按压“起动”按钮x_1）

x_{02}为“0”（不按压“停止”按钮x_2）

y寄存器　y_{01}为“1”（电动机y_1正在运转中）

因此，上述执行顺序中的（4）和（6）的情况如下：

（4）y_{01}的值是“1”，所以A＝“1”。

（6）x_{01}的值即使是“0”，仍然是A＝“1”，运算结果仍然是“1”，y_{01}保持“1”状态。

上述程序执行到若干次以后，在顺序控制中按压“停止”按钮x_{21}时，这一次的执行结果就变成“1”，电动机y_{01}进入停止状态。也就是：

x寄存器　x_{01}为“0”（不按压“起动”按钮）

x_{02}为“1”（按压“停止”按钮x_{21}）

y寄存器　y_{01}为“1”（电动机y_1运转）

在执行这样的顺序时，（2）、（4）、（6）就要变成如下情况：

（2）因x_{02}的值为“1”，取反码后变成“0”，A＝“0”。

（4）即使y_{01}的值为“1”，因A＝“0”，所以，运算结果为“0”，A仍等于“0”。

（6）因为x_{01}的值为“0”，A寄存器也为“0”，所以，运算结果仍是“0”，A＝“0”。

其结果，（8）中y_{01}从“1”变成“0”。

以上说明了编好的程序是如何执行的。至于YODIC—S的系统结构、操作面板的内容以及硬件的详细情况可参阅另外的资料（五—1、五—5等）。

第二章　语句说明

所谓编程序就是用机器能够理解的语言编写文章，从而使机器进行工作。YODIC—S是用语句记述程序的。按语句的种类，每一种都有其特定的书写规定，但也有不分语句种类，对编制任何程序都是通用的规定。本章将说明这种共同的规定。此外，本章也说明关于编程序必须了解的基本事项。下章将说明各种语句的规定和关于语句的注意事项。

为了有效地进行顺序控制，本章首先说明YODIC—S可编程序控制器所具有的功能和编程序用到的各种语句之间的关系。

2.1 程序設計和硬件功能

顺序控制的特点是所处理的信息都是用“1”和“0”表示的二进制信息，与模拟量控制的情况有很大差别。具有这种特点的装置可有效地进行顺序控制。在这个意义上讲，数字电子计算机被认为是适合顺序控制的，可是目前的通用计算机在顺序控制方面，有许多不必要的指令和功能，未免大材小用。此外，编程序要求具有一定的知识，在经济上也不合算等。

为了有效地进行控制，顺序控制器至少必须具有以下功能：

（1）驱动操作端的输出功能；

（2）接受检测端信号的输入功能；

（3）逻辑运算功能；

（4）判断分支功能；

（5）信息存储功能；

（6）计时功能。

YODIC—S就是利用计算机技术建立起来而又克服了通用计算机缺点的一种顺序控制器。它不仅具有本章说明的必不可少的功能，而且使用简便，编程序容易。下面从编程序的角度说明上述功能。

（1）输出功能

顺序控制的最终目的是驱动操作端，所以我们把输出功能列为第一个最基本的功能，无论哪一种顺序控制器，不能没有输出功能。例如简单的转鼓式顺序控制器起码备有输出功能和计时功能。在YODIC—S中是由开关输出装置执行这种功能的。

从编程序方面考虑，尽管也有输出逻辑运算结果的装置（硬件），但这样的装置往往大大地限制了输出功能，而给编程序带来困难。因此，必须有一套无条件地自由输出的软件系统。

YODIC—S除了有输出逻辑运算结果的逻辑语句以外，还有设定语句。在顺序图表中要设定的输出用○表示，要复位的输出用×表示。YODIC—S是利用设定语句把输出设定为○或×的。

（2）输入功能

如不检测输入条件，有时很难决定输出是○，还是×。因此，必须有接收控制对象状态

（检测端的信号）的功能。YODIC—S是由开关输入装置来执行这种功能的。

（3）逻辑运算功能

接收的输入信息需与其它控制信息进行逻辑运算，以便按照运算结果决定输出。YODIC—S是由运控器进行逻辑运算的。在程序上使用逻辑语句。逻辑运算功能不单单是将运算结果输出，而且还要把运算结果作为判断条件，以变更控制内容等，这样就能制成动态的程序。YODIC—S还可以用条件转移语句写出与逻辑语句相同的逻辑式，作为判断条件。

（4）判断分支功能

在顺序控制中判断分支功能也是极为重要的，是编程序中不可缺少的一种功能。YODIC—S是用条件转移语句来执行这种功能的。

由条件转移语句分支后的程序，必需再汇合，所以还备有无条件转移语句。

（5）信息存储功能

在顺序控制中不能单凭现在的输入组合决定输出，往往还要参考过去的状态和现在输出的状态。为了存储这种一次消失的过去的输入输出信息和现在怎样进行操作的操作信息，在YODIC—S中备有z寄存器。此外，开关输出装置还有保存现在的输出信息的功能。输出一旦设定为○，就可保持○状态，直到用程序复位（×状态）为止。

保持这类输出的y寄存器，在YODIC—S接通电源时，马上自动清零并返回到初态。

（6）计时功能

掌握时间经过，对于顺序控制也是很重要的。为了明确时间的前后关系经常使用各式各样的计时元件，从微小延时的延时继电器起到准确计时的时钟为止，应有尽有。

采用YODIC—S时，操作人员完全不需要经常计时，而是由装在机内的定时器来执行顺序控制的计时功能。

内部定时器也是在通电时自动地清零复位。

上面说明了顺序控制的功能，但仅备有必要的功能并不意味着容易编程序。为使程序设计简化，在设计YODIC—S时特别注意了以下几点：

（1）不了解硬件也能编程序；

（2）没有计算机知识也能编程序；

（3）用与日常语言相近的语句编程序；

（4）不需要特别的编译程序，机器即可接受。

2.2 語句

YODIC—S的程序同日常会话一样，以完整的句子作为程序的最小单位，并把它叫做语句。下面列举语句的特点和规定：

（1）语句以符号开始（也有以语句号开始的）；

（2）语句以终止符号（；）结束；

（3）必要时，可用语句号来表示语句的名称；语句号（简称为ST．NO）作为语句的名字；

（4）语句号写在语句的前面；

（5）语句的内容用符号和数字书写；

（6）除特定的语句外，语句的长度不定。

语句的形式示于图2.1。

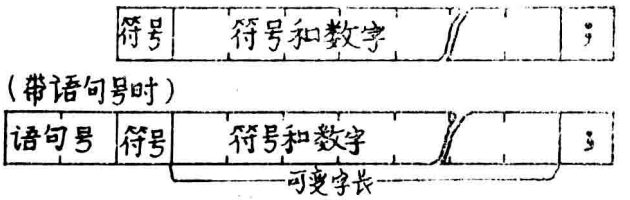


图2.1 语句的形式

语句的写法如下。

例：$x_{01}+z_{10}=y_{02}$;

$Sy_{01}\ y_{02}\ n(y_{03}\ z_{11})$;

F03 x_{23};

G01;

$D_{000000000000}$;

下面列举用语句编写程序时的一些特点。

(1)在语句和语句间，除语句以外不写任何东西，而且也不需留有空白（程序是由一系列连续不断的语句所构成）。

(2)程序由磁芯存储器来存储，而且从存储器的开头往下存放。

(3)程序的长度不定（可变长度）。

但存储容量需在16K字以内。

(4)程序上的每一符号和数字，都作为一个字。

在程序编完之前，要知道占用多少存储单元是很困难的，只能根据顺序控制的操作步数和输出点数等系统规模，大致估算所需字数。编程序时原则上应考虑容易看懂。程序完成后再检查所占用的单元数，必要时应设法节省占用的内存单元数量（参看第十一章）。

程序的形式如图2.2所示。

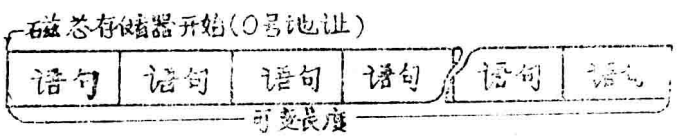


图2.2 程序的形式

程序可以分成若干小程序，其最小单位就是语句。但程序也不单是语句的罗列，如果只是语句的拚凑，而不产生意义的话，纵然有完整的形式，也不能称为程序。

程序的简例如下：

例： $D_{000000000000}$;

01 F01 nz_{05};

$Sz_{03}\ z_{04}$;

$nx_{02}\ y_{01}+x_{01}=y_{01}$;

G01;

02

2.3 語句号

语句号在程序中用于程序的分支和汇合。

此外，在各种操作中如把程序用打字机打印出来时也要使用。语句号是在程序中把特定的语句定一个名字。预先写在语句的前面。语句号的使用规定如下。

（1）语句号所使用的字数

语句号用十六进制2位数的数字表示（参考附录一）。

（2）语句号００禁止使用。

（3）语句号０１写在开始执行的语句之前。可供任意使用的语句号共有254个。

（4）终端语句号

为了增补程序和进行各种操作，在程序的终端必须写上终端语句号。

（5）同一语句号不得重复使用。

（6）语句号的使用顺序是自由的，但一般从小的语句号开始使用为好。

用语句号表示的程序如下：

```
    D0000000000000000;
01  F01 nz05;                    (开始执行语句号)
    Sz03 z04;
    F02 x01;                     (分支语句号)
    x02 = y01;
    G01;                         (汇合语句号)
02  x03 = y01;                   (转移处所语句号)
    G01;                         (汇合语句号)
03                               (终端语句号)
```

2.4 語句号区域

下面说明语句号区域的设定。

在YODIC—S的程序中需要使用语句号，所以，磁芯存储器的地址号就失去了一切意义。而自由命名的语句号却起程序地址的作用。

另一方面，从机器本身来看，程序还要由磁芯存储器来存储，所以，要参照磁芯存储器的地址，进行程序的分支和汇合。因此，在程序写入的时候，机器本身能自动地查出每个语句号存储在磁芯存储器的哪个地址中，并作出语句号和磁芯存储器地址的对照表。该对照表就是语句号表，编制对照表的工作就称为语句号的登记工作。

像这样得到的语句号表也必须在某处存储，以便随时能参考。语句号表的存储处所称为语句号区域，它存在磁芯存储器的开始部分。使用的语句号小时，所占用的区域就小。

在编程序时，首先就要设定这个语句号区域，一般从简单的程序开始写。这种语句号区域的程序如下进行编码。

（1）语句号区域的设定程序

D0000000000……00；或者

；D0000000000……00；（参考注4）

（2）数字“0”的数目

设使用语句号的最大号码为Nmax，则数字“0”的数目＝（4Nmax＋3个以上）。