

YODICS

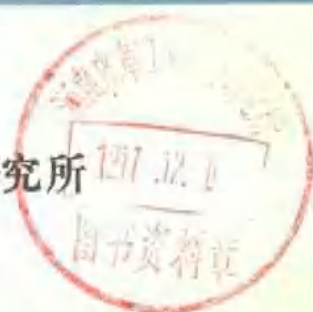
可编程序控制器

工作原理



第一机械工业部机械研究院机电研究所

一九七五年九月



YODIC-S 可 编 程 序 控 制 器

工 作 原 理

参 考 资 料 之 二

前 言

可编程序控制器是最近几年发展起来的一种新型顺序控制装置，在国外称为可编程序逻辑控制器（PLC）、可编程序顺序控制器（PSC）或可编程序矩阵控制器（PMC）。它是利用小型计算机的一些原理和技术，设计制造而成的，但却比通常的小型计算机结构简单、编程容易、价格低廉。因此，目前已在外国各种自动生产线、水火电站中的一些操作子回路和自动立体仓库等方面得到了应用。

为了配合机械工业工厂技术改造，不断提高自动化水平，我们遵照毛主席关于“洋为中用”的教导，翻译出版了《YODIC-S可编程序控制器工作原理》，供有关同志工作中参考。为了方便读者，在资料最后附有YODIC-S逻辑符号说明、固体组件图表和主要英文缩写符号表。关于可编程序控制器的一般应用情况请参阅我所编译出版的《可编程序控制器及其应用》。

由于水平有限，加之时间仓促，文内难免有错误和不当之处，欢迎读者批评指正。

本资料的翻译出版得到了天津电气传动设计研究所、上海工业自动化仪表研究所、上海电器科学研究所和重庆工业自动化仪表研究所的大力支持，在此表示深切谢意。

编 者

毛 主 席 语 录

鼓足干劲，力争上游，多快好省地建设社会主义。

对于外国文化，排外主义的方针是错误的，应当尽量吸收进步的外国文化，以为发展中国新文化的借镜；盲目搬用的方针也是错误的，应当以中国人民的实际需要为基础，批判地吸收外国文化。

学习外国的东西，是为了研究和发展中国的东西。

目 录

第一篇 中央处理装置 (CPU)

第一章 CPU 概述	(1)
第二章 控制器	(6)
第三章 运算器	(51)
第四章 中间变数存储器	(63)
第五章 主存储器	(71)
第六章 操作面板	(82)
第七章 位总线 (B总线)	(85)
第八章 程序总线 (PG总线)	(89)
第九章 主机架电源	(92)

第二篇 输入输出装置 (BIO)

第一章 BIO概述	(101)
第二章 开关输入装置	(103)
第三章 开关输出装置	(123)
第四章 内部定时器	(133)

第三篇 程序输入输出装置 (PGIO)

第一章 PGIO 概述	(148)
第二章 程序设定台	(149)
第三章 盒式磁带	(180)

附录一 YODIC-S 逻辑符号说明	(215)
--------------------------	-------

附录二 YODIC-S 固体组件图表	(229)
--------------------------	-------

附录三 YODIC-S 主要英文缩写符号表	(246)
-----------------------------	-------

第一篇 中央处理装置(CPU)

第一章 CPU 概述

1.1 YODIC-S系统结构

YODIC-S型可编程序控制器,按功能可分为中央处理装置(以下简称CPU)、输入输出装置(以下简称BIO)和程序输入输出装置(以下简称PGIO)三大部分。

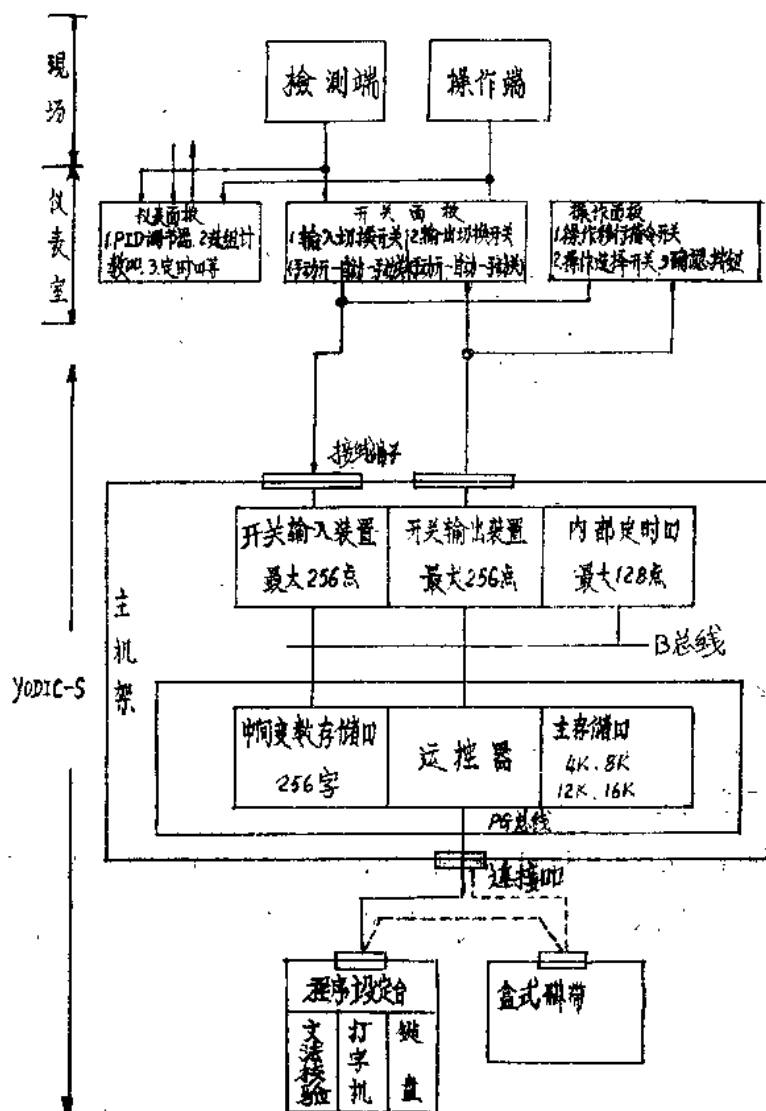


图1.1 YODIC-S的系统结构图

BIO包括开关输入装置(x)、开关输出装置(y)、内部定时器(t)和中间变数存储器(z)。
PGIO包括程序设定台(PC)和盒式磁带(CT)。

YODIC-S型可编程序控制器,按结构则可分为主机架、程序设定台和盒式磁带三部分。

在主机架中装有CPU、BIO和主机架电源。

YODIC-S的系统结构如图1.1。

CPU与BIO之间以及CPU与PGIO之间固然可以传送各种信号,但前者尚需使用位总线(以下简称B总线),后者尚需使用程序总线(以下简称PG总线或P总线)。

1.2 中央处理装置的结构

1.2.1 CPU的结构

CPU的方块图示于图1.2。

CPU(CENTRAL PROCESSING UNIT)系由控制器、运算器和主存储器构成。

控制器用来产生控制CPU内部的一系列控制信号。它设有起停控制电路、时钟脉冲发生器和只读存储器等。

在YODIC-S中,控制CPU内部的一系列控制信号需作成微程序(微程序是由一条条的微指令构成)。为此,采用了指定地址和读出固定数据用的只读存储器(控制存储器)。

在执行逻辑运算的运算器中,设有指挥和控制整个运算器的逻辑运算控制器(LU控制器)、暂存语句符号的状态寄存器、进行逻辑运算的逻辑运算电路以及与BIO之间传送信息的电路。

主存储器用来记忆顺序控制程序。它的性能指标如下。

主存储器种类:磁芯存储器

存储容量:4K字(4096字)/模块(插件) 最大16K字(按4K字为1单位扩展)

字长:5位

存取周期:2微秒

存储保护:有通停电存储保护电路

1.2.2 语句及其符号

YODIC-S的语句包括逻辑语句、控制语句、输入输出语句、数据语句以及其它语句。

语句一览表示于表1.1,符号和数字位的构成示于表1.2,运算执行时间示于表1.3。

表1.1 语句

语句种类	形 式	符号	备 注
逻辑语句	左边 右边 $f(x, y, z, t) = v$; f : 逻辑式 v : y, z 或 t	n	否定
		x	输入变量 (开关输入)
		y	输出变量 (开关输出)
		z	中间变数
		t	内部定时器变量
		$+$	“或”运算 (“与” 运算无符号)
		$($	左括号
		$)$	右括号
控制语句	GN ; N : 语句号	G	GO TO语句
			无条件转移到语句号 N
	$FNf(x, y, z, t)$; N : 语句号 f : 逻辑式	F	IF语句 逻辑式 f 的结果为“1”时跳到语句号 N ; 其结果为“0”时, 则执行下一个语句
输入输出语句	$SV_1V_2\cdots$ $n(V_m V_{m+1}\cdots)$; v : y, z (变量)	S	SET语句 n 左侧的变量设定为“1”, 右侧的变量复位为“0”
	$E\cdots\cdots$; (未使用)	E	使用通道时的语句
数据语句	$D00\cdots 0$; 0 : 数字0	D	确保空白区域, 以备增加程序之用; 空白区域由“0”的数目+2字构成
其它语句	$;$ (分号)	$;$	语句的终止符号
	$*$	$*$	空操作

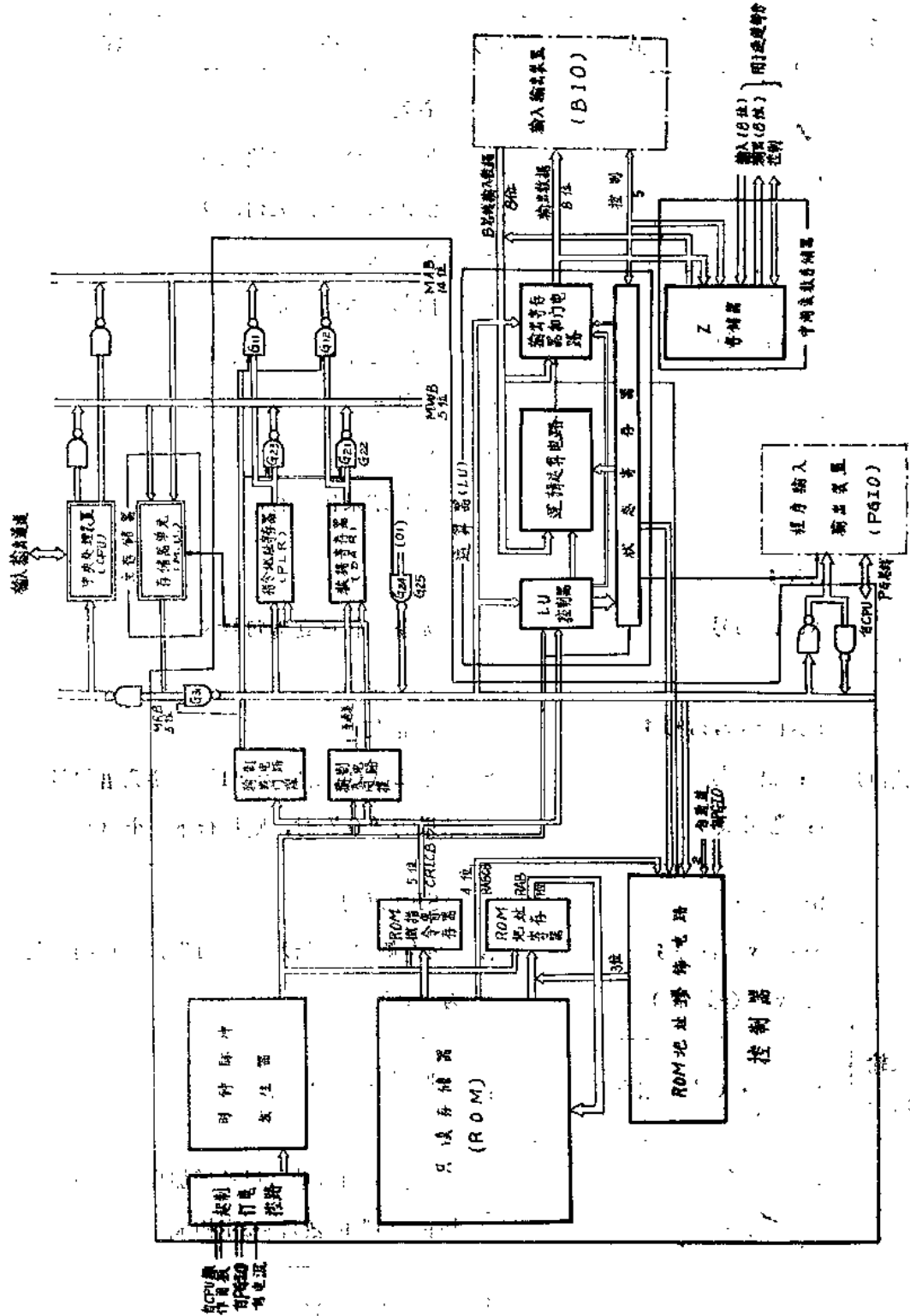


图1.2 中央处理装置方块图

表1.2 符号和数字位的构成

符 号	内 容	位 的 构 成	数 字	十 进 制 数	位 的 构 成
		0 1 2 3 4			0 1 2 3 4
t	内部定时器变量	1 0 0 0 0	0	0	0 0 0 0 0
x	输入变量	1 0 0 0 1	1	1	0 0 0 0 1
y	输出变量	1 0 0 1 0	2	2	0 0 0 1 0
z	中间变数	1 0 0 1 1	3	3	0 0 0 1 1
G	无条件转移	1 0 1 0 0	4	4	0 0 1 0 0
F	条件转移	1 0 1 0 1	5	5	0 0 1 0 1
E	未定义	1 0 1 1 0	6	6	0 0 1 1 0
D	数据	1 0 1 1 1	7	7	0 0 1 1 1
n	否定	1 1 0 0 0	8	8	0 1 0 0 0
+	“或”运算	1 1 0 0 1	9	9	0 1 0 0 1
(	左括号	1 1 0 1 0	A	10	0 1 0 1 0
)	右括号	1 1 0 1 1	B	11	0 1 0 1 1
=	代入	1 1 1 0 0	C	12	0 1 1 0 0
;	终止符号	1 1 1 0 1	D	13	0 1 1 0 1
S	设定	1 1 1 1 0	E	14	0 1 1 1 0
*	空操作	1 1 1 1 1	F	15	0 1 1 1 1

表1.3 运算执行时间

分 类			运算执行时间(理论值, 微秒)
1	逻辑式		$12.5 \times (\text{变量的数目}) + 2.5 \times (+ \text{和 } n \text{ 的数目}) + 6.0 \times (\text{左右括号的数目})$
2	逻辑语句	右边无n	逻辑式的执行时间 + 21.0
		右边有n	“ ” “ ” “ ” “ ” “ ” “ ” + 23.5
3	设定语句		$14.0 \times (\text{变量的数目}) + 2.5 \times (n \text{ 的数目}) + 6.0 \times (\text{左右括号的数目}) + 7.0$
4	条件转移语句	运算结果为“0”	逻辑式的执行时间 + 13.0
		“ ” “ ” “ ” “ ” “ ” “ ” “ ” “ 1 ”	“ ” “ ” “ ” “ ” “ ” “ ” “ ” + 21.0
5	无条件转移语句		20.5
6	* (空操作)		2.0/1字
7	语句号		5.5

续表1.3

8	仅;的语句	4.5
9	平均运算执行时间	推算值 4.0/1字

注：包含主存储器的读出时间。

	运算执行时间计算例子	字数	理论值 (微秒)	推算值** (微秒)
1	$x_{01} = y_{01};$	8	33.5	32
2	$x_{01} + nx_{02}y_{02} = y_{02};$	16	63.5	64
3	$x_{01}x_{02}(nx_{03} + x_{04}x_{05}x_{06}nx_{07}) = ny_{03};$	32	124.5	128
4	$Sy_{01}ny_{02};$	9	37.5	36
5	$Sy_{01}y_{02}y_{03}n(y_{04}y_{05}y_{06});$	23	99.5	92
6	$F_{04}x_{01};$	7	25.5	28
	运算结果为“0” 运算结果为“1”		33.5	
7	$F_{02}n(x_{01}x_{02} + nx_{01}nx_{02});$	22	79.0	88
	运算结果为“0” 运算结果为“1”		87.0	
8	$01F_{01}nz_{06};$	10	33.5	40
	运算结果为“0” 运算结果为“1”		31.5	
9	$G_{01};$	4	20.5	16

**——推算值等于平均运算执行时间（4.0微秒）×字数。

第二章 控制器

2.1 控制器结构

控制器是由进行起停控制和产生基准时钟脉冲的时间控制电路、形成各种控制信号用的只读存储器的控制电路以及主存储器传送信号用的信号传送电路等构成。

2.2 时间控制电路

时间控制电路及其时间波形分别示于图2.1和图2.2。

时间控制电路系由时钟脉冲电路、起动控制电路、复位控制电路、工作状态控制电路以及报警接点输出电路构成。

2.2.1 时钟脉冲电路

基本的时钟脉冲是由8兆周的晶体振子产生，其周期为125毫微秒，经二次分频后产生

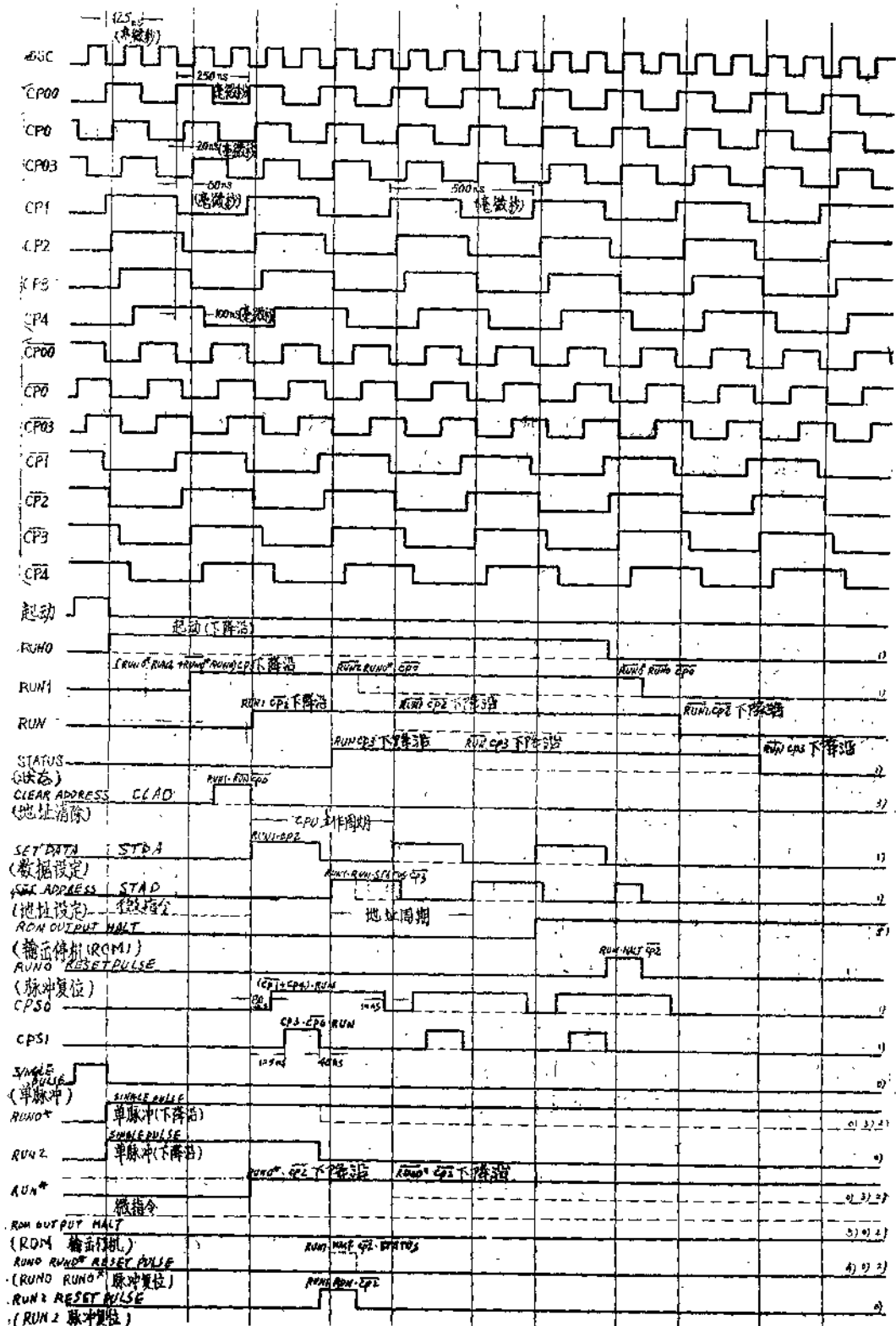


图 2.2 控制时间波形

- 1) 表示单脉冲时的波形, 1) 的虚线表示单脉冲时; 2) 的虚线表示行机; 3) 的实线表示第一个单脉冲, 虚线表示第二个以后的单脉冲;
4) 表示同一回路的信号; 5) 的虚线表示启动时即是行机时。

周期为250毫微秒的 CP_0 波形和周期为500毫微秒的 CP_1 波形。在500毫微秒周期中, 还有比 CP_1 延迟20毫微秒的 CP_2 、延迟60毫微秒的 CP_3 和延迟100毫微秒的 CP_4 时间波形。

YODIC-S的微指令工作周期为500毫微秒。

2.2.2 起动控制电路

起动控制电路是通过按下CPU操作面板上的起动 (START) 按钮或按照PGIO的起动指令开始工作。该电路可由以下条件式表达:

$$\overline{CS}_0 \cdot START + \overline{CS}_0 \cdot ONSRV + SINGLE \cdot START + SINGLE \cdot CHECK$$

条件式中的第1项“ $\overline{CS}_0 \cdot START$ ”, 表示未与PGIO连接, 按下操作面板上的起动按钮后立即开始工作。 $\overline{CS}_0$ 信号, 当与PGIO连接时为“1”。由此可见, 在与PGIO连接时, 尽管按下操作面板上的起动按钮也无效。

第2项“ $\overline{CS}_0 \cdot ONSRV$ ”表示与PGIO连接, 需按照PGIO的指令起动。其中ONSRV表示操作中。

第3项“ $SINGLE \cdot START$ ”为单脉冲起动; 第4项“ $SINGLE \cdot CHECK$ ”为单脉冲校验。两项都表示在调试时用单脉冲来驱动, 调试完毕后不能使用。

在“单脉冲”方式时, 每按一次起动按钮或者给出一个脉冲 (例如脉冲发生器) 只能执行一条微指令。

在起动控制电路中, 还包含一个能够使上述信号与 CP_0 信号同步的输出电路, 其输出由两个触发器产生。

起动控制电路的时间波形示于图2.3。

2.2.3 复位控制电路

复位控制电路是由CPU完全恢复到初始状态的INZRS电路和BIO恢复到初始状态的BCLC电路所组成。

INZRS电路在满足下列条件时有输出:

$$INZRS = IRS + \overline{CS}_0 \cdot RGSF \cdot RESET + \overline{RUN} \cdot RESET + \overline{RUN} \cdot PRST$$

条件式中的第1项“ IRS ”系指由主机架电源发出的“初始状态复位”信号。在电源接通或断开时发出此信号。

第2项“ $\overline{CS}_0 \cdot RGSF \cdot RESET$ ”表示未与PGIO连接, 在工作中按下操作面板上的复位 (RESET) 按钮时, 便执行到语句的终止符号“;”, 然后返回到初始状态。

但与PGIO连接时, 按下操作面板上的复位按钮无效。

第3项“ $\overline{RUN} \cdot RESET$ ”表示在停机时按下复位按钮后便立即恢复到初始状态。

第4项“ $\overline{RUN} \cdot PRST$ ”表示一旦与PGIO连接, 立即按PGIO的复位信号恢复到初始状态。

使BIO复位的BCLC信号系由主机架电源发出的“电源准备”信号 (PRS) 产生。

2.2.4 工作状态控制电路

按下起动按钮并起动后, 表示CPU工作的运行触发器和状态触发器立即工作。

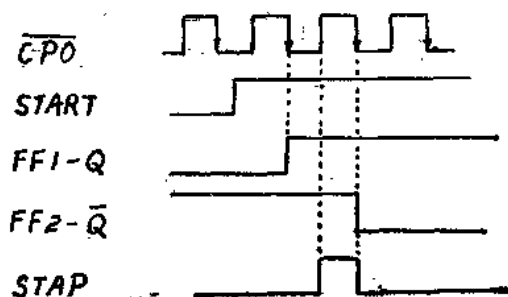


图2.3 起动控制电路的时间波形

下面示出各触发器的设定和复位条件。但由于RUN2、RUN0*和RUN*等运行触发器是在单脉冲时工作的，故此处从略。

设定 $RUN0 = STAP \cdot \overline{SINGL}$

复位 $RUN0 = INZRS + STATUS \cdot RUN1 \cdot HALT \cdot \overline{CP_2}^{**}$

RUN0触发器是在产生起动信号时设定，而在INZRS时和在工作中执行停机微指令时复位。STAP—表示步进。

设定 $RUN1 = RUN0 \cdot \overline{RUN0^*} \cdot CP_3 + SINGL$

复位 $RUN1 = \overline{RUN0} \cdot \overline{RUN0^*} \cdot CP_0 + SINGL$

设定 $RUN = RUN1 \cdot \overline{CP_2}$

复位 $RUN = \overline{RUN1} \cdot \overline{CP_2} + INZRS$

设定 $STATUS = RUN \cdot CP_3$

复位 $STATUS = \overline{RUN} \cdot CP_3 + INZRS$

利用RUN运行触发器和STATUS状态触发器可产生各种信号。

$CPS_0 = RUN \cdot (CP_1 + CP_4)$

CPS₀是产生微指令的时钟脉冲。

$CPS_1 = RUN \cdot CP_0 \cdot CP_3$

$STDA = RUN \cdot \overline{CP_3}$

STDA是只读存储器（以下简称ROM）的微指令寄存器设定信号。

$STAD = RUN1 \cdot RUN \cdot STATUS \cdot \overline{CP_3}$

STAD是ROM地址寄存器的设定信号。

$CLAD = RUN1 \cdot \overline{RUN} \cdot \overline{RUN^*} \cdot CP_0$

CLAD是ROM地址寄存器的复位信号（在起动后ROM地址寄存器复位）。

$EXEC = \overline{CS_0} \cdot RUN0$

EXEC是表示未与PGIO连接，而是正在执行中的信号。

2.2.5 报警接点输出电路

在用YODIC-S进行控制时，如果BIO的某局部不能工作，或者CPU事故停机，或者主机架电源发生异常等等均需报警，为此采用了晶体管接点输出电路（异常时接点断开）。

报警接点输出电路示于图2.4。

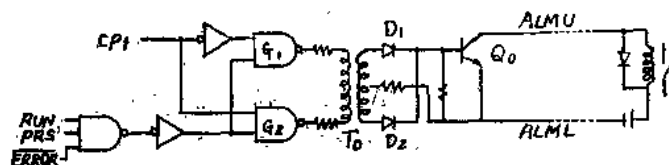


图2.4 报警接点输出电路

如上图所示，当满足 $RUN \cdot PRS \cdot ERROR$ 条件时（ $ERROR = BIOR$ ，即BIO处于可以工作的状态），G₁和G₂按照CP₁信号交替接通，驱动T₀隔离脉冲变压器，在T₀的次级侧（二次侧）产生感应交流电压。该电压经D₁和D₂二极管进行全波整流后送至Q₀晶体管。

**——原文中的CP₂系为 $\overline{CP_2}$ 之误，已改正（译注）。

当 Q_0 有基极电流流过时,立即处于接通状态。

在发生异常时,不能满足 $RUN \cdot PRS \cdot ERROR$ 条件。这时 T_0 次级侧的感应电流消失, Q_0 处于断开状态。

G_1 、 G_2 和 T_0 一次侧(初级侧)的详细线路示于图2.5。

2.3 只读存储器及其外围电路

2.3.1 ROM

ROM是不能自动写入而专用于读出的固定存储器(JIS C6230~1970)。

在YODIC-S中采用了四个64字 $\times$ 8位(8位/1字,共64字)的ROM,用来构成128字 $\times$ 16位。

首先说明64字 $\times$ 8位的ROM。

ROM的方块图示于图2.6。

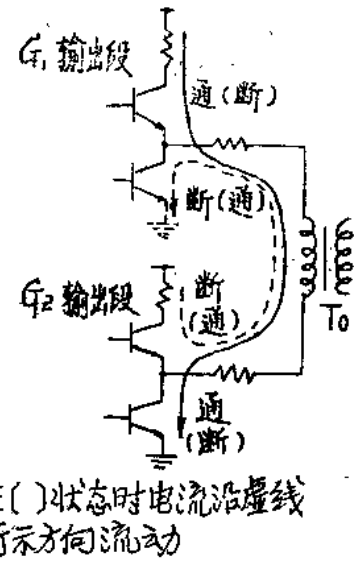


图2.5 G_1 、 G_2 和 T_0 一次侧的线路图

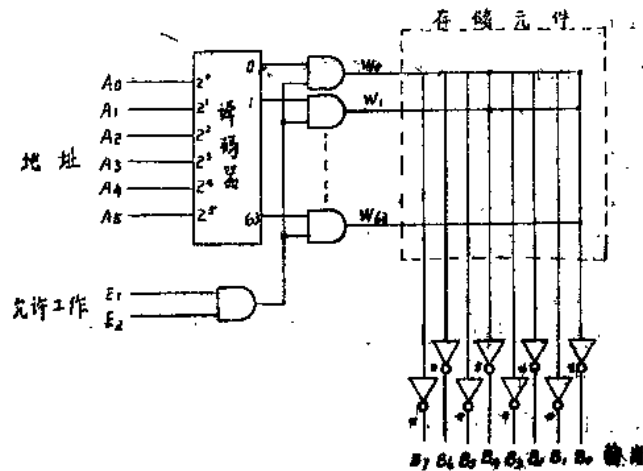


图2.6 只读存储器(ROM)方块图

如图所示,ROM的地址信号 $A_0 \sim A_5$ 经译码后成为0~63。

当 E_1 和 E_2 为高电位时,则输出字的选择信号 $W_0 \sim W_{63}$ 。依据这64条字线与8条位线直交的交点是否连接来决定ROM的输出 $B_0 \sim B_7$ 。

本图所示的存储元件部分,未被字选择之处,表示没有输出。

例如 W_0 被选择时, B_0 和 B_4 均成为低电位, W_1 被选择时,仅 B_4 处于低电位。

YODIC-S的ROM外围电路示于图2.7。

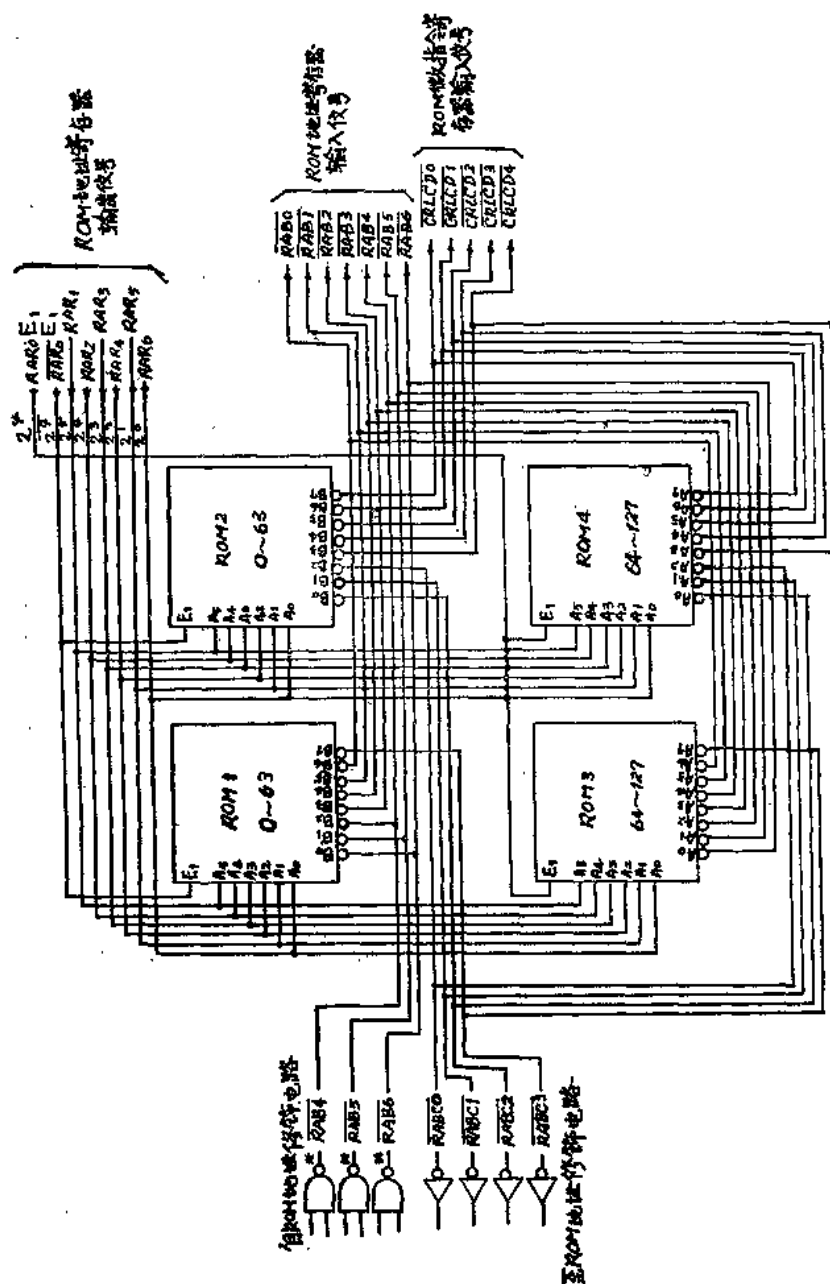


图2.7 只读存储器 (ROM) 的外围电路

如图所示,是由四个ROM组合成128字 $\times$ 16位的电路。RAR₀~₆是来自ROM地址寄存器的信号。当产生RAR₀信号时,ROM1和ROM2工作,输出ROM的0~63字的16位数据。

当产生RAR₀信号时,ROM3和ROM4工作,输出64~127字的16位数据。

在16位数据中,7位RAB信号为ROM地址寄存器的输入信号,4位RABC信号为ROM地址修饰电路的输入信号,其余的5位CRLCD信号为ROM微指令寄存器的输入信号。

所谓ROM的地址修饰，就是指在执行微程序过程中按照外部条件来修饰ROM地址寄存器。ROM的地址修饰是依靠改变RAB（ROM地址总线）信号来进行的。例如ROM地址为00场合的输出如下所示。

	CRLCD					RABC				RAB						
ROM 1.2	0	1	2	3	4	0	1	2	3	0	1	2	3	4	5	6
的输出	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1

这时由于 $\overline{RABC}_3$ 有指定信号，所以ROM地址修饰电路工作。当外部条件未满足时，只有 $\overline{RAB}_3$ 信号。这时ROM地址寄存器的 $\overline{RAR}_3$ 被设定，ROM地址成为08。

当外部条件满足并且ROM地址修饰电路的 $\overline{RAB}_3$ 有指定信号时，由于 $\overline{RAB}_3$ 与ROM的输出接成“接线-或”的方式，因此 $\overline{RAB}_3$ 和 $\overline{RAB}_3$ 都有信号，下一ROM地址成为09。即ROM的输出08被修饰为09。

ROM的地址与其输出的关系可参考2.3.5节。

2.3.2 ROM地址寄存器

ROM地址寄存器用来指定ROM的地址，它是一个7位寄存器。ROM地址寄存器示于图2.8。

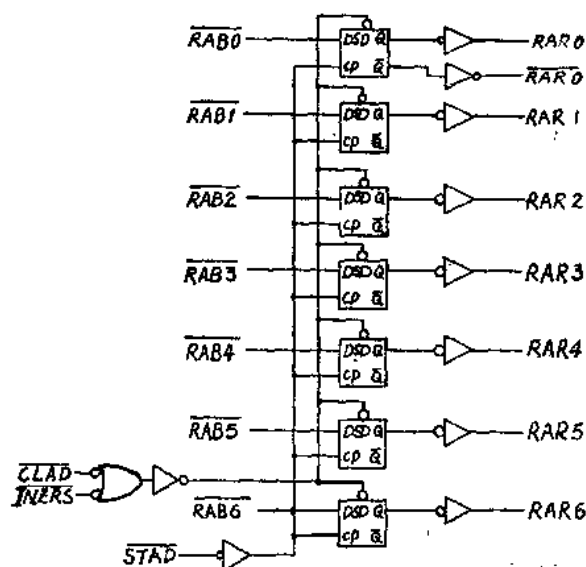


图2.8 ROM地址寄存器

该图中，信号是从各D触发器的Q端取出，尽管接有反相器，但仍需注意输出为 $\overline{RAR}_0 \sim \overline{RAR}_6$ 信号。

这是因为输入条件是反相的，触发器也必须反相工作（后述2.3.3节ROM微指令寄存器也须同样考虑）。

ROM地址寄存器，首先按CLAD或INZRS信号直接设定，这时 $\overline{RAR}_0 \sim \overline{RAR}_6$ 全部成为低电位，只有 $\overline{RAR}_3$ 成为高电位，即ROM指定00号地址。

其次，接收 $\overline{RAB}_0 \sim \overline{RAB}_6$ 信号，并按STAD信号依次设定ROM地址寄存器，指定ROM地