

电子数字计算机原理

第 四 册

(试用教材)

372

北京大学计算机专业

1974 年 11 月

目 录

第一章 通道控制

第一节	通道在计算机中的作用	2
第二节	通道的基本结构	4
第三节	分时操作和排队器	5
第四节	控制字	10
第五节	通道工作过程	14
第六节	子通道接口	19

第二章 光电输入设备

第一节	光电输入设备的用途	21
第二节	光电输入机的基本工作原理	22
第三节	代码输入的工作过程	25

第三章 宽行打印设备

第一节	宽行打印机的打印系统	30
第二节	宽行打印机的走纸系统	33
第三节	宽行打印机的色带系统	34
第四节	宽行打印机的逻辑控制线路	35

第四章 磁带存贮器

第一节	磁带机及记录原理和方式	46
第二节	磁带带面的讯息分布	50
第三节	代码的检验和校正	53
第四节	时标脉冲和同步讯号	58
第五节	找区的过程	62
第六节	代码交换的过程	67
第七节	磁带机伺服电路工作原理	74

第五章 磁盘存贮器

第一节	记录原理和机械结构概况	89
第二节	基本控制原理	92

第三节	定位控制字操作过程.....	101
第四节	写盘控制字操作过程.....	111
第五节	读盘控制字的操作过程.....	115

第一章 通道控制

随着计算机技术的迅速发展，外部设备的控制及其信息传输已逐渐成为一个独立的组成部份，由于每台机器的设计考虑和应用范围不同，这一部份的构造也有所差别，有的机器称为通道，有的称为交换器，有的称为外部控制器，但是它们总的功能都是一样的，就是实现对外部设备的统一管理，以提高机器的使用效率。这里讲的外部设备包括光电机、控制打字机等向主存贮器传送信息的输入设备，和打印机、绘图仪等接受由主存贮器发来信息的输出设备，以及磁带、磁盘等外部存贮器。

目前使用的通道类型大约可以分为三种：

1. 字节多路通道

这种通道可以接8、16、32或更多的子通道，不同的子通道所接的外部设备可以同时工作，以字节为单位分时与通道交换代码。在速度较高，信息流量较大的机器中，字节多路通道一般都设立一个高速缓冲存贮器，用来存放控制字和被交换的代码，通过这个缓冲存贮器可将若干个字节拼成一个或几个字长，再与内存交换，这样，就可以大大减少与内存的交换次数。同时由于缓冲存贮器的速度高，也就提高了通道的工作效率，增加了通道的信息流量。

在速度不是很高，信息流量要求不是很大的中、小型机器中也可以不设缓冲存贮器，控制字放在内存的专用单元，字节代码经过通道直接与内存交换。

字节多路通道适用于连接大量工作频率较低的外部设备，如宽行打印机、光电输入机、控制打字机等，如果通道工作频率较高，也可以接磁带，磁鼓等工作频率较高的设备。在本章中我们主要以这种通道为典型，说明通道的工作原理。

2. 选择通道

选择通道用于连接工作频率较高，使用又很频繁的外部设备，如磁盘。由于磁盘的工作频率比较高，如果将它连接在字节多路通道上，就会大大增加通道的信息流量，影响其它外部设备的交换，所以必须从字节多路通道中独立出来。

选择通道一个通道可接几台外部设备，这几台外部设备不能同时工作，当选择其中的一台进行交换时，其余几台就不能交换。

3. 数组多路通道

这种通道适用于工作频率较高的旋转性的外存设备如磁带、磁盘。这种通道也可以接多台外部设备，它与选择通道不同的是当某台设备进行代码交换时，其它几台可以进行查找工作。由于磁盘、磁带等外存设备真正读写的时间并不太长，而化在查找数据所在位置的时间往往比较长，因此设备在查找时把通道让给别的设备进行交换，这样就提高了通道的利用效率。当然比起选择通道所化的硬设备元件相应要多一些。

第一节 通道在计算机中的作用

通道是实现对外部设备统一管理的一个部件，它代替了运算控制器对输入输出操作的控制过程，使输入输出操作和运算控制器能并行和各自独立的工作。

一、通道在计算机中的作用

早期的计算机都以运控为中心，外部设备也由运控统一管理，这种机器的结构形式如图 1.1 所示。

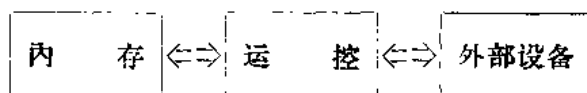


图 1.1

这类机器在内存与外部设备交换代码时，运控都必须停止对数据的运算和处理，等待交换完毕再进行算题。因为目前的一些外部设备受马达转速和机械传动设备的限制，它们的速度相对机器的运算速度来讲是很慢的，像光电输入机是 $1ms$ 输入一排孔，一排孔相当于一个字符或一个代码，而 $1ms$ 对运控来讲是相当长的，在 $1ms$ 中对每秒10万次的机器可以运算100次，对每秒百万次的机器可运算1000次。假如在某台机器上算一个题，纸带输入化了2秒时间，运算化了10秒时间，宽行打出数据化了10秒，那么输入和输出用去的12秒时间对运算控制器来讲就白白浪费了。

所以像上面一类输入输出和主机运算串行的机器，尽管计算机的速度很高，但真正算题的时间是很少的，大量的时间要化在输入输出上，机器的效率得不到充分的发挥。

后来的机器采取了中断的方式，也就是一般情况下机器仍在算题，当外部设备需要输入或输出一个代码时，发一中断信号，要求主机暂时停止一下运算，把正在运算的数据保存起来，先进行输入或输出，在这一个数码输入或输出完毕后再进行运算。例如光电机是每 $1ms$ 送一个代码，每送一个代码时，先将纸带上的码子通过光电转换放在一个快速的缓冲寄存器里，然后向主机发一中断信号，在很短时间里打断一次主机运算。先进行输入，但其所占的时间是很短的，因此 $1ms$ 中的大部份时间仍可用于算题，如图 1.2 所示。



图 1.2

这样的机器在效率上虽有提高，但是中断次数频繁，外部设备的管理仍然通过运控来实现，这就带来了以下两个缺点。

第一，由于输入输出操作串行在主机的程序之中，每次使用外部设备都要占用主机时

间，通过主机的部件来完成，这样主机的效率仍然得不到充分的发挥。

第二：输入输出都由运控统一管理，这就使得运控的逻辑设计复杂，零散，并且没有灵活性，因为一种外部设备确定后，它的逻辑线路也就固定了，以后机器用户想要增加或减少外部设备都是很困难的。

程序系统的进一步发展，一般都需要机器配有高速的外存设备（磁带、磁盘）来存放软件系统，这样内存与外部设备之间的代码交换就显得更加频繁，上面的矛盾也就显得更加突出，这些矛盾促使机器的结构发生了变化，即由原来以运控为中心，变成以内存为中心，结构形式如图 1.3 所示。

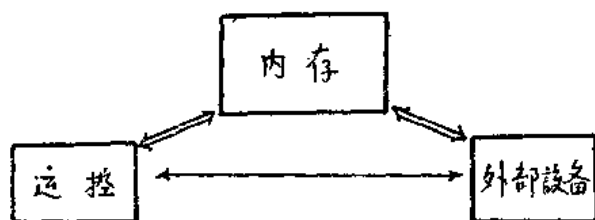


图 1.3

图 1.3 是内存、运控、外部三者关系的示意图，实际上由于计算机应用的日益广泛和多道程序的使用，一般都要求一台机器配有多种多台外部设备，并且要求这些外部设备能够同时工作，所以在内存与外部设备之间就出现了一个新的组成部份称为“通道”，其结构框图如图 1.4 所示。

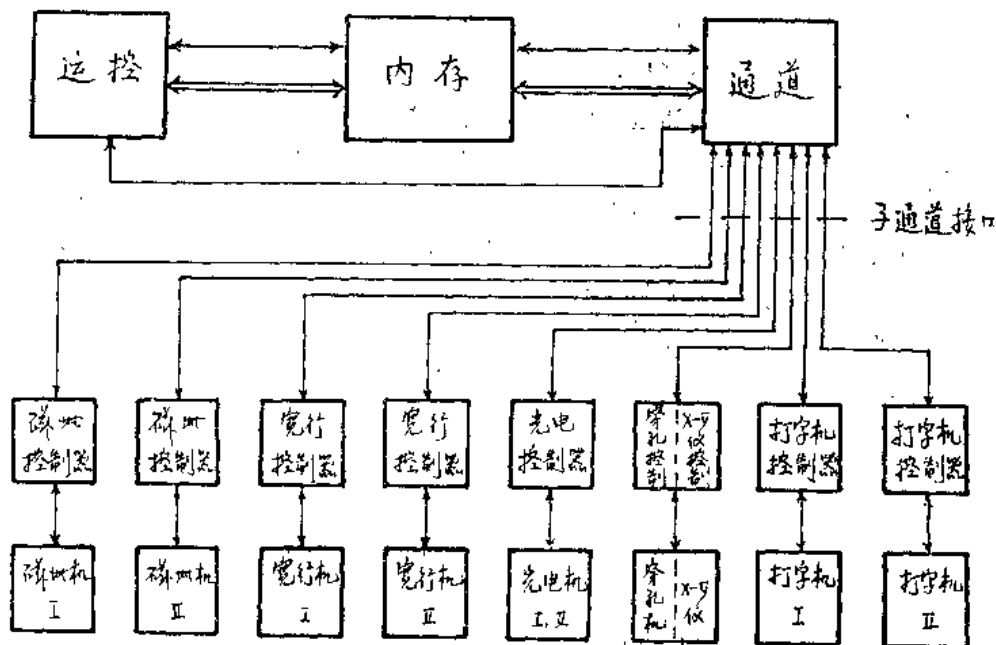


图 1.4

这样的机器结构，在算题程序中需要使用某台外部设备时，运控只要在开始使用前和使用结束后两次化极短的时间参与控制管理工作，而在整个外部设备输入输出过程中完全可以由通道独立地对外部设备进行控制，输入或输出的代码也不必经过运控而直接和内存交换，从而保证了外部设备工作时运控也能继续工作。同时，由于通道的工作频率比外部设备的工

作频率高得多，通道可以用极短的时间处理一台外部设备输入输出的代码，通道用分时的办法可以保证各外部设备之间同时工作。

二、子通道

通道设有若干个子通道，图1.4上画了8个，根据实际需要可以是64、128、256等。这些子通道与外部设备控制器的接口是标准的，也就是不管外部设备的品种类型不同，要求通道与外部设备控制器的连接方式都是一样的，具体的说都是数字式的，如果是模拟信号，在外部设备控制器中就应该将模拟信号转换成数字式的信号，再与子通道相连，外部设备与子通道间代码线和控制线的条数和内容也应该是统一的。各外部设备和通道之间线缆连接方式目前有并行和串行两种，这在后面还要讲到，图1.4是并行方式的，在本章中我们是以并行方式为典型来介绍通道的工作方式。

一个子通道可以接几台同类型设备，如图1.4中的第4子通道，接了两台光电机，也可以接几台不同类型的设备，如图1.4中的第5子通道，接了X—Y仪和穿孔机两台设备，这几台设备一般只能串行工作，不再分时使用子通道。

标准接口有利于机器用户根据需要增加或更换新的外部设备，而主机部份不需作任何变动。

三、通道的具体功能

通道在管理程序的调度下具体完成下列任务：

1. 接受主机的访外指令(FW)，确定所要访问的子通道。
2. 从内存选取属于该子通道的通道控制字，根据控制字向外部设备控制器和设备发送工作命令，工作命令包括启、停，总清，读写等工作性质，控制码……。
3. 对各通道来的字节交换请求(S_0)按优先次序进行排队，实现分时工作。
4. 实现外部设备和内存之间的代码传送，根据控制字指出代码在内存的地址，和在输入输出设备上的区域，并控制交换个数。
5. 将外部设备的中断请求和通道的中断请求，送往主机。
6. 控制外部设备进行某些控制性的动作，如宽行走纸等。

第二节 通道的基本结构

由于计算机的设计要求和应用范围不同，各类机器的通道结构是不完全一样的，尤其是随着计算机使用的日益广泛，计算机由单机使用发展到双机使用，以及多台机器组成计算机网联合起来使用，计算机通道结构也就各有不同。

在这里我们只介绍一种比较简单的字节多路通道，其结构框图如图1.5所示，图中虚线内的部份属于通道。

从图上可以看出，通道主要由控制字寄存器，代码缓冲寄存器，排队器，拍节发生器，通道传输和操作控制等几部份组成。

控制字寄存器 (J_K)：用来存放正在执行的控制字。

代码缓冲寄存器 (J_N)：是外部设备与内存进行代码交换时，用来存放外部设备和内存传来的代码。

排队器 (P)：根据规定的优先次序，确定优先执行的通道。

拍节发生器 (PF)：产生通道的工作拍节。

通道传输：用于外部设备和通道之间代码和讯号的传输。

操作控制：根据控制字内容，排队结果和通道拍节产生通道微操作。

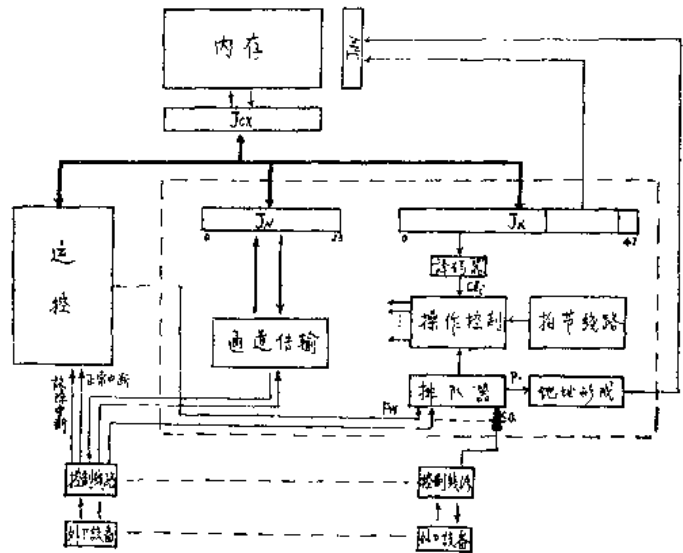


图 1.5

以上几部份的具体功能及彼此之间的有机联系，将在下面几节中详细谈到。

第三节 分时操作和排队器

一、什么叫分时操作

在日长生活中我们经常碰到这样的例子：就是几项工作都要公用一个场地和设备，例如从京沪、京广、京包……各线来的列车都要在北京站上、下旅客，北京站就是各列火车的公用场地，如果各列火车的到达和开出没有一定的时间和次序，那么车站就会混乱不堪，列车在铁轨上也会发生事故。因此每个火车站都有一个时间表，安排各列火车在不同的时间到达和开出，这样一个火车站就能充分发挥它的效率，有条不紊的接待各列火车的旅客。

在计算机里也经常遇到这样的情况，就是许多操作要公用一个部件。

例如内存是运控和通道的公用部件，从图 1.5 中我们可以看到运控，通道和内存都有代码总线连系，运控每执行一条指令都要从内存取指令、取数或存数，而外部设备输入指令和数据，或者输出运算结果也要向内存存数和取数，但是内存每次只能读出一个数，或存入一个数，因此通道和运控不能同时使用内存，只能“分时”使用。

又如通道的 8 个子通道接有多种多台外部设备，这些外部设备同时工作时，都要通过通道和内存进行数据交换，也就是通道和内存又是外部设备输入输出的公用部件，而通道与内存每一时刻又只能允许交换一个数据，这样各外部设备同时工作都要求和内存交换数据时，就要排一排队，分先后次序分时与通道、内存交换代码。

下面，我们具体分析访内分时和通道分时的原理及其实现方法，通过具体分析进一步理解分时的意义。

二、访内分时

上面讲到运控工作时要经常地访问内存，通道工作时也要经常地访问内存，怎样协调它们的工作呢？这里需要满足以下两条原则。

第一：通道和运控都要求访内时，通道应该优先访内，因为通道连接着许多转动性的外部设备，如磁带、光电等，它要求不断的和内存交换数据，时间性比较强，如果上一个不响应，下一个代码来了就会冲掉上一个代码，造成丢码。例如磁带机输入时是每 $25\mu s$ 向主机送一个字节，一个字节一般是 6 位（或 8 位）二进制码。每从磁带上读出一个字节时，磁带控制器就将它放在一个缓冲寄存器 J_h 里，如图 1.6 所示，同时向通道发出一个交换代码的请求 S_Q ，置“1”通道相应的请求触发器 C_{SQ} ，如果在 $25\mu s$ 内 C_{SQ} 得不到响应， J_h 里的代码就不能送到内存，那么下一个码子就进入了 J_h ，冲掉了上一个代码，而第二个 S_Q 又一次置“1” C_{SQ} ，相当于两个 S_Q 合成一个 S_Q 了，这就造成了丢码。

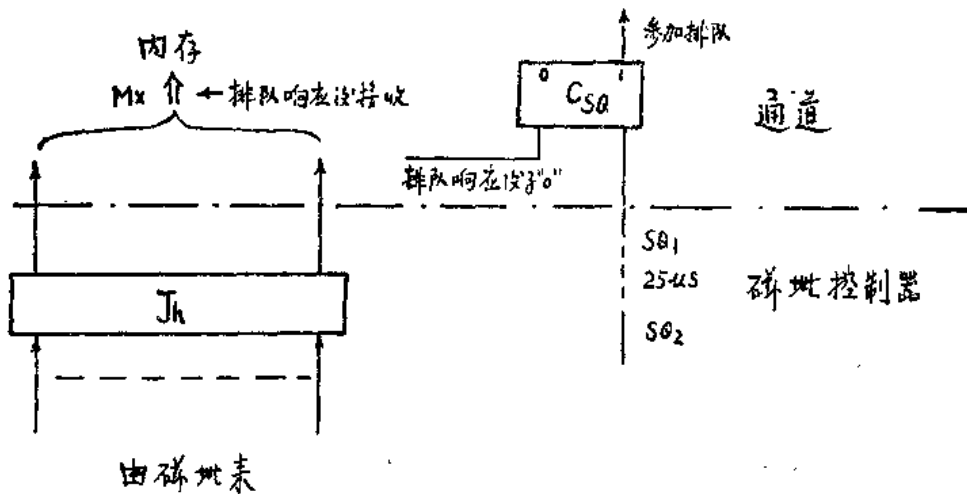


图 1.6

第二：内存是公用部件，设计安排应尽可能发挥内存的作用，使它尽可能忙碌地工作，通道访内完毕后应立即供给运控访内，运控工作过程中有不访问内存的时候应尽可能给通道访问内存提供条件。

为了按照上述原则分时访问内存，可以设置一个触发器叫做允许访问内存触发器 C_{YN} 。

$C_{YN}=1$ 允许通道访问内存

$C_{YN}=0$ 允许运控访问内存

如图 1.7 所示， C_{YN} 触发器采用维持阻塞触发器，图中 W_{QW} 表示通道请求访内电位，

$W_{QW}=1$ 时表示通道有访内请求，在计数端加有 m_{YN} 脉冲，当 W_{QW} 一旦有请求， C_{YN} 就可很快地置为“1”态。但是并不是通道一有请求 C_{YN} 立即可以置为“1”态，运控在向内存取指令或存（取）数的一个周期内是不允许切换的，这样会造成工作混乱。那么什么时候切换呢？在以下三种情况下只要 $W_{QW}=1$ ， C_{YN} 便可置为“1”态。

第一：在运控取完一条指令后（假设为 W_5 拍），如果这时 $W_{QW}=1$ ， m_{YN} 脉冲就可置“1” C_{YN} ，运控访内暂停。

第二：运控取来的指令属于不读不写指令，如转移，移位等类指令，执行这条指令不需要取数和存数，这时不读不写触发器 $C_{\overline{RX}}=1$ ，这时 C_{YN} 也允许切换。

第三：运控取来的指令是“ $\times$ ”、“ $\div$ ”一类需要转局部的指令，这类指令运控需要较多的拍节才能完成，而不必继续访内，这时局部操作触发器 $C_{JK}=1$ ， C_{YN} 允许切换。

综合以上三种情况， C_{YN} 置“1”的条件是

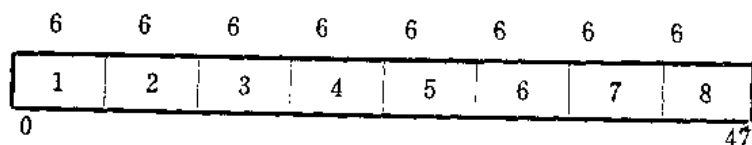
$$W_{QW}(W_5 + C_{\overline{RX}} + C_{JK})m_{YN}$$

在二、三两种情况运控和通道访问内存并不冲突，所以运控通道处于并行工作状态。对于第一种情况，运控处于暂时等待状态，通道访问内存结束时，给一结束脉冲置“0” C_{YN} ，运控便可又可继续工作了。

三、通道分时

通道是各外部设备的公用部件，当各子通道的外部设备同时工作时，通道也以分时的方法和外部设备交换代码，它和访内分时的道理是一样的，但情况有所不同，在访内时，通道访内运控可以等待，而在通道分时中由于各种外部设备都以一定的工作频率自行工作着，每个交换请求必须在一定的时间内处理完，如超过等待时间就会造成丢码的错误，所以通道分时必须要求通道有较高的工作频率，使得所有子通道的外部设备同时工作时，能接待它们的交换请求而不丢码。

外部设备一般是以字节为单位和主机进行交换的，一个字节一般是6位（或8位）二进制码，表示一个字符。全字长为48位的二进制码以图示的顺序和内存交换。



如果某些外部设备的工作频率比较高，而内存、通道的工作频率不够高，这些设备仍以字节为单位进行交换，内存和通道将接应不暇，为了降低交换频率，对工作频率较高的外部设备可以以二字节，半字长，或全字长为单位进行交换，扩展字节的工作在哪里进行根据

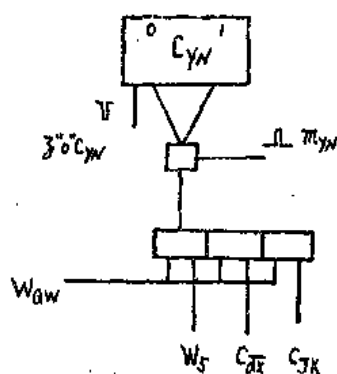


图 1.7

具体情况而定，如果通道没有专门设立一个缓冲存储器，那么扩展字节的工作可在外部设备控制器中进行，如果通道设有缓冲存储器，那么就可利用缓冲存储器将输入的字节代码拼成全字长再送入内存，输出时将内存的全字长代码先送到通道的缓冲存储器，再以字节为单位送到外部设备。

通道怎样分时接待各外部设备的交换请求呢？这就需要排排队，排队的原则是工作频率高的应该优先于工作频率低的，因为工作频率高的 S_Q 请求如不尽快响应，下一个 S_Q 就会很快到来容易造成丢码。

下面我们用法具体说明通道是怎样分时工作的，图 1.8 中我们假设有四个子通道的外部设备在同时工作：

1 台磁带每隔 $25\mu s$ 发 $-S_Q$ 请求

2 台宽行每隔 $150\mu s$ 发 $-S_Q$ 请求

1 台光电每隔 $1 ms$ 发 $-S_Q$ 请求

通道 $10\mu s$ 可以交换一个字节

假设在 t_0 时刻四个子通道的外部设备都同时发出 S_Q 请求。

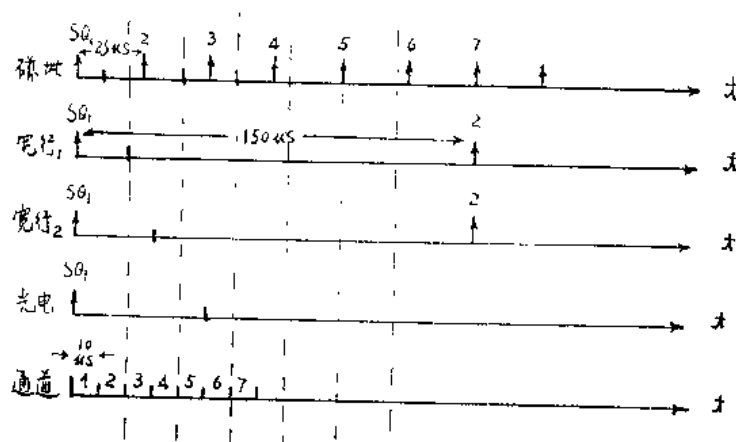


图 1.8

图中的箭头表示外部设备 S_Q 请求到来的时间，小短竖线表示 S_Q 完成的时间。

根据排队原则，通道第一个 $10\mu s$ 接待磁带的 S_{Q1} ，第二个 $10\mu s$ 接待宽行1的 S_{Q1} ，第三个 $10\mu s$ 接待宽行2的 S_{Q1} ，第四个 $10\mu s$ 接待磁带的 S_{Q2} ，第5个 $10\mu s$ 接待光电的 S_{Q1} ，第6个 $10\mu s$ 接待磁带的 S_{Q3} ，第7个 $10\mu s$ 通道就是空闲的了……，通道就是这样与外部设备交换信息。

四、排队器

上面讲到排队器的任务就是根据预先确定的优先次序对各子通道的 S_Q 请求进行排队，确定下一次通道和那个子通道的外部设备进行交换。

排队器的逻辑考虑就是优先权高的子通道有请求时，封锁优先权低的通道，让优先权高的先排上队。

图1.9是排队器的逻辑原理图， $C_{SQ0}, C_{SQ1} \dots C_{SQ7}$ 是与8个子通道的 S_Q 相对应的请求源，子通道的外部设备有 S_Q 请求时就置“1”相应的触发器 C_{SQi} ， i 表示了通道号。 k_0 到 k_7 是8个开关，用来控制与通道的分联，若 k_1 的2点与1点相连，则外部设备发来的 S_Q 不起作用，第1子通道和外部设备处于断开状态，若 k_7 的2点与3点相连，表示第1子

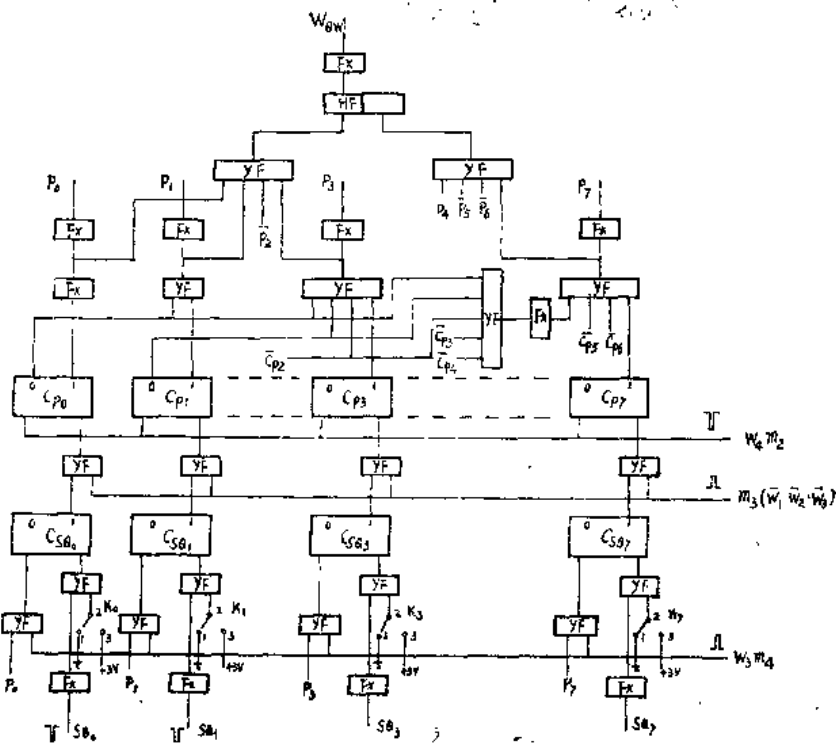


图 1.9

通道与外部设备处于接通状态，这些通道开关的设置是为了便利机器的调整和维护。

$C_{P0}, C_{P1} \dots C_{P7}$ 是排队触发器，在排队前先将请求源通过脉冲 m_3 ($\bar{W}_1 \cdot \bar{W}_2 \cdot \bar{W}_3$) 打入 C_{Pi} ，打入脉冲什么时候来是有条件的，条件是 $\bar{W}_1 \cdot \bar{W}_2 \cdot \bar{W}_3$ ，在第四节讲通道工作节拍时还要讲到， W_1, W_2, W_3 其中有一个是高电位表示其它通道正在交换，排队器状态不能更变，也就不能发打入脉冲。

为什么需要设一排 C_{Pi} 触发器，而不能由 C_{SQi} 直接参加排队呢？这是因为外部设备发来的 S_Q 信号是随机的，随时有来 S_Q 的可能，比如开始排队时 $C_{SQ0} = C_{SQ1} = 0$ ，而 $C_{SQ2} = 1$ ，显然应该 C_{SQ2} 排上， P_2 为高电位，由第二子通道的外部设备进行交换，但可能第2子通道正在交换过程中，又来了 S_{Q0} 置“1” C_{SQ0} ，如果没有 C_{Pi} 这排触发器屏蔽，而由 C_{SQi} 直接排队马上又会变成 S_{Q0} 排上， P_0 变成高电平， P_2 变成低电平，这样就造成了工作的混乱。加了 C_{Pi} 这排触发器，那么在这次打入后到下次打入前 C_{Pi} 的内容就不会改变，排上队的请求源也就固定了。

C_{SQi} 触发器在这次交换完毕，下次打入前的 W_4m_2 清“0”。

排上队的请求源在代码交换完毕后的 W_3m_4 便可清“0”，没有排上的请求源不应请

“0”。

C_{pi} 的排队是通过一排逻辑门来实现的，每个子道道的 S_0 请求排上队的条件由下面的逻辑式表示，即 0 号子通道排上队的条件是 $C_{p0}=1$ ；1 号子通道排上队的条件是 $C_{p0}=0$ ，而 $C_{p1}=1$ 。……

$$P_0 = C_{p0}$$

$$P_1 = \overline{C_{p0}} \cdot C_{p1}$$

$$P_2 = \overline{C_{p0}} \cdot \overline{C_{p1}} \cdot C_{p2}$$

$$P_3 = \overline{C_{p0}} \cdot \overline{C_{p1}} \cdot \overline{C_{p2}} \cdot C_{p3}$$

$$P_4 = \overline{C_{p0}} \cdot \overline{C_{p1}} \cdot \overline{C_{p2}} \cdot \overline{C_{p3}} \cdot C_{p4}$$

$$P_5 = \overline{C_{p0}} \cdot \overline{C_{p1}} \cdot \overline{C_{p2}} \cdot \overline{C_{p3}} \cdot \overline{C_{p4}} \cdot C_{p5} = \overline{C_{p0}} \cdot \overline{C_{p1}} \cdot \overline{C_{p2}} \cdot \overline{C_{p3}} \cdot \overline{C_{p4}} \cdot C_{p5}$$

$$P_6 = \overline{C_{p0}} \cdot \overline{C_{p1}} \cdot \overline{C_{p2}} \cdot \overline{C_{p3}} \cdot \overline{C_{p4}} \cdot \overline{C_{p5}} \cdot C_{p6} = \overline{C_{p0}} \cdot \overline{C_{p1}} \cdot \overline{C_{p2}} \cdot \overline{C_{p3}} \cdot \overline{C_{p4}} \cdot \overline{C_{p5}} \cdot C_{p6}$$

$$P_7 = \overline{C_{p0}} \cdot \overline{C_{p1}} \cdot \overline{C_{p2}} \cdot \overline{C_{p3}} \cdot$$

$$\overline{C_{p4}} \cdot \overline{C_{p5}} \cdot \overline{C_{p6}} \cdot C_{p7} = \overline{C_{p0}} \cdot \overline{C_{p1}} \cdot \overline{C_{p2}} \cdot \overline{C_{p3}} \cdot \overline{C_{p4}} \cdot \overline{C_{p5}} \cdot \overline{C_{p6}} \cdot C_{p7}$$

将这些逻辑式，落实成逻辑图，就是图上的排队线路

以上是排队器的原理图，随着排队源数目增加，或优先情况不同，排队器的形式也会有所不同。

第四节 控 制 字

通道的主要功能是管理和协调外部设备的工作，而通道的工作又是听从全机统一调度的，控制字相当于通道的工作命令，通道的工作是通过执行控制字来实现的。

一、控制字

由于机器类型和设计考虑的不同，各类机器控制字的形式和条数是不一样的，基本的控制字一般有输入，输出，控制和停机四条，我们举例说明控制字的形式和内容

控制字的一般形式：

0	CZ	23	T	5	m	7	N	23	F	27	28	DN	44	JH
3		3		1			17		1			17		2

CZ : 表示操作码, 操作码用多少位根据每台机器设计而定, 如果要求通道功能强, 控制字条数多, 操作码的位数相应也就多一些。这里用 3 位二进制表示, 最多可允许有 8 条控制字。

T : 为台号, 表示某个通道的第几台工作, 一个通道可接若干台。

m : 为特征位, $m=1$, 表示交换个数满。

$\bar{N}$: 为交换个数, 以半字长为单位, 补码表示。 $\bar{N}$ 的位数根据机器内存存贮量而定, 最大交换个数就是内存总单元数, 若内存容量为 13 万个半字长, 交换个数就占 17 位

F : 也是特征位, 可以根据外部设备和管理程序的约定, 不同设备表示不同含意。

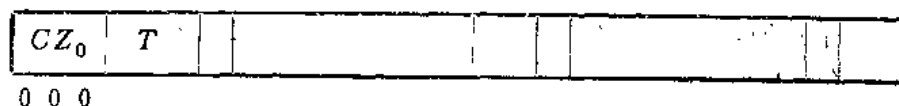
DN : 表示正在交换的代码在内存存放的地址。

JH : 为字节号, 如全字长为 48 位, 那么半字长 24 位就可以分成 4 个字节, 每个字节 6 位。

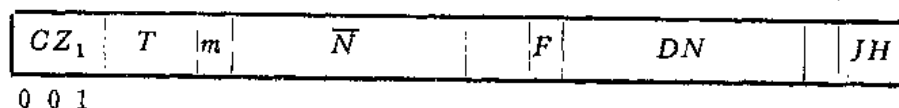
没有加说明的位为空位。

这里说明一下控制字中没有设通道号, 原因是在工作过程中通道号已由排队器决定。

具体的控制字有以下四条:



CZ_0 为停外部设备控制字, 作为管理程序强迫某台外部设备停机用。

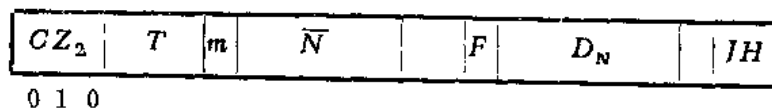
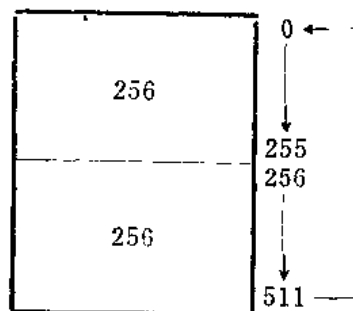


CZ_1 为外部设备代码输入到内存的控制字。

对于光电输入在有管理程序下使用往往是轮流输入到内存的二个缓冲区, 如图示每个缓冲区 256 个半字长, 但在机器未配上管理程序时往往按内存地址依次输入。所以这里约定

$F=1$ 时交换完 256 个半字长后正常中断 ZCD 。以便程序对已满的缓冲区进行处理。交换完 512 个半字长时后发正常中断, 内存地址变到起始地址。

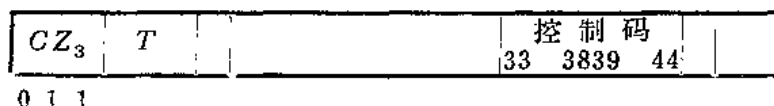
$F=0$, 按内存地址输入。



这条为内存代码输出到外部设备的控制字。

对 $X-Y$ 仅约定 $F=1$ 出界发故障中断

$F=0$ 否



这条是控制型的控制字，对不同外部设备可作不同的约定。例如对磁带表示找区，控制码即为区号12位（33—44），对宽行表示空走纸，控制码即为走纸行数，用6位码子（39—44）表示

二、控制字专用地址：

因为8个子通道的外部设备，可以同时并行工作，因此每个子通道工作时都对应有一条控制字，控制字放在内存的专用单元中，8个子通道就占用8个全字长单元，这些专用单元是机器设计时，事先约定好的，比如放在内存60—77号8个单元（八进制半字长编码）。如果通道设有专门的高速缓冲存储器也可将控制字放在缓冲存储器里，这样可以提高通道的工作频率。

控制字平时放在内存的专用单元中，当某个子通道的外部设备需要交换一个代码时，就需要将该子通道对应的控制字取到控制字寄存器 J_K 中执行，在这次交换完毕后修改控制字，并将它送回到内存专用单元。

那么控制字地址是怎样形成的呢？由于子通道号与控制字专用地址一一对应，我们就可以利用某个子通道排上队后的排队信号 P_i 来形成控制字地址。例如“1”子通道的请求排上队后， P_1 是高电平，利用 P_1 来形成“62”号，见图1.10， W_1 和 W_4 拍送往内存的地址线（ DX ）上产生00...0110010，即“62”号。图中 $\sum P_i = P_0 + P_1 + \dots + P_7$

图1.10中 W_2 和 W_3 拍所连接的一排门，是向内存存取代码时由控制字所决定的内存地

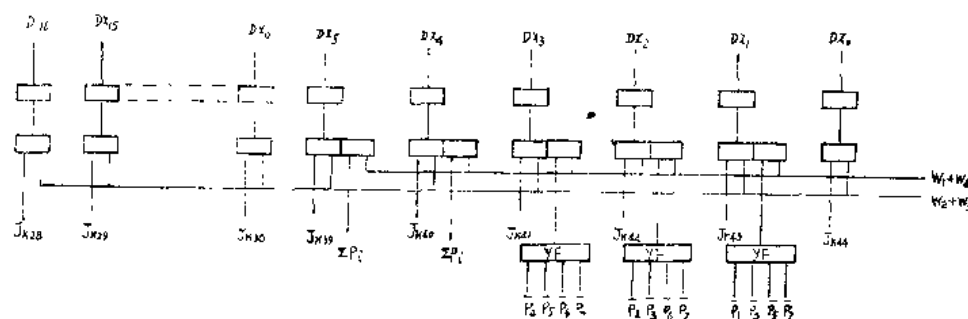


图 1.10

址，控制字的内存地址对应 J_{K28-44} 。

三、控制字的修改

对于 CZ_1 、 CZ_2 两条控制字在执行过程中对交换个数和内存地址需要不断加以修改，也就是每交换一个字节后，字节号 JH 就要加1，表示下次输入的字节代码，放到内存单元的下一个字节位置，每交换完4个字节，表明已交换完一个半字长，访内地址和交换个数也随

着加1。

CZ_1 、 CZ_2 中的交换个数是以补码填写的。为什么要用补码填写呢？假如用原码填写那么每次交换完一个半字长，就要将交换个数减“1”，交换个数减到17位为全“0”时，交换就完毕了。因此用原码表示在逻辑线路上要求有减“1”和判全“0”的功能。如果用补码表示那么每交换完一个半字长就加“1”，加到 m 这一位是“1”时交换就完毕了，因此用补码表示在逻辑线路上要求有加“1”和判 m 位是否为“1”的功能。在逻辑线路上加“1”比减“1”要简单容易，判一位是否为“1”比判17位是否为“0”容易，这就是为什么要用补码填写的道理。

在此顺便提一下 CZ_3 用于宽行走纸时，它的控制码也用补码填写，其道理和上面说的一样，因为宽行每走一行也需要对控制码加以修改，最后也要判走纸行数是否够了，用补码填写在逻辑上比较方便。

控制字的修改工作在控制字寄存器 J_K 中执行。

四、 J_K 寄存器

J_K 寄存器用一排维持阻塞触发器构成，可以接收从内存数码寄存器，或开关寄存器送来的控制字，开关寄存器是通道分调面板上设有的一排开关构成的寄存器，分调时可以手扳控制字模拟从内存取来的控制字，便于调机。

J_{K7-23} 、 J_{K29-44} 、 J_{K46-47} 根据控制字要求构成计数状态，由于通道的一拍相当于内存的一个周期（约 $2\mu s$ ），时间比较长，因此对进位速度要求并不高，根据杆件排列的方便17位采用三位一组，共6组，组内并联进位，组间串联进位，每组的简图如下：

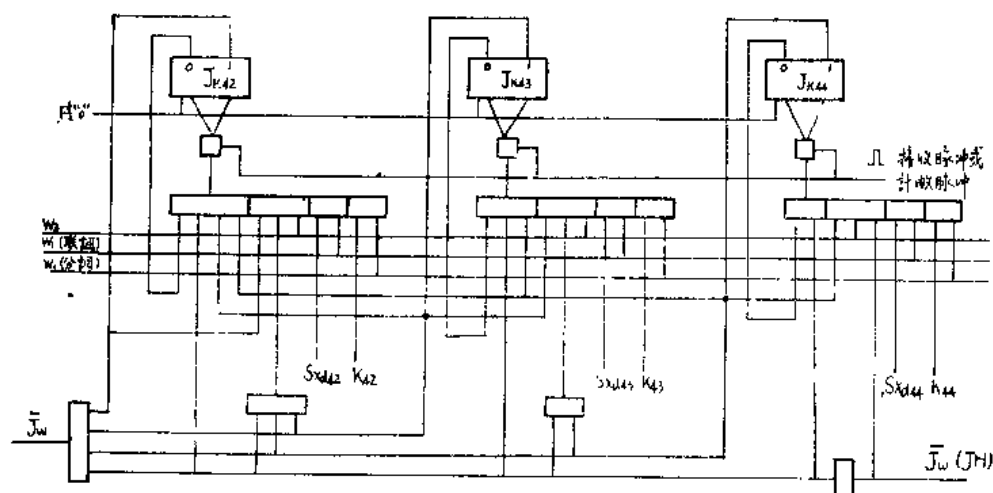


图 1.17

图中 S_{x42} 代表从内存数码寄存器42位来的读数线，联调时存 W_1 拍内被接收到 J_{K42} 中去，其它位也同样。 K_{42} 表示开关寄存器42位，在分调 W_1 拍接受到 J_{K42} 中去，其它位也同样。

W_3 是修改控制字的拍节，我们将修改前的数码用 J_{K42} 、 J_{K43} 、 J_{K44} 表示，修改后的数

码用 J_{K42}^* 、 J_{K43}^* 、 J_{K44}^* 表示，那么他们应该有下面的逻辑关系式：

$$J_{K44}^* = J_{K44} \cdot \bar{J}_w + \bar{J}_{K44} \cdot J_w$$

$$J_{K43}^* = J_{K43} \cdot (\bar{J}_w + \bar{J}_{K44}) + \bar{J}_{K43} \cdot J_{K44} \cdot J_w$$

$$J_{K42}^* = J_{K42} \cdot (\bar{J}_w + \bar{J}_{K43} + \bar{J}_{K44}) + \bar{J}_{K42} \cdot J_{K43} \cdot J_{K44} \cdot J_w$$

划成逻辑图就是图1.11的有关部分，其中 J_w 表示进位信号， J_w 中已包含了 W_3 的因素，所以在 YHF 门的第一或可免去 W_3 的条件。

第五节 通道工作过程

一、交换过程和通道的工作节拍

通道的工作节拍主要根据代码交换过程的特点进行安排的，我们先分析一下代码的交换过程。

假如控制字是放在内存的，那么通道每把一个字节送往外部设备需要三次访问内存：

第一次：由内存专用地址取出控制字到 J_K 寄存器。

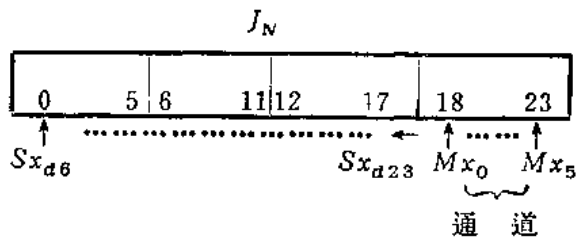
第二次：根据控制字中的内存地址从内存读出24位代码，再根据控制字中的字节号将所要交换的一个字节送到外部设备。

第三次：将修改后的控制字写回内存专用单元。

在输入操作时，如果内存有字节存取的功能（即每次可以存取一个字节，其它字节内容不变），那么输入一个字节也可以类似上面三次访问内存完成。如果内存只能是半字长存取，那么输入一个字节需要四个内存周期：

第一次：从内存取出控制字到 J_K 寄存器

第二次：将通道送来的6位数码送到 J_N （数码缓冲寄存器的最后六位，即18—23位），同时根据控制字的内存地址读出24位数码，左斜六位送到 J_N 寄存器0—17位，内存读出的前六



位即 Sx_{d0-5} 位不接，内存读出的第七位 Sx_{d6} ，送入 J_{N0} ，第八位 Sx_{d7} 送入 J_{N1} ，依此类推。

第三次：根据控制字的内存地址，将 J_N 的内容写回内存。

第四次：将修改后的控制字写回内存。

这样经过4个 S_Q 后内存半字长单元就充满了输入的代码，这时内存地址被修改了(+1)，下次输入的内容又送到另一单元。每执行1个 S_Q ，内存单元的内容变化如下，1—4代表输