

試 用 讲 义

电子数字计算机主控制器

(指控、数存、存控)

研究所四室指控组 编

长 沙 工 学 院

前 言

本文包括指令控制分机，超快速数码缓冲存贮器和存贮控制分机三部分内容。这三部分内容统称为中央处理机的主控分机，即先行控制分机。

先行控制分机是处于主存贮器与运算控制部件之间的一个匹配部分。它有如下几个作用：

一、协调各分机之间的工作，提高主机的速度。

从运控的角度看，在中小型机器里，运控执行指令是直接和存贮器打交道，因此每执行一条指令要分如下几步：

1. 取指令
2. 处理指令 <完成变址及取操作数等>
3. 完成指令所规定的操作。

由上看出，每执行一条指令，一般要访问二次存贮器。在同步控制的机器中，作为执行指令的周期一般为大于或等于二次访问存贮器的周期。这就限制了指令的最短执行时间，直接影响了机器速度的提高。为了提高速度，充分发挥运算部件的效率，把执行指令过程的三步分别由不同部件完成。即：取指令和处理指令工作单由先行控制部件完成，运控仅完成指令操作，这样，运控所需要的指令和操作数，顺手从先行控制部件提取，而不需要考虑存贮器的影响。从而使运控部分的作用可以充分发挥。

从存贮器的角度看：存贮器的发展目前也很快，但由于大机器对主存的容量和速度都要求很高，相比之下，存贮器的速度仍然是慢的，本机存贮容量为26万，访问周期 $1\sim 1.2\mu s$ ，从速度上看，是不能满足百万次机器的要求的。本机把26万主存分成八个模块，八个模块可以同时存取，以此提高速度。本机为双机系统，访问存贮器的请求源有八个，这八个请求源可能同时访问存贮器，甚至可能同时访问某一个模块，（3.2万为一个模块），这就需要对请求源进行排队处理。对八个请求源和八个模块进行综合控制处理，按轻重缓急使每个请求都能各得其所，又使八个模块密切配合，有效工作，这个控制就由主控分机中的存控来完成。

先行控制部件在完成运控之间速度匹配的过程中，超快速数码缓冲存贮器又起着它独特的作用。它是一个数据缓冲存贮器，计算机在算题过程中，使用内存单元数据时，对某一数据而言可能重复使用多次，对内存单元来说又有一定的相关性（若访问了 a 单元，往往又要访问 $a+i$ ，而 i 往往又是很小的）。这样，当指令需要取数时，首先向数存取，若有，就不要访问主存；若没有，则由数存访问主存，并且每访问一次就取一组（4个单元）数据。该组数据要在数存中存放相当一段时间，在该组数据被作废前，指令可以重复使用多次，这就大大减少了主存的访问次数。另外，数存的存取时间为主频

周期，所以数据往往可以随要随取，这对先行控制部分的工作是十分有利的。

二、执行一些特殊指令，完成机器的控制作用。

对转移等一些不需要进行运算的特殊指令均在指控中完成。

软件所需要的一些控制信息及中断问题由指控实现。

本文编写有两个目的：一是作为教材用；二是作为部件的说明书。编写的指导思想是：与教材相比要增加一些本机的具体内容和实现方法；和说明书相比，在基本原理的说明上有所增加。但由于编写时受时间和水平所限，肯定会有错误和不足之处，请批评指正。

编 者

1975. 9

目 录

第一章 指令控制器

§ 1.1 概 述	1
§ 1.2 指控基本部件	4
§ 1.3 几种典型操作的控制	19
§ 1.4 指控调试面板	24
§ 1.5 指控部件的检查	32
§ 1.6 中断的有关问题	39

附 录

1. 标准计数器	57
2. 简易寄存器	62
3. 全加器方案	64
4. 模三余数校验	67

第二章 数码缓冲存储器

§ 2.1 概 述	71
§ 2.2 双极型集成电路存储器的工作原理	72
§ 2.3 自动调度方案	75
§ 2.4 数码缓冲存储器的自检	86

第三章 存访控制器

§ 3.1 存控的功能和技术指标	89
§ 3.2 排队器和信息通道	91
§ 3.3 系统的六种工作状态	99
§ 3.4 地址变换	102
§ 3.5 回答、中断、发送	106
§ 3.6 分调与联调	111

附 录

1. 主存速度的估计	116
2. 奇偶校验方法的可信概率	116
3. 原码与反码的奇偶校验	117

第一章 指令控制器

§ 1.1 概 述

1 任务及要求

本机是一台多外设通道的双机系统的大型计算机，它采用双运算器方式，可同时分别完成两单字长运算或联合完成双字长运算。本机是一台以专为主、专通结合的多功能计算机。

本机在双机同时完成单字长指令时，每秒钟可完成四百万条以上的指令。存储器访问周期 $1\sim 1.2\mu s$ ，虽采用模8工作方式也难以和运算控制器匹配。为了满足机器对指令快速处理的要求，使机器的运算部件处于良好而又紧张的工作状态，要求指令控制部件及时地向运算部件提供指令，为此指令控制部件采用了先行控制技术。在运算部件对指令处理的同时，指令控制部件将对以下要执行的指令进行预先的加工处理，为运算部件对指令的处理作好充分而必要的准备。因此，只要指令控制部件能及时地供上运算部件处理指令的需要，而不致使运算部件有空闲时间，则机器的运算速度就能获得最好的发挥。根据本机速度要求，在极端情况下，运算部件每拍都要指令，为此，指令控制部件要每拍能供给指令。

指令部件的作用可归纳为如下几个方面：

1) 对指令进行排队：将变字节的指令编成标准形式的指令。

本机的指令根据其字节的多少可分成两大类，即双字节（16位）指令和4字节（32位）指令，为了节省主存容量，加快指令的存取速度，但又为不致使机器对指令的处理过于复杂，对指令存储方式采取两种形式：双字节指令，每存储单元存放两条；4字节指令每存储单元存放一条。不允许将一条4字节指令分开存放于两个存储单元，为此，在程序中连续L型指令的条数为奇数条时，应补以空操作指令（双字节）。

指令处理部件对指令的排队处理就是将指令进行分析，将一单元中存放的两条短指令分开，以逐条进行处理。

2) 对变址型指令，形成绝对地址。本机器变址型指令（即B型指令）对操作数地址给出两个信息，即、变址寄存器单元号和相对地址 d 。此外，对于每道程序，管理程序都给出一个上、下界地址（ J_{sj}, J_{xj} ）。获得操作数绝对地址的方法：

$$D = (B) + d + J_{sj}$$

若为取数型指令，则将 D 送入先行，进而通过先行将该地址的数从主存取出并寄存于先行。若为存储型指令，则将绝对地址送入后行送数地址寄存器中，供存数选址用。

3) 为配合主存的工作,本部件的入口和出口采用了缓冲技术,即充分利用主存的空闲时间,将新指令从主存中取出,暂存于缓冲寄存器中,供指令排队时使用。将整理好的指令暂存于堆站中,作为进入运算部件的缓冲用。对取操作数和运算结果的存贮也采用缓冲技术,用先行取数寄存器存放操作数,用后行存数寄存器存放被存的运算结果。

4) 对某些特殊指令进行一些必要的处理。

2 基本组成

指令控制部件的总结构是考虑了上述各项要求,即尽可能地每拍处理一条指令;注意高速运算部件与低速存贮部件之间的匹配关系;加上个别指令的特殊要求,构成指令控制部件的总框图;见图 1-1 所示:

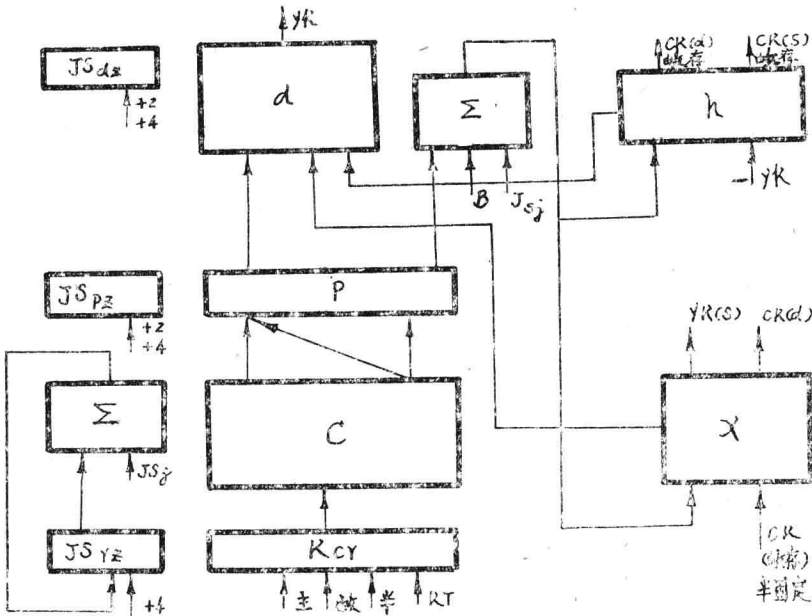


图 1-1 指令控制器总框图

它是由指令缓冲站、指令排队器、指令堆站、先行取数寄存器、后行送数寄存器和变址全加器等部分组成,各部分的功能及构成后面介绍。

3 基本工作过程

为了能对指令控制部件有一个完整的基本的了解,现将指令控制部件的基本工作过程简述如下:

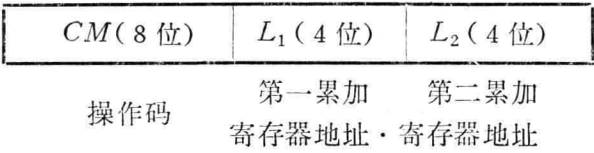
根据对指令部件提出的要求,将存于主存中的指令取出,进行适当的排队整理后以标准的形式送入运算部件,同时准备好指令所要求的操作数,为此我们将指令分步进行处理。

首先,由缓冲站入口根据缓冲站的可容情况,由 JS_{rz} 提供指令地址向主存发取指令请求命令,在得到请求回答信号后,改变 JS_{rz} 状态(+4),在缓冲站可容情况($JS_{rn} < 12$)下,继续发取指令请求命令,直至当缓冲站不可容时($JS_{rn} = 12$),停止发请求,主存读出的指令是根据请求次序顺序到来,并依次送入缓冲站中。

指令进入缓冲站后，排队器就可对缓冲站中的指令依次进行排队整理，它是根据已排指令的形式决定被排指令从何开始，而后根据被排指令的形式决定被排指令由缓冲站输出开关向排队器传送的方式，从而完成指令的排队整理工作。

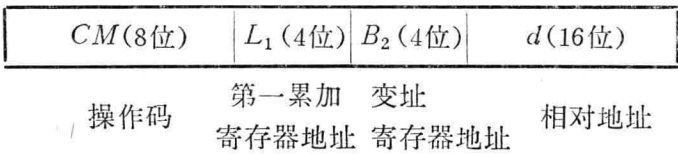
获得了完整的指令，即可开始对各指令进行不同的处理。本机指令分为四类（详见指令系统）：

*L*型指令，由 16 位组成，形式如下：

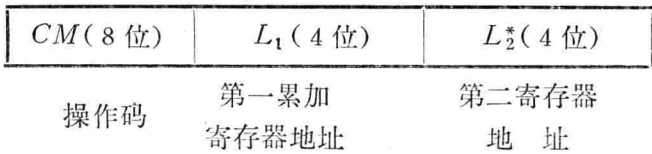


此类指令，指令控制部件不必作更多的整理工作，指令所需的操作数完全由累加寄存器提供。

*B*型指令，由 32 位组成，形式如下：



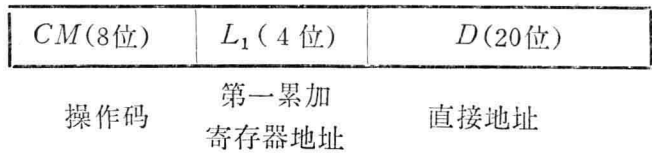
对此类指令的处理步骤为：首先由(*B*₂)、*d* 和上界寄存器 (*J*_{*s**j*}) 的内容在变址加法器上求出绝对地址： $D = (B_2) + d + J_{sj}$ ，进而判别 *D* 是否越界，若越界，则发越界中断，否则，将 *D* 分别送入先行取数寄存器中（取数型指令）或后行送数地址寄存器中（存数型指令），再将寄存地址的先行（后行）寄存器的单元号送入堆站组成 *L*₂^{*}，代替原来的 *B*₂ 和 *d*，构成如下的指令的统一形式暂存于堆站：



取数地址送入先行取数寄存器后，即可根据送入的地址向主存（或数存）进行请求取数，数读出后即存入本单元寄存器（此时地址被送入的数取代。）

存数地址送入后行送数地址寄存器后，尚需待运算部件执行完该指令后，将被存结果送入对应的后行送数寄存器之后，再向主存请求存数。

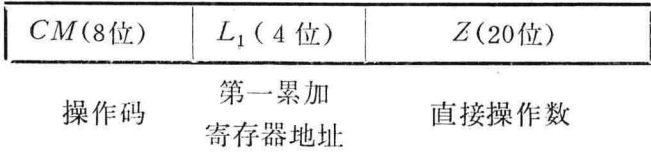
*D*型指令，由 32 位组成，形式如下：



该类指令与 *B* 型指令基本相同，区别仅在于其指令本身提供直接取数（存数）地

址，不需象 B 型指令那样由 B_2 和 d 求得。其它完全与 B 型相同。

Z 型指令，由 32 位组成，形式如下：



这类指令的 Z 部分为直接操作数，因此不需向主存请求取操作数，在处理该类指令时，只需将 Z 直接送入先行取数寄存器（已是操作数），待运算部件执行该指令时使用。

经过排队器处理过的指令以统一的形式送入指令堆站，待运算部件需要时，指令连同所需的操作数一起送入运算部件。

§ 1.2 指控基本部件

1 入口部件：

入口部件主要担负指控部件向主存（和半固定存储器）请求取指令，并选择处理开门、关门、 D 型无条件转移和 D 型转子等指令，最后将接收来的指令送入缓冲站。

为了完成上述功能，又考虑尽可能节省器材，减少指令处理时间，入口部件采用开关形式直接将主存（或半固定存储器）来的指令传送给缓冲寄存器，其各部分由下列组成，逻辑框图如图 1-2 所示：

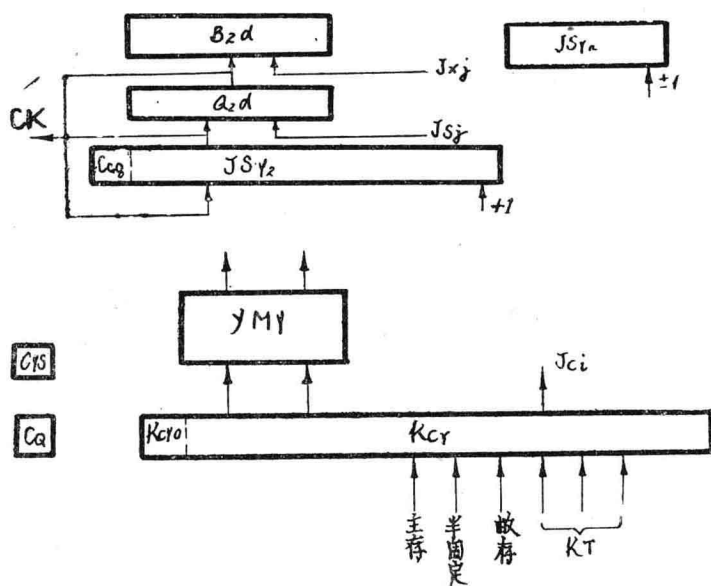


图 1-2 入口部件逻辑框图

(1) 指令入口开关 K_{cr} ：

由六或开关逻辑线路组成。用以接收主存、半固定存储器来的指令；在分机调整时

还能接收数存来的指令；在分部件调整时还可接收操作面板的指令（占三个通道）。

(2) 有无指令到来的标志位 K_{cro} ：

K_{cro} 用作标志 K_{cr} 上的信息可不可以送至缓冲寄存器中：

$$K_{cro} = \overline{K_{cro}'} \cdot \overline{C_{rs}} \cdot \overline{C_{cx}}.$$

其中： K_{cro}' 由各输送指令部件的发送信号组合而成，只要 K_{cr} 的输入通道有一路有送指令信号， K_{cro}' 就为高电位，表示有指令到 K_{cr} ， $\overline{C_{rs}}$ 、 $\overline{C_{cx}}$ 分别表示不执行转移动作和不是小循环状态的标志，即当 K_{cro}' 、 $\overline{C_{rs}}$ 和 $\overline{C_{cx}}$ 均为高电位时，标志着 K_{cr} 有新指令且可以送至缓冲站。

(3) 入口指令计数器 JS_{rz} ：

由标准计数器组成见附录一。共 18 位，表示主存字地址，存放被取指令的绝对地址。

当 K_{cr} 之内容为关门或 D 型无条件转移指令时，它直接接收 K_{cr} 上的关门或 D 型无条件转移指令所给定的地址。在处理关门或 D 型无条件转移以外的转移指令及中断恢复时可通过开关 K_{zk} 接收来自排队指令计数器 JS_{pz} 、先行输入开关 K_{xr} 和先行输出开关 K_{xi} 的内容。在正常情况下，它每得到一个请求取指令回答信号即“+1”以便请求下一单元的指令。“+1” JS_{rz} 的控制逻辑式见“逻辑控制式入口部分”第 6 条（以下称〈控、入〉6）。

(4) 请求主存或半固定标志 C_{cq}

它是一个触发器构成的标志位（借用组成 JS_{rz} 的标准计数器插件的一位）。其“0 状态”表示向主存请求；“1 状态”表示向半固定请求。其信息的产生在排队器部分。

(5) 入口可逆计数器 JS_{rn}

由标准计数器组成，共四位计 0~12 共 13 种状态。它表示缓冲站中应有指令的单元数（并非实际已有指令的单元数）。其作用是控制是否继续向主存（或半固定存储器）请求取指令，当获得一个请求回答之后就“+1” JS_{rn} ，当缓冲站给排队器送一个单元指令后就“-1” JS_{rn} 。当 $JS_{rn}=12$ 时，表示缓冲站已占满，不可再请求。 JS_{rn} 的“+”、“-”控制式见〈控、入 9,10〉。

(6) 取指令地址加法器 Q_{zd} ：

由全加器线路组成见附录三。它仅在转移指令的第二拍将 JS_{rz} 的高 12 位与 J_{sj} 相加求得绝对地址送给 JS_{rz} ，同时利用 Q_{zd} 的高位进位信号判断是否越界（有进位表示越界）。控制式见〈控、入 21,22〉。

(7) 指令地址比较器 B_{zd} ：

由全加器线路组成，完成 $(J_{xj} + Q_{zd})$ 反动作，用 B_{zd} 的进位信号判断 Q_{zd} 是否超越 J_{xj} ，有进位代表不越界，无进位代表越界。见〈控、入 21,22〉

(8) 入口双拍控制触发器 C_{rs} ：

入口执行成功性转移指令时，需清除已请求但还未到来的那些指令，即入口、存控都需要一个转换时间，完成切断请求和封锁即将到来的指令。同时，因转移指令的相对地址还需变为绝对地址放入 JS_{rz} 。这些动作都需两拍才能完成，因而用 C_{rs} 进行分拍。

C_{rs} 的控制见〈控、入29,30〉

(9) 入口指令奇偶错控制触发器 C_Q :

在入口设有从存控到入口传送指令奇偶错检查,这个地方一旦出了毛病,机器就没有自身检查的可能,为此,在入口用硬件设法将机器偶然跳动引起的奇偶错去掉,即加 C_Q 判断,若连续两条指令均发生奇偶错时,发中断信号。当入口发现第一次奇偶错时: $Z“1”C_Q$, 清缓冲站,按 JS_{pz} 的内容重新请求。当新请求的指令到入口,若仍奇偶错,发中断;若无奇偶错, $Z“0”C_Q$, 认为第一次奇偶错是偶然错误。入口按新的请求继续工作即可。

入口部分的基本工作过程:

1) 判断缓冲站是否有空闲单元,即 $JS_{rn} \neq 12$ 若 $JS_{rn} = 12$, 则根据 JS_{rz} 的内容和 C_{cq} 状态决定向主存(或半固定存储器)请求取指令。

2) 主存(半固定存储器,以后略)根据请求信号进行排队,若该单元可读时,发出请求回答信号,该信号到入口完成“ $+1JS_{rz}$ ”和“ $+1JS_{rn}$ ”工作。再根据 JS_{rn} 的状态判断是否可发出第二个请求。

3) 在请求回答信号发出一定时间后,主存随之读出指令,发出来指令信号同时指令信息到 K_{cr} , 并判断是否奇偶错。

4) 指令到达 K_{cr} 后,判断是否为开门、关门、 D 型无条件转移和 D 型转子指令。若不是,将指令传送给缓冲寄存器。若是,则分别完成:

a) 若为关门、 D 型无条件转移和 D 型转子指令,作 $K_{cr13-30} \rightarrow JS_{rz}$. 进而根据机器状态决定是否通过 Q_{zd} 获得绝对地址。

b) 若为开门指令,作 $Z“1110”J_{CQ}$ (其作用见后面小循环形成)。

c) 将指令送入缓冲寄存器。

其它形成小循环的动作;以后待续。这里需要说明的是,本机主存采取 M_8 (或 M_4) 方式工作,各模块对一个请求源来说是至少相差一拍进行工作,指控入口向主存请求,每拍只请求一个单元,待获得请求回答之后,再请求下一单元指令。而由于主存工作周期较长,主存发出请求回答到真正发出指令,有一段时间间隔,在这段时间里请求是一直在进行,因而对缓冲站来说,请求到的个数与真正得到的指令数之间往往有一个差值,为了保证凡主存发出的指令,缓冲站一定有单元存放,故加 JS_{rn} 以控制请求个数。由于主存工作周期一定,因而主存给出指令也一定是按照请求次序进行的。

再需要说明的一点是:从图 1-2 中看出,入口还有从数存来的通道。该通道只有在本部件与运控联调时使用,在未与主存联调时,要与数存先联,此时,若不加该通道,指控与数存不便于工作,加此通道之后,指令、操作数均通过数存获得,便可以看成一个完整的机器。

2 指令缓冲站

指令缓冲站用作快速指令处理部件与慢速主存的时间缓冲匹配,为充分利用设备,缓冲站又兼作建立小循环的控制。

为实现上述功能,其由下列各部分组成。逻辑框图见图 1-3 所示: (图见下页)

(1) 指令缓冲寄存器 $J_{co \sim 11}$:

寄存器由简易单拍双门触发器组成见附录二，用以接收寄存 K_{cr} 的指令。

(2) 小循环标志 C_{cx} :

C_{cx} 为“1”，表示缓冲站已为小循环工作状态， C_{cx} 为“0”，表示为非小循环工作状态。

(3) 指令入口计数器 JS_{cr} :

由标准计数器组成，用以表示 K_{cr} 的内容传送到 J_{ci} 的单元号，除建立小循环时关门指令到来不“+1 JS_{cr} ”外， J_c 每接收一单元指令，即“+1 JS_{cr} ”， JS_{cr} 由 0~11 循环计数以保证 J_{ci} 顺序接收。具体控制见〈控、缓 8〉。

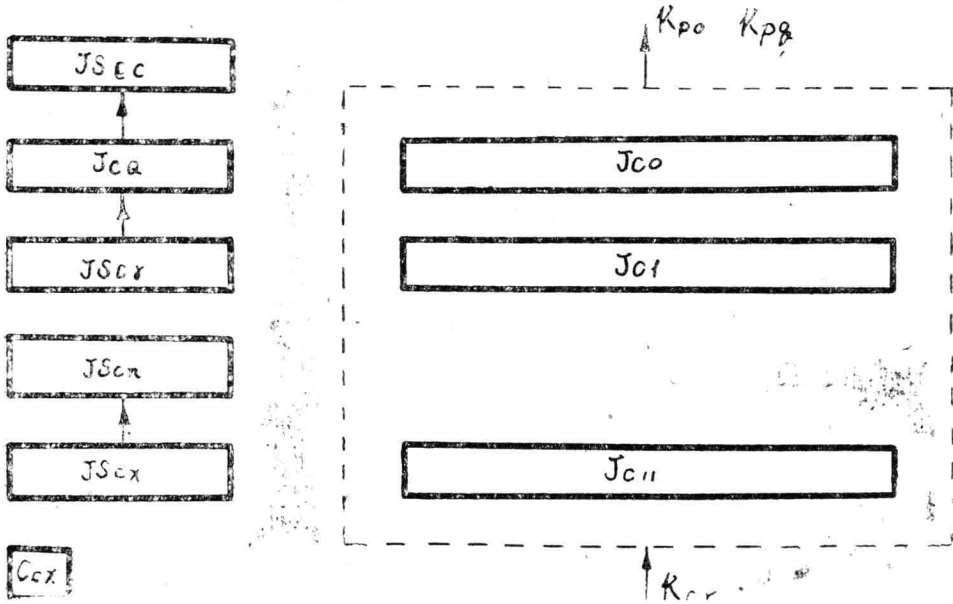


图 1-3 指令缓冲站逻辑框图

(4) 指令出口计数器 JS_{cc} :

由标准计数器组成，用以表示缓冲站中输出指令的单元号，每输出一单元指令后即，“+1 JS_{cc} ”， JS_{cc} 由 0~11 循环计数，以保证顺序输出：

$$+1JS_{cc} = t_{j-1} \cdot \overline{J_{cq}} \rightarrow JS_{cc}$$

(5) 小循环起始地址寄存器 J_{cq}

用标准计数器组成，用以记录小循环起始指令在缓冲站中的单元号，在小循环程序执行一周后恢复到该单元。其内容是在形成小循环过程中，当循环的第一条指令进入缓冲站时由 JS_{cr} 传入 J_{cq} 。

$$JS_{cr} \rightarrow J_{cq} = (J_{cq} = 1110) \cdot K_{cro} \cdot \text{开门}_r \cdot \text{清缓冲站}$$

(6) 缓冲站可逆计数器 JS_{cn}

由标准计数器组成，用以记录缓冲站实际保存的指令单元数。 JS_{cn} 记录的只是应有的（实有的加上已请求但未进入缓冲站）指令单元数。当循环预报时需要知道缓冲站中实际有的单元数，以便于实现继续请求的控制。另外，在入口处理转移指令（包括关

门指令)时,该转移指令后面已请求的指令应全部作废,此时, JS_{rn} 的内容就不确切,故:需要 $JS_{cn} \rightarrow JS_{rn}$.

(7) 小循环所用单元计数器 JS_{cx} :

由标准计数器组成,设置的原因仍然是由于循环预报需要在循环情况下知道循环程序在缓冲站中所占的确切单元数,在一般情况下, JS_{cn} 表示的为缓冲站中实有指令单元数。而在建立小循环过程中,缓冲站也在不断地给排队器输送指令(包括循环程序在内)这样,当小循环建立时, JS_{cn} 所表示的内容为循环程序在缓冲站未向排队器输送的指令单元数,而不是小循环的实际指令单元数。加了 JS_{cx} 后,在小循环建立过程中,就记录循环指令单元数,从 $Z“1110”J_{CQ}$ 开始到 $Z“1”C_{cx}$ 止。 $Z“1110”J_{CQ}$ 时 $Z“0001”JS_{cx}$, $Z“1”C_{cx}$ 时 $JS_{cx} \rightarrow JS_{cn}$ 使 JS_{cn} 在建立小循环之后记录小循环的真实条数。

指令缓冲站的基本作用是供指令缓冲用,其控制过程为:

1) 当 K_{cro} 为1时,向缓冲寄存器输送指令。指令的输入单元受 JS_{cr} 控制,利用 JS_{cr} 的顺序改变,顺次向各寄存器存放指令。

2) 指令缓冲站中的指令由 JS_{cc} 控制输出,由排队器选择接收,每取出一单元指令就“ $+1JS_{cc}$ ”,以控制顺序输出。

由于 JS_{cr} 和 JS_{cc} 的联合控制,避免了指令在缓冲寄存器中的流动,从而简化了寄存器的结构,提高了可靠性。

指令缓冲站的另一个重要作用是建立小循环,其控制过程为:

在通常情况下,当 K_{cr} 为“开门”指令时,则表示即将进入循环程序,此时, $Z“1110”J_{CQ}$,再来指令时,由于 $J_{CQ}=1110$,表示为循环的起始指令,作 $JS_{cr} \rightarrow J_{CQ}$,此后不断接收指令,直至关门指令到来,即表示循环程序结束,若该循环程序长度在缓冲站寄存范围之内,则关门指令到来之后,即建立小循环($Z“1”C_{cx}$),停止继续取指令($Z“1100”JS_{rn}$),若循环程序长度超过缓冲站寄存的范围,则不能建立小循环,在关门指令到来之前就出现 $JS_{cr}=J_{CQ}$ (表示循环程序已大于12单元),此时, $Z“1111”J_{CQ}$,撤消建立小循环的预备状态。

若在执行循环程序过程中出现中断,在中断恢复时,由于开门指令已执行完毕,不再出现,故不能应用上述办法实现小循环的恢复。为此,采用如下办法:利用关门指令的特点建立小循环,当关门指令到达 K_{cr} 之后,若 $J_{CQ}=1111$,表示在此之前,没有出现开门指令或是处于大循环状态,此时作建立小循环的预备状态, $Z“1110”J_{CQ}$,以下的动作与上述相同。这样,若是大循环程序,则会反复执行 $Z“1110”J_{CQ}, JS_{cr} \rightarrow J_{CQ}, Z“1111”J_{CQ}$.而对于中断之后恢复小循环状态便可以得到解决。

小循环程序中出现转移指令时,按大循环处理。

对于多重循环,即连续几个“开门”指令进入缓冲站,始终以后来者先执行。连续几个“关门”指令到达缓冲站时,则自然是先执行前者,然后由前到后,逐个执行。

3 排队器

排队器的作用,主要负责对来自缓冲站的指令进行编排并作适当处理,其基本作用可归结为两大部分:

1) 整理指令:

(4) 缓冲站输出开关指令译码器 $YM_{p1}, YM_{p2}, YM_{p3}, YM_{p4}$:

译出被排指令为 L 型还是非 L 型, 用以控制 J_p 接收和 JS_{pz} 的“+2”还是“+4”。

(5) 排队寄存器有无新指令标志触发器 C_{pz} :

用于标志 J_p 有无新指令待处理, $C_{pz}=1$ 为有新指令, 为 0 时, 表示 J_p 已无新指令。具体的控制见〈控、排 9, 10〉。

(6) 排队指令计数器 JS_{pz} :

由标准计数器组成, 共 19 位。用以表示即将进入排队器的指令所在的相对地址。根据判断接收的指令是 L 型还是非 L 型进行“+2”或“+4 JS_{pz} ” (JS_{pz} 表示字符地址)。在执行“转移类型”和中断恢复指令时它通过 K_{zk} 接收 K_{xr} 和 K_{xt} 的转移地址。

(7) 排队器双拍控制触发器 C_{ps} :

用六门触发器构成, 用以对在排队器执行双拍的指令的分拍控制。

(8) 变址寄存器 J_B :

由标准计数器组成, 其作用是: a) 暂存变址量 (M 和 A), b) 在执行输入指令时, 供寄存和修改地址用。

变址量是从通用累加寄存器取来, 在这里作中间寄存的主要目的是为了减少排队器执行 B 型指令的逻辑控制级。设了 J_B 之后: 根据 K_p 上的指令决定选择累加 (变址) 寄存器内容, 在 J_p 接收该指令的同时将变址量取入 J_B , 待该指令在 J_p 执行变址时, 变址量就直接从 J_B 中获得。否则, 在不设 J_B 情况下, J_p 执行变址操作时, 首先要根据 J_p 指令进行译码、选址, 取变址量, 而后实现变址, 这样在一拍中的逻辑控制级数大大超过总体要求的级数。增加 J_B 之后, 虽然器材增加, 但换来的逻辑控制简单和可靠性的提高。

(9) 排队器开关 K_p :

它是汇总 K_{po}, K_{pq} 的操作码和 L_2 部分内容, 共 12 位, 主要用来选择变址量。

(10) 变址加法器 Q_{b1}, Q_{b2} :

由全加器逻辑线路组成参见附录三。主要用来求得取数 (或存数) 的绝对地址, 它的原始数据来自 J_B, J_p 和上界地址寄存器 J_{sj} , 同时, 根据所得结果判断是否越上界。

(11) 变址比较器 B_{sd} :

由全加器线路组成。实现 Q_{b2} 所求得绝对地址和下界地址寄存器 J_{xj} 比较、判别 Q_{b2} 内容是否越下界。

(12) M_3 余数比较设备:

在调机或调程序过程中, 难题之一就是程序乱套, 造成的原因: 如果是设备问题, 则主要又在于变址设备。为了很好监视 Q_{b1}, Q_{b2} 的工作, 采用了对 Q_{b1}, Q_{b2} 进行 M_3 余数检查, 即如果工作正常, 则 Q_{b2} 的 M_3 余数应等于加数、被加数 M_3 余数和的 M_3 余数, 若不等, 则必有错误, 用以进行动态监视。 M_3 余数检查的具体实现见附录四。

(13) C'_{ps} :

主要用来控制在机器改变状态过程中, 不再发出第二次改变状态的请求。当由于中断, 进管条件满足 (t_{i4} 成立) 的情况下, $Z=1$ 时 C'_{ps} , 当新的程序状态字取来之后, $Z=0$ 时 C'_{ps}, C_{ps} 为 1 时, 除执行进管动作和允许 $Z=1$ 中断位外, 指控部件不处理任何

指令。

(14) 半固定的使用控制:

半固定存储器容量为 32768×32 , 用字节地址表示只要 17 位地址, 具体使用规定: 对 B 型指令, 在指令的变址量中用 M (24 位) 的最高位判别: 为“1”, 访问半固定; 为“0”, 访问主存。对 D 型指令, 用 D 的高三位判别: 若全为“1”, 访问半固定, 否则, 访问主存。获得的地址送入先行时, 占用 K_{xr9} 作标志:

$$\rightarrow K_{xr9} = B_p \cdot J_{B9} + D_p \cdot J_{p13} J_{p14} \cdot J_{p15}$$

概括起来, 排队器工作分为两步:

1) 指令的排队整理:

首先根据 JS_{cc4} 和 JS_{p219} 确定选择在 K_{p0} 和 K_{pq} 中的指令, 进而根据所选择的 K_{p0} 和 K_{pq} 中的指令决定给 J_p 的接收方式。

2) 指令的处理:

对于指令的处理, 首先根据 K_p 判断是否为变址型指令, 若是, 则选择累加寄存器的变址量送入 J_B , 然后对 J_p 上的指令进行处理:

a) 若为 B 型指令, 作 $D^* = J_B + d_2 + J_{sj}$

b) 若为 D 型指令, 作 $D^* = D + J_{sj}$

c) 根据不同指令作些特殊处理。

随后将获得的地址或数送入先行或后行, 进而形成第二地址送入 K_{dr} , 最后通过 K_{dr} 将 K_{dr} 上的标准形式的指令送入 J_{d0} 。

4 指令堆站

指令堆站是指控部件与运算部件的构通部件, 缓冲取操作数的时间, 待操作数取来, 便为运算部件提供条件完全具备的指令。另外, 堆站出口也是部分指令的处理部件。如对开门、关门和转移等吸收型指令的处理。指令堆站的另一个重要作用是: 指令执行过程中的平滑作用, 在排队器执行转移等指令时, 对缓冲站中原取来的指令全部作废, 重新请求取指令, 在请求取指令过程中, 运算部件的指令来源, 全靠堆站的积蓄, 以避免运算部件处理指令的中断。从而提高了机器的运算速度。

指令堆站由下列各部分组成, 逻辑框图如图 1-5 所示:

(1) 指令寄存器 J_d :

由简单元拍双门触发器组成, 共 6 个单元, 每单元 18 位, 用以寄存指令及字符地址。

(2) 堆站指令入口开关 K_{dr} :

它根据不同的指令, 选取来自不同部件的内容汇集成标准的指令形式。 K_{dr1-12} 除 K_{dr1-8} 在犯规中断产生时置“犯规”操作码外, 均由 J_{p1-12} 直接传送, $K_{dr13-16}$ 则分别由 J_{p13-16} (L 型)、 JS_{xr} (取数型)、 C_{hr} (存数型) 等传送。

(3) 堆站输出开关 K_{dc} :

根据 JS_{dc} 的控制, 分别将堆站各寄存器的内容通过 K_{dc} 将指令在出口执行或送入运算部件, 其中 K_{dco} , 表示 K_{dc} (实为堆站) 有无可供执行的新指令。

(4) 堆站入口计数器 JS_{dr} :

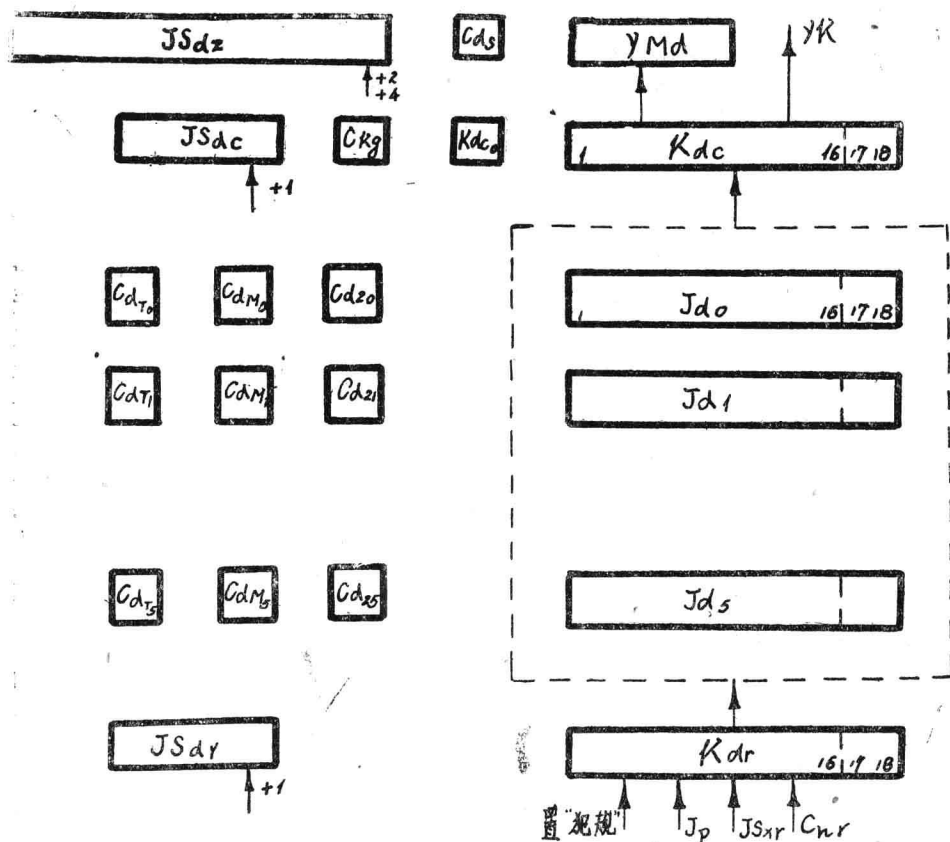


图 1-5 堆站框图

由标准计数器组成，用以控制来自排队器的指令送入堆站指令寄存器的顺序。 JS_{dr} 由 0~5 循环计数。堆站每接收一条指令就“+1 JS_{dr} ”。

(5) 堆站出口计数器 JS_{dc} :

由标准计数器组成，用以控制指令从堆站输出的顺序。它与 JS_{dr} 配合以保证程序中指令的执行顺序。

(6) 出口指令计数器 JS_{dz} :

由标准计数器组成，共 20 位，其中地址 19 位，表示半字地址。半固定和主存的标志位一位。它表示即将由堆站输出的 (K_{dc} 的指令) 指令所在地址，以供转子和中断恢复时用， JS_{dz} 本身属于程序状态字 (PSW) 的一部分。堆站每输出一条指令就 +2 (L 型指令) 和“+4 JS_{dz} ” (非 L 型)，(上述“+2”和“+4”都是以字节地址为单元)。若输出“转移型指令，则将转移地址由 $K_{xt} \rightarrow JS_{dz}$ ” (由 K_{dc13} 控制转移与否)。

(7) 开门、关门指令标志触发器 C_{kg} :

为了保证循环时，关门指令判断的准确性，即关门指令在排队器执行时，堆站应既

无开门也无关门指令,专设 C_{Kg} 作标志。当开门、关门指令由排队器进入堆站时, $Z“1”C_{Kg}$ 。开门、关门指令由堆站输出时, $Z“0”C_{Kg}$ 。当 $C_{Kg}=1$ 时,排队器出现开门或关门指令,则不允许该指令在排队器处理。

(8) “修改 M ”指令标志位 $C_{dM0} \sim C_{dM5}$:

为提高变址效率,需对“修改 M ”指令作特殊处理(见§1-3的5),这样有必要标志出堆站寄存器所放的指令是否为“修改 M ”指令。当排队器给堆站寄存器送到的是“修改 M ”指令时,将对应的 $C_{dMi}Z“1”$,当该单元的指令从堆站输出时, $Z“0”C_{dMi}$ 。

(9) 堆站有无指令标志触发器 $C_{dz0} \sim C_{dz5}$:

主要用于标志堆站各寄存器是否有新指令。“1”表示有新指令,“0”表示无新指令。 J_{di} 接收新指令时, $Z“1”C_{dzi}$; J_{di} 中指令送出堆站时, $Z“0”C_{dzi}$ 。

(10) $Z“CT”$ 指令标志位 $C_{dT0} \sim C_{dT5}$:

由于堆站有无 $Z“CT”$ 指令,影响排队器中条件转移指令的执行。排队器条件转移指令是要在堆站,运算器均无 $Z“CT”$ 指令情况下才能执行。堆站接收一条 $Z“CT”$ 指令就将与该单元对应的标志位 C_{dT_i} 置“1”。该单元指令出堆站时,对应该单元的 C_{dT_i} 置“0”。

(11) 双拍控制触发器 C_{ds} :

某些指令,如开门指令等,在堆站出口需两拍才能执行完,故用 C_{ds} 进行分拍控制。

(12) 堆站指令译码器 YM_d :

由于有些指令是在堆站出口执行,有些送入运算器的指令在堆站出口要作一些处理。故需判别 K_{dc} 上所放置的是什么指令。此判别由 YM_d 来实现。

(13) 字符地址的寄存:

运算部件执行字符运算指令时,需要知道字符地址,为此,在堆站中,从 K_{dr} , J_{di} 到 K_{dc} 均增设 17,18 位作为字符地址的寄存。 $K_{dr17,18}$ 接收 $Q_{b219,20}$, $K_{dc17,18}$ 同 $K_{dc1 \sim 16}$ 一起送至运算部件。

(14) 堆站与先行的连系:

堆站出口是指令在指控部件执行的最后一步,也是与运控部件的交接口。当指令由指控向运控部件发送的同时,也必须将该指令所用到的有关信息——进运控部件的指令的操作数和转移型指令的转移地址等一起准备好。因此,凡用到先行的指令,出堆站的条件之一就是先行将其信号准备好没有。判断的方法是:堆站出口指令的第二地址 ($K_{dc13 \sim 16}$) 所指的先行单元(双倍字长运算指令应是 $K_{dc14 \sim 16}$ 和 $K_{dc14 \sim 16+1}$ 两个先行单元)有无数,有数表示已准备好。

(15) 运算部件可取指令的条件 t_{j-6} :

指控部件与运控部件交接指令的方法是:堆站出口的指令若与运控部件有关,首先形成一个“运算部件可取的条件(指控部件已准备好) t_{j-6} ”,并告知运控,此时运控若能接收,则返回一个接收信号(运 A 接收或运 A' 接收),由此信号进一步合并形成指令出堆站的条件 t_{j-5} 。

t_{j-5} 的实际使用逻辑式见指控部件实际使用逻辑式,这里将其含义介绍一下: