工业控制计算机及其应用

于常友　张宝臣　编

目　　录

第 一 章　工业控制计算机系统介绍

概述：工业控制计算机除了包括一般数字计算机所具有的存储器、控制器、运算器和一般外部设备外，还具有一套工业外围装置，如：外部时钟装置、计数装置、外部中断装置、开关量输入输出装置、数字量输入输出装置和模拟量输入输出装置。可以应用这些工业外围装置来搜集工业现场的信息，显示工业现场的数据，监视工业现场各执行机构的工作情况和控制工业生产过程。

如图 1.1 所示，输入输出总线包括：设备地址选择线、数据输入线、数据输出线、控制线、状态报告线和请求线。系统包括：磁带机、人机对话打字机、过程打字机、外部时钟装置、计数装置、外部中断请求装置、模拟量输入和输出装置、数字量输入和输出装置及开关量输入和输出装置。

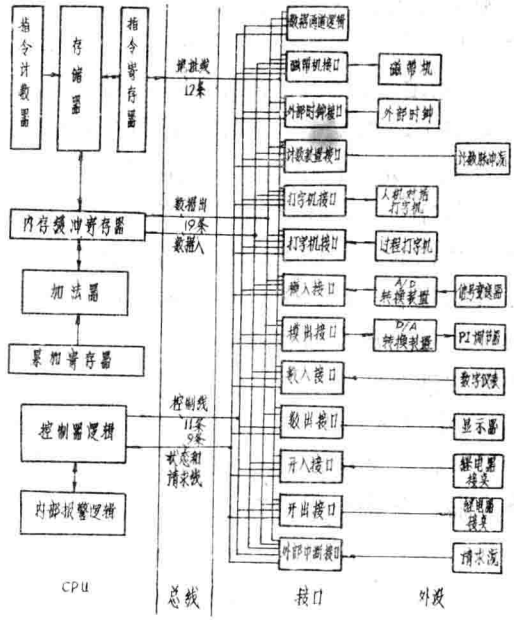


图 1.1

这是一个典型的工业控制计算机系统，它基本上包括了所有的工业外围装置。本章重点介绍该系统的中央处理装置、指令系统、主机与外设间信息交换的方式、中断的种类和操作控制台。这里介绍的计算机系统并不针对某一个特定的生产过程，它可以对各种不同的生产过程进行管理和控制。

1.1 中 央 处 理 装 置

如图 1.2 所示，A：累加寄存器，用于寄存参加运算的数和运算结果；B：扩展的累

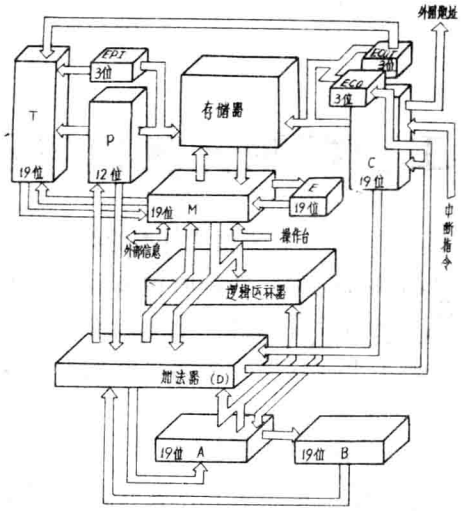


图 1.2

加寄存器，在进行乘除法运算和移位时使用；M：存储器缓冲寄存器、控制台、CPU或外设向存储器输入数据时，数据首先打入M，然后送存储器，存储器向控制台、CPU或外设输出数据时，数据亦首先打入M，然后送控制台、CPU或外设；P：指令计数器，程序开始执行前，要把程序的开始地址打入P，在程序执行的过程中，每执行一条指令，在取指令周期中，P的内容自动加1，指出下一条要执行指令的地址，在执行转移指令时，把要转移的地址送P；EPI：指令页面寄存器，该机存储器容量可扩充到32K字，分成八个（0—7）页面，每一页为4K字，EPI寄存器中始终存放执行指令所在的页面号；C：指令寄存器，要执行的指令总是在取指令周期内从存储器取出打入C，在执行指令的周期内，总是执行C寄存器中存放的指令；ECO：操作数页面寄存器，对于要访问存储器的指令，其操作数有效地址的页面号存放在该寄存器中；ECOT：亦是操作的数页面寄存器，该寄存器仅在间接寻址和中断处理时使用；X：变址寄存器，该机的变址寄存器是借助于存储器0页的0号单元，所以图中没有画出；T：断点缓冲寄存器，中断时把P，EPI，和ECO三个寄存器的内容都打入T暂时保存起来，中断处理程序再把T的内容送存储器指定单元保存起来；E：状态寄存器，该寄存器中存放着中断屏蔽状态字。

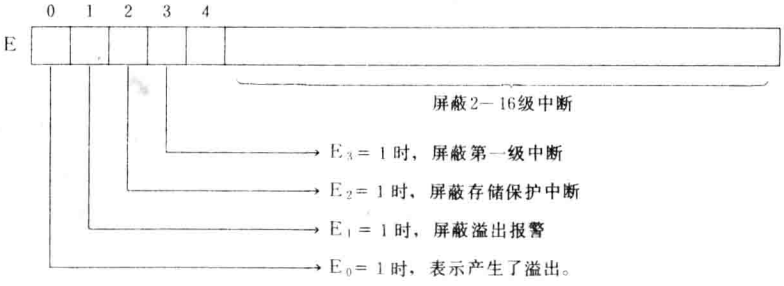


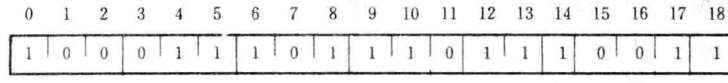
图 1.3

图1.3中标出了中断屏蔽状态字各位的功能。当中断屏蔽状态字为3777　774，则主机处于关中断状态，即主机不能响应任何中断级的请求。当状态字为0000　000时，则主机处

于开中断状态，即可以按优先级响应任何级别的中断请求。中断屏蔽是指某一级或某几级的中断被屏蔽，主机不能响应已被屏蔽级的中断请求，但可以按优先级响应没有被屏蔽级的中断请求。

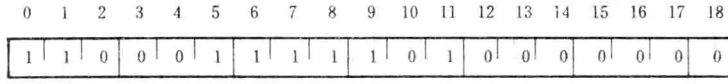
该机字长为19位。有长字和短字之分，长字19位，从第0位开始每三位用一个八进制位表示，最后，第18位为"1"表示是一个八进制位的4，为"0"表示是一个八进制位的0。短字12位，从第0位开始每三位用一个八进制位表示，共4个八进制位，12到18位不用。

例一　长字的表示法：



用八进制表示为：4356　714

例二　短字的表示法



用八进制表示为：6175

1.2 指　令　系　统

1.2.1　寻址方式：

机器指令包括两部分的内容，一部分是操作码，它决定指令的功能；另一部分是操作数，它可以是操作数本身，也可以是操作数所在的地址。所谓寻址方式就是找到为了实现操作码规定的功能所需要的操作数的方法。本机有五种寻址方式：

（1）隐含寻址：在这种寻址方式中，指令操作码的意义中包含有所需操作数的地址址（通常是寄存器的名字）。

例　AGB；A寄存器的内容送给B寄存器。

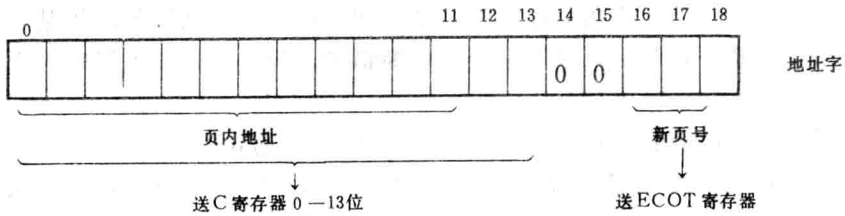
（2）立即寻址：指令的操作数部分就是操作数本身。

例　L_{DD}　X，n；n是一个12位的二进制数，执行这条指令，直接把n送到X寄存器的0—11位。

（3）变址寻址：指令操作数部分的内容是有效地址M的位移量，用d来表示，而有效地址M的基地址在变址寄存器X中。M＝X＋d。

例　设X＝1000，A＝7，执行L_D[X＋3]，A指令，则有效地址M＝(X＋3)＝1000＋3＝1003，于是A＝7送1003号单元。

（4）间接寻址：指令操作数部分的内容是一个地址码，记为m，用m来寻找有效地址M。间接寻址首先从内存里取出来的是一个地址字其格式为：



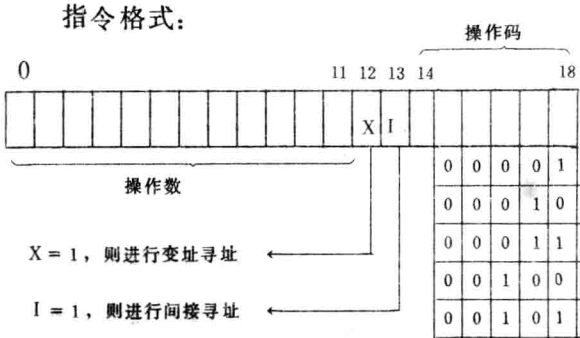
假设地址字既不要求变地址，也不要求间址，即第12位和13位都为0，则寄存器ECOT的内容就是操作数所在内存的页面号，而指令寄存器C的0—11位的内容就是操作数的有效地址M，这种情况下M=(m)。

例　设1020单元的内容为2000，而2000单元的内容为5。执行L_D A,(1020)指令，则M=(1020)=2000，于是(2000)=5送A。

(5) 直接寻址：在既不变址也不间址的条件下，指令操作数部分的内容就是操作数所在内存单元的地址，即有效地址M。

例　L_D A，M；则把M的内容送A。

指令格式：



操作码		指令名称
八进制表示	符号表示	
004	A_{DD}	加法
010	SuB	减法
014	AND	逻辑乘
020	OR	逻辑加
024	XOR	按位加
040	ADX	与X加
044	JP	无条件转移
050		
054		
060	EI	执行
064	HJ	停止后转移
100	L_D X, M	装X
104	L_D A, M	装A
110	L_D E, M	装E
114	MP	乘法
120	DV	除法
124	INC	增量
140	L_D M, X	存X
144	L_D M A	存A
150	L_D M, E	存E
154	L_D M, T	存T
160	LDE	清零

注：在寻址方式中，用寄存器的名字表示寄存器的内容，如：A，就表示A寄存器的内容；用字母M表示有效地址，(M)表示有效地址的内容；用(X+d)表示变址寻址的有效地址；在地址字不要求变址和间址的情况下，用(m)表示间接寻址的有效地址。

1.2.2 指令系统

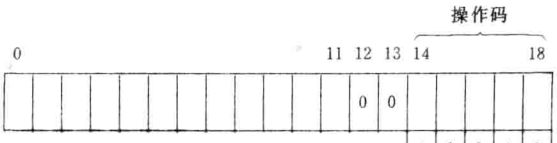
(1) 访问指令：即要求访问内存的指令。

指令功能：

指令名称	汇编指令码	指令功能
加法	ADD M;	A+(M)→A
减法	SUB M;	A−(M)→A
乘法	MP M;	A×(M)→A，B。A中存积的高位部分；B中存积的低位部分
除法	DV M;	A，B÷(M)→A，B。被除数高位部分在A中，低位部分在B中；结果：商在A中，余数在B中
逻辑乘	AND M;	A∧(M)→A
逻辑加	OR M;	A∨(M)→A
按位加	XOR M;	A⊕(M)→A
装A	L_D A, M;	(M)→A
装X	L_D X, M;	(M)→X
装E	L_D E, M;	(M)→E
存A	L_D M, A;	A→M
存X	L_D M, X; X;	X→M
存E	L_D M, E;	E→M
存T	L_D M, T;	T→M
与X加	ADX M;	$X_{0-11}+(M_{0-11})\rightarrow X_{0-11}$。X=0时跳步
无条件转移	JP M;	(M)→P，EPI，ECO。无条件转到M的内容所指定的地址
执行	EI M;	(M)→C。有效地址的内容送C，作为一条指令来执行。
停机后转移	HJ M;	(M)→P，EPI，ECO，然后停机，再启动时，从转移地址开始执行
增量	INC M;	$(M_{0-11})+1\rightarrow M_{0-11}$。当(M)=0时，跳步
清零	LDE M;	将有效地址M的内容清零

(2) 输入输出指令和隐指令：输入输出指令是和外设打交道的指令，又称访外指令。隐指令是中断响应后，由硬件形成的指令，该指令从外设接口打入指令寄存器C，并执行之。由于这类指令在程序中是看不到的，所以叫作隐指令。

指令格式：



操作码					操作码		指令名称
					八进制表示	符号表示	
0	1	0	1	0	0 5 0	OUT	输出
0	1	0	1	1	0 5 4	IN	输入
0	0	1	1	1	0 3 4	ICT	中断计数
0	1	1	1	1	0 7 4	IJP	中断无条件转移
1	0	1	1	1	1 3 4	IOUT	存储器输出
1	1	1	1	1	1 7 4	IIN	向存储器输入

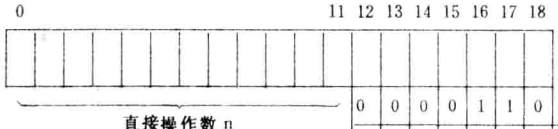
指令功能：

指令名称	汇编指令码	指令功能
输出	OUT M ；	A→M指定外设接口缓冲寄存器
输入	IN M ；	M指定外设接口缓冲寄存器的内容→A
中断计数	ICT M ；	$(M_{0-18})+1\rightarrow M$
中断无条件转移	IJP M ；	转M的内容所指定的单元去执行
存储器输出	IOUT M ；	(M)→高速外设缓冲寄存器
向存储器输入	IIN M ；	高速外设缓冲寄存器的内容→M

注：对于I/O指令中的有效地指M不在是存储器的有效地址了，而是外设的设备码。

(3) 立即寻址指令：此类指令的特点是指令的0—11位的内容就是直接参加运算的操作数。

指令格式



操作码								操作码		指令名称
								八进制表示	符号表示	
0	0	0	0	1	1	1	0	0 3 0	LDD X, n,	直接装X
0	0	0	1	1	1	1	0	0 7 0	ADXD	直接与X加
0	0	1	0	1	1	1	0	1 3 0	HJD	停机后直接转
0	0	1	1	1	1	1	0	1 7 0	JPD	直接无条件转
0	1	0	0	1	1	1	0	2 3 0	JPDP	A＞0转
0	1	0	1	1	1	1	0	2 7 0	JPDZ	A＝0转
0	1	1	0	1	1	1	0	3 3 0	JPDN	A＜0转
0	1	1	1	1	1	1	0	3 7 0	JZD	跳转并清“0”
1	0	0	0	1	1	1	0	4 3 0	MPD	直接乘
1	0	0	1	1	1	1	0	4 7 0	DVD	直接除
1	0	1	0	1	1	1	0	5 3 0	ADDD	直接加
1	0	1	1	1	1	1	0	5 7 0	ANDD	直接逻辑乘
1	1	0	0	1	1	1	0	6 3 0	ORD	直接逻辑加
1	1	0	1	1	1	1	0	6 7 0	XORD	直接按位加
1	1	1	0	1	1	1	0	7 3 0	LDD E, n,	直接装E
1	1	1	1	1	1	1	0	7 7 0	LDD A, n,	直接装A

指令功能：

指令名称	汇编指令码	指令功能
直接装X	L_{DD} X, n ;	$C_{0-11} \rightarrow X_{0-11}$
直接与X加	ADXD n ;	$C_{0-11} + X_{0-11} \rightarrow X_{0-11}$。当X = 0时跳
停机后直接转	HJD n ;	$C_{0-11} \rightarrow P$后停机，再启动，从转移地址执行
直接无条件转	JPD n ;	$C_{0-11} \rightarrow P$
正转	JPDP n ;	A＞0，则$C_{0-11} \rightarrow P$；否则顺行
零转	JPDZ n ;	A = 0，则$C_{0-11} \rightarrow P$；否则顺行
负转	JPDN n ;	A＜0，则$C_{0-11} \rightarrow P$；否则顺行
溢出转并清零	JZD n ;	当E_0= 1时，$C_{0-11} \rightarrow P$，并将E_0清“0”
直接乘	MPD n ;	$n \times A_{0-18} \rightarrow A, B$。结果：高位部分在A中，低位部分在B中。
直接除	DVD n ;	A, B ÷ n →。A，B被除数高位部分在A中，低位部分在B中；结果：商在A中，余数在B中。
直接加	ADDD n ;	A + n → A
直接逻辑乘	ANDD n ;	$A_{0-11} \wedge n \rightarrow A_{0-11}$
直接逻辑加	ORD n ;	$A_{0-11} \vee n \rightarrow A_{0-11}$
直接按位加	XORD n ;	$A_{0-11} \oplus n \rightarrow A_{0-11}$
直接装E	L_{DD} E, n ;	$n \rightarrow E_{0-11}$
直接装A	L_{DD} A, n ;	$n \rightarrow A_{0-11}$

注：表中$C_{0-11} = n_{。}$都表示指令0—11位的内容。

(4) 带扩冲操作码指令：此类指令只有一个操作码，其中的每一种指令由补助操作码决定。

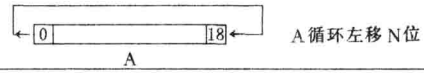
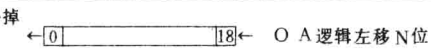
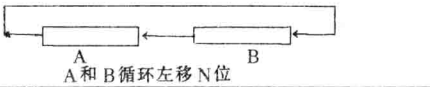
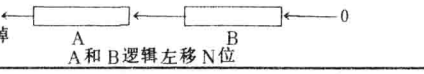
指令格式：

辅助操作码　　操作码

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	八进制表示	符号表示	指令名称
1	1	0	0	0	0	0	0	新页号			0	0	0	0	0	0	0	0	600␣ 0 0 0	COP N;	改变操作数页号
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2000 0 0 0	AGB ;	A→B
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000 0 0 0	BGA	B→A
0	0	移位指令				移位次数															
0	0	1	0	1	1							0	0	0	0	0	0	0	13␣␣ 0 0 0	RLCA	A循环左移
0	0	1	0	1	0							0	0	0	0	0	0	0	12␣␣ 0 0 0	RLAB	A·B循环左移
0	0	1	1	1	1							0	0	0	0	0	0	0	17␣␣ 0 0 0	LLA	A逻辑左移
0	0	1	1	1	0							0	0	0	0	0	0	0	16␣␣ 0 0 0	LLAB	A、B逻辑左移

注：表中“␣”表示是一个由程序人员填写的八进制位。

指令功能：

指令名称	汇编指令码	指令功能
改变操作数页号	COP N ；	C_{9-11}→ECO （即N→ECO）
A的内容送B	AGB ；	A→B
B的内容送A	BGA ；	B→A
A循环左移	RLCA N ；	
A逻辑左移	LLA N ；	
A和B循环左移	RLAB N ；	
A和B逻辑左移	LLAB N ；	

1.2.3 关于汇编语言的几点说明

用汇编语言表示的指令，通常叫作汇编语句，它包括：标号，操作码，操作数和注释。

例如：

标号	操作码	操作数	注释
LOOP：	ADD	1000；	A＋(1000)→A

标号指出了该语句在内存中的相对地址，其它语句可以用这个标号寻找该语句，标号的后面必须跟有冒号。操作码规定指令的功能。操作数与操作码之间要用空格分开，操作数的后面要跟有分号，分号后面是注释。关于操作码的汇编语言表示前面已经作了交待，下面仅对操作数的一些规定加以说明。

(1) **：表示现行指令所在的存储器地址；**－N：表示现行指令所在的存储器地址减N；**＋N：表示现行指令所在的存储器地址加N。其中N是十进制的整数。

(2) BO*：表示现行操作数页号。

(3) I ：表示简接寻址

(4) 用十进制表示操作数，数字的前面要带上正负号；用八进制表示操作数，则不需要加符号。

(5)；表示空指令，执行这样的指令，不做任何工作。

1.3 主机与外设间息交换的方式

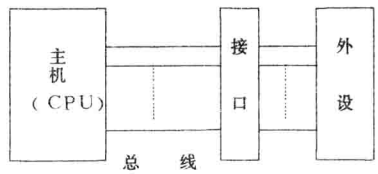


图 1.4

概述：如图 1.4 所示，CPU 的信息总是通过总线、接口，然后传送给外设；而外设的信息总是通过接口，总线，然后送给 CPU。两者间的信息交换又都是通过输入输出指令来实现的。所以首先介绍一下总线结构、接口的概念和输入输出指令的工作过程是有必要的。

总线结构如图1.5所示。

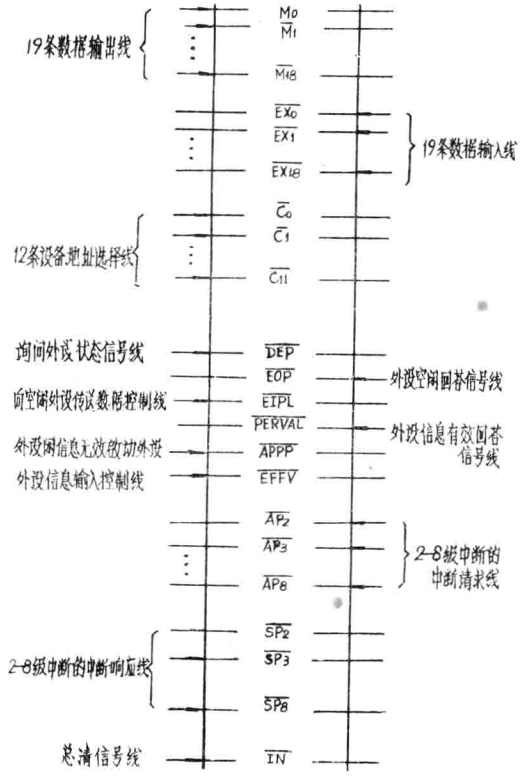


图 1.5

总线就是 CPU 和外设间输入输出数据和传输信息的一组传输线。一般说来总线包括：设备地址选择线、数据传输线、控制线、状态报告线、请求线和总清信号线。图 1.5 示出了一种典型的总线结构。

接口概念：

所谓接口就是主机和外设的交接部分。接口的主要功能是控制信息的交换方式、数据的传送形式和数据传送的通路。接口包括的电路有设备码译码器：译码选中要求进行信息交换的外设；数据缓冲寄存器：暂时存放准备交换的数据；设备状态触发器：寄存设备的现行状态，如设备空闲还是忙碌，信息有效还是无效；程序中断控制部件：控制用程序中断方式进行信息交换的逻辑部件；数据通道控制部件：控制用数据通道方式进行信息交换的逻辑部件。

输入输出指令的工作过程：

前面我们已经介绍了输入输出指令的格式和功能，现在假设输入或者是输出指令已经取到了指令寄存器 C（如图1.6所示）。对于输入输出指令，指令操作数部分的内容是外设的设备码，用来译码选中请求交换信息的外设。用指令的3—7位来指定输入输出单元，每一个输入输出单元包括16个接口；用指令的8—11位来指定接口，0 号接口作为通用接口，

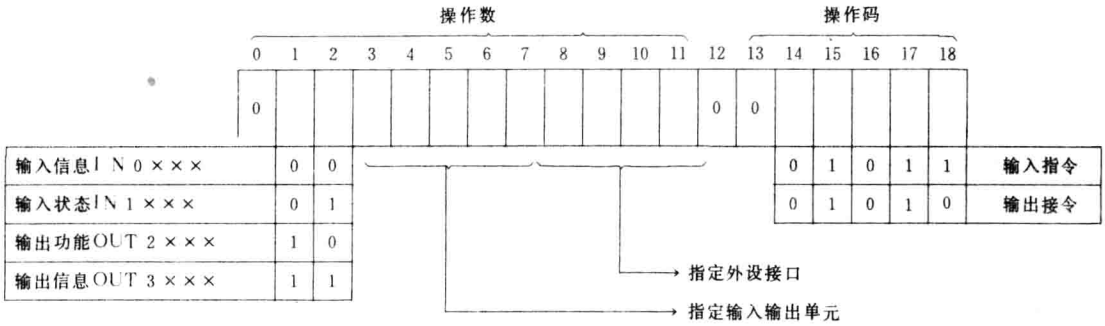


图 1.6

1—15号接口叫作外设接口，每一个外设接口可以带一个外部(围)设备；用指令的1，2位来指定输入输出的内容，包括：输入信息(IN 0 × × ×)，输入状态(IN 1 × × ×)，输出功能(OUT 2 × × ×)，输出信息(OUT 3 × × ×)。

如图 1.7 所示，每一个输入输出单元都有这样一个设备码译码器，第一级译码是多一

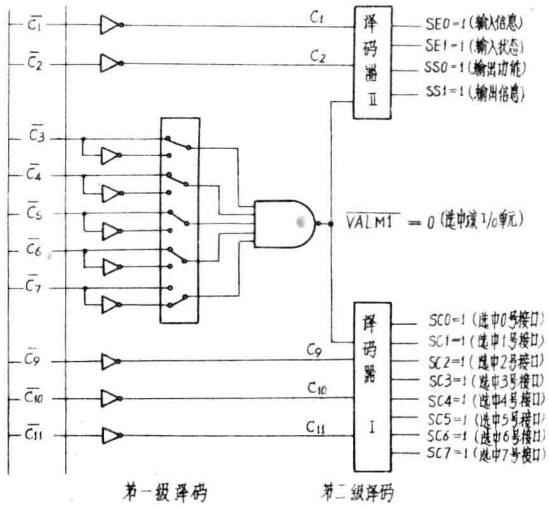


图 1.7

译码器，译码产生输入输出单元选中信号VALMi = 0。图 1.7 的连线，则选中 1 号输入输出单元，即产生VALMi = 0。第二级译码有两个译码器，均采用多一译码器，译码器生效的条件是$\overline{\text{VALMi}} = 0$。

输入			输出
C_9	C_{10}	C_{11}	
0	0	0	$SC_0 = 1$
0	0	1	$SC_1 = 1$
0	1	0	$SC_2 = 1$
0	1	1	$SC_3 = 1$
1	0	0	$SC_4 = 1$
1	0	1	$SC_5 = 1$
1	1	0	$SC_6 = 1$
1	1	1	$SC_7 = 1$

译码器Ⅰ的真值表

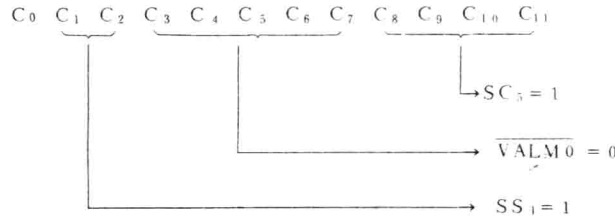
输入		输出
C_1	C_2	
0	0	$SE_0 = 1$
0	1	$SE_1 = 1$
1	0	$SS_1 = 1$
1	1	$SS_1 = 1$

译码器Ⅱ的真值表

译码器Ⅰ是用来译码选择外设接口的，一般情况下，一个I/O单元中的外设接口不超过7个，所以用C_9，C_{10}，C_{11}进行译码即可。

译码器Ⅱ是用来确定输入输出内容的。

例：执行OUT 3025指令，则指令0—11位的内容为：



设备码译码后，$\overline{VALM1}=0$，$SC_5=1$，$SS_1=1$，表示选中了1号输入输出单元的5号外设接口，并且进行信息输出。

输入信息：执行IN 0 ×××指令，就将选中外设接口中输入缓冲寄存器的内容通过输入数据线送CPU的累加器A。

输出信息：首先将要输出的信息送累加器A，然后执行OUT 3 ×××指令，就把A的内容通过输出数据总线送选中外设接口的输出缓冲寄存器，从而送外设。

输入状态：状态字存放在外设接口状态寄存器里，执行IN1×××指令，则把选中外设接口状态寄存器的内容通过输入数据线送累加器A。

输出功能：首先在累加器A中存放一个功能字，功能字各位的含意如下表所示：

0	1	2	3	A
1				$A_0=1$，把接口中的功能生效触发器置“1”，功能字生效。
	1			$A_1=1$，在功能字生效的情况下，去启动外设。
		1		$A_2=1$，把中断有效触发器置“1”，要求以程序中断方式进行信息交换。 $A_2=0$，要求以程序传送方式进行信息交换。
			1	$A_3=1$，打字机键盘输入有效。 $A_3=0$，打字机阅读器输入有效。

于是功能字＝6000；以程序传送方式启动外设

＝7000；以程序中断方式启动外设

＝4000；停止外设工作

＝6400；打字机打印输出

然后执行OUT 2 ×××指令，则把功能字通过输出数据总线送选中外设的接口，去设置接口中各功能触发器的状态，实现功能字所规定的功能。

例：在累加器A中装入功能字7000，执行OUT 2025指令，则以程序中断方式启动1号输入输出单元中的5号外设。

1.3.1 程序传送方式

(1) 定义：利用程序中的输入输出指令来完成主机和外设间的信息传送，称之为程序传送方式。

(2) 传送过程：如图1.8所示。

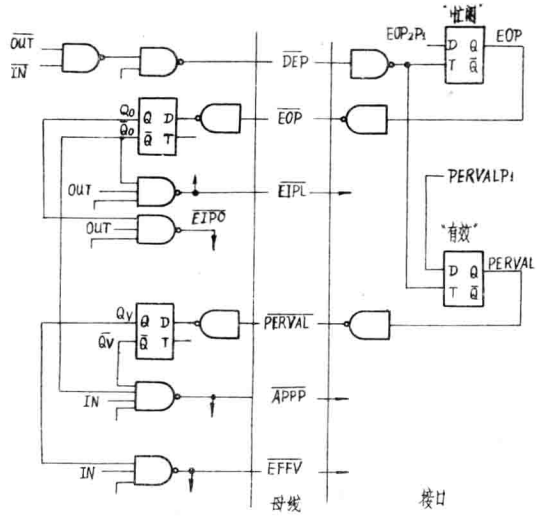


图 1.8

① 执行输出信息指令（OUT 3×××）的传输过程：执行OUT 3××× 指令，CPU首先发出询问外设状态信号 DEP = 1。若此时外设是处于忙碌状态（即$EOP_2P_1 = 1$），则把接口中的忙闲触发器置“1”，于是接口向主机发出外设忙碌信号EOP = 1（$\overline{EOP} = 0$），把主机的外设忙闲触发器Q_0置“1”。于是发出EIPO = 1信号，这就是指令的结束信号。虽然指令结束了，但要输出的信息并没有送到外设的接口缓冲寄存器。反之，若此时外设是处于空间状态(即$EOP_2P_1 = 0$)，则把接口中的忙闲触发器置“0”，于是接口向主机发出外设空闲信号EOP = 0（$\overline{EOP} = 1$），把主机的外设忙闲触发器Q_0置“0”。于是发出EIPL = 1信号，它一方面送外设，去打开接口缓冲寄存器的输入门，把总线上的数据打入接口缓冲寄存器；另一方面去控制指令计数器的内容再进行一次加“1”(指令计数器的内容在取指令周期中已经进行了一次加“1”)，使程序跳步。

② 执行输入信息指令（IN 0×××）的传输过程：执行IN 0×××指令，主机亦发出询问外设状态信号 DEP = 1，该信号不仅去询问外设忙闲（询问外设忙闲同输出信息指令），而且还去询问外设的信息是否有效。所谓信息有效即外设输入的信息已经打入了接口缓冲寄存器，此时外设向主机发出 $PERVALP_1=1$ 信号。DEP = 1信号的前沿正跳，把接口中的信息有效触发器置“1”，于是接口向主机发出信息有效信号PERVAL = 1，否则接口向主机发出信息无效信号PERVAL = 0。主机亦有一个外设信息有效触发器Q_V，当信息有效时，由PERVAL = 1把Q_V置“1”，于是发出EFFV = 1信号。这个信号一方面送接口，去打开接口缓冲寄存器的输出门，把要输入的信息打入总线。从而送CPU的累加器A；另一方面去控制指令计数器的内容再进行一次加1，使程序跳步。当信息无效时，由PERVAL = 0把Q_V置“0”。于是发出 APPP = 1信号，这个信号表示外设空闲但信息无效，它一方面送外设去重新启动一次外设，另一方面又是指令的结束信号，但要输入的信息并没有打入输入数据线。

③ 执行输出功能指令（OUP 2×××）的传送过程：执行OUT 2×××指令，主

机也要发出询问外设状态信号DEP＝1。当外设空闲时，外设向主机发出$\overline{EOP}$＝1，于是主机发出EIPL＝1，一方面送外设去控制将功能字的各位打入接口中各功能触发器，使功能生效；另一方面去控制指令计数器的内容再进行一次加1，使程序跳步。反之，若外设忙碌，则功能字不能被输出，这和执行输出信息指令的过程是一样的。

④ 执行输入状态指令（IN 1×××）的传送过程：因为外设接口状态寄存器的内容是在外设的工作过程中随机形成的，所以当执行输入状态指令时，外设接口中状态寄存器的内容总是有效的，因而执行一次IN 1×××指令就能确保把接口中状态寄存器的内容送累加器A。所以执行IN 1×××指令，总是要跳步的。

例一

```
0500: LDD   A, 6000 ;
   1: OUT       2025 ;   输出功能字6000，以程序传送方式启动打字机。
   2: JPD       0501 ;
   3: LDD   A, 1020 ;
   4: OUT       3025 ;   输出B的ASCII码1020，打字机打印B。
   5: JPD       0504 ;
   6: HJD       0500 ;   停机，再启动，从0500执行。
```

例二

```
0500: LDD   A, 6400 ;
   1: OUT       2025 ;   输出功能字6400，以程序传送和输入方式启动打字机。
   2: JPD       0501 ;
   3: IN        0025 ;   将一个键盘字符送累加器A。
   4: JPD       0503 ;
   5: LDD   A, 4000 ;
   6: OUT       2025 ;   输出功能字4000，停止打字机工作。
   7: JPD       0506 ;
0510: HJD       0500 ;
```

1.3.2 程序中断方式

利用程序传送方式来进行主机和外设间的信息交换。当要交换下一个信息时，首先要询问前一个信息是否交换结束，如果没有结束，程序需要循环等待，这样就降低了主机的效率；同时往往因为程序等待，而不能交换其他要求交换的信息，因而使这些信息失去了实时性。程序中断方式克服了上述两种缺点，当主机传送一个信息后，就转去执行其他程序，当外设把一个信息交换结束后，由外设提出交换下一个信息的要求（即申请中断）。主机接受到此请求后，马上停止正在执行的程序，转到给该外设交换信息的程序（即中断处理程序）去执行，中断处理程序执行完毕，再转到被中断的程序去继续执行。这就避免了主机对外设的询问和等待，从而提高了主机的效率；同时可以把要求交换的各种信息按优先级进行排队，使每一种要求交换的信息都不失实时性。

（1）定义：同主机交换信息的设备发出中断请求，主机响应中断后，转中断处理程序，在中断处理程序中进行信息交换。

（2）程序中断的过程：

如图1.9所示，本系统共有八级中断：第一级是计算机内部故障中断；第二级是数据通道中断；第三级和第四级不用；第五级是时钟中断；第六级是计数中断；第七级是程序中断；第八级不用。下面仅对程序中断进行分析：

1　首先必须输出功能字，以程序中断的方式启动所需要的外设：

```
LDD     A,     7000  ;
```

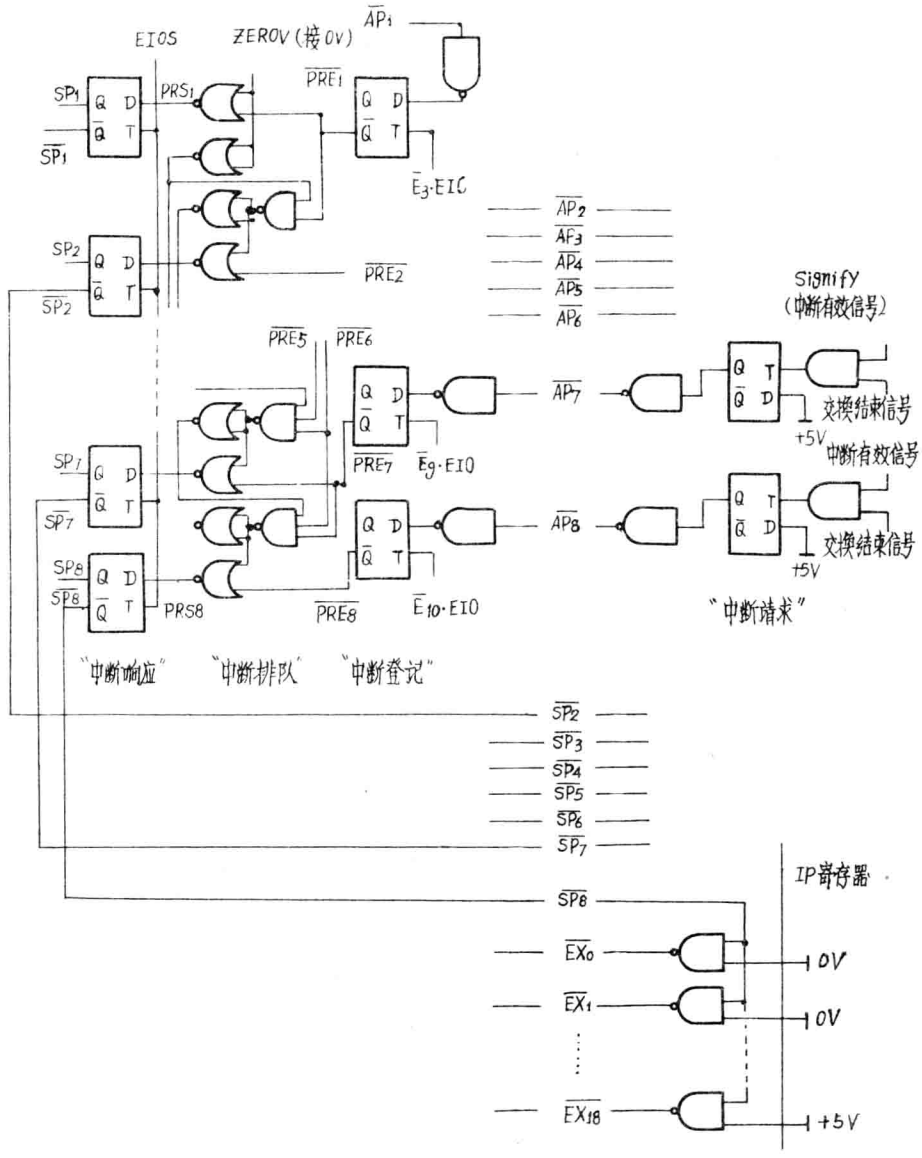


图 1.9