

# HDLC 规约及其硬软件技术

上海交通大学 提光旭

## 一、前言

运动技术的计算机化,是势所必然,运动技术的通信规约,也必将向计算机靠拢,计算机数据通信规律,世界上已有许多种,其中用的较多的是美国IBM公司的BSC规约。而这种规约有如下缺点:

- (1) 对每一信息块必须一一响应,效率低。
- (2) 为了建立同步要用去传输信息中的部分码元,二进制信息传输困难。
- (3) 传送的电文不能进行出错控制,可靠性差。

基于上述原因,在高速信息传送的计算机时代,它就不能完全适应这一要求,因此各公司都相继开发了种种高效的规约,其中被定为国际标准之一的就是HDLL规约,它是High Level Data Link Control Procedure(高级数据链控制顺序)的缩写。它具有以下几方面的优点。

- (1) 它以帧为基础,建立同步是由叫做符号顺序(Hag Segurce)来完成的,可传送任何二进制的信息,而不必做任何特别说明,即透明性较好。
- (2) 可以连续发送7或127个信息块之后,要求响应,导线使用效率高。
- (3) 能对全部电文进行差错控制,可靠性高。
- (4) 采用CRC检错,检错能力好。
- (5) 不同机种之间联接较容易,兼容性好。

所以HDLC规约,目前是一种有生命力的规约,有许多国家定于选用。下面就其具体规约进行说明,之后就目前的硬软件在实现HDLC上的情况,进行分析,最后谈一谈用SIO的软件。

## 二、HDLC规约

### 1. 帧的结构与传送

在HDLC中,信息全部以帧为单位传送的,帧的构成如下:

图1 帧结构

F	A	C	I	FCS	F
8bit	8bit	8bit	任意bit	6bit	8bit

F: 帧标志 (F Lag Sequence)

A: 地址区 (Address field)

I: 信息区 (Information field)

FCS: 帧检查顺序标志 (frame Check sequence)

每个帧由F开始, A、C、I、Fcs按顺序连接, 以F结束。

一帧的开始和结束, 都有F来加以识别, F共八位, 即“01111110”组成, 以此来建立收发双方的同步, 同时也表示帧的开始, 根据需要帧的最后F也可以省略。

地址域是八位, 在1:N方式下, 决定子站地址的, 在主站→子站传送信息时, 以指明子站的地址, 当子站→主站传送信息时, 通知主站信息的来源。当地址为全0时为非站地址, 子站不作任何响应, 当为全“1”时, 称全地址, 各子站均接收。因此这种情况下, 可有254个地址, 在不够时, 也可扩为16位。可做广扩令之用。

控制区有八位, (在需要时也可以扩展) 它主要表示命令或响应的, 其内容如表1。有三种形式: 信息传送, 监视和非编码。

表 1 控制区的内容

控制部内容	帧名	各位内容							
		b <sub>1</sub>	b <sub>2</sub>	b <sub>3</sub>	b <sub>4</sub>	b <sub>5</sub>	b <sub>6</sub>	b <sub>7</sub>	b <sub>8</sub>
信息(I)	信息(I)帧	0	N(s)			P/F	N(R)		
监视(S)	监视(S)帧	1	0	S		P/F	N(R)		
非编码(U)	非编码(U)帧	1	1	M		P/F	M		

表中N(S): 发信侧所发信息帧的编号。

N(R): 收信侧接收信号帧的编号。

P/F: 是控制权标志

M: 修饰功能码

信息帧是由b<sub>1</sub> = “0”来定义的, 其他二种帧是由b<sub>2</sub>来区别的。发信侧在b<sub>2</sub>~b<sub>4</sub>位中的N(s)中, 标有数据帧的编号, 以防止在信息传输中信息块的丢失或重叠, 它可记录8种。

当主站向子站发布指令时, P/F = “1”, 当子站接收到命令而响应时, P/F = “1”以表示通信良好。在发送几幅帧时, 是在最后帧的位置上出现。如图2所示。

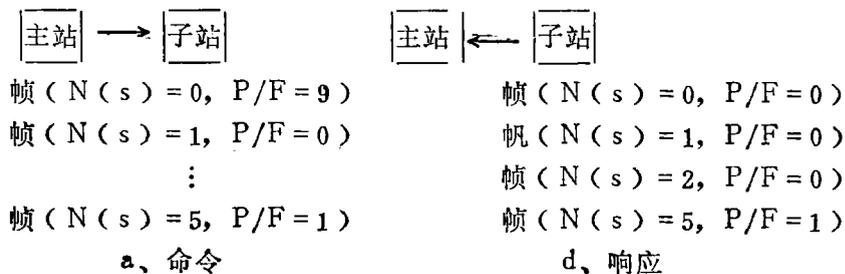
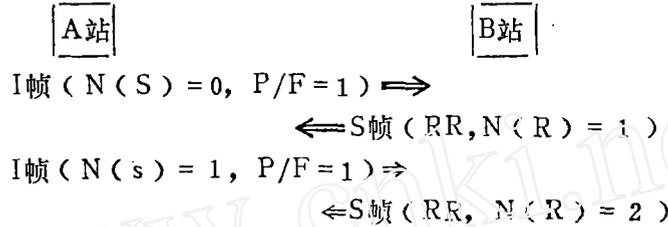


图 2 P/F位的作用

当 $b_1b_2 = 10$ 时, 代表监视帧, 它的用途之一是对信息帧的应答。例如图 3 表示 $b_3b_4 = 00$ 时, 代表信息帧接收准备好的响应。



RR是S帧的4种当中之一, 即图 3  $b_3b_4 = 00$ , 表示 $N(s) = i$ 帧信息在B站已正确接收, 而且也为接收 $N(s) = i + 1$ 帧信息, 做好了准备之意。所以叫做RR: Receive Ready。在接收端没有接收到时,  $N(R)$  仍为前一个数字, 使对方再发。如图4所示。

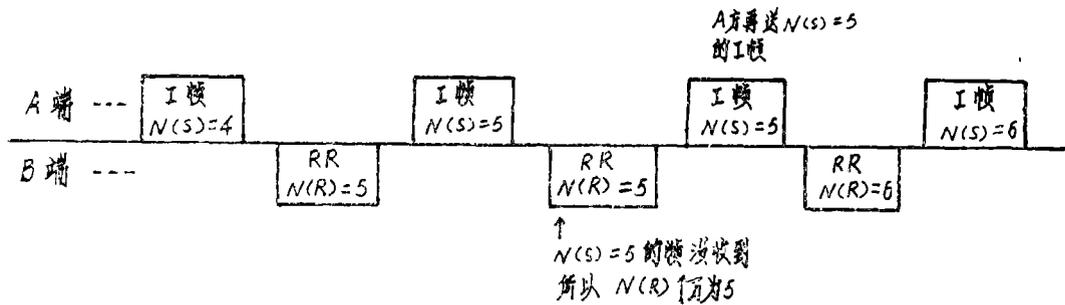


图4 按RR要求再送示意图

$b_3b_4 = 10$ 时, 称作接收没准备好RNR (Receive not ready) 其动作过程如图 5 所示。

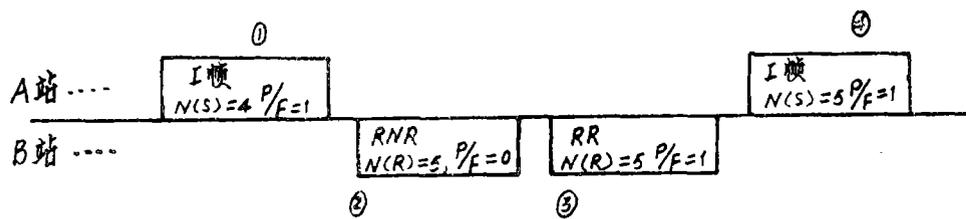


图5 RNR动作示意图

在①处, 因为 $P/F = 1$ , 表示下一帧由B站发出, 而B站由于某些原因没准备好, 因而使得 $P/F = 0$ , 并作成RNR, 第五个要自己发出, 这是②状态。③时B站发出 $N(R) = 5$ , RR, 而且又是 $P/F = 1$ , 表示下次请A站发, ④是A站发出信息。显而易见RNR 是表示对方没有作好准备, 而发出的响应信号。

$b_3b_4 = 01$ 时, 称作请求从指定编号开始再送功能。REJ (Reject) 当处于REJ 的 $N(R) = i$ 时, 表示对方 $(i - 1)$ 之前的帧已全部接收, 而 $i$ 号帧以后没有收到, 所以

要求从第i号帧开始，再发送一次，这当然可以用RR命令，但REJ命令更为明确。

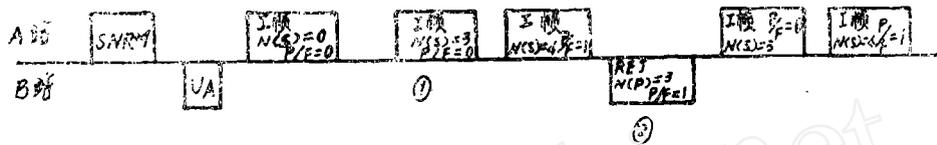


图 6

$b_3b_4 = 11$ 时，称作SREJ (Selective Reject)，它同REJ大体相同，区别是SREJ只要求重新传送指定的帧号，此帧以后的不用传送。

利用上述命令和响应，可以进行对话式的信息传送。其动作过程可参看图7。

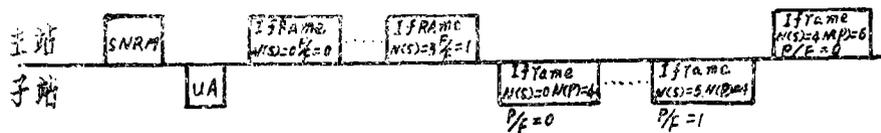


图 7

其他命令和响应情况，请参看表2

## 2、关于HDLC的透明性

所谓数据的透明性，是指信息经过传输以后，仍能保持原样的特性。这里讲的不是对信道干扰的抗拒性，而是指数据信息有能力抗拒伪监控码的干扰。伪监控码是指某些监控码（如同步码，STX、ETX等控制码）图形相同的数据码。在IBM的BSC中是采用信息段内填字码的形式加以区别，在HDLC中是采用自动插入和删除“0”的办法。叫作填位法 (bit-stuffing)

因为在HDLC中的帧标法F中，其码为  $(7E)_{16}$ ，它有6个连续的1。为了不使信息部分同F混杂，因此规定帧的其他部分（非F部分），若有连续5个“1”时，后面插入一个“0”在接收端遇到5个1时，就将后面的0去掉，以恢复原来的信息码。例如，发送的数据为011110111111011110，发送时和接收时情况如下图8。



图 8 插入0与除0的传输情况

这里顺便指出HDLC判断帧出错的方法。由帧结构看出，在F之后是8位地址，再后8位控制码，F之后共8位，从后面的F看F之前是16位检错码，因此二个F之间的最小距离应该为32位二进制码，这时表示无信息传送，如果此时收到的码小于32倍，那么肯

定这帧是有错。

帧检错部分是CRC-CcITT16位的循环码即见余多项式为  $G(x) = X^{16} + X^{12} + X^5 + 1$ ，它有很强的检错和纠错能力，对于信息传输中的码元，1、2或3位非连续出错或出错码为奇数，或突发性错码长度 $\leq 16$ 比特，其错码可检率为100%，突发错长度=17比特的检错率为99.9995%，对于大于突发错的长度 $\geq 18$ 比特的检错率为99.99847%。

### 3、HDLC的模式

HDLC通信中能完成二种形式的通信方式①主站同254个子站之间的通信，即1：N形式，另一种是二个都成为主站，或称为复合站之间的通信，这种形式下只能是1：1的形式。

按HDLC规约进行数据通信的系统，要设定通信模式，HDLC共有六种响应模式，它们是：

- 1、正规响应模式NRM (Normal Response Mode)
- 2、异步响应模式ARM (Asynchronous Response Mode)
- 3、异步平衡响应模式ABM (Asynchronous Balanced Mode)
- 4、扩张式正规响应模式NRME (Normal Response Mode Extended)
- 5、扩张异步响应模式ABME (Asynchronous Balanced Mode Extended)
- 6、扩张异步平衡响应模式ABME(Asynchronous Balanced)Mode Extended

其中扩张的三种方式。可以在未取的对方响应的情况下，连续发送127个信息帧“因为较复杂而使用率低，故此处省略可参阅有关文献。一般情况都使用前三种模式。

所谓正规响应模式是指主站召唤子站，允许子站响应时，子站才可以发出响应的模式。因此即使终端很多的情况下，由于主站可以控制响应的发送，是唯一的1：N方式。一条线上连接多个子站，大大提高了导线利用率。这种模式下，正如前面所说的，线路控制权是由 $b_4$ 的P/F置1来完成的。

ARM模式是子站未经主站同意可以发送响应，但不能发送命令的模式。由于响应无设置模式的功能，所以只能在主站的控制下来改变。关于P/F位，由于子站发送响应与主站无关系，因而用不着像正规响应方式那样，在响应发送的最后一帧中，使 $P/F = 1$ ，而是在发送第一帧的P/F中置1就可以了，其作用是一样的。主站也不用等待相应命令的响应，就可继续通信。

异步平衡模式是主、子站都有发送命令和响应的复合模式。只适合于复合通信方式下，P/F位也同异步模式一样，这种模式是最复杂的一种。

HDLC模式的设定是由主站发出模式设定命令，子站送回UA响应，以示完成，之后就开始了数据通信。

关于信息流的控制。

在通信中传送装置的处理速度并不一定完全相同，有的快些，有的慢。一般情况下，慢装置发信，快装置接收，通信是没什么问题的，但反过来那就比较困难了，而HDLC规约也能使这种装置的通信顺利进行。IBM公司的BSC中。是根据ACK/WACK码来

完成这种控制的,在HDLC中,正如前面所说,当接不到信息时,就发RNR帧,取得后就再发RR帧,使其继续通信,接到RNR的站必须停止信息帧的发送,如图5所示。

关于模式的解除。传送的信息完成时,或者是因什么理由不让其继续进行传输时,就要解除模式,这可以在主站发布DISC命令,或根据子站发出的RD响应信息,促使主站发出DISC命令。在接到DISC命令时,发信UA响应,移向切断模式。但这种切断只是从逻辑上,而非电气上,如果需要,还须从电气上加以措施。

HDLS规约如上所述,由于它的特点而被定为国际标准规约之一。另一方面,像IBM公司也推出新的规约叫SDLC基本同HDLC规约,只是帧的结构中,最后并不是F,而是GA(Go ahead) polling码:“01111111”,七个连续的1,8个1以上也作为GA,用它可以在更高的速度下完成应答通信。同时也可以完成环路通讯。

### 三、HDLC用的LSI

正如上面所述,在HDLC中,在发信侧要发送F等标志顺序,插入0码,产生CRC保护等,在接收侧要检出F等帧顺序,去掉多余的0码,检出CRC码等,如果用一般电路来完成是较困难的。目前各公司已相继产生出适用于HDLC或SDLC的LSI电路。从传送信息的速度上从数十K比特,到数M比特,在功能上,有进行地址比较功能,有SDLC的GAPolling功能以及和其他规约相兼容的功能等。下面就其代表的加以简述。

#### 1. Motorola公司的MC6854ADLC。

它的传送速度只能达到660kbit/秒,而它的MC68B54可以达到1.5Mbit/秒,但无地址比较功能。

#### 2. Zilog公司的Z80SIO。

这个SIO也和UART和USRT兼容的,什么也可以做的万能规约器件。它有二个通道,而且均可用于全双工工作,但在DMA情况下,只能作为半双工工作。它使用Z80的中断模式2。

该片电路有三种SIO,如图9,由于它只有40只脚,因此少了一只脚,A通道满了,而B通道就少一只。当然仔细考虑一下,总会有些不用的脚,所以经过考虑还是可以弥补的。它的传送速度,对2.5MHz的Z80,500kbit/秒,对于4MHz的Z80,可达800kbit/秒,对于6MHz的Z80可达1.2Mbit/秒。在子站有地址比较的功能,但无环路功能。而且它很难同其他cpu芯片相连接。

#### 3. Intel公司的8273。

8273的传输速度较低,只能64kbit/秒,(8273—4是56K,8273—8是48kbit/秒)但在功能上具有地址比较也有环路的功能。它的最大特点是具有数字锁相电路PLL(Phase Lock Loop)这对保证传送的同步具有极大的好处,但它需要32倍的传输比特位的时钟,因此传输速度不可能很高。

Intel的8274(MPSC)和Z80SIO一样,可以说它是Z80SIO的翻版,它可以和8085,

8088、8086芯片兼容、它可以达到880kbit/秒的传输速度。

此外日本NEC日本电气的uPD7201基本同Intel的8274。Intel公司的8251是不具备HDLC功能的。

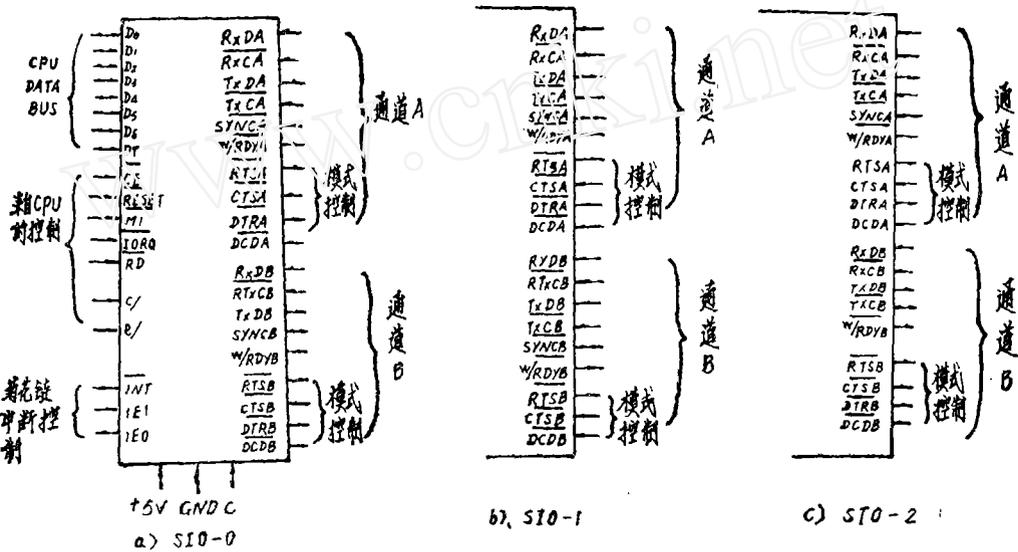


图9、Z80 SIO的三种排列

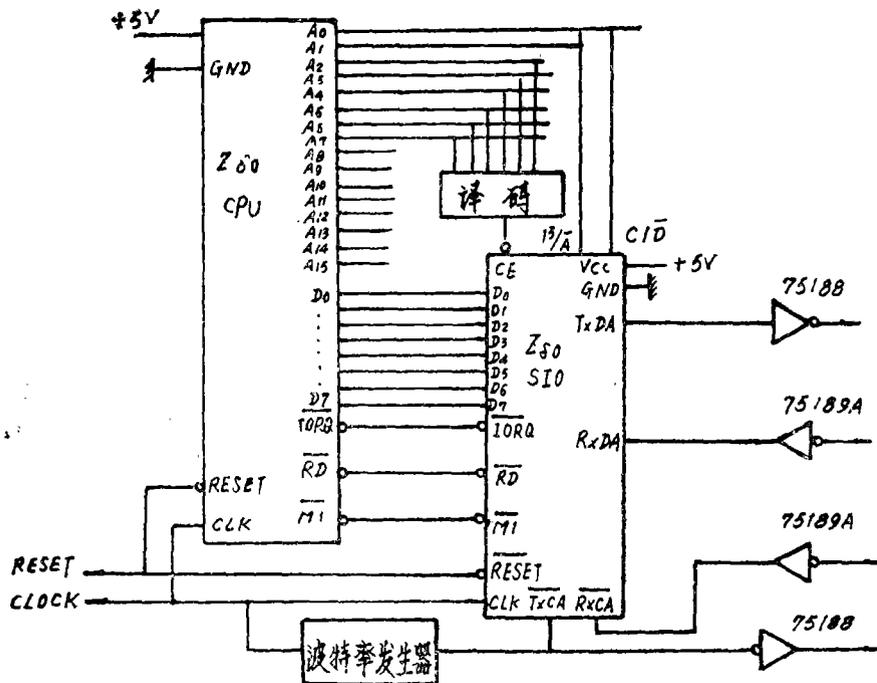


图10 Z80SIO的连接

#### 四、HDLC的软件举例:

HDLC的规约已如上述,能完成它的片子也有许多,用的较多的还是Z80的SIO,因此以下的程序就以Z80 SIO来加以说明。由于篇幅的限制,有关Z80 SIO的详细使用说明,此处予以省略,请读者另找参考书参考。本文认为读者已对Z80 SIO的使用比较了解的。

使用Z80SIO完成HDLC时,其基本的思路如下。

发信侧要自动地发送出特征帧F,数据中需要插入零的地方要自动插入零,产生CRC,并在全部信息帧完成后,发出CRC和F帧。帧发送完后,不发送字,这样有意发送空,用这种方法来使CPU通知Z80SIO,这种方法比较简单,也是SIO的一个特点。

在接收侧,自动检出标志符帧F,作为主站时,要接收全部帧。作为子站时,只把自己的地址,即WR<sub>0</sub>中的字,和地址帧相比较,来判断接收否,是就将接收的信息自动地除去0,做CRC检查。否则不接收。

HDLL时的汇编程序如下,现在简单地加以说明:

行19~22,是初始化程序,使Z80SIO初始化,行24~52是对它的各个寄存器进行写入,这时SIO不能通信。在HDLC中主站和子站不同,本文列出的为子站的情况,也就是在接收初始时,将地址、查找模式(写寄存器WR3中)设为“1”,而主站时,由于要接收全部帧,所以要它设为“0”,而不要检查地址。

RSTRT是接收开始子程序。RECEIV是接收子程序,用它来等待接收字符,收到了就返回。异步和正规同步的情况下,接收完了的字符用软件来判断,利用收信子程序,来判断是否应接收下一字符,是的话再调用接收子程序。对HDLC而言,结束是由SIO来告知,在该子程序中,了解SIO是否发出结束标志,它是73~85行。因此返回时,就将结果通知主程序。

A寄存器=0:表示接收了信息(信息在C寄存器)

A寄存器=FF:表示接收完结(C寄存器存有结束条件)

C寄存器=0:表示正常结束

C寄存器=FE:表示是由失效(abort)而结束

C寄存器=FF:表示通过CRC结束。

所谓失效“abort”的概念,是说当信息传送中中途因某些原因而切断了,发信侧不发标志符。而是送出连续的“1”。接收侧接到连续8个“1”以上时,就作为退化或失效(abort)而宣布作为该帧信息不正确。

REND是接收结束子程序,TSTRT是发信开始初始化程序,TRANSMT是发信程序,发送字送到SIO,等到发送缓冲器空时,进行发送。

TEND为发信结束子程序。对于Z80SIO来说,发信缓冲器空时,自动地判断帧的结束,就发送CRC和特征帧F,结束帧的发送。

136~141行是检出信息中有插入字符时(同步时插入同步码)正确地在发信二个字符之间为取得正确的时序而编的程序。因为HDLC要做插入零的动作,所以要根

表2 HDLC规约的命令与子站响应表

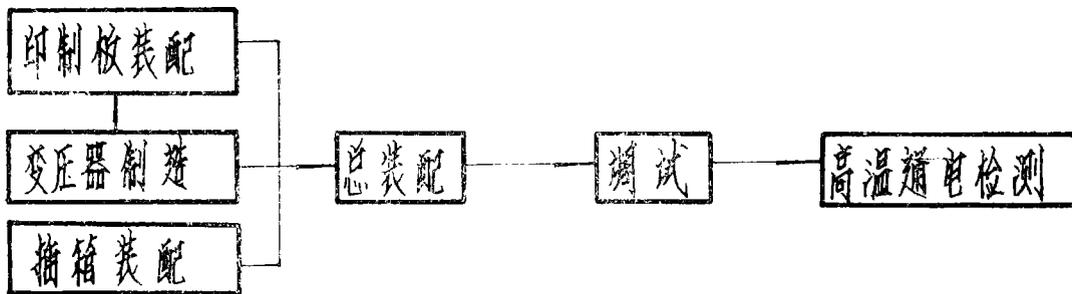
命令	响应	b <sub>1</sub>	b <sub>2</sub>	b <sub>3</sub>	b <sub>4</sub>	b <sub>5</sub>	b <sub>6</sub>	b <sub>7</sub>	b <sub>8</sub>	命令及响应的功能说明
I	I	0	N(S)			P/F	N(R)			I 信息
RR	RR	1	0	0	0	P/F	N(R)			RR 接收帧准备好
RNR	RNR	1	0	1	0	P/F	N(R)			RNR 接收未准备好
REJ	REJ	1	0	0	1	P/F	N(R)			REJ 要求以指定的帧号开始再送
SREJ	SREJ	1	0	1	1	P/F	N(R)			SREJ 要求发送指定编号的帧
UI	UI	1	1	0	0	P/F	0	0	0	UI 无编号信息
SIM	SIM	1	1	1	0	P/F	0	0	0	SIM 置子站于初始状态
NRO		1	1	0	1	P/F	0	0	0	RIM 子站请求SIM
SARM	DM	1	1	1	1	P/F	0	0	0	SARM 将子站置于ARM形式
UP		1	1	0	0	P/F	1	0	0	PM 子站通知主站为非连接模式
SABM		1	1	1	1	P/F	1	0	0	UP 向子站询问响应
OISC	RD	1	1	0	0	P/F	0	1	0	SABM 置成异步平衡模式(复合式)
NRI		1	1	0	1	P/F	0	1	0	DISC 将对方是为切断连接方式
SARME		1	1	1	1	P/F	0	1	0	RD 请子站发送DISC
	OA	1	1	0	0	P/F	1	1	0	SARME 置子站设为扩张ARM
SABME		1	1	1	1	P/F	1	1	0	UA 对U帧给予发定响应
SNRM		1	1	0	0	P/F	0	0	1	SABME 置复合式两站设为扩张ABM
FRMR	FRMR	1	1	1	0	P/F	0	0	1	SNRM 置子站设为NRM正规模式
NR2		1	1	0	1	P/F	0	0	1	FRMR 告知接收到不正确的帧
RSET		1	1	1	1	P/F	0	0	1	RSET 置接收编号从头开始
XID	XID	1	1	1	1	P/F	1	0	1	XID 请求发信识别码
NR3		1	1	0	1	P/F	0	1	1	SNRME 将子站设为扩张正规模式
SNRME		1	1	1	1	P/F	0	1	1	

# 西门子工厂装配工艺概况

许昌继电器厂 周 涛

最近我们赴联邦德国西门子公司进行了近两个月的装配工艺的技术培训，总的印象是西门子各工厂设备先进、自动化程度高，工艺先进、具有严格的质量控制系统，管理先进、普遍采用计算机进行生产管理和技术管理。

这些工厂的产品装配主要由印制板装配、变压器制造、插箱装配和总装配组成，基本工艺流程如图一所示。



图一

将几个方面的装配工艺见闻的几点意见叙述如下。

## 一、印制板装配

这些工厂的印制板装配工艺的基本流程如图二所示。

据它来定时间。该程序是针对CPU为4MHz时9600bit/秒的程序。

### 参考文献

- (1) 宫崎诚二著 微机数据传送的基础与实际
- (2) Ziog MiCRoco Mpu TER COMPONENT DATA Book 1981.
- (3) 接口 1984.11.第11号 P.268~280
- (4) Z80汇编程序设计手册。
- (5) Intel 微型计算机器件手册。上海交通大学1983.12.