

计算机通信接口研究

于云程 苏建志

提 要 设计信息系统时必须解决远程终端与主机之间的通信问题, 本文具体讨论以 Z80 微型机系统为终端与主机 FACOM—230/38 之间的通信接口设计问题, 采用的通信规程是富士通公司的 BSC 规程。文中主要讨论了接口设计应考虑的问题, 规程功能分析, 论证电文块长度的选择, 最后分析 BSC 规程特点, 并以软件实现了该规程, 给出部分通信程序流程图。

一、概 述

本文要解决的问题是把两种不同类型的计算机, 即 Cromemco 公司的微型机同日本富士通公司的 FACOM—230/38 机, 按照富士通提供的 BSC 通信规程文本, 用专用电话线路把它们连接起来, 系统的配置如图 1。

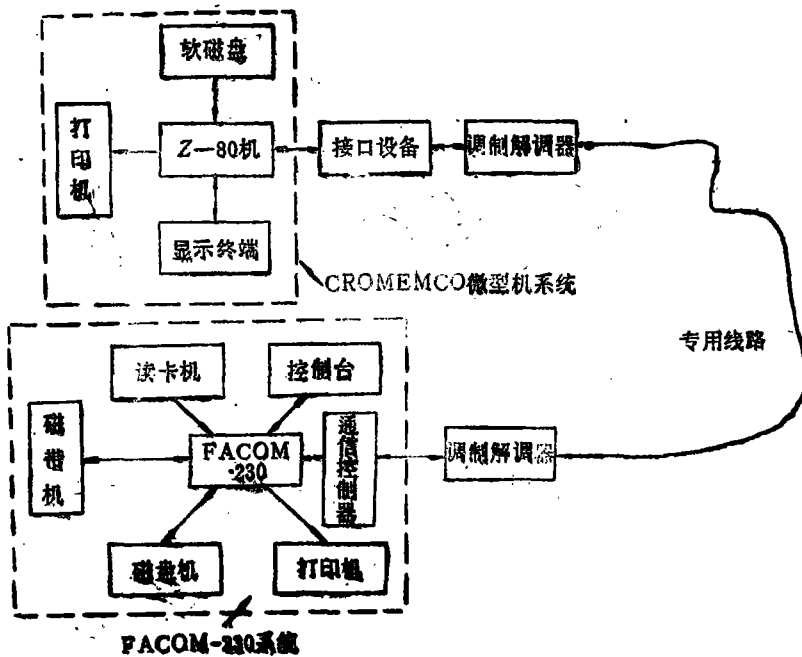


图 1 系统配置图

图中两虚框内分别表示 Cromemco 微计算机系统和 FACOM—230/38 机系统。所采用的调制解调器为日本的 MD—1230, 速率 1200bps, 同步型的, 按标准接口同数据设备联接。我们的任务就是在上述的约束条件下, 给微型机设计一接口, 使整个系统协调工作。接口包括硬、软件两部分。

二、设计中应弄清的几个问题

接口设计有两种: 一种是在系统设计时, 接口设计是总的系统任务之一, 统一解决; 另一种是给已定型的系统新加接口, 前者简单, 后者比较复杂, 本文主要讨论后者。为此必须搞清下面问题:

1. 要配接口计算机是单用户还是多用户

对于单用户没有资源的分配问题, 而对于多用户资源一般分为用户独占部分, 如内存和公用部分, 如系统程序、外设等。设计的通信资源——即硬件和软件部分如何为各用户服务呢? 从经济和灵活来看最好为各用户公用, 大家分时享用。

2. 如何解决通信软件和系统软件间的关系

为了使新增加的通信程序置于系统软件控制之下, 有两种解决办法: 解剖操作系统, 找出相应的入口, 这种方法很有效, 但难度大; 对于目前市场上出售的微型机系统, 多数给用户的功能扩展留有入口, 如 Cromemco 系统的各种“系统调用”, 利用它们把通信程序的运转纳入操作系统控制之下, 这种方法简单易行, 我们采用的就是这种方法。

3. 原系统的硬软件在多大程度上为新增加的通信功能所用

此问题和头一个问题有关。对于单用户所有的硬软件资源均可用, 如果不考虑通信和其它作业的并行运行, 只要附加一简单的标准接口硬件设备。软件由原 CPU 执行就足够了。对于多用户的系统, 为了满足通信和其它作业并行运行, 而尽可能少占主 CPU 的工作时间, 采用功能强的硬件设备比较适宜。正是因为这样的原因采用 Cromemco 公司的“IOP”和“QUADART”板组成专门的通信设备, 进一步提高原系统的分布处理能力。

IOP 主要由 Z—80、16K 内存和 2K 的监控程序组成, 由它接收来自主 CPU 的命令、程序和数据, 另外由它控制 QUADART 板的工作, 把远程来的数据送给 CPU。

“QUADART”完成收发、CRC 校验和定时、提供标准的 RS—232C 接口。

增加 IOP 和 QUADART 板后, Z—80 微型机框图如图 2。

4. CPU 同 IOP 间的数据交换方式

有两种方式可选择: 查询和中断两种方式。查询方式的优点是灵活、便于调试、但占用 CPU 工作时间长, 影响多用户工作能力。中断方式和查询比较, 主要优点是节约运行时间, 因此采用中断方式。Z—80 有三种中断方式, IM2 方式中断最灵活, 功能也最强, 因此采用此种方式。

交换数据块大小由许多因素决定, 下面将给出较详细的分析。

5. 通信软件的组成及要求

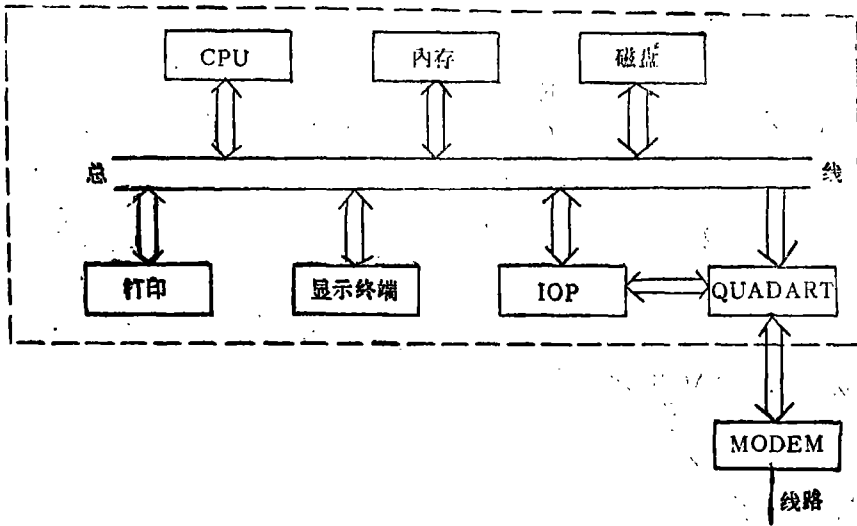


图 2 微型机系统框图

除了在 IOP 中运行的通信程序外，通信软件还必须包括在 CPU 内运行的通信管理程序，其功能是：完成同 IOP 间的数据发和收，向 IOP 传送程序和命令，接收 IOP 的服务请求，对外设的读和写以及必要时的代码转换功能。

通信程序可用 ROM 的方式常住在 IOP 或由磁盘调入，除了“QUADART”硬件完成字符的分解组装，CRC 校验，同步字符和空闲字符的插入以及群同步的检测等功能外，通信规程的功能主要由 IOP 内的程序完成。对通信程序的基本要求之一是时间性：在一个字符的收发时间内要完成规程规定的动作。

设线路的（即系统要求的）数据率为 R (bps—比特/秒)，IOP 的执行时钟频率为 F_2 ，则 IOP 完成规程规定动作所需的节拍数 N 应满足：

$$N \leq \frac{F_2}{R}$$

取 $F_2 = 4\text{MHz}$ ， $R = 1200\text{bps}$ ，则

$$N \leq 3.3 \times 10^3 \text{ (节拍数)}$$

若不满足上述关系，将使系统的效率下降或者完全破坏系统的工作。

三、规程功能的选择

一种规程文本往往能在各种场合下运用，但具体的运用场合又往往比较简单，通常制定规程文本时就考虑到这种情况，除了基本的功能外还提供一些可选功能。基本功能包括链路的建立和拆除，数据正确无误的传输以及在任何条件下不能发生无限循环和死锁现象。至于文本中的其它功能要么是提高系统的效率，要么可能根本是不必要的，因此这些功能属于可选的。下面对文本中可选性进行具体讨论。

1. 信息序列

文本中列出了下面五种格式：

- (1) SOH (标题信息) ETB CRC
- (2) STX (电文信息) ETB/ETX CRC
- (3) SOH (标题信息) STX (电文信息) ETB/ETX CRC
- (4) DLESTX (电文信息) DLEETB/DLEETX CRC
- (5) SOH (标题信息) DLESTX (电文信息) DLEETB/DLEETX CRC

考虑到我们要完成的任务是点对点智能终端的接口设计，数据格式不限，即要求具有透明性传输的能力，因此选用(1),(4)两种形式或它们的组合形式(5)就足够了。

2. WACK 和 TTD 序列

延迟应答序列 WACK 和延迟发送序列 TTD 分别使从站和主站有使用低速 I/O 设备的能力。如果从站的输出设备能够很快地完成数据的转移，因而不会影响下面立即来的数据接收，这种延迟应答的能力就不需要采用。同样采用了相对来说速率比较高的输入设备或者采取其它一些措施，不会出现数据输入低于发送速率的问题，采用 TTD 也就成为多余。

对于我们的系统来说，由于是多用户，采用 WACK 和 TTD 序列也就十分必要。

3. 反向传输请求功能 RVI

能够加速从站变为主站，可根据具体要求取舍。

4. 否定应答 NAK

从规程完整性来说没有 NAK 而代之以不响应，不会受影响，但有了它会提高系统的工作效率。

5. 计时和计数值的选择

计时和计数本身是不可少的，但其值根据系统的部件和配置可作适当的变动，不仅不会影响系统的工作，相反还会提高系统的性能。

以上仅对主要可选性进行了讨论。

四、发送电文块长度的讨论

电文块的长度是重要的系统参数之一，它直接影响到系统的通信能力和软硬件的复杂程度，因此从系统的观点看，电文块的长度既不能过长也不能过短，应该按照系统的本身的特性选择，我们用电文字符数 m 来表示电文的块长。

1. 从系统的有效吞吐量选择最佳电文块长 m_{op}

若链路规程为 BSC，按半双工工作，纠错用等待应答，出错重传，干扰为随机型的热噪声，根据资料[1]的结果，加以修改得：

$$m_{op} = \frac{-(\Delta TR + K_2 C)}{2K_2} + \sqrt{\left(\frac{K_2 C - \Delta TR}{2K_2}\right)^2 + \frac{\Delta TR C - (\Delta TR + K_2 C)/L_n(1-P_w)}{K_2}} \quad (1)$$

其中 R 为系统的数据率，以每秒的比特数表示； K_2 为每个字符的传送比特数； C 为电文传输时增加的字符开销数，一般为(7~8)个； P_w 为传输过程的出错概率，若系

统的误码率为 P_K , 则

$$P_W = 1 - (1 - P_K)^{K_2} \approx 8P_K$$

ΔT 为相邻两个电文块间的时间, 其大小与线路的延迟, 收发调制解调器的同步建立时间, 收发站对输入的响应时间和反向应答传输所需的时间有关。若系统限于国内的线传输, 信号的延迟可以不计。

准确计算收发站的响应时间比较困难, 因为它与计算机本身的性能, 作业的种类和数量等许多因素有关, 而且从性质上说它是一随机量。对于本文所讨论的计算机系统近似认为数据的存取只是对磁盘系统, 且响应时间主要由磁盘的访问时间决定, 另外在一次电文块的传输中只计主站的响应时间。Z-80 系统软盘存取时间可视为它的延迟时间 (LATENCY), 其值约为 70ms (毫秒), FACOM 用的是硬盘, 访问时间近似为 40ms, 因两机工作于主站的机会相等, 故计算机的响应时间取为 55ms。

MD-1230 型调制解调器的同步建立时间为 20~70ms, 取其中间值 45ms。

反向应答序列格式为:

PAD SYN SYN ACK 0/1 PAD.

略去 PAD 字符, 字符按 8 位传输, $R=1200\text{bps}$, 不准计算出反向传输时间。考虑以上因素, ΔT 近似为:

$$\begin{aligned} \Delta T &= \frac{5 \times 8}{1200} \times 2 + 55 \\ &= 178.3\text{ms} \end{aligned}$$

上式第二项中的因子是考虑到传送电文一次同步两次。

根据资料 [2] 提供的线路实验数据:

$$P_K = 10^{-3} \sim 10^{-5}$$

取 10^{-4} 作为计算值。

把上述各值代入式 (1) 计算得

$$m_{op} = 189 \text{ (字符数)}$$

2. 电文长度的最佳分段

一般电文长度是随机的, 对于不同的系统相应的随机性也不完全相同。具体特性同用户的种类和性质等许多因素有关, 但是电文长度的分布规律相似, 资料 [3] 给出的分布如图 3 所示, $W(M)$ 表示电文长度的概率密度函数。

另外资料 [1] 给出系统的吞吐量 Q 成正比, Q 和电文块长 m 的关系为

$$Q = \frac{K_1 m (1 - 8P_K)^{m+c}}{K_2 (m+C) + \Delta TR} = Q(m) \quad (2)$$

其中 K_1 为字符的信息比特数, ASC11 码其值为 7。

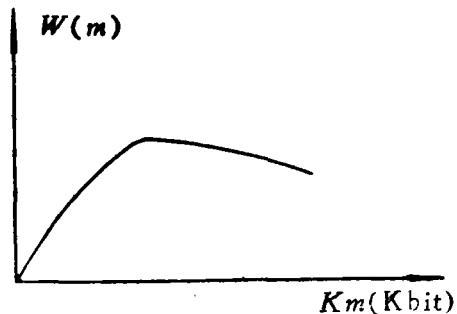


图 3 电文长度的分布

对于给定的 P_K 、 ΔT 、 R 和 C ， Q 和 m 的关系如图 4 所示，其特点是在 $m=m_{op}$ 时， Q 取最大值，随着 m 偏离 m_{op} ， $Q(m)$ 逐渐下降。

现在讨论对于给定的电文长度 M ，怎样分块传输最有利。

假定把 M 分为段长分别为 m_1 ， $m_2 \cdots m_N$ ，则相应系统的有效吞吐量正比于 Q_M ：

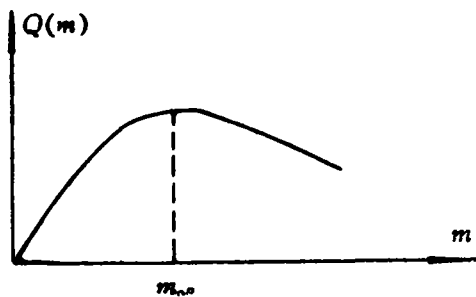


图 4 $Q(m) \sim m$ 曲线

$$Q_M = \frac{\sum_{i=1}^N [Q(m_i) \cdot m_i]}{\sum_{i=1}^N m_i} \quad (3)$$

$$\sum_{i=1}^N m_i = M \quad (4)$$

从数学上讲，对电文长度 M 的最佳分块，就是使式 (3) 在约束条件 (4) 下出现最大值。由于 Q 是 m 的非线性函数，只能用递推方法计算，因此计算起来很繁琐，而且这种严格的计算方法对我们的问题未必必要。故促使我们寻求另外的解法：从 $Q(m)$ 的几何图形出发，辅助以计算，具体作法如下：

A. 对于任何电文长度 M ，存在一种分法使式 (3) 出现极大值。直观地看电文块分的过长或过短均使 $Q(m)$ 下降，相反使各 m_i 的值尽可能地接近 m_{op} ， Q_M 就增加，因此 Q_M 对分法存在极大值。

B. 对于给的 M ，最佳的分法应满足：

$$m_{op} - m_i \geq 0 \quad i = 1, 2 \cdots N$$

$$\text{或} \quad m_{op} - m_i \leq 0 \quad i = 1, 2 \cdots N$$

即最佳分法，使分得的各段长 m_i 要么大于或等于 m_{op} ，要么均小于或等于 m_{op} 。证明如下：

假定有一分法，使 Q_M 最大，且存在一对 m_j 和 m_K 满足：

$$m_{op} - m_j > 0$$

$$m_{op} - m_K < 0$$

那么我们肯定作到：使 m_j 增加， m_K 减小，使两段长向 m_{op} 接近，而同时又保持二者的和不变。由于 $|m_{op} - m_i|$ 越小，对应的 $Q(m_i)$ 越大，从而说明原来的分法不是最好，因而最好的分法必须满足上面的条件。

C. 最佳分法一定满足：

$$|m_{op} - m_i| < m_{op} \quad i = 1, 2, \cdots N.$$

即

$$0 < m_i < 2m_{op}.$$

证明方法同前面相似，此结果说明最佳分法可以使 m_i 大于或小于 m_{op} ，但距 m_{op} 的距离不能超过 m_{op} 。

D. 最佳分法是等分 M 的方法, 即

$$m_i = \frac{M}{N}$$

假定存在一种最佳分法, 使

$$m_j < m_K < m_{op} \quad (\text{也可设 } m_{op} < m_i < m_j)$$

那么可以再分 M , 使 m_j 增加, m_K 减小, 且保持 $m_j + m_K$ 不变, 这时 Q_M 的增加量 ΔQ 为:

$$\begin{aligned} \Delta Q &= Q(m_j + \Delta m) + Q(m_K - \Delta m) - Q(m_j) - Q(m_K) \\ &\approx [Q'(m_j) - Q'(m_K)] \Delta m. \end{aligned}$$

$$\because \Delta m > 0 \quad Q'(m_j), Q'(m_K) > 0$$

且 $Q'(m_j) > Q'(m_K)$

故 $\Delta Q > 0$

所以不等分 M 不是最好的分法。

由上面分析, 最佳分法使下式成立:

$$\begin{aligned} Q_M &= \frac{\sum_{i=1}^N Q(m_i) \cdot m_i}{M} = \frac{\sum_{i=1}^N Q(m) \cdot m}{N \cdot m} \\ &= Q\left(\frac{M}{N}\right) \end{aligned} \quad (5)$$

其中 N 为正整数, 为了求出 N 的最佳值, 我们采用资料[4]中所介绍的线性规划方法之一, 先用 n 代替 N 作连续变化, 求出 n_{op} , 之后把 n_{op} 整数化得出 N 。

由式(5)看出使 Q_M 最大的条件显然是:

$$\left(m_{op} - \frac{M}{n}\right)^2 \text{ 最小}$$

$$\text{故 } \frac{d}{dn} \left(m_{op} - \frac{M}{n}\right)^2 = 2 \left(m_{op} - \frac{M}{n}\right) \frac{-M}{n^2} = 0$$

$$\therefore n_{op} = \frac{M}{m_{op}} \quad (6)$$

n_{op} 整数化就是取最接近于 n_{op} 的正整。故

$$N = \begin{cases} [M/m_{op}] - 1, & \text{当 } [M/m_{op}] - M/m_{op} \geq \frac{1}{2}, M/m_{op} > 1 \\ [M/m_{op}], & [M/m_{op}] - M/m_{op} \leq \frac{1}{2}, M/m_{op} > 1 \\ 1, & M/m_{op} < 1. \end{cases} \quad (7)$$

其中 $[M/m_{op}]$ 表示大于或等于 M/m_{op} 的最小正整数。

若在传输数据之前, 先对线路的误码率 P_K 进行测试, 算出 m_{op} , 就能使系统对环境

自动适应,为此先产生测试码送入信道,并把信道联接成测试所需的形式,测出 P_K ,自动标出 N ,使系统对消息长度和环境处于最佳运行。

3. 从缓存的经济性来考虑电文块长的选取问题

A. 收发站的缓存量至少等于电文块的最大长度,因此最大块长 m_{\max} 越大,硬件费用越高。

B. 电文块长对缓存利用率 η 的影响

由于电文长度是随机的,分出的电文块长并不总是等于缓存的容量。

设缓存的容量为 m_R ,电文长度 M 的概率分布为 $W(M)$,分块原则为:

$$M = nm_R + m$$

$$\text{则近似有: } \eta = \sum_{n=1}^{\infty} \int_0^{m_R} \frac{nm_R + m}{(n+1)m_R} W(nm_R + m) dm$$

不难看出 m_R 越小,缓存利用的越好。

4. 电文块长度同软件的复杂性间的关系

按照规程文本规定,电文块的传输时间超过一秒要插入同步序列,对于传输速率为1200bps,字符的位长为8比特的系统,不需加同步序列插入的电文块为150个字符。

另外对于Cromemco微型机的软盘系统,读写以记录为单位,每个记录是128个字符。若不采用以该记录的整倍数为单位进行传输,同样会加大软件工作量。

综合前面的讨论,对于我们所讨论的具体系统和我国目前有线信道的具体情况,采用128个字符为电文块长度较为合理。

五、通信程序设计

我们以软件实现通信规程,程序设计中应考虑的问题讨论如下。

1. 主站与从站

本规程为争夺式,即终端可以作主站也可以作从站。所谓主站就是取得消息发送权的站。主站与从站的处理要求不相同,因此必须分别设计主站与从站的通信处理程序,然后将两个程序合并在一起,使主、从站在一定条件下可以转换,这样就构成了整个通信程序。具体的作法是设一个“发送键”,只要按过“发送键”终端便进入主站,并使内存中主/从站标识单元 M/S 置1。

2. 控制字符

本规程规定了14个控制字符(详见附表2),除同步字符SYN外,一般来说其中ENQ、SOH、STX、ETB、ETX、TTD和EOT为主站发送控制字符,也就是从站必要判断并作出相应处理的字符;ACK0、ACK1、WACK、RVI、NAK、EOT为从站对主站的应答或响应,也就是主站必需判断并作出相应处理的字符。此外还有DLE,它是透明电文标识符,主站从站都应判断,以便按透明电文格式处理。

另外,ACK0、ACK1、WACK、RVI、TTD都是双字节字符,必需连续接收两个字符才能判明一个控制字符。

3. 传输控制矩阵

终端侧的传输控制矩阵(详见附表3)概括了整个规程的传输控制过程,有两点特别值得注意:

(1) 传输控制矩阵的行代表终端在通信过程中所处的状态,共有八个状态。其中状态②、③、④、⑤属于从站,而⑥、⑦、⑧属于主站。状态①为中间状态,若此时终端处于从站,则该状态为选择序列接收状态;若此时要求发送电文,则该状态为选择序列发送状态,然后转为主站状态⑥。

(2) 矩阵的列代表接收到的控制字符。行列的交点给出了相应的处理,同时也给出了状态转换的条件。

综合以上两点,通信程序的数据结构是二维数组,按行存贮形成八个转移表,与之对应设计八个状态处理程序。算法是以控制字符为键字检索转移表。

4. 子程序、指示器

字符的发送与接收操作对于各种状态都是共同的,所以可设计为两个子程序。

- 接收子程序负责接收一个字符,当接收以中断方式工作时这相当于接收字符中断(RCA)服务子程序。

- 发送子程序负责初始化接口板,编排电文格式,发送字符。

- 代码转换也是应考虑的特殊问题,当异种机相联时尤其如此。本课题终端Z-80使用ASCII代码,而主机FACOM使用EBCDIC码,所以必需进行代转换。这个任务由代码转换子程序完成。

- 指示器,为便于发送与状态转移处理,我们设置一个指示器。指示器就是在Z-80内存用户区指定5个单元(见图5),它们的用途是:

- FADD存放要发送电文的起始地址,两个字节。
- DCNT存放发送的字节数,一个字节。
- MSJP存放状态转移参数,即转移表首地址,两个字节。

5. 计时器、计数器

规程规定了在各种情况下电文等待的计时值:2秒、3秒、25秒。种类较多并且计时的时间长,如果用软件实现,则终端微型机势必长时间空转运行,不但不能发挥机器应有的效能,甚至使处理工作无法进行,故我们采用硬件计数器(CTC)配合以软件的方法解决。

规程还对某些应答的重复次数作了限制,因此在处理中必须对这些应答的次数计数。计数方法由软件实现比较合适,在内存中设置一个计数单元,与程序配合对不同应答公用。

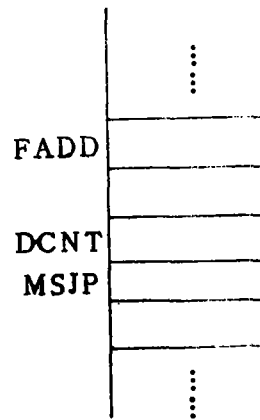
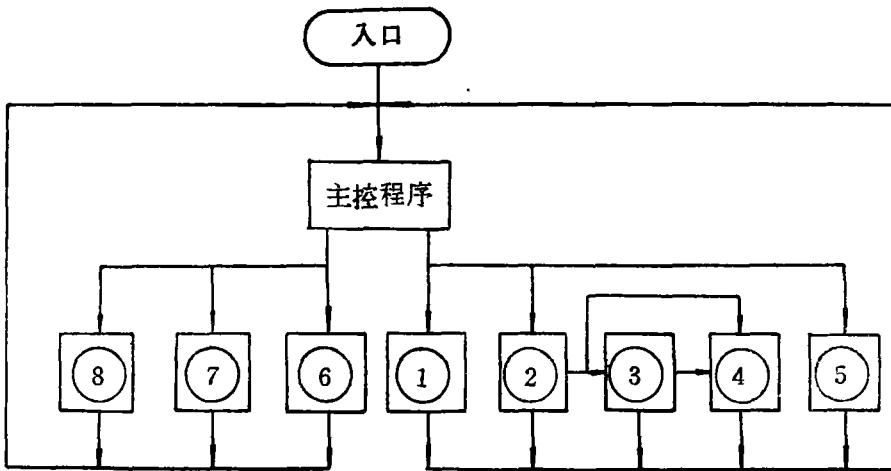


图5 指示器

六、程序块

根据以上各点的考虑,整个通信程序由9个程序块和几个子程序组成:



①~⑧代表状态①~⑧处理程序

图 6 主控程序与状态处理程序的关系

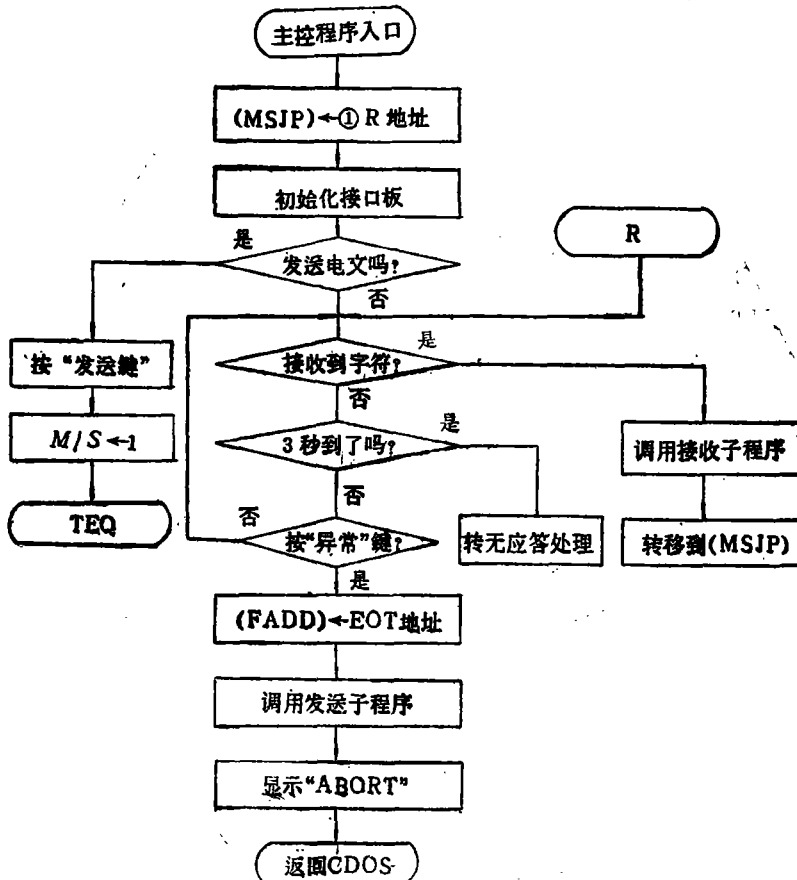


图 7 主控程序简化流程图

1. 主控程序。
2. 状态①——⑧共八个状态处理程序。
3. 发送接收、代码转换等子程序。

主控程序的功能是：

- 初始化接口板；
- 接收一个字符；
- 主/从站转换；
- 状态转移；
- 处理异常结束。

状态处理程序的功能是：

- 对接收到的控制字符作出反应及处理；
- 标明状态转移参数。

主控程序与各状态处理程序之间的关系示于图 6。

我们对每个程序块都进行了设计，这里给出主控程序（图7）、选择序列发送与状态①R处理程序（图8）、发送子程序（图9）的简化流程图，以说明我们的设计。

关于流程图指出以下几点：

- (1)各状态处理完毕均从入口R返回主控程序。
- (2)状态转移由MSJP的内容决定，MSJP的内容是变化的，即每个状态处理程序均添写适当的转移地址。
- (3)前面已经说

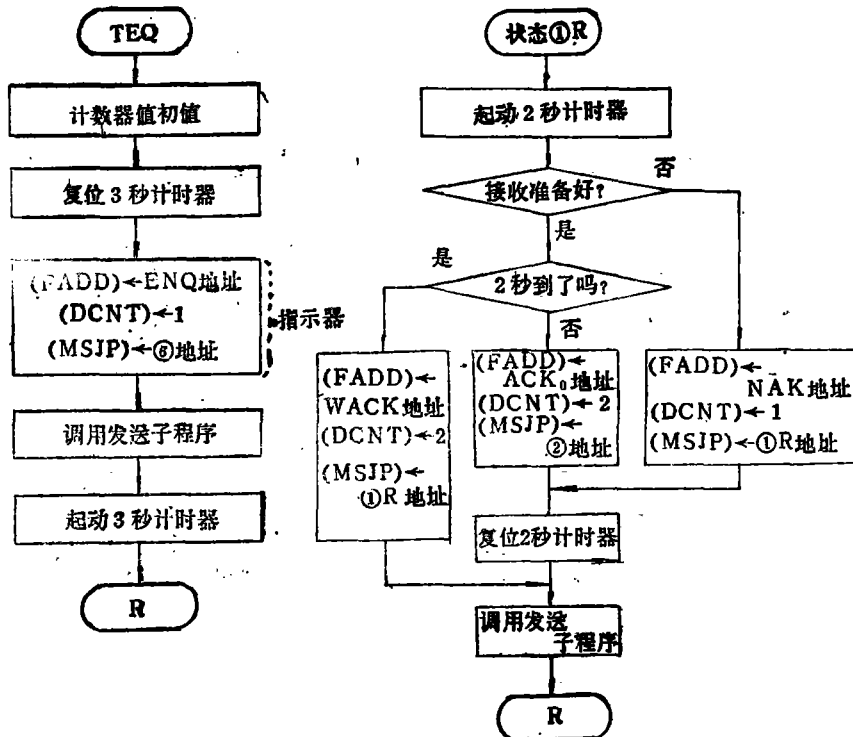


图 8 选择序列发送程序、状态①R处理程序简化流程图

过状态①是中间状态，根据要求终端可为主站或从站，TEQ 和①R 分别为这两种情况的处理程序。

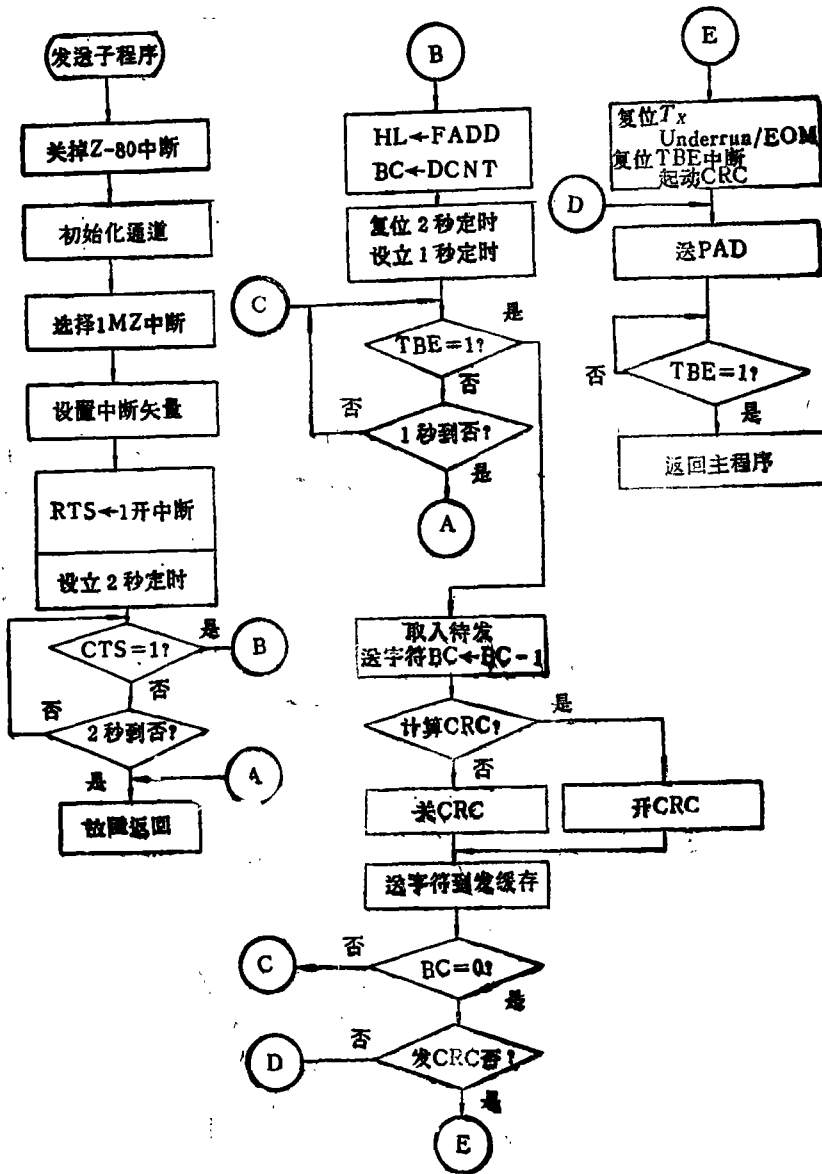


图9 送发子程序简化流程图

七、结 语

以上讨论了计算机通信接口的设计原则和通信程序设计的有关问题。用软件实现了BSC 规程。此外，我们还作了两个实验（文中没有涉及），一是用一台Z-80 微型机系统模拟两机远程对话，距离1.2公里；另一个是两台Z-80 微型机系统相距15公里

作对话实验。实验结果证明所考虑的问题是正确。

我们试图通过具体实例阐明通信接口设计的一般原则。但由于所讨论的实例范围有限，加之我们水平不高，不全面和不妥之处一定不少，请读者批评指正。

富士通BSC规程的基本条件 附表 1

项 目	条 件
线路种类	专用通信线路
传输速率	1200/2400/4800 bps
通信方式	半双工
同步方式	SYN 同步
连接式	争夺(Contention)方式
应答方式	用 ACK0/ACK1 和 NAK 交替应答
通信控制码	EBCDIC 码
通信码系	EBCDIC 或透明码
误差控制	CRC [$G(x) = x^{16} + x^{15} + x^2 + 1$]
信息发送顺序	先发送的位 $b_7b_7b_6b_5b_4b_3b_2b_1b_0$ CRC
适用的调制解调器	同步 MODEM

传输控制字符

附表 2

字 符	16进代码	$b_7b_6b_5b_4b_3b_2b_1b_0$	$b_7b_6b_5b_4b_3b_2b_1b_0$
SYN	32	0 1 0 0 1 1 0 0	
SOH	01	1 0 0 0 0 0 0 0	
STX	02	0 1 0 0 0 0 0 0	
ETB	26	0 1 1 0 0 1 0 0	
ETX	03	1 1 0 0 0 0 0 0	
EOT	37	1 1 1 0 1 1 0 0	
ENQ	2D	1 0 1 1 0 1 0 0	
ACK0	1070	0 0 0 0 1 0 0 0	0 0 0 0 1 1 1 0
ACK1	1061	0 0 0 0 1 0 0 0	1 0 0 0 0 1 1 0
NAK	3D	1 0 1 1 1 1 0 0	
WACK	106B	0 0 0 0 1 0 0 0	1 1 0 1 0 1 1 0
RVI	107C	0 0 0 0 1 0 0 0	0 0 1 1 1 1 1 0
TTD	022D	0 1 0 0 0 0 0 0	1 0 1 1 0 1 0 0
DLE	10	0 0 0 0 1 0 0 0	

注：上述以外的代码，不能作传输控制字符使用。

终端一侧的传输控制矩阵 附表 3

状态	发送指令和内部要求		ENQ	SOH	(DLE)STX	(DLE)ETB/X	TTD	EOT
			A	B	C	D	E	F
中间的	①	a→+RSP② b→NAK① c→WACK①	√	√	√	√	√	√
SOH 等待	②	前面的应答	③	③	④	√	15次 NAK② 16次 EOT①	①
(DLE)STX 等待	③	NAK②	③	③	④	d→+RSP② e→NAK② f→WACK⑤ g→RVI②	③	③
(DLE)ETB/X 等待	④	n→NAK② o→④	④	④	④	同上	NAK②	④
发出WACK后 ENQ 等待	⑤	a→+RSP② f→WACK⑤ g→RVI②	√	√	√	√	√	①
选择序列的应答等待	⑥	h→7次 ENQ⑥ 8次 EOT① i→①—A 相同	√	√	√	√	√	7次 ENQ⑥ 8次 EOT
信息组的应答等待	⑦	√	√	√	√	√	√	①
TTD的应答等待	⑧	√	√	√	√	√	√	①

终端一侧的传输控制矩阵 续附表 3

状态	发送指令和内部要求		+RSP G	NAK H	WACK I	RVI J	越 时 K	内部要求		备 注
								L		
中间的	①	∨	∨	∨	∨	∨	∨	ENQ⑥	∨	∨不予理睬
SOH等待	②	∨	∨	∨	∨	∨	EOT① (25秒)	∨	∨	
(DLE)STX等待	③	③	③	③	③	③	NAK② (3秒)	∨	∨	
(DLE)ETB/X等待	④	④	④	④	④	④	同上	∨	∨	
发出WACK后ENQ等待	⑤	∨	∨	∨	∨	∨	EOT① (25秒)	∨	∨	
选择序列的应答等待	⑥	BLOCK⑦ m→EOT① ENQ⑥	7次ENQ⑥ 8次EOT①	15次ENQ⑥ 16次EOT①	⑥-H 相同	⑥-H 相同	⑥-H 相同 (3秒)	∨	∨	
信息组的应答等待	⑦	j→BLOCK⑦ p→EOT① i→TTD⑧ m→⑦K相同	7次BLOCK⑦ 8次EOT①	15次ENQ⑦ 16次EOT①	j→BLOCK⑦ h→EOT① l→TTD⑧	7次ENQ⑦ 8次EOT① (3秒)	7次ENQ⑦ 8次EOT① (3秒)	∨	∨	
TTD的应答等待	⑧	TTD⑧	⑦~J 相同	TTD⑧	TTD⑧	TTD⑧	7次TTD⑧ 8次EOT① (25秒)	∨	∨	

状态: a. 接收准备状态正常; b. 不可接收; c. 暂不可接收; d. 原信息组已正确接收, 下一信息组接收准备正常; e. 信息组有错; f. 原信息组正确接收, 下一信息组暂不能接收; g. 原信息组正确接收, 而本端现有发送要求; h. 优先端; i. 非优先端; j. 有待发的信息组; k. 无待发的信息组; l. 待发的信息组未准备完毕; m. 交互性混乱; n. 无透明码; o. 有透明码。
标志: BLOCK: 信息组; +RSP: ACK0/ACK1

参 考 文 献

- [1] 苏建志: 通讯规程性能分析, 国防科技大学学报, 1981, No.3.
- [2] 中国科学院, 线路实验报告.
- [3] Davies, D.W., Borber, D.L.A, "Cemmunication Network for Computers", 1973.
- [4] 计算机应用与应用数学, 1976, No.9.
- [5] 富士通公司 BSC 规程文本。

Consideration of Computer Communication Interface

Yu Yun-cheng Su Jian-zhi

Abstract

This report mainly discusses design problems of the interface for computer communication on the basis of the engineering task which needs to link the microcomputer Z80 and FACOM—230/38 remotely. The Fujitsu's BSC protocol is used. In this paper, the aspects in design of communication interface are discussed and some functions of the protocol are discussed, the formulas evaluating optimum length of the message block are deduced from effective line throughput, optimum division of the message block, usage of memory and so on. Finally, characteristics of the BSC protocol is analysed for communieation programming, the protocol is realized by software and some communication program flow charts are presented.