

微机接口技术大作业

一、概述

1. 基本功能

本设计为一个以8086CPU为核心的模型机，利用基本的芯片能够完成一些经过简化的常用功能。CPU能够执行预先设定的程序，管理与其连接的硬件资源，使整个设计的系统有条不紊的运行。利用系统总线将CPU与输入输出接口电路相连，再连接外围设备，使模型机实现数据的输入输出。根据对CPU和外围设备之间的通信要求——以查询方式进行并行输出、以中断方式进行串行输入，设计了合理的外围设备。

并行传输由于多位数据共同传输，可以使传输速度很快，但也因此需要大量的数据线，成本很高。所以其广泛应用于微机系统中，是微机系统中最基本的信息交换方法，其常用于与并行接口打印机、磁盘驱动器进行通信。磁盘驱动器等为简单连续传送，常常利用DMA进行通信，以节省CPU查询等工作以提高CPU利用效率。所以在此选择并行接口打印机作为外设；

串行传输由于使用的数据线少，在远距离通信中可以节约通信成本。而现成的公共电话网是通用的长距离通信介质，它虽然是为传输声音信号设计的，但利用调制解调技术，可使现成的公共电话网系统为串行数据通信提供方便、实用的通信线路。在此将CPU的串行输入接口经过电平转换和调制解调之后连接到公共电话网，以实现远距离的传送。串行接口的调制解调控制电路连接到调制解调器上，其用于传输的接口连接到RS-232-C收发器上，将TTL电平转化为公共电话网传输标准所接受的RS-232-C电平。

2. 组成及各部件基本原理

- 以8086CPU为微处理器，工作在最小模式；
 - CPU内部通过内部总线将算术逻辑部件、累加器和寄存器组及控制器连接起来，使CPU可以进行算术运算逻辑运算、保存较少量的数据、对指令进行译码并致敬规定的动作、能和存储器 and 外设交换数据等功能；
 - 在此模型机中CPU工作在最小工作模式中，系统中只有8086一个微处理器，所有总线信号由8086直接产生，使系统总线控制电路简化到最少。
- 以8251作为串行输入接口，经过调制解调连接公共电话网进行远距离传输；
 - 在本模型机中8251采用异步通信方式，其特点是：首发双方不用统一的时钟进行定时，没有固定的时间关系，所以每个字符前后要用若干为作为分隔符进行识别，即每次需要重复建立同步信号传送一个字符。当在传输率相同时，可见同步方式信息有效率比异步方式高，但同步传输的要求比异步传输高，必须用同一个时钟进行协调，而异步传输的收发方时钟频率比较接近即可；
 - 在此通信方式下，发送端和接受端都需要时钟来决定每一位对应的时间长度，两个时钟频率为为传输率的波特率因子倍，波特率由系统软件设置，时钟脉冲由8254芯片提供；
 - 8251芯片还可以设定字符位数，增加奇偶校验等；

- 作为异步输入接口的工作方式：当其准备接受一个字符时，就检测RxD是否为低电平，将检测到的低电平作为起始位，启动接受控制电路中的内部计数器进行计数，该计数脉冲就是接收器时钟脉冲RxC。当计数器进行到响应于半位的传输时间（在本模型机中设定波特率因子为64，所以在此为计到第32个时钟脉冲），又对RxD进行检测，如果此时仍为低电平，则确认收到一个有效起始位。8251开始进行常规采样和字符装配，即每隔一位的传输时间（32个时钟脉冲）对RxD进行一次采样。数据进入串并转换器中的输入移位寄存器移位并进行奇偶校验和去停止位，变成并行数据，送到接受缓冲器中，同时接受控制电路发出中断申请RxRDY信号送给8259，利用中断CPU通知已收到一个可用数据；
- RS-232-C收发器和调制解调器
 - 调制解调器及公共电话网络的电平要求都是RS-232-C标准，所以在8251和调制解调器之间加上一个电平转换器，将8251输出的TTL电平变换为RS-232-C要求的响应信号；
 - 调制是将各种数字基带信号转换成适于信道传输的数字调制信号(已调信号或频带信号)，解调为在接收端将收到的数字频带信号还原成数字基带信号，再经过调制解调之后，就可以连入公共电话网络进行远距离传输了。
- 以8254作为波特率发生器，为8251的TxC和RxC端口提供时钟频率，作为定时器，以便与CPU对8255并行输出设备进行管理；
 - 在串行传输时，数据是一位一位进行传输，每一位数据都占据一个固定的时间长度，接受端和发送端需要根据时钟来决定何时读取和发送一个信号，所以时钟脉冲对串行传输的准确率和稳定性起非常关键的作用。利用8254将晶振产生的1MHz时钟脉冲进行分频之后连接到8251的TxC和RxC端口作为8251异步传输的时钟频率，采取国际常用波特率1200bps，波特率因子为64，得到收发时钟频率为76.8KHz；
 - 典型的计数器含有4个寄存器，即初值寄存器、计数寄存器、控制寄存器和状态寄存器，CPU可以通过访问这些计数器来设定计数器的工作方式和工作状态。计数器从初值寄存器中获得计数初值，送到计数执行部件，从初值开始减一计数最后达到零。输入时钟信号CLK决定计数速率，门控脉冲GATE对时钟进行控制；
 - 在此模型机中，计数器二工作在模式2——分频器。写入控制字后，以高电平作为初始电平，当写入初值并收到触发后，开始作减一计数，当减到一时，OUT端变为低电平，输出一个负脉冲，脉冲宽度为一个时钟周期。将1MHz的晶振产生信号分频到76.8KHz，初值设为13。
- 以8259作为中断管理，管理输入设备与CPU之间进行的以中断方式的通信；
 - 8259 芯片工作方式：中断请求寄存器IRR接受外部的中断请求，8位IRR对应引脚 $IR_7 \sim IR_0$ ，接到某一引脚的中断申请后，IRR即对该中断申请锁存。之后逻辑电路根据中断屏蔽寄存器IMR决定是否屏蔽该申请。中断优先级判别电路将未受屏蔽的中断申请与当前受理的中断比较。如果该中断申请优先级更高，则使8259 INT端置1，向CPU发送中断请求。当前中断服务寄存器ISR存放当前正在处理的中断请求。
 - 输入设备连接到8259的IR6端，设定该片8256型号为60~67H，则该8259中断号为66H，中断向量为0000:0198H。
- 以8255作为并行输出接口，连接打印机，CPU用查询方式与其进行通信；
 - 8255 有三个数据端口，每个端口能收发八位数据。控制电路能根据CPU对8255写入的控制字进行对三个端口工作方式的控制。工作方式有三种，分别为基本输入输出方式、选通的输入输出方式、双向传输方式。A、B、C高四位、C低四位可以独立工作。
 - 模型机中的8255设置为A端口输出连接打印机可以用打印斤并行通信，C端口高四位输出，其中一位连接到打印机驱动端，C端口高四位输入，可以读取打印机的状态，查询其是否空闲。

- 以74LS138控制各个芯片片选信号。
 - 为各个芯片通过将地址线连接到74LS138上，并将译码后的信号连接到各个芯片的片选端，得出各个芯片内寄存器地址。

3. 协同工作

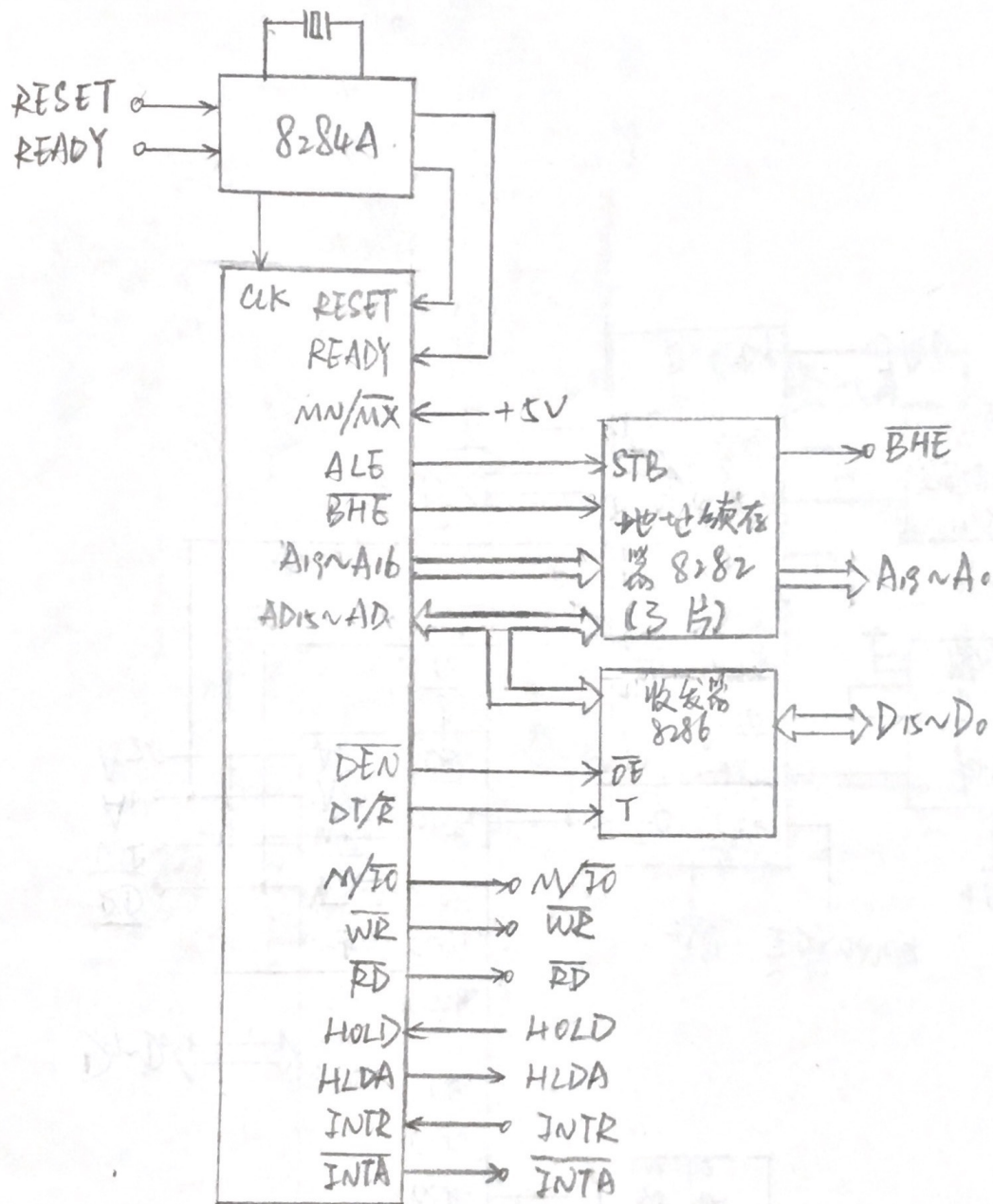
8086 CPU与各个芯片之间的协同工作：

- 8284A芯片作为8086的时钟发生器，有3片8282作为地址锁存器，8286作为总线收发器；
- 8259、8255、8254及8251四个芯片，都与系统数据总线相连，不仅传输了一般的数据，还传输了CPU对芯片的编程命令和芯片送往CPU的状态信息；
- CPU的一部分地址信号经过74LS138译码器译码后得到连接到片选信号，CPU可以通过不同的地址信号选中不同的芯片，对其进行控制。某些低地址位连接到了芯片不同寄存器的选中端口，实现不同的地址对芯片不同端口的访问；
- 各个芯片都与CPU的部分控制总线相连：每个芯片都连接了读写信号，8259还连接了INTA和INTR等中断控制信号；

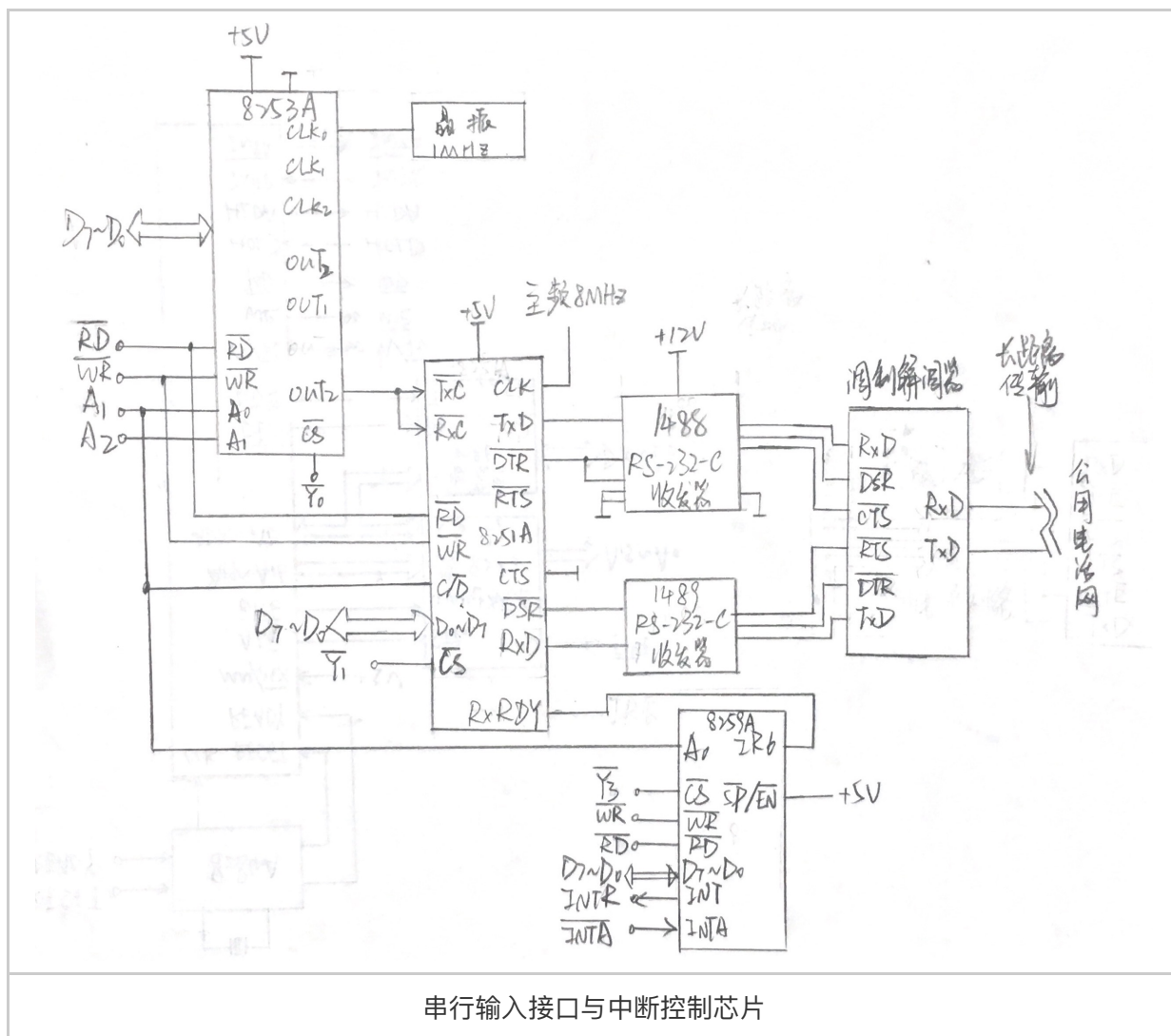
芯片之间的协同工作：

- 8254提供时钟信号，作为8251的接受端时钟频率和发送端时钟频率；
- 8251的接收器准备好信号RxDY，表示当前8251已从调制解调器接收到一个字符，正在等待CPU取走。将作为中断信号，连接到8259其中一个 IR_n 端口；

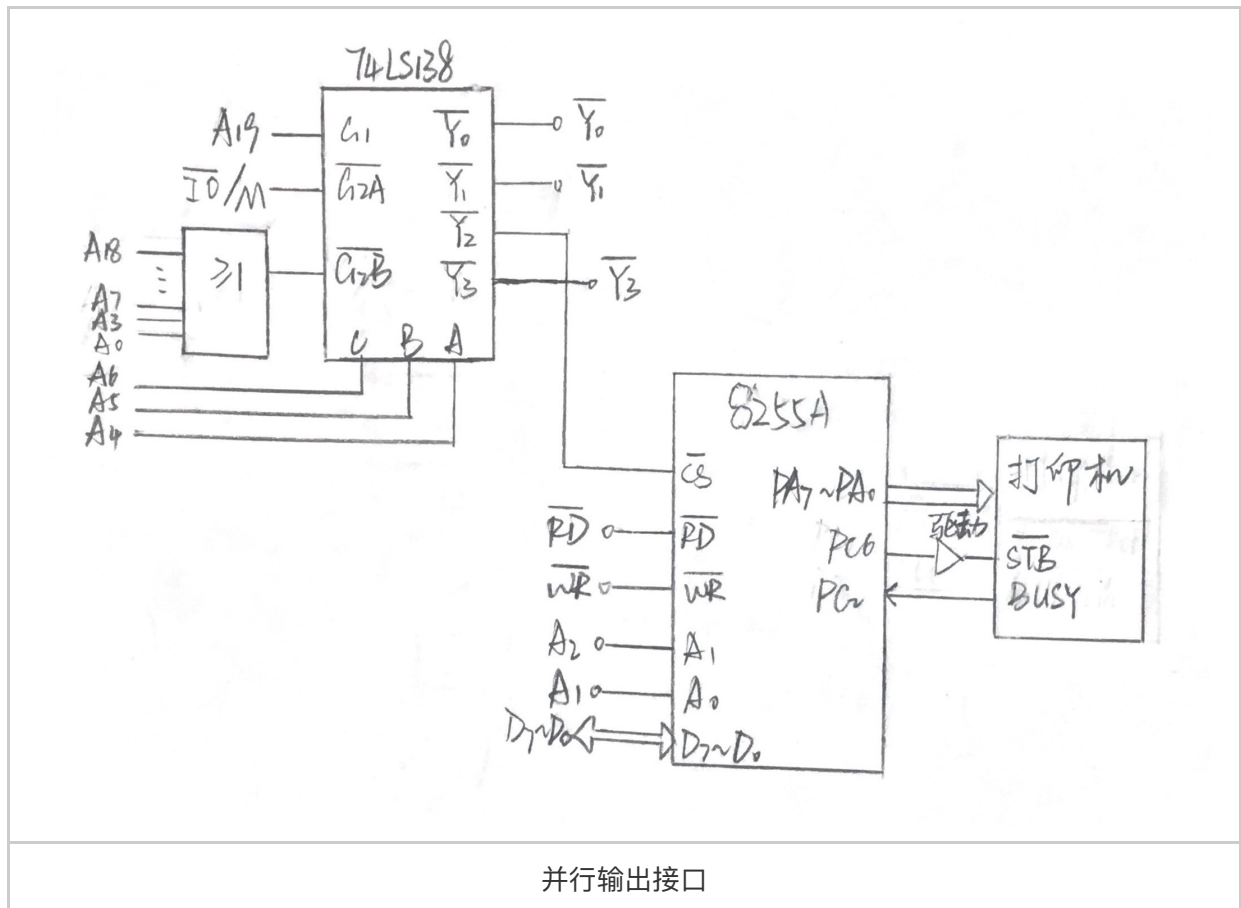
二、芯片连接图



8086处于最小工作模式



串行输入接口与中断控制芯片



三、各芯片工作方式

- 8086 最小模式；
- 8251 工作于异步方式，波特率为1200bps，收发时钟频率为76.8KHz，数据位7位，采用偶校验，两位终止符；
- 8254 计数器2工作在方式2，进行分频，初值为13；
- 8259 为单片使用，上升沿触发，非缓冲连接，非自动中断结束，采用普通嵌套；
- 8255 A端口输出，C高四位输入，低四位输出；

四、输入输出控制程序

```

1  io8253_2 equ 80004h
2  io8253_ctl equ 80006h
3  io8251_d equ 80010h
4  io8251_c equ 80012h
5  io8255_a equ 80020h
6  io8255_c equ 80024h
7  io8255_ctl equ 80026h
8  io8259_icw1 equ 80030h
9  io8259_ocw1 equ 80032h
10 intr_invadd equ 00198h
11
12 data segment
13     buff db 200 dup(0)

```

```

14 data ends
15
16 stack segment
17     db 100 dup(?)
18 stack ends
19
20 code segment
21     assume:cs:code,ds:data,ss:stack
22 start:
23     mov ax,data
24     mov ds,ax
25     ;define 8251
26     mov dx,io8251_c
27     mov al,0fbh      ;异步, 波特率64, 7个数据位, 偶校验, 2个停止位
28     out dx,al
29     mov al,34h      ;设置控制字, 接收启动, 并设置有关信息
30     out dx,al
31     ;define 8255
32     mov dx,io8255_ctl
33     mov al,81h      ;ABC端口均工作与方式0, A为输出, C高四位输出, 低四位输入
34     out dx,al
35     mov al,0Dh      ;置PC6为1, 即STB高电平
36     out dx,al
37     ;define 8259
38     mov dx,io8259_icw1
39     mov al,13h
40     inc dx
41     mov al,60h
42     out dx,al
43     mov al,01h
44     out dx,al
45     mov al,0bfh
46
47     push ds
48     xor ax,ax
49     mov dx,ax
50     mov ax,offset irq6
51     mov si,intr_invadd
52     mov [si],ax
53     mov ax,seg irq6
54     mov [si]2,ax
55     pop ds
56
57     mov si,offset buff ;si是取出数的下一位
58     mov di,offset buff ;di是存放数的下一位
59 lt2:cmp di,si          ;当di大于si时输出
60     jle lt2
61
62     mov dx,io8255_c

```

```
63  lt1:in al,dx
64      test al,04h
65      jnz lt1
66      lodsb
67      mov dx,io8255_a
68      out dx,al
69      mov al,0ch      ;使STB置0
70      mov dx,io8255_c
71      out dx,al
72      inc al
73      out dx,al      ;使STB置1
74
75      mov ah,4ch
76      int 21h
77 code ends
78      end start
79
80 irq6 proc near
81      push dx
82      push al
83      mov dx,io8251_d
84      in al,dx
85      stosb
86      pop al
87      pop dx
88      iret
```