微型计算机原理和接口

下　册

何维中　张治中　编

北京工业学院九系

1983，4

内　容　简　介

本书是为计算机专业《微型计算机原理和接口》课所编教材的下册。上册的内容包括：微处理器和微计算机的概论及Z80微处理器的结构（第一章）；Z80微处理器的指令系统（第二章）；微计算机的汇编语言（第三章）；Z80汇编语言程序设计（第四章）；微型计算机的存贮器（第五章）；输入输出和外围芯片（第六章）。

下册的内容包括：接口技术（第七章）；微型计算机系统（第八章）；16位和32位微处理器（第九章）；微型计算机系统开发（第十章）和微计算机的应用（第十一章）。

参加本书编写的还有李书涛同志（第十章）和蔡建新同志（第十一章）

目　录

第七章　微型计算机的接口技术

§ 7.1 概述……7.1

§ 7.2 与开关的接口……7.2

一、按钮开关……7.3

二、拨钮开关……7.8

三、多位开关……7.12

§ 7.3 与发光二极管显示器的接口……7.17

一、与单个发光二极管显示器的接口……7.17

二、与七段发光二极管显示器的接口……7.20

§ 7.4 与键盘的接口……7.24

一、概述……7.24

二、与非编码键盘的接口……7.25

三、与编码键盘的接口……7.36

§ 7.5 与CRT接口……7.45

一、电视机的典型结构和微处理机的接入方法……7.45

二、电视机的工作原理和字符显示方法……7.46

三、微型计算机与CRT接口……7.50

四、Intel 8275 CRTC ……7.69

§ 7.6 与数/模（D/A）和模/数（A/D）转换器的接口……7.99

一、D/A转换原理及实际芯片……7.99

二、与D/A转换器的接口……7.112

三、A/D转换原理及实际芯片……7.127

四、与A/D转换器的接口……7.145

§7.7 与针式点阵打印机的接口……7.150
一、概述……7.150
二、M—5微型机与TEXAS 810 针式点阵
打印机的接口……7.154
第八章 微型计算机系统
§8.1 概述……8.1
一、微型机系统的支持电路……8.1
二、微型机系统的规整连接……8.2
三、微型机系统连接的电气性能要求……8.3
§8.2 微型机系统的辅助电路……8.9
一、直流稳压电源……8.9
二、时钟发生器……8.10
三、译码电路……8.11
四、复位再启动电路……8.14
五、指令单条执行电路……8.15
§8.3 μC最小系统……8.15
§8.4 单板μC系统实例……8.19
一、实例单板机的组成……8.19
二、实例单板机的存贮器……8.22
三、实例单板机的EPROM 写入……8.25
四、实例单板机的I/O接口……8.30
§8.5 实例单板机的外部设备和驱动程序……8.35
一、LED显示器和系统的连接……8.35
二、十六进制键盘和系统的连接……8.42
三、键盘驱动程序……8.48

§ 8.6 标准总线……8.72
一、概述……8.72
二、S-100 总线系统……8.73
三、IEEE—488总线系统……8.94
四、RS-232-C 总线系统……8.112
§ 8.7 普通录音机作为外存贮器……8.119
一、录音机磁带中数字信息的记录格式……8.119
二、录音机的读写……8.123
三、普通录音机作外存贮器时的接口电路……8.125
§ 8.8 监控程序……8.130
一、监控程序的功能……8.130
二、ZBUG 监控程序概貌……8.131
三、ZBUG 监控程序细则……8.138
四、ZBUG 监控程序的主控程序……8.143
五、ZBUG 监控程序的命令执行程序—ZBUG
ZBUG 的分支程序……8.151
1。“返回监控命令”执行程序—MON键程序……8.151
2。“主寄存器显示和更改命令”执行程序—
EXAM REG 键程序……8.157
3。“单步命令”执行程序—SINGL STEP
键程序……8.173
4。“断点命令”执行程序—BREAK POINT
键程序……8.179
5。“启动命令”执行程序—EXEC 键程序……8.186

此为试读，需要完整PDF请访问：www.ertongbook.com

第九章　16位和32位微处理机
§9.1　16位微处理机概况……9.1
一、16位机的发展……9.1
二、8086微处理机概要……9.2
三、M68000微处理机概要……9.3
§9.2　Z8000 微处理机……9.4
一、概要……9.4
二、Z8000 的结构……9.7
三、存贮器管理……9.23
四、中断和陷井……9.31
§9.3　32 机简介……9.40
一、iAPX 432 ……9.42
二、发展趋势……9.46
第十章　微型计算机系统开发
§10.1微型计算机系统开发的主要步骤……10.1
§10.2微型计算机系统开发的辅助工具……10.11
§10.3微型计算机开发系统（MDS）……10.12
§10.4微型计算机系统开发举例……10.20
第十一章　微型计算机的应用
§11.1国内外微处理器与微计算机应用的概况……11.1
一、国外微处理器和微计算机应用的现状与特点……11.1
二、国内微处理器和微计算机的应用概况……11.5
§11.2微计算机的优点和应用的基本设计思想……11.9
一、微计算机的优点……11.9

二、微计算机应用的基本设计思想…… 11.10
§ 11.3微计算机应用的实例…… 11.14
一、Z80单板机在线切割机床上的应用…… 11.14
二、SDK-85单板机用于测量温度和定温控制…… 11.21
§ 11.4微计算机应用的发展动向…… 11.25

第七章　微型计算机的接口技术

§7.1　引言

上一章已经讨论了微型计算机的输入输出技术，也就是微型计算机与外界交换信息采用何种数据形式，在时间上如何配合，采用何种方式来管理交换过程以及微型计算机的I／O指令方式等。这一章要讨论的微型计算机的接口技术，就是微处理机和各种外围设备之间的具体联接，其中既包括硬件问题，也包括软件问题。关于接口技术的讨论需先说明如下几点。

第一点是，由于讨论接口技术是具体的，所以首先涉及的是微处理器CPU的类型和外围设备的特性。目前，“标准的微处理器”是8位微处理器，例如：Intel 8080，8085，Zilog Z80，Motorola 6800，Signetics 2650等等。而其中Z80的功能是最强的。当前已经开始大量推出的16位微处理器和数量尚少的32位微处理器，由于相应的软件和接口芯片的跟不上，尚不能作为“标准的微处理器”。当然，可以预期，若干年后（如二、三年后），16位微处理器将成为下一个“标准的微处理器”。但是，即使到那时，8位微处理器还会有它自己的用途。所以，下面讨论中，都是8位微处理器的接口技术，而且以Z80-CPU为主。至于外围设备，则是以微型计算机系统中常用的为主（如开关，发光二极管显示器，键盘，行式打印机，阴极射线管显示器，模数和数模转换器，磁盘等）。

第二点是，硬件和软件折衷的问题。下面要讨论的接口问题，通常，其多数是既可用软件方法（程序），也可用硬件来实现，或者是由软硬件的结合来实现。一般来讲，用软件方法可节省硬设备，但降低CPU的效率。用硬件方法可以提高效能，但要增加硬设备。系统设计者的责任就是在两者之间找到一个合理的折衷方案，使硬件和软件之间达到某种平衡。即硬

件和软件的开销最小。当然，这种平衡将随着新的元器件的引入而发生变化。

第三点是，微型计算机的接口技术，一开始是按照习惯的做法，设计一些由SSI，MSI片组成的复杂的逻辑电路板，以管理（即控制）微处理机和外部设备之间的数据传送和发送必须的同步信号。但是由于大规模集成电路（LSI）工艺的极为快速的发展，这种设计已经显得过时了。现在，不仅整个CPU集成在一个芯片上，而且，各个微处理器生产公司还同时配套生产了大量的I/O接口芯片（LSI芯片），正如上一章中介绍的，由于它们是可编程序的，因而是通用的，通过编程，可以加接于不同的外围设备。因此，下面讨论中，多数是采用LSI接口芯片来实现的。目前，LSI接口芯片还没有达到成熟，主要表现在两个方面，一方面是没有标准化，即各个不同的微处理器生产公司均为自己的微处理器配套了大量的LSI接口芯片，但这些芯片与别的公司的微处理器一般是不兼容的，这给使用上造成限制和困难。另一方面，这些LSI接口芯片的多数仍然是"没有头脑的"芯片，即使如上章介绍的DMA芯片，也只能执行有限指令功能。它们必须在CPU控制下工作。但是，由于LSI工艺的飞速发展，在LSI接口芯片中已经开始配置微处理器，成为所谓"智能"接口。虽然现在的LSI接口芯片还未成熟，但是下面要讨论的接口技术对将来的智能接口都同样是行之有效的。

§7.2 与开关的接口

这里说的开关，是指的机械开关，如按钮开关，搬钮开关和多位开关（旋转开关、选择开关或拨轮开关）等。这是最简单的输入外围设备之一。它们与微处理器之间的接口，是最简单的也是最基本的接口。根据计算机的标准，这类机械开关的位置变化是很慢的，所以不需要锁存器。一般只

需要一个可寻址的三态缓冲器即可。但为了使讨论更具有通用性，下面讨论中均采用可编程序的并行接口PIO来实现。使用机械开关的唯一问题是它闭合（或断开）得不干脆。也就是说，这种开关会颤动，即在建立良好的接触之前有一个不规则的开闭过程，最后才达到稳定。这个过程叫颤动时间，颤动时间的长短取决于具体的开关，一般为几个毫秒（如2ms）。微处理器必须在开关达到稳定后才能读取数据，否则会读入错误的数据。颤动可用两种方法来处理，以确保读入正确的数据。一种是用软件方法，使CPU在发现一次闭合后直到颤动停止为止有一段延迟，这种延迟可用一个延迟子程序来实现。在此时间内CPU处于等待状态，也可以去完成其他任务。另一种是用硬件方法。即采用硬件电路来实现。如一个单稳电路或交叉耦合的与非门等。下面将讨论各种机械开关接口的具体实现（与Z80-CPU）。

一、按钮开关

下面讨论一个按钮开关（或一个单刀单掷开关）与Z80-CPU的接口，电路如图7－1所示。

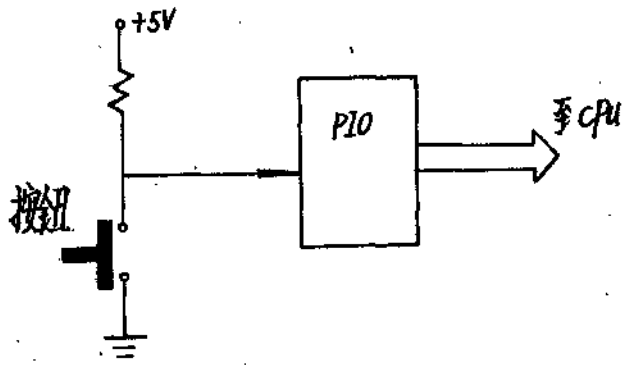

图7－1　按钮开关的接口电路

图7－1中，按钮开关用Z80PIO的一位作缓冲器接至Z80CPU。

由于按钮开关在许多个CPU时钟周期内保持闭合，故不需锁存。按钮开关被按下时，PIO的输入位便接地。上拉电阻保证按钮开关未被按下时输入位置1。这里作为接口的Z_{80}PIO是作为输入口用的，具体的与Z_{80}-CPU及按钮开关的连线如前面Z_{80}-PIO一节所介绍的，此处从略。下面利用上述接口电路完成两个功能，并编出相应的程序。这两个功能是：①根据按钮的状态设置一个存贮单元的内容。②计数按钮按下的次数。

1、确定开关闭合

按钮开关未按下时使存贮单元0040置“1”；

按钮开关已按下时使存贮单元0040置“0”。

实现上述功能的程序流程如图7－2所示。

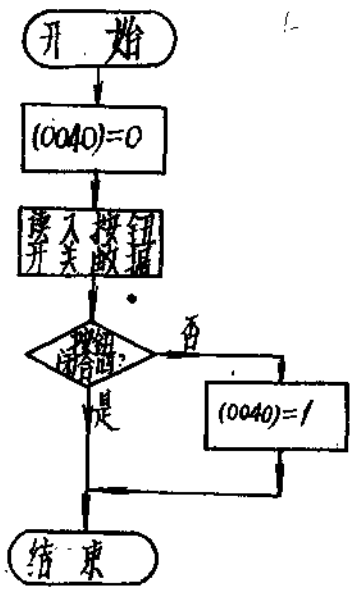

图7－2　确定开关闭合给已知单元置数的程序流程图

源程序：

```
LD   A,01001111B      ;  定义口A为输入口
OUT (PIOCRA) , A      ;
```


此为试读，需要完整PDF请访问：www.ertongbook.com

```
        LD   HL , 40H          ;  标志＝0
        LD   (HL), 0           ;
        IN   A , (PIODRA)      ;  读按钮位置
        AND  MASK              ;  按钮闭合吗?
        JR   Z , DONE          ;  是，转DONE
        INC  (HL)              ;  否，标志＝1
DONE:   HALT                   ;
```

对源程序的说明：

①口地址PIOCRA，PIODRA的具体值决定于PIO如何接入微处理机。

②此程序中PIO A口被定义为输入口，工作于字节输入方式。也可以使PIO工作于位控方式而完成上述功能，只要利用下列初始化程序就可以使PIO A口工作于位控方式。

```
        LD   A , 11001111B     ;  定义口A为控制口
        OUT  (PIOCRA), A       ;
        LD   A , 0FFH          ;  全部位为输入
        OUT  (PIOCRA), A       ;
```

③MASK取决于按钮开关接入的是PA0～PA7中的哪一位，在按钮接入位置为1，否则为0，如表7－1所示。

④上程序中利用指令AND MASK来判断开关是否闭合，而当接入PA0或PA7时，则利用移位指令来判断也很方便，同学可自行编出程序。

按钮接入位置	MASK	
	2 进 制	16 进制
PA0	00000001	01
PA1	00000010	02
PA2	00000100	04
PA3	00001000	08
PA4	00010000	10
PA5	00100000	20
PA6	01000000	40
PA7	10000000	80

表 7－1　MASK 与按钮接入位置对应表

2、计数按钮开关闭合次数

在按钮开关每次闭合后，使存贮单元0040的内容加1，以记录它的闭合次数。

为了记录开关按下次数，必须确保每次闭合产生一次状态转移。但是，由于机械开关触点在稳定于最终位置之前要弹跳（颤动），即按钮开关每次按下不止产生一次状态转换。所以，CPU在发现一次闭合后等待一段时间，便可以消除按钮开关的弹跳。所需的延时称为消除弹跳时间，此时间是按钮开关技术指标之一，它一般为几个毫秒。在此时间内，程序不应检查开关的状态，CPU不能读取数据，因为这样会把弹跳误认为是闭合。程序可以进入延时例程或在此时间内执行其他任务。即使程序延迟一段消除弹跳时间后，它仍然必须等待本次闭合终止，然后才能去查看是否发生一次新的闭合。这样可避免重复计数。图 7－3 是实现上述功能的程序流程

图。

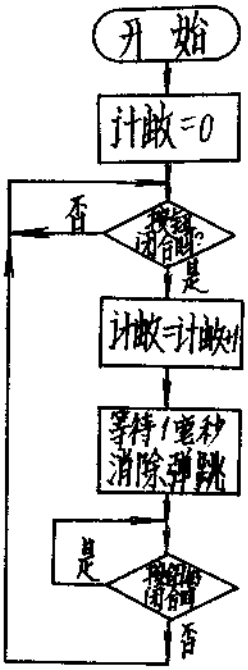

图7－3　计数按钮开关闭合次数程序流程图

源程序：

```
        LD   A , 01001111B      ; 定义PIO A口为输入口
        OUT ( PIOCRA), A        ;
        LD   HL , 40H           ; 闭合计数＝0
        LD   (HL), 0            ;
CHKCL:  IN A , (PIODRA)         ; 读按钮位置
        AND MASK                ; 按钮按下(0)吗?
        JR NZ, CHKCL            ; 否，等待到按钮按下
        INC (HL)                ; 是，闭合计数加1
        CALL DELAY              ; 等待1ms 消除弹跳
CHKOP:  IN    A , (PIODRA)      ; 读按钮位置
```

```
        AND   MASK          ; 按钮仍在按下吗?
        JR    Z , CHKOP     ; 是, 等待释放
        JR    CHKCL         ; 否, 查看下一次闭合
DELAY:  LD C , MSCNT        ; 取相当于1ms延时的计数值
DLY1 :  DEC C               ; 计数=计数-1
        JR NZ , DLY1        ; 计数到C
        RET                 ;
```

如前所述，对于这种简单的接口，实际上并不需要PIO，只要一个可寻址的三态缓冲器即可。

二、搬钮开关

单刀双掷开关是最常用也是最简单的搬钮开关，它是一种不是处于常闭状态就是处于常开状态的机械开关。下面讨论一个单刀双掷搬钮开关经PIO接到Z80CPU，电路如图7－4所示。

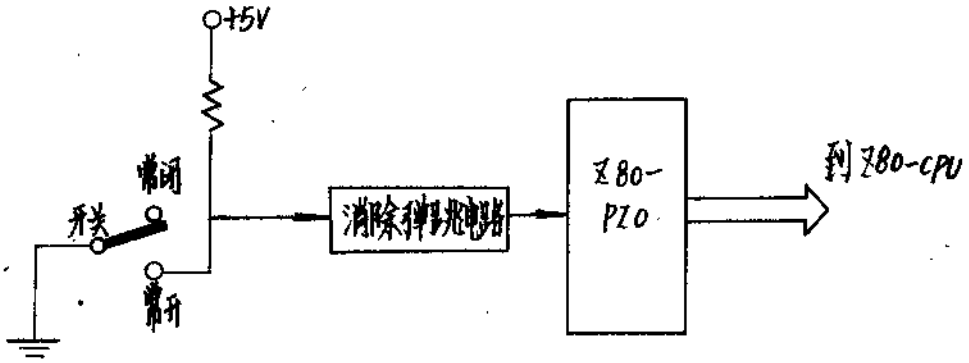

图7－4　搬钮开关接口电路

与按钮开关相类似，这里搬钮开关用Z80PIO的一位作为可寻址缓冲器，当开关处于常闭位置时，输入到PIO的为“1”；当开关处于常开位置时，输入到PIO的为“0”。上图中的消除弹跳电路可用脉宽几个毫秒

的单稳电路或一对交叉耦合的与非门来实现。因为即使开关在新状态（位置）上稳定下来前发生弹跳。但这些电路仍将只产生一个阶跃或脉冲来响应开关位置的一次变化。下面利用上面接口电路完成两个功能：①开关闭合时使一存贮单元置1。②开关状态改变时使一存贮单元置1。

1、等待开关闭合

开关闭合前存贮单元0040置0，闭合后置1，即处理机使存贮单元0040清零，等待开关闭合，闭合后存贮单元0040置1。其程序流程如图7－5所示。

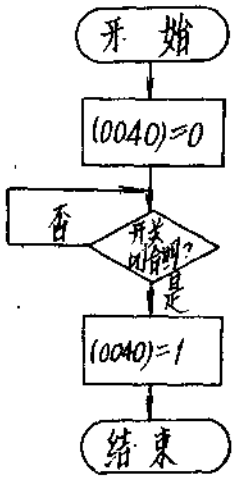

图7－5　等待开关闭合给已知单元置数的流程图

源程序：

```
        LD   A , 01001111B    ;  定义PIO A口为输入口
        OUT  (PIOCRA), A      ;
        LD   HL , 40H         ;
        LD   (HL), 0          ;  标志=0
WAITC:  IN   A,(PIODRA)       ;  读开关位置
        AND  MASK             ;  开关闭合(0)吗?
```

```
        JR      NZ , WAITC      ;  否，等待开关闭合
        INC     (HL)            ;  是，标志＝1
        HALT                    ;
```

这里的MASK决定于开关接入PIO的位置。

2、等待开关位置改变

直到开关位置改变前存贮单元0040一直为零，即处理机等待开关位置改变后使存贮单元0040置1。其流程图如图7－6所示。

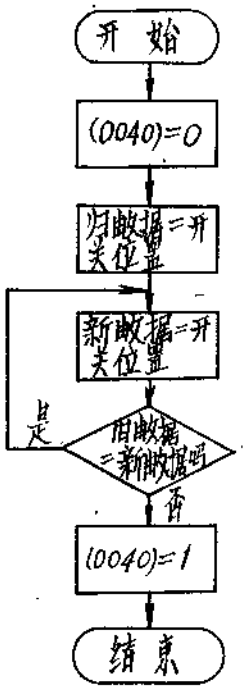

图 7-6 等待开关位置改变给已知单元置数程序流程图

源程序：

```
        LD      A , 01001111B   ;  定义PIO A口为输入口
        OUT     (PIOCRA), A     ;
        LD      HL , 40H        ;
        LD      (HL), 0         ;  标志＝0
```


此为试读，需要完整PDF请访问：www.ertongbook.com