

《单片机原理及接口技术》复习纲要

第0章 绪论+计算机的基础知识

1. [理解]何谓**单片机**? (P3) 微型计算机的系统结构 (P6 图 0-1 或<参考>\戴梅萼的《微型计算机技术及应用》教材图 1.1 微处理器、微型计算机和微型计算机系统三者的关系)。微型计算机的**三总线结构** (P7 图 0-2、<参考>\戴梅萼教材的图 1.2 微型计算机的基本结构或<参考>\周明德教材第 1 章 PPT45~46)。
2. [了解]在系统编程 **ISP** 和在应用编程 **IAP** (P5)
3. [了解]微型计算机的基本工作原理 (P7§0.1.2)
4. [掌握、重点、难点]符号数的原码、反码、补码表示 (重点)。计算机中, 凡是符号数一律用补码表示。两数相减是化归为补码的加法完成的。即 $[x-y]_{\text{补}} = [x]_{\text{补}} + [-y]_{\text{补}}$ 。但要注意**减法借位标志**与补码加法的**进位标志**之间的**互反关系**。理解此互反关系是理解多字节 BCD 码减法程序 (P54[例 3-12]) 的基础。[例程]例 0-2,<Demo>\Chap00 [补充教学]**减法与补码**: 从补码的几何实质入手, 讲透了补码及其上溢、下溢条件。
5. [理解、重点] 溢出标志 **OV** 和进位标志 **CY**。 $OV = CY \oplus CY_{-1}$, 最终的 **CY** 在加法运算中表现为**进位**, 在减法运算中表现为**借位**。[习题 0.5]
6. [掌握、重点]**BCD 码** (尤其是**压缩 BCD 码**)。[难点]BCD 的运算。以加法为例, 两个压缩 BCD 码的整数相加后, 结果不是 BCD 码, 若想保持其结果仍为压缩 BCD 码表示, 必须进行十进制调整。这便是 51MCU 中 DA 指令的由来, 8086CPU 的 DAA 指令也作同样的事。甚至 8086 还为 BCD 码减法提供了 DAS 调整指令。[习题 0.6]

第1章 51 单片机结构

1. [了解]51MCU 内部结构 (P15 图 1-1)。
2. [识记、理解]**普林斯顿结构**和**哈佛结构** (P15 图 1-2)。[难点]51 单片机的存

存储空间物理上分四块：片内/片外程序存储器、片内/片外数据存储器。逻辑上有三个存储器地址空间：片内数据存储器、片外数据存储器及片内、片外统一编址的程序存储器（P16 图 1-3）。[扩展]两种结构不是对立关系，很容易由哈佛结构模拟出普林斯顿结构。51 单片机本身采用哈佛结构，但通过将 51 输出的控制信号 \overline{PSEN} 和 \overline{RD} “相与”后接到同一个存储器芯片的 \overline{OE} 引脚，就演变为普林斯顿结构（P157 图 9-11）。

3. [理解、重点]对于 51 系列，片内 RAM 为 128 个字节；对于 52 系列，片内 RAM 有 256 个字节，如何区分 80H~FFH 访问的是 RAM 还是 SFR？靠寻址方式区分！（P17 图 1-4）[识记]片内 RAM 的分配（P18 表 1-1）[习题 1.6]
4. [理解、熟悉]特殊功能寄存器 SFR（P19 表 1-2）[习题 1.6]
5. [理解、重点、难点]程序状态字 PSW。这是 CPU 实现条件转移指令的基础，而后者又是高级语言实现分支和循环结构的基础。要点：①CY 对于加法为进位，对于减法为借位；②RS1 和 RS0，选择工作寄存器组；[习题 1.7] ③SP 和 P0~P3 的复位状态（重点），其余大多数复位值为 0。（P22 表 1-4）[习题 1.11]。[编程忠告]若程序用到堆栈，通常要初始化 SP。
6. [识记、重点]晶振周期、机器周期和指令周期三者的关系（P21§1.4.2）。[参考]李广弟教材第 2 章 PPT25 [编程影响]软件延时就利用了此关系（P52 例 3-9）。[注意]实验开发板 QX-mini51 所用单片机 STC12C5A60S2 对传统 8051 的 111 条指令执行全面提速，因此精确延时变得困难。详见芯片手册 STC12C5A60S2.pdf 的附录 I（p446/463）。

第 2 章 51 单片机的指令系统

1. [理解、掌握]51MCU 指令的寻址方式有七种：立即寻址、直接寻址、寄存器寻址、寄存器间接寻址、变址寻址、相对寻址、位寻址。[习题 2.4]，[编程影响]变址寻址（`MOVC A, @A+DPTR` 或 `MOVC A, @A+PC`）对于查表技术很关键。寄存器间接寻址（比如读操作 `MOVX A, @DPTR` 和写操作 `MOVX @DPTR, A`）对于访问外部 RAM 或外部 I/O 接口很重要。
2. [理解、重点、熟悉] 51MCU 共有 111 条指令，大致分为四类：数据传送与交换指令、算术运算和逻辑运算指令、控制转移指令、位操作指令。[习题] 2.6、

2.11、2.12、2.15。

3. [熟记、重点]算术运算和逻辑运算指令对 PSW 的影响 (P33§2.3.1): ①P (奇偶) 标志仅对累加器 A 操作的指令有影响。它是**偶校验**, 即数据位加上校验位本身, 其中的“1”的个数必为偶数。**奇校验**与之相反。[P60 习题 3.8]; ②**传送指令、加 1、减 1 指令、逻辑运算指令不影响 CY、OV、AC 标志位**; ③**加、减运算指令影响 P、OV、CY、AC 四个测试标志位, 乘、指令使 CY=0, 当乘积大于 255, 或除数为 0 时, OV=1。**[难点]注意到指令系统中无 DEC DPTR 指令, 如何实现? (本章**补充题 2**)
4. [重点、掌握]DA 指令、DIV AB、MUL AB。前两者对于计数值拆分为百位、十位和个位很重要。应用见[实验 3]。[难点]DA 指令的实现原理 (本章**补充题 3**)。
5. [理解、难点] 控制转移指令 LCALL addr16, 先将**断点** (下一条指令的地址) 压栈, 再令 $PC \leftarrow \text{addr16}$ 。保护断点的细节是先 PUSH PCL, 再 PUSH PCH。RET 指令恢复断点继续执行, 其实质是先 POP PCH, 再 POP PCL。间接转移指令 JMP @A+DPTR 对于多分支**散转**程序很有用。[例程]P30 例 2-22、P50 例 3-7、P51 例 3-8。

第 3 章 51 单片机汇编语言程序设计

1. [了解]51 单片机编程主要是用**汇编和 C 语言**, 最终都必须转换为**机器码**才能执行。(P45 图 3-1)。关于 51 单片机各指令的机器码、字节数及指令周期可参见 P293 附录 A 51 单片机指令表。上一章[习题 2.10]展示了**手工汇编**的技术, 有助于理解**汇编器 (Assembler)**是如何工作的?
2. [理解、识记]构建程序的三种基本结构:**顺序结构、选择结构、循环结构**。它们的共同点:**单入口、单出口**。这有效降低了**调试和测试的难度**。
3. [熟悉]51 汇编常用**伪指令** (P46§3.2): ORG、DB、DW、DS、EQU、BIT、END。注意: 伪指令主要用于为汇编器提供一些信息帮助其汇编, 本身并没有对应的机器码。
4. [重点]**查表技术**。**表的本质为数组这种数据结构**。借助索引 (数组下标) 访问数组元素。其基本手法是**以空间换时间**。[P48 例 3-3]、[实验 3]

5. [重点、难点]**多分支散转程序**。基于查表技术,多利用 **JMP @A+DPTR** 指令。有两类设计方法:①查**转移地址表**;②查**转移指令表**。[例程]P50 例 3-7, P51 例 3-8。
6. [重点]**延时子程序**。[P52 例 3-9]。[应用注意]当今的 51 芯片多数已优化提速,与原先的 51 内核并不完全兼容,因此延时值可能并不精确,具体查看芯片手册。如果要精确延时,请借助第 7 章的定时器。
7. [重点、难点]**冒泡法排序**。[实验 1]
8. [重点]利用位逻辑运算,软件可以**模拟硬件电路**。[例 3-14]、[习题 3.12]。
9. [掌握]**奇校验和偶校验**。[P60 习题 3.8]

第 5 章 输入、输出接口 P0~P3

1. [理解、重点、难点]为何 CPU 与 I/O 外设之间需要 **I/O 接口**? CPU 与 I/O 外设之间交换三类信息:**数据、状态和控制**。其中数据又有 3 种基本形式:**数字量、模拟量、开关量**。引入 I/O 接口后,接口中存放这三类信息的寄存器分别称作**数据口、状态口和控制口**。并非所有外围芯片必须都有这三类端口,比如 8255 没有状态口,而 8155 则三类口全有。[参考]戴梅萼的微机原理及接口技术教材的图 1.2 给出了当今微型计算机的基本结构 (**What?**);李广弟教材第 7 章 PPT3~4 对引入 I/O 接口的必要性作了精彩的讲解 (**Why?**);周明德教材第 5 章 PPT11~15 对 CPU 与 I/O 外设之间传送的信息有清晰的分类 (**How?**)。计算机本质上是二进制机器,只能处理离散量。那模拟量如何进入计算机进行处理,反过来计算机又如何控制模拟设备?这便是 **A/D 转换**和 **D/A 转换**的产生背景。
2. [了解、重点、难点]CPU 与 I/O 外设之间的数据传送方式:**程序方式、中断方式、DMA 方式**。程序方式又细分为**无条件**和**条件传送**两种方式。一般来说,在无条件程序方式下,输入外设需接三态缓冲器、输出外设需接锁存器。可以认为它们时刻为输入或输出作好准备。这对于极慢速设备是没问题的。本章的开关和 LED 灯、数码管属于典型的无条件传送程序方式。[参考]戴梅萼教材 P182(有照片)§5.5 概括了 **CPU 和外设之间的三种数据传送方式**、戴梅萼教材的图 5.3 **无条件传送方式**的工作原理、图 5.4 **查询式输入**的接口、

图 5.5 查询式输出的接口、图 5.7 中断方式输入的接口电路。周明德教材第 5 章 PPT121 图 5-26 (查询式输入接口电路)、PPT125 图 5-29 (查询式输出接口电路)、PPT134 图 5-33 (中断式输入接口电路)。

3. [重点、理解、难点]P0~P3 的位结构 (P87 图 5-2)。[要点]①系统扩展时, 51 对外开放内部总线。具体地 P2 提供高 8 位地址, P0 提供低 8 位地址, 同时复用为 8 位数据总线。这两个端口中的多路切换开关及总线控制信号就是用于这个目的。注意: 当 P2 口的几位作为地址使用时, 剩下的 P2 口线就不能作为普通 I/O 口线使用; P0 无上拉电阻, 作普通 I/O 口使用时须外接上拉电阻。②P3 口有很多第二功能。可以某些口线设为第二功能, 余下的用作普通 I/O 口。③P1 常作普通 I/O 口使用。④51 的 P0~P3 为**准**双向口, 即当用作输入口线时, 为了能准确读取引脚的信号, 必须先向锁存器写“1”, 以让电路图中引脚右下方的场效应管 VT 截止, 确保它不被拉低为低电平“0”。[区分]MOV P1,A (写锁存器) 和 MOV A,P1 (读引脚电平) [实验 2]、[例 5-2] [引申]**跑马灯实验**是个基础但重要的实验, 实验 2 是用软件延时方式实现, 其思考题 3 引出按键的查询式编程 (而实验 3 就是按键的中断式编程); 习题 6.6 就是用按键触发外部中断的方式实现跑马灯的移位; 实验 4 基础内容及思考题 1 就是利用定时器分别以查询方式和中断方式实现跑马灯。李忠的《穿越计算机的迷雾》§8.3 给出了跑马灯的纯硬件实现方式: 利用上升沿 D 触发器构建循环移位寄存器来实现之。**循环移位寄存器**本身是个非常有用的概念, 可用在 CPU 的 CU (Control Unit) 设计中, 参考李忠第 10-11 章和 Charles Petzold 的名著“CODE”的 Technical Addendum; 也可利用它构造一般的 n 分频器 (n>1)。剪掉首尾之间的连线形成的**移位寄存器**又是串并转换的硬件基础。
4. [重点、难点]七段数码管。[要点]①**共阴接法**和**共阳接法**。段码表互补; ②**静态显示**和**动态显示**; ③动态显示时, 需两个输出接口, 一个输出**字形码**。一个提供**位选码**。[实验 3]、[P98 习题 5.8]
5. [重点、难点]独立式按键、行列式键盘。键盘扫描程序流程 (P96 图 5-10)。引出**按键去抖**的有趣问题 (实际工程的复杂性总是出人意料!)。软件去抖最简单的方法是**延时**, 高级的做法是基于**状态机**和定时器来实现 (可参见马

潮编著的《AVR 单片机嵌入式系统原理与应用实践》（第 3 版）P280§9.2）。硬件消抖电路有**基本 RS 触发器**、**RC 积分电路**等。关于基本 RS 触发器消抖电路，应用见 P104 例 6-1、例 6-2，其消抖原理分析见 P186§10.5.1。[参考]李广弟教材第 7 章 PPT29（键闭合和断开时的电压抖动）[实验 4]涉及如何在中断服务程序中消抖的难题。

第 6 章 51 单片机的中断系统

1. [理解、重点、难点]中断技术使用**异步**的**编程**范式，而**查询**使用**同步**的编程范式。[参考]Comer 的 ECA 课件 P607-644 讨论了程序方式和中断方式的 I/O。尤其请关注 P609-610、618-624，两者比较见 P620。[编程忠告]**不要以同步思维来进行异步编程**。教材 P104 例 6-2 的方法一就是反面教材。
<Demo>\<Chap06>有关于此程序的持续改进，其中 eg6-2v2(2).asm 为最终版本。教材 P180-181 的 AD 转换例程是另一个反面教材，其改写见 <Demo>\<Chap10>\P180.asm。
2. [理解、重点、难点]中断处理的过程（P102 图 6-2）：中断**请求**、中断**响应**、中断**服务**、中断**返回**。[编程忠告]中断返回指令是 RETI，不能用 RET 代替。原因是中断处理完后，**中断激活触发器**须清零，否则以后就无法响应中断了。
[参考]请参考周明德教材第 167 页的照片理解之。
3. [理解、重点]51 单片机的**中断源**（P100 表 6-1）和**中断系统**（P101 图 6-1）。
[习题]6.2、6.3
4. [掌握、重点、难点]中断服务程序（ISR，Interrupt Service Routine）的编写。
要点：①**程序员提供 ISR，而中断系统决定何时调用它**。这与 GUI 编程的**回调函数**概念类似：程序员提供处理消息的窗口过程，而操作系统根据消息队列调用相应的窗口过程。只不过由于中断可以嵌套（中断服务程序在执行过程中有可能被更高级的中断打断），从而更复杂。②**注意清除中断请求标志**，以避免 CPU 重复响应。有些中断请求标志系统在响应中断时会自动清除（比如定时器 T0 的溢出标志 TF0，下降沿触发方式的外部中断请求标志 IE1 等），而有些则需要程序员手动清除。比如 51 为了支持全双工通信，串行发送和串行接收共用一个中断向量地址 0023H，显然中断响应时系统不能自动清除

它们, 否则程序员无法判断当前该处理哪种中断(发送 or 接收)? 此时程序员在利用完 TI 或 RI 标志后应该记得清除之。③注意中断随时可能发生, 因此应确保中断返回时, 原先被中断的程序从断点处继续往下执行时语义不变, 就像不曾被中断一样。这就要求编写 ISR 时特别小心。关键之处是确保中断返回时不会影响 PSW。方法就是将 PSW 视作现场的一部分。[参考]Comer 的 ECA 课件 P624, [实验 3], [习题]6.6、6.7

5. [了解、难点]外部中断源的扩展问题。方法①采用查询法扩展外部中断源。技术细节(见本章 PPT27 补充例 3): 硬件电路中, 它们通过“或非门”接到 51 外部中断引脚, 然后它们又各自接到 51 的某个输入口线上。软件编程时, 在中断服务程序中再查询各输入口线的状态并作相应处理; 方法②利用 T0 和 T1 的方式 2 计数功能(初值设为 255)。(描述见下一章 PPT36), [实例]<Demo>\Chap08\LEDController.asm。

第 7 章 单片机的实时/计数器

1. [了解、难点]计数器电路是理解 51 定时器/计数器的基础。[参考]李忠的《穿越计算机的迷雾》P142 图 9.7 以及 Charles Petzold 的名著《编码的奥秘》(CODE 的中译本) P125~128 (尤其 P127 的电路及时序图)。
2. [了解]定时的 3 种主要方法: 软件定时、数字电路硬件定时、可编程定时/计数器(如 8253 芯片、51 单片机内的 T0 和 T1)。
3. [理解]51 单片机 T0/T1 的内部结构和控制信号(P109 图 7-1, 方式 1)。
4. [熟悉、重点]方式寄存器 TMOD 和控制寄存器 TCON (P110 表 7-1、P111 表 7-2)。[习题 7.1]
5. [重点、识记]定时/计数器的 4 种工作方式。[参考]李广弟教材 Ch5 课件 PPT28 (方式 0)、35 (方式 2)、40 (T0 方式 3)、42 (T0 方式 3 下的 T1)。重点掌握方式 1 和方式 2。
6. [重点、难点]定时器/计数器的编程。要点: ①查询方式和中断方式。[实验 4]、[习题 7.5] [实例] <Demo>\Chap07\MyTimer.asm 展示了同时利用两个定时器分别以查询方式和中断方式在两个指定引脚输出周期为 4ms 且占空比为 0.25 的方波。②可以利用方式 2 计数器功能扩展外部中断源(本章 PPT36)。[实

例] <Demo>\Chap08\LEDController.asm。

第8章 单片机的串行接口

1. [理解]并行通信和串行通信的概念 (P121 图 8-1)。
2. [理解、重点、难点]异步通信和同步通信的概念 (P122 前两幅图)。重点是异步通信。一帧数据由起始位、数据位、校验位 (可选) 和停止位构成。
3. [理解、识记]3 种通信线路形式: 单工、半双工和全双工。(见 P122 图 8-2)。
4. [理解、难点]串并转换的硬件基础是移位寄存器。[参考] 李忠的《穿越计算机的迷雾》P129 图 8.14 上剪掉首尾之间的连线 (原图是循环移位寄存器), 就得到移位寄存器。
5. [理解]波特率、比特率、波特率发生器 (P123§8.1.5)。[要点]51MCU 根据波特率发生器的移位时钟来自系统时钟分频值还是定时器 T1 的溢出标志区分为固定波特率和可变波特率方式。[习题]8.2、8.3、8.4
6. [理解、重点]51 串行口结构 (P127 图 8-7, 注意发送 SBUF 和接收 SBUF 这两个寄存器地址都是 99H, 但实际上是两个独立的硬件)、串行传送示意图 (P128 图 8-8)。
7. [理解、识记]51 单片机的波特率发生器及串行口的工作方式 (P130§8.4)。[参考]波特率发生器的示意图请参考周明德教材第 9 章 PPT74 图 9-34。[要点]四种工作方式: 方式 0 用于同步移位寄存器, 可扩展 I/O 口 ([P137 例 8-4])、方式 1 用于双机通信、方式 2 和方式 3 用于多机通信。方式 0 和方式 2 是固定波特率, 方式 1 和方式 3 为可变波特率。重点掌握方式 1。
8. [理解]控制寄存器 SCON 及电源控制寄存器 PCON 中的 SMOD 位 (波特率加倍位)。[难点]多机通信的关键是区分地址帧和数据帧。TB8、RB8 和 SM2 发挥了关键作用 (P129§8.3.1)。
9. [重点、难点]串行口的编程。两种方式: ①查询方式; [要点]先发后查、先查后收; 程序流程参考 P131 图 8-9。[习题 8.5] ②中断方式。程序流程参考 P132 图 8-10。[习题 8.6] [实例]<Demo>\Chap08\eg8-1 (发送+接收、中断和查询两个版本)、eg8-2、LEDTerminal+LEDController。

第9章 单片机总线与系统扩展

1. [理解、重点、难点]三总线（地址总线、数据总线和控制总线）与系统扩展。

所谓单片机的系统扩展就是将外接数据存储器、程序存储器和 I/O 接口与三总线相连 (P149 图 9-1)。51MCU 的 P0 和 P2 端口可对外开放内部总线 (P87 图 5-2)，P2 提供高 8 位地址线，P0 为低 8 位地址线和数据线分时复用总线（引脚 ALE 的输出信号接地址锁存器）。控制信号主要是 \overline{PSEN} （外扩 ROM）、 \overline{RD} 和 \overline{WR} （外扩 RAM 或 I/O 接口）。[参考]关于三总线结构，请参考戴梅萼教材的图 1.2 微型计算机的基本结构或周明德教材第 1 章 PPT45~46。周明德教材第 6 章 PPT5 图 6-1 图示了微型机中各级总线的分类。总线扩展（尤其数据总线）必然会遇到三态逻辑的问题。所谓“三态”是指除了可输出高电平状态或低电平状态外，还可处于断开（高阻）状态。李广弟教材第 7 章 PPT4 提到的数据总线隔离就是利用三态门。李忠的《穿越计算机的迷雾》在 §10.2 轮流使用总线中利用继电器形象示意了三态传输门的构造原理（图 10.7）。外围芯片通常有一个片选引脚 \overline{CS} （或 \overline{CE} ），就是基于三态门利用它设置芯片与总线之间隔离与否。

2. [理解、重点、难点]系统扩展的方法。①数据线的连接。外接芯片的数据线 D0~D7 接单片机的 P0 端口。当驱动力不够时，在两者之间加驱动器；②控制线的连接。51 的 \overline{PSEN} 接外扩 ROM 的 \overline{OE} ，51 的 \overline{RD} 接外扩 RAM 或 I/O 接口的 \overline{OE} 或 \overline{RD} ，51 的 \overline{WR} 接外扩 RAM 或 I/O 接口的 \overline{WR} 或 \overline{WE} ；③地址线的连接。许多外围芯片（比如 8255、8155）提供地址线选择内部的存储单元或端口，本身还有一个片选引脚 \overline{CS} 或 \overline{CE} 。此时 51 连接外围芯片的地址线概念上分为两部分：字选地址（片内选择）和片选地址。通常，字选地址直接接 MCU 从 A0 开始的低位地址线，而片选引脚的接法有 3 种（P150 图 9-3）：线选法、译码法、直接接地。前两者是重点。译码法又分为部分译码和全译码；地址译码器既可以用逻辑门搭建（P151 图 9-4），也可以选用译码器芯片（如 74LS138，P152 图 9-6 和 9-7）。

3. [理解、重点]系统扩展的原则：使用相同控制信号的芯片之间不能有相同的

地址, 使用相同地址的芯片之间控制信号不能相同。一句话, 不能有总线冲突。比如, 外接 I/O 和外扩 RAM 接法一样 (51 为统一编址), 靠地址区分; 外扩 RAM 和外扩 ROM 地址空间重叠, 靠控制信号区分。[习题 9.1] [扩展]51 单片机本身采用哈佛结构, 但通过将 51 输出的控制信号 \overline{PSEN} 和 \overline{RD} “相与” 后接到同一个存储器芯片的 \overline{OE} 引脚 (P157 图 9-11), 就演变为普林斯顿结构。

李广弟教材第 6 章 PPT 24 基于此原理, 给出了一个有趣的应用实例。

4. [了解、难点]几个重要时序: 从外部 ROM 取指令时序 (P156 图 9-10)、外部 RAM 读时序 (P159 图 9-12) 和写时序 (图 9-13)。[参考]更精确的描述见周明德教材第 6 章 PPT16-17、21、26 和 28。原始出处可能是孙育才编著的《MCS-51 系列单片机及其应用》(第 6 版) 的 §2.3.3 “振荡器、时钟电路及时序” (P20~24) 以及 §3.3 “外部存储器与访问” (P38~41)。
5. [重点、难点]并行 I/O 接口的扩展。51 对扩展的 I/O 接口和外部 RAM 采用统一编址的方案: 相同的控制信号、寻址方式和指令。①通用锁存器、缓冲器的扩展。三态缓冲器常用于输入, 锁存器常用于输出。[例程 9-3] [参考] 戴梅萼教材的图 5.3 无条件传送方式的工作原理。②可编程并行接口芯片的扩展。由于它专为和计算机接口而设计, 它有和计算机接口的三总线引脚, 与 CPU 或 MCU 连接非常方便。如 8255、8155 芯片。[比较]8086CPU 采用独立编址方案。RAM 寻址空间 (1MB): 00000H~FFFFFFH, 而 I/O 端口寻址空间 (64KB): 0000H~FFFFH。地址重叠也没关系, 因为 CPU 的输出引脚 $\overline{M}/\overline{IO}$ 就是用于区分它们, 请参考戴梅萼教材的图 2.3 8086 的引脚信号结合图 5.4 查询式输入的接口、图 5.5 查询式输出的接口来理解之。
6. [熟悉、重点]并行接口芯片 8255。要点: ①片内地址 A1A0 为 00、01、10 和 11 时, 分别对应三个数据口 A 口、B 口、C 口和控制口; ②[重点]根据控制口的 D7 为 1 或 0, 对应工作方式控制字 (P164 表格) 或 C 口置位/复位控制字 (P165 表 9-4); ③[难点]三种工作方式: 方式 0 (基本方式)、方式 1 (选通方式)、方式 2 (双向传输方式)。[重点]方式 0 最简单, 无须联络信号。当 CPU 和 I/O 外设之间数据传送方式为无条件的程序方式时, 可以选择 8255 作为两者之间的 I/O 接口, 选用方式 0 即可; 当 CPU 和 I/O 外设之间数据传

送为条件查询的程序方式或者中断方式时, 选用方式 1 或方式 2。电路连接参考 P166 图 9-20。[参考]关于 8255 工作方式 1 的完整时序见<参考>\戴梅萼教材的图 6.23 和图 6.25。[习题 9.9]

7. [了解]多功能接口芯片 **8155**。它集成了 256B 的静态 RAM、2 个 8 位和 1 个 6 位的可编程并行 I/O 端口、一个 14 位的有多种工作方式的减法计数器以及 1 个地址锁存器。其内部结构及地址分配见 P169 图 9-24。注意 $\text{IO}/\overline{\text{M}}$ 为端口/存储器选择信号。从概念上讲, 它构成字选地址的一位。8155 靠它来确定是访问内部的 RAM 还是端口。
8. [重点、掌握]存储器和 I/O 口的综合扩展电路。[线选法举例]P168 图 9-23 (PPT50~51)、[译码法举例]本章 PPT60~62。[习题] 9.3、9.5

第 10 章 单片机应用接口技术

1. [理解、重点]A/D 转换和 D/A 转换的基本概念 (P174 图 10-1)。当 CPU 和 I/O 外设之间交换的数据为模拟量 (即连续量) 时, 就会自然引出 A/D 转换和 D/A 转换的问题。因此, 它们具有基本的重要性, 极大地扩展了计算机的应用领域。技术指标一般了解即可。[参考] 戴梅萼的《微型计算机技术及应用》教材的图 10.1 一个包含 A/D 和 D/A 转换环节的实时控制系统。自然引出了测量系统和程序控制系统的概念。[题外话]构成信息技术的三大支柱: 计算机技术、测控技术和通信技术。单片机应用在这 3 个领域均有涉及, 特别是测控技术。
2. [重点、难点]DAC0832 芯片。要点: ①逻辑结构框图 (P175 图 10-2), 典型的两级缓冲结构; ②根据两级缓冲是直通或锁存, 得到三种工作方式: 直通方式、单缓冲、双缓冲。重点是单缓冲。双缓冲工作方式可以使数据接收和启动转换异步进行, 即在 D/A 转换的同时接收下一个待转换数据, 因而提高了通道的转换速率 (其实质就是流水线的思想)。双缓冲还有另一个重要应用: 要求多路 D/A 转换同步输出的场合 (比如 X-Y 绘图仪)。[参考]李广弟教材第 10 章 PPT28~31; ③如何输出锯齿波和三角波? [例 10-2]、[习题 10.1]
3. [重点、难点]ADC0809 芯片。要点: (1)结构框图 (P179 图 10-9), 它有 8 路模拟信号输入通道 (IN0~IN7), 可进行分时选通。通道选择通过地址

ADDC, ADDB 和 ADDA 来指定。A/D 转换的输入时钟信号由 CLK 提供。(2)几个重要的信号。START: **转换启动信号**。ADC0809 是正脉冲启动转换, START 上跳沿时, 所有内部寄存器清 0; START 下跳沿时, 开始进行 A/D 转换; 在 A/D 转换期间, START 应保持低电平。EOC: **转换结束信号**。启动转换后, EOC 变为低电平, 转换完成后变成高电平。OE: **输出允许信号**, 高电平时打开三态输出缓冲器, 使转换后的数字量从 $D_0 \sim D_7$ 输出; (3)CPU 获取 A/D 转换结果的方式。A/D 转换结束后, A/D 转换芯片会输出转换结束信号, 通知 CPU 读取转换数据。CPU 一般可采用四种方式和 A/D 转换器进行联络来实现对转换数据的读取: ①**延时方式**; ②**查询方式**; ③**中断方式**; ④**CPU 等待方式**。其中 CPU 等待方式需要 CPU 的支持(比如 8086 CPU 支持, 而 51 MCU 不支持)。其技术原理是利用 CPU 的 READY 输入引脚的功能, 设法在 A/D 转换期间使 READY 信号为低电平, 以使 CPU 处于等待状态, 而转换结束后, 则使 READY 成为高电平, 从而唤醒 CPU 去读取转换结果。此时的编程会相当简单。**[参考]**关于 CPU 等待方式见戴梅萼《微型计算机技术及应用》教材的 AD 转换相关照片。**[例题]**P180 图 10-10 为中断方式, 而 P79 例 4-15 为查询方式。**[习题]**10.5、10.6。

4. **[了解、难点] 按键去抖** (P186§10.5.1)。通常有**软件去抖**和**硬件去抖**两类方法。软件去抖最简单的方法是**延时**, 高级但复杂的做法是基于**状态机**和**定时器**来实现(可参见马潮编著的《AVR 单片机嵌入式系统原理与应用实践》(第 3 版) P280§9.2)。硬件去抖通常采用**基本 RS 触发器**来实现, 此外还有 RC 积分电路等方法。**[题外话]**关于状态机的系统阐述见离散数学教材。**[实验 4]**的选作内容“秒表”, 其最优雅的实现方式就是基于**有限状态自动机**。

第 12 章 以 MCU 为核心的嵌入式系统的设计与调试

1. **[了解]** 嵌入式系统的一般开发步骤 (P227 § 12.1.1): 分析课题要求→确定方案→**软硬件设计**→**软硬件联调**及纠错→绘制印制电路板→焊接形成产品。
2. **[了解]** 典型的以 MCU 为核心的嵌入式系统硬件结构 (P227 图 12-1)。
3. **[了解]** 嵌入式应用系统的设计原则 (P228 § 12.1.2): ①选择单片机机型; ②充分利用 MCU 的片内硬件资源, 简化系统扩展, 提高**可靠性**; ③了解服务对象特性, 进行一体化设计, 在性能指标上留有余地; ④在满足功能和性能要

求的前提下, 控制成本, 提高性价比; ⑤软件采用模块设计、便于编写、测试、调试、移植及维护; ⑥考虑应用系统的使用环境, 采取相应措施, 如抗干扰等。

4. [识记]嵌入式系统的开发工具 (P228 § 12. 1. 3): **PC、编程器和仿真机**。主要的单片机仿真调试集成软件包是 **Keil μ Vision**。
5. [识记]嵌入式系统的开发有两种方法 (P229 § 12. 1. 4): ①**PC+模拟仿真软件+编程器**; ②**PC+在线仿真器+编程器**。目前, **Proteus** 是流行的模拟仿真软件。
6. [了解]嵌入式系统的**抗干扰技术** (P230 § 12. 2)。有硬件抗干扰和软件抗干扰两类措施。软件抗干扰措施中重点是**看门狗 (Watch Dog Timer) 技术**。