

第一部分 单片机基础



本部分既是单片机系统的硬件基础,也是嵌入式系统学习的入门基础。介绍单片机系统结构与基本组成、存储结构,系统讲述单片机内部寄存器组及特殊功能寄存器,使学生掌握程序存储器、数据存储器地址分布、读写时序和读写操作,理解基本 I/O 端口结构与应用接口设计。

本部分包括:

第 1 章 单片机与嵌入式系统

本章简要介绍嵌入式系统学习中的几个基本概念及基本原理,包括微型计算机系统、单片机的技术特点、嵌入式系统、复杂指令集(CISC)与精简指令集(RISC)、嵌入式系统开发环境与交叉编译等。

第 2 章 体系结构

本章是单片机学习的重点和难点,是单片机系统设计的基础。介绍单片机内部逻辑结构、基本组成单元、工作方式、输入/输出端口结构与工作过程,讲述单片机基本工作电路的部结构、工作原理及应用设计。

第 3 章 存储结构

本章介绍单片机程序存储器基本特性及中断的硬跳转原理,讲述片内数据存储器的结构、地址分布及特殊功能寄存器。



1.1 嵌入式系统

将计算机基本组成单元集成在一个芯片上,是嵌入式系统的硬件基础。单片机的出现将计算机应用扩展到了消费电子、仪器仪表和检测与控制领域。

1.1.1 单片机与嵌入式系统

计算机系统由中央处理单元(CPU)、数据存储器(RAM)、程序存储器(ROM)、输入/输出接口以及系统总线构成。在微型计算机系统中,这些组成单元被集成为若干集成芯片或模块,组合安装在主板上,形成完整的计算机系统,简称微机系统。在单片机系统中,这些组成单元被集成在一片集成电路芯片中,称为单芯片计算机,简称单片机。

单片机具有体积小、功耗低、价格低、易于开发的特点,尤其适合作为控制器嵌入工业控制、家电产品、智能仪表、汽车电子、手持设备等系统中,构成嵌入式系统。伴随着在消费电子、汽车电子、网络通信、工业控制、航空航天等领域的广泛应用,嵌入式系统已成为计算机应用的热点。

嵌入式系统根据其内核嵌入的微处理器性能不同而应用于不同领域。在系统简单、速度慢、成本低等中低性能领域,采用以 MCS-51 为代表的 8 位机为主。在 PDA、手机、路由器、视频编解码器、图像处理等复杂高性能领域,采用以 ARM 为内核的 32 位机为主。

1.1.2 存储结构

嵌入式计算机系统存储结构有别于通用微机存储结构。

1. 普林斯顿结构

普林斯顿结构也称为冯·诺伊曼结构,如图 1-1 所示。程序存储器和数据存储器在物理上是同一个存储器,如 PC 中的主存储器 RAM,根据使用需要,把内存划分为程序存储区和数据存储区,取指令和取数据通过相同的地址总线 and 数据总线进行。以 Intel 80x86 CPU 为核心的微机系统是典型的普林斯顿存储结构计算机。

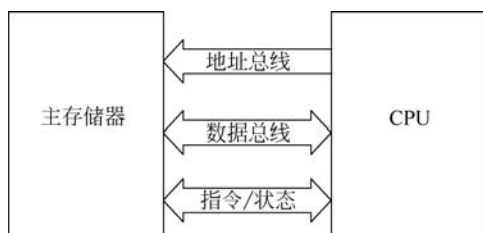


图 1-1 普林斯顿结构

2. 哈佛结构

哈佛存储结构如图 1-2 所示。程序存储器和数据存储器在物理上是两个相互独立的单元，它们共用地地址总线和数据线，但具有各自独立的读写控制指令。程序存储器采用非易失性存储器 EPROM、EEPROM 和 Flash 存储器，数据存储器采用易失性存储器 SRAM、DRAM 和 SDRAM。嵌入式系统普遍采用哈佛存储结构。

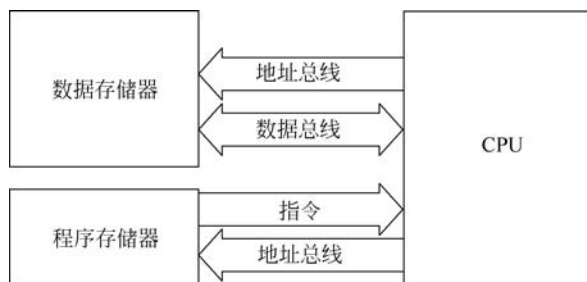


图 1-2 哈佛结构

1.1.3 CISC 和 RISC

指令集是 CPU 设计的核心，CISC 和 RISC 是两种不同设计理念的产物。

1. CISC

CISC(Complex Instruction Set Computer, 复杂指令集系统)是 CPU 指令集向后兼容的必然结果,具有完备而复杂的机器指令集合。复杂指令集 CPU 的复杂性在于微控制器的设计与实现,具有不等长指令编码及不固定的指令周期,程序设计方便,代码量小。

2. RISC

RISC(Reduced Instruction Set Computer, 精简指令集系统)是 CISC 指令集精简优化的子集,只包含使用频率高的少量指令(约占总指令数的 20%),配备大量通用寄存器,强调对指令流水线的优化,寻址方式少且简单。精简指令集 CPU 具有等长指令编码和单一指令周期,其复杂性在于编译程序和优化。

随着计算机技术的发展,CISC 和 RISC 在理念上相互借鉴,逐步趋同。RISC 设计正变得越来越复杂,而 CISC 在微处理器内部采用 RISC 架构,把复杂的指令由微程序通过多条

精简指令来实现。

1.1.4 分类与应用

嵌入式系统与通用计算机系统相比具有不同的技术要求和发展方向。通用计算机系统需要运算速度、总线速度的提升和存储容量的扩大,而嵌入式系统是以与控制对象密切相关的嵌入特性(体积、功耗、成本)、控制能力及可靠性为目标。

1. 嵌入式微处理器(MPU)

嵌入式微处理器的硬件基础是计算机中的微处理器。根据具体应用需求,将微处理器及其必备的 ROM、RAM、总线及外设接口装备在专门设计的印制电路板(Printed Circuit Board, PCB)上,以减小体积、功耗和成本,并且在系统散热、电磁兼容性、可靠性方面做特殊处理。因为其处理器、存储器、接口设计在一块 PCB 上,所以嵌入式微处理器也称为单板机。

2. 嵌入式微控制器(MCU)

以经过性能与功能裁剪的微处理器为核心,以满足某种特定的嵌入式控制需求为目的,将微处理器及其必备的 ROM、RAM、总线及外设接口集成在一个芯片上,成为单芯片计算机,从而使体积、功耗、成本大幅度降低,可靠性提高。

1980 年,Intel 推出 MCS-51 系列 8 位单芯片计算机,奠定了嵌入式系统的典型结构体系和应用模式。

随着超大规模集成电路工艺和集成制造技术的不断发展,出现了 8 位的 MCS-51、Z80、uPD78、MC68HC 系列,16 位的 MCS-96/98、M68HC16、NEC783 系列,32 位的 ARM、M68300、NECSH 系列等。

3. 数字信号处理器(DSP)

数字信号处理器(DSP)是嵌入式系统在信号处理领域的应用。随着嵌入式系统的智能化和网络化,具有智能化功能的产品(智能家电、智能家居终端、智能仪器仪表、图像图形处理、实时语音压缩/解压缩、虚拟现实、人工智能等)不断出现,使嵌入式系统从以控制为主向以信号处理为主、控制为辅的方向发展。嵌入式 DSP 有两个技术来源:一是以德州仪器(TI)为核心的 DSP 生产商,将 DSP 处理器单芯片化,增加片上资源和外围接口,成为嵌入式 DSP 处理器;二是原来的单片机生产商在单片机中增加信号处理协处理器(如 Intel MCS-296),成为具有 DSP 能力的单片机。

4. SoC

SoC(System on Chip, 片上系统)是高度集成化、固件化的系统集成技术,其核心思想是针对具体应用需求,从系统的整体功能和性能出发,用软硬结合的设计方法,把整个系统集成在一个芯片中。各种微处理器内核作为 SoC 设计的标准器件,用标准的硬件描述语言 VHDL 来描述,尽可能地将整个嵌入式系统集成到一片或几片集成电路芯片中。

SoC 采用芯片内部信号传输,大幅度降低功耗、体积和生产成本,提高系统性能和可靠性。

1.2 交叉编译

作为嵌入式系统的控制器的单片机,不管是8位、16位还是32位,由于受到其本身资源限制,应用程序均不能在其自身上开发,而需要一台具有高性能和高配置的通用计算机,如常用的PC和Windows操作系统,256MB以上内存+1GB以上硬盘,安装专用的开发平台(Keil μ Vision、MDK等),这样的通用计算机称为宿主机,嵌入式控制器的单片机称为目标机。应用程序在宿主机上开发,在目标机上运行,这种开发方式称为交叉编译。宿主机和目标机通过通信口(RS232/USB/并口)相连,调试好的程序可从宿主机下载(称为烧录或编程或固化)到目标机。

程序固化是单片机开发的最后一步,之后目标机(单片机)就可以独立执行嵌入式控制器的工作。

1.3 产品系列

嵌入式系统以满足特定需求为目标,以经过性能和功能裁剪的微处理器为核心,形成了众多系列产品。

1.3.1 MCS-51 系列

以控制为主,主要应用于数据处理量小而简单、速度要求在微秒级(指令周期几微秒或者十几微秒)、功耗低(工作电流在毫安级)、成本低(成本低于10美元)等场合,大多采用以MCS-51为代表的8/16位单片机,并以8位机为主。

目前主要8/16位机系列包括以下系列。

(1) PIC系列:美国微芯片公司(Microchip)研制生产,采用哈佛总线结构、二级流水作业、精简指令系统以及多种内嵌模块(WDT、ADC、CCP模块等)。

(2) AVR系列:美国Atmel公司研制生产,一种高性能、高速度和低功耗产品,常见的有TA90系列。

(3) MCS-51系列:美国Intel公司研制生产,应用最为广泛、最成熟的产品。配套的各种开发系统非常丰富,开发出多种系列产品。其中AT89系列为国内8/16单片机主流系列产品。

1. AT89 系列基本特性

AT89系列单片机是美国ATMEL公司研制生产的8位Flash系列单片机,与Intel公司生产的MCS-51兼容,是目前国内单片机开发主流单片机。具有如下特点:

- (1) 片内含有Flash程序存储器,可实现在系统编程。
- (2) 软硬件全面兼容MCS-51系列单片机。
- (3) 具有静态时钟方式,功耗降低,适合嵌入式及便携式系统。

(4) Flash 程序存储器可以多次快速擦写,适合学习、开发。

(5) 是目前国内最流行的单片机系列。

2. AT89 系列产品

标准型: 程序存储器由 EPROM 升级到 Flash 工艺,基本特性与 MCS-51 相同。AT89 标准型系列产品见表 1-1。

表 1-1 AT89 标准型系列产品

型号	ROM/RAM	中断源/定时器	其他
AT89C51	4KB+128B	5/2	
AT89LV51	4KB+128B	5/2	低电压
AT89C52	8KB+256B	8/3	
AT89LV52	8KB+256B	8/3	低电压
AT89C55	20KB	8/3	
AT89LV55	20KB	8/3	低电压

低配置型: 除 I/O 端口减少外,其他配置同 AT89C51,采用 DIP20 封装。AT89 低配置型系列产品见表 1-2。

表 1-2 AT89 低配置型系列产品

型号	ROM/RAM	中断源/定时器	其他
AT89C1051	1KB+64B	3/2	无串口
AT89C2051	2KB+128B	5/2	

高配置型: 在标准型的基础上增加了功能模块。Flash 程序存储器可通过 SPI 串行接口实现在线编程,具备看门狗、双数据指针。AT89 高配置型系列产品见表 1-3。

表 1-3 AT89 高配置型系列产品

型号	ROM/RAM	中断源	其他
AT89S53	12KB+128B	9	
AT89S8252	8KB+128B	9	2KB EEPROM
AT89S4D12	4KB+128B	9	Flash RAM

1.3.2 ARM 系列

随着嵌入式系统与 Internet 的结合,PDA、智能手机、路由器、调制解调器等应用不断出现,对嵌入式系统在信号处理能力、运算速度、通信时延等方面提出了更高要求,其代表为以具有 32 位精简指令系统(RISC)的 ARM 微处理器为核心的嵌入式系统,如意法半导体(ST)的 STM32 系列和韩国三星(Samsung)的 S3C2410 系列,其中 STM32 为目前国内主

流系列产品。

STM32 为基于 ARMCM3 内核的微控制器产品系列,在 2007 年由意法半导体 (STMicroelectronics)公司发布。STM32 提供了一个完整的 32 位产品系列,在保持高性能、低功耗、低电压的同时,继续保持高度的系统集成性和易开发性。STM32 具有完整的产业开发支持环境,包括免费的标准软件库、评估板和开发套件。

目前 STM32 主要应用于 PDA、路由器、调制解调器、网卡、通信等高速低功耗的高端应用。

1.3.3 TMS320 系列

数字滤波、谱分析、快速傅立叶变换等信号处理算法正越来越多地应用于嵌入式系统,DSP 应用从通用单片机中以普通运算指令实现 DSP 功能发展到嵌入式 DSP 处理器阶段。主流产品包括德州仪器(TI)的 TMS320C2000 和 TMS320C5000 系列,其中 C2000 系列主要应用于工业控制,C5000 系列主要应用于移动通信。

习题

1. 计算机如何改变我们的生活?
2. 简述嵌入式计算机系统与 PC 系统的异同。
3. 简述嵌入式系统开发过程。
4. CISC 与 RISC 的区别是什么?

第 2 章



体 系 结 构



微课视频

本章以 AT89C51 为模型机,介绍 MCS-51 系列单片机体系结构。

2.1 基本特性

AT89C51 为目前国内最为流行的 51 系列单片机,片内基本资源如下。

内部程序存储器(ROM): 4KB。

内部数据存储器(RAM): 256B。

寄存器组: 4 个寄存器组,每组有 R0~R7 8 个 8 位工作寄存器。

8 位并行 I/O 端口: P0、P1、P2 和 P3。

定时/计数器: 2 个 16 位定时/计数器 T/C0、T/C1。

串行通信端口: 一个全双工串行端口,RXD 接收、TXD 发送。

中断系统: 5 个中断源,定时/计数器 T/C0 和 T/C1、外部中断 INT0 和 INT1,串行通信中断 ES。

系统扩展能力: 可扩展 64KB 程序存储器(ROM)和 64KB 数据存储器(RAM)。

堆栈: 设在 RAM 单元,位置可以浮动,通过指针 SP 确定堆栈在 RAM 中的位置,系统复位时 SP=07H。

指令系统: 共 111 条汇编语言指令,按功能分为数据传送、算术运算、逻辑运算、控制转移和布尔操作 5 大类。

2.2 内部结构

系统内部结构见图 2-1。

系统由中央处理单元(CPU)、程序存储器(ROM)、数据存储器(RAM)、时序电路、并行端口、串行端口、中断系统及系统总线组成。

1. CPU

CPU 为整个系统的控制中心,由控制器和运算器(ALU)组成。其中控制器包括微程

序控制器、程序计数器(PC)、指令寄存器(IR)、指令译码寄存器(ID)、定时与控制单元。

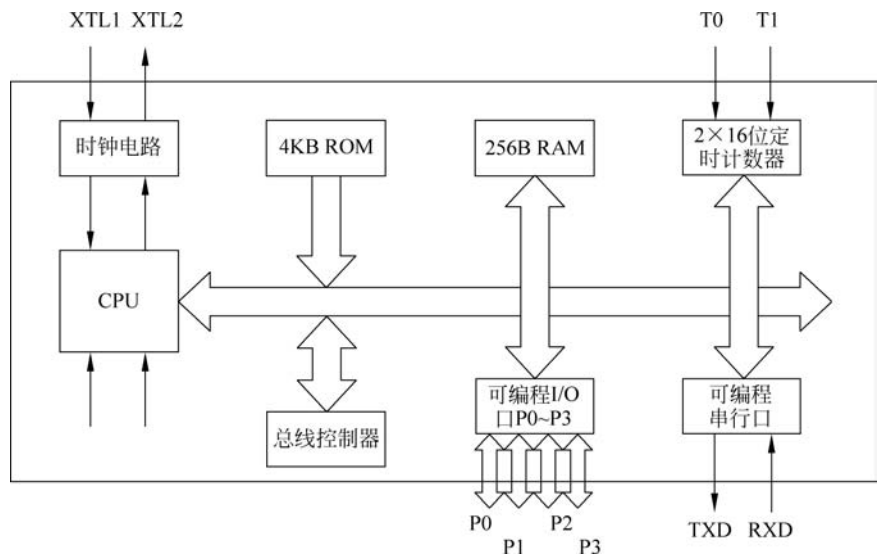


图 2-1 内部结构

运算器包括算术和逻辑运算单元(ALU)、累加器(ACC)、暂存器(B)、程序状态字(PSW)寄存器,实现算术和逻辑运算。

程序计数器(PC): 16 位程序计数器,可寻址空间 64KB。存放待执行指令在程序存储器中的存放地址。

指令寄存器(IR): 存放取指周期取出的当前指令。

指令译码寄存器(ID): 对来自 IR 的指令进行译码,产生与二进制指令代码相应的控制信号。

数据指针寄存器(DPTR): 16 位数据指针寄存器,可寻址外部数据存储区,也可寻址外部程序存储区的数据表格。

程序状态字(PSW): 8 位寄存器,记录当前运算器的状态信息,各位定义见表 2-1。

表 2-1 PSW 定义

D7	D6	D5	D4	D3	D2	D1	D0
CY	AC	F0	RS1	RS0	OV	—	P
进位	半进位	自定义	寄存器组选择		溢出		奇偶

CY: 进位标志。当运算器最高位有进位/借位时,CY=1; 否则,CY=0。

AC: 半进位标志。当运算器 D3 有进位/借位时,AC=1; 否则,AC=0。

F0: 用户自定义标志位。

RS1/RS0: 当前寄存器组选择标志位,见表 2-2,具体内容参见片内数据存储器(RAM)部分。

表 2-2 工作寄存器组定义

RS1	RS0	当前寄存器组
0	0	0 组(字节地址: 00H~07H)
0	1	1 组(字节地址: 08H~0FH)
1	0	2 组(字节地址: 10H~17H)
1	1	3 组(字节地址: 18H~1FH)

P: 奇偶校验位。当累加器中 1 的个数为奇数个时, $P=1$; 否则, $P=0$ 。

2. 存储器

与 80x86 系列微机不同, MCS-51 采用哈佛结构组织存储器, 程序存储器与数据存储器在物理上是两个相互独立的单元, 二者有独立的控制信号、指令和寻址方式。I/O 端口与数据存储器统一编址, 内部寄存器和端口映射在片内数据存储单元, 与内部数据存储器有相同的访问时序和访问指令。

3. 时钟

产生系统工作所需要的时序信号, 在取指周期, 控制系统各单元从程序存储器中取出指令; 在指令执行周期, 控制系统各单元执行指令所指定的操作。

4. 定时/计数器

片内有 2 个 16 位定时/计数器 T/C0 和 T/C1, 用于定时/计数控制。T/C1 可用于产生串行通信所需波特率信号。

5. 可编程 I/O 并行端口

片内提供 P0、P1、P2、P3 4 个 8 位并行 I/O 端口, 每个引脚可独立用于输入/输出, 是系统主要的数据传输和输入/输出控制引脚。

6. 可编程串行端口

片内提供一路 TTL 电平全双工串行通信端口, 实现与其他设备的通信。

7. 中断系统

提供 5 个中断源, 包括 2 个外部硬件中断、2 个定时/计数器中断和 1 个串行通信中断。片内具有中断管理寄存器, 实现中断优先权、屏蔽/允许、中断标志等功能管理。

2.3 封装与引脚

40 脚 DIP 封装及引脚如图 2-2 所示。

1. 电源引脚及晶振

VDD(40): 电源端。

VSS(20): 接地端。

XTAL1(19)/XTAL2(18): 片内振荡器反相放大器输入/输出端, 产生系统时钟信号。

2. 输入/输出

P0 端口(32~39): 8 位并行 I/O 端口, 开漏输出, 可作为 8 位双向 I/O 口使用。在系统

扩展时,分时复用为 8 位外部数据总线 D[7..0]和 16 位外部地址线总线 AB 的低 8 位地址 A[7..0],可用 ALE 信号锁存。

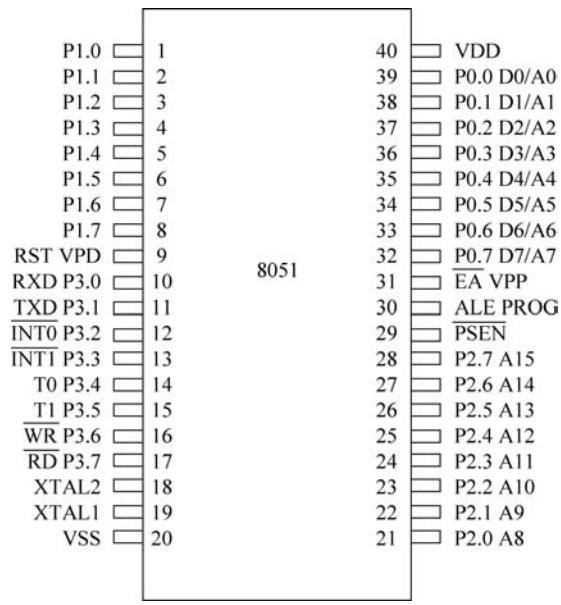


图 2-2 封装及引脚

P1 端口(1~8): 8 位双向 I/O 端口,读引脚时需要先写入 1。

P2 端口(21~28): 8 位并行 I/O 端口,可作为 8 位双向 I/O 端口使用。在系统扩展时,作为 16 位外部地址线总线 AB 的高 8 位地址线 A[15..8]。

P3 端口(32~39): 8 位并行 I/O 端口,可作为 8 位双向 I/O 端口使用。具有第二功能,见表 2-3。

表 2-3 P3 口第二功能

P3 引脚	第二功能
P3.0	串行通信 RXD
P3.1	串行通信 TXD
P3.2	外部中断 $\overline{\text{INT0}}$
P3.3	外部中断 $\overline{\text{INT1}}$
P3.4	定时/计数器 0 计数脉冲输入
P3.5	定时/计数器 1 计数脉冲输入
P3.6	外部数据存储器写信号 $\overline{\text{WR}}$
P3.7	外部数据存储器读信号 $\overline{\text{RD}}$

3. 控制引脚

ALE/PROG(30): 地址锁存允许。访问外部存储器时,ALE 用于锁存 P0 口作为 16 位

地址总线 AB 的低 8 位地址线 A[7..0]。Flash 存储器编程时,用于输入编程脉冲 PROG。

$\overline{\text{PSEN}}$ (29): 外部程序存储器读选通信号。

$\overline{\text{EA}}/\text{VPP}$ (31): 外部程序存储器访问选择端。当 $\overline{\text{EA}}=0$ 时,CPU 从外部 ROM 的 0000H 单元开始执行程序。当 $\overline{\text{EA}}=1$,CPU 从内部 ROM 的 0000H 单元执行程序,当程序长度超过 4KB 时会自动转向外部程序存储器 ROM 的 1000H 单元。

RST(9): 系统复位。

2.4 工作方式

有复位、程序运行、单步、低功耗多种工作方式。

2.4.1 复位

复位引脚 RST 持续 24 个振荡周期的高电平使系统复位,进入复位工作方式。

在复位状态下,程序计数器 PC=0000H,堆栈指针 SP=07H,SFR 内容清 0,P0~P3 输出全 1。复位操作不影响内部数据存储器(RAM)内容。

2.4.2 程序运行

程序运行方式是系统基本工作方式,程序可存放在片内 ROM 和片外 ROM。

当 $\overline{\text{EA}}=0$ 时,CPU 从外部 ROM 的 0000H 单元开始执行程序。当 $\overline{\text{EA}}=1$ 时,CPU 从内部 ROM 的 0000H 单元执行程序,当程序长度超过 4KB 时会自动转向外部 ROM 的 1000H 单元。

复位后程序计数器 PC=0000H,程序总是从程序存储器(ROM)的 0000H 单元开始运行程序。根据系统设计,在 0000H 单元存放有一条无条件转移语句 SJMP,使系统复位后直接跳转到主程序的入口地址,即主程序首地址。

2.4.3 单步

单步工作方式使程序的执行处于外部脉冲的控制下,一个脉冲执行一条指令,用于程序单步调试,可利用外部中断 $\overline{\text{INT0}}$ 实现。

设置 $\overline{\text{INT0}}$ 为低电平中断请求有效,保持为低。将按键操作产生的脉冲加到 $\overline{\text{INT0}}$ 端,按键按下时产生高电平。

在 $\overline{\text{INT0}}$ 中断处理程序中执行如下指令:

```
JNB P3.2, $      ; 若  $\overline{\text{INT0}}=0$ ,循环,等待按键按下
JB P3.2, $       ; 若  $\overline{\text{INT0}}=1$ ,循环,等待按键释放
RETI             ; 返回主程序
```

利用按键产生脉冲,加在 $\overline{\text{INT0}}$ 端,不按为低电平,按下为高电平。

$\overline{\text{INT0}}$ 保持为低,即处于中断请求状态,触发 $\overline{\text{INT0}}$ 中断,执行中断处理程序。在中断处理程序中,执行第一条指令,等待按键按下,显示内部寄存器内容。当按键按下并释放后,返回主程序。

根据中断规定,从中断服务程序返回后,至少要执行一条指令才能重新进入中断。因此,返回主程序后,执行一条指令,显示内部存储器内容,又重新进入中断服务程序,实现程序的单步调试。

2.4.4 低功耗

系统有节电和掉电两种低功耗工作方式,由电源控制寄存器(PCON)控制,其定义见表 2-4。

表 2-4 PCON 寄存器定义

D7	D6	D5	D4	D3	D2	D1	D0
						PD	IDL

PD=1,进入掉电工作方式。

IDL=1,进入节电工作方式。

1. 节电工作方式

执行一条 IDL=1 的指令使系统进入节电模式,CPU 时钟被切断,但时钟信号仍然提供给 RAM、定时器、中断系统和串行口,栈指针(SP)、程序计数器(PC)、程序状态字(PSW)、累加器(ACC)和通用寄存器内容保持。工作电流由正常工作时的 24mA 下降为 3.7mA。

中断请求和硬件复位可使系统退出节电工作方式。

中断方式:任一中断发生,IDL 被硬件复位,节电状态结束。中断返回时将回到进入节电方式时的指令的后一条指令,继续运行程序。

硬件复位:结束节电工作模式,系统进入复位状态。

2. 掉电工作方式

当 PD=1 时,系统进入掉电模式,片内振荡器停止,时钟冻结,所有操作停止,片内 RAM 内容保持,SFR 内容被破坏。掉电方式下,VCC 可降到 2V,工作电流降为 50 μ A。

硬件复位使系统退出掉电模式,进入复位工作模式。

2.5 复位电路

复位引脚 RST 持续 24 个振荡周期的高电平为系统复位信号,当 $f_{\text{osc}}=6\text{MHz}$ 时,复位信号至少应持续 4 μ s 以上才能使系统复位。复位完成后应使 RST 保持低电平,否则系统循环复位。

复位后,07H 写入堆栈指针寄存器(SP),P0~P3 口置 1,允许输入,程序计数器(PC)和

特殊功能寄存器(SFR)清零。复位操作不影响内部 RAM 内容。

当 RST 由高变低,系统由程序存储器(ROM)的 0000H 单元开始执行主程序。

1. 上电复位

上电复位电路如图 2-3 所示。

上电时,电容相当于开路,电源通过电阻对电容充电,充电曲线如图 2-4 所示。

$$u_C = 5(1 - e^{-\frac{t}{RC}})V$$

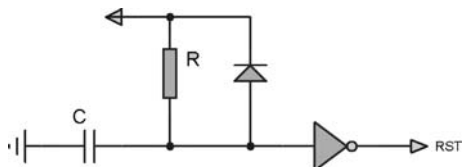


图 2-3 上电复位电路

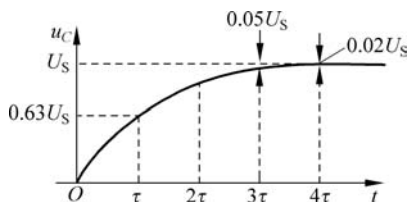


图 2-4 复位电路参数计算

在 $0 < u_C < 0.8V$ 区间,74LS04 输出高电平,使系统复位。根据系统复位要求,需要 24 个振荡周期的高电平为系统复位信号。当 $f_{OSC} = 6MHz$ 时,复位信号至少应持续 $4\mu s$ 以上。根据充电特性曲线,可计算电路参数值。

2. 按键复位

按键复位电路见图 2-5,可根据电容充放电曲线计算按键复位电路参数。

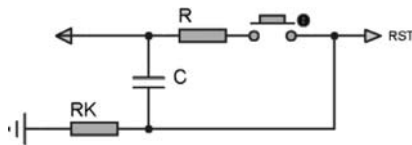


图 2-5 按键复位电路

2.6 时序

控制器在时钟脉冲驱动下,按照一定的时间顺序发出控制信号,控制各组成单元工作,完成某项基本操作,称为 CPU 工作时序。

2.6.1 时钟电路

时钟电路提供系统工作时钟信号,是控制器的基本构成单元。

1. 内部时钟

系统内部时钟见图 2-6。系统内部有一个高增益反相放大器,用于构成振荡器。反相放大器输入端为 XTAL1,输出端为 XTAL2。在 XTAL1 和 XTAL2 两端跨接石英晶体及两个电容构成稳定的自激振荡器。电容 C1 和 C2 通常都取 $30pF$,对振荡频率有微调作用。频率越高,工作速度越快,功耗越大,产生的高次谐波也会对系统内部的模拟电路产生干扰。

2. 外部时钟

可使用外部振荡脉冲信号,外部时钟见图 2-7。由 XTAL2 端引脚输入,直接送至内部时钟电路。XTAL2 逻辑电平与 TTL 电平不兼容,应接一个上拉电阻(5.1kΩ)。

外部时钟方式常用于多片单片机同时工作,便于系统同步。

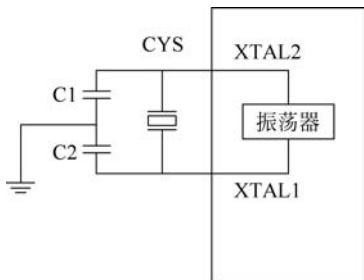


图 2-6 内部时钟

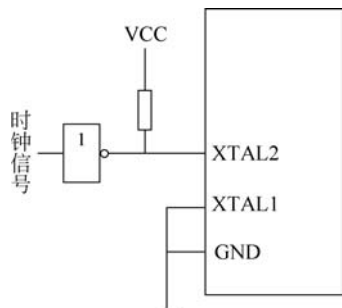


图 2-7 外部时钟

2.6.2 时序周期

- 系统基本时序周期包括以下几种。
- (1) 振荡周期(P): 振荡源的周期,若为内部时钟方式,则为石英晶体的振荡周期。
 - (2) 时钟周期(S): 2 个振荡周期构成 1 个时钟周期,即 $S = P1 + P2$ 。
 - (3) 机器周期: 6 个时钟周期构成 1 个机器周期,如图 2-8 所示。

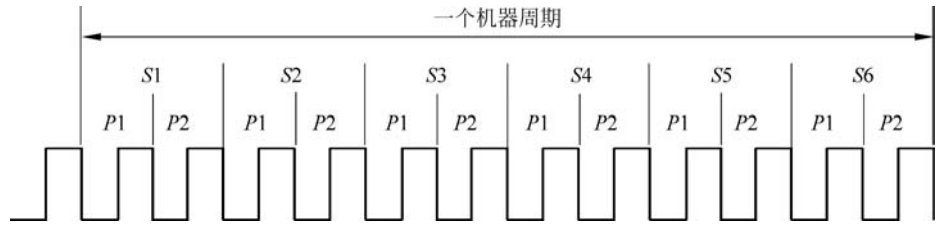


图 2-8 基本时序周期

- (4) 指令周期: 完成一条指令占用的全部时间称为指令周期。MCS-51 的指令周期包含 1~4 个机器周期。
- 若 $f_{osc} = 6\text{MHz}$,则系统周期参数见表 2-5。

表 2-5 周期参数

振荡周期/ μs	时钟周期/ μs	机器周期/ μs	指令周期/ μs
1/6	1/3	2	2~8

2.7 输入/输出端口

共有 4 个 8 位并行双向 I/O 端口,分别定义为 P0、P1、P2 和 P3 口。各端口具有锁存、输出驱动和输入缓冲能力,基本特性见表 2-6。

表 2-6 输入/输出端口基本特性

端口	P0	P1	P2	P3
宽度/b	8	8	8	8
字节地址	80H	90H	A0H	B0H
位地址	80H~87H	90H~97H	A0H~A7H	B0H~B7H
引脚灌电流/mA	10	10	10	10
引脚拉电流/ μ A		50	50	50

P0、P1、P2 和 P3 口映射片内数据存储器的一个存储单元,分别占用地址 80H、90H、A0H 和 B0H,可以位寻址,每个 I/O 引脚都具有位地址。

2.7.1 P0 口

P0 口可作为一般 I/O 端口使用,在外部总线扩展时可作为地址线/数据线复用。

1. 逻辑结构

P0 口位逻辑结构见图 2-9,由输出锁存器、三态输入缓冲器、输出驱动电路及控制电路组成,工作状态受控制信号 C 控制。当 CPU 使控制信号 $C=0$ 时,输出级上拉场效应管 T1 截止,输出级漏极开路,切换开关 MUX 连接输出锁存器,P0 口作一般输入/输出口使用。当 $C=1$ 时,切换开关 MUX 连接反相器 3 输出端,P0 口分时作为地址/数据总线使用。

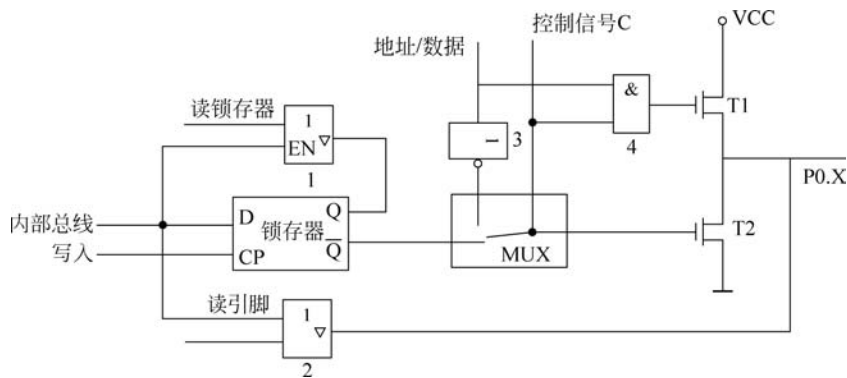


图 2-9 P0 口位逻辑结构

2. 工作过程

P0 口既可作一般 I/O 端口使用,又可作地址/数据总线使用。

P0 口作为一般 I/O 端口输出时,输出级属开漏电路,必须外接上拉电阻。

P0 口作 I/O 端口输入时,须先向锁存器写入 1,使 T2 截止,然后才能读引脚状态,因此称为准双向 I/O 端口。

外部扩展时,P0 口分时复用为外部地址总线 AB 的低 8 位地址线 $A[7..0]$ 和外部数据总线 $DB[7..0]$ 。

2.7.2 P1 口

P1 口可作为一般 I/O 端口使用。

1. 逻辑结构

P1 口位逻辑结构见图 2-10。输出级有内部上拉电阻与电源相连,当 P1 口输出高电平时向外提供拉电流。

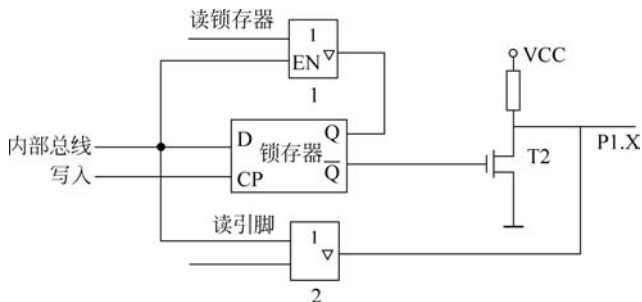


图 2-10 P1 口位逻辑结构

2. 工作过程

内部总线输出高电平时,D 锁存器反向端 $\overline{Q}$ 输出低电平,使 T2 截止,引脚通过上拉电阻输出高电平。内部总线输出低电平时,D 锁存器反向端 $\overline{Q}$ 输出高电平,使 T2 导通,引脚通过 T2 接地,输出低电平。

P1 口作输入时,须先向锁存器写入 1,使 T2 截止,才能读引脚状态,因此称为准双向 I/O 端口。

2.7.3 P2 口

P2 口可作为一般 I/O 端口使用,在外部总线扩展时作为地址线使用。

1. 逻辑结构

P2 口位逻辑结构见图 2-11。输出级由场效应管和上拉电阻组成,可直接驱动拉电流负载,不需要外接上拉电阻。多路开关 MUX 使 P2 口具有通用 I/O 和地址总线输出两种功能。

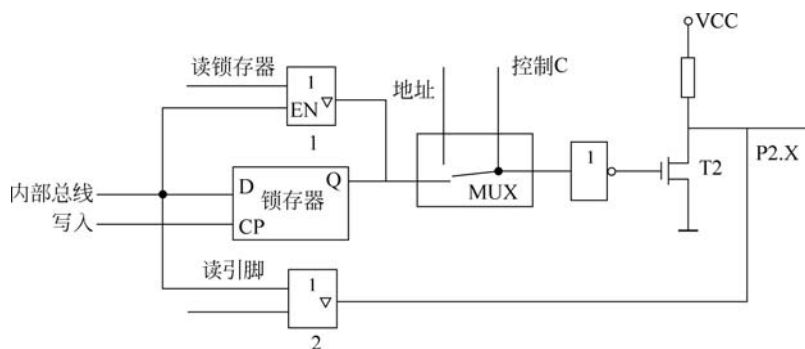


图 2-11 P2 口位逻辑结构

2. 工作过程

当控制信号 $C=0$ 时,多路开关 MUX 连接地址线,程序计数器 PC 的高 8 位地址 PCH 经反相器和 T2 同相出现在引脚,送出高 8 位地址信息 $A8 \sim A15$ 。

当控制信号 $C=1$ 时,多路开关 MUX 连接 D 锁存器,D 锁存器输出 Q 经反相器接输出 T2。当 Q 为高,T2 截止,引脚输出高电平。当 Q 为低,T2 导通,引脚输出低电平。

P2 可作一般 I/O 端口使用,在系统扩展外部存储器或端口时,提供高 8 位地址信号 $A[15..8]$ 。

2.7.4 P3 口

P3 口为多功能端口。

1. 逻辑结构

输出驱动由与非门和场效应管组成。输入电路由一个三态门和一个输入缓冲器组成,逻辑结构见图 2-12。P3 口具有通用 I/O 端口功能和第二功能。

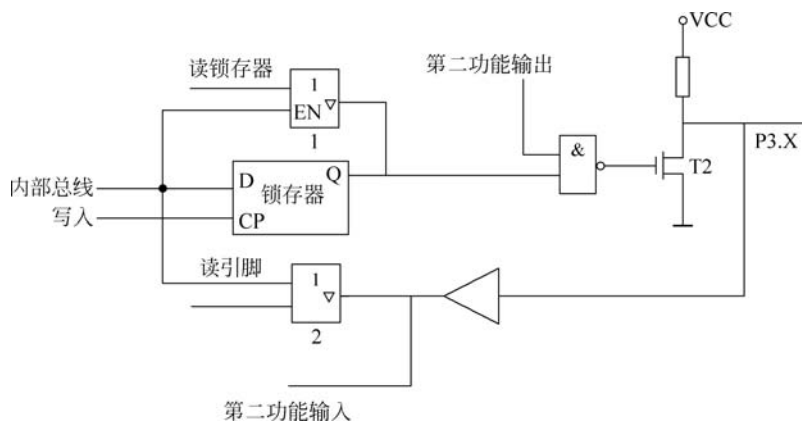


图 2-12 P3 口位逻辑结构

2. 工作过程

当 P3 口作为通用 I/O 端口使用时,第二输出功能端置 1,D 锁存器输出 Q 经与非门和场效应管传输到引脚。当作为第二功能时,D 锁存器输出高电平,第二功能信号经与非门和场效应管送达引脚。

在用作输入时,先向对应的锁存器写入 1,使场效应管截止,外部信号经读引脚缓冲器输入。若是第二功能输入信号,则从第二功能输入引脚输入。

P3 口第二功能见表 2-7。

一般在系统设计时,P3 口不作通用 I/O 端口,以充分利用单片机的内部资源。

表 2-7 P3 口第二功能

引脚	定义	说 明	引脚	定义	说 明
P3.0	RXD	串行数据输入	P3.4	T0	计数器 0 外部输入
P3.1	TXD	串行数据输出	P3.5	T1	计数器 1 外部输入
P3.2	INT0	外部中断 0 输入	P3.6	$\overline{\text{WR}}$	外部数据存储器写信号
P3.3	INT1	外部中断 1 输入	P3.7	$\overline{\text{RD}}$	外部数据存储器读信号

2.8 指令系统

程序由一系列机器指令构成。指令是使 CPU 执行某种操作的命令。对一个具体的 CPU,所有机器指令的集合称为该 CPU 的指令集。指令系统是表征 CPU 性能的核心指标,其格式与功能不仅决定整个计算机系统的硬件结构,而且直接影响系统软件和应用领域。

2.8.1 寻址方式

针对一个 CPU 和其指令系统,规定在机器指令中如何表示以及在指令运行中如何根据指令中给出的地址码寻找操作数的方式,称为寻址方式。MCS-51 规定了 7 种寻址方式。

1. 立即寻址方式

指令中的地址码部分直接给出操作数,取出指令的同时立即得到操作数。立即数作为指令的一部分,存储在代码段。

【应用举例】

```
MOV A, #20H          ; 将立即数 20H 送入寄存器 A
MOV DPTR, #1234H     ; 将立即数 1234H 送给 DPTR 寄存器
```

2. 寄存器寻址方式

操作数保存在某寄存器中,指令地址码给出的是该寄存器的编号。

【应用举例】