第 3 章


微课视频

存 储 结 构

本章以 AT89C51 为模型机，介绍 MCS-51 系列单片机存储结构。

3.1 基本特性

采用哈佛型存储结构，程序存储器和数据存储器在物理上是分开的，具有独立的寻址机构和寻址指令。程序存储器采用 EPROM、EEPROM 或 Flash 存储器，数据存储器采用静态存储器(SRAM)或动态存储器(DRAM)。

内部集成 4KB 程序存储器(ROM)、256B 数据存储器(RAM)，可外部扩展 64KB(最大)程序存储器(ROM)和 64KB(最大)数据存储器(RAM)。

在物理结构上，存储系统可以分为 4 个存储空间，即片内 RAM、片内 ROM 和片外 RAM、片外 ROM，如图 3-1 所示。

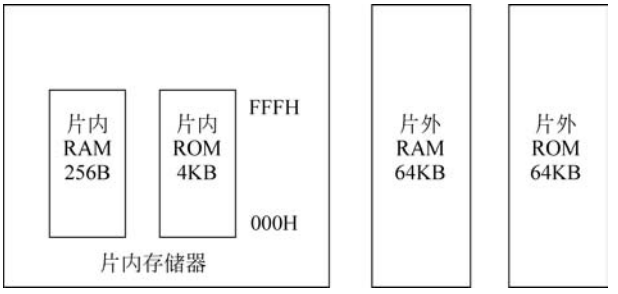

图 3-1 存储系统地址分布

3.2 程序存储器

程序存储器用来存放程序和数据表格，采用 16 位程序计数器(PC)和 16 位地址总线，最大寻址空间为 64KB。

程序存储器物理上可分为片内和片外两部分，共享 64KB 程序存储地址空间。

用引脚 $\overline{EA}$ 确定 CPU 对内部程序存储器和外部程序存储器的使用选择。

当 $\overline{EA}$=0 时，CPU 从外部 ROM 的 0000H 单元开始执行程序。当 $\overline{EA}$=1 时，从内部 ROM 的 0000H 单元开始执行，当地址超过 4KB 时自动转向外部 ROM 的 1000H 单元。

片内程序存储区 0000H～002AH 作为系统保留使用。

0000H～0002H 单元：主程序入口地址。复位时程序计数器 PC=0000H，指向该单元。该单元存放一条绝对跳转指令(SJMP)，跳转到主程序首地址。

0003H～000AH 单元：外部中断 $\overline{INT0}$ 中断服务程序入口地址。该单元存放一条绝对跳转指令(SJMP)，跳转到 $\overline{INT0}$ 中断处理程序首地址。

000BH～0012H 单元：定时/计数器 T/C0 溢出中断处理程序入口地址。该单元存放一条绝对跳转指令(SJMP)，跳转到 T/C0 溢出中断处理程序首地址。

0013H～001AH 单元：外部中断 $\overline{INT1}$ 中断服务程序入口地址。该单元存放一条绝对跳转指令(SJMP)，跳转到 $\overline{INT1}$ 中断处理程序首地址。

001BH～0022H 单元：定时/计数器 T/C1 中断服务程序入口地址。该单元存放一条绝对跳转指令(SJMP)，跳转到 T/C1 溢出中断处理程序首地址。

0023H～002AH 单元：串行口通信中断服务程序入口地址。该单元存放一条绝对跳转指令(SJMP)，跳转到串行通信中断处理程序首地址。

3.3 数据存储器

系统中 RAM 分为内部 RAM 和外部 RAM 两部分，用 MOV 指令访问内部数据存储单元，用 MOVX 指令访问外部数据存储单元。

3.3.1 数据存储器地址分布

片内和片外数据存储器地址分布如图 3-2 所示。

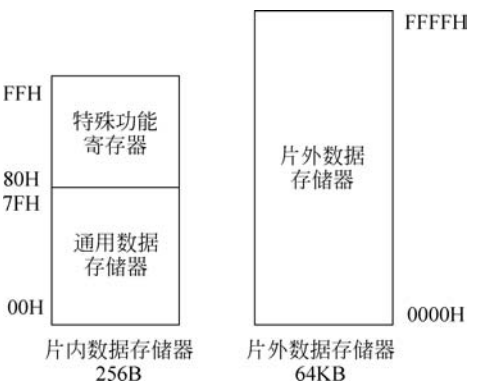

图 3-2 片内和片外数据存储器地址分布

3.3.2 片内 RAM

片内 RAM 按功能划分为工作寄存器区、位寻址区、数据缓冲区(RAM)和特殊功能寄存器(SFR)区四个区域，地址空间分配见图 3-3。

图 3-3 片内 RAM 地址分布

1. 工作寄存器区

地址范围为 00H～1FH，32 字节，分为 4 组，每组 8 个寄存器，标识为 R0～R7。任何时刻只有一组作为当前工作寄存器组使用，由程序状态字 PSW 中的 RS1 和 RS0 选择，见表 3-1。

表 3-1 当前寄存器组选择

RS1	RS0	工作组寄存器(地址)
0	0	0 组(00H～07H)
0	1	1 组(08H～0FH)
1	0	2 组(10H～17H)
1	1	3 组(18H～1FH)

通过软件设置 RS1、RS0 以设置当前寄存器组。单片机复位后，默认 0 组为当前工作寄存器组。

2. 位寻址区

地址范围为 20H～2FH，16 字节，共计 16×8＝128 位。每个单元有一个字节地址，字节地址范围为 20H～2FH。每位有一个位地址，位地址范围为 00H～7FH。位寻址区每一位都可作为一个软件触发器使用，通常把各种状态、位控制变量保存在位寻址区。

3. 数据缓冲区

地址范围为 30H～7FH，用户数据区，共 80 字节单元，用作用户的数据存储区。堆栈也可设置在该区域。

4. 特殊功能寄存器区

地址范围为 80H～FFH，特殊功能寄存器(SFR)区。系统把片内寄存器和 I/O 端口映射在该区域存储单元，对 I/O 端口的操作实际上就是对相应存储单元的操作。

SFR 字节地址见表 3-2。

表 3-2 SFR 字节地址

标识符	名　　称	地址	标识符	名　　称	地址
ACC	累加器	E0H	DPTR	数据指针	83H/82H
B	B 寄存器	F0H	P0	端口 0	80H
PSW	程序状态字	D0H	P1	端口 1	90H
SP	堆栈指针	81H	P2	端口 2	A0H

续表

标识符	名　　称	地址	标识符	名　　称	地址
P3	端口 3	B0H	TL0	定时/计数 0 初值	8AH
IP	中断优先级	B8H	TH1	定时/计数 1 初值	8DH
IE	中断允许	A8H	TL1	定时/计数 1 初值	8BH
TMOD	定时/计数方式	89H	SCON	串口控制寄存器	98H
TCON	定时/计数控制	88H	SBUF	串口数据寄存器	99H
TH0	定时/计数 0 初值	8CH	PCON	电源控制寄存器	97H

3.3.3 特殊功能寄存器

系统特殊功能寄存器映射为内部数据存储器的存储单元，每个特殊功能寄存器都有相应单元地址。

1. 程序计数器(PC)

16 位地址寄存器，存放将要执行的指令地址，寻址范围为 0000H～FFFFH(64KB)。

复位时 PC=0000H，即系统复位后，CPU 从程序存储器 ROM 的 0000H 单元开始执行程序。

2. 数据指针(DPTR)

由 2 个 8 位寄存器构成，高 8 位寄存器 DPH 和低 8 位寄存器 DPL 构成 16 位的寄存器 DPTR，用来存放外部 RAM 的地址，作为 CPU 访问外部 RAM 的数据指针。

CPU 的查表指令使用 DPTR 提供 ROM 中表格的首地址。CPU 访问外部 RAM 中的数据或 ROM 中的表格、常数，必须借助 DPTR 作指针来实现数据的读取访问。

3. 程序状态字(PSW)

8 位寄存器，表征程序执行的状态信息，各位定义如图 3-4 所示。

D7	D6	D5	D4	D3	D2	D1	D0
CY	AC	F0	RS1	RS0	OV	—	P

图 3-4　8 位寄存器定义

CY(PSW.7)：进位标志。在加减法运算中，累加器 A 的最高位 D7 有进位或借位，则 CY=1，否则 CY=0。

AC(PSW.6)：辅助进位位。在加减法运算中，累加器 A 的低 4 位向高位 4 位有进位或借位，则 CY=1，否则 CY=0。

F0(PSW.5)：用户标志位。由用户来定义和使用。

RS1，RS0：当前工作寄存器组选择位。选定当前工作寄存器组。

OV(PSW.2)：溢出标志位。当运算结果溢出时，OV 置 1。

P(PSW.0)：奇偶标志位。累加器 A 中 1 的个数为奇数个时为 1，A 中 1 的个数为偶数时为 0。

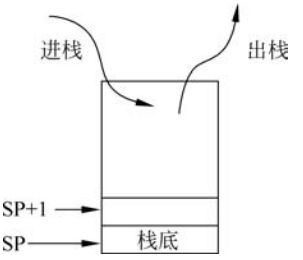

图 3-5　堆栈结构

4. 堆栈指针(SP)

8 位寄存器，指示堆栈栈顶地址。

堆栈是在内存单元专门用来按照先进后出方式存取的区域。在使用堆栈之前，先给 SP 赋值，以规定堆栈的起始位置，称为栈底。在数据压入堆栈后，SP 内容自动加 1，随着数据不断进栈，SP 向上增长，如图 3-5 所示。

系统复位后，SP 总是初始化到内部 RAM 地址 07H。堆栈有专门访问指令 PUSH 和 POP。

3.4　最小系统

由 AT89C51 构成的最小系统见图 3-6，包括时钟电路、上电复位电路和按键复位电路。系统运行点亮一个 LED，R1 为 LED 限流电阻。

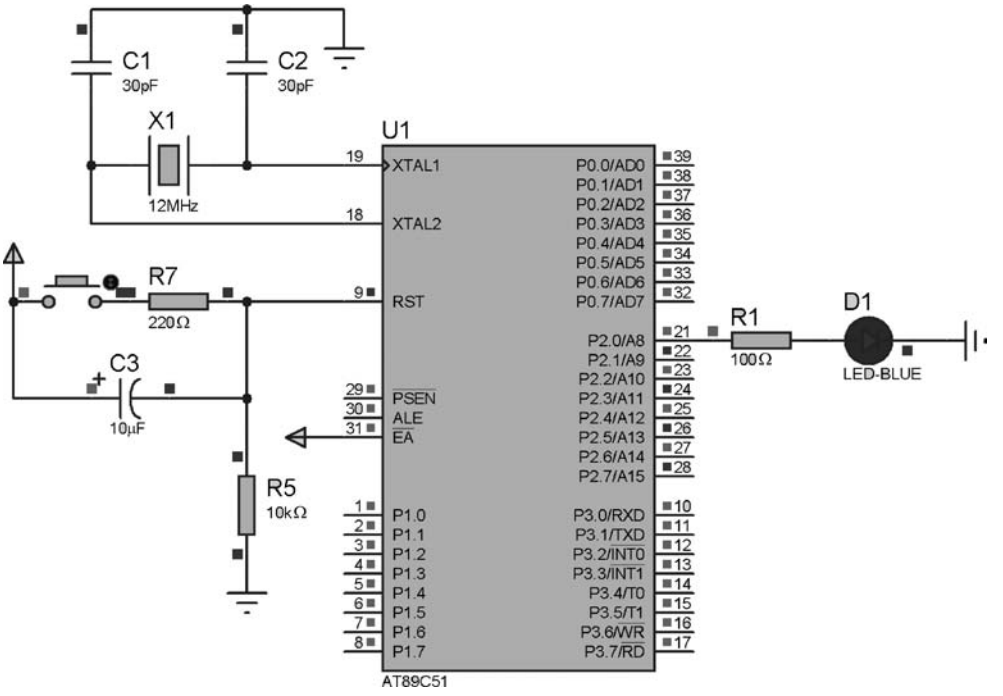

图 3-6　最小系统

注：本书电路图由 Proteus 7.0 绘制，器件符号遵循国际规范。

【应用举例】

```
#include <reg51.h>          //C51 包含文件
sbit LED = P2^0;            //定义 LED 控制位
void main()
```

```
{
    LED = 1;                    //点亮发光二极管
    while(1);
}
```

习题

1. 片内程序存储器 00H～2AH 系统保留单元的作用是什么？
2. 简述哈佛存储结构的特点和意义。
3. 简述工作寄存器组的结构及设置工作寄存器组的意义。
4. 什么是堆栈？其作用是什么？
5. 简述程序计数器(PC)的工作过程。

第二部分　单片机C语言程序设计

用C语言进行嵌入式系统开发已成为业界主流，也是单片机教学的首选。C语言程序设计是“单片机原理与技术”课程的先修课程，用于MCS-51系列单片机开发的C语言称为C51，在数据类型、程序结构、基本语句上，与C语言有非常大的相似度。本部分在先修课程C语言程序设计基础上，介绍C51的数据类型、程序结构和函数设计，突出C51特有的数据类型、存储类型及其对特殊功能寄存器、位寻址的定义和应用。

本部分包括：

第4章　数据类型与基本运算

本章介绍C51的程序结构与数据类型，介绍变量和常量的存储器类型定义及其在单片机程序设计中的作用，讲述C51的基本运算和复合运算。

第5章　程序控制语句

本章介绍条件语句if、开关语句switch的基本语法，以实现程序的有条件跳转与分支。讲述while、do while和for语句设计方法和语法要求，实现循环体结构的程序设计。

第6章　函数

函数是模块化程序设计的核心，也是提高程序设计水平的关键技能。本章介绍C51函数的定义、说明、调用及参数传递。

第 4 章

数据类型与基本运算

微课视频

4.1 C51 程序结构

用于 MCS-51 系列单片机开发的 C 语言称为 C51，在数据类型、程序结构、基本语句、平台使用等方面，C51 与 C 语言有非常大的相似度。

1. C51

由于具有良好的结构性和模块化，用 C 语言书写的程序容易阅读和维护，具有良好的可移植性，功能化的代码能够很方便地从一个工程移植到另一个工程。

相对于汇编语言，用 C 语言编写程序更符合人类的思考习惯，开发者可以更专心考虑算法，而不必十分熟悉处理器的工作过程。

C51 支持所有 51 系列单片机开发，Keil μVision 提供了良好的 C51 开发和调试环境。

2. 程序结构

下面为一个简单而完整的 C51 程序，实现最小系统点亮一个 LED 的功能，接口电路见图 4-1。

```
#include <reg51.h>          //C51 包含文件
sbit LED = P2^0;
void main()                 //主程序
{
    LED = 1;                //点亮发光二极管
    while(1);
}
```

该程序在 KEIL 平台上编辑、编译、连接、运行后，可驱动连接在 P2.0 引脚上的 LED 发光。

C51 程序数据类型、语法、语句、函数定义等类似于一般 C 语言程序。

C51 应用程序有且只有一个主函数 main()，作为程序运行的入口。

{}中间的内容为程序的主体，称为程序体。

C51 用//和/*...*/对程序中的任何部分作注释。