

## 第3章

### CHAPTER 3

# 51单片机及其内部结构

第1章和第2章简要介绍了计算机系统的基本知识。从本章开始,将结合第1章和第2章的内容对51单片机的内部结构进行详细讲解。

## 3.1 单片机简介

单片机是一种集成电路芯片,是采用超大规模集成电路技术把具有数据处理能力的中央处理单元(Central Processing Unit,CPU)、随机存取存储器(Random Access Memory,RAM)、只读存储器(Read-Only Memory,ROM)、多种I/O接口和中断系统、定时器/计时器等功能(可能还包括显示驱动电路、脉宽调制电路、模拟多路转换电路、A/D转换电路等)集成到一块芯片上构成的一个小而功能完善的微型计算机系统,单片机在工业控制领域得到了广泛应用。从20世纪80年代开始,单片机从4位、8位,发展到现在的32位。与常用的PC相比,单片机只缺少了外部设备(如键盘、鼠标等)。它体积小、质量轻、价格低,为学习、应用和开发提供了便利条件。

### 3.1.1 单片机的发展

1971年,Intel公司的霍夫研制成功世界上第一块4位微处理器芯片Intel 4004,标志着第一代微处理器问世,微处理器和微机时代从此开始。

1971年11月,Intel推出MCS-4微型计算机系统(包括4001 ROM芯片、4002 RAM芯片、4003 移位寄存器芯片和4004微处理器),其中4004包含2300个晶体管,尺寸规格为3mm×4mm,计算性能远远超过当年的ENIAC,最初售价为200美元。

1972年,霍夫等开发出第一个8位微处理器Intel 8008。由于8008采用的是P沟道MOS微处理器,因此仍属第一代微处理器。

1973年,霍夫等研制出8位微处理器Intel 8080,以N沟道MOS电路取代了P沟道,第二代微处理器就此诞生。主频2MHz的8080芯片的运算速度比8008快10倍,可访问64KB存储器,使用了基于6μm技术的6000个晶体管,处理速度为0.64MIPS(Million Instructions Per Second)。

1975年,MITS发布第一个通用型Altair 8800,售价为375美元,带有1KB存储器。这是世界上第一台微型计算机。

1976年,Intel公司研制出MCS-48系列8位的单片机,标志着单片机的问世。Zilog公司

也于 1976 年开发了 Z80 微处理器。当时,Zilog、Motorola 和 Intel 在微处理器领域三足鼎立。

20 世纪 80 年代初,Intel 公司在 MCS-48 系列单片机的基础上,推出了 MCS-51 系列 8 位高档单片机。MCS-51 系列单片机在片内 RAM 容量、I/O 接口功能和系统扩展方面都有了很大的提高。

随着工业控制领域要求的提高,开始出现了 16 位单片机,但因为其性价比不理想,并未得到很广泛的应用。20 世纪 90 年代,随着 Intel 的 i960 系列特别是后来的 ARM 系列的广泛应用,32 位单片机迅速取代 16 位单片机的高端地位,并且进入主流市场。在此阶段,传统的 8 位单片机的性能也得到了飞速提高。

此外,单片机系统已经不再只在裸机环境下开发和使用,大量专用的嵌入式操作系统得到广泛应用,部分高端单片机甚至可以直接使用专用的 Windows 和 Linux 操作系统。

3.1.2 单片机的分类

单片机可分为通用型单片机和专用型单片机两大类。通用型单片机是把可开发资源全部提供给使用者的微控制器。专用型单片机则是为过程控制、参数检测、信号处理等方面的特殊需要而设计的单片机。通常所说的单片机即指通用型单片机。

1. 51 单片机的分类

51 单片机源于 Intel 公司的 MCS-51 系列。在 Intel 公司将 MCS-51 系列单片机的技术开源后,许多公司,如 Philips、Dallas、Siemens、OKI、Atmel、华邦、LG 等都以 8051 为基核推出了许多各具特色、具有优异性能的单片机。这些厂家以 8051 为基核推出的各种型号的兼容型单片机统称为 51 系列单片机,8051 是其中最基础的单片机型号,该系列还在不断发展和完善。随着各种新型号系列产品的推出,51 单片机越来越被广大用户所接受。

MCS-51 系列单片机有超过 20 种芯片,从芯片内部资源的角度,MCS-51 系列单片机分为多个种类,如表 3.1 所示。

表 3.1 MCS-51 系列单片机分类

| 型 号     | 程序存储器/KB | 数据存储器/B | 寻址范围 (RAM) /KB | 寻址范围 (ROM) /KB | 并行接口 | 串行接口 | 中断源 | 定时器计数器 | 晶振 /MHz |
|---------|----------|---------|----------------|----------------|------|------|-----|--------|---------|
| 8051AH  | 4        | 128     | 64             | 64             | 4×8  | UART | 5   | 2×16   | 2~12    |
| 8751H   | 4        | 128     | 64             | 64             | 4×8  | UART | 5   | 2×16   | 2~12    |
| 8031AH  | —        | 128     | 64             | 64             | 4×8  | UART | 5   | 2×16   | 2~12    |
| 8052AH  | 8        | 256     | 64             | 64             | 4×8  | UART | 6   | 3×16   | 2~12    |
| 8752H   | 8        | 256     | 64             | 64             | 4×8  | UART | 6   | 3×16   | 2~12    |
| 8032AH  | —        | 256     | 64             | 64             | 4×8  | UART | 6   | 3×16   | 2~12    |
| 80C51BH | 4        | 128     | 64             | 64             | 4×8  | UART | 5   | 2×16   | 2~12    |
| 87C51H  | 4        | 128     | 64             | 64             | 4×8  | UART | 5   | 2×16   | 2~12    |
| 80C31BH | —        | 128     | 64             | 64             | 4×8  | UART | 5   | 2×16   | 2~12    |
| 83C451  | 4        | 128     | 64             | 64             | 7×8  | UART | 5   | 2×16   | 2~12    |
| 87C451  | 4        | 128     | 64             | 64             | 7×8  | UART | 5   | 2×16   | 2~12    |
| 80C451  | —        | 128     | 64             | 64             | 7×8  | UART | 5   | 2×16   | 2~12    |
| 83C51GA | 4        | 128     | 64             | 64             | 4×8  | UART | 7   | 2×16   | 2~12    |
| 87C51GA | 4        | 128     | 64             | 64             | 4×8  | UART | 7   | 2×16   | 2~12    |
| 80C51GA | —        | 128     | 64             | 64             | 4×8  | UART | 7   | 2×16   | 2~12    |

续表

| 型 号        | 程序存储器/KB | 数据存储器/B | 寻址范围<br>(RAM)<br>/KB | 寻址范围<br>(ROM)<br>/KB | 并行<br>接口 | 串行<br>接口 | 中断源 | 定时器<br>计数器 | 晶振<br>/MHz |
|------------|----------|---------|----------------------|----------------------|----------|----------|-----|------------|------------|
| 83C152     | 8        | 256     | 64                   | 64                   | 5×8      | GSC      | 6   | 2×16       | 2~17       |
| 80C152     | —        | 256     | 64                   | 64                   | 5×8      | GSC      | 11  | 2×16       | 2~17       |
| 83C251     | 8        | 256     | 64                   | 64                   | 4×8      | UART     | 7   | 3×16       | 2~12       |
| 87C251     | 8        | 256     | 64                   | 64                   | 4×8      | UART     | 7   | 3×16       | 2~2        |
| 80C251     | —        | 256     | 64                   | 64                   | 4×8      | UART     | 7   | 3×16       | 2~12       |
| 80C52      | 8        | 256     | 64                   | 64                   | 4×8      | UART     | 6   | 3×16       | 2~12       |
| 8052AH     | 8        | 256     | 64                   | 64                   | 4×8      | UART     | 6   | 3×16       | 2~12       |
| AT89C51    | 4        | 128     | 64                   | 64                   | 32       | UART     | 5   | 2×16       | 0~24       |
| AT89C52    | 8        | 256     | 64                   | 64                   | 32       | UART     | 6   | 3×16       | 0~24       |
| AT89LV51   | 4        | 128     | 64                   | 64                   | 32       | UART     | 5   | 2×16       | 0~24       |
| AT89LV52   | 8        | 256     | 64                   | 64                   | 32       | UART     | 6   | 3×16       | 0~24       |
| AT89C1051  | 1        | 64      | 4                    | 4                    | 15       | —        | 3   | 1×16       | 0~24       |
| AT89C1051U | 1        | 64      | 4                    | 4                    | 15       | UART     | 5   | 2×16       | 0~24       |
| AT89C2051  | 2        | 128     | 4                    | 4                    | 15       | UART     | 5   | 2×16       | 0~24       |
| AT89C4051  | 4        | 128     | 4                    | 4                    | 15       | UART     | 5   | 2×16       | 0~24       |
| AT89C55    | 20       | 256     | 64                   | 64                   | 32       | UART     | 6   | 3×16       | 0~33       |
| AT89S53    | 12       | 256     | 64                   | 64                   | 32       | UART     | 7   | 3×16       | 0~33       |
| AT89S8252  | 8        | 256     | 64                   | 64                   | 32       | UART     | 7   | 3×16       | 0~33       |
| AT88SC54C  | 8        | 128     | 64                   | 64                   | 32       | UART     | 5   | 2×16       | 0~24       |

注：UART—通用异步接收发送器；R/E—MaskROM/EPROM；GSC—全局串行通道。

表 3.1 列出了 MCS-51 单片机的芯片型号以及它们的性能指标,下面在表 3.1 的基础上对 MCS-51 单片机作进一步的说明。

#### 1) 按片内不同程序存储器的配置来划分

MCS-51 单片机按片内不同程序存储器的配置来分,可以分为 3 种类型。

(1) 片内带掩模 ROM 型：8051、80C51、8052、80C52。此类芯片是由半导体厂商在芯片生产过程中,将用户的应用程序代码通过掩模工艺制作到 ROM 中。其应用程序只能委托半导体厂商“写入”,一旦写入后不能修改。此类单片机适合大批量使用。

(2) 片内带 EPROM 型：8751、87C51、8752。此类芯片带有透明窗口,可通过紫外线擦除存储器中的程序代码,应用程序可通过专门的编程器写入单片机中,需要更改时可擦除重新写入。此类单片机价格较高,不适合大批量使用。

(3) 片内无 ROM(ROMLess)型：8031、80C31、8032。由于此类芯片的片内没有程序存储器,使用时必须在外部并行扩展程序存储器芯片,因此此类单片机系统电路复杂,目前较少使用。

#### 2) 按片内不同容量的存储器配置来划分

按片内不同容量的存储器配置来分,可以分为两种类型。

(1) 51 子系列型：51 子系列是 MCS-51 系列基本型产品。片内带有 4KB ROM/EPROM(8031、80C31 除外)、128BRAM、2 个 16 位定时器/计数器、5 个中断源等。

(2) 52 子系列型：52 子系列是 MCS-51 系列增强型产品。片内带有 8KB ROM/

EPROM(8032、80C32 除外)、256BRAM、3 个 16 位定时器/计数器、6 个中断源等。

3) 按芯片在半导体制造工艺上的不同来划分

按芯片在半导体制造工艺上的不同来分,可以分为两种类型。

(1) HMOS 工艺型: 8051、8751、8052、8032。HMOS 工艺,即高密度短沟道 MOS 工艺。

(2) CHMOS 工艺型: 80C51、83C51、87C51、80C31、80C32、80C52。此类芯片型号中都包括字母 C。

此两类器件在功能上完全兼容,但采用 CHMOS 工艺的芯片具有低功耗的特点,它所消耗的电流要比 HMOS 器件小得多。CHMOS 器件比 HMOS 器件多了两种节电的工作方式(掉电方式和待机方式),常用于构成低功耗的应用系统。

## 2. 51 单片机生产厂商

目前,生产 51 单片机的厂商和相应产品如下。

1) ATMEL 公司

在众多的 51 单片机中,ATMEL 公司的 AT89C51、AT89S51 应用最广泛。ATMEL 公司的单片机芯片不但和 8051 指令、引脚完全兼容,且片内带 4KB Flash 存储器,其中的内容可以用电的方式瞬间擦除、改写,同时,写入单片机内的程序可以进行加密。同时,AT89C51、AT89S51 售价比 8031 低,市场供应充足。

2) Philips 公司

Philips 公司推出的含存储器的 80C51 系列和 80C52 系列单片机,此产品都为 CMOS 型工艺的单片机。Philips 公司推出的 51 单片机与 MCS-51 系列单片机兼容,但增加了程序存储器 Flash ROM、可编程计数器阵列 PCA、I/O 接口的高速输入/输出、串行扩展总线 I<sup>2</sup>C 总线、ADC、PWM、I/O 接口驱动器、程序监视定时器(Watch Dog Timer, WDT)等功能的扩展。

3) 华邦公司

华邦公司推出的 W78C××和 W78E××系列单片机,此产品与 MCS-51 系列单片机兼容,每个指令周期只需要 4 个时钟周期,工作频率最高可达 40MHz。同时增加了程序监视定时器、6 组外部中断源、2 组 UART 等资源。

4) Dallas 公司

Dallas 公司推出的 Dallas-HSM 系列单片机,主要包括 DS80C×××、DS83C×××和 DS87C×××等。此产品除了与 MCS-51 系列单片机兼容外,还具有高速结构(1 个机器周期只有时钟周期,工作频率范围为 0~33MHz)、更大容量的内部存储器(内部 ROM 有 16KB)、两个 UART、13 个中断源、程序监视定时器等功能。

5) LG 公司

LG 公司推出的 GMS90C××、GMS97C××和 GMS90L××、GMS97L××系列单片机。与 MCS-51 系列单片机兼容。

6) 凌阳公司

凌阳公司推出的单片机具有高速度、低价、可靠、实用、体积小、功耗低和简单易学等特点。SPCE061A 型单片机内嵌 32KB Flash,处理速度高,适用于数字语音播报和识别等应用领域。SPMC75 系列单片机是凌阳公司开发的具自主知识产权的 16 位单片机,集成了能产生变频电动机驱动的 PWM 发生器、多功能捕获比较模块、BLDC 电动机驱动专用位置侦

测接口、两相增量编码器接口等硬件模块；以及多功能 I/O 接口、同步和异步串行接口、ADC、定时计数器等功能模块，在这些硬件模块的支持下，SPMC75 可以完成诸如家电用变频驱动器、标准工业变频驱动器、多环伺服驱动系统等复杂应用。

以上 ATME1、Philips、华邦、Dallas、凌阳、LG 等公司生产的系列单片机与 Intel 公司的 MCS-51 系列单片机具有良好的兼容性，包括指令兼容、总线兼容和引脚兼容。但各个厂家开发了许多功能不同、类型不一的单片机，给用户提供了广泛的选择空间，其良好的兼容性保证了选择的灵活性。

### 3.1.3 单片机应用等级

单片机用途广，使用环境差别大，如何保证单片机控制系统或装置的可靠性是设计者和使用者最为关注的问题。根据单片机的运行温度范围，单片机的应用大致划分为 3 级。

(1) 军用级：运行温度范围为  $-50\sim+125^{\circ}\text{C}$ ，适用于军用品要求苛刻的应用环境，芯片的价格比较昂贵。例如，Intel 公司的 MCS-51 系列单片机 MD80C51FB。该型号以 MD 标识开头，M 代表军品，D 代表直插封装。

(2) 工业级：早期的单片机产品大多为工业级，运行温度范围为  $-45\sim+85^{\circ}\text{C}$ ，介于商业级和军用级之间，适宜在工业生产环境下使用。其特点是可靠性远高于商业级，但价格远低于军用级。MCS-51 系列单片机的普通产品均属于工业级。

(3) 商业级：运行温度范围为  $0\sim+70^{\circ}\text{C}$ ，主要限于机房、办公及住宅环境，适用于民用产品，例如，家电、玩具等。商业级产品价格低廉，品种齐全，应用最为广泛。

### 3.1.4 单片机应用领域

#### 1. 在工业控制中的应用

工业控制是最早采用单片机控制的领域之一，在测控系统、过程控制、机电一体化设备中主要利用单片机实现逻辑控制、数据采集、运算处理、数据通信等用途。单独使用单片机可以实现一些小规模的控制功能，作为底层检测、控制单元可以与上位计算机一起构成大规模工业控制系统。特别是在机电一体化技术中，单片机的结构特点使其更容易发挥其集机械、微电子和计算机技术优势于一体的优势。

#### 2. 在智能仪器中的应用

内部含有单片机的仪器统称为智能仪器，也称为微机化仪器。这类仪器大多采用单片机进行信息处理、控制及通信，与非智能化仪器相比，功能得到了强化，增加了诸如数据存储、故障诊断、联网集控等功能。以单片机为核心的智能仪器已经是自动化仪表发展的一种趋势。

#### 3. 在家用电器中的应用

单片机功能完善、体积小、价格低廉、易于嵌入，非常适用于对家用电器的控制。嵌入单片机的家用电器实现了智能化，是传统家用电器的更新换代，现已广泛应用于洗衣机、空调、电视机、视盘机、微波炉、电冰箱、电饭煲以及各种视听设备中等。

#### 4. 在信息和通信产品中的应用

信息和通信产品的自动化和智能化程度很高，其中许多功能的完成都离不开单片机的参与。这里最具代表性和应用最广的产品就是移动通信设备，例如，手机内的控制芯片就属于专用型单片机。另外，在计算机外部设备，如键盘、打印机中也离不开单片机。新型单片

机普遍具备通信接口,可以方便地与计算机进行数据通信,为计算机和网络设备之间提供连接服务创造了条件。

### 5. 在办公自动化设备中的应用

现代办公自动化设备中大多数嵌入了单片机作为控制核心,如打印机、复印机、传真机、绘图机、考勤机及电话等。通过单片机控制不但可以实现设备的基本功能,还可以完成与计算机之间的数据通信。

### 6. 在医用设备领域中的应用

单片机在医疗设施及医用设备中的用途也相当广泛,例如,在医用呼吸机、各种分析仪、医疗监护仪、超声诊断设备及病床呼叫系统中都得到了实际应用。

### 7. 在汽车电子产品中的应用

现代汽车的集中显示系统、动力监测控制系统、自动驾驶系统、通信系统和运行监视器等装置都离不开单片机。特别是在采用现场总线的汽车控制系统中,以单片机担当核心的节点通过协调、高效的数据传送不仅实现了复杂的控制功能,而且简化了系统结构。

## 3.2 51 单片机的内部结构

MCS-51 单片机在一块芯片中集成了 CPU、RAM、ROM、定时器/计数器和多种功能的 I/O 接口等,包含下列几个部件:

- (1) 一个 8 位 CPU;
- (2) 一个片内振荡器及时钟电路;
- (3) 4KB ROM 程序存储器;
- (4) 128B RAM 数据存储器;
- (5) 2 个 16 位定时器/计数器;
- (6) 可寻址 64KB 外部数据存储器器和 64KB 外部程序存储器空间的控制电路;
- (7) 32 根可编程的 I/O 线(4 个 8 位并行 I/O 接口);
- (8) 一个可编程全双工串行接口;
- (9) 5 个中断源、两个优先级嵌套中断结构。

8051 单片机内部框图如图 3.1 所示。各功能部件由内部总线连接在一起。在图 3.1 中,4KB 的 ROM 部分用 EPROM 替换就成为 8751;图 3.1 中去掉 ROM 部分就成为 8031。

CPU 是单片机的核心部件。它是一个 8 位数据宽度的处理器,负责控制、指挥、调度和协调整个单元系统的工作。CPU 由运算器、控制器和寄存器等部件组成。

(1) 运算器:运算器的主要功能是算术运算和逻辑运算。例如,能完成加、减、乘、除、加 1、减 1、BCD 码十进制调整、比较等算术运算以及与、或、异或、求补、循环等逻辑操作,操作结果的状态信息送至程序状态字(Program Status Word,PSW)寄存器中。

8051 运算器还包含有一个布尔处理器,可执行置位、复位、取反等位操作,也能对进位标志位与其他可位寻址的位进行逻辑与、或操作。运算器虽然是 CPU 中的一个核心部件,但是用户编程时无法直接访问。

(2) 程序计数器(Program Counter,PC):PC 用来存放即将要执行的指令地址,共 16 位,它计数(寻址)的范围是 0000H~FFFFH(64KB),可对 64KB 程序存储器直接寻址。执

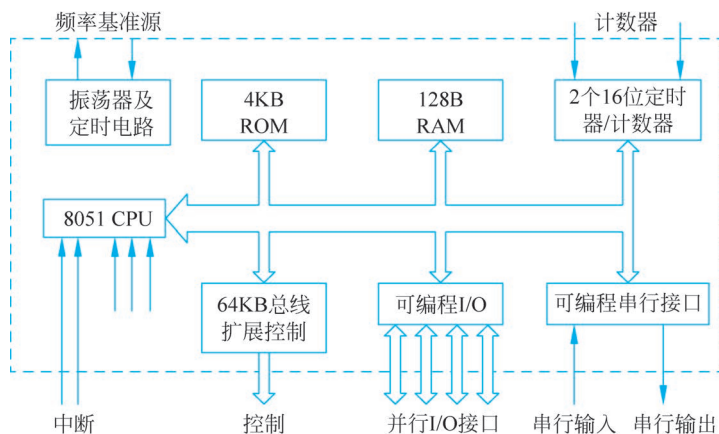


图 3.1 8051 单片机内部框图

行指令时,PC 内容的低 8 位经 P0 口输出,高 8 位经 P2 口输出。由于 PC 中存放的是即将执行的指令的地址,因此一般情况下不允许用户编程修改它,否则会导致程序运行紊乱。

(3) 指令寄存器: 指令寄存器中存放 CPU 从 ROM 中取出的即将执行的指令机器码。CPU 执行指令时,由程序存储器中读取的指令机器码送入指令寄存器,经译码后发出相应的控制信号,完成指令功能。

(4) 累加器(Accumulator, ACC): ACC 为 8 位寄存器(简称 A 累加器),是 CPU 中使用最频繁的寄存器。CPU 要执行算术运算和逻辑运算,就必须有累加器 ACC 的参与,并且执行结果也将暂时存放在 ACC 中。

(5) B 寄存器: B 寄存器通常与 A 累加器配合使用,用于存放第二操作数。在乘、除运算中,运算结束后,存放乘法的乘积高 8 位或除法的余数部分。若不进行乘除操作,则 B 寄存器可作为通用寄存器。

单片机芯片中除 CPU 外的其他部件,将在后续章节中详细介绍。

## 3.3 存储器

### 3.3.1 51 单片机的存储器组织

51 单片机的存储器分为 ROM 和 RAM 两种,其中,ROM 用来存放程序、表格和一些常数;RAM 用来存放程序运行过程中所需要的数据(变量)或临时存放运算的结果。在介绍 51 单片机的存储器结构前,首先介绍计算机系统存储空间的结构形式。

(1) 统一编址: ROM 和 RAM 共用一个存储器逻辑空间,实行统一编址。比如对 8086 计算机系统而言,总共有 20 根地址线,那么 8086 的 CPU 总共能访问  $2^{20}$ B(1MB)的地址空间,此时,ROM 和 RAM 一起共同占用 1MB。如果 RAM 占用 64KB 的地址空间,那么 ROM 只能占用剩下的 1MB 减去 64KB 的地址空间,如图 3.2(a)所示。

(2) 分开编址: ROM 和 RAM 分为两个独立存储器逻辑空间,分开编址。比如对 51 单片机系统而言,总共有 16 根地址线,那么 51 单片机 CPU 总共能访问  $2^{16}$ B(64KB)的地址空间,此时,ROM 和 RAM 分别拥有 64KB,如图 3.2(b)所示。



视频讲解

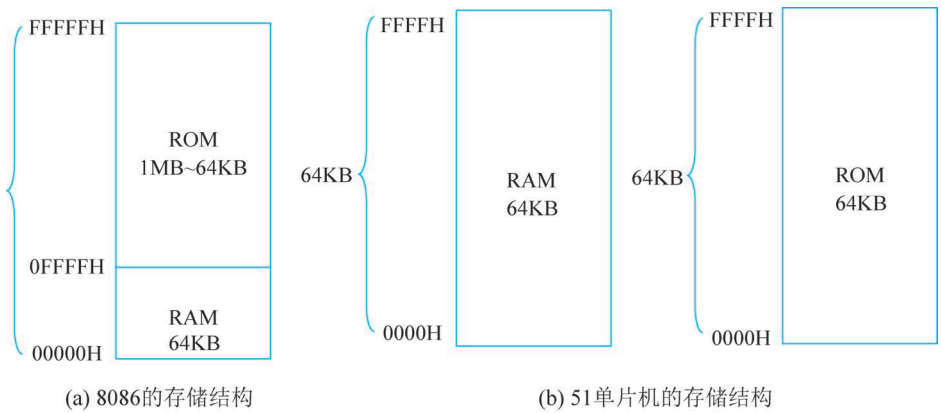


图 3.2 统一编址和分开编址

51 单片机内部有 4KB 的 ROM 和 128B 的 RAM,当片内的 ROM 和 RAM 不能满足使用需求时,可以购买 ROM 芯片和 RAM 芯片进行扩展。因此,从物理角度来看,51 单片机有 4 个存储空间,即片内 RAM、片外 RAM、片内 ROM 和片外 ROM。

当 CPU 执行程序时,芯片上的 EA 引脚的电平决定了片内 ROM 和片外 ROM 的选择。

(1) EA=1: CPU 首先自动访问内部 ROM(4KB: 0000H~0FFFH),当程序的存储范围超过了 4KB 时,CPU 自动继续访问片外 ROM。此时,片外 ROM 的地址空间范围为 4~64KB(60KB: 1000H~FFFFH)。

(2) EA=0: 此时内部 ROM 被忽略,CPU 总是从外部 ROM 访问,外部 ROM 空间的范围是 0~64KB,其地址范围为 0000H~FFFFH。

由此可见,尽管从物理的角度来看 51 单片机存在片内 ROM 和片外 ROM,但是 CPU 总共能访问的 ROM 只有 64KB,因此从逻辑的角度来看,51 单片机有 3 个存储空间,即片内 RAM、片外 RAM 和片内片外统一的 ROM,如图 3.3 所示。

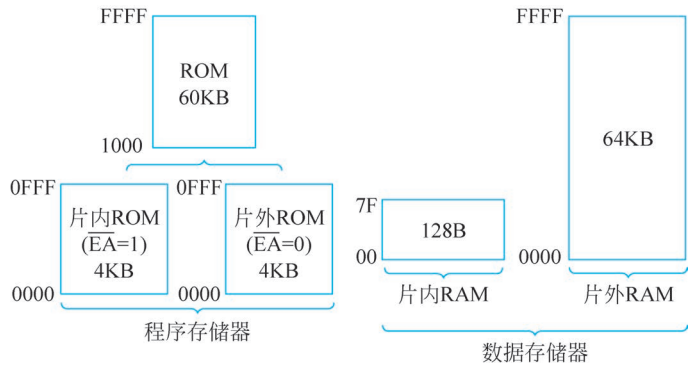


图 3.3 51 单片机逻辑上的 3 个存储空间

3.3.2 程序存储器

在 51 单片机中,程序存储器被用来存放程序、常数或表格等。8051 系列单片机片内有 4KB 的 ROM 存储单元,地址为 0000H~0FFFH。8751 有 4KB 的 EPROM,而 8052 和 8752 则有 8KB 的片内存储器。8031 和 8032 无片内程序存储器,因此 CPU 只能从片外

ROM 开始访问,故 8031 和 8032 芯片的 EA 引脚必须接地。

在 64KB 的程序存储器中,0000H~0002AH(地址向量区)区域具有特殊用途,是保留给系统使用的。0000H 是单片机的入口地址(启动地址),51 单片机复位后程序计数器 PC 的内容为 0000H,故系统必须从 0000H 单元开始读取第一条指令,以执行程序(类似 C 语言中的 main()函数)。

除 0000H 单元外,其他的 0003H、000BH、0013H、001BH 和 0023H 单元分别对应 5 种中断源的中断子程序的入口地址。其含义是:

- (1) 0003H~000AH,为外部中断 0( $\overline{\text{INT0}}$ )的中断地址区;
- (2) 000BH~0012H,为定时器/计数器 0(T0)的中断地址区;
- (3) 0013H~001AH,为外部中断 1( $\overline{\text{INT1}}$ )的中断地址区;
- (4) 001BH~0022H,为定时器/计数器 1(T1)的中断地址区;
- (5) 0023H~002AH,为串行接口(TI,RI)中断地址区。

CPU 在系统中断响应之后,程序将自动转到各中断入口地址处执行,由于各个中断的地址区域有限,因此当中断服务函数超出这个区域长度时,需要在中断入口地址处往往存放一条无条件转移指令进行跳转,以便执行中断服务程序。

### 3.3.3 外部数据存储器

在 51 单片机中,其扩展的外部 RAM 和扩展的 I/O 接口在地址空间 0000H~FFFFH 是重叠的。也就是说,所有扩展的外部 RAM 和扩展的 I/O 接口共同占用 64KB 的地址单元。

在 8051 单片机中采用 MOV 和 MOVX(Move External)两种指令来区分片内外 RAM 空间,其中,片内 RAM 使用 MOV 指令,片外 RAM 和 I/O 接口使用 MOVX 指令。当访问片内和片外统一编址的 ROM 时,使用 MOVC(Move Code byte)指令。

### 3.3.4 内部数据存储器

51 单片机的片内 RAM 的地址范围是 00H~FFH,其中,80H~FFH 属于特殊功能寄存器(Special Function Register,SFR)。在普通型 51 单片机中,只有低 128B 的 RAM,地址范围为 00H~7FH;80H~FFH 是特殊功能寄存器。在增强型 52 单片机中,共有 256B 的片内 RAM,地址范围是 00H~FFH;高 128B(80H~FFH)与特殊功能寄存器共用。CPU 通过不同的寻址方式以区分高 128B 属于 RAM 或者 SFR。访问高 128B 的 RAM 采用寄存器间接寻址方式,而访问 SFR 只能采用直接寻址方式。访问低 128B 的 RAM 空间时,两种寻址方式均可采用。内部 RAM 和 SFR 如图 3.4 所示。



图 3.4 内部 RAM 和 SFR

51 单片机片内 RAM 的 00H~7FH 地址范围的使用情况如图 3.5 所示。

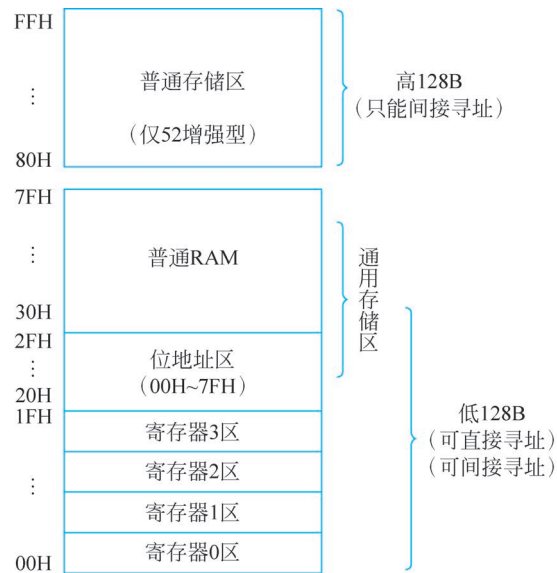


图 3.5 00H~7FH 地址范围的使用情况

(1) 00H~1FH: 前 32B 为通用寄存器区。共分为 4 组通用寄存器(0~3 组),每组有 R0~R7 共 8 个寄存器。这 4 组通用寄存器在某一时刻只能使用其中的一组,其他没有使用的寄存器组作为普通 RAM 使用。由程序状态字(PSW)寄存器中的 RS1、RS0 两位决定具体使用哪一组作为通用寄存器,如表 3.2 所示。注意,在 PSW 寄存器中,RS1 是高位,RS0 是低位。在 CPU 初始化时,默认选择第 0 组作为通用寄存器。

表 3.2 当前工作组寄存器的选择

| PSW. 4 RS1 | PSW. 3 RS0 | 当前使用的工作寄存器组 R0~R7 |
|------------|------------|-------------------|
| 0          | 0          | 组 0(00H~07H)      |
| 0          | 1          | 组 1(08H~0FH)      |
| 1          | 0          | 组 2(10H~17H)      |
| 1          | 1          | 组 3(18H~1FH)      |

寄存器与 SRAM 的地址对应关系如表 3.3 所示。

表 3.3 R0~R7 寄存器与 SRAM 的地址对应关系

| 寄存器名称 | 地 址 |     |     |     |
|-------|-----|-----|-----|-----|
|       | 组 0 | 组 1 | 组 2 | 组 3 |
| R0    | 00H | 08H | 10H | 18H |
| R1    | 01H | 09H | 11H | 19H |
| R2    | 02H | 0AH | 12H | 1AH |
| R3    | 03H | 0BH | 13H | 1BH |
| R4    | 04H | 0CH | 14H | 1CH |
| R5    | 05H | 0DH | 15H | 1DH |
| R6    | 06H | 0EH | 16H | 1EH |
| R7    | 07H | 0FH | 17H | 1FH |

系统在复位后,堆栈指针 SP 自动赋为 07H,堆栈操作的数据将从 08H 开始存放。这样默认情况下寄存器组 1~组 3 就无法使用。为了解决这一问题,在程序初始化时应该先设置 SP 的值。例如,将 SP 置成 70H,以后堆栈的数据就从 71H 存放。

(2) 20H~2FH: 这 16 字节单元既可以字节寻址,也可以位寻址,每个单元有 8 位,总共 128 位。每个位拥有一个位地址,如表 3.4 所示。在表 3.4 中,以最后一行为例,字节地址为 20H 的单元总共有 8 位,其位地址从 D7~D0 分别为 07H、06H、05H、04H、03H、02H、01H、00H。

在表 3.4 中,存在一个字节地址 20H 和一个位地址 20H。CPU 如何判断 20H 是属于字节地址还是位地址呢? 这需要通过指令的另外一个操作数来判断。比如,

```
MOV C, 20H;
```

C 是一个位,故 20H 是位地址。

```
MOV A, 20H;
```

A 是一个 8 位的寄存器,故 20H 是字节地址。

表 3.4 SRAM 中的位寻址区地址分配

| SRAM 字节地址 | D7  | D6  | D5  | D4  | D3  | D2  | D1  | D0  |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|
| 2FH       | 7FH | 7EH | 7DH | 7CH | 7BH | 7AH | 79H | 78H |
| 2EH       | 77H | 76H | 75H | 74H | 73H | 72H | 71H | 70H |
| 2DH       | 6FH | 6EH | 6DH | 6CH | 6BH | 6AH | 69H | 68H |
| 2CH       | 67H | 66H | 65H | 64H | 63H | 62H | 61H | 60H |
| 2BH       | 5FH | 5EH | 5DH | 5CH | 5BH | 5AH | 59H | 58H |
| 2AH       | 57H | 56H | 55H | 54H | 53H | 52H | 51H | 50H |
| 29H       | 4FH | 4EH | 4DH | 4CH | 4BH | 4AH | 49H | 48H |
| 28H       | 47H | 46H | 45H | 44H | 43H | 42H | 41H | 40H |
| 27H       | 3FH | 3EH | 3DH | 3CH | 3BH | 3AH | 39H | 38H |
| 26H       | 37H | 36H | 35H | 34H | 33H | 32H | 31H | 30H |
| 25H       | 2FH | 2EH | 2DH | 2CH | 2BH | 2AH | 29H | 28H |
| 24H       | 27H | 26H | 25H | 24H | 23H | 22H | 21H | 20H |
| 23H       | 1FH | 1EH | 1DH | 1CH | 1BH | 1AH | 19H | 18H |
| 22H       | 17H | 16H | 15H | 14H | 13H | 12H | 11H | 10H |
| 21H       | 0FH | 0EH | 0DH | 0CH | 0BH | 0AH | 09H | 08H |
| 20H       | 07H | 06H | 05H | 04H | 03H | 02H | 01H | 00H |

(3) 30H~7FH: 这 80 字节单元是通用 RAM 区。

## 3.4 特殊功能寄存器

51 单片机总共有 21 个特殊功能寄存器,位于内部数据存储器的高 128 个单元,该区也称作特殊功能寄存器区,它们主要用于存放控制命令、状态或数据,其地址空间为 80H~FFH。这 21 个寄存器中有 11 个特殊功能寄存器具有位寻址能力,它们的字节地址刚好能被 8 整除,如表 3.5 所示。



视频讲解

表 3.5 SFR 特殊功能寄存器地址表

| 符号名(SFR)          | 位地址与位名称 |       |       |       |       |       |       |       | 字节地址 |
|-------------------|---------|-------|-------|-------|-------|-------|-------|-------|------|
|                   | D7      | D6    | D5    | D4    | D3    | D2    | D1    | D0    |      |
| 并行接口 P0           | P0.7    | P0.6  | P0.5  | P0.4  | P0.3  | P0.2  | P0.1  | P0.0  | 80H  |
|                   | 87H     | 86H   | 85H   | 84H   | 83H   | 82H   | 81H   | 80H   |      |
| 堆栈指针 SP           |         |       |       |       |       |       |       |       | 81H  |
|                   |         |       |       |       |       |       |       |       |      |
| 数据指针低字节 DPL       |         |       |       |       |       |       |       |       | 82H  |
|                   |         |       |       |       |       |       |       |       |      |
| 数据指针高字节 DPH       |         |       |       |       |       |       |       |       | 83H  |
|                   |         |       |       |       |       |       |       |       |      |
| 电源控制 PCON         | SMOD    | —     | —     | —     | GF1   | GF0   | PD    | IDL   | 87H  |
|                   |         |       |       |       |       |       |       |       |      |
| 定时器/计数器控制 TCON    | TF1     | TR1   | TF0   | TR0   | IE1   | IT1   | IE0   | IT0   | 88H  |
|                   | 8FH     | 8EH   | 8DH   | 8CH   | 8BH   | 8AH   | 89H   | 88H   |      |
| 定时器/计数器方式控制 TMOD  | GATE    | C/T   | M1    | M0    | GATE  | C/T   | M1    | M0    | 89H  |
|                   |         |       |       |       |       |       |       |       |      |
| 定时器/计数器 0 低字节 TL0 |         |       |       |       |       |       |       |       | 8AH  |
|                   |         |       |       |       |       |       |       |       |      |
| 定时器/计数器 1 低字节 TL1 |         |       |       |       |       |       |       |       | 8BH  |
|                   |         |       |       |       |       |       |       |       |      |
| 定时器/计数器 0 高字节 TH0 |         |       |       |       |       |       |       |       | 8CH  |
|                   |         |       |       |       |       |       |       |       |      |
| 定时器/计数器 1 高字节 TH1 |         |       |       |       |       |       |       |       | 8DH  |
|                   |         |       |       |       |       |       |       |       |      |
| 并行接口 P1           | P1.7    | P1.6  | P1.5  | P1.4  | P1.3  | P1.2  | P1.1  | P1.0  | 90H  |
|                   | 97H     | 96H   | 95H   | 94H   | 93H   | 92H   | 91H   | 90H   |      |
| 串行接口控制 SCON       | SM0     | SM1   | SM2   | REN   | TB8   | RB8   | TI    | RI    | 98H  |
|                   | 9FH     | 9EH   | 9DH   | 9CH   | 9BH   | 9AH   | 99H   | 98H   |      |
| 串行接口数据缓冲器 SBUF    |         |       |       |       |       |       |       |       | 99H  |
|                   |         |       |       |       |       |       |       |       |      |
| 并行接口 P2           | P2.7    | P2.6  | P2.5  | P2.4  | P2.3  | P2.2  | P2.1  | P2.0  | A0H  |
|                   | A7H     | A6H   | A5H   | A4H   | A3H   | A2H   | A1H   | A0H   |      |
| 中断允许控制器 IE        | EA      | —     | ET2   | ES    | ET1   | EX1   | ET0   | EX0   | A8H  |
|                   | AFH     | AEH   | ADH   | ACH   | ABH   | AAH   | A9H   | A8H   |      |
| 并行接口 P3           | P3.7    | P3.6  | P3.5  | P3.4  | P3.3  | P3.2  | P3.1  | P3.0  | B0H  |
|                   | B7H     | B6H   | B5H   | B4H   | B3H   | B2H   | B1H   | B0H   |      |
| 中断优先级控制 IP        | —       | —     | PT2   | PS    | PT1   | PX1   | PT0   | PX0   | B8H  |
|                   | —       | —     | BDH   | BCH   | BBH   | BAH   | B9H   | B8H   |      |
| 程序状态字 PSW         | Cy      | AC    | F0    | RS1   | RS0   | OV    | F1    | P     | D0H  |
|                   | D7H     | D6H   | D5H   | D4H   | D3H   | D2H   | D1H   | D0H   |      |
| 累加器 ACC           | ACC.7   | ACC.6 | ACC.5 | ACC.4 | ACC.3 | ACC.2 | ACC.1 | ACC.0 | E0H  |
|                   | E7H     | E6H   | E5H   | E4H   | E3H   | E2H   | E1H   | E0H   |      |
| 寄存器 B             | B.7     | B.6   | B.5   | B.4   | B.3   | B.2   | B.1   | B.0   | F0H  |
|                   | F7H     | F6H   | F5H   | F4H   | F3H   | F2H   | F1H   | F0H   |      |

在所有 51 单片机中,表 3.5 所列出的特殊功能寄存器的名称、地址和符号都一样。对于 52 增强型单片机而言,增加了一些新的寄存器内容,比如,在 C8H 处增加了定时器/计数器 2 的设置,如表 3.6 所示。

表 3.6 8052 定时器/计数器 2 的地址表

| 符号名   | 位地址与位名称 |      |      |      |       |     |      |        | 字节地址 |
|-------|---------|------|------|------|-------|-----|------|--------|------|
|       | D7      | D6   | D5   | D4   | D3    | D2  | D1   | D0     |      |
| T2CON | TE2     | EXF2 | RCLK | TCLK | EXEN2 | TR2 | C/T2 | CP/PL2 | C8H  |
|       | CFH     | CEH  | CDH  | CCH  | CBH   | CAH | C9H  | C8H    |      |
| RLDL  |         |      |      |      |       |     |      |        | CAH  |
| RLDH  |         |      |      |      |       |     |      |        | CBH  |
| TL2   |         |      |      |      |       |     |      |        | CCH  |
| TH2   |         |      |      |      |       |     |      |        | CDH  |

下面将对部分特殊功能寄存器作简要介绍。

(1) 累加器 A: 累加器 A 是 8 位寄存器,是程序中最常用的专用寄存器,在指令系统中累加器的助记符为 A。大部分单操作数指令的操作数取自累加器,很多双操作数指令的一个操作数也取自累加器。加、减、乘和除等算术运算指令的运算结果都存放在累加器 A 或 AB 寄存器中,在变址寻址方式中累加器被作为变址寄存器使用。由于大部分数据操作都是通过累加器进行的,故累加器的使用十分频繁。累加器 A 自身带有全零标志, $A=0$  则  $Z=1$ (真);  $A \neq 0$  则  $Z=0$ (假)。

(2) B 寄存器: B 寄存器为 8 位寄存器,主要用于乘除指令中。乘法指令的两个操作数分别取自 A 累加器和 B 寄存器,其中, B 为乘数,乘法结果的高 8 位存放于 B 寄存器中。在除法指令中,被除数取自 A,除数取自 B,除法的结果商数存放于 A,余数存放于 B 中。在其他指令中, B 寄存器也可作为一般的数据单元来使用。

(3) 程序状态字(PSW): 程序状态字是一个 8 位寄存器,它包含程序运行的状态信息。在状态字中,有些位状态是根据指令执行结果,由硬件自动完成设置的,而有些状态位则必须通过软件方法设定。PSW 中的每个状态位都可由软件读出,PSW 的各位定义如表 3.7 所示。

表 3.7 PSW 位定义

| 位序  | PSW. 7 | PSW. 6 | PSW. 5 | PSW. 4 | PSW. 3 | PSW. 2 | PSW. 1 | PSW. 0 |
|-----|--------|--------|--------|--------|--------|--------|--------|--------|
| 位标志 | CY     | AC     | F0     | RS1    | RS0    | OV     | —      | P      |

① CY: 进位/借位标志位: 反映了当前运算中的最高位是否有进位/借位情况。加法为进位,减法为借位。如果最高位有进位/借位,则  $CY=1$ (真); 无进位/借位,则  $CY=0$ 。

② AC: 辅助进位标志: 进行加法或减法操作时,当发生低 4 位向高 4 位的进位或借位时,  $AC=1$ ; 如果低 4 位没有向高 4 位进位或借位,则  $AC=0$ 。在进行十进制调整指令时,将借助 AC 状态进行判断。

③ F0 用户标志位,该位为用户定义的状态标记。

④ RS1 和 RS0：寄存器区选择控制位，由设置这两位为 0 或 1 来选择当前工作寄存器区。

⑤ OV：溢出标志位：在带符号的加减运算中，OV=1 表示加减运算结果超出了累加器 A 所能表示的符号数有效范围(−128~+127)，即运算结果溢出；反之，OV=0 表示运算正确，即无溢出产生；

⑥ P：奇偶标志位：累加器 A 中 1 的个数的奇偶性，若累加器中 1 的个数为奇数，则 P=1，否则 P=0。

(4) 数据指针寄存器(Data Point Register, DPTR)：数据指针 DPTR 是一个 16 位的特殊功能寄存器，其高 8 位用 DPH 表示，其低 8 位用 DPL 表示。它既可以作为一个 16 位的寄存器使用，也可作为两个 8 位的寄存器 DPH 和 DPL 使用。DPTR 在访问外部数据存储器时可用来存放 16 位地址，作地址指针使用，例如，MOVX @DPTR, A。

(5) 堆栈指针(Stack Pointer, SP)寄存器：堆栈是一块存储区域，它是用来临时存放调用子程序或进入中断服务程序时主程序断点的位置和其他寄存器内容的存储区域。比如，在调用子程序时，主程序会暂时停止执行，直接跳转到子程序入口地址。当子程序执行完毕会再次回到主程序，在主程序中调用子程序的下一条语句继续执行，如图 3.6 所示。

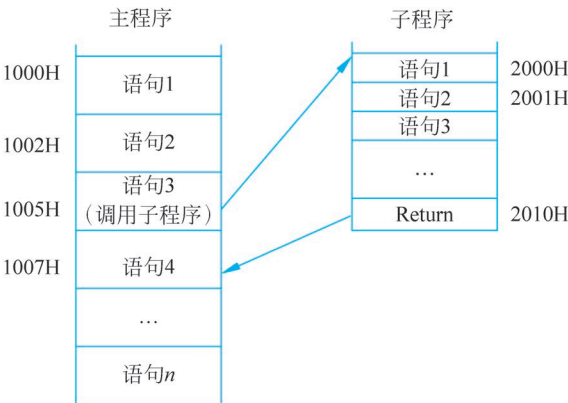


图 3.6 子程序调用

在图 3.6 中，主程序存放在 ROM 存储器的 1000H 处，并按照顺序执行。当执行到 1005H 处的语句调用子程序时，程序会自动跳转到子程序存放在 ROM 存储器中的入口地址 2000H 处执行。直到子程序执行完毕(即执行到 ROM 中存放在 2010H 处的最后一条语句 Return)，程序会自动返回到 1007H，然后继续在主程序中顺序执行。

CPU 是如何在子程序执行完毕后自动返回到主程序中调用子程序的下一条语句的位置处呢？这就是堆栈的作用。在图 3.6 中，当执行到主程序第 3 条语句时，PC=1007H(PC 总是指向下一条需要执行的语句地址)。由于这是一条子程序调用语句，此时 CPU 会将语句 4(调用子程序的下一条语句)的地址 1007H 存放于堆栈，然后将子程序的入口地址 2000H 加载到 PC，从而转向子程序的语句 1 执行。当子程序执行到 2010H 处的最后一条语句，由于该语句是 Return，此时 CPU 将堆栈中存放的 1007H 取出加载到 PC，从而程序转向到主程序 1007H 处的语句 4 执行，即主程序中调用子程序的下一条语句继续执行。

在计算机中(不一定是单片机)，堆栈可以是寄存器，也可以是 RAM。由寄存器构成的堆栈称为硬堆栈，其优点是寄存器位于 CPU 内部，因而访问速度快；缺点是 CPU 内部的寄

寄存器数量有限,因而堆栈的存储量受到限制。用 RAM 构成的堆栈称为软堆栈,理论上软堆栈的存储量没有限制,因而在计算机系统中,绝大多数堆栈都是软堆栈。堆栈具有以下特点:

(1) 堆栈只允许在其一端进行数据插入和删除操作,不是按字节任意访问的,是一个特殊的存储区。它是按“先进后出,后进先出”的原则存取数据,这里的“进”与“出”是指进栈与出栈操作,如图 3.7 所示。

(2) 堆栈指针 SP 用于存放最后一个被存放在堆栈中的数据的地址。每存入(或取出)1 字节数据,SP 就自动加 1(或减 1)。SP 始终指向新的被操作的数据。

(3) 堆栈有栈底和栈顶之分,栈底就是 CPU 初始化时,SP 指向的地址,栈底地址一旦初始化后将保持不变,它决定了堆栈在 RAM 中的位置;栈顶是最后一个被存入堆栈的数据的地址,因此,可以说 SP 总是指向栈顶的。

(4) 堆栈栈底确定后,就可以向堆栈存入数据。存入数据时,有向上生长和向下生长两种方式。如果存入一个数据,堆栈指针增加,则是向上增长方式;如果存入一个数据,堆栈指针减少,则是向下增长。

(5) 有专门的堆栈操作指令。

PUSH: 数据进入堆栈称“进栈(压栈)”。

POP: 数据从堆栈取出称“出栈(弹栈)”。

对 51 单片机来说,其堆栈的特性如下:

(1) 51 单片机的堆栈也服从“先进后出”原则;

(2) 初始化时,SP=07H,即栈底为 07H,此时栈底与栈顶一样都是 07H,堆栈中没有数据,只能执行 PUSH 操作,无法执行 POP 操作;

(3) SP 是一个 8 位的寄存器,即堆栈位于片内 RAM 区域;

(4) 51 单片机的堆栈操作采用向上增长方式,且以“字节”为单位进栈或出栈。

例如,CPU 初始化时,SP=07H,进行如图 3.8 所示的操作。

(1) MOV A, #12H; 将数据 12H 送入 A 中。

(2) MOV SP, #10H; 初始化 SP 指针,栈底为 10H。

(3) PUSH ACC; 将 ACC 中的内容压栈,即将 12H 送入堆栈。在入栈前 SP 先加 1,SP=11H,然后将数据 12H 存放在地址 11H 处,此时栈底仍为 10H,栈顶变为 11H。

(4) MOV 0EH, #34H; 将数据 34H 送入片内 RAM 地址 0EH 处。

(5) PUSH 0EH; 将片内 RAM 地址 0EH 处存放的内容压栈,即将 34H 送入堆栈。在入栈前 SP 先加 1,SP=12H,然后将数据 34H 存放在地址 12H 处。此时栈底仍为 10H,栈顶变为 12H。

(6) POP ACC; 将堆栈的数据弹出到 ACC 中。由于堆栈是先进后出,故先将数据 34H 弹出到 ACC 中,然后 SP 减 1,SP=11H。此时栈底仍为 10H,栈顶变为 11H。

(7) POP 0FH; 将堆栈的数据弹出到片内 RAM 地址 0FH。此时先将数据 12H 弹出到地址 0FH 中,然后 SP 减 1,SP=10H。此时栈底仍为 10H,栈顶变为 10H,栈顶等于栈底。

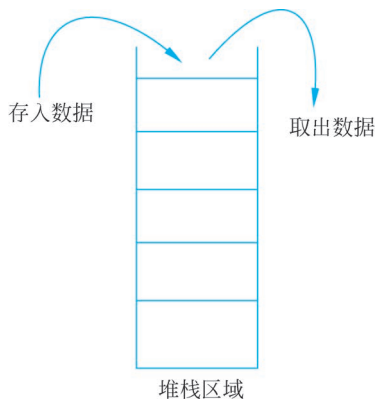


图 3.7 堆栈区域

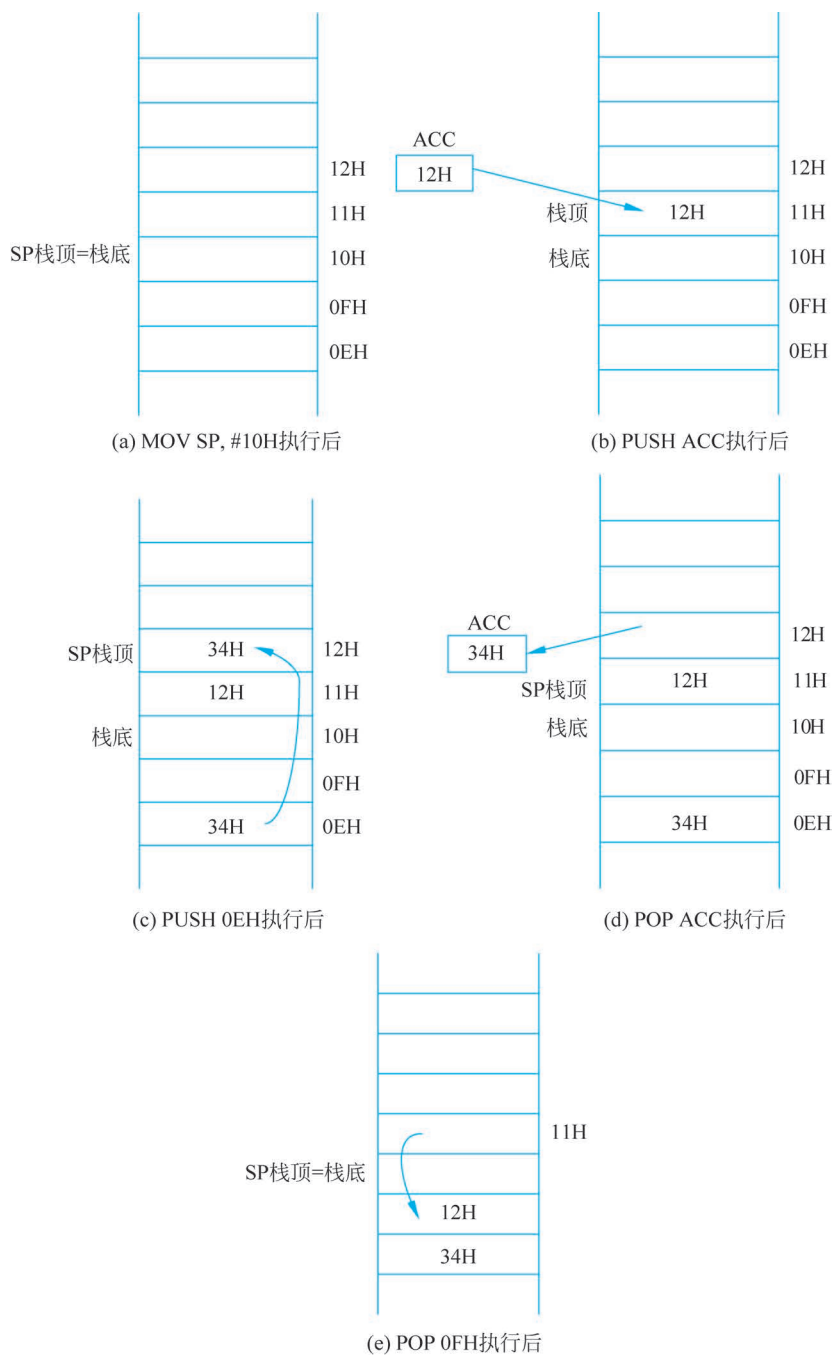


图 3.8 堆栈的操作



视频讲解

## 3.5 时钟电路与复位电路

### 3.5.1 时钟电路

51 单片机的时钟电路有两种方式：内部振荡方式和外部振荡方式。

(1) 内部振荡方式：51 单片机片内有一个用于构成振荡器的高增益反相放大器，引脚 XTAL1 和 XTAL2 分别是此放大器的输入端和输出端。单片机片内的放大器与外部连接的晶体振荡器和微调电容一起构成了稳定的自激振荡器，从而产生振荡脉冲，并输入单片机的内部时钟电路，如图 3.9 所示。

- ① XTAL1(19 脚)：接外部晶振的一个引脚(内部反相放大器的输入端)。
- ② XTAL2(18 脚)：接外部晶振的一个引脚(内部反相放大器的输出端)。

晶体振荡器的振荡频率决定单片机的时钟频率。通常晶体振荡器的时钟频率范围为 1.2~12MHz。电容  $C_1$  和  $C_2$  为微调电容，可起频率稳定、微调作用，一般取值为 5~30pF。

(2) 外部振荡方式：不借助于单片机芯片内的反相放大电路，直接从外部输入振荡脉冲的方式，此时 XTAL1 引脚接地。通常，为了各单片机之间时钟信号的同步，引入唯一的共用外部脉冲信号作为各单片机的振荡脉冲。

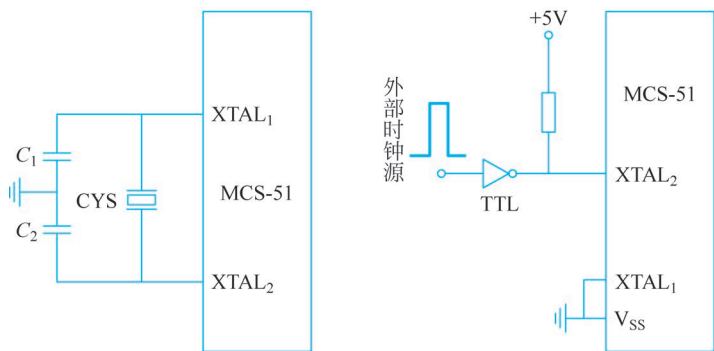


图 3.9 晶振电路

### 3.5.2 基本时序单位

CPU 执行指令的一系列动作是在时序电路控制下一拍一拍进行的，由于各种指令的机器码字节数不同、指令功能不同，因而不同的指令执行时间不一定相同。单片机中有以下几个基本时序单位，如图 3.10 所示。

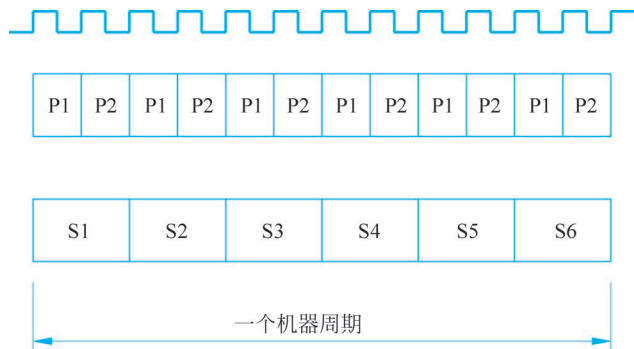


图 3.10 时序单位

(1) 振荡周期(拍节)：振荡脉冲(晶振)的周期称为振荡周期或拍节(用 P 表示)，它是晶振频率的倒数，是单片机中最小的时间单位。

(2) 状态周期(时钟周期)：振荡脉冲经过二分频以后，就是单片机的时钟信号，把时钟

信号的周期定义为状态周期(用 S 表示)。一个状态包含两个节拍,其前半周期对应的节拍叫节拍 1(P1),后半周期对应的节拍叫节拍 2(P2)。

(3) 机器周期: 机器周期是指 CPU 访问一次存储器或 I/O 接口电路所需要的最小时间单位。一个机器周期由 6 个状态 S1~S6(12 个振荡周期)组成。

(4) 指令周期: CPU 取出一条指令至该指令执行完所需时间,不同指令其指令周期不同,但都以机器周期为时间单位。通常,一个指令周期由 1~4 个机器周期组成。

例如,单片机外接晶振为 12MHz,则:  
振荡周期=1/12MHz=(1/12) $\mu$ s=0.0833 $\mu$ s  
状态周期=2 $\times$ 1/12MHz=(2/12) $\mu$ s=0.167 $\mu$ s  
机器周期= 12 $\times$ 1/12MHz =1 $\mu$ s  
指令周期=1~4 $\mu$ s

3.5.3 复位电路

计算机在启动时都需要复位,使系统中的所有部件都处于一个确定的初始状态,并从该状态开始工作。对单片机而言,要实现复位,必须使复位引脚 RST 保持至少两个机器周期(即 24 个振荡周期)的高电平。一般有两种复位方法,如图 3.11 所示。

- (1) 上电复位: 加电瞬间电容充电,RST 出现高电平(最少 2 个机器周期),然后降为低电平。
- (2) 开关复位: 在上电自动复位电路上增加了人工复位。



图 3.11 复位电路

在如图 3.11 所示的电路中,在通电瞬间由于 RC 的充电过程,在 RST 端出现一定宽度的正脉冲,只要该正脉冲保持 2 个机器周期以上,就能使单片机自动复位,在 6MHz 时钟下,通常 CR 取 22 $\mu$ F,R<sub>1</sub> 取 200 $\Omega$ ,R<sub>2</sub> 取 1k $\Omega$ ,这时能可靠地进行上电复位和开关复位。

单片机复位后,不影响片内 RAM 的内容。堆栈指针 SP 为 07H,P0~P3 口都为 1(即 P0~P3 口的内容均为 FFH),当 RST 引脚返回低电平以后,PC 寄存器变为 0000H,即 CPU 从 0000H 地址开始执行程序。复位后其他寄存器的状态如表 3.8 所示。

表 3.8 复位后的寄存器状态

| 寄存器  | 内容    | 寄存器                            | 内容                         | 寄存器             | 内容                                        |
|------|-------|--------------------------------|----------------------------|-----------------|-------------------------------------------|
| PC   | 0000H | P <sub>0</sub> ~P <sub>3</sub> | FFH                        | TL <sub>0</sub> | 00H                                       |
| ACC  | 00H   | IP                             | $\times\times\times0000$ B | TH <sub>1</sub> | 00H                                       |
| B    | 00H   | IE                             | $0\times\times0000$ B      | TL <sub>1</sub> | 00H                                       |
| PSW  | 00H   | TMOD                           | 00H                        | SCON            | 00H                                       |
| SP   | 07H   | TCON                           | 00H                        | SBUF            | 不定                                        |
| DPTR | 0000H | TH <sub>0</sub>                | 00H                        | PCON            | $0\times\times\times\times\times\times$ B |

## 3.6 引脚功能



视频讲解

51 单片机最常见的封装是标准型 DIP(双列直插)40 脚。凡封装相同的 51 单片机,其引脚定义和功能与 8051 兼容,使用时绝大部分器件可以互换。图 3.12 是 40 脚封装的逻辑功能与实际引脚排列图,其中图 3.12(a)是逻辑功能图,图 3.12(b)是实际引脚图。各个引脚的功能如表 3.9 所示。

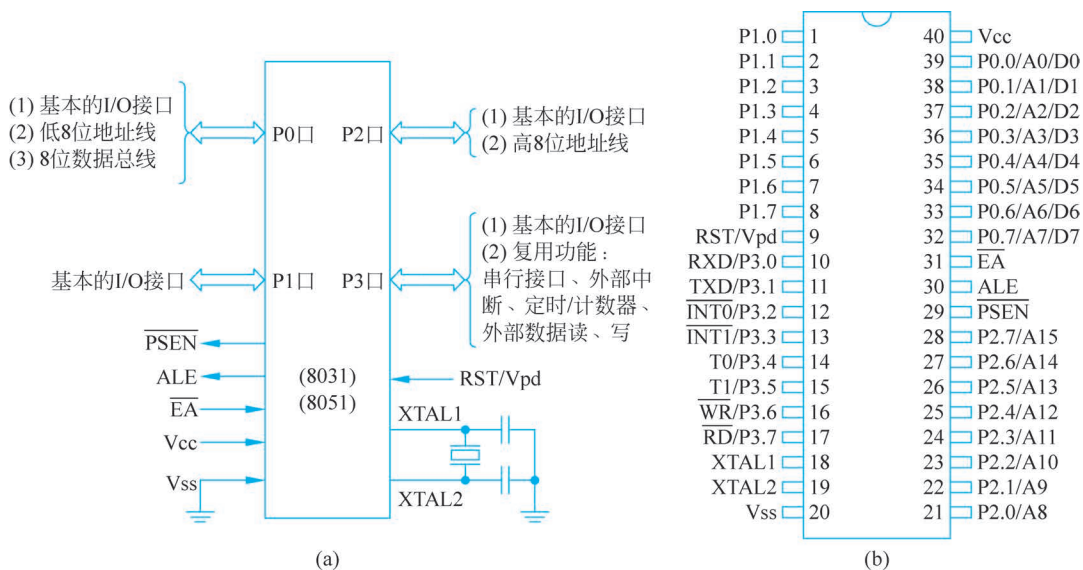


图 3.12 引脚分布

表 3.9 MCS-51 单片机的引脚功能和含义

| 引脚名称      | 脚号    | 功能描述                                                                                                                                                                                |
|-----------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vcc       | 40    | +5V 电源端(+4.5~+5.5V)                                                                                                                                                                 |
| Vss       | 20    | 接地端                                                                                                                                                                                 |
| P0.0~P0.7 | 39~32 | ① P0 口是一个 8 位开漏极双向 I/O 接口;②在使用外部存储器时,低 8 位的地址和数据总线复用这个口                                                                                                                             |
| P1.0~P1.7 | 1~8   | ① P1 口是一个 8 位准双向 I/O 接口,内部已有上拉电阻;②P1 口能够驱动 4 个 LSTTL 负载;③作为输入端的那些引脚,必须先对该位输出 1(即对其编程为输入状态);④在 8052 中,P1.0 和 P1.1 引脚还有 T2 和 T2EX 功能,T2 是定时器/计数器 2 的外部输入端,T2EX 是定时器/计数器 2 的“记录方式”的输出端 |
| P2.0~P2.7 | 21~28 | ① P2 口是一个 8 位准双向 I/O 接口,内部已有上拉电阻;②作为输入端的那些引脚,必须先对该位输出 1(即对其编程为输入状态);③在总线操作时,P2 口为高 8 位地址线                                                                                           |
| P3.0~P3.7 | 10~17 | ① P3 口是一个 8 位准双向 I/O 接口,内部已有上拉电阻;②作为输入端的那些引脚,必须先对该位输出 1(即对其编程为输入状态);③在作特定的复用功能时,P3 复用脚功能                                                                                            |

续表

| 引脚名称                | 脚号 | 功能描述                                                                                                               |
|---------------------|----|--------------------------------------------------------------------------------------------------------------------|
| RST/V <sub>pd</sub> | 9  | ① 复位信号的输入端,高电平有效(非 TTL 电平);②当单片机的主电源 V <sub>cc</sub> 断开时,RST 引脚还是芯片内部 RAM 的辅助电源端(只对 HMOS 芯片有效)                     |
| ALE                 | 30 | ① 地址锁存允许信号,高电平有效;当对外部存储器操作时,ALE 的高电平时,发出地址信号,ALE 的下降沿锁定由 P0 口输出的低 8 位地址信号。②当不用外部存储器时,ALE 可作为标准的脉冲信号(其频率是晶振频率的 1/6) |
| PSEN                | 29 | 是外部程序存储器的选通信号,低有效,能驱动 8 个 LSTTL 负载                                                                                 |
| EA                  |    | ① 当 EA 为高电平时,单片机先执行片内程序存储器的程序,当地址超出内部程序存储器范围时,再执行外部程序存储器的程序;②当 EA 为低电平时,单片机只执行外部程序存储器的程序                           |
| XTAL1               | 19 | ① 内部振荡器的输入端;②当使用外部时钟时,对 CMOS 芯片,XTAL1 为输入端;对 HMOS 芯片,XTAL1 应接地                                                     |
| XTAL2               | 18 | ① 内部振荡器的输出端;②当使用外部时钟时,对 CMOS 芯片,XTAL2 应悬浮不用;对 HMOS 芯片,XTAL2 为输入端                                                   |

3.7 单片机最小系统

单片机的最小系统是指用最少的元件组成的、可以工作的单片机系统。对 51 单片机来说,最小系统的必备条件是需要 ROM 存储程序,需要 RAM 保存中间数据,需要时钟、电源和复位。由于 51 单片机片内有 ROM 和 RAM,所以只要外接晶振电路、复位电路即可构成单片机的最小系统电路(见图 3.13)。注意,单片机的 EA 引脚需要接电源,以保证启动时从片内 ROM 开始执行程序。

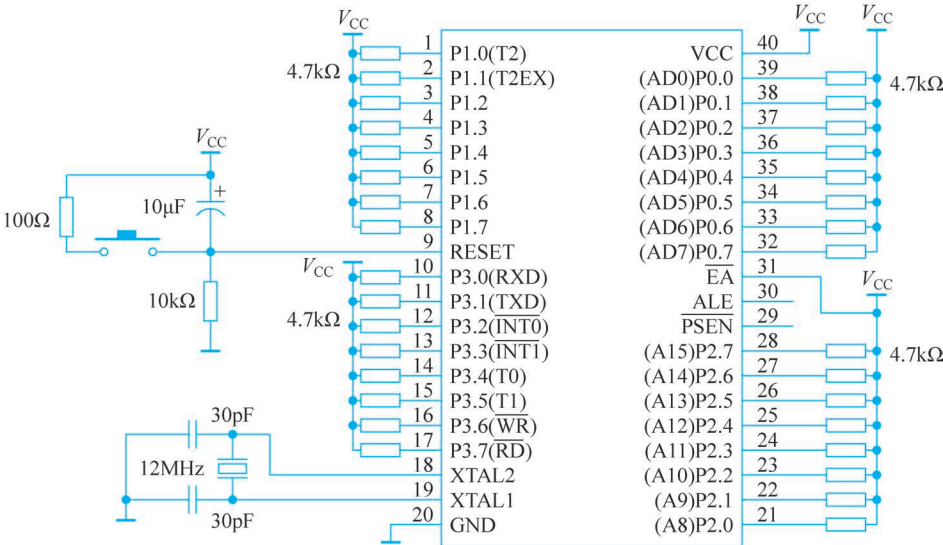


图 3.13 51 单片机最小系统

## 习题

### 1. 填空题

- (1) 若不使用 MCS-51 片内的存储器,引脚  $\overline{EA}$  必须接\_\_\_\_\_。
- (2) MCS-51 中有 4 组工作寄存器,它们的地址范围是\_\_\_\_\_。
- (3) 若单片机的使用的频率为 6MHz 的晶振,那么状态周期为\_\_\_\_\_; 机器周期为\_\_\_\_\_; 指令周期为\_\_\_\_\_。
- (4) 在 MCS-51 系统中,可以进行位寻址的 RAM 区域是\_\_\_\_\_单元。
- (5) MCS-51 系统复位控制信号的有效电平是\_\_\_\_\_。

### 2. 选择题

- (1) 程序计数器在( )中。
 

|            |            |
|------------|------------|
| A. CPU 控制器 | B. CPU 运算器 |
| C. 外部程序存储器 | D. 外部数据存储器 |
- (2) 单片机应用程序一般存放在( )。
 

|        |        |        |        |
|--------|--------|--------|--------|
| A. RAM | B. ROM | C. 寄存器 | D. CPU |
|--------|--------|--------|--------|
- (3) 单片机上电后或复位后,工作寄存器 R0 是在( )。
 

|               |               |
|---------------|---------------|
| A. 0 区 00H 单元 | B. 0 区 01H 单元 |
| C. 0 区 09H 单元 | D. SFR        |
- (4) 进位标志 CY 在( )中。
 

|             |             |
|-------------|-------------|
| A. 累加器      | B. 算术逻辑运算部件 |
| C. 程序状态字寄存器 | D. DPTR     |
- (5) MCS-51 单片机的堆栈区应建立在( )。
 

|                      |            |
|----------------------|------------|
| A. 片内数据存储区的低 128B 单元 | B. 片外数据存储区 |
| C. 片内数据存储区的高 128B 单元 | D. 程序存储区   |
- (6) 用户不能直接使用的寄存器是( )。
 

|        |         |       |      |
|--------|---------|-------|------|
| A. PSW | B. DPTR | C. PC | D. B |
|--------|---------|-------|------|
- (7) MCS-51 复位后,程序计数器 PC=( )。即程序从此处开始执行指令。
 

|          |          |          |          |
|----------|----------|----------|----------|
| A. 0001H | B. 0000H | C. 0003H | D. 0023H |
|----------|----------|----------|----------|
- (8) 当程序状态字寄存器中 RS1 和 RS0 分别为 0 和 1 时,系统先用的工作寄存器组为( )
 

|        |        |        |        |
|--------|--------|--------|--------|
| A. 组 0 | B. 组 1 | C. 组 2 | D. 组 3 |
|--------|--------|--------|--------|

### 3. 判断题

- (1) 当向堆栈压入一个字节的数据后,SP 中的内容减 1。( )
- (2) 程序计数器 PC 中的内容是当前正在执行指令的地址。( )
- (3) 8031 与 8051 的区别在于内部是否有程序存储器。( )
- (4) 8051 单片机,程序存储器数和数据存储器扩展的最大范围都是一样的。( )
- (5) 对于 8051 单片机,当 CPU 对内部程序存储器寻址超过 4KB 时,系统会自动在外部程序存储器中寻址。( )

(6) 当 8051 单片机晶振频率为 12MHz, ALE 地址锁存信号端的输出频率为 2MHz 的方脉冲。( )

(7) 8051 单片机片内 RAM 为 00H~1FH 的 32 个单元, 不仅可以作为工作寄存器使用, 而且可作为 RAM 来读写。( )

(8) MCS-51 的数据存储器在物理上和逻辑上都分为两个地址空间: 一个是片内的 256B 的 RAM, 另一个是片外最大可扩充 64KB 的 RAM。( )

#### 4. 简答题

(1) 单片机包含哪些主要的逻辑部件?

(2) 8051 内部 RAM 分为哪 4 部分?

(3) 使单片机复位有哪几种方式? 单片机复后的初始状态如何?

(4) PSW 的作用是什么? 常用标志位有哪些?

(5) 8051 的时钟周期和振荡周期有什么关系?

(6) 什么叫堆栈? 其作用是什么? 8051 最大的堆栈容量为多少?

(7) 8051 的存储器分哪几个空间? 如何区别不同空间的寻址?