

# 第 3 章

## MCS-51 单片机强健的体魄

### 硬件结构详解

#### 本章概要及学习目标

MCS-51 单片机以其强健的体魄、精妙的构造,在电子世界的舞台上演绎着属于自己的传奇。本章介绍 MCS-51 单片机的硬件结构,如果把单片机比作一个“人”的话,我们接下来的任务就是围绕“大脑(存储器)、心脏(CPU)、五官(I/O 接口)”等“人体”重要的组成部分,逐步了解这个“人”的身体状况。

通过本章的学习,读者应该了解以下知识。

#### 1. 了解 MCS-51 单片机的引脚功能与连接方式

熟悉 MCS-51 单片机的引脚布局和编号规则。

理解每个引脚的主要功能和用途。

掌握引脚的输入输出特性、电压电流规范等。

#### 2. 理解 MCS-51 单片机的内部结构

了解 CPU 的基本组成和工作原理。

理解存储器(ROM 和 RAM)的作用和区别。

掌握 I/O 口、定时器/计数器、中断系统等核心模块的功能。

#### 3. 熟悉 MCS-51 单片机的时钟、复位及工作时序

理解时钟及复位功能的工作原理。

掌握时钟电路及复位电路的连接。

熟悉指令的执行周期和时序。

达到这些学习目标后,我们便能深入理解 MCS-51 单片机的工作原理和操作方法,为后续的学习和实践提供坚实的基础。

#### 开启“硬件结构详解”学习旅程

### 3.1 MCS-51 单片机的外部引脚及功能

#### 本节重点、难点及知识脉络

“MCS-51 单片机的外部引脚及功能”可换一种方式描述,即“单片机长什么样?有什么用?”

**重点：**

长什么样：一个芯片(DIP封装40个引脚,PLCC封装44个引脚……)

有什么用：并行通信、串行通信、定时、计数、中断……

**难点：**每个引脚的功能及输入输出特性。

.....开启“MCS-51单片机的外部引脚及功能”学习旅程.....

### 3.1.1 MCS-51单片机的引脚

单片机长什么样？单片机在外观上通常表现为一个集成了复杂电子电路的微小芯片，其尺寸从几毫米到几十毫米不等，具体形状和大小会根据不同的型号和制造商有所差异，芯片周围分布一些金属针。简单来说，就是一个周边排布金属针的芯片。这些金属针称为“引脚”，引脚是实现单片机功能与外部世界交互的关键接口，它们使得单片机能够控制各种外部设备，并响应来自这些设备或其他外部源的信号。不同芯片的引脚功能会略有区别。常见的封装形式有40脚(排列成两列)的直插封装(DIP)和44脚的方形封装(PLCC)两种。

使用HMOS制造工艺的MCS-51单片机通常采用直插封装方式，如图3.1(a)所示。而采用CMOS工艺制造的80C51和80C52除了可以使用常见的直插封装外，有时也会采用方形封装，如图3.1(b)所示。这两种封装各有自己的使用场景和原因。

#### 1. 40脚的直插封装(DIP)

(1) 这是一种比较常见的封装方式，可以在很多不同的应用中使用。

(2) 对于手工焊接来说很方便，可以通过插入直插式插座来固定在电路板上，更换和维修也比较容易。

(3) 在初学者、教育和实验领域中非常常见，因为它的引脚易于观察和连接。

#### 2. 44脚的方形封装(PLCC)

(1) 这种封装设计得比较紧凑，适用于空间受限的应用场景。

(2) 它含有更多的引脚，引脚密度更高，可以集成更多的功能和接口。

(3) 很容易进行自动化焊接，适合批量生产和表面贴装技术。

(4) 在某些商业产品中很常见，特别是那些需要小型化和高集成度的场合。

总结起来，直插封装(DIP)适用于初学者、教育和实验应用，而方形封装(PLCC)适用于空间受限的商业产品和需要更高集成度的场景，应该根据具体的应用需求和生产要求来选择合适的封装方式。

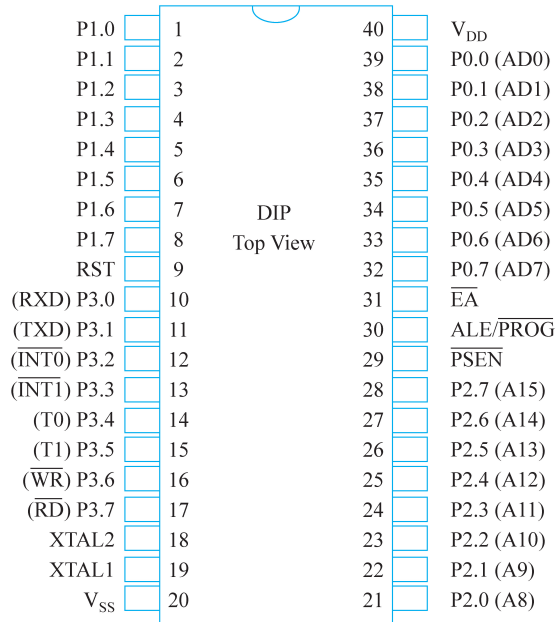
### 3.1.2 MCS-51外部引脚功能

单片机每个引脚都有自己特有的使命。同时，由于引脚数目的限制，有一部分引脚还会具有第二功能，为了方便理解，可以用拟人的方式进行描述。以40脚双列直插封装形式的MCS-51单片机为例，我们可以将引脚的功能进行人性化的解释。根据引脚功能的不同，40个引脚可分为4类。

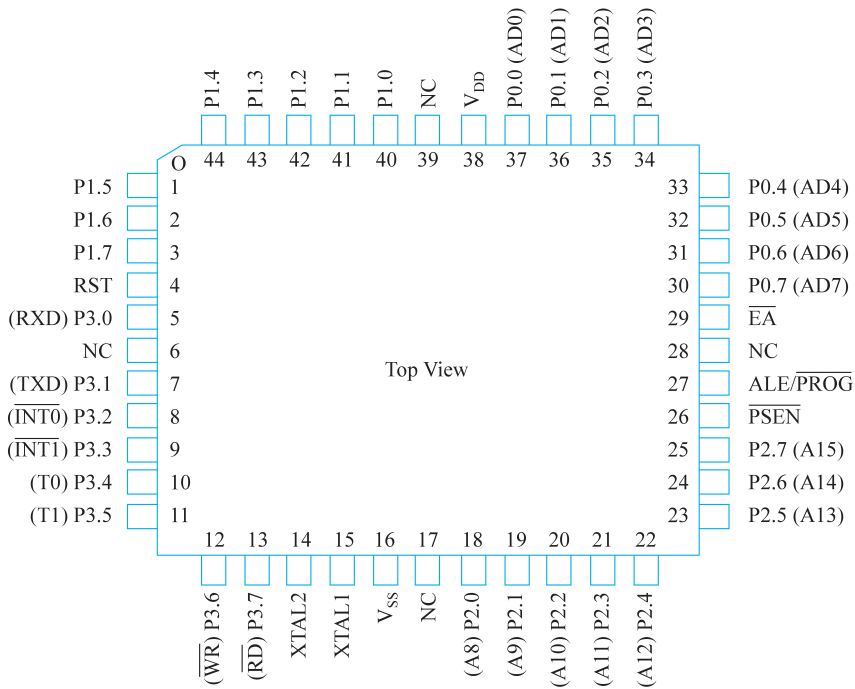
**电源引脚：**人靠吃饭储存体能，单片机靠电源维持活动。

**时钟引脚：**人靠心跳输送血液，单片机靠时钟协调工作。

**控制引脚：**人靠思维支配行为，单片机靠控制发号施令。



(a) 直插封装



(b) 方形封装

图 3.1 单片机的引脚

**I/O 引脚:** 人靠五官传递感知,单片机靠 I/O 传送信息。

### 1. 电源引脚( $V_{CC}$ 和 $V_{SS}$ )

(1)  $V_{CC}$  (40 脚): 电源引脚,像单片机“吃饭”,接收+5V 能量供给。

(2)  $V_{SS}$  (20 脚): 接地引脚,与大地相连,使单片机的电路回路完整。

### 2. 时钟引脚(XTAL1 和 XTAL2)

MCS-51 单片机可以使用内部时钟电路或外部时钟电路提供时钟信号。

当使用内部时钟电路时,需要在 XTAL1 和 XTAL2 之间外接晶体和微调电容。这个连接构成一个谐振电路,晶体的固有频率即为振荡频率,可以通过微调电容对频率进行微调。

使用外部时钟时,外部时钟脉冲由 XTAL2 输入,同时 XTAL1 必须接地。

(1) **XTAL1(19 脚):** 它类似单片机的心跳,是内部振荡电路的反相放大器输入端,接外部晶体和微调电容的一端,确保单片机按正确的频率运行。若采用外部时钟,此引脚必须接地。

(2) **XTAL2(18 脚):** 它像是单片机的心跳检测器,是内部振荡电路的反相放大器输出端,接外部晶体和微调电容的另一端。若采用外部时钟,此引脚为外部时钟脉冲的输入端。可以通过观察此引脚的脉冲信号来判断单片机的振荡电路是否正常工作。

### 3. 控制引脚( $RST/V_{PD}$ 、 $ALE/PROG$ 、 $PSEN$ 、 $EA/V_{PP}$ )

#### 1) $RST/V_{PD}$ (9 脚)

**引脚功能(复用):** 复位信号输入端、备用电源输入端。

**复位信号输入端** 可以将其比喻为单片机的“重启”按钮。如果我们向这个引脚提供一个高电平的信号,并且保持这个信号的持续时间超过 24 个时钟振荡周期(也就是两个机器周期),那么就相当于按下了单片机的“重启”按钮,完成了对单片机的复位操作。

**备用电源输入端** 可以将其想象成单片机的“备用电池”。当主电源  $V_{CC}$  发生故障并降低到特定的电平后,+5V 备用电源会自动连接到这个引脚上,为单片机内部的 RAM 提供备用电源。这样做的目的是防止内部 RAM 中的数据丢失,并确保在主电源供电恢复后单片机可以正常工作。

#### 2) $ALE/PROG$ (Address Latch Enable/Programming, 30 脚)

**引脚功能(复用):** 地址锁存允许信号端、编程脉冲输入端。

**地址锁存允许信号端** 当单片机对片外存储器进行访问时,系统需要给每个外部存储器单元分配唯一的地址。由于引脚数量的限制,地址信息只能复用单片机其他引脚,其中 P0 引脚复用低 8 位地址信息,P2 引脚复用高 8 位地址信息。为了确保地址信息传输的准确性和抗干扰能力,系统需要对地址信息进行锁存,而 ALE 引脚的输出信号被用作低 8 位地址信息的锁存控制信号。

**编程脉冲输入端** 作为片内带有 EPROM 的单片机(8751 等)的编程脉冲输入端。

**引脚特点:**

当单片机不访问外部存储器时,ALE 引脚自动输出频率为时钟振荡器频率的 1/6 的正脉冲信号,这种脉冲信号可以作为时钟基准。

当单片机访问外部存储器时,在每两个机器周期中只会出现一个 ALE 信号,即丢失一个 ALE 脉冲信号。这种情况下,ALE 输出的脉冲信号不适合作时钟基准使用。

无论怎样,只要单片机上电正常工作,ALE 引脚必定会输出脉冲信号。我们可以使用示波器观察 ALE 端输出状态来判断单片机的好坏。如果有正脉冲输出,基本上确定单片机是好的。

3)  $\overline{\text{PSEN}}$ (Program Store Enable,29 脚): 程序存储器允许输出信号端

可以将 $\overline{\text{PSEN}}$ 想象成单片机和外部程序存储器之间的一个“送信员”。当单片机需要和外部程序存储器交流时,“送信员”会定时输出一种特殊的信号“负脉冲”,就像敲门一样。这个信号告诉外部程序存储器:“你好,我要和你说话啦!”外部程序存储器听到信号后,就知道单片机想要读取指令码了。

为了检查单片机是否工作正常,我们可以使用示波器来关注 $\overline{\text{PSEN}}$ 是否在按时输出信号,就好像看他是否敲门一样。如果我们能听到负脉冲声音,那一般意味着单片机状态良好。

4)  $\overline{\text{EA}}/\text{V}_{\text{PP}}$ (Enable Address/Voltage Pulse of Programming,31 脚): 片外程序存储器地址允许/固化编程电压输入

**引脚功能(复用): 程序存储器的访问方式、编程电压输入端。**

**程序存储器的访问方式** 可以将 $\overline{\text{EA}}/\text{V}_{\text{PP}}$ 想象成单片机和内部程序存储器之间的一扇“门”。

**$\overline{\text{EA}}/\text{V}_{\text{PP}}$ 接收到高电平** 单片机和内部程序存储器之间的“大门被打开”。单片机会乖乖地执行内部程序存储器的指令。当单片机把内部程序存储器的指令都执行了一遍还没接到“收工”信号,也就是超过了 0FFFH 这个地址(8051/8751 只有 4KB 的内部程序存储器),单片机就会自动去找外部程序存储器,并从外部程序存储器 1000H 地址里面的内容开始继续执行指令。

**$\overline{\text{EA}}/\text{V}_{\text{PP}}$ 接收到低电平** 单片机和内部程序存储器之间的“大门被关闭”。单片机完全忽略内部程序存储器,直接去执行外部程序存储器的指令,而不管是否有内部程序存储器存在。对于没有内部程序存储器的单片机 8031 来说,它必须扩展外部程序存储器才能有效工作,所以这个时候, $\overline{\text{EA}}/\text{V}_{\text{PP}}$ 一定为低电平,呈现出“大门被关闭”的状态,这样才能正确地访问到外部程序存储器。

**编程电压输入端** 如果要对 8751 内部 EPROM 进行编程,则需要给它一个较高的电压输入。这个时候,就需要把这个高电压连接到  $\text{V}_{\text{PP}}$  身上。

#### 4. I/O(输入/输出)引脚(P0、P1、P2、P3)

四组并行口,每一组都有 8 个完全一样的位。可以将每一组并行口想象成运动场上的 8 条并行跑道。每条跑道上容纳 1 个人;运动场的并行跑道上最多同时存在 8 个人;每个人在各自的跑道上,既可以顺时针“跑”,也可以逆时针“跑”。同理,每个 I/O 口可以传输 1 个二进制位;一组并行口最多同时传送 8 个二进制位;每个二进制位在各自的 I/O 口线上,既可以输入(从外部设备传入至单片机),也可以输出(从单片机输出至外部设备)。

1) P0 口(P0.0~P0.7,39 脚~32 脚): 8 位双向三态 I/O 口

P0 口可以驱动 8 个 LS 型 TTL 负载,当 CPU 访问外部存储器或进行 I/O 口扩展时,P0 口作为**低 8 位地址总线**和**数据总线**使用。需要注意的是,由于 P0 口的漏极开路电路特性,它内部没有固定的上拉电阻。因此,在将其用作普通输出口时,**须外接上拉电阻**以确保正确输出高电平。

2) P1 口(P1.0~P1.7,1 脚~8 脚): 8 位准双向 I/O 口

P1 口内部有上拉电阻,每一位都可以驱动 4 个 LS 型 TTL 负载。

3) P2 口(P2.0~P2.7,21 脚~28 脚): 8 位准双向 I/O 口

P2 口内部有上拉电阻,每一位都可以驱动 4 个 LS 型 TTL 负载。在单片机访问外部程序存储器时可分时,P2 口分时复用为高 8 位地址总线。

4) P3 口(P3.0~P3.7,10 脚~17 脚): 8 位准双向 I/O 口,第二功能复用口

P3 口内部有上拉电阻,每一位都可以驱动 4 个 LS 型 TTL 负载。与其他 I/O 口不同的是,P3 口的每个引脚除了可以作为普通 I/O 口使用外,还具有第二功能,具体功能如表 3.1 所示。

表 3.1 P3 口各引脚第二功能表

引 脚	第 二 功 能
P3.0	RXD(串行输入端)
P3.1	TXD(串行输出端)
P3.2	$\overline{\text{INT0}}$ (外部中断 0 输入端)
P3.3	$\overline{\text{INT1}}$ (外部中断 1 输入端)
P3.4	T0(定时器 0 外部输入端)
P3.5	T1(定时器 1 外部输入端)
P3.6	$\overline{\text{WR}}$ (外部数据存储器写选通输出端)
P3.7	$\overline{\text{RD}}$ (外部数据存储器读选通输出端)

需要注意的是,P1 口、P2 口、P3 口内部均带有上拉电阻,当它们作为输入口使用时,需要先向该端口各位写入“1”,这也是它们被称为准双向的原因。

MCS-51 并行口主要特性如表 3.2 所示。

表 3.2 MCS-51 并行口主要特性

并行口	I/O 功能	其他 功 能	驱 动 能 力	上 拉 电 阻	输 入 条 件
P0 口	双向 I/O 口	地址总线低 8 位 数据总线共 8 位	8 个 LS 型 TTL	需要外接	先向外写“1”
P1 口	准双向 I/O 口	无	4 个 LS 型 TTL	无须外接	先向外写“1”
P2 口	准双向 I/O 口	地址总线高 8 位	4 个 LS 型 TTL	无须外接	先向外写“1”
P3 口	准双向 I/O 口	第二功能	4 个 LS 型 TTL	无须外接	先向外写“1”

## 3.2 MCS-51 单片机的内部结构

### 本节重点、难点及知识脉络

主线:

单片机内部三大组成部分(CPU、存储器、I/O 接口)。

**CPU**(运算器、控制器)。

**存储器**(程序存储器、数据存储器)。

**I/O 接口**(P0、P1、P2、P3)。

**重点:**

1. CPU 工作原理。
2. 存储器单元分配。
3. I/O 接口工作原理。

**难点:**

I/O 接口电路组成及分析。

### ..... 开启“MCS-51 单片机的内部结构”学习旅程 .....

MCS-51 单片机的内部结构主要包含 CPU(中央处理器)、存储器和 I/O 接口这三部分。CPU 内部包含运算器和控制器模块。存储器则被划分为数据存储器 and 程序存储器两部分。此外,I/O 接口用于外部设备与单片机之间的数据传输。下面我们将详细了解这些部分的功能和注意事项。

## 3.2.1 MCS-51 单片机的 CPU

CPU(中央处理器)由**运算器**和**控制器**组成。运算器负责执行各种算术和逻辑运算,使单片机能够进行复杂的数据处理。控制器则负责指挥和协调单片机的操作流程,在执行程序时控制命令的顺序和时序。

### 1. 运算器

运算器主要用于执行算术运算、逻辑运算和位操作,使单片机能够进行复杂的数据处理。它包含以下几个关键部件。

#### 1) 算术逻辑部件(ALU)

ALU 可以进行各种算术运算,如加、减、乘、除等,也可以执行逻辑操作,如与、或、异或、取反等。它是实现各种数学和逻辑运算的核心部件。

#### 2) 累加器 A

累加器 A 又称为 ACC,是一个 8 位的特殊用途寄存器。在 ALU 进行运算时,通常将其中一个操作数放入累加器 A 中,并将运算结果存回累加器 A。此外,累加器 A 还可作为数据传输的中转站,在 CPU 内部的数据传送过程中经常使用累加器 A。

#### 3) 寄存器 B

寄存器 B 是一个 8 位的通用寄存器。在乘法和除法运算中,寄存器 B 用来存放其中一个操作数(另一个放在累加器 A 中),并且与累加器 A 一起来保存运算结果。在乘法运算中,寄存器 B 保存结果的高 8 位,累加器 A 保存结果的低 8 位;而在除法运算中,寄存器 B 保存余数,累加器 A 保存商。除乘除运算外,寄存器 B 也可以作为普通的数据寄存器使用。

**举例说明:**

假设我们要将 251(十六进制为 0xFB)除以 10(十六进制为 0x0A),并关注商和余数。

准备阶段:

将 251(0xFB)放入累加器 A。将 10(0x0A)放入寄存器 B。

执行除法：

触发除法操作，产生商和余数。

结果存储：

商(25,十六进制为 0x19)被存储在累加器 A 中。余数(1,十六进制为 0x01)被存储在寄存器 B 中。

#### 4) 程序状态字寄存器(PSW)

PSW 是一个 8 位的特殊功能寄存器，主要用于“发送命令和显示状态”，负责存储 CPU 的状态信息，如标志位(表示运算结果的状态)、控制位等。它在程序执行过程中起着重要的作用，影响着程序的运行和结果。PSW 位于片内的特殊功能寄存器区，字节地址为 D0H，此外 PSW 的每一位都有位地址，可进行位寻址。在学习过程中，应该掌握并牢记其各位的含义。PSW 的格式如图 3.2 所示。



图 3.2 PSW 的格式

PSW 各位含义如下。

##### (1) CY(PSW.7)：进位标志位。

进位标志位 CY 用于指示运算结果中是否发生了进位或借位。当运算结果的低半字节向高半字节进位或借位时，CY 位由硬件置为“1”，否则清零。CY 位主要用于在算术运算、移位操作、逻辑运算等指令中进行进位或借位的判断。

##### (2) AC(PSW.6)：辅助进位标志位。

辅助进位标志位 AC 用于指示运算结果的低半字节向高半字节是否发生了进位或借位。当运算结果的低半字节向高半字节进位或借位时，AC 位由硬件置为“1”，否则清零。AC 位的主要作用是辅助实现 BCD 码加减运算时的进位调整。

##### (3) F0(PSW.5)：用户标志位。

用户标志位 F0 可以通过软件设置为“1”或“0”，以满足特定的用户需求。软件可以根据需要给 F0 位赋予某种特殊意义，可以用作某个条件标志或其他自定义标志的状态。

##### 举例说明：

在某个应用中，F0 位可以表示一个特定条件是否满足。例如，F0 位置为“1”表示温度超过设定阈值，清零表示温度未超过阈值。

##### (4) RS0、RS1(PSW.3、PSW.4)：工作寄存器组选择位。

RS0 和 RS1 是两个联合使用的位，用于选择 MCS-51 单片机中四组工作寄存器中的某一组作为当前的工作寄存器组。通过改变 RS0 和 RS1 的状态，软件可以实现对当前工作寄存器组的选择，从而方便地在不同的程序段之间切换，其组合关系如表 3.3 所示。

表 3.3 RS0 和 RS1 的组合关系

RS1	RS0	工作寄存器组	片内 RAM 地址
0	0	0	00H~07H
0	1	1	08H~0FH

续表

RS1	RS0	工作寄存器组	片内 RAM 地址
1	0	2	10H~17H
1	1	3	18H~1FH

(5) OV(PSW.2): 溢出标志位。

OV 位用于指示当前运算是否发生了溢出。当运算结果超出了有限范围时,OV 位会由硬件自动置为“1”,否则清零。

(6) -(PSW.1): 保留位。

这是保留位,未被使用,其值没有特定含义。

(7) P(PSW.0): 奇偶标志位。

P 位用于标识累加器 A 中“1”的个数。当 A 中“1”的个数为奇数时,则置为“1”;当 A 中“1”的个数为偶数时,则置为“0”。P 位常用于检验串行通信中数据传输是否出错。

#### 举例说明:

假设累加器 A 中的数据为 57H(01010111B),其中包含 5 个“1”。当进行奇偶校验时,对该字节进行统计,发现有奇数个“1”,因此 P 位会被置为“1”,表明校验结果为奇校验。

#### 运算器需要注意的几点:

ALU 能够执行基本的二进制算术和逻辑运算,但只能处理固定位宽的数据。例如,在 MCS-51 单片机中,ALU 的操作位宽为 8 位。

累加器 A(Accumulator)是一个重要的寄存器,用于存储运算结果或操作数。

寄存器 B 被用作辅助存储器,用于一些特定的指令操作。

程序状态字(Program Status Word,PSW)用于存储运行时的各种标志位,如进位标志(Carry Flag)、零标志(Zero Flag)等。

在进行计算时,需要注意位宽匹配、溢出和进位等问题。

#### 运算器小结:

(1) 勤劳的累加器 A 不断地“传数”,热心肠的寄存器 B 不时地“来帮忙”,无所不能的算术逻辑部件(ALU)主打一个字“干”,负责监控的程序状态字(PSW)时刻“监督工作状态”。

(2) 运算器中涉及的 3 个特殊功能寄存器为 A、B、PSW。特殊功能寄存器的作用后面会介绍。

## 2. 控制器

MCS-51 单片机中的控制器是其内部结构中一个非常重要的部分,它负责处理指令的执行顺序和时序。为了更好地理解控制器的作用,我们可以将其比喻成一位聪明的指挥官,在单片机执行任务时负责协调各个部件的工作。

首先,我们来介绍一下控制器中的几个重要组成部分。

### 1) 程序计数器 PC

假设一本厚厚的书代表存储程序代码的存储器。那么,程序计数器 PC 就像是一个会自动跳转的书签,它保存了下一条即将执行的指令的地址。当单片机需要执行下一条指令时,它会通过程序计数器 PC 找到下一条指令所在的地址,并将该地址发送给内存单元。这

样,单片机就可以按照顺序依次执行指令了。

#### 2) 指令寄存器 IR

指令寄存器 IR 可以看作控制器的大脑,它临时存储着从内存中读取的指令。当指令被加载到指令寄存器 IR 中后,控制器就会开始对指令进行解析和译码,然后生成相应的控制信号,用于执行指令所规定的操作。可以说,指令寄存器 IR 起着传递指令和控制信号的桥梁作用。

#### 3) 指令译码器 ID

如果把控制器比作战场上的指挥官,那么指令译码器 ID 就是他的智囊团。指令译码器 ID 负责解析和识别指令的含义,并根据指令的要求生成相应的操作信号。例如,当控制器读取到一条加法指令时,指令译码器 ID 会识别这是一条加法指令,并发送信号给其他部件,让它们做好执行加法运算的准备。

#### 4) 数据指针 DPTR

数据指针 DPTR 可以看作一个特殊的仓库管理员,负责管理与数据存储器相关的地址信息。单片机在执行指令时,可能需要读取或写入存储器中的数据,这时就需要使用数据指针 DPTR 来指示具体的内存地址。通过改变数据指针 DPTR 的值,单片机可以实现对存储器中数据的读写操作。

#### 5) 堆栈指针 SP

堆栈指针 SP 就像控制器的备忘录,用来记录当前的执行状态。当单片机执行子程序或遇到中断时,它会将当前的执行状态(如 PC、寄存器等)保存在堆栈中,并通过堆栈指针 SP 来指示当前保存的位置。当子程序执行完成或中断处理结束后,单片机会根据堆栈指针 SP 的指示,将之前保存的状态恢复,继续执行主程序。

控制器中的这些部分相互协作,共同完成指令的执行和任务的处理。

#### 控制器需要注意的几点:

(1) PC 是由两个 8 位计数器 PCH 及 PCL 组成的 16 位计数器,可对 64KB 范围内的程序存储器寻址。CPU 每读取指令的一字节,PC 值就自动加 1。由此,实现了单片机程序的自动执行。PC 在物理结构上是独立的,不属于特殊功能寄存器,它没有地址,无法读写,只能通过转移、调用或返回等指令改变其内容,以实现程序的转移。

(2) DPTR 是由两个 8 位寄存器 DPH 及 DPL 组成的 16 位特殊功能寄存器,也可以作为两个 8 位的寄存器来使用。它有固定的地址,可以进行读写操作。

(3) SP 指示栈顶的位置,采用“后进先出”的结构方式,主要用于:CPU 响应中断或进行子程序调用时,对现场和断点地址进行保护,以便稍后 CPU 能顺利返回、正确执行。SP 可指向片内 RAM 中 00H~7FH 的任何位置。堆栈中每压入一字节的数据,SP 自动加 1;每弹出一字节的数据,SP 自动减 1。CPU 复位后,SP 的值被初始化到片内 RAM 地址 07H。

#### 控制器小结:

CPU 通过 PC 指针,从程序存储器中提取指令,然后将指令送入指令寄存器 IR;接着,指令译码器 ID 对指令进行解析和译码;最后,通过定时和控制逻辑电路,根据规定的时间序列释放控制信号,使各个部件协调工作,完成指令所规定的任务。简单来说:PC 找“程序”,DPTR 找“数据”,IR 用来“存储”,ID 进行“分析”,控制电路负责“干活”。

### 3.2.2 MCS-51 单片机的存储器

存储器相当于人类的大脑,拥有记忆功能,主要用来存储程序和数据。此外,给某些存储单元赋予特定的含义便可通过对其访问实现控制及状态查询等功能。MCS-51 单片机采用了哈佛结构的存储器,这意味着程序存储器和数据存储器在物理上是分开的。它一般包括片内程序存储器、片外程序存储器、片内数据存储器和片外数据存储器。最基本的存储空间为 1B,即 1 个存储单元内存放 1 字节的信息。从使用的角度具体来看,MCS-51 单片机的存储器地址空间可以分为四类。

(1) **片内和片外统一编址的 64KB 程序存储器空间(0000H~FFFFH)**: 这个存储器空间用于存储程序代码。4KB 以内的存储空间,不管是单片机内部还是外部,都可以通过相同的地址范围进行访问。

(2) **片外 64KB 数据存储器空间(0000H~FFFFH)**: 这个存储器空间用于存储数据。它是与芯片连接的外部存储器,可以用来扩展单片机的数据存储能力。

(3) **片内 128B 数据存储器空间(00H~7FH)**: 这个存储器空间也用于存储数据,但容量相对较小。它位于单片机内部,用于存储临时变量、函数的局部变量等。

(4) **片内 128B 特殊功能寄存器存储空间(80H~FFH)**: 这个存储器空间用于存储具有特殊功能的寄存器,其中有 21 个存储单元已被定义,可根据需求直接访问。

MCS-51 单片机内部的存储器空间存在重叠的情况,如图 3.3 所示。为了区分不同的存储器空间,CPU 在数据操作过程中使用不同的指令。指令系统中设计了特定的数据传送指令。

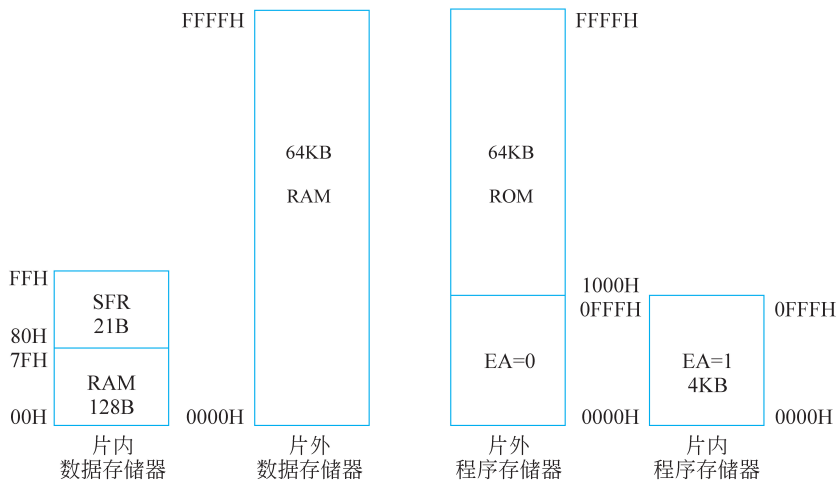


图 3.3 存储器空间分配图

(1) **MOVC(Move Code)**用于访问片内和片外程序存储器空间。

(2) **MOVX(Move External)**用于访问片外数据存储器空间。

(3) **MOV(Move)**用于访问片内数据存储器 and SFR(Special Function Register)空间。

有关这些指令的详细内容将在后面的章节中进行介绍。通过这些指令,单片机可以方便地进行数据的读取和存储操作。

程序存储器可以比作一个大仓库。这个仓库可以存放各种物品,比如程序和表格等常数。由于采用16位的程序计数器PC和16位的地址总线,因此**可访问的程序存储器空间最大为 $2^{16}$ B即64KB**。整个程序空间分为片内和片外两部分,**CPU访问的程序存储器究竟是片内存储器还是片外存储器可由EA引脚所接电平来确定**。

对于8051、8751单片机而言,内部带有4KB的ROM、EPROM。如果用内部的存储单元,则需要将EA引脚接高电平,此时,当内部存储空间都执行完还没有收到单片机的结束指令时,则CPU转向外部程序存储器1000H(0FFFH+1H)地址处继续执行。如果只想用外部的存储单元,则EA引脚接低电平。CPU从片内程序存储器和片外程序存储器中执行程序的速度是相同的。需要注意的是当使用8031单片机时,由于其内部没有程序存储器,所以EA引脚必须接低电平,CPU只能到片外程序存储器执行程序,别无选择。

程序存储器中有一些特殊用途的单元,被预留给系统使用。这些单元的功能如下。

0000H~0002H: 复位后初始化。

0003H~000AH: 外部中断0。

000BH~0012H: 定时器0溢出中断。

0013H~001AH: 外部中断1。

001BH~0022H: 定时器1溢出中断。

0023H~002AH: 串行口中断。

002BH: 定时器2溢出中断(52系列单片机)。

0000H~002AH这部分存储空间与MCS-51单片机有关。0003H~002AH这部分空间被划分为5个段,用作5种中断服务程序的入口地址。表3.4给出了MCS-51单片机的中断入口地址。

复位后,程序计数器PC的值为0000H,仔细观察可以发现,除了“复位后初始化”占用0000H~0002H3字节单元,其他每一种中断区域都是8字节单元,而仅有的3字节或8字节极有可能装不下对应功能的指令程序。因此,通常情况下会在每一段的入口地址处放一条转移指令,避免和下一个特殊功能空间冲突。即在0000H~0002H这部分存储单元中放入转移指令,引导程序跳转到程序存储器的特定位置执行程序。在每个中断入口处也会放入转移指令,以便引导程序跳转到指定的中断服务程序位置执行中断服务程序。

表 3.4 中断入口地址

中断入口地址	中 断 源
0003H	外部中断0(INT0)
000BH	定时器0(T0)
0013H	外部中断1(INT1)
001BH	定时器1(T1)
0023H	串行口

数据存储器顾名思义是用来存储数据信息的。MCS-51单片机内部数据存储器容量为

128B,即最多可存储 128 字节的数据信息。52 系列单片机内部数据存储器容量为 256B,当内部数据存储器空间不足时,可进行外部扩展,外扩的数据存储空间采用 16 位的数据指针 DPTR 和 16 位的地址总线,因而可访问的外部数据存储器空间最大为  $2^{16}$  B 即 64KB。内部数据存储器地址范围是 00H~7FH,外部数据存储器地址范围是 0000H~FFFFH。0000H~007FH 这一地址范围是重叠的,既可以对应内部数据存储器,也可以对应外部数据存储器。为了区分这两个存储区,MCS-51 单片机使用不同的指令来操作它们。对于内部数据存储器,使用 MOV 指令来进行读写操作;而对于外部数据存储器,则需要使用 MOVX 指令来进行读写操作。

**MCS-51 单片机的片内数据存储器为 128B,地址范围为 00H~7FH。**它包含 3 部分,分别是通用工作寄存器区、位寻址区及用户 RAM 区。对于它的寻址可采用直接寻址和间接寻址来实现。其结构如图 3.4 所示。



图 3.4 片内数据存储器的结构

### 1. 通用工作寄存器区(00H~1FH)

MCS-51 单片机共有 4 组通用工作寄存器,每组由 8 个工作寄存器组成,编号为 R0~R7,共占用 32 个存储单元。4 组通用工作寄存器都可作为 CPU 的当前工作寄存器组,可通过指令对 PSW 中的 RS0、RS1 修改来实现当前工作寄存器组的选择。这 4 组通用工作寄存器也可以作为一般 RAM 单元使用。当 CPU 复位后,第 0 组被选为当前工作寄存器组。工作寄存器地址如表 3.5 所示。

表 3.5 工作寄存器地址

工作寄存器组	RS1	RS0	R0	R1	R2	R3	R4	R5	R6	R7
0	0	0	00H	01H	02H	03H	04H	05H	06H	07H
1	0	1	08H	09H	0AH	0BH	0CH	0DH	0EH	0FH
2	1	0	10H	11H	12H	13H	14H	15H	16H	17H
3	1	1	18H	19H	1AH	1BH	1CH	1DH	1EH	1FH

#### 举例说明:

假设当前工作寄存器组选择第 2 组,则需要对“RS1、RS0”赋值“1、0”,此时 R0 的字节地址为 10H,R7 的字节地址为 17H。

### 2. 位寻址区(20H~2FH)

16 个片内 RAM 单元可进行位寻址操作,每个单元由 8 位组成,共 128 位,位地址范围为 00H~7FH。同时这 16 个单元也可进行字节寻址操作。位寻址区地址分配如表 3.6 所示。

表 3.6 位寻址区地址分配

字节地址	位地址							
	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

**举例说明：**

假设要访问字节地址 21H 里面存放 8 位中的 D0 位,则可以访问位地址 08H。如果要访问字节地址 2FH 里面存放 8 位中的 D7 位,则可以访问位地址 7FH。

**3. 用户 RAM 区(30H~7FH)**

用户 RAM 区有 80 字节,供用户自由使用。

**4. 52 系列单片机高 128B 片内 RAM(80H~FFH)**

MCS-51 单片机中的 8032、8052 和 8752 片内 RAM 容量为 256B,高 128B 地址与 SFR 的地址重叠,靠寻址方式区分,SFR 使用直接寻址,内部 RAM 的 80H~FFH 使用寄存器间接寻址。

**特殊功能寄存器 SFR(80H~FFH)：**

MCS-51 单片机的 CPU 对各功能部件的控制是采用特殊功能寄存器(Special Function Register, SFR)的集中控制方式。SFR 实质上是一些具有特殊功能的片内 RAM 单元,字节地址区间为 80H~FFH。其中分布着 PC 及其他 21 个 SFR。可采用直接寻址的方式对 SFR 进行寻址,同时个别 SFR 还可以进行位寻址操作。表 3.7 列出了各 SFR 的名称、功能及地址。



续表

符号	名称	字节地址	位地址							
			D7	D6	D5	D4	D3	D2	D1	D0
* RCAP2H	定时器/计数器 2 记录寄存器(高字节)	CBH								
* RCAP2L	定时器/计数器 2 记录寄存器(低字节)	CAH								
SCON	串行口控制寄存器	98H	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H
			SM0	SM1	SM2	REN	TB8	RB8	TI	RI
SBUF	串行数据缓冲寄存器	99H								
PCON	电源控制寄存器	97H	SMOD	—	—	—	GF1	GF0	PD	IDL

注：带“\*”号的 SFR 是 8052 单片机才有的特殊功能寄存器。

从表 3.7 可以看出, MCS-51 单片机的 21 个 SFR 在此区间内的分布是离散的, 在 80H~FFH 这段存储空间内有部分单元未占用, 不能使用。同时, 所有可位寻址的 SFR 的特点是, 其地址可以被 8 整除。

#### 举例说明：

字节地址为 80H 的 P0 可进行位寻址, 即可对 P0.0~P0.7 每一位独立访问。

字节地址为 97H 的 PCON 不可进行位寻址, 即想改变 PCON 中最高位 SMOD 的值, 只能通过字节传送指令对 PCON 整体重新赋值。

### 3.2.3 MCS-51 单片机的 I/O 接口

I/O 接口相当于人类的五官, 拥有传输功能, 主要用来实现人机交互(人和单片机之间沟通联络)。MCS-51 单片机具有 4 个 8 位并行 I/O 口(如同 4 个运动场, 每个运动场都有 8 条并行跑道), 共计 32 根口线(即 32 条可传送数据的“跑道”), 各 I/O 口分别记作 P0、P1、P2 和 P3。每个端口都包括 8 位双向口, 各 I/O 口线都可独立地用于输入或输出。每个端口都有自己的锁存器(即特殊功能寄存器 P0~P3)、输出驱动器和输入缓冲器, 可以完成对输入数据的缓冲及对输出数据的锁存。4 个 I/O 口的结构略有不同, 各自所能完成的功能也有一定的差异。

#### 1. P0 口

P0 口中各位具有结构完全相同但又相互独立的内部逻辑电路。图 3.5 给出了 P0 口中某一位的内部电路结构。

由图 3.5 可以看出, 它由一个输出锁存器、两个三态输入缓冲器和输出驱动电路及控制电路组成。其工作状态受控制电路与门、反相器及转换开关 MUX 控制。

P0 口可分时作为地址/数据总线使用, 也可作为一般 I/O 口使用。

##### 1) P0 口作为地址/数据总线

如果系统内部存储器空间不足, 则需要扩展外部存储器。当 CPU 需要对外部存储器进行读写操作时, 需要提前解决三个问题。第一, 要访问的外部存储器地址是多少(外扩了那么多存储单元, 我们要访问的到底是哪个)? 第二, 谁来传送具体的数据信息(这里的数据信息包括指令和数据, 指令也以二进制数据的形式进行存储)? 第三, 要进行读操作还是写

操作(信息是由 CPU 传送给外扩存储器的写操作,还是由外扩存储器传送给 CPU 的读操作)? 此时 P0 口功不可没。它既要充当**低 8 位的地址总线**提供地址信息,又要作为**8 位真正**的**双向数据总线**分时复用传送数据信息。此时,CPU 内部的硬件会自动将控制线设置为“1”,MUX 拨向图 3.5 中上方的位置,与反向器的输出端相连,同时与门应处于开启状态。

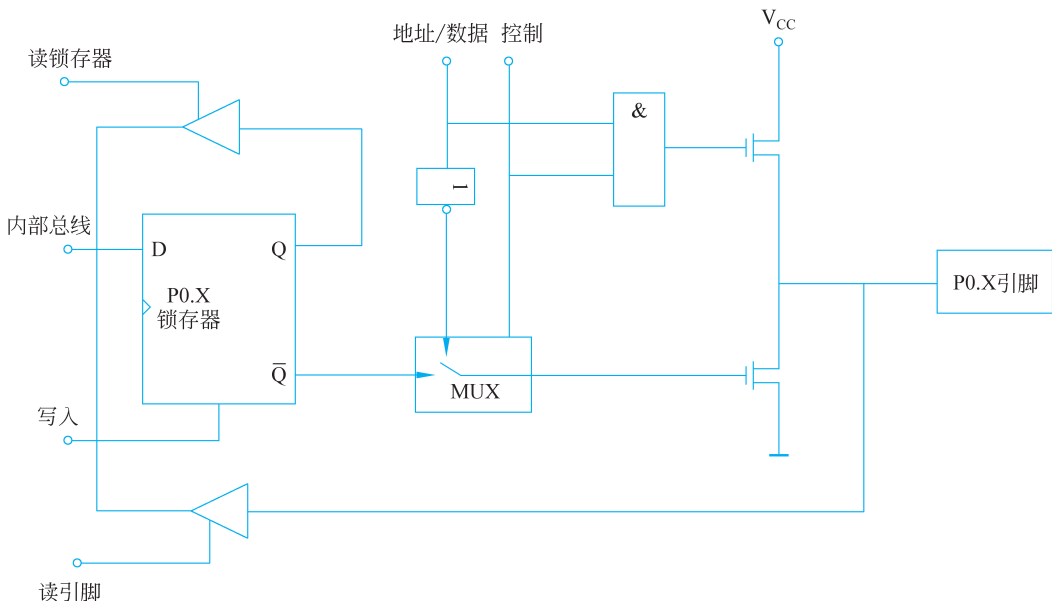


图 3.5 P0 口位结构原理图

由图 3.5 可以看出,右侧两个 FET 以反相形式连接,构成了推挽结构,这种电路结构在一定程度上增加了 P0 口的负载能力。需要输出的地址/数据信息通过与门及非门实现了对这两个 FET 的驱动。例如,当输出地址/数据信息为“0”时,与门的输出为“0”,非门的输出为“1”,这就实现了对上侧 FET 的封锁,而下侧 FET 则处于开启状态,这样由 I/O 引脚输出的信号就为“0”;同理,当输出的地址/数据信息为“1”时,引脚输出的信号为“1”。当输入数据时,信号直接经由引脚通过输入缓冲器进入内部总线。

## 2) P0 口作为一般 I/O 口

P0 口的数据输出通路由锁存器及输出驱动电路构成。在执行输出操作时,CPU 首先将写脉冲加到 D 锁存器的 CL 端,这样锁存器将内部总线上的信息取反后由  $\bar{Q}$  输出,再由输出级 FET 进行取反,最终由引脚输出。但需要注意的是,此时的输出级为漏极开路的形式,如想输出高电平,需要在引脚上外接上拉电阻。

当 P0 口用作输入口时,上侧 FET 处于截止状态,进行输入操作时,“读引脚”指令会打开三态缓冲器,这样输入数据可以经由三态缓冲器进入内部总线。但需要注意,当数据输入时,输入信号在接到三态缓冲器输入端的同时,也接到了下侧 FET 的漏极上。如果锁存器之前锁存过“0”,则此时下侧 FET 就会处于开启状态,这样会使输入的高电平信号被拉为低电平,造成误读。所以,在进行端口输入操作前,应先向锁存器中写入“1”,这样两个 FET 都会处于截止状态,此时端口处于高阻状态,可以避免误读的发生。

上述的读操作是针对引脚上的信息的读取,称为“读引脚”。另外,MCS-51 单片机对 I/O 口还可以进行“读端口”操作。我们应注意二者之间的区别,这两种读操作的信号输入时通过

两个不同的输入缓冲器实现,而且两种读操作的结果可能不同。例如,我们通过端口输出一高电平信息,这时锁存器中的信息应为高电平,而与 I/O 口相连的电路可能将 I/O 口处电平拉低为低电平,如果我们这时分别进行“读引脚”和“读端口”操作,就会得到不同的结果。

## 2. P1 口

图 3.6 给出了 P1 口中某一位的内部电路结构。从图 3.6 可以看出,它是由锁存器、输入缓冲器、FET 及内部上拉电阻组成的。对比图 3.5,P1 口与 P0 口在内部结构上存在一定区别,P1 口内部缺少 MUX 及相关控制电路,且输出级是由 FET 和上拉电阻组成的。

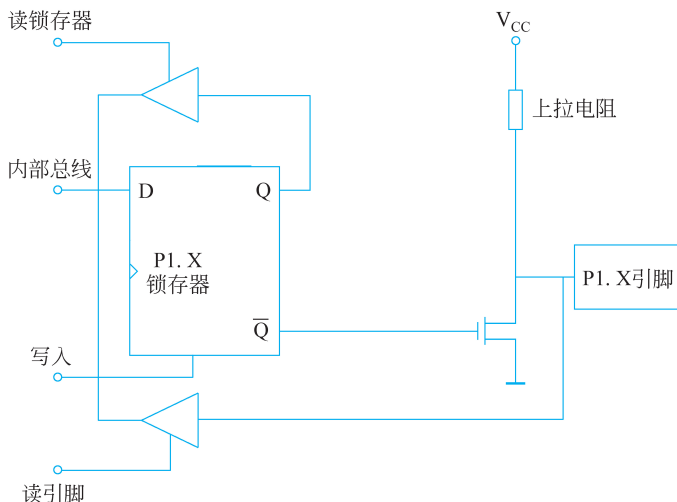


图 3.6 P1 口位结构原理图

P1 口仅用作一般 I/O 口使用,其输出电路仅由上拉电阻和场效应管组成,不能实现三态输出,所以 P1 口为准双向口。但由于其输出电路部分通过上拉电阻与电源相连,所以在输出时无须外接上拉电阻。但当它进行输入操作时,也需要先向锁存器写“1”,否则也会出现误读。

## 3. P2 口

图 3.7 给出了 P2 口中某一位的内部电路结构。从图 3.7 可以看出,P2 口的输出电路与 P1 口类似,同时它还具有与 P0 口类似的 MUX 及相关控制电路,所以 P2 口除了具有一般

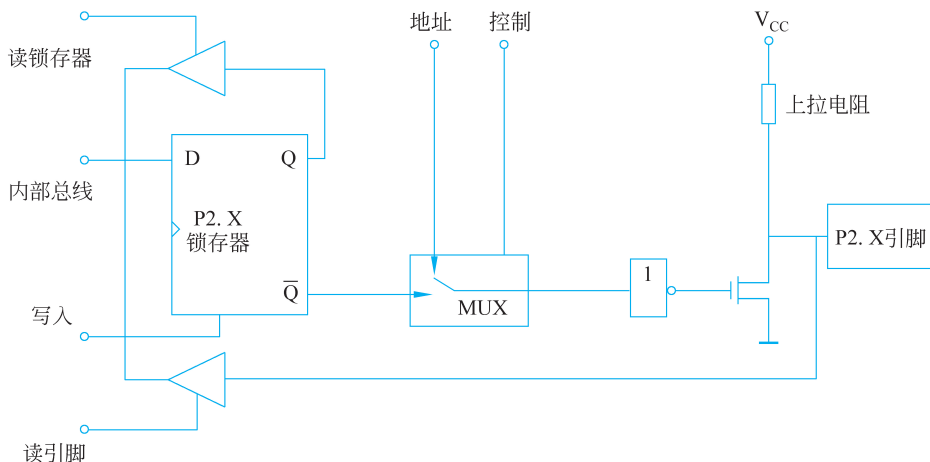


图 3.7 P2 口位结构原理图

I/O 口的功能外,还可以作为系统高 8 位地址总线使用。当 MUX 的开关接到下方时,P2 口作为一般 I/O 口使用,当 MUX 的开关接到上方时,**P2 口作为系统高 8 位地址总线使用**。

当 P2 口作为一般 I/O 口使用时,与 P1 口的功能相同;当 P2 口作为地址总线使用时,它输出高 8 位的地址信息,与 P0 口提供的低 8 位地址信息共同组成 MCS-51 单片机 16 位地址信息。

#### 4. P3 口

P3 口为多功能端口,除了可作为一般 I/O 口使用,**P3 口的各引脚还有第二功能**,图 3.8 给出了 P3 口中某一位的内部电路结构。P3 口在结构上与 P1 口存在的不同主要在于加入了一个与非门和一个缓冲器。通过加入与非门,我们可以对 P3 口的输出信号的性质进行设定,当“第二功能输出”为“1”时,与非门的输出为锁存器的 Q 端信号,此时 P3 口作为一般 I/O 口进行输出;当我们通过设置使 Q 端输出为“1”时,与非门的输出即为第二功能输出,此时 P3 口作为第二功能输出口使用。

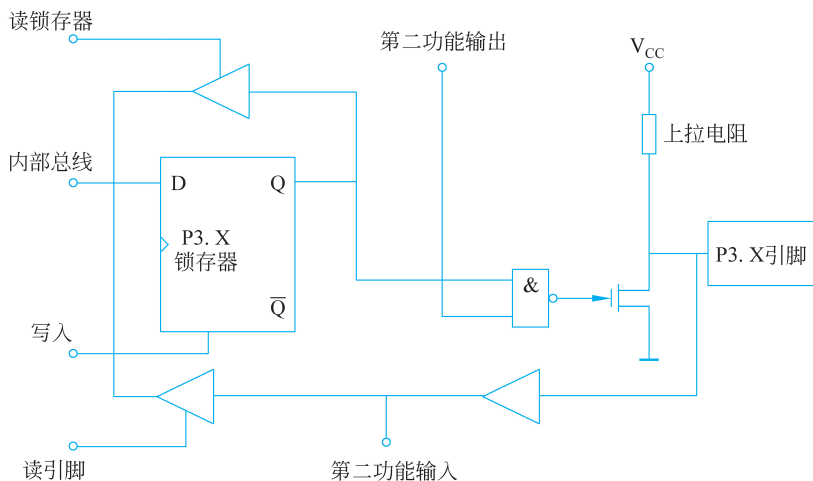


图 3.8 P3 口位结构原理图

当 P3 口作为一般 I/O 口使用时,其功能与 P1 口相同,P3 口的第二功能定义如表 3.8 所示。

表 3.8 P3 口的第二功能定义

P3 口引脚	引脚第二功能
P3.0	RXD 串行输入端口
P3.1	TXD 串行输出端口
P3.2	$\overline{\text{INT0}}$ 外部中断 0 输入端
P3.3	$\overline{\text{INT1}}$ 外部中断 1 输入端
P3.4	T0 定时器 0 外部输入端
P3.5	T1 定时器 1 外部输入端
P3.6	$\overline{\text{WR}}$ 外部数据存储器写选通输出端
P3.7	$\overline{\text{RD}}$ 外部数据存储器读选通输出端

### P0~P3 口功能总结如下。

(1) P0 口：作为并行 I/O 口，支持数据输入输出。其独特之处在于同时用作数据总线和低 8 位地址总线，通过 MUX(多路选择器)实现地址与数据的复用。它是唯一的真正双向口，其输出缓冲器采用三态门设计，确保数据传输时内外电路连通，非传输时则隔离，以增强数据传输的可靠性。

(2) P1 口：标准的并行 I/O 口，用于数据输入输出。它不参与地址总线构建，无 MUX，且为准双向口设计。

(3) P2 口：同样是并行 I/O 口，但主要用于提供系统的高位地址线。它不涉及数据复用，仅接收地址信号，也为准双向口设计。

(4) P3 口：除了作为并行 I/O 口外，还具有独特的第二功能，为系统提供控制信号。因此，在 P3 口电路中增加了额外的控制逻辑以支持这些功能，这是它与 P0、P1、P2 口的主要区别。它同样是准双向口设计。

总结：P0~P3 口均为 MCS-51 单片机的并行 I/O 口。除此之外，P0 口因其数据总线和低 8 位地址总线信息复用，以及真正的双向性而特殊；P2 口用于高 8 位地址总线信息提供；P3 口具备控制信号功能，增加了系统的灵活性。P0 口是真正的双向口，P1~P3 口均为准双向口。

## 3.3 MCS-51 单片机的时钟电路及复位电路

### 本节重点、难点及知识脉络

#### 思考：

1. 单片机包含众多模块，功能及构造各不相同，如何让它们按照统一步伐协调工作？
2. 单片机内部功能模块的初始状态是怎样的？单片机在工作工程中如果受到干扰出现故障而不受控时怎么办？

#### 重点：

1. 时钟电路搭建(内部时钟、外部时钟、时钟发生器)。
2. 复位功能及复位电路的设计(上电复位、手动复位)。
3. 几个周期的概念(时钟周期、机器周期、指令周期)。

#### 难点：

MCS-51 单片机的工作时序(CPU 取指、执行指令的时序)。

### 开启“MCS-51 单片机的时钟电路及复位电路”学习旅程

#### 3.3.1 时钟电路

时钟在单片机中扮演着至关重要的角色，可以类比成人类的心跳。心跳是维持人体生命活动的基本节奏，它确保了血液循环、氧气输送和废物排出的有序进行。同样地，时钟在单片机中也是一个不可或缺的“生命节奏”，它驱动着单片机内部的各种操作按照预定的时序和频率执行。

- (1) **节奏控制**：就像心跳为人体提供了稳定的节奏一样，时钟为单片机提供了稳定的