

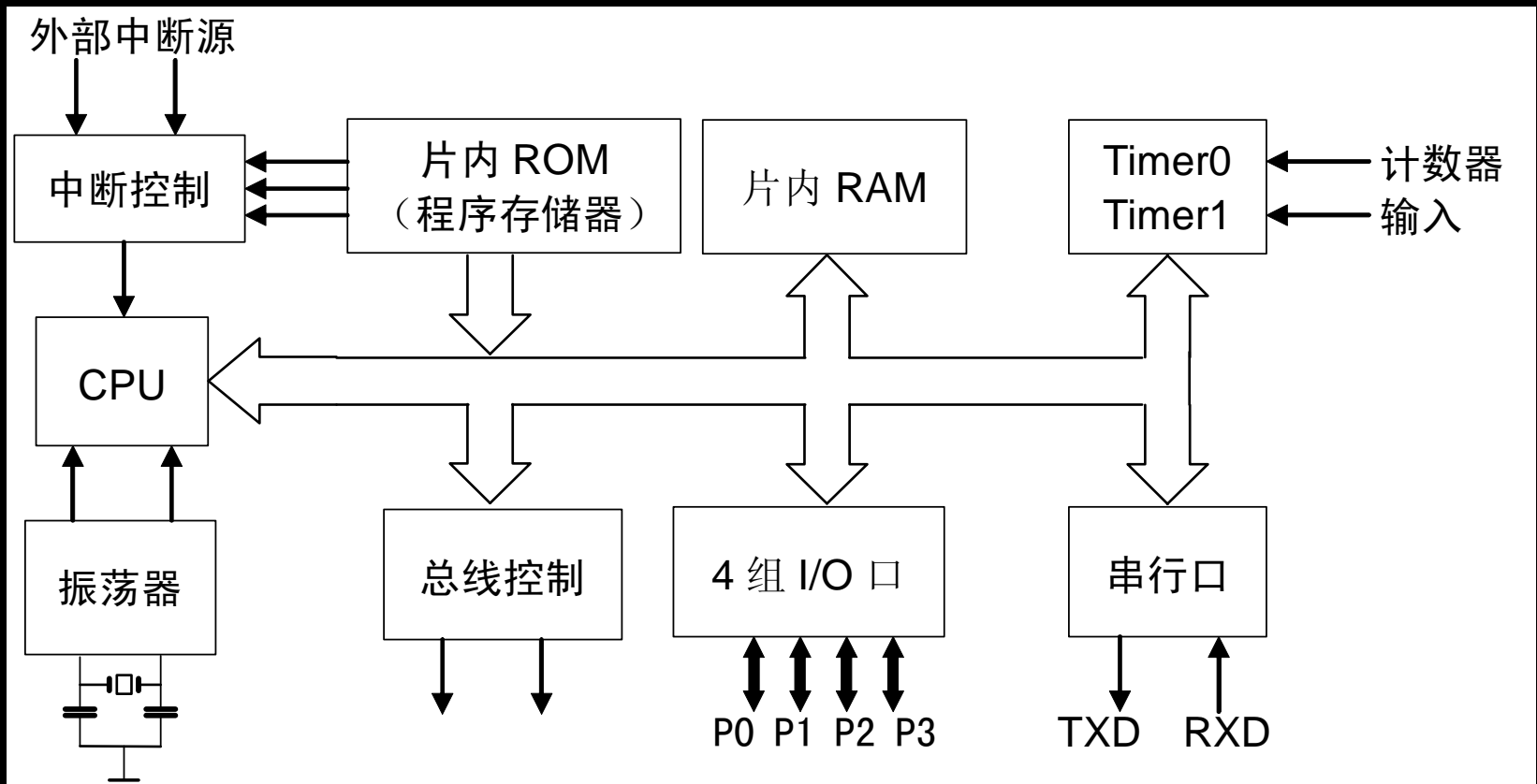
# 第6章 探寻单片机内部 ——存储器组织

欢迎访问 电路飞翔网

<http://www.circuitfly.com> 获取更多信息

- 6.1 单片机的程序存储器
- 6.2 单片机的数据存储器
- 6.3 单片机的特殊功能寄存器SFR
- 6.4 实例点拨——用取表方式控制流水灯

- Intel公司首推的8051单片机有128 Bytes的片内RAM、4K Bytes的片内ROM、两个Timer、1个串行口和4组I/O口。



8051单片机的内部结构框图

# 6.1 单片机的程序存储器

- 6.1.1 单片机的程序下载到哪里

程序经过成功地汇编后，就会生成一个**.HEX**的文件（十六进制代码），十六进制代码很容易就转换成二进制代码，按一定的顺序（地址）下载到单片机的程序存储器（**program memory**）中。

**AT89S51**单片机中的程序存储器就是一个**FLASH**存储器。

除了使用**FLASH**存储器作为程序存储器外，**PROM**、**EPROM**等存储器也用在一些单片机中。

还有一些单片机甚至连程序存储器都没有，这种单片机称为**ROMless**单片机，必须通过读取外部的程序存储器中的程序才能正常使用。

# 6.1 单片机的程序存储器

## • 6.1.2 AT89S51单片机的程序存储器

4K Bytes可在线编程的FLASH存储器，可用0000H~0FFFH来指向，如果把这1000H个字节的存储空间比喻成一幢大楼，大楼的楼层可表示为：0000H层~0FFFH层，共1000H层。

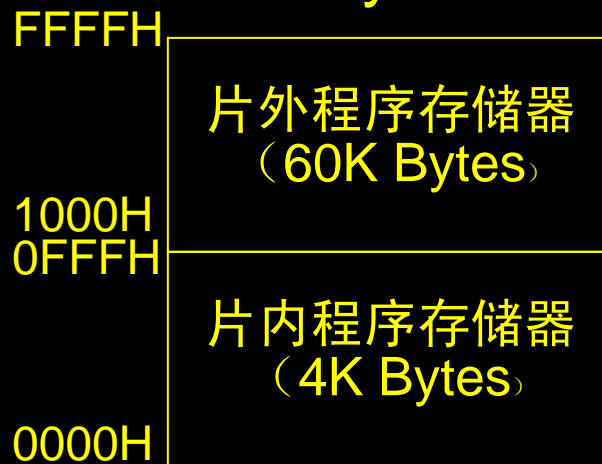
0FFFH	1	1	0	0	1	0	1	1
0FFEH	0	0	1	1	0	1	0	1
...	.....							
...								
0002H	1	1	1	0	0	0	1	1
PC → 0001H	1	0	0	0	0	0	0	1
0000H	0	0	0	0	1	1	0	0

## AT89S51单片机的片内程序存储器

# 6.1 单片机的程序存储器

## • 6.1.2 AT89S51单片机的程序存储器(容量)

在单片机中，有一个程序计数器（**program counter, PC**）用于告诉单片机下一条将要执行的指令的地址。程序计数器**PC**是一个两个字节（**16位**）的寄存器，故**AT89S51**单片机最大的寻址范围是**0000H~FFFFH**，共**64K Bytes**。

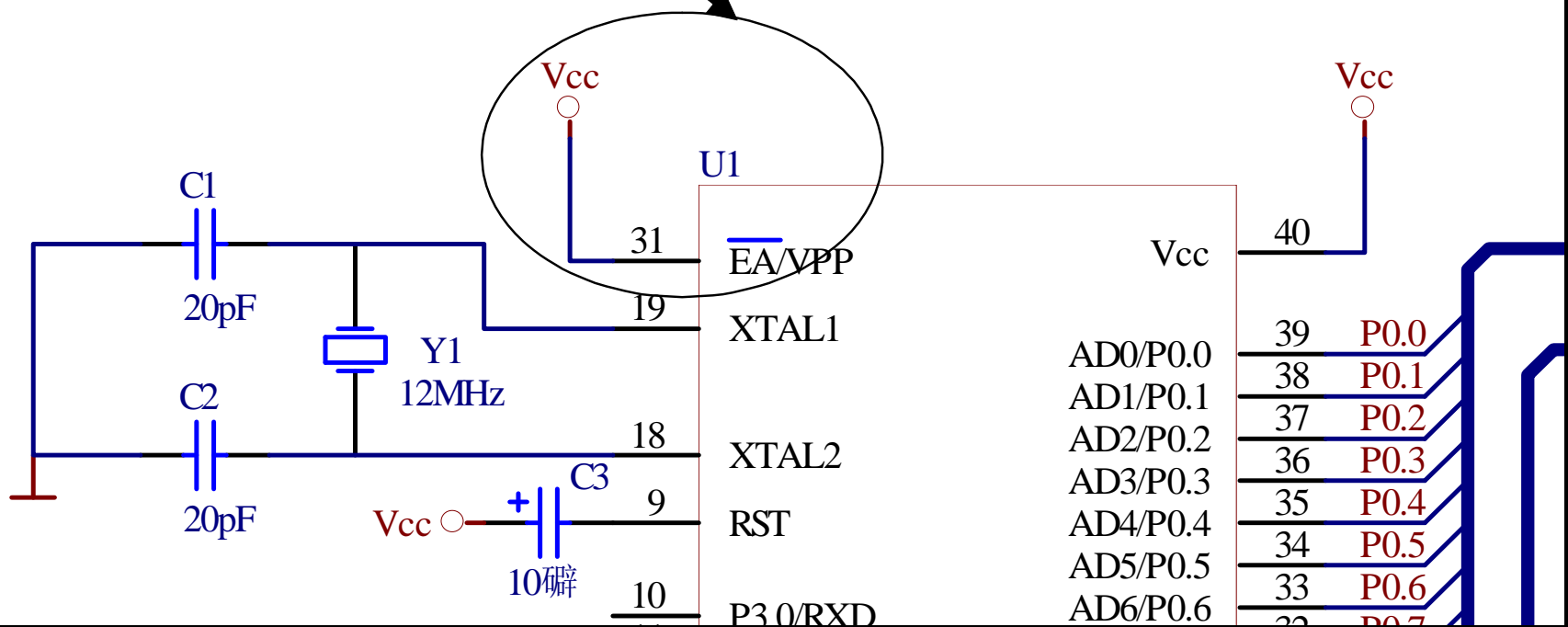


**AT89S51单片机的片内片外程序存储器总容量**

# 6.1 单片机的程序存储器

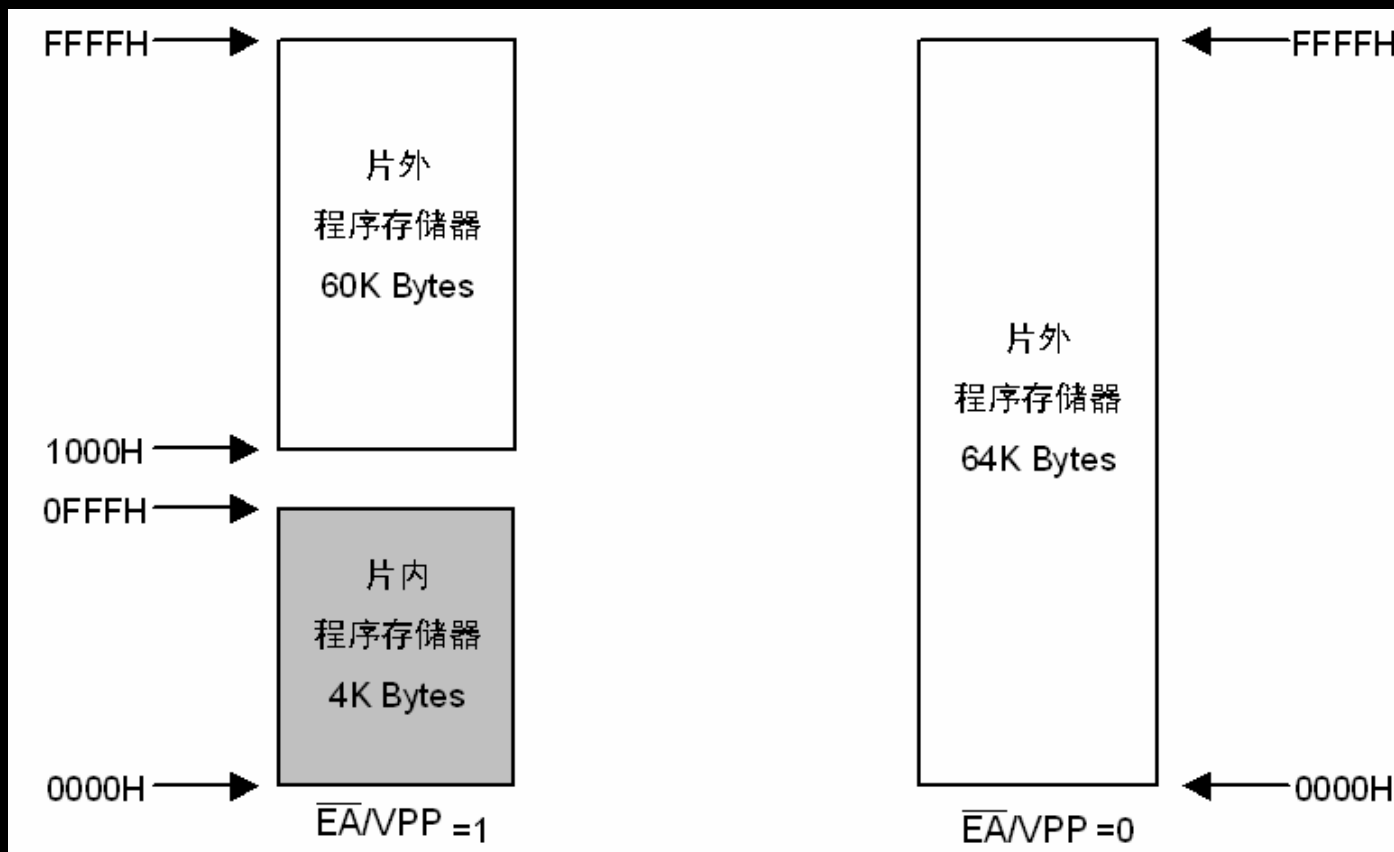
## • 6.1.3 是片内还是片外程序存储器

$\overline{\text{EA/VPP}}$  管脚，接高电平，单片机  
执行片内程序存储器中的程序



# 6.1 单片机的程序存储器

## • 6.1.3 是片内还是片外程序存储器



$\overline{EA}/VPP=0$ 时，全部自动执行片外程序存储器

$\overline{EA}/VPP=1$ 时，先执行片内程序存储器，执行完4K Bytes后，自动执行片外程序存储器。

## 6.2 单片机的数据存储器

- 6.2.1 数据存储器用来作什么

AT89S51单片机提供容量为 $128 \times 8\text{-bit}$ 的片内RAM，用于存储运行时产生的数据的地方，也称这个片内RAM为单片机的数据存储器。

例如指令“MOV 30H, R1”在执行时，工作寄存器R1中的数据载入地址30H中，这30H就是指单片机数据存储器的地址。



## 6.2 单片机的数据存储器

- 6.2.2 片内和片外数据存储器

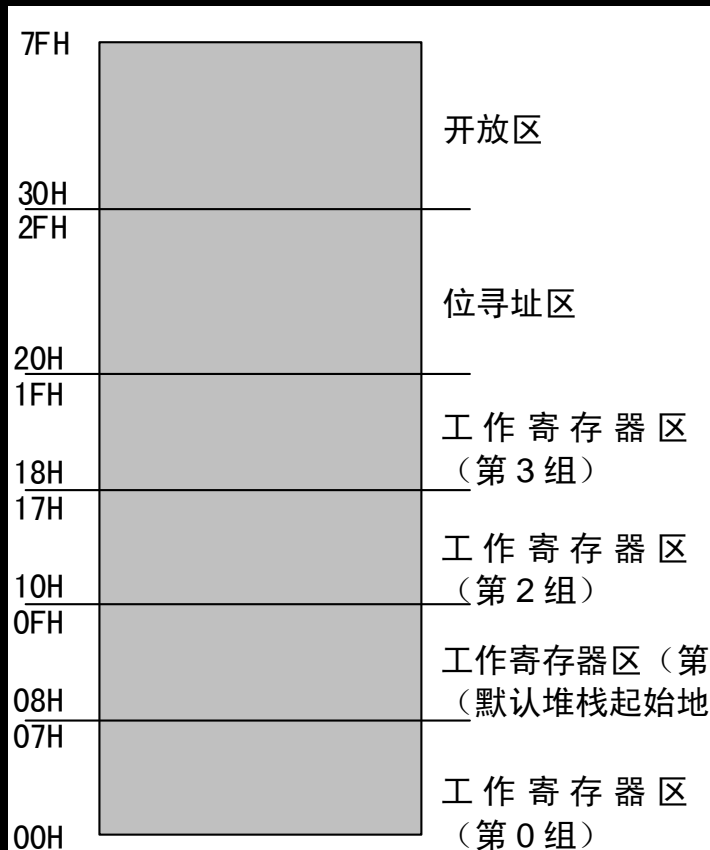
片内数据存储器就是单片机中原有的数据存储器，即片内**RAM**。

片外数据存储器是外部扩展的部分，它可以由**SRAM**来充当。

访问数据存储器时可通过数据指针**DPTR**进行。数据指针**DPTR**是一个长度为两个字节（**16位**）的寄存器。最大的数据存储器寻址范围也为**0000H~FFFFH**。

## 6.2 单片机的数据存储器

### • 6.2.2 片内和片外数据存储器



### AT89S51单片机数据存储器结构 (128×8-bit)

在单片机运行期间，**128×8-bit**的片内数据存储器并不是完全开放给用户使用的，只有其中的**20H~7FH**共**96**个字节（开放区+位寻址区）是给用户使用的，而**00H~1FH**一般先不去直接使用。

## 6.2 单片机的数据存储器

- 6.2.3 00H~1FH——工作寄存器区(组别映射)

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

工作寄存器组别映射的片内数据存储器地址

## 6.2 单片机的数据存储器

- 6.2.3 00H~1FH——工作寄存器区(组别选择)  
更改程序状态字PSW中的第3位(RS0)和第4位RS1)，改变当前程序使用的工作寄存器组别。

程序状态字PSW								当前工作寄存器组别
CY	AC	F0	RS1	RS0	OV	保留	P	
×	×	×	0	0	×	×	×	第0组 (00H~07H)
×	×	×	0	1	×	×	×	第1组 (08H~0FH)
×	×	×	1	0	×	×	×	第2组 (10H~17H)
×	×	×	1	1	×	×	×	第3组 (18H~1FH)

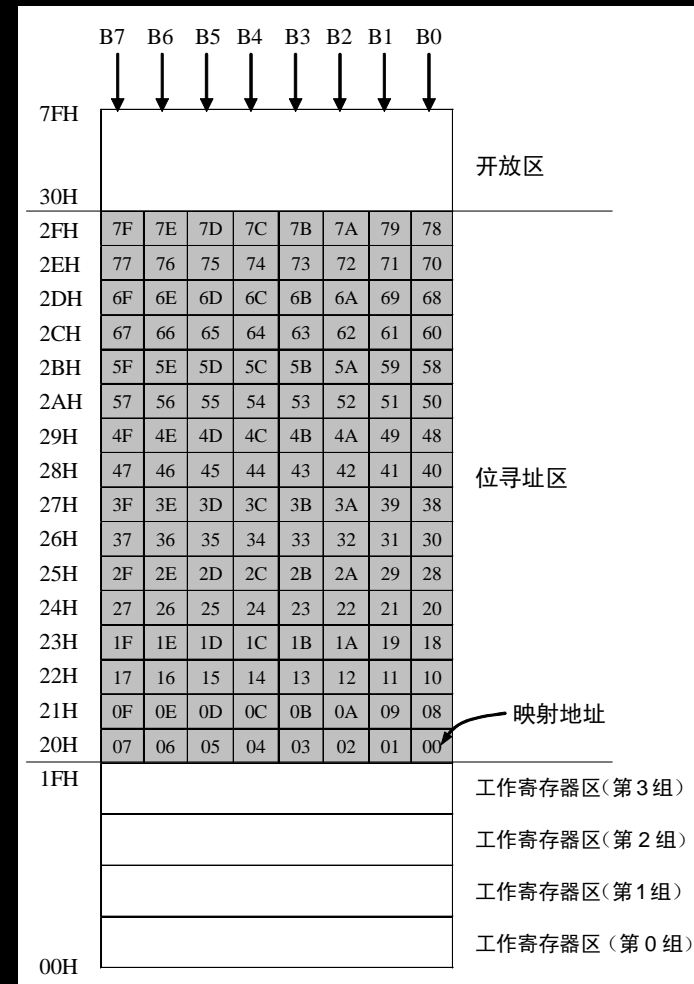
程序状态字PSW与工作寄存器组别

## 6.2 单片机的数据存储器

### • 6.2.4 20H~2FH——位寻址区

片内数据存储器的20H~2FH（共16个字节）为位寻址区，这16个字节共有 $16 \times 8 = 128$ 位可进行位寻址，也就是可以以位为单位进行操作。所用到的指令是SETB——置1；CLR——清0；CPL——取反；ANL——“与”操作等。

20H~2FH位寻址区也可以当成是一般的开放区使用。



每个地址的映射地址

## 6.2 单片机的数据存储器

- 6.2.5 30H~7FH——开放区

30H~7FH是片内数据存储器中开放给用户使用的地址空间。用户可以在这个空间里存储系统运行时产生的数据，也可以读取存储的数据到工作寄存器、累加器等中。

一般来说，AT89S51单片机的开放区可以包括位寻址区20H~2FH，所以，AT89S51单片机开放给用户使用的空间为20H~7FH共96个字节。

## 6.3 单片机的特殊功能寄存器SFR

特殊功能寄存器SFR (special function registers)是用来对单片机内各功能模块进行管理、控制的控制寄存器，也是指示单片机运行状态的状态寄存器，它是一个具有特殊功能的RAM区。

累加器ACC，程序状态字PSW，I/O口P0~P3都在特殊功能寄存器中。

其地址映射空间紧接着片内数据存储器(00H~7FH)为80H~0FFH。

# 6.3 单片机的特殊功能寄存器SFR

## • 6.3.1 特殊功能寄存器SFR分布图

0F8H									0FFH
0F0H	B								0F7H
0E8H									0EFH
0E0H	ACC								0E7H
0D8H									0DFH
0D0H	PSW								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP								0BFH
0B0H	P3								0B7H
0A8H	IE								0AFH
0A0H	P2		AUXR1				WDTRST		0A7H
98H	SCON	SBUF							9FH
90H	P1								97H
88H	TCON	TMOD	TL0	TL1	TH0	TH1	AUXR		8FH
80H	P0	SP	DP0L	DP0H	DP1L	DP1H		PCON	87H

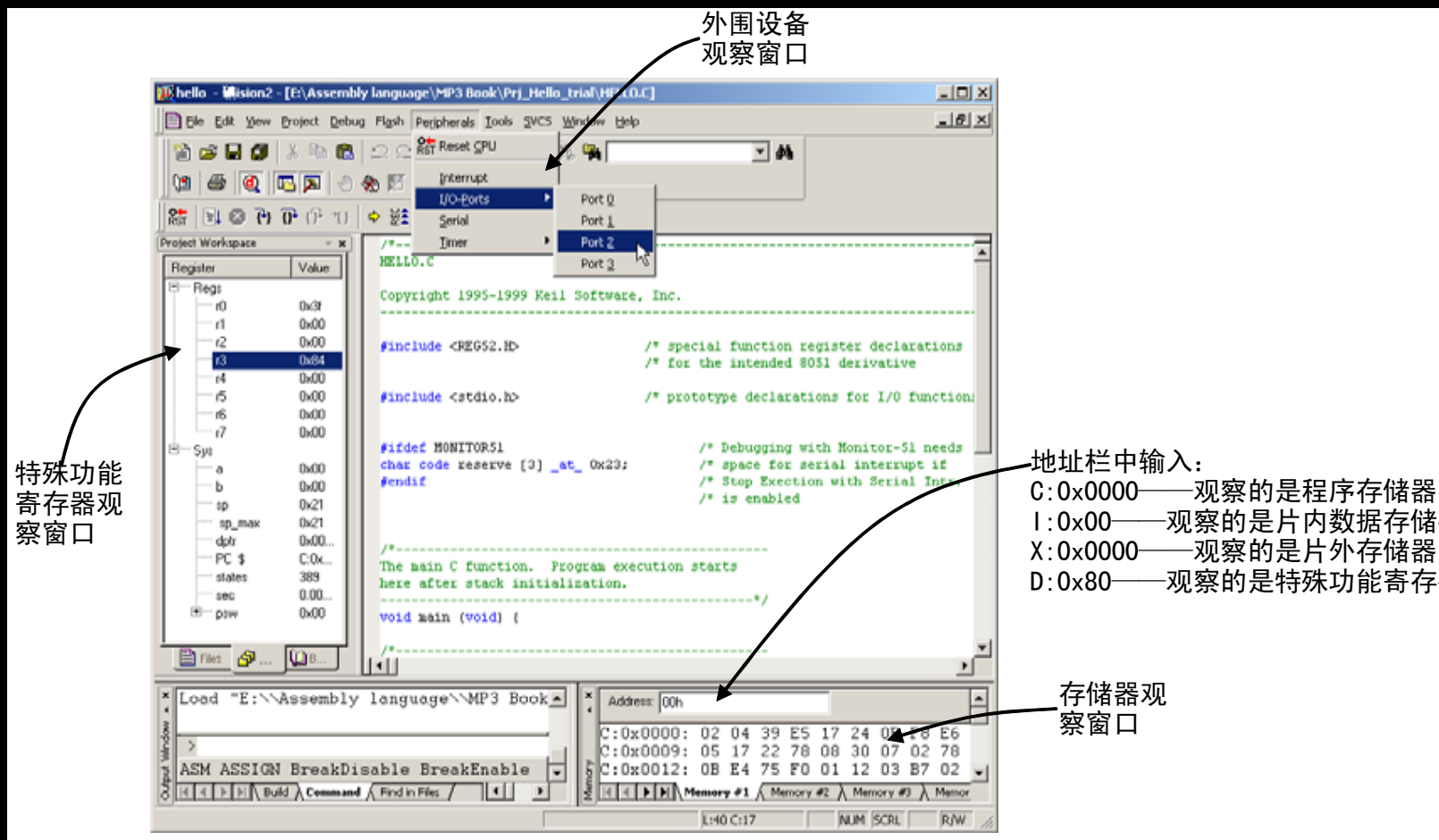
没有被占用的地址有可能在单片机中不存在。

## AT89S51单片机特殊功能寄存器分布图



# 6.3 单片机的特殊功能寄存器SFR

## • 6.3.1 特殊功能寄存器SFR分布图



## μVision中的特殊功能寄存器及外围设备观察窗口

# 6.3 单片机的特殊功能寄存器SFR

## • 6.3.2 特殊功能寄存器的操作方法

### ——字节操作与位操作

AT89S51单片机中支持位操作的特殊功能寄存器是：

- ✓P0（P0口锁存器）
- ✓TCON（定时/计数器控制寄存器）
- ✓TMOD（定时/计数器模式控制寄存器）
- ✓P1（P1口锁存器）
- ✓SCON（串行口控制寄存器）
- ✓P2（P2口锁存器）
- ✓IE（中断使能寄存器）
- ✓P3（P3口锁存器）
- ✓IP（中断优先控制寄存器）
- ✓PSW（程序状态字寄存器）
- ✓ACC（累加器）
- ✓B（B寄存器）

字节操作  MOV P1,#00H

位操作  CLR P0.0  
CLR P0.1  
CLR P0.2  
CLR P0.3  
CLR P0.4  
CLR P0.5  
CLR P0.6  
CLR P0.7

## 6.3 单片机的特殊功能寄存器SFR

### • 6.3.3 特殊功能寄存器简介

AT89S51单片机的特殊功能寄存器共有26个。

- ✓ P0 (80H) ——P0口锁存器
- ✓ SP (81H) ——堆栈指针
- ✓ DPxL和DPxH (82H~85H) ——数据指针 (x代表0或1)
- ✓ PCON (87H) ——电源控制寄存器
- ✓ TCON (88H) ——定时/计数器控制寄存器
- ✓ TOMD (89H) ——定时/计数器模式控制寄存器
- ✓ TL0、TL1、TH0、TH1 (8A~8DH) ——Timer寄存器

## 6.3 单片机的特殊功能寄存器SFR

### • 6.3.3 特殊功能寄存器简介

AT89S51单片机的特殊功能寄存器共有26个。

- ✓ AUXR (8EH) ——辅助寄存器
- ✓ P1 (90H) ——P1口锁存器
- ✓ SCON (98H) ——串行口控制寄存器
- ✓ SBUF (99H) ——串口缓冲寄存器
- ✓ P2 (0A0H) ——P2口锁存器
- ✓ AUXR1 (0A2H) ——辅助寄存器1
- ✓ WDTRST (0A6H) ——看门狗复位控制端

## 6.3 单片机的特殊功能寄存器SFR

### • 6.3.3 特殊功能寄存器简介

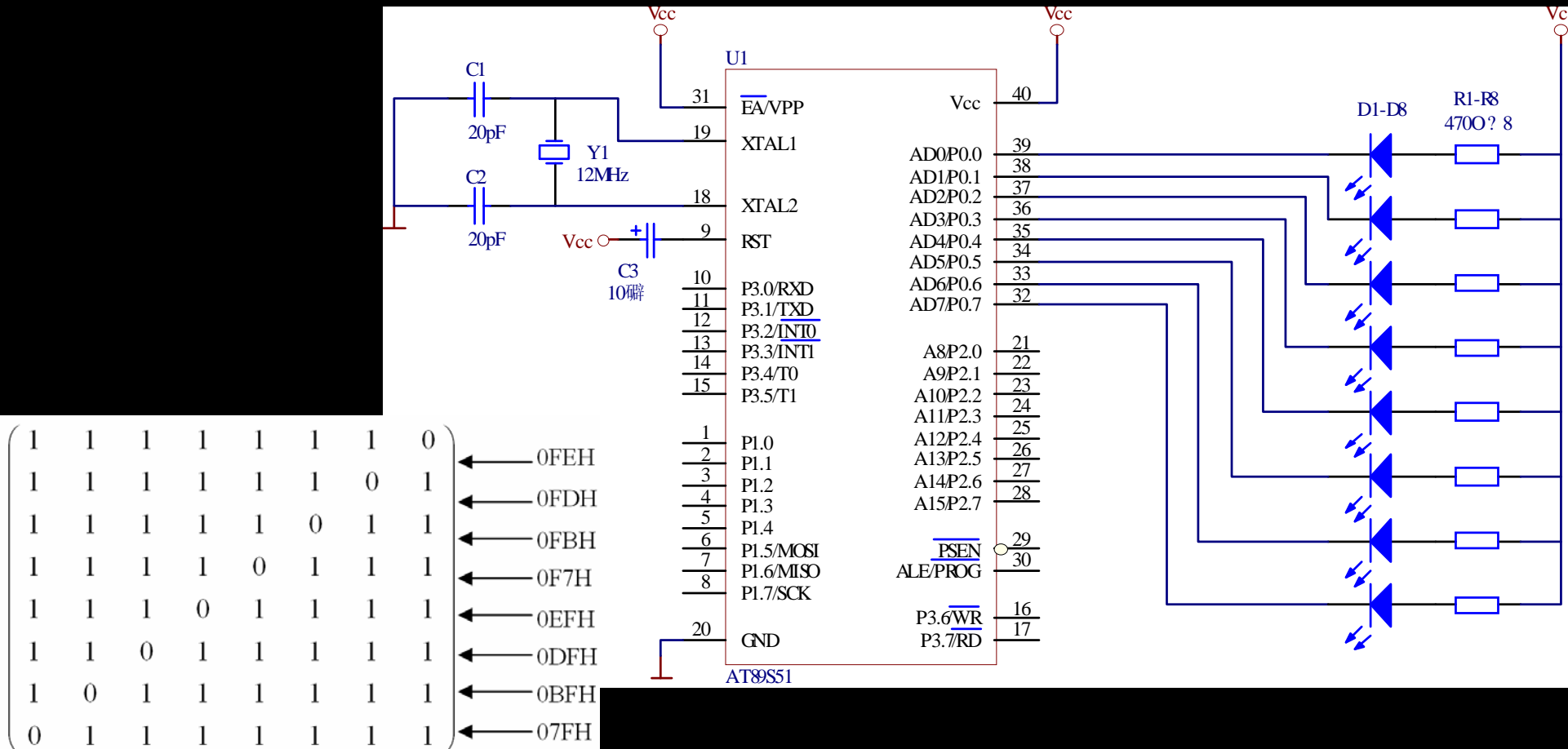
AT89S51单片机的特殊功能寄存器共有26个。

- ✓ IE (0A8H) ——中断使能寄存器
- ✓ P3 (0B0H) ——P3口锁存器
- ✓ IP (0B8H) ——中断优先级控制寄存器
- ✓ PSW (0D0H) ——程序状态字
- ✓ ACC (0E0H) ——累加器
- ✓ B (0F0H) ——B寄存器

## 6.4 实例点拨

# ——用取表方式控制流水灯

### • 6.4.1 流水灯的显示数据转换



## 6.4 实例点拨

# ——用取表方式控制流水灯

### • 6.4.2 取表法

数据存在依次取用的特点，一般用取表的方法。

在汇编程序末尾附加一个数据表，这个数据表可以是字节型的数据，也可是英文字母等数据。数据表与汇编程序一起被汇编，数据表就成为程序的一部分包含在生成的.HEX文件中。

```
TABLE:                                ;本程序的数据表
DB      0FEH, 0FDH, 0FBH, 0F7H        ; 左移
DB      0EFH, 0DFH, 0BFH, 07FH
DB      07FH, 0BFH, 0DFH, 0EFH        ; 右移
DB      0F7H, 0FBH, 0FDH, 0FEH
DB      88H                            ; 结束码
```

## 6.4 实例点拨

### ——用取表方式控制流水灯

#### • 6.4.2 取表法

当程序下载到单片机后，指令的执行代码和数据表中的数据一起保存在程序存储器中，存在存储器的某一段地址空间中。而数据表的名称**TABLE**就像一个指针，指向第一个数据**0FEH**所在的地址。

程序存储器地址	程序存储器中的指令代码																
0000H	90	00	1F	E4	93	B4	01	02	-	80	F6	F5	80	7B	14	12	00
0010H	14	A3	80	EF	7C	14	7D	F8	-	DD	FE	DC	FA	DB	F6	22	FE
0020H	FD	FB	F7	EF	DF	BF	7F	7F	-	BF	DF	EF	F7	FB	FD	FE	88

TABLE

数据表中的数据依次排列在程序存储器



# 6.4 实例点拨

## ——用取表方式控制流水灯

### • 6.4.2 取表法(取表程序介绍)

```

START:                                ;装表, 即 DPTR 指向表 TABLE 的表头
MOV    DPTR, #TABLE                    ; 数据指针寄存器 DPTR 指向表 TABLE 头
LOADTABLE:                             ;取表程序段, 表中的数据载入累加器 ACC 中, 并判断是否已经完成
CLR    A                                ; 累加器 ACC 清 0
MOVC   A, @A+DPTR                      ; 基址加变址寻址, 将表中数据载入 ACC
CJNE   A, #88H, DISPLAY                ; 如果取到 88H 表明表已经取完, 否则显示
JMP    START                           ; 如果取完, 跳回 START 重新开始
DISPLAY:                               ; 显示程序段, 累加器 ACC 的数据输出到 P0
MOV    P0, A                           ; 将 ACC 中数据输出至 P0 口
CALL   DELAY                           ; 调延时子程序
INC    DPTR                             ; DPTR 加 1, 准备取表中的下一个数据
JMP    LOADTABLE                       ; 跳回取表程序段

```

## 6.4 实例点拨

# ——用取表方式控制流水灯

### • 6.4.3 方法总结

取表法的实质是把存放在程序存储器中的数据，通过@A+DPTR指针来寻址，并载入ACC中，再经过指令“MOV P0,A”将数据从I/O口输出。



## 取表法设计思路