

# 《单片机原理及应用》

## 第2章：SoC单片机的结构与原理



2.1 C8051F系列单片机总体结构

2.2 C8051F020存储器组织

2.3 中断系统

2.4 端口输入/输出

2.5 电源管理方式

2.6 复位与时钟



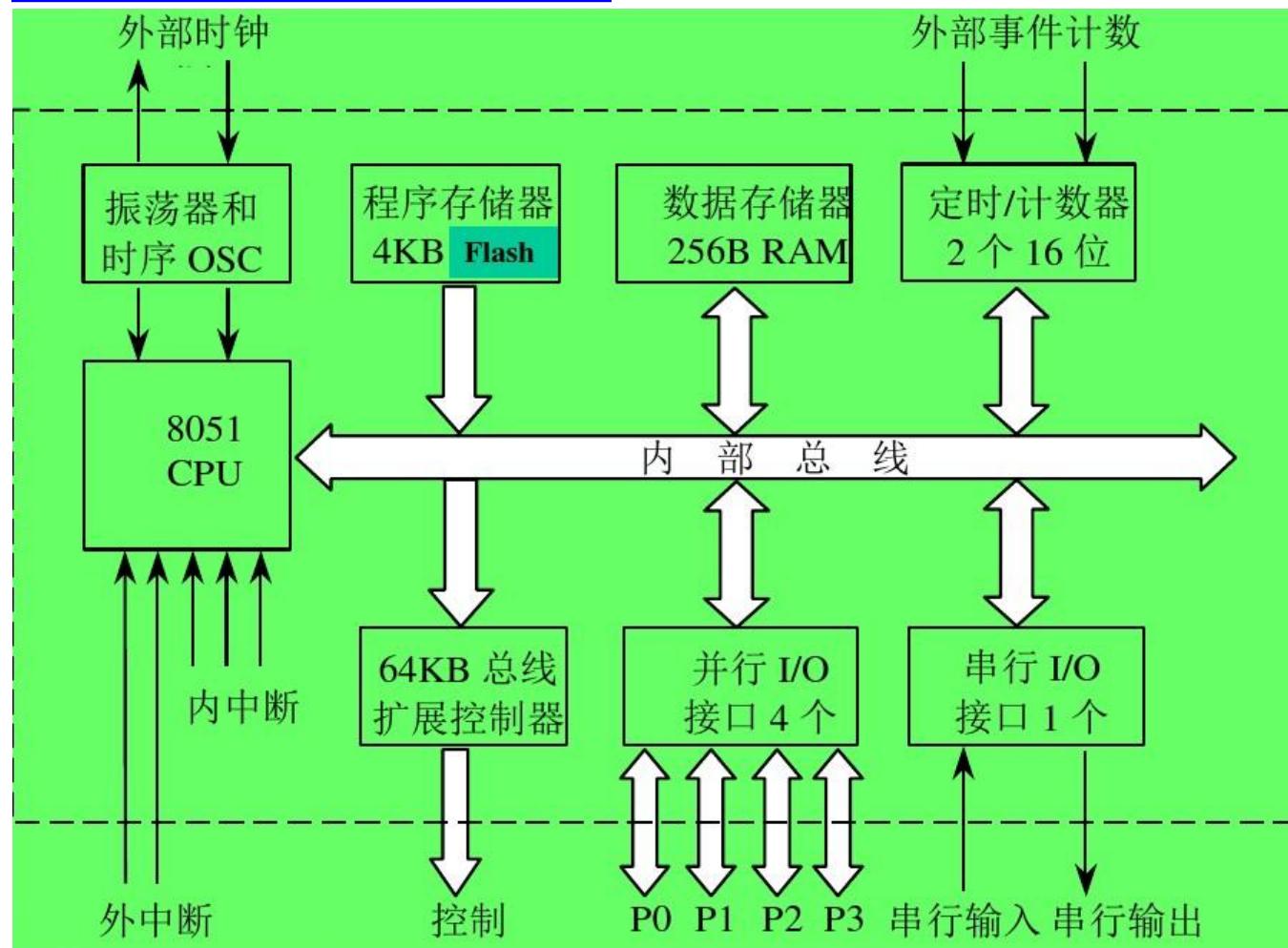
# 2.1 C8051F系列单片机总体结构



100脚的TQFP封装



## AT89C51的基本组成

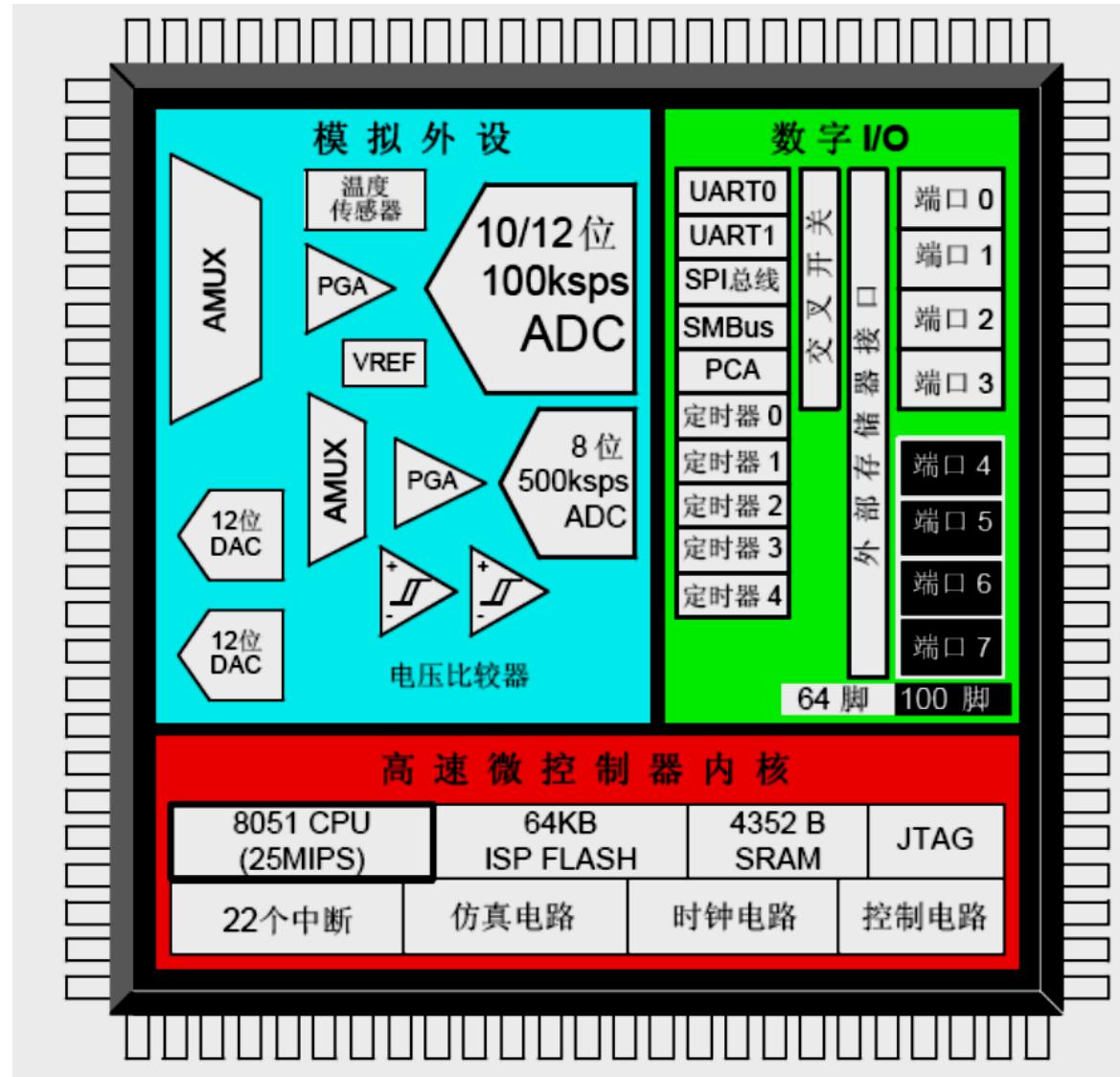


<https://www.bilibili.com/video/BV16741137Ag?p=16>



## • C8051F系列单片机简介

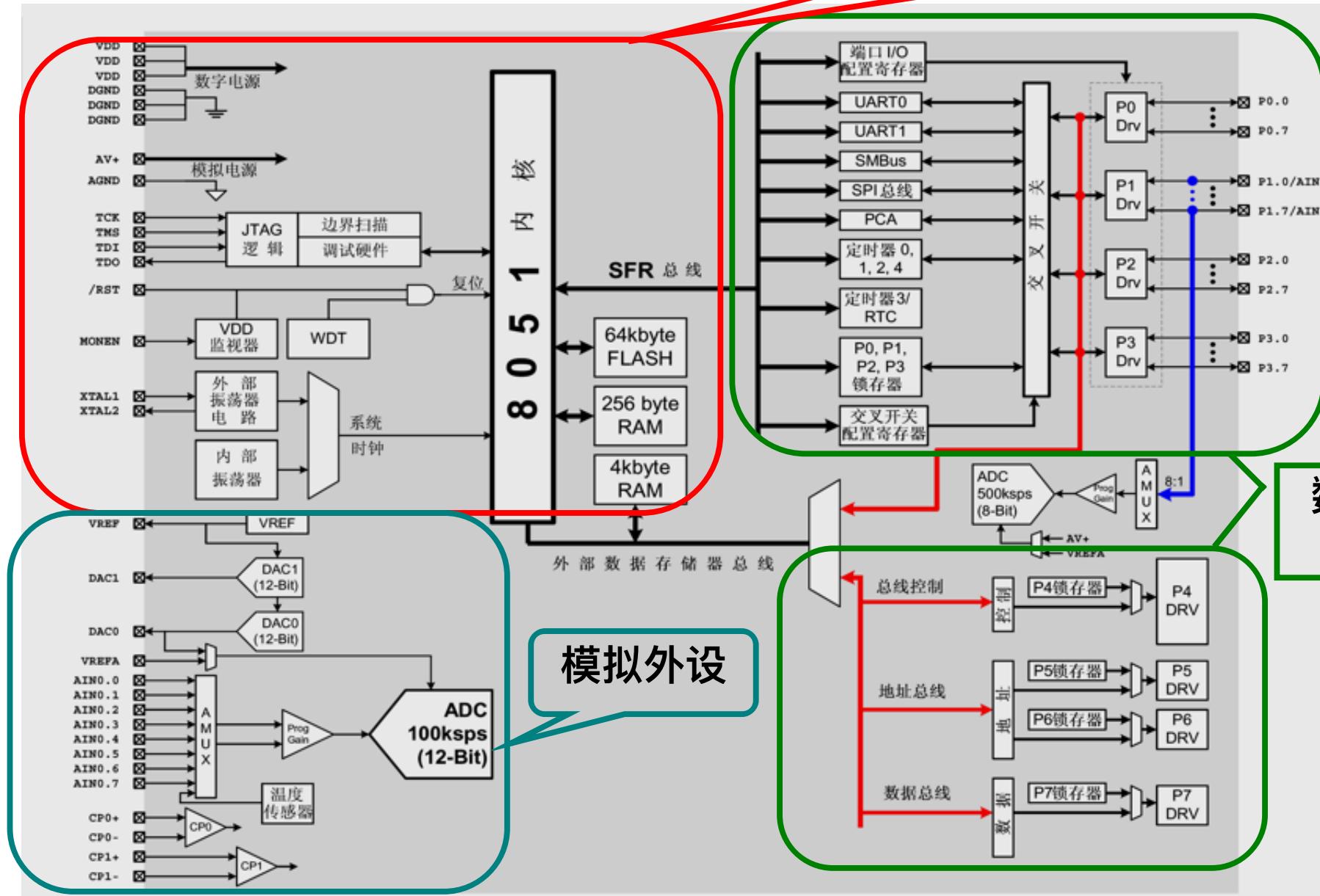
- C8051F系列单片机是集成混合信号片上系统SOC (System on chip)。
- 具有与MCS-51内核及指令集完全兼容的微控制器。
- 除了具有标准8051的数字外设部件之外，片内还集成了数据采集和控制系统中常用的模拟部件和其他数字外设及功能部件。
- 由原来美国Cygnal公司（已被Silicon Labs公司并购）设计与制造。



# C8051F020原理框图

使用说明书 datasheet

## 高速微控制器内核



# 1 高速微控制器内核



- 高速、流水线结构的8051 兼容的CIP-51 内核（可达25MIPS）
- 22 个矢量中断源
- 64K字节可在系统编程的FLASH 存储器
- 4352 (4096+256) 字节的片内RAM
- 可寻址64K字节地址空间的外部数据存储器接口
- 片内调试电路提供全速、非侵入式的在系统调试接口（不需要仿真器），时钟电路
- 精确的VDD 监视器和欠压检测器
- 专用的看门狗定时器；
- 双向复位引脚



# 2 模拟外设

## ► 12位SAR ADC

±1LSB INL

- ? 可编程转换速率，最大100ksps
- ? 可多达8个外部输入；可编程为单端输入或差分输入？

可编程放大器增益：16、8、4、2、1、0.5

数据相关窗口中断发生器

- ? 内置温度传感器 ( $\pm 3^{\circ}\text{C}$ )

## ► 8位ADC

- ? 可编程转换速率，最大500ksps

- ? 8个外部输入

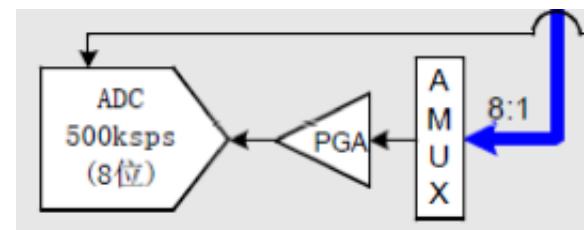
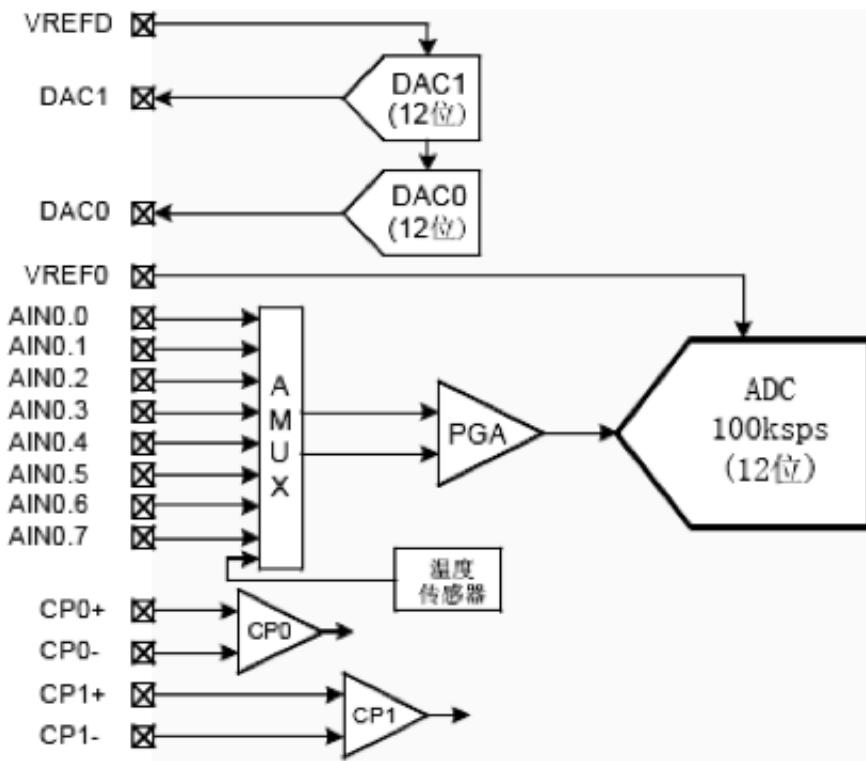
可编程放大器增益：4、2、1、0.5

## ► 两个12位DAC

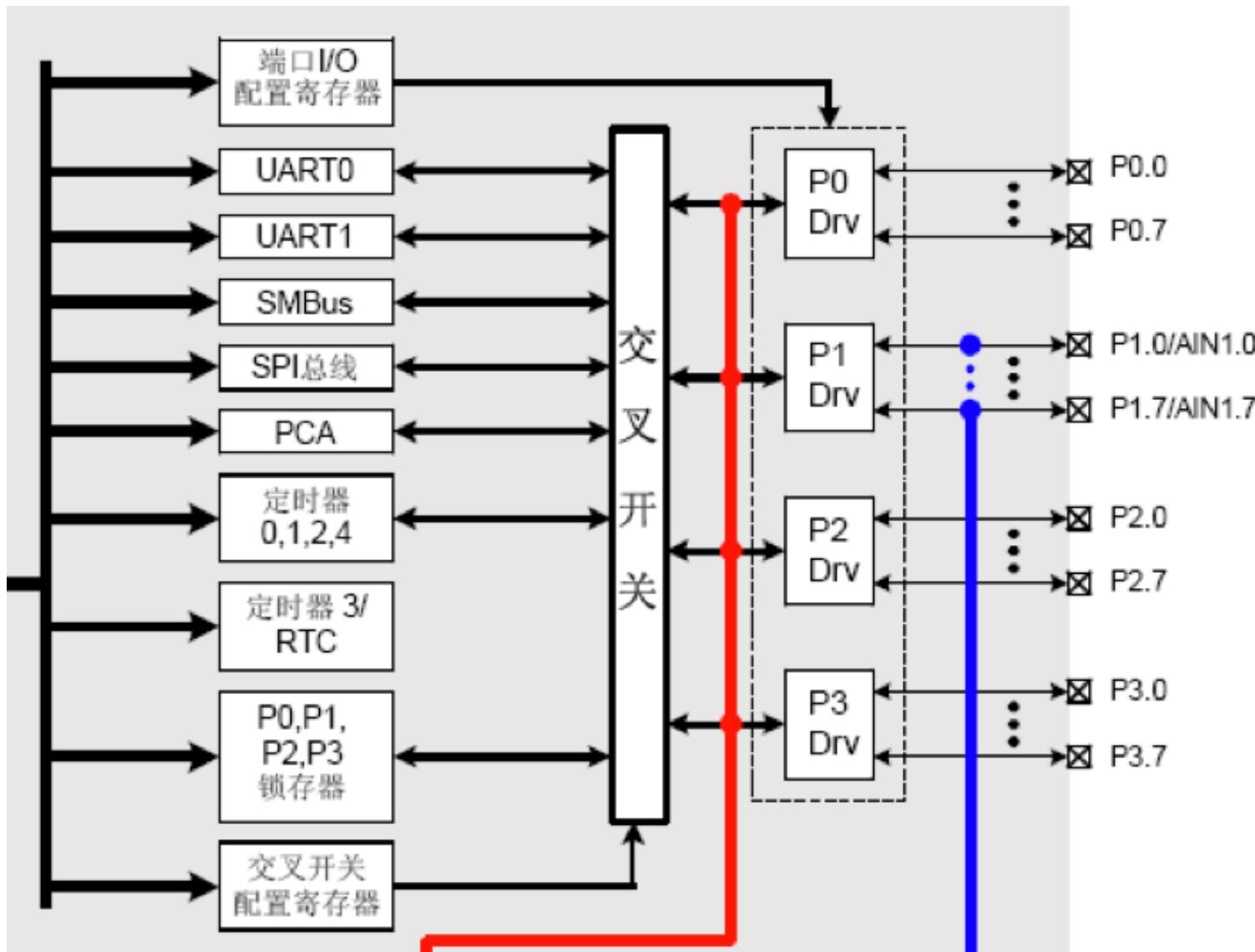
- ? 可以同步输出，用于产生无抖动波形

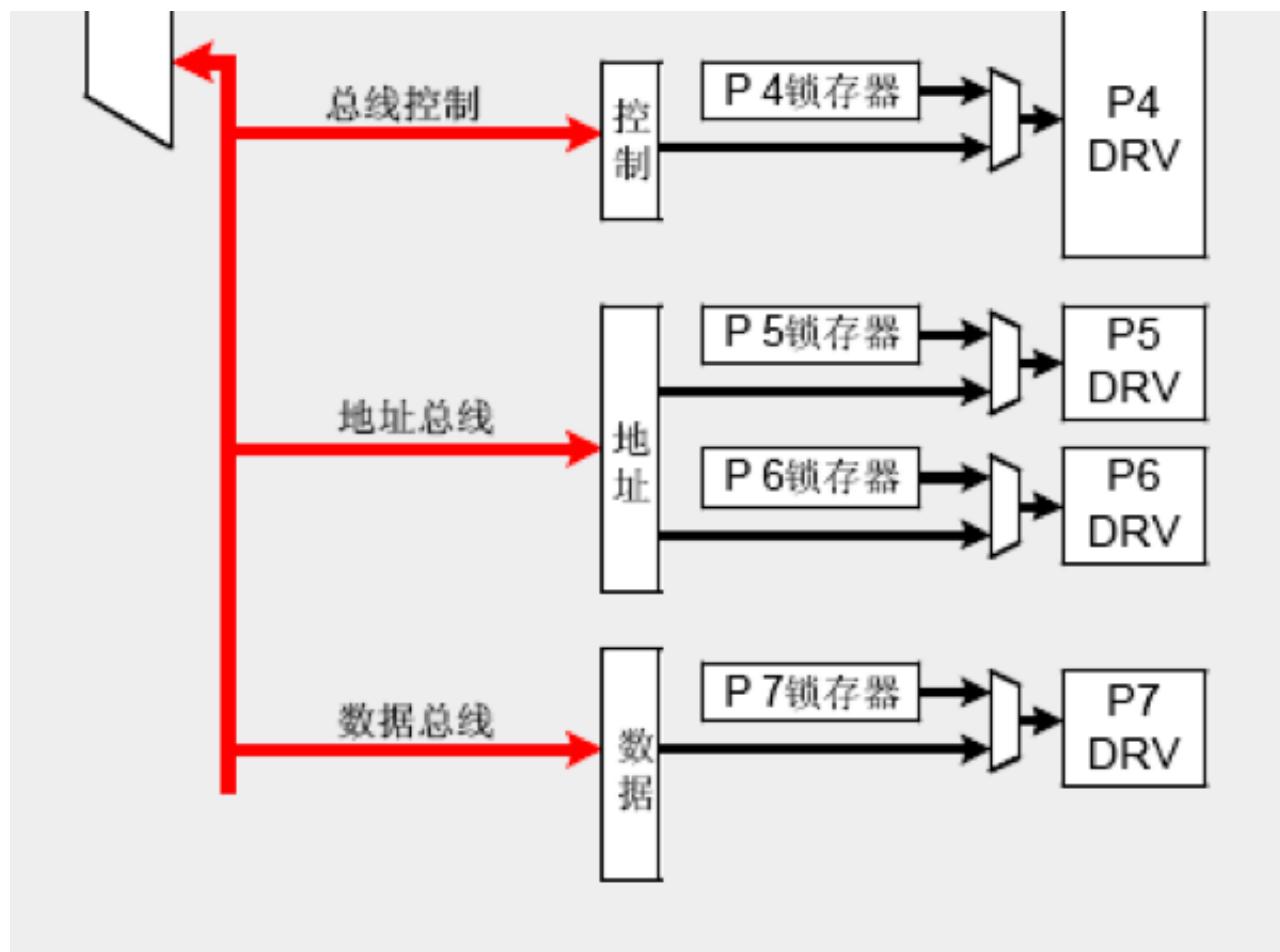
## ► 两个模拟比较器

## ► 电压基准



# 3 数字外设





# 芯片的极限参数

表 2.1 极限参数\*

参 数	条 件	最 小 值	典 型 值	最 大 值	单 位
环境温度（通电情况下）		-55		125	℃
储存温度		-65		150	℃
任何引脚相对DGND的电压（VDD和端口I/O除外）		-0.3		VDD + 0.3	V
任何端口I/O引脚或/RST相对DGND的电压		-0.3		5.8	V
VDD引脚相对DGND的电压		-0.3		4.2	V
通过VDD、AV+、DGND和AGND的最大总电流				800	mA
任何端口引脚的最大输出灌电流				100	mA
任何其它I/O引脚的最大输出灌电流				50	mA
任何端口引脚的最大输出拉电流				100	mA
任何其它I/O引脚的最大输出拉电流				50	mA

\*注：超过这些列出的“极限参数”可能导致器件永久性损坏。



- CIP-51对标准8051的改进

- 采用流水线结构，指令速度大大提高

CIP-51中，70%指令的执行周期为1或2个系统时钟周期。

标准8051中，除MUL和DIV以外所有指令需要12或24个系统时钟周期

执行周期数	1	2	2/3	3	3/4	4	4/5	5	8
指令数	26	50	5	16	7	3	1	2	1

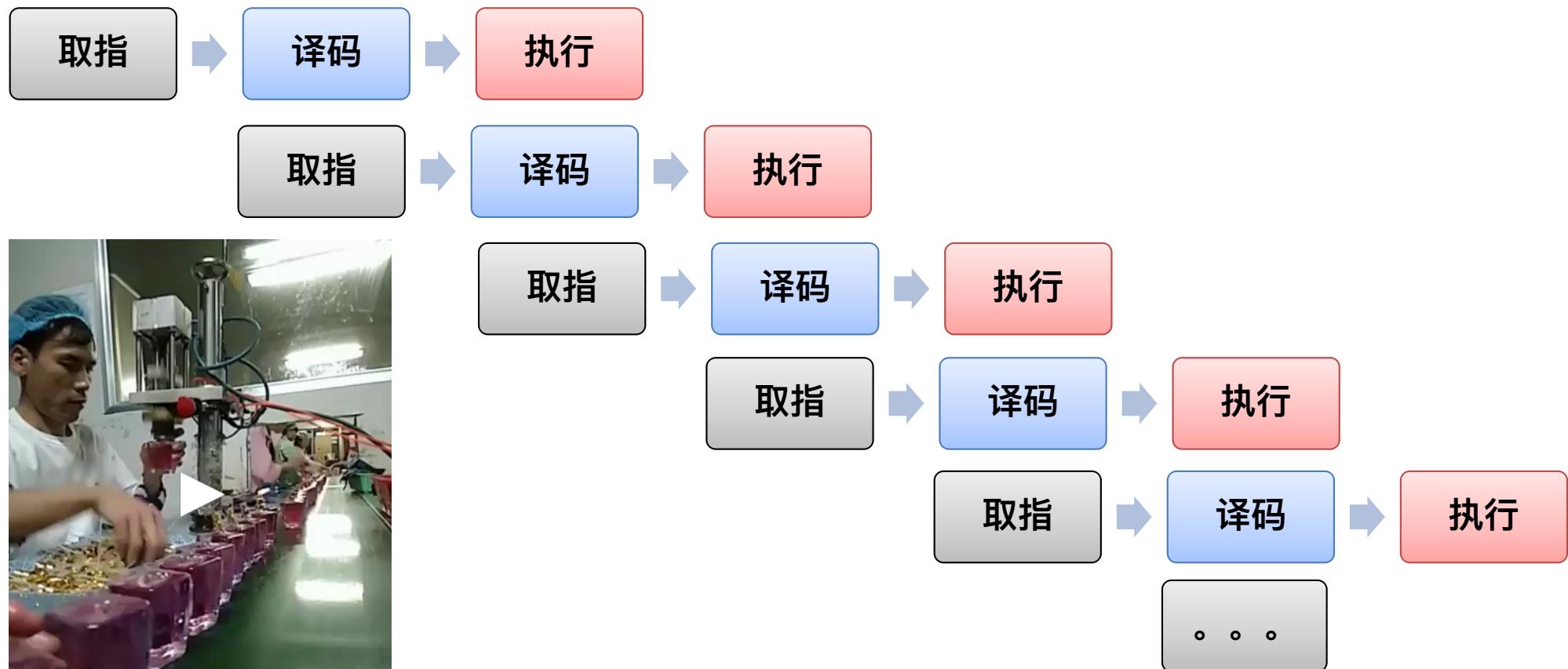
- 扩展的中断系统向CIP-51提供22个中断源，允许大量的模拟和数字外设中断控制器，标准8051只有7个中断源
- MCU可有多达7个复位源：一个片内VDD监视器、一个看门狗定时器、一个时钟丢失检测器、一个由比较器0提供的电压检测器、一个软件强制复位、CNVSTR引脚以及/RST引脚



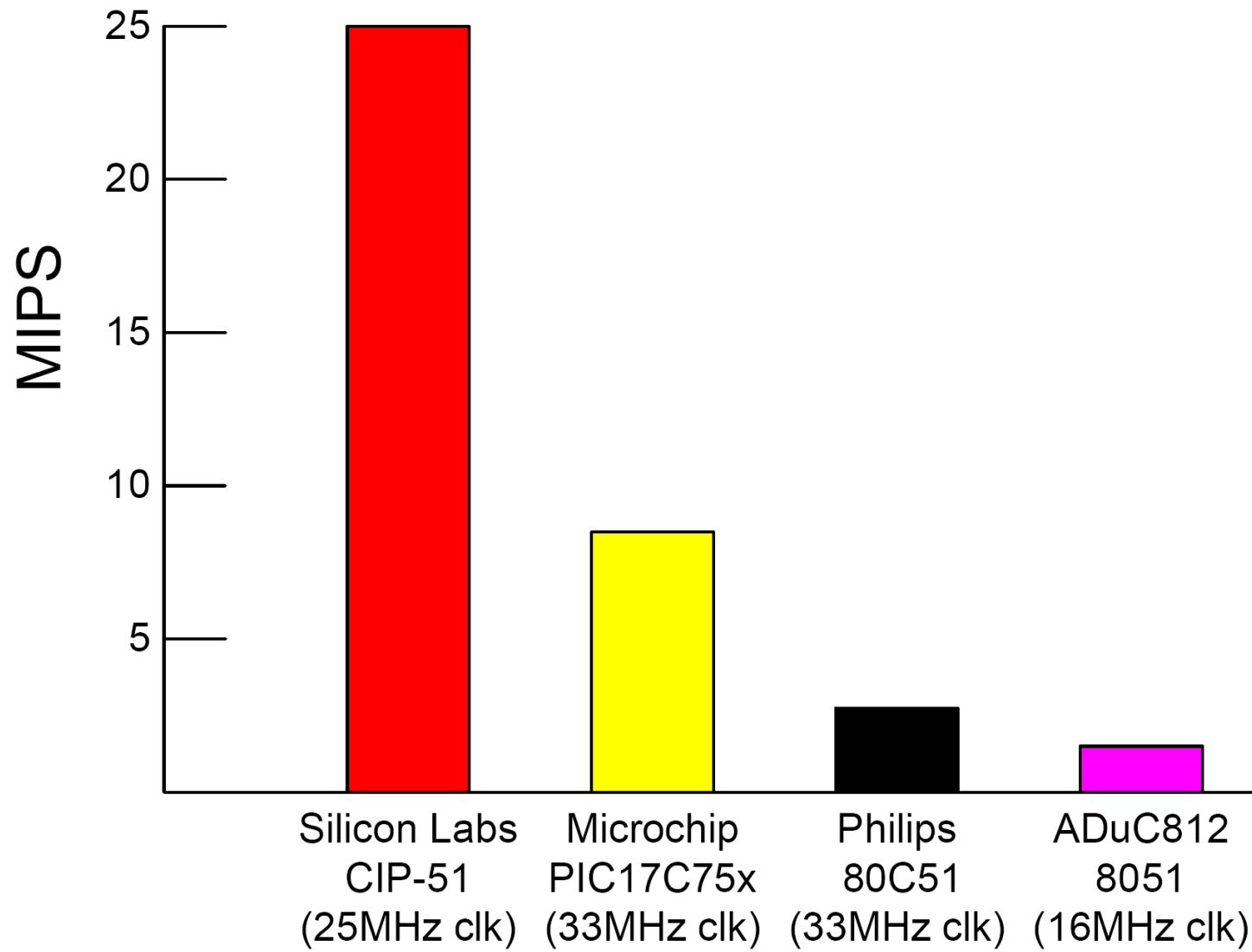
## • 顺序结构



## • 流水线结构（以三级为例）

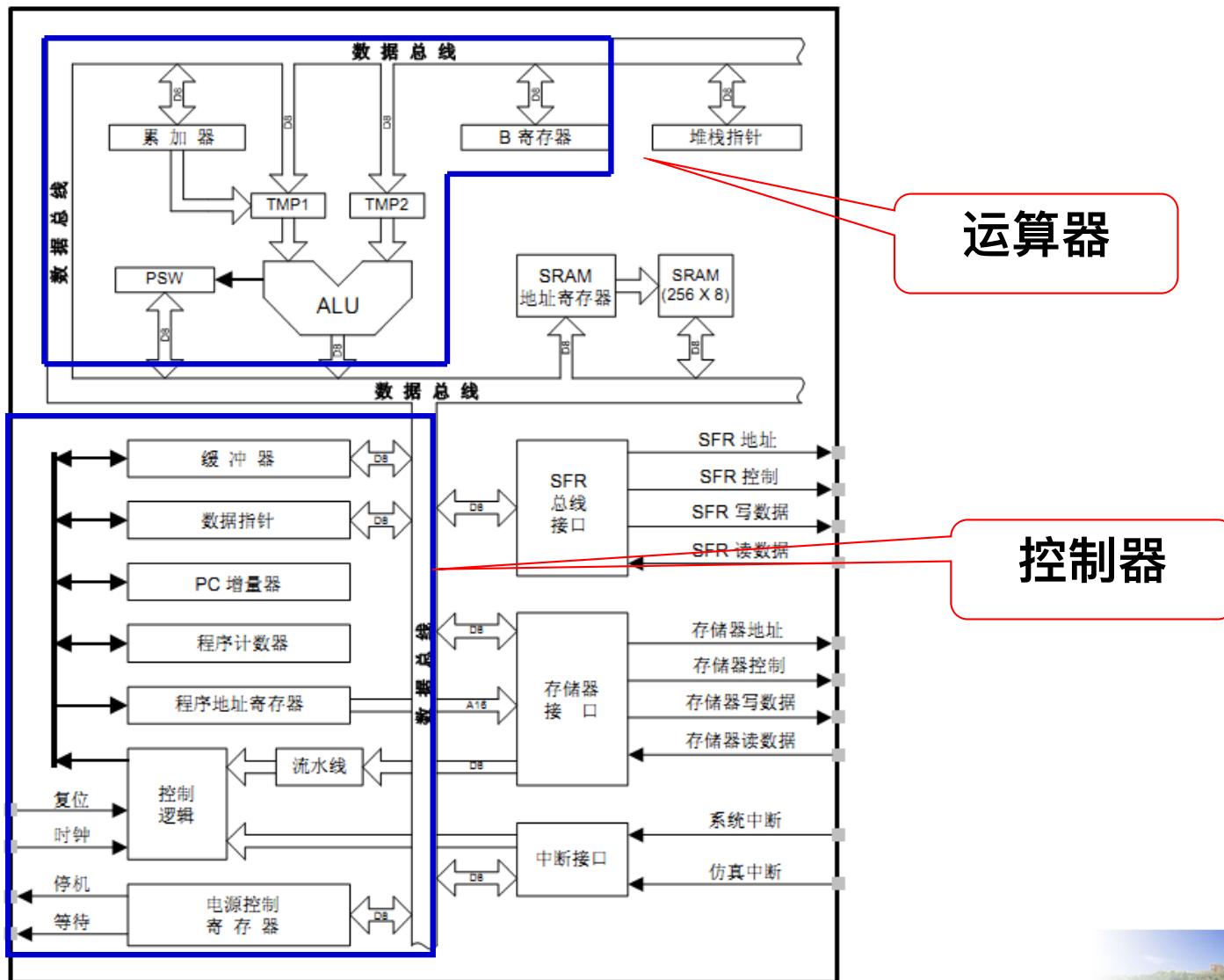


# 微控制器内核峰值执行速度比较



# C8051F单片机CPU结构

- CPU是计算机的核心部件，它由**运算器**、**控制器**组成，还包括一些**特殊功能寄存器**完成计算机的运算和控制功能



# 1、以ALU为中心的运算器

运算器又称算术逻辑部件 (ALU, Aithmctieal Logic Unit) , 主要完成对数据的算术运算和逻辑运算。

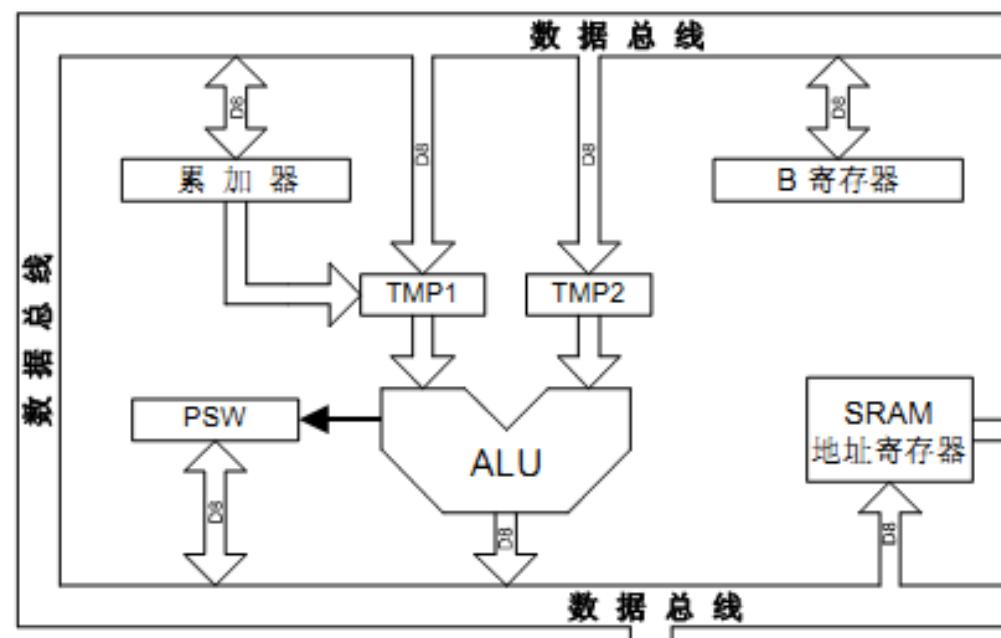
运算器相关的寄存器:

ACC — 累加器

B — 寄存器

TEMP—暂存器

PSW—程序状态字寄存器



**ACC—累加器：**运算器中最重要的工作寄存器，**用于存放参加运算的操作数和运算结果。**指令系统中常用A表示累加器。

**B—寄存器：**运算中的工作寄存器，在乘法和除法运算中存放操作数和运算结果。在其他运算中，可以作为一个中间结果寄存器使用。

**TEMP—暂存器：**用来暂存由数据总线或通用寄存器送来  
的操作数，并把它作为另一个操作数

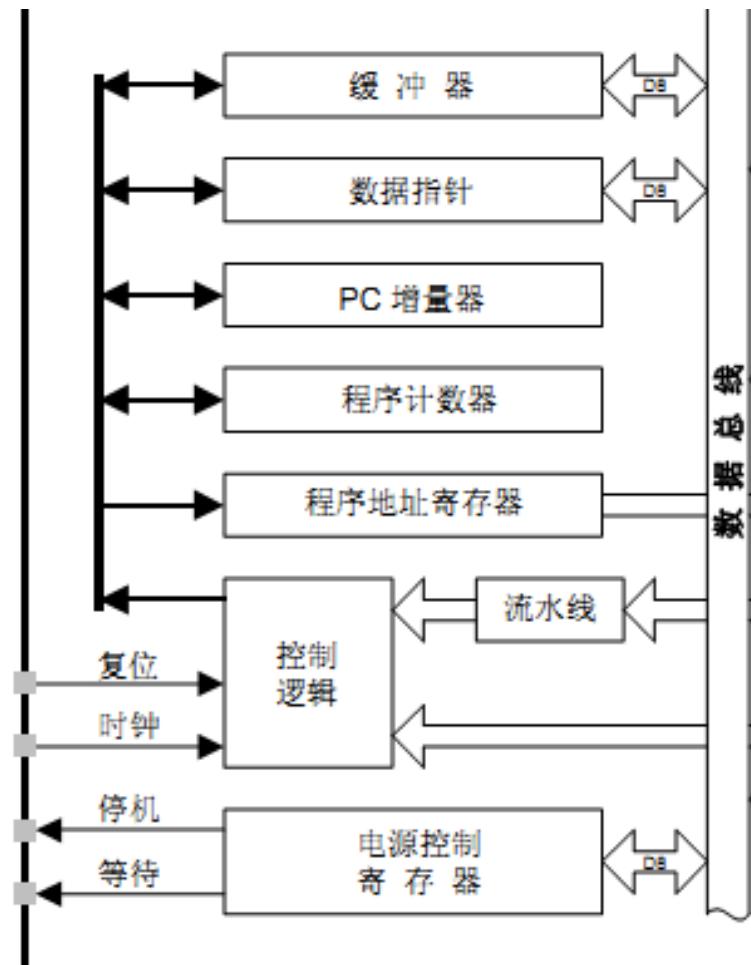
**PSW—程序状态字寄存器：**8位的寄存器，用于存放程序  
运行中的各种状态信息。

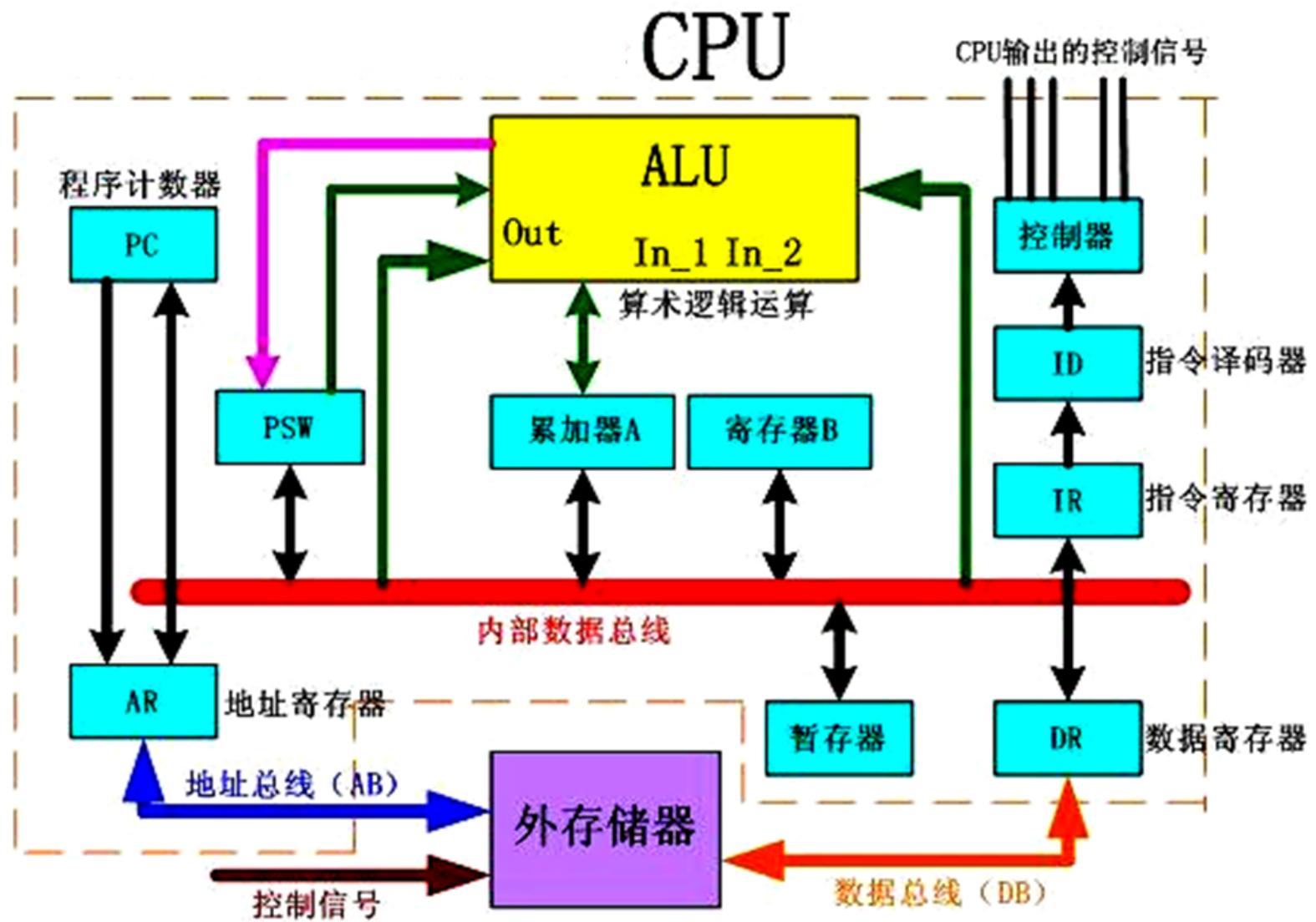


## 2、控制器 (Controller)

控制器 (Controller) 主要包括定时和控制逻辑、指令寄存器、译码器、地址指针DPTR和程序计数器PC等。

控制器是整个计算机的指挥中心, 它负责从内部存储器中取出指令并对指令进行分析、判断, 并根据指令发出控制信号, 使计算机的有关部件及设备有条不紊地协调工作, 保证计算机能自动、连续地运行。





CPU内部结构框图

## 程序计数器PC：

- 它是16位的按机器周期自动增1计数器
- 总指向下一条指令所在首地址(当前PC值)
- 一切分支/跳转/调用/中断/复位 等操作的本质就是:改变 PC 值

## 指令寄存器IR：

是用来存放当前正在执行的指令。

## 指令译码器ID：

是对存放在IR中的指令操作码进行解释，产生相应的控制信号。

## 数据指针DPTR：

是一个16位的寄存器，可以用来寻址外部数据存储空间，也可以寻址外部程序存储空间，寻址范围是64KB。



2.1 C8051F系列单片机总体结构

2.2 C8051F020存储器组织

2.3 中断系统

2.4 端口输入/输出

2.5 电源管理方式

2.6 复位与时钟



## 2.2 C8051F020存储器组织

- 兼容标准的8051存储器组织 (**哈佛结构**)
- 哈佛结构：程序和数据存储在不同的存储空间中，且独立编址、独立访问。
  - 物理结构上可分为：
    - 片内、片外**程序存储器**
    - 片内、片外**数据存储器**
  - 从功能寻址上可分为：
    - 程序存储器（空间）
    - 内部数据存储器
    - 特殊功能寄存器
    - 位寻址存储器
    - 外部数据存储器

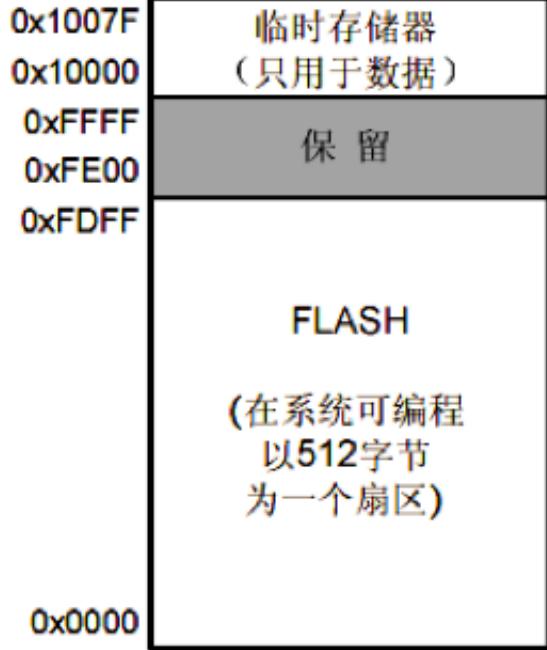
<https://www.bilibili.com/video/BV16741137Ag?p=19>



# C8051F020存储器结构

## 程序存储器

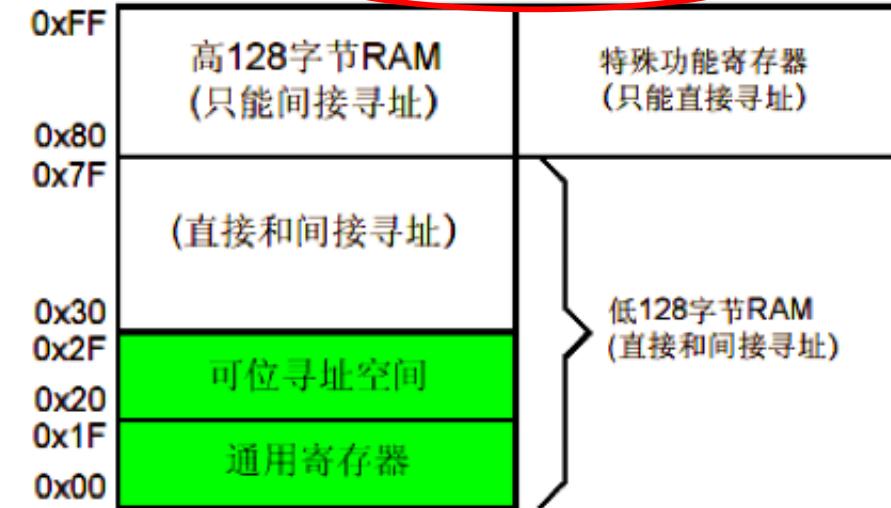
(FLASH)



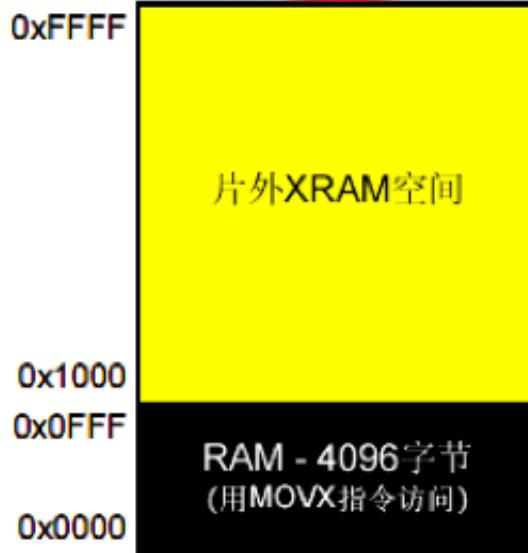
注意：地址+空间

## 数据存储器

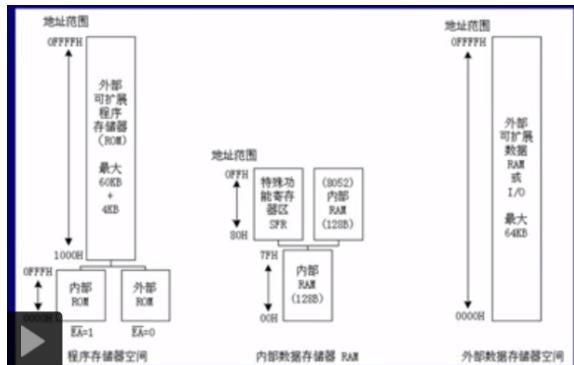
内部数据地址空间



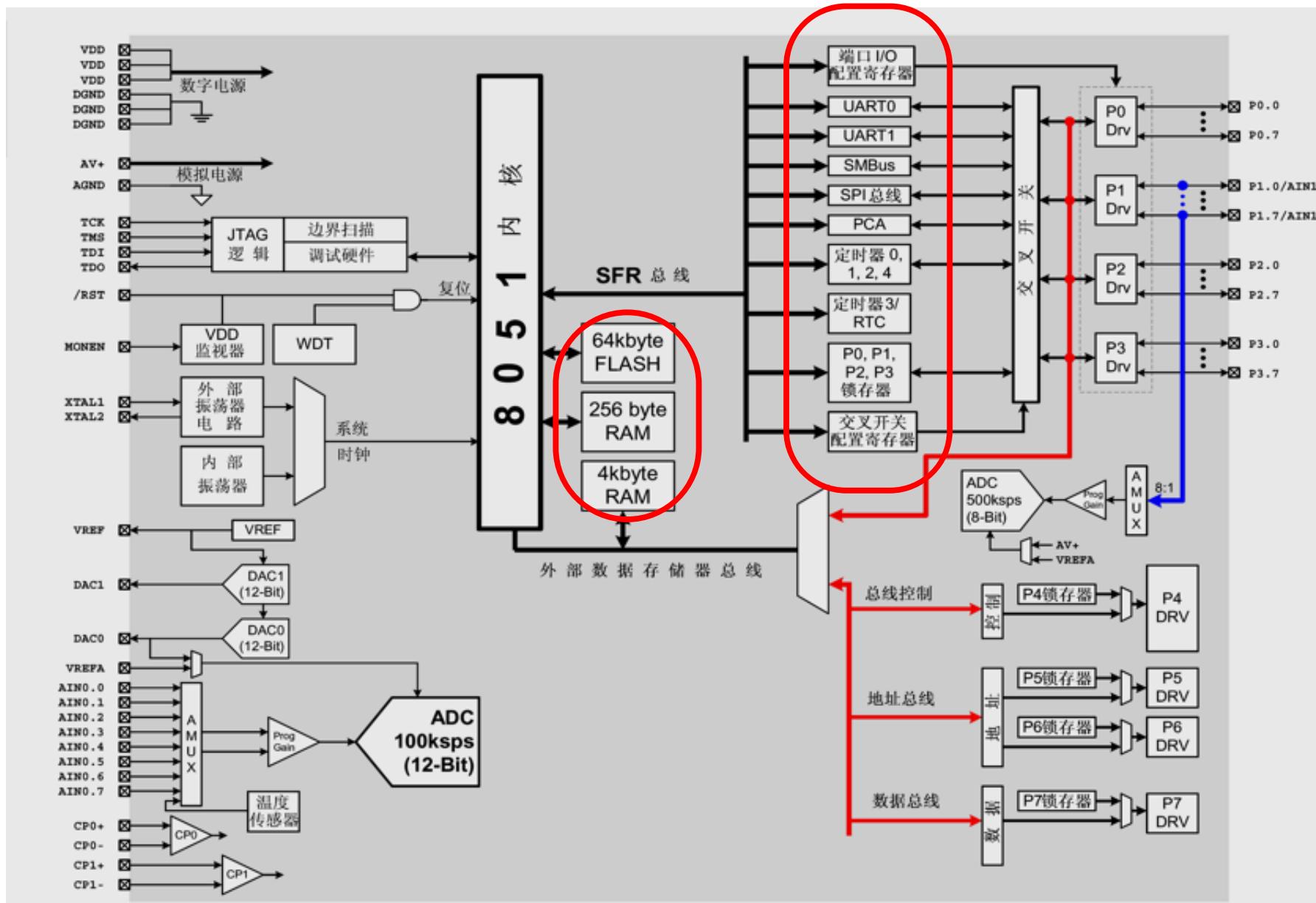
外部数据地址空间



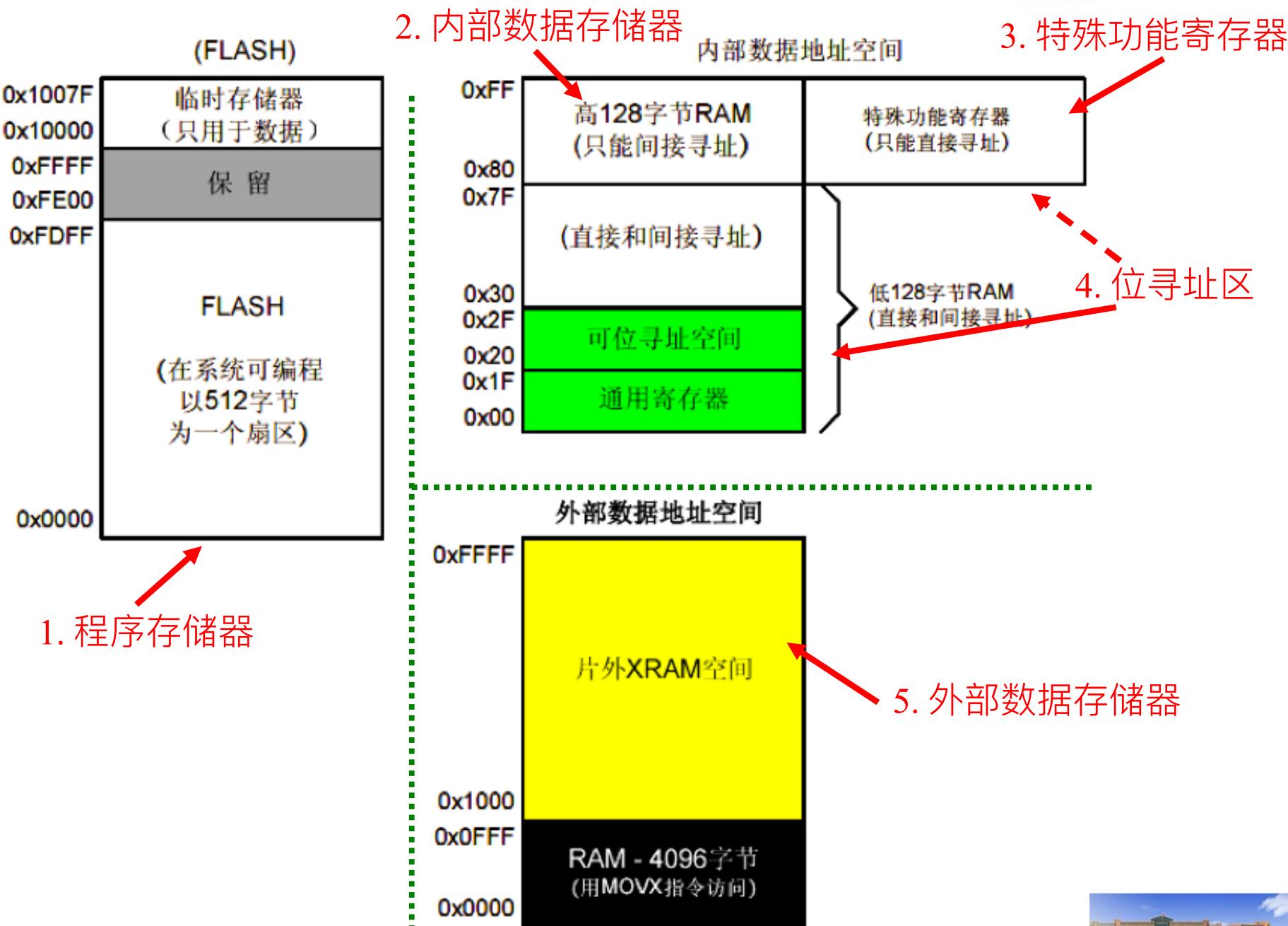
原理框图上如何对应?



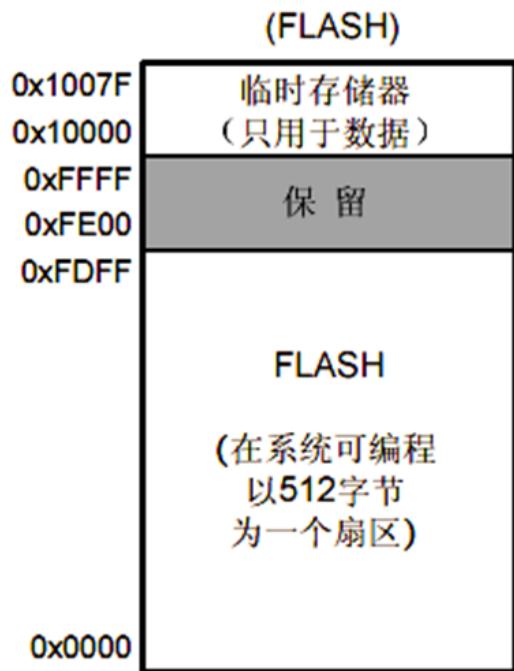
# C8051F020原理框图



# C8051F020存储器结构



# 1 程序存储器



- CIP-51 有64K+128字节的程序存储器空间。
- FLASH 存储器, 64K字节, 地址0x0000 到 0xFFFF。除了存储程序代码之外还可以用于存储非易失性数据。
- FLASH存储器中, 地址0xFE00 – 0xFFFF, 共 512 字节, 保留在工厂使用, 不能用于存储用户程序。
- FLASH存储器中, 有一个附加的128字节的扇区, 可用于非易失性数据存储, 如小的软件常数表。地址范围是从0x10000到0x1007F。
- 可读可写, 用MOVC指令读, 用MOVX指令写。编程前设置程序存储写允许PSWE位 (PSCTL.0) 为1, 以允许 FLASH 写操作。
- 使用编程工具通过JTAG接口对Flash存储器编程

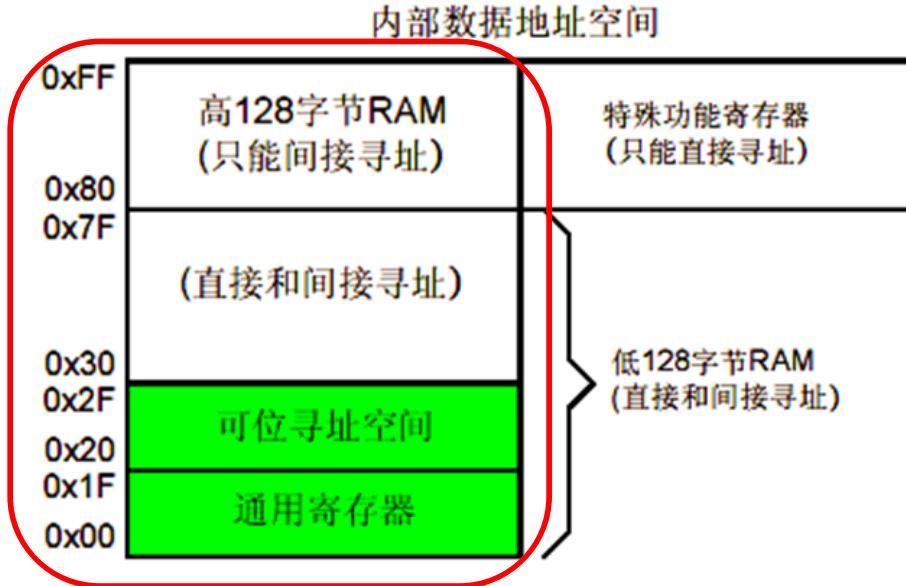
## 软件对FLASH的编程过程

- 禁止中断
- 置位LFWE (FLSCL.0) , 允许擦/写FLASH
- 置位PSEE (PSCTL.1) , 允许FLASH扇区擦除
- 置位PSWE (PSCTL.0) , 允许FLASH写
- 用MOVX向待擦扇区的任意地址写任意数据
- 清PSEE, 禁止FLASH擦除
- 用MOVX写入数据
- 清PSWE, 禁止写
- 开中断

Flash正在被编程或者擦除期间，CIP-51中的程序停止执行。  
在FLASH写/擦除操作期间发生的中断被挂起，等FLASH操作完成后按优先级顺序得到服务。



## 2 内部数据存储器

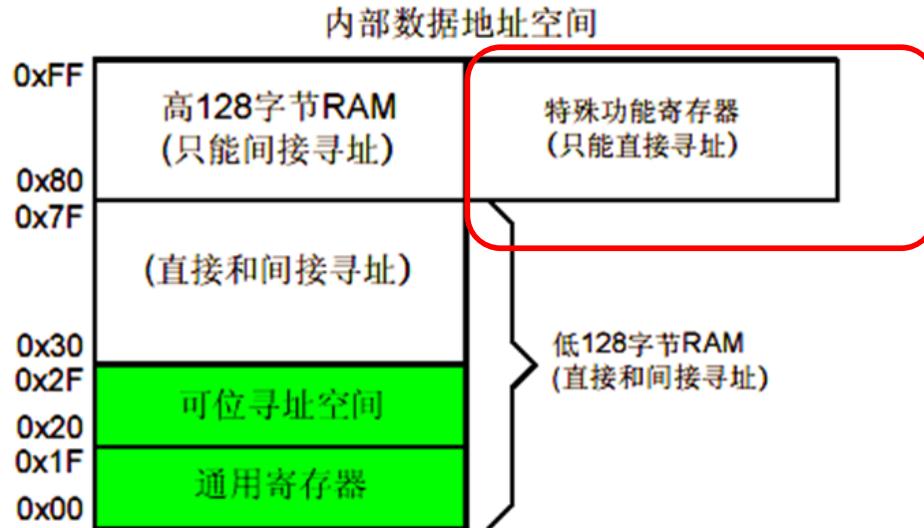


- 256字节的RAM，地址0x00到0xFF。
- 低128字节用于通用寄存器和临时存储器。
- 寻址方式：直接或间接。
- 0x00 – 0x1F：4个通用寄存器区，每个区有8个寄存器（R0~R7）。
- 通用寄存器（R0~R7）任一时刻，CPU只能使用其中一组，由程序状态字寄存器PSW中的RS1、RS0位决定。
- 0x20 – 0x2F：既可以按字节寻址又可以作为128个位地址用直接位寻址方式访问。
- 高128字节只能用间接寻址访问（R0、R1）。
- 高128字节与特殊功能寄存器（SFR）占据相同的地址空间，但物理空间不同，以寻址方式区分

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x59
sp_max	0x59
dptr	0x0000
PC	C:0x0003
states	389
sec	0.00019450
psw	0x00



### 3 特殊功能寄存器 (SFR)



- 0x80 ~ 0xFF : 直接寻址存储器空间为特殊功能寄存器SFR(Special Function Registers)的地址空间。
- 与内部RAM高128地址在物理上是两个空间。
- SFR 提供对CIP-51的资源和外设的控制及CIP-51与这些资源和外设之间的数据交换。
- CIP-51有120多个SFR，除了标准8051中的全部SFR，还增加了一些用于配置和访问专有子系统的SFR。具体见SFR表格2-3。编程时一般在头文件中定义。
- 地址能被8整除，即以0x0、0x8结尾的SFR可以按位寻址，其余只能按字节寻址



- **ACC (Accumulator)累加器：**存放运算的操作数和结果
- **B (B Register)寄存器：**乘除法中与ACC配合使用
- **SP (Stack Pointer) 8位堆栈指针：**
  - 复位值为0x07，使用时需要重新赋值。
  - 数据进入堆栈前，SP加1，数据退出堆栈后，SP减1。
  - **如果不对SP设置初值，则堆栈在0x08开始的区域。**
  - 理论上堆栈可以位于256字节数据存储器的任何位置
  - 一般在内部RAM的0x30~0xFF单元中开辟堆栈
  - SP一经确定，堆栈的位置也就跟着确定下来
  - SP可以初始化为不同的值，因此堆栈的位置是浮动的。
- **DPTR (Data Pointer)16位的数据指针：**访问外部存储器时使用，用于寄存器间接寻址。分为DPH、DPL。



## PSW—程序状态字寄存器



- **CY(PSW.7)进位/借位标志位。**若ACC在运算过程中发生了进位或借位，则CY=1；否则=0。它也是**布尔处理器的位累加器**，可用于布尔操作。
- **AC(PSW.6)半进位/借位标志位。**若ACC在运算过程中，最后一次算术操作向高半字节有进位（加法）或借位（减法）时，该位置1。其他算术操作将其清0。
- **F0 (PSW.5)可由用户定义的标志位。**



PSW.7

PSW.4 PSW.3

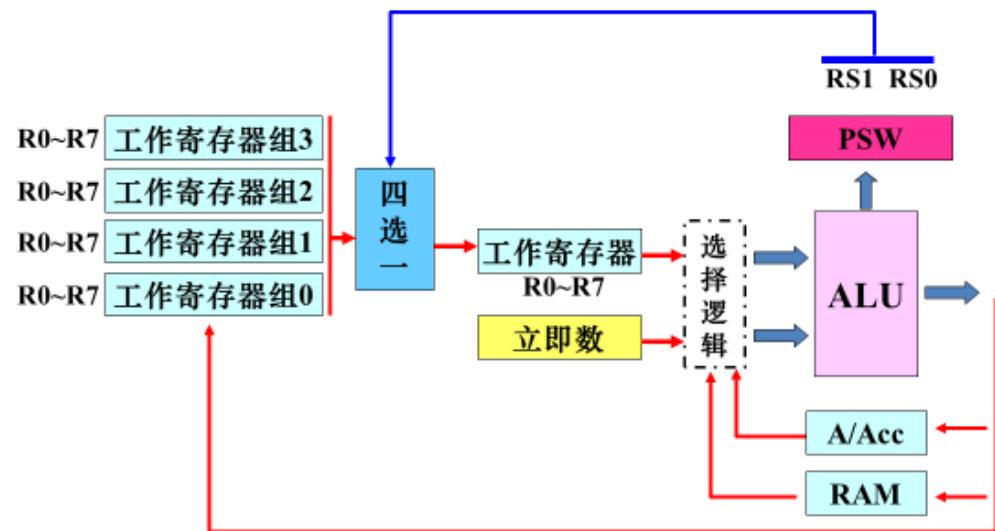
PSW.0



## RS1和RS0 ( PSW.4和 PSW.3) : 寄存器选择位

- RS1, RS0 = 0 0 则选择了工作寄存器组 0 , R0~R7地址为00H ~ 07H
- RS1, RS0 = 0 1 则选择了工作寄存器组 1 , R0~R7地址为08H ~ 0FH
- RS1, RS0 = 1 0 则选择了工作寄存器组 2 , R0~R7地址为10H ~ 17H

内部数据地址空间



PSW.7

PSW.2 PSW.1 PSW.0

CY	AC	F0	RS1	RS0	OV	—	P
----	----	----	-----	-----	----	---	---

➤ **OV (PSW.2)溢出标志位**

OV=1时特指累加器在进行带符号数(-128—+127)运算时出错 (超出范围)；

OV=0时未出错。

➤ **P (PSW.0)奇偶标志位。**

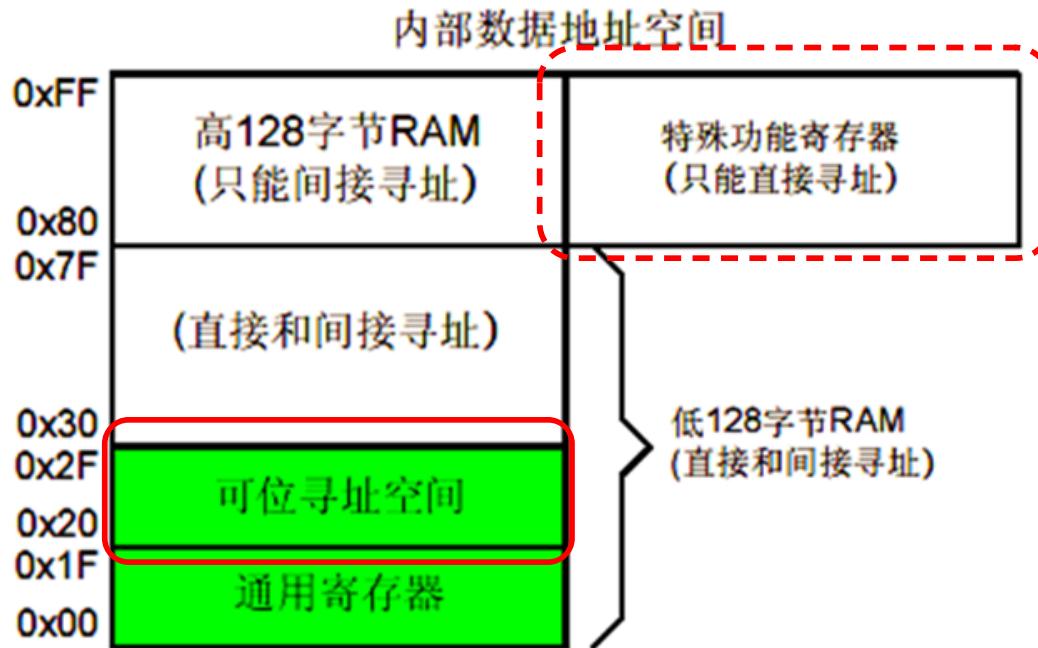
P=1表示累加器中“1”的个数为奇数

P=0表示累加器中“1”的个数为偶数

CPU随时监视着ACC中的“1”的个数，并反映在PSW中



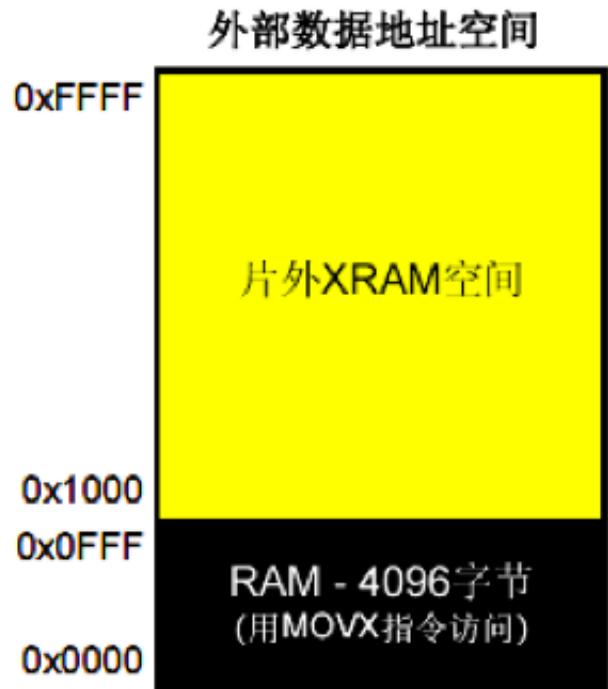
# 4 位寻址区



- 内部RAM中**0x20-0x2F单元**以及**特殊功能寄存器中地址为8的倍数的特殊功能寄存器**可以位寻址，它们构成了CIP-51的位存储器空间。
- 内部RAM中**0x20-0x2F共16个单元的每一位**都有一个**位地址**，它们占据地址空间**0x00~0x7F**，**具体见表2-4**。
- CIP-51内的布尔处理器，能对位地址空间中的位存储器直接寻址，对它们**执行置“1”、清“0”、取反、测试等操作**。



## 外部RAM和片内 XRAM



- 标准的51单片机只有128或256字节的内部数据存储器，不够时可在芯片外部扩展64KB的外部数据存储器。
- C8051F020单片机同样有可扩展的64KB外部数据空间。但在芯片内部有外部数据存储器空间的4096字节RAM (XRAM)，还有外部数据存储器接口(EMIF)可用于访问芯片外存储器和存储器映射的I/O器件。
- 外部存储器空间可以用外部传送指令(MOVX) 和数据指针(DPTR)访问，或者通过使用R0或R1用间接寻址方式访问。



## 方式1：MOVX+DPTR指针访问

16位形式的MOVX指令访问由DPTR寄存器的内容所指向的存储器单元。

- MOV DPTR. #1234h ; 将待读单元的16位地(0x1234)  
; 装入DPTR
- MOVX A, @DPTR ; 将地址0x1234的内容装入累加  
器A



## 方式2：R0或R1间接寻址访问

8位形式的MOVX指令用特殊功能寄存器EMIOCN的内容给出待访问地址的高8位，由R0或R1的内容给出待访问地址的低8位。

### 8位形式的MOVX指令

- MOV EMIOCN, #12h ; 将地址的高字节装入EMIOCN
- MOV R0, #34h ; 将地址的低字节装入R0 (或R1)
- MOVX A, @R0 ; 将地址0x1234的内容装入累加器A



# 外部存储器接口控制寄存器EMIOCN的格式如下：

R/W	复位值							
PGSE L7	PGSE L6	PGSE L5	PGSE L4	PGSE L3	PGSE L2	PGSE L1	PGSE L0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xAF

- 位7-0： PGSEL[7: 0]: XRAM页选择位
- 当使用8位的MOVX指令时， XRAM实际是选择256字节的RAM页

0x00: 0x0000-0x00FF

0x01: 0x0100-0x01FF

...

0xFE: 0xFE00-0xFEFF

0xFF: 0xFE00-0xFEFF



2.1 C8051F系列单片机总体结构

2.2 C8051F020存储器组织

2.3 中断系统

2.4 端口输入/输出

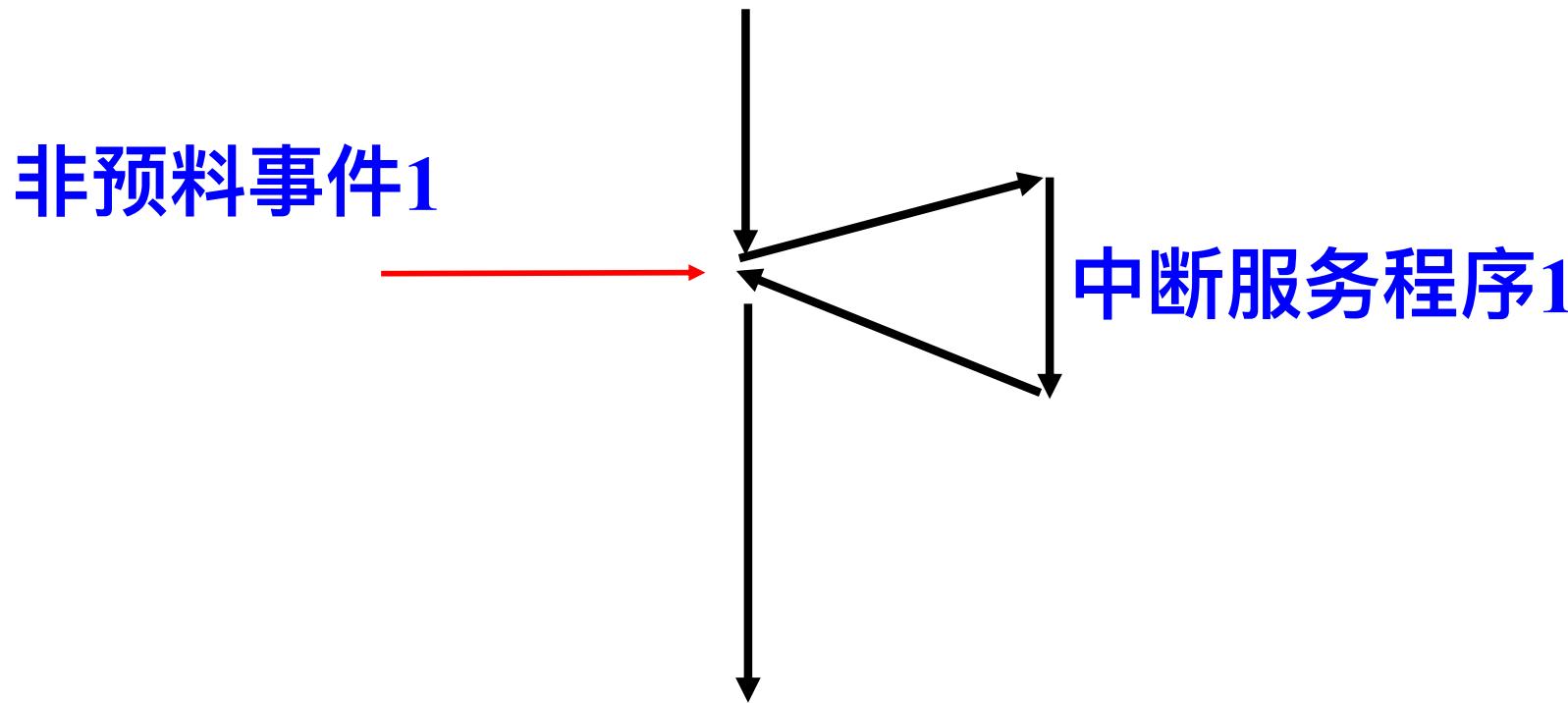
2.5 电源管理方式

2.6 复位与时钟



## • 2.3.1 中断概念

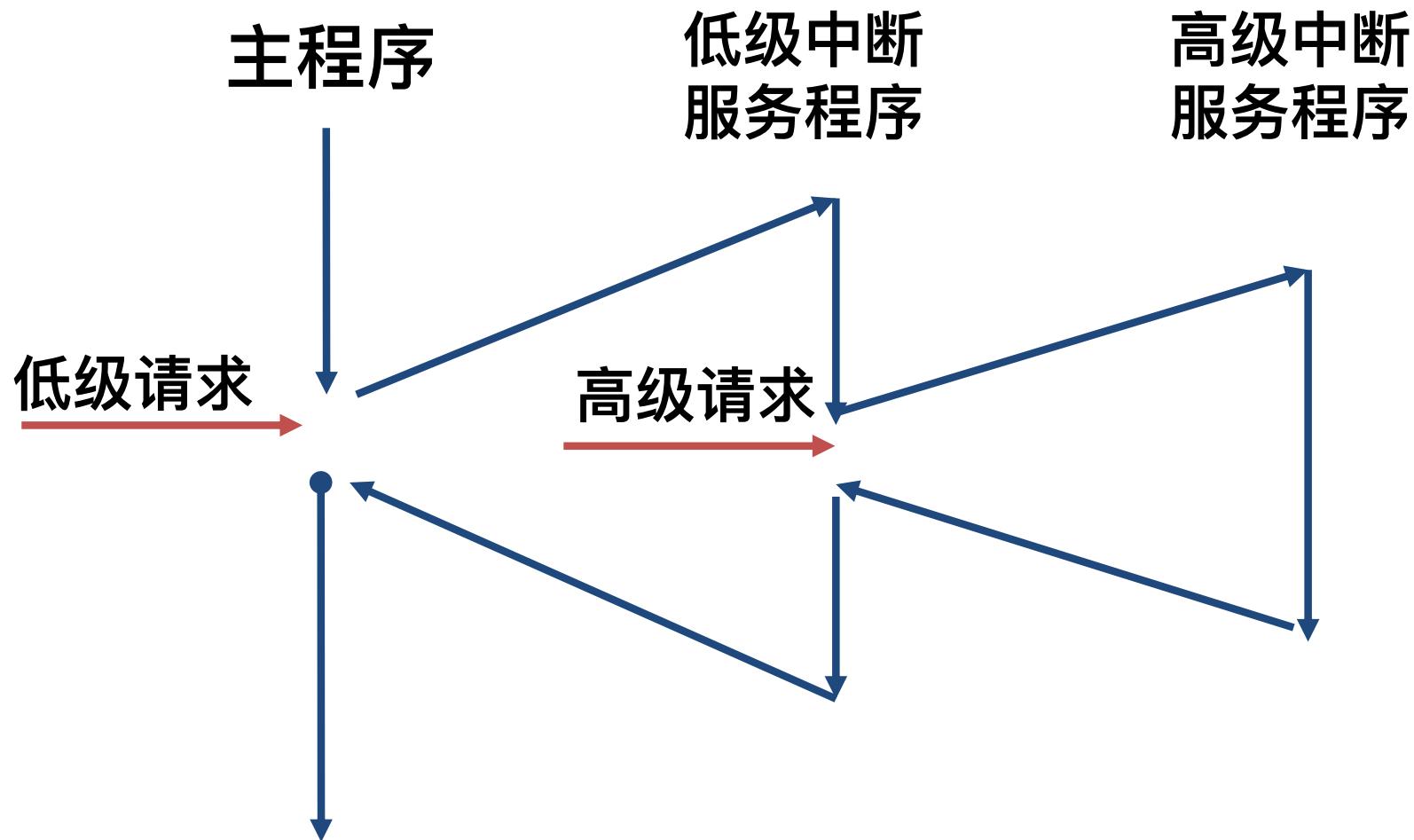
### CPU执行流程



中断的定义：在计算机运行过程中，发生紧急请求（如按键盘），CPU暂停当前工作，转而处理紧急请求，处理完毕后再回到原来被中断的地方，继续运行原来的工作。



# 二级中断服务程序嵌套（优先级）

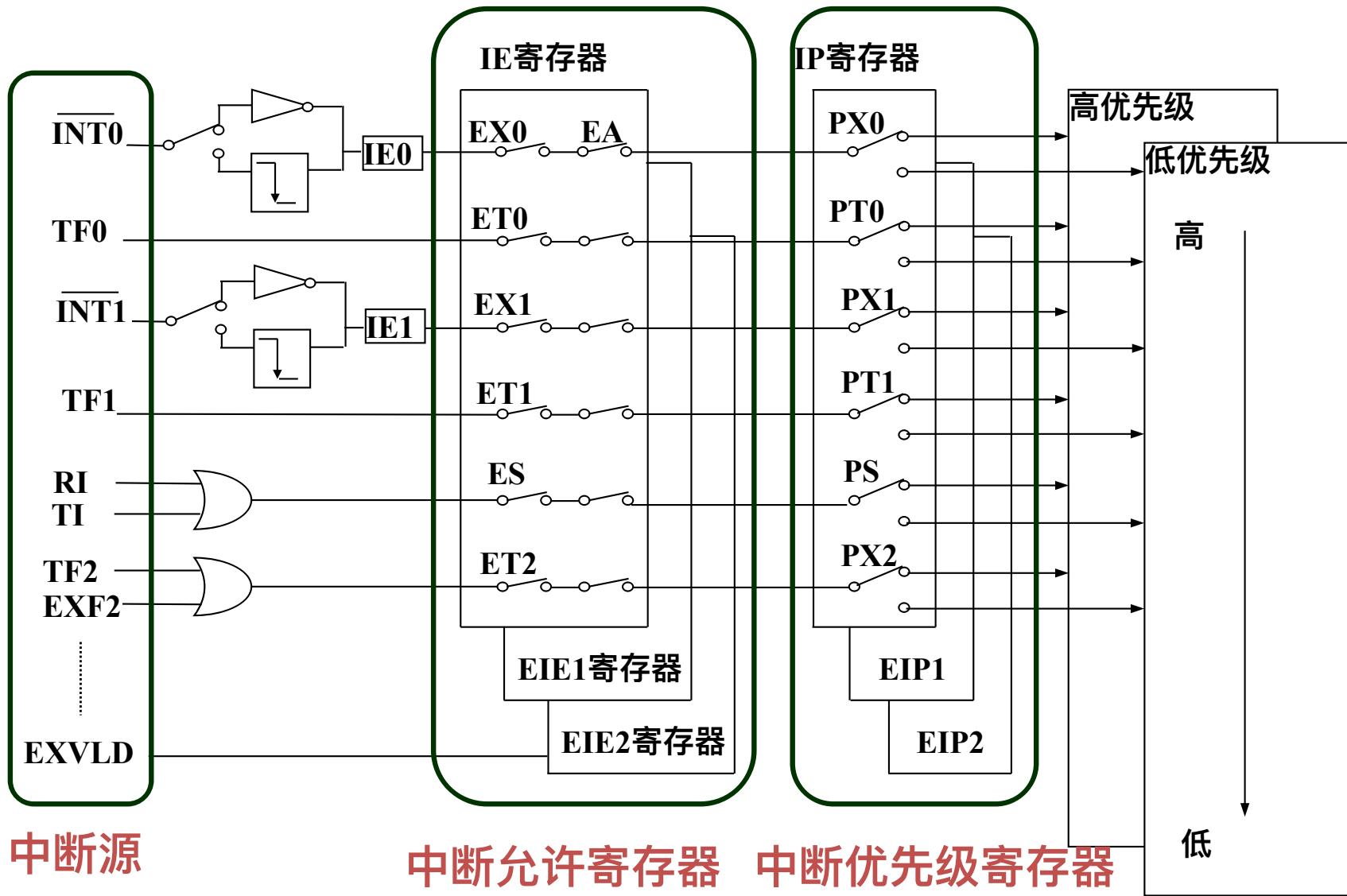


## 2.3.2 C8051中断系统

- C8051单片机有**22个中断源**，每个中断源有两个优先级
- 每个中断源可以在一个SFR中有一个或者多个**中断标志**，当一个外设或者外部源满足有效的中断条件时，相应的中断标志被置为逻辑“1”
- 如果中断被允许，在中断标志**被置位**时将产生中断，一旦**当前指令执行完，开始执行中断服务程序（ISR）**
- 每个ISR必须以RETI指令结束，使程序回到中断前执行完的那条指令的下一条指令
- 如果中断未被允许，中断标志将被硬件忽略，程序继续执行
- 每个中断源都可以用一个SFR中的相关中断允许位允许或者禁止，但是必须先置“1”EA位以保证每位中断允许位有效。
- 清零EA位将禁止所有 中断
- 某些中断标志在CPU进入ISR时被自动清除，但大多数中断标志不是硬件清除的，必须在ISR返回前用软件清除
- 如果中断标志在CPU执行完中断返回指令（RETI）后仍然保持置位状态，则会立即产生一个新的中断请求，CPU将在执行完下一条指令后重新进入ISR



# C8051F中断系统



# C8051F单片机的中断源

中断源	中断向量	优先级	中断标志	使能	优先级控制
复位	0x0000	最高	无	始终使能	总是最高
外部中断0 (INT0)	0x0003	0	IE0 (TCON.1)	EX0 (IE.0)	PX0 (IP.0)
定时器0溢出	0x000B	1	IF0 (TCON.5)	ET0 (IE.1)	PT0 (IP.1)
外部中断1 (INT1)	0x0013	2	IE1 (TCON.3)	EX1 (IE.2)	PX1 (IP.2)
定时器1溢出	0x001B	3	IF1 (TCON.7)	ET1 (IE.3)	PT1 (IP.3)

每个中断向量（中断服务程序入口地址）的间隔为8个字节，这8个字节是放不下对应的中断服务程序的。通常在中断入口，安排一条相应的跳转指令，以跳到用户设计的中断处理程序入口。



# 1. 中断源

## (1) 外部中断源

- 与MCS-51兼容的两个外部中断源，外部中断0 (/INT0) 和外部中断1 (/INT1)。
  - 可配置为低电平触发或下降沿触发。由IT0 (TCON.0) 和 IT1 (TCON.2) 的设置决定。
  - 如果被配置为边沿触发，CPU转向ISR时自动清除相应的中断标志
  - 如果被配置为低电平触发，中断标志将跟随外部中断输入引脚的状态
- C8051F020中还有另外2个外部中断源 (外部中断6、7) 为边沿触发输入，可以被配置为下降沿触发或上升沿触发。



## (2) 定时器/计数器溢出中断源

- 由内部定时器中断源产生，故它们**属于内部中断**
- C8051F020内部**有5个16位定时器/计数器 (T0~T4)** ,其中3个 (T0~T2) 与MCS-51兼容。
- 受内部定时脉冲或引脚上输入的外部定时脉冲控制，向CPU提出溢出中断请求。



### (3) 串行口发送 / 接收中断

- 由内部串行口中断源产生，故也是一种内部中断。  
C8051F020中有2个UART串行口（UART0、UART1）。
- 分为发送中断和接收中断，分别是对TI和RI中断位置1
- 串行口中断后通过判断TI和RI标志，区分是串行口发送中断还是串行口接收中断。
- 此外还有串行外设接口SPI、SMbus两种串口接口的中断源

### (4) 其它中断源

- C8051F020单片机还有电压比较器、A/D转换、晶振准备好等中断源。



## 2. 中断控制

### (1) 中断使能控制

- C8051F对所有的中断源的开放和屏蔽（总控制位EA）
- 每个中断源都有中断允许控制
- 受中断允许寄存器IE、EIE1、EIE2控制。

### (2) 中断优先级别的设定

- 每个中断源都可以设置为高优先级和低优先级，由中断优先级寄存器IP、EIP1、EIP2统一管理。



# 中断允许寄存器IE

位7	位6	位5	位4	位3	位2	位1	位0	复位值
EA	IEGF0	ET2	ES0	ET1	EX1	ET0	EX0	00000000
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	SFR地址: 0xA8

位7： EA位， EA=0： 禁止所有中断， EA=1： CPU开放中断， 每个中断位是开放还是屏蔽由各种的允许位确定

位6： IEGF0， 通用标志0。用作软件控制的通用标志位

位5： ET2， 定时器2中断允许位

位4： ES0， 串行口UART0中断允许位

位3： ET1， 定时器1中断允许位

位2： EX1， 外部中断1（/INT1）中断允许位

位1： ET0， 定时器0中断允许位

位0： EX0， 外部中断0（/INT0）中断允许位



## 例：开放定时器T0的溢出中断：

位7	位6	位5	位4	位3	位2	位1	位0	复位值
EA	IEGF0	ET2	ES0	ET1	EX1	ET0	EX0	00000000
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	SFR地址： 0xA8
1	0	0	0	0	0	1	0	

可以采用字节传送指令

**MOV IE, #82H**

也可以用位寻址指令，则需采用如下两条指令实现同样功能：

**SETB EA**

**SETB ET0**



## • 中断允许寄存器 EIE1

位7	位6	位5	位4	位3	位2	位1	位0	复位值
ECP1R	ECP1F	ECP0R	ECP0F	EPCA0	EWADC0	ESMB0	ESP10	00000000
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	SFR地址: 0xE6

- 位7: ECP1R, 允许比较器1 (CP1) 上升沿中断
- 位6: ECP1F, 允许比较器1 (CP1) 下降沿中断
- 位5: ECP0R, 允许比较器0 (CP0) 上升沿中断
- 位4: ECP0F, 允许比较器0 (CP0) 下降沿中断
- 位3: EPCA0, 允许可编程计数器阵列 (PCA0) 中断
- 位2: EWADC0, 允许ADC0窗口比较中断
- 位1: ESMB0, 允许SMBus0中断
- 位0: ESP10, 允许串行外设接口0中断



## • 中断允许寄存器 EIE2

位7	位6	位5	位4	位3	位2	位1	位0	复位值
<b>EXVLD</b>	<b>ES1</b>	<b>EX7</b>	<b>EX6</b>	<b>EADC1</b>	<b>ET4</b>	<b>EADC0</b>	<b>ET3</b>	<b>00000000</b>
<b>R/W</b>	<b>R/W</b>	<b>R/W</b>	<b>R/W</b>	<b>R/W</b>	<b>R/W</b>	<b>R/W</b>	<b>R/W</b>	<b>SFR地址: 0xE7</b>

位7: EXVLD, 允许外部时钟源有效 (XTLVLD) 中断

位6: ES1, 允许UART1中断

位5: EX7, 允许外部中断7

位4: EX6, 允许外部中断6

位3: EADC1, 允许ADC1转换结束中断

位2: ET4, 允许定时器4中断

位1: EADC0, 允许ADC0转换结束中断

位0: ET3, 允许定时器3中断



# 中断优先级寄存器

中断优先级寄存器 IP各位的定义如下：

中断优先级寄存器 EIP1各位的定义如下：

位7	位6	位5	位4	位3	位2	位1	位0	复位值
<b>PCP1R</b>	<b>PCP1F</b>	<b>PCP0R</b>	<b>PCP0F</b>	<b>PPCA 0</b>	<b>PWADC 0</b>	<b>PSMB 0</b>	<b>PSP10</b>	<b>00000000</b>
<b>R/W</b>	<b>R/W</b>	<b>R/W</b>	<b>R/W</b>	<b>R/W</b>	<b>R/W</b>	<b>R/W</b>	<b>R/W</b>	<b>SFR地址： 0xF6</b>

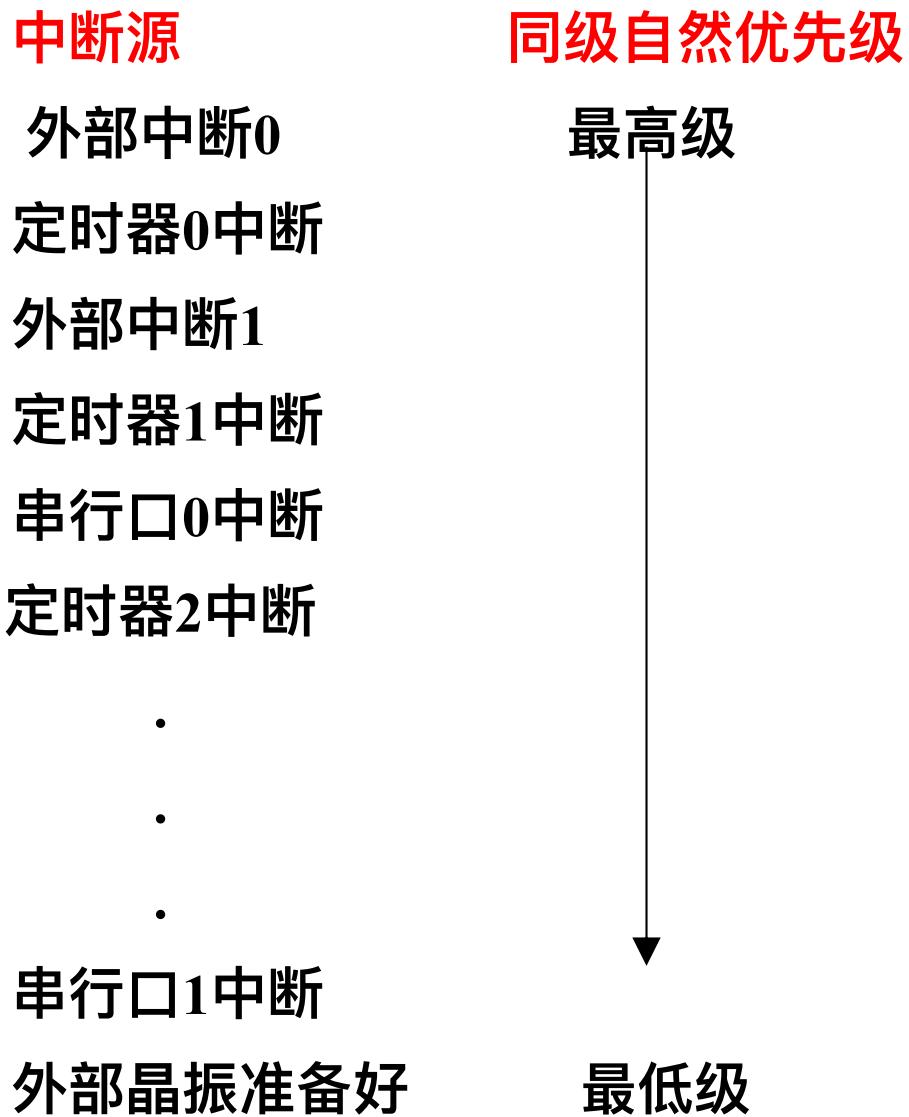
中断优先级寄存器 EIP2各位的定义如下：

# 优先级管理

- 中断优先级只有高低两级。若有两个或两个以上中断源处于同一中断优先级，内部中断系统对各中断源的处理遵循以下两条基本原则：
  1. 低优先级中断可以被高优先级中断所中断，反之不能。
  2. 一种中断（不管是什优先级）一旦得到响应，与它同级的中断不能再中断它。
- 由两个不可寻址的“优先级激活”触发器实现上述基本原则。
- 同时收到几个同一优先级的中断请求时，哪一个请求将得到服务，取决于内部的硬件查询次序，CPU将按自然优先级顺序确定该响应哪个中断请求。



# 自然优先级由硬件形成



### 3. 中断响应的时间

- 最快的响应时间为5个系统时钟周期：一个周期用于检测中断，4个周期完成对ISR的长调用（LCALL）。
- 如果中断标志有效时CPU正在执行RETI指令，则需要再执行一条指令才能进入中断服务程序。
- 最长的响应时间是下一条指令是DIV的情况。响应时间为18个系统时钟周期：
  - 1个时钟周期检测中断；
  - 5个时钟周期执行RETI；
  - 8个时钟周期完成DIV指令；
  - 4个时钟周期执行对ISR的长调用（LCALL）



# CPU响应中断时：

1. 先置位相应的优先级状态触发器（不可寻址，该触发器指出CPU开始处理的中断优先级别）
2. 执行一条硬件子程序调用指令，控制转移到相应的入口，清零中断请求标志（有些中断源如RI、TI不能被清除，需要中断服务程序软件清除）。
3. 把程序计数器PC的内容压入堆栈（但不保护PSW），将被响应的中断服务程序的入口地址送程序计数器PC，各中断源服务程序的入口地址可参见中断向量栏。



## 4. 中断响应过程：

1. 当中断产生时，首先由硬件实现返回地址（PC值）压入堆栈操作。
2. 然后中断响应函数被调用，用软件方式实现以下处理：
  - 使用语句将ACC、B、DPH、DPL、PSW这些特殊功能寄存器的值保存在堆栈中。
  - 如果中断响应函数未使用using属性进行修饰，中断响应函数所使用的通用寄存器的值保存到堆栈中。
  - 对中断进行处理。
  - 恢复保存寄存器的值，退出中断响应。
3. 执行RETI语句，硬件实现返回地址的载入，并跳转到对应的语句执行。



# 中断响应函数

中断响应函数的定义：

**函数类型 函数名 (形式参数列表) interrupt n**

注意：中断属性带一个值n为0~31的整形参数，用来表示中断响应函数所对应的中断号。（表2-5 优先级）

例：

```
void falarm (void) interrupt 1//中断号1对应中断源T0
{
    alarm_count++;
}
```

注意：仅能在函数定义时使用interrupt函数属性，不能在函数声明时使用interrupt函数属性



## 5. 外部中断触发方式的选择

外部中断0 (/INT0) 、外部中断1(/INT1)有两种触发方式:

### (1) 电平触发方式

外部中断输入信号必须有效（保持低电平），直至CPU响应该中断为止，同时在中断服务程序返回之前必须使外部中断输入信号无效，否则CPU在中断返回后又会再次响应中断。

### (2) 边沿触发方式

外部中断标志触发器锁存外部中断输入线上的负跳变，即使CPU暂时不响应，中断申请标志也不会丢失。



2.1 C8051F系列单片机总体结构

2.2 C8051F020存储器组织

2.3 中断系统

2.4 端口输入/输出

2.5 电源管理方式

2.6 复位与时钟



## 2.4 端口输入/输出

C8051F020有按8位端口组织的64个数字I/O引脚。

低端口（P0、P1、P2和P3）既可以按位寻址也可以按字节寻址。

高端口（P4、P5、P6和P7）只能按字节寻址。

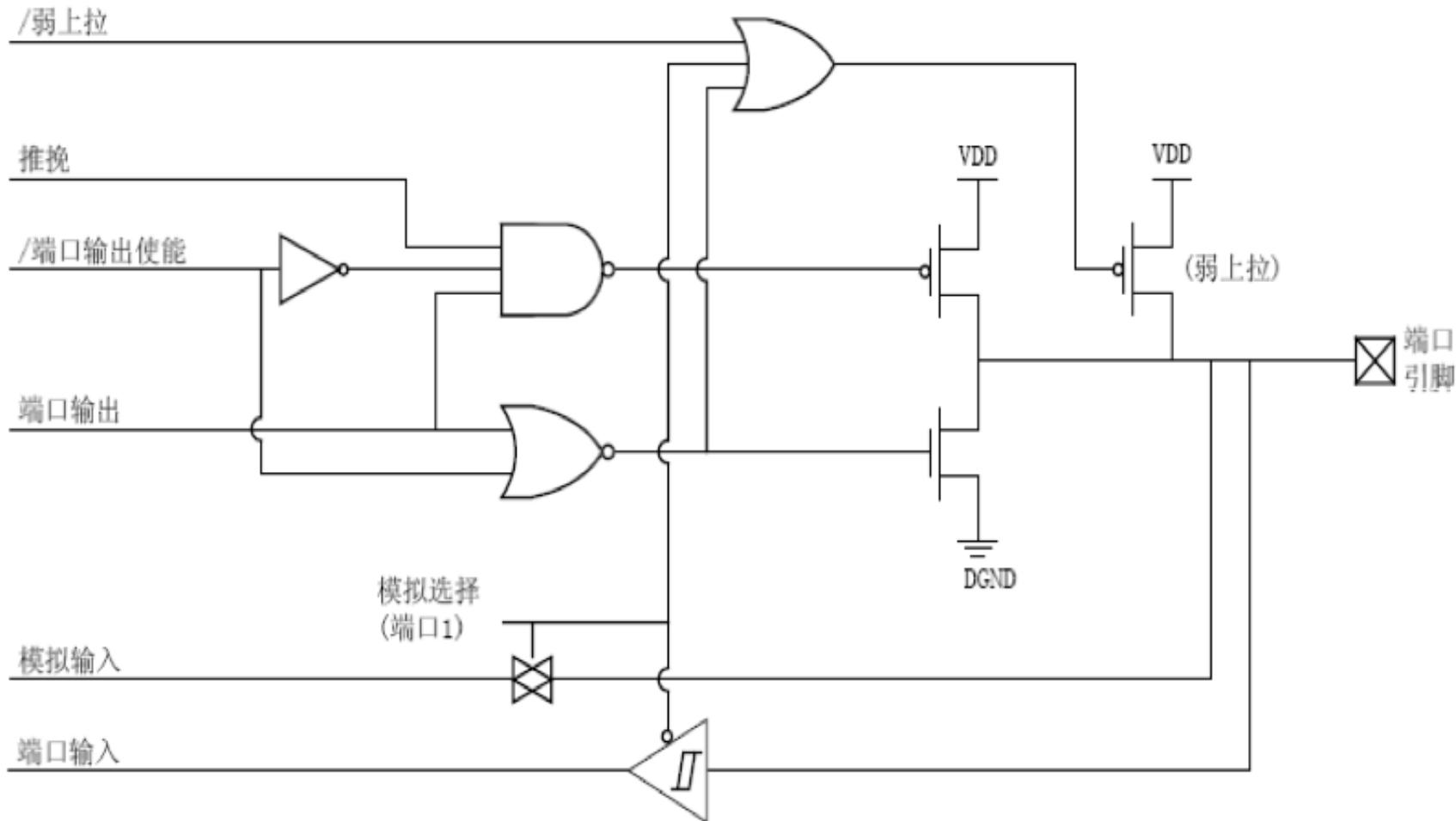
所有引脚都耐5V电压，都可以被配置为：

漏极开路或推挽输出方式和弱上拉。

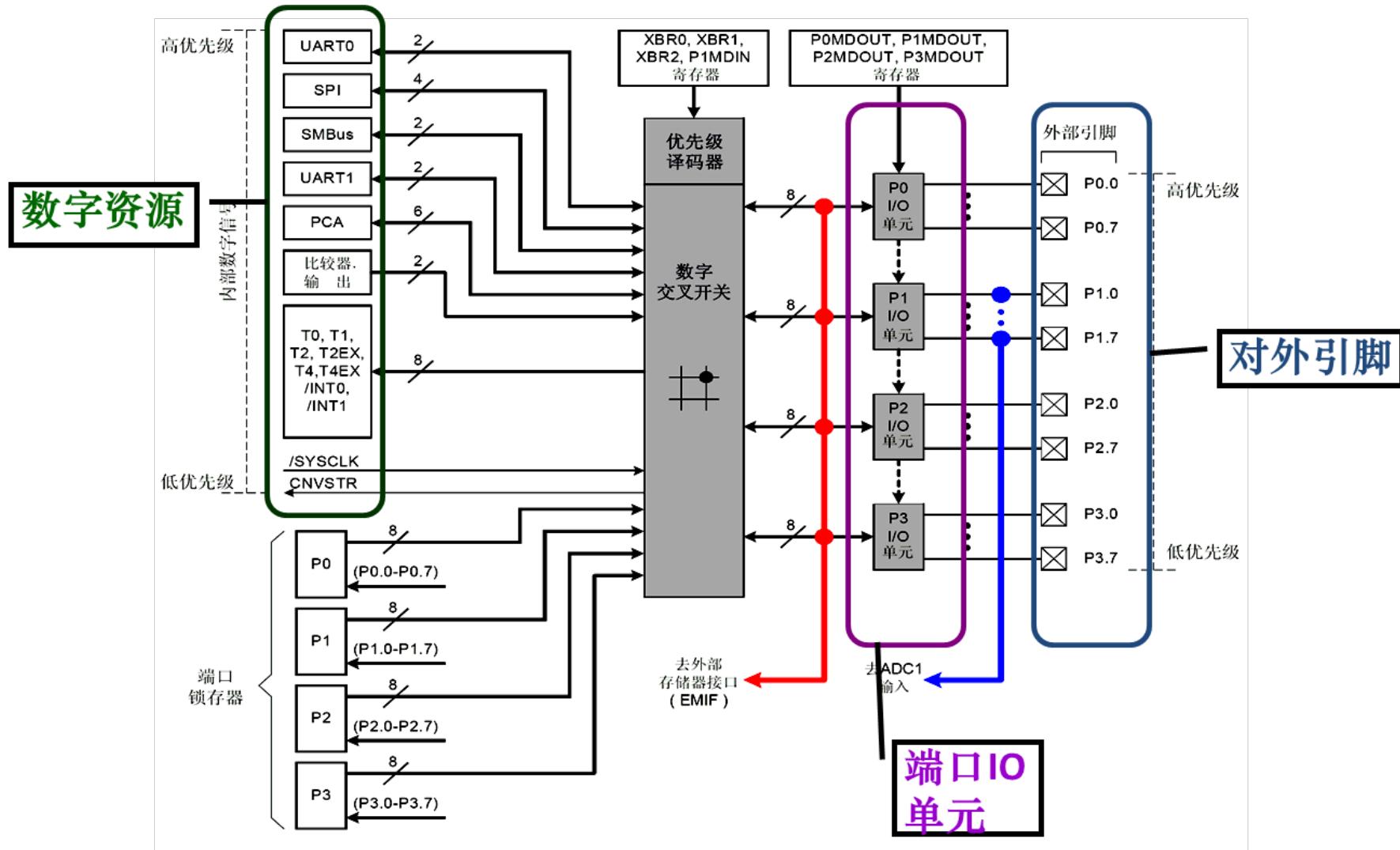


# 1 I/O 端口结构

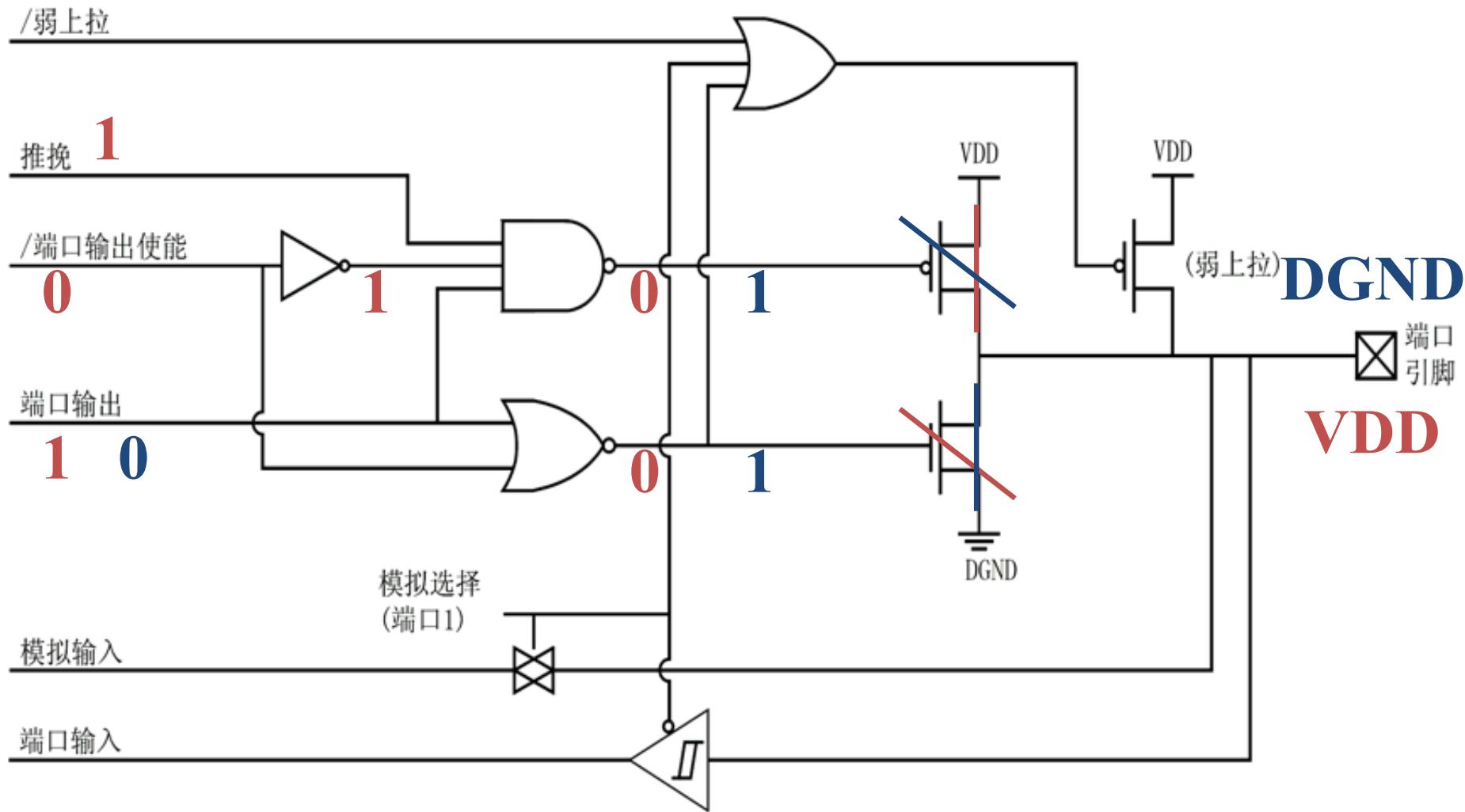
图 17.1 端口 I/O 单元功能框图



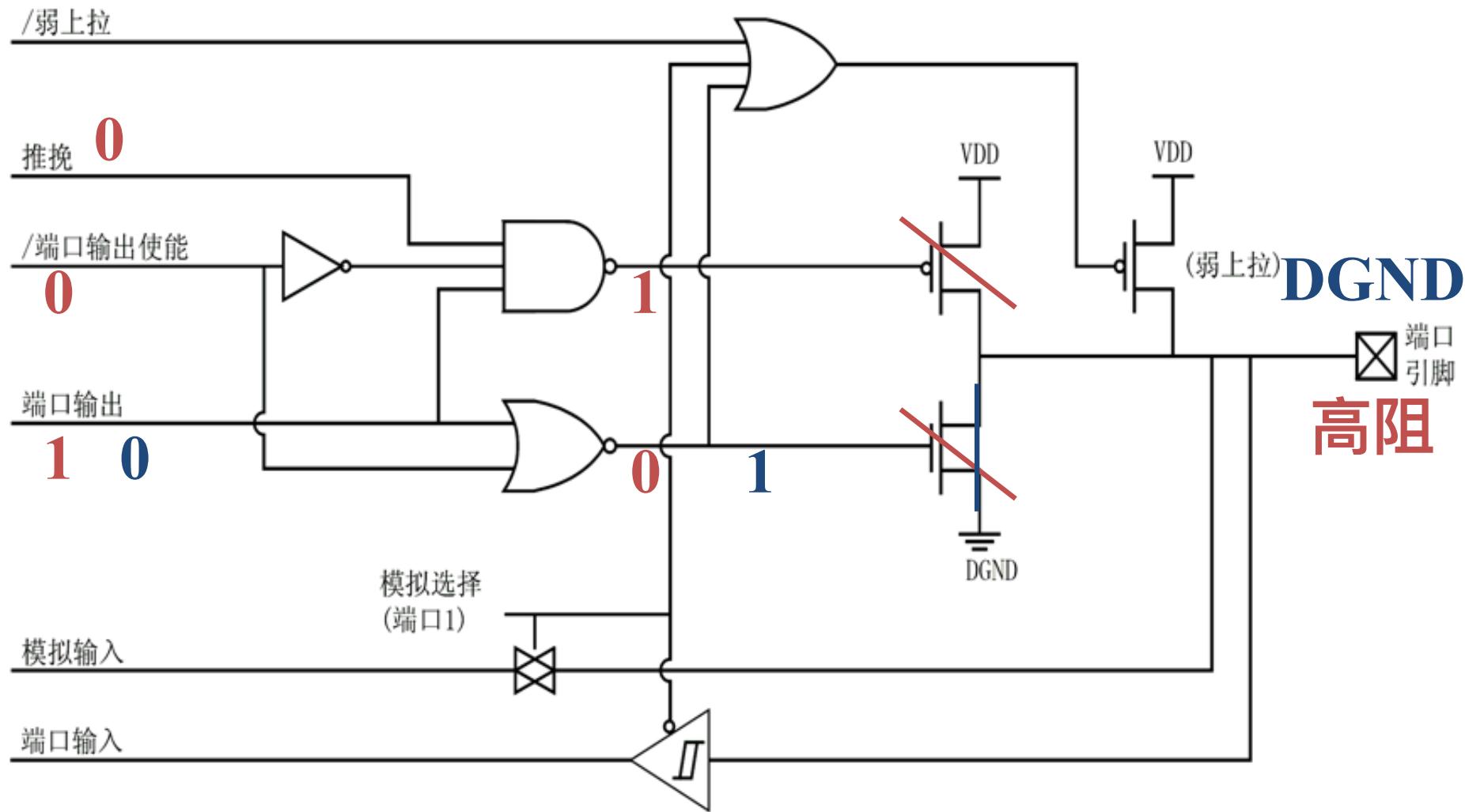
# 1 I/O 端口结构



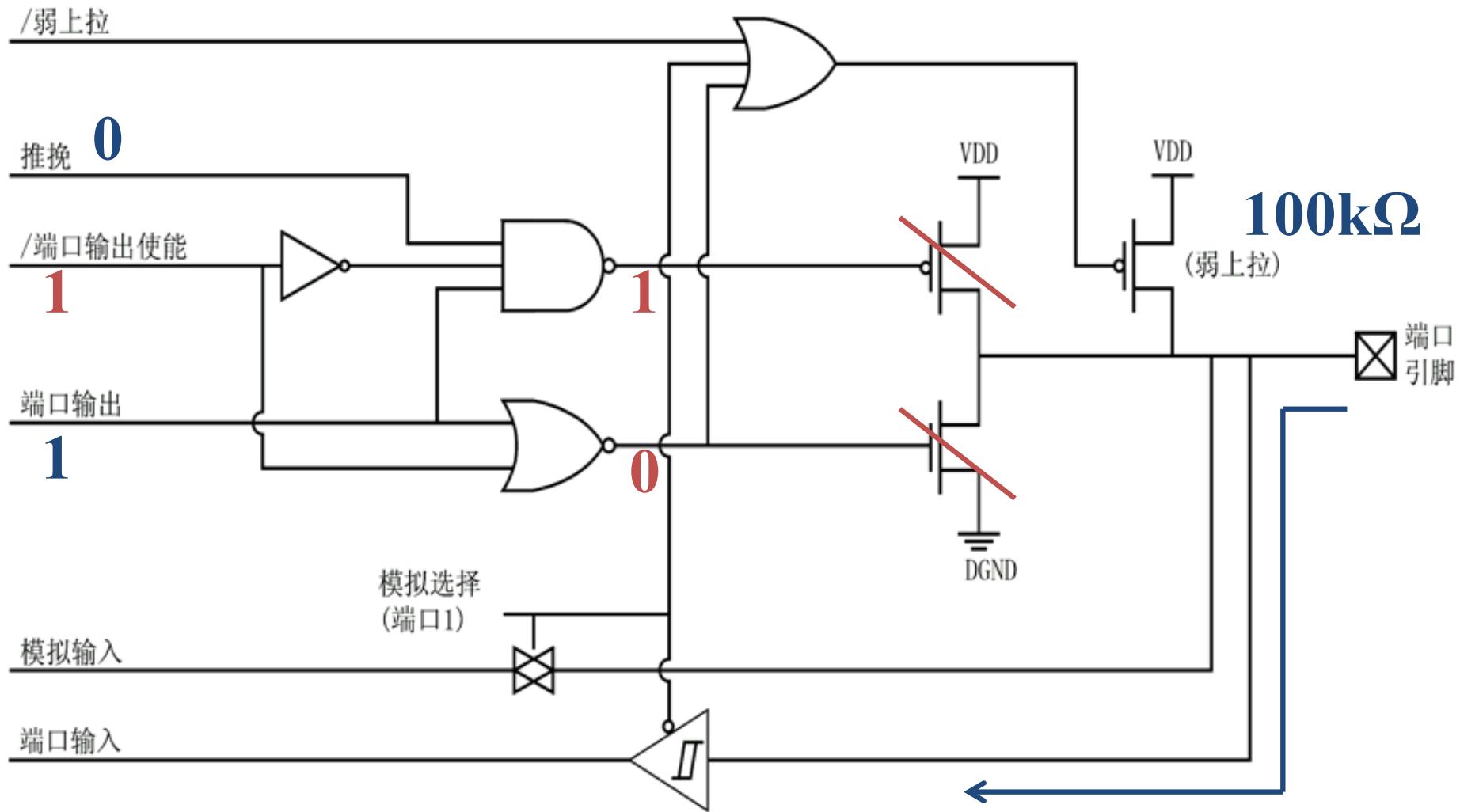
## 推挽输出方式



## 漏极开路输出



## 数字输入



## 2 I/O 端口直流电气特性

表 17.1 端口 I/O 直流电气特性

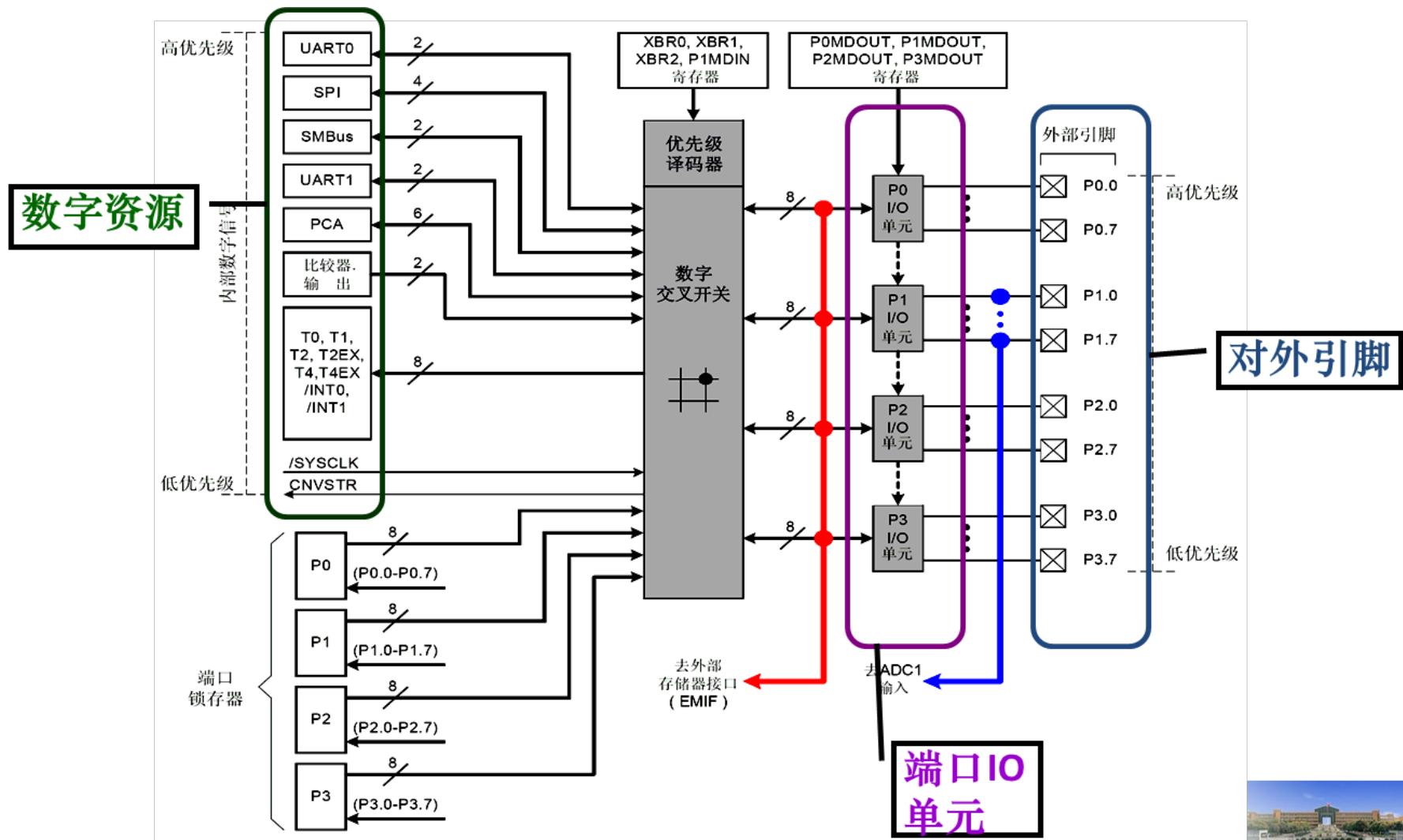
VDD = 2.7V – 3.6V, -40°C 到 +85°C (除非另有说明)。

参数	条件	最小值	典型值	最大值	单位
输出高电压 (V <sub>OH</sub> )	I <sub>OH</sub> = -10μA, 端口 I/O 为推挽方式 I <sub>OH</sub> = -3mA, 端口 I/O 为推挽方式 I <sub>OH</sub> = -10mA, 端口 I/O 为推挽方式	VDD-0.1 VDD-0.7 VDD-0.8			V
输出低电压 (V <sub>OL</sub> )	I <sub>OL</sub> = 10μA I <sub>OL</sub> = 8.5mA I <sub>OL</sub> = 25mA			0.1 0.6 1.0	V
输入高电压 (V <sub>IH</sub> )		0.7×VDD			V
输入低电压 (V <sub>IL</sub> )				0.3×VDD	V
输入漏电流	DGND < 端口引脚 < VDD, 高阻态 弱上拉禁止 弱上拉使能			±1 10	μA
输入电容			5		pF



### 3 低端口功能结构

C8051F020的数字资源需要通过4个低端I/O端口才能使用。每个引脚既可定义为通用的端口I/O (GPIO) 引脚，又可以分配给一个数字外设或功能。



# 4 优先权交叉开关译码器

- C8051F020 单片机有大量的数字资源需要通过4个低端I/O端口P0、P1、P2和P3才能使用。P0、P1、P2和P3中的每个引脚既可定义为通用的端口I/O (GPIO) 引脚，又可以分配给一个数字外设或功能（例如：UART0 或/INT1）。
- 这种资源分配的灵活性是通过使用优先权交叉开关译码器实现的。
- 或称为“交叉开关”，按优先权顺序将端口0 -3 的引脚分配给器件上的数字外设 (UART、SMBus、PCA、定时器等)。
- 端口引脚的**分配顺序是从P0.0 开始**，可以一直分配到P3.7。为数字外设分配端口引脚的优先权顺序为UART0具有最高优先权，而CNVSTR具有最低优先权。
- 优先权交叉开关的配置是通过3个特殊功能寄存器XBR0、XBR1、XBR2来实现的，对应使能位被设置为逻辑‘1’时，交叉开关将端口引脚分配给外设。



# 优先权交叉开关译码表

高

UART0

SPI

SMBus

UART1

PCA

↓  
低

引脚 I/O	P0				P1				P2				P3				交叉开关寄存器位											
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7				
TX0	●																								UART0EN:XBR0.2			
RX0		●																										
SCK	●	●																							SPI0EN:XBR0.1			
MISO		●	●																									
MOSI			●	●																								
NSS				●		●																						
SDA	●	●	●	●																					SMB0EN:XBR0.0			
SCL		●	●	●		●																						
TX1	●	●	●	●		●																			UART1EN:XBR2.2			
RX1		●	●	●		●																						
CEX0	●	●	●	●		●																						
CEX1		●	●	●		●																						
CEX2			●	●		●																						
CEX3				●		●																						
CEX4					●	●																			PCA0ME:XBR0.[5:3]			
ECI	●	●	●	●		●																			ECI0E:XBR0.6			
CP0	●	●	●	●		●																			CP0E:XBR0.7			
CP1	●	●	●	●		●																			CP1E:XBR1.0			
T0	●	●	●	●		●											●	●							T0E:XBR1.1			
/INT0	●	●	●	●		●																			INT0E:XBR1.2			
T1	●	●	●	●		●																			T1E:XBR1.3			
/INT1	●	●	●	●		●																			INT1E:XBR1.4			
T2	●	●	●	●		●																			T2E:XBR1.5			
T2EX	●	●	●	●		●																			T2EXE:XBR1.6			
T4	●	●	●	●		●																			T4E:XBR2.3			
T4EX	●	●	●	●		●																			T4EXE:XBR2.4			
/SYSCLK	●	●	●	●		●																			SYSCKE:XBR1.7			
CNVSTR	●	●	●	●		●																			CNVSTE:XBR2.0			
	ALE	/RD	/WR		AIN1.0/A8	AIN1.1/A9	AIN1.2/A10	AIN1.3/A11	AIN1.4/A12	AIN1.5/A13	AIN1.6/A14	AIN1.7/A15	A8m/A0	A9m/A1	A10m/A2	A11m/A3	A12m/A4	A13m/A5	A14m/A6	A15m/A7	AD0/D0	AD1/D1	AD2/D2	AD3/D3	AD4/D4	AD5/D5	AD6/D6	AD7/D7
	AIN1 输入/非复用地址	复用地址高/非复用地址	低		高																							



寄存器XBR0、XBR1 和XBR2用于为数字I/O资源分配物理I/O引脚。

## XBR0：端口I/O交叉开关寄存器0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CP0E	ECI0E		PCA0ME		UART0EN	SPI0EN	SMB0EN	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址:

位 7: CP0E: 比较器 0 输出使能位

0: CP0 不连到端口引脚。

1: CP0 连到端口引脚。

位 6: ECI0E: PCA0 外部计数器输入使能位

0: PCA0 外部计数器输入不连到端口引脚。

1: PCA0 外部计数器输入连到端口引脚。

位 5-3: PCA0ME: PCA0 模块 I/O 使能位

000: 所有的 PCA0 I/O 都不连到端口引脚。

001: CEX0 连到端口引脚。

010: CEX0、CEX1 连到 2 个端口引脚。

011: CEX0、CEX1、CEX2 连到 3 个端口引脚。

100: CEX0、CEX1、CEX2、CEX3 连到 4 个端口引脚。

101: CEX0、CEX1、CEX2、CEX3、CEX4 连到 5 个端口引脚。

110: 保留。

111: 保留



R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CP0E 位7	ECI0E 位6	位5	PCA0ME 位4	位3	UART0EN 位2	SPI0EN 位1	SMB0EN 位0	00000000 SFR地址:

位 2: UART0EN: UART0 I/O 使能位。

0: UART0 I/O 不连到端口引脚。

1: UART0 的 TX 连到 P0.0, RX 连到 P0.1。

位 1: SPI0EN: SPI 总线 I/O 使能位。

0: SPI0 I/O 不连到端口引脚。

1: SPI0 的 SCK、MISO、MOSI 和 NSS 连到 4 个端口引脚。

位 0: SMB0EN: SMBus 总线 I/O 使能位

0: SMBus0 I/O 不连到端口引脚。

1: SMBus0 的 SDA 和 SCL 连到 2 个端口引脚。



## XBR1：端口I/O交叉开关寄存器1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
SYSCKE	T2EXE	T2E	INT1E	T1E	INT0E	T0E	CP1E	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：

位 7: SYSCKE: /SYSCLK 输出使能位  
 0: /SYSCLK 不连到端口引脚。  
 1: /SYSCLK 连到端口引脚。

位 6: T2EXE: T2EX 使能位  
 0: T2EX 不连到端口引脚。  
 1: T2EX 连到端口引脚。

位 5: T2E: T2 使能位  
 0: T2 不连到端口引脚。  
 1: T2 连到端口引脚。

位 4: INT1E: /INT1 使能位。  
 0: /INT1 不连到端口引脚。  
 1: /INT1 连到端口引脚。

位 3: T1E: T1 使能位  
 0: T1 不连到端口引脚。  
 1: T1 连到端口引脚。

位 2: INT0E: /INT0 使能位  
 0: /INT0 不连到端口引脚。  
 1: /INT0 连到端口引脚。

位 1: T0E: T0 使能位  
 0: T0 不连到端口引脚。  
 1: T0 连到端口引脚。

位 0: CP1E: 比较器 1 输出使能位  
 0: CP1 不连到端口引脚。  
 1: CP1 连到端口引脚。



## XBR2：端口I/O交叉开关寄存器2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
WEAKPUD	XBARE	-	T4EXE	T4E	UART1E	EMIFLE	CNVSTE	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址:

位 7: **WEAKPUD:** 弱上拉禁止位。

0: 弱上拉全局使能。

1: 弱上拉全局禁止。

位 6: **XBARE:** 交叉开关使能位

0: 交叉开关禁止。端口 0、1、2 和 3 的所有引脚被强制为输入方式。

1: 交叉开关使能。

位 5: 未用。读=0, 写=忽略。

位 4: **T4EXE:** T4EX 输入使能位

0: T4EX 不连到端口引脚。

1: T4EX 连到端口引脚。

位 3: **T4E:** T4 输入使能位

0: T4 不连到端口引脚。

1: T4 连到端口引脚。



R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
WEAKPUD	XBARE	-	T4EXE	T4E	UART1E	EMIFLE	CNVSTE	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址:

位 2:      UART1E: UART1 I/O 使能位

0: UART1 I/O 不连到端口引脚。

1: UART1 TX 和 RX 连到两个端口引脚。

位 1:      EMIFLE: 外部存储器接口低端口使能位

0: P0.7、P0.6 和 P0.5 的功能由交叉开关或端口锁存器决定。

1: 如果 EMIOCF.4 = ‘0’ (外部存储器接口为复用方式)

则 P0.7 (/WR)、P0.6 (/RD)和 P0.5 (/ALE)被交叉开关跳过，它们的输出状态由端口锁存器和外部存储器接口决定。

1: 如果 EMIOCF.4 = ‘1’ (外部存储器接口为非复用方式)

则 P0.7 (/WR)和 P0.6 (/RD)被交叉开关跳过，它们的输出状态由端口锁存器和外部存储器接口决定。

位 0:      CNVSTE: 外部转换启动输入使能位

0: CNVSTR 不连到端口引脚。

1: CNVSTR 连到端口引脚。

交叉开关寄存器被正确配置后，通过将XBARE (XBR2.6) 设置为逻辑‘1’来使能交叉开关。



## 1. 配置端口引脚的输出方式

每个端口引脚的输出方式都可被配置为**漏极开路或推挽方式**，缺省状态为漏极开路。

**端口0-3 引脚的输出方式由PnMDOUT 寄存器中的对应位决定。**

图 17.11 P0MDOUT：端口 0 输出方式寄存器

R/W	复位值 00000000 SFR地址： 0xA4							
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-0: P0MDOUT.[7:0]: 端口 0 输出方式位。  
0: 端口引脚的输出方式为漏极开路。  
1: 端口引脚的输出方式为推挽。

注: 当 SDA、SCL、RX0 (当 UART0 工作于方式 0 时) 和 RX1 (当 UART1 工作于方式 0 时) 出现在端口引脚时, 总是被配置为漏极开路输出。

例: P3MDOUT.7为逻辑'1'时将P3.7 配置为**推挽方式**;

P3MDOUT.7为逻辑'0'时将P3.7 配置为**漏极开路方式**。



## 2. 配置端口引脚为数字输入

通过设置输出方式为“**漏极开路**”并向端口数据寄存器中的相应位写‘1’将端口引脚配置为数字输入。

**例如：**

设置P3MDOUT.7 为逻辑‘0’，并设置P3.7 为逻辑‘1’  
即可将P3.7 配置为数字输入。

如果一个端口引脚被交叉开关分配给某个数字外设，并且该引脚的功能为输入（例如UART0的接收引脚RX0），则该引脚的输出驱动器被自动禁止。



### 3. 外部中断 (IE6和IE7)

- 除了外部中断/INT0和/INT1外，P3.6和P3.7可以被配置为边缘触发的中断源。
- 用IE6CF (P3IF.2) 和IE7CF (P3IF.3) 位可以将这两个中断源配置为下降沿或上升沿触发
- 当检测到P3.6和P3.7有下降沿或者上升沿发生时，P3IF寄存器中对应的外部中断标志将被置1。如果对应的中断被允许，将会产生一个中断，CPU将转向对应的中断向量地址



## 4. 弱上拉

- 每个端口引脚都有一个弱上拉部件，在引脚与VDD之间提供阻性连接（约 $100\text{k}\Omega$ ），在默认情况下该上拉部件被使能。
- 弱上拉可以被总体禁止，通过向弱上拉禁止位（WEAKUP，XBR2.7）写“1”实现
- 当任何引脚被驱动为逻辑“0”时，弱上拉自动取消；即输出引脚不能与其自身的上拉部件冲突
- 对于端口1的引脚，将引脚配置为模拟输入时，上拉部件也被禁止



## 5. 配置端口1的引脚为模拟输入 (AIN.[7:0])

端口1的引脚可以用作ADC1模拟多路开关的模拟输入。通过向P1MDIN寄存器中的对应位写‘0’即可将端口引脚配置为模拟输入（只有P1可以，P0、P2~P4不可以）。缺省情况下端口引脚为数字输入方式。

图 17.13 P1MDIN：端口 1 输入方式寄存器

R/W	复位值 11111111							
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xBD

位 7-0： P1MDIN.[7:0]：端口 1 输入方式位。  
 0： 端口引脚被配置为模拟输入方式。数字输入通路被禁止（读端口位将总是返回‘0’）。引脚的弱上拉被禁止。  
 1： 端口引脚被配置为数字输入方式。读端口位将返回引脚的逻辑电平。弱上拉状态由 WEAKPUD 位 (XBR2.7, 见图 17.9) 决定。

**注意：**被配置为模拟输入的引脚所对应的P1MDOUT位应被设置为逻辑‘0’（漏极开路方式），对应的端口数据位应被设置为逻辑‘1’（高阻态）。



## I/O引脚初始化—输出：推挽还是开漏；输入：模拟还是数字

### 1. 数字输出

- 设置输出方式为“推挽”

### 2. 数字输入（计数器）

- 设置输出方式为“漏极开路”，逻辑0
- 向端口数据寄存器中的相应位写‘1’

### 3. 模拟输入（ADC）

- 设置输出方式为“漏极开路”，逻辑0
- 向端口数据寄存器中的相应位写‘1’
- 向P1MDIN 寄存器中的对应位写‘0’

例如：

设置 $P3MDOUT.7 = 1$ , 即可将P3.7 配置为数字推挽输出。 ( $P3MDOUT = 0x80$ )

例如：

设置 $P3MDOUT.7= 0$ , 且 $P3.7 =1$ , 即可将P3.7 配置为数字输入。

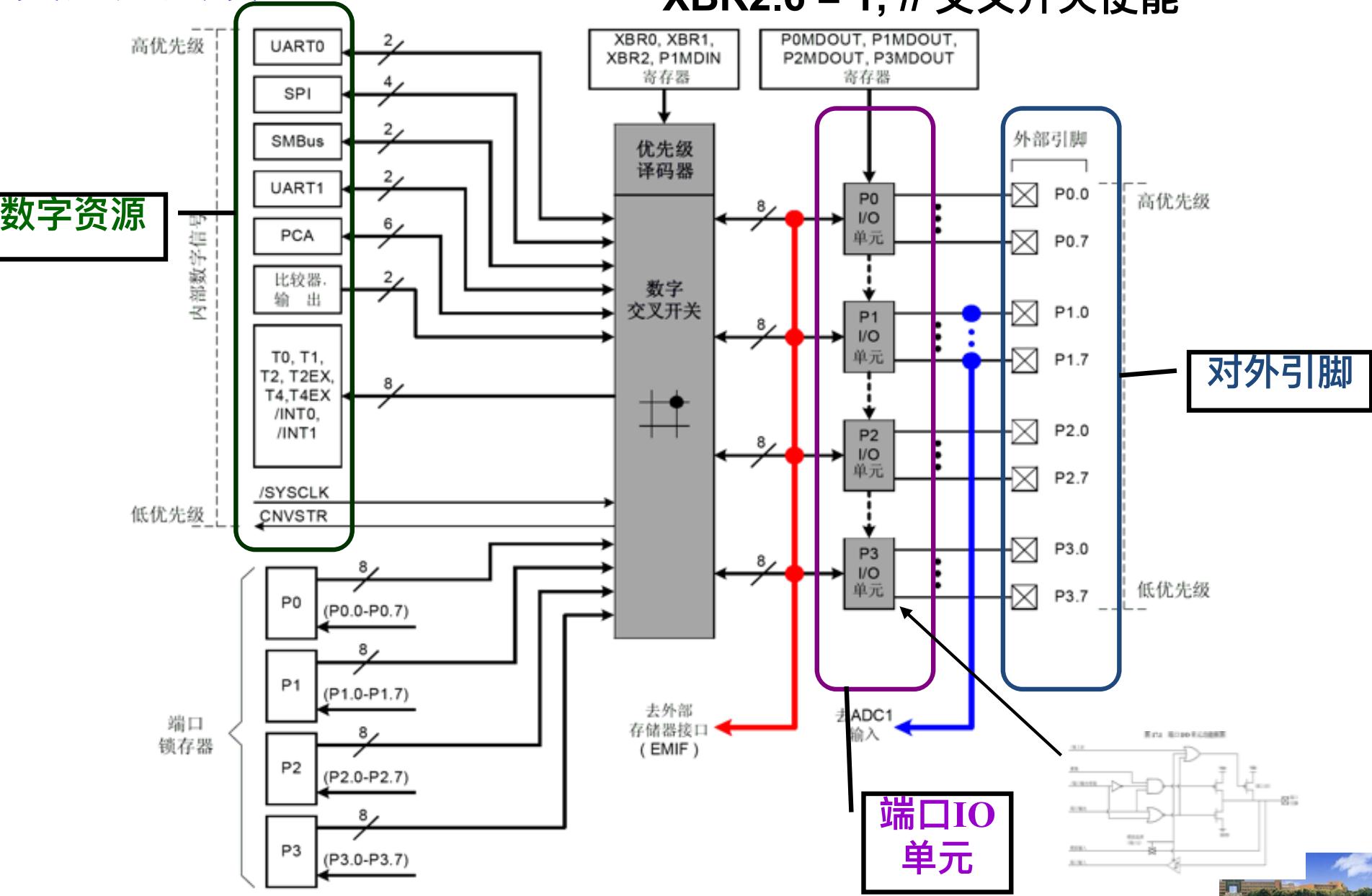
例如：

设置 $P1MDOUT.7= 0$ , 且 $P1.7 =1$ , 且 $P1MDIN.7 = 0$ , 即可将P1.7 配置为模拟输入。



# 总结 (2/2)

## 交叉开关设置



## 6. 外部存储器接口引脚分配

- 如果外部存储器接口（EMIF）被设置在低端口，  
EMIFLE（XBR2.1）位应被置“1”，以使交叉开关不  
将P0.7（/WR）、P0.6（/RD）和P0.5（/ALE）（复  
用方式）分配给外设。
- 如果外部存储器接口被设置在低端口并且发生一次片  
外MOVX操作，则在该MOVX指令执行期间外部存储  
器接口将控制有关端口引脚的输出状态，而不管交叉  
开关寄存器和端口数据寄存器的配置如何。
- 端口引脚的配置输出不受EMIF操作的影响，但读操  
作将禁止数据总线上的输出驱动器。



# 复用方式时的交叉开关译码表

不能分配

(EMIFILE = 1; EMIF 工作在复用方式; P1MDIN = 0xFF)

引脚 I/O	P0		P1		P2		P3		交叉开关寄存器位												
	0	1	2	3	4	5	6	7													
TX0	●																				
RX0		●							UART0EN:XBR0.2												
SCK	●	●																			
MISO		●	●																		
MOSI		●		●																	
NSS		●	●	●		●			SPI0EN:XBR0.1												
SDA	●	●	●	●																	
SCL		●	●	●																	
TX1	●	●	●	●																	
RX1		●	●	●					UART1EN:XBR2.2												
CEX0	●	●	●	●																	
CEX1		●	●	●																	
CEX2		●		●																	
CEX3			●	●																	
CEX4			●						PCA0ME:XBR0.[5:3]												
ECI	●	●	●	●					ECI0E:XBR0.6												
CP0	●	●	●	●					CP0E:XBR0.7												
CP1	●	●	●	●					CP1E:XBR1.0												
T0	●	●	●	●					T0E:XBR1.1												
/INT0	●	●	●	●					INT0E:XBR1.2												
T1	●	●	●	●					T1E:XBR1.3												
/INT1	●	●	●	●					INT1E:XBR1.4												
T2	●	●	●	●					T2E:XBR1.5												
T2EX	●	●	●	●					T2EXE:XBR1.6												
T4	●	●	●	●					T4E:XBR2.3												
T4EX	●	●	●	●					T4EXE:XBR2.4												
/SYSCLK	●	●	●	●					SYSCKE:XBR1.7												
CNVSTR	●	●	●	●					CNVSTE:XBR2.0												
	ALE	RD	WR	AIN1.0/A8	AIN1.1/A9	AIN1.2/A10	AIN1.3/A11	AIN1.4/A12	AIN1.5/A13	AIN1.6/A14	AIN1.7/A15	AD0/D0	AD1/D1	AD2/D2	AD3/D3	AD4/D4	AD5/D5	AD6/D6	AD7/D7		
				AIN1 输入/非复用地址高	AIN1 输入/非复用地址低	复用地址高/非复用地址高	复用地址高/非复用地址低	复用地址低/非复用地址高	复用地址低/非复用地址低	复用地址高/非复用地址高	复用地址高/非复用地址低	复用地址低/非复用地址高	复用地址低/非复用地址低	复用地址高/非复用地址高	复用地址高/非复用地址低	复用地址低/非复用地址高	复用地址低/非复用地址低	复用地址高/非复用地址高	复用地址高/非复用地址低	复用地址低/非复用地址高	复用地址低/非复用地址低



# 非复用方式时的交叉开关译码表

不能分配

(EMIFLE = 1; EMIF 工作在非复用方式; P1MDIN = 0xFF)

引脚 I/O	P0								P1								P2								P3								交叉开关寄存器位
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
TX0	●																														UART0EN:XBR0.2		
RX0		●																															
SCK	●	●																													SPI0EN:XBR0.1		
MISO		●	●																														
MOSI		●	●																														
NSS		●	●																														
SDA	●	●	●	●	●	●	●	●																						SMB0EN:XBR0.0			
SCL		●	●	●	●	●	●	●																									
TX1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●														UART1EN:XBR2.2			
RX1		●	●	●	●	●	●	●																									
CEX0	●																																
CEX1		●																															
CEX2			●																														
CEX3				●																													
CEX4					●																												
ECI	●	●	●	●	●	●	●	●																						ECI0E:XBR0.6			
CP0	●	●	●	●	●	●	●	●																						CP0E:XBR0.7			
CP1	●	●	●	●	●	●	●	●																						CP1E:XBR1.0			
T0	●	●	●	●	●	●	●	●																						T0E:XBR1.1			
/INT0	●	●	●	●	●	●	●	●																						INT0E:XBR1.2			
T1	●	●	●	●	●	●	●	●																						T1E:XBR1.3			
/INT1	●	●	●	●	●	●	●	●																						INT1E:XBR1.4			
T2	●	●	●	●	●	●	●	●																						T2E:XBR1.5			
T2EX	●	●	●	●	●	●	●	●																						T2EXE:XBR1.6			
T4	●	●	●	●	●	●	●	●																						T4E:XBR2.3			
T4EX	●	●	●	●	●	●	●	●																						T4EXE:XBR2.4			
/SYSCLK	●	●	●	●	●	●	●	●																						SYSCKE:XBR1.7			
CNVSTR	●	●	●	●	●	●	●	●																						CNVSTE:XBR2.0			
	ALE	RD	WR	AIN1.0/A8	AIN1.1/A9	AIN1.2/A10	AIN1.3/A11	AIN1.4/A12	AIN1.5/A13	AIN1.6/A14	AIN1.7/A15	A8m/A0	A9m/A1	A10m/A2	A11m/A3	A12m/A4	A13m/A5	A14m/A6	A15m/A7	AD0/DO	AD1/D1	AD2/D2	AD3/D3	AD4/D4	AD5/D5	AD6/D6	AD7/D7						
	AIN1 输入/非复用地址高								复用地址高/非复用地址低								复用数据/非复用数据																



- 端口4-7 的所有端口引脚都可用作通用I/O (GPIO) , 通过读和写相应的端口数据寄存器访问每个端口，这些端口数据寄存器是一组按字节寻址的特殊功能寄存器。与端口0-3一样有：
  - 1. 配置端口引脚的输出方式
  - 2. 配置端口引脚为数字输入
  - 3. 弱上拉
  - 4. 外部存储器接口



2.1 C8051F系列单片机总体结构

2.2 C8051F020存储器组织

2.3 中断系统

2.4 端口输入/输出

2.5 电源管理方式

2.6 复位与时钟



## 2.5 电源管理方式

- IP-51 有两种可软件编程的电源管理方式：
- **空闲方式（等待方式）**  
CPU 停止运行，而外设和时钟处于活动状态。 ( $10\mu A \sim 5mA$ )
- **停机方式（掉电方式）**  
CPU 停止运行，所有的中断和定时器（时钟丢失检测器除外）都处于非活动状态，系统时钟停止。 ( $0.2\mu A$ )



- 功耗 =  $CV^2f$
- 其中： C – CMOS的负载电容；
- V – 电源电压
- f – 系统时钟的频率
- 功率大小与系统的电压高低、系统时钟频率高低成正比。一个低功耗的设计应尽量采用最低的电源电压、最低的系统频率，并尽可能地使用电源管理方式，以最大限度地节省功耗。



# 电源控制寄存器 (PCON)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
<b>SMOD0</b>	<b>SSTAT0</b>	—	<b>SMOD1</b>	<b>SSTAT1</b>	—	<b>STOP</b>	<b>IDLE</b>	<b>00000000</b>
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0x87

- 位1: STOP: 停机方式选择。置1强制掉电
- 位0: IDLE: 空闲方式选择。置1进入空闲方式



## 2.5.1 空闲方式

- 空闲位IDLE (PCON.0) 置1，进入空闲方式。
  - 1 . 内部寄存器和存储器的数据不变。
  - 2 . 所有模拟和数字外设保持活动状态。
  - 3 . 有被允许的中断发生或/RST (系统复位) 有效将结束空闲方式。



## 2.5.2 停机方式

- 将停机方式选择位（PCON.1）置1，导致CIP-51 进入停机方式。
- 在停机方式，CPU和振荡器都被停止，实际上所有的数字外设都停止工作。
- 只有内部或外部复位能结束停机方式。



2.1 C8051F系列单片机总体结构

2.2 C8051F020存储器组织

2.3 中断系统

2.4 端口输入/输出

2.5 电源管理方式

2.6 复位与时钟



## • 2.6.1 复位源

### 1、复位状态

复位电路可以将控制器置于一个预定的状态——复位状态。

在进入复位状态时，将发生以下过程：

- CIP-51 CPU停止程序执行
- 特殊功能寄存器（SFR）被初始化为所定义的复位值
- 外部端口引脚被置于一个已知状态
- 中断和定时器被禁止

端口I/O锁存器的复位值为0xFF（全部为逻辑‘1’），  
外部I/O引脚处于高电平状态。复位之后弱上拉被使能。

复位后，SFR初始为预定值

内部数据储存器内的内容在复位过程中保持不变

堆栈指针复位，堆栈丢失



## 在退出复位状态时：

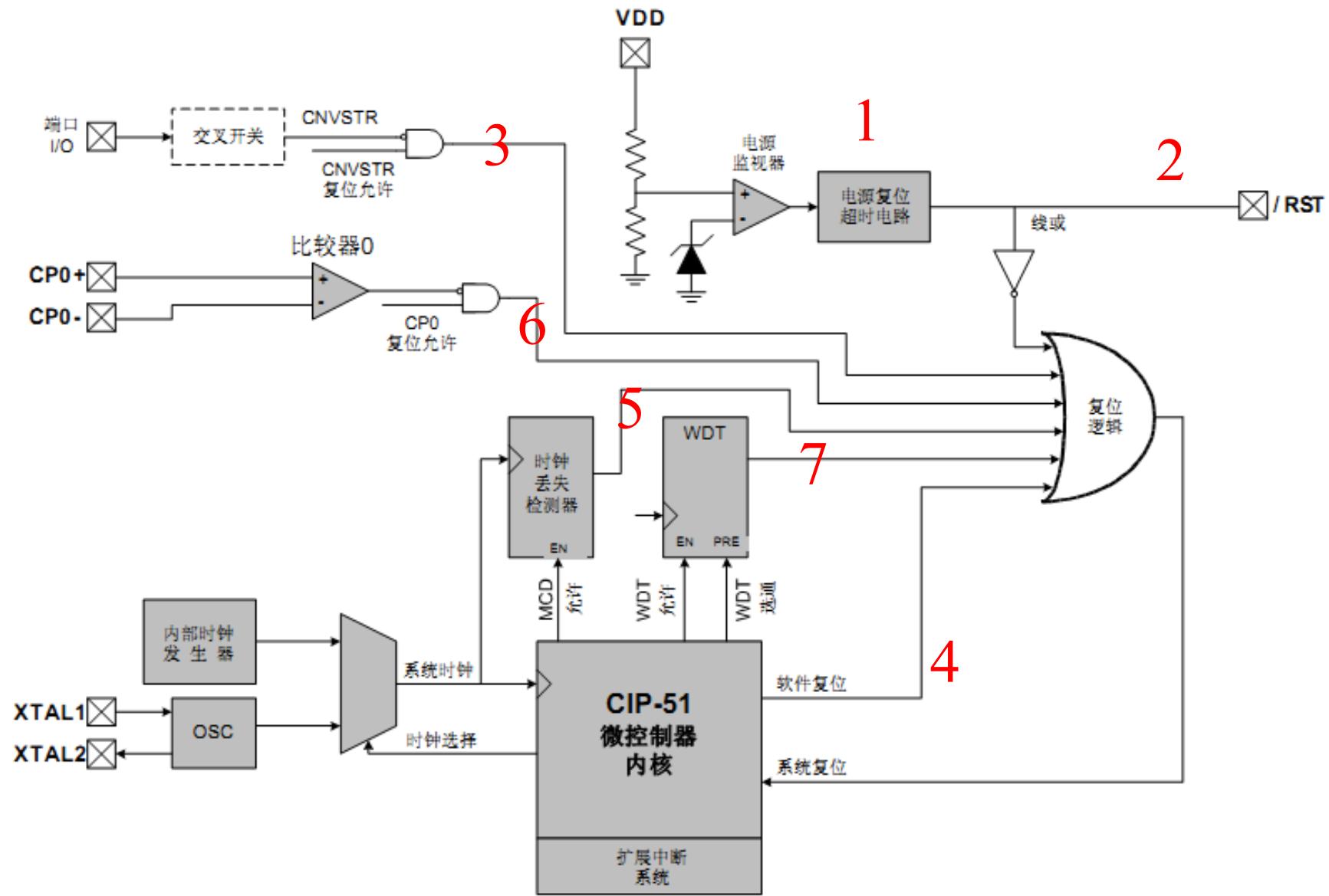
- 程序计数器（PC）被复位， $PC=0000H$ ；
- CIP-51使用内部振荡器作为默认的系统时钟,约为2MHz；
- 看门狗定时器被使能，用系统时钟的12分频作为其时钟源；
- 程序从地址0x0000开始执行。

## C8051F020的7个复位源：

- 1.上电/掉电复位
- 2.外部/RST引脚复位
- 3.外部CNVSTR信号复位
- 4.软件命令复位
- 5.比较器0复位
- 6.时钟丢失检测器
- 7.看门狗定时器超时复位



# 复位源框图



# RSTSRC: 复位源寄存器

R	R/W	R/W	R/W	R	R	R/W	R
-	CNVRSE F	C0RSEF	SWRSE F	WDTRS F	MCDRS F	PORSF	PINRSF
位7	位6	位5	位4	位3	位2	位1	位0

位7: 保留

位6: CNVRSEF: 转换启动复位源使能和标志

位5: C0RSEF: 比较器0复位使能和标志

位4: SWRSF: 软件强制复位和标志

位3: WDTRSF: 看门狗定时器复位标志

位2: MCDRSF: 时钟丢失检测器标志

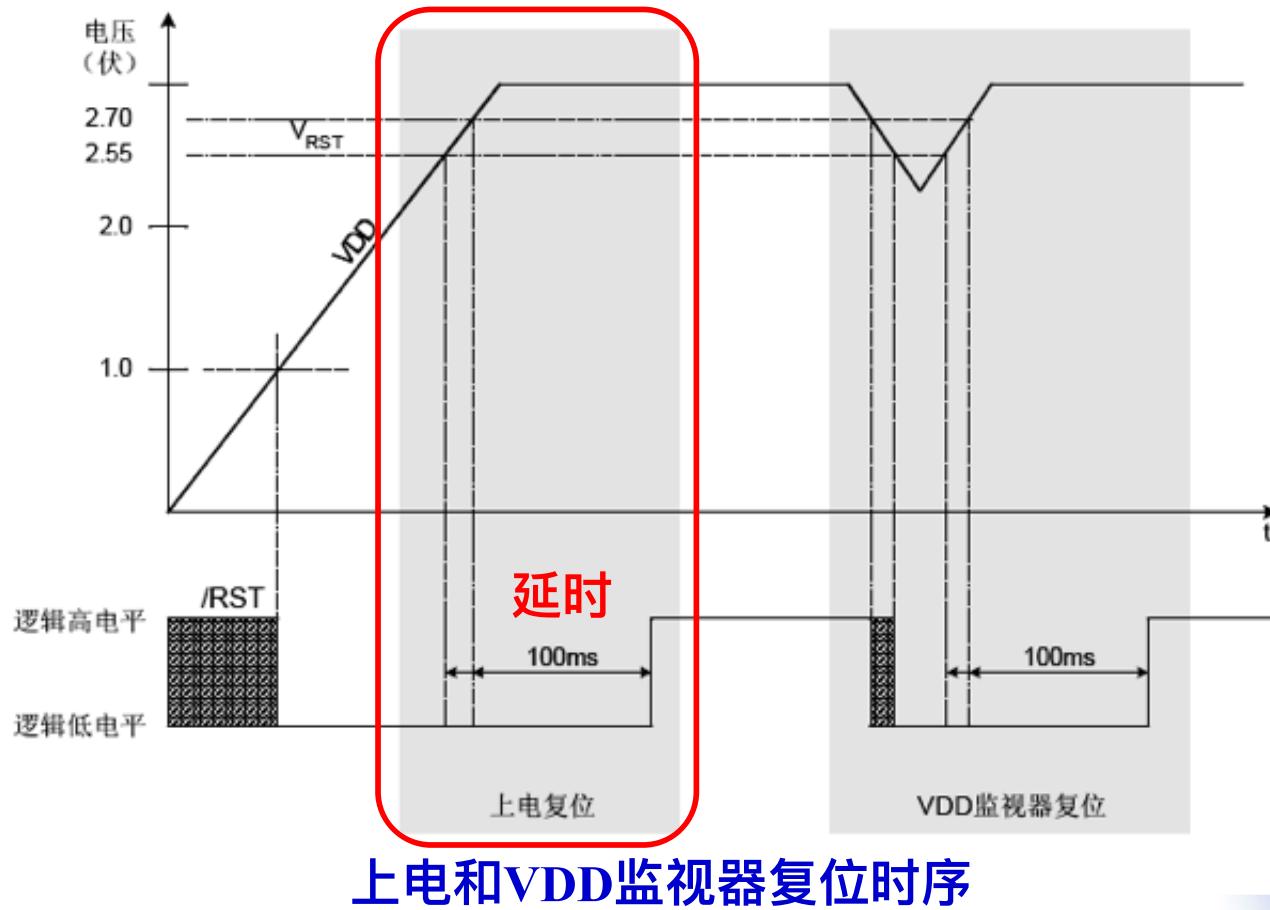
位1: PORSF: 强制上电复位和标志

位0: PINRSF: 硬件引脚复位标志

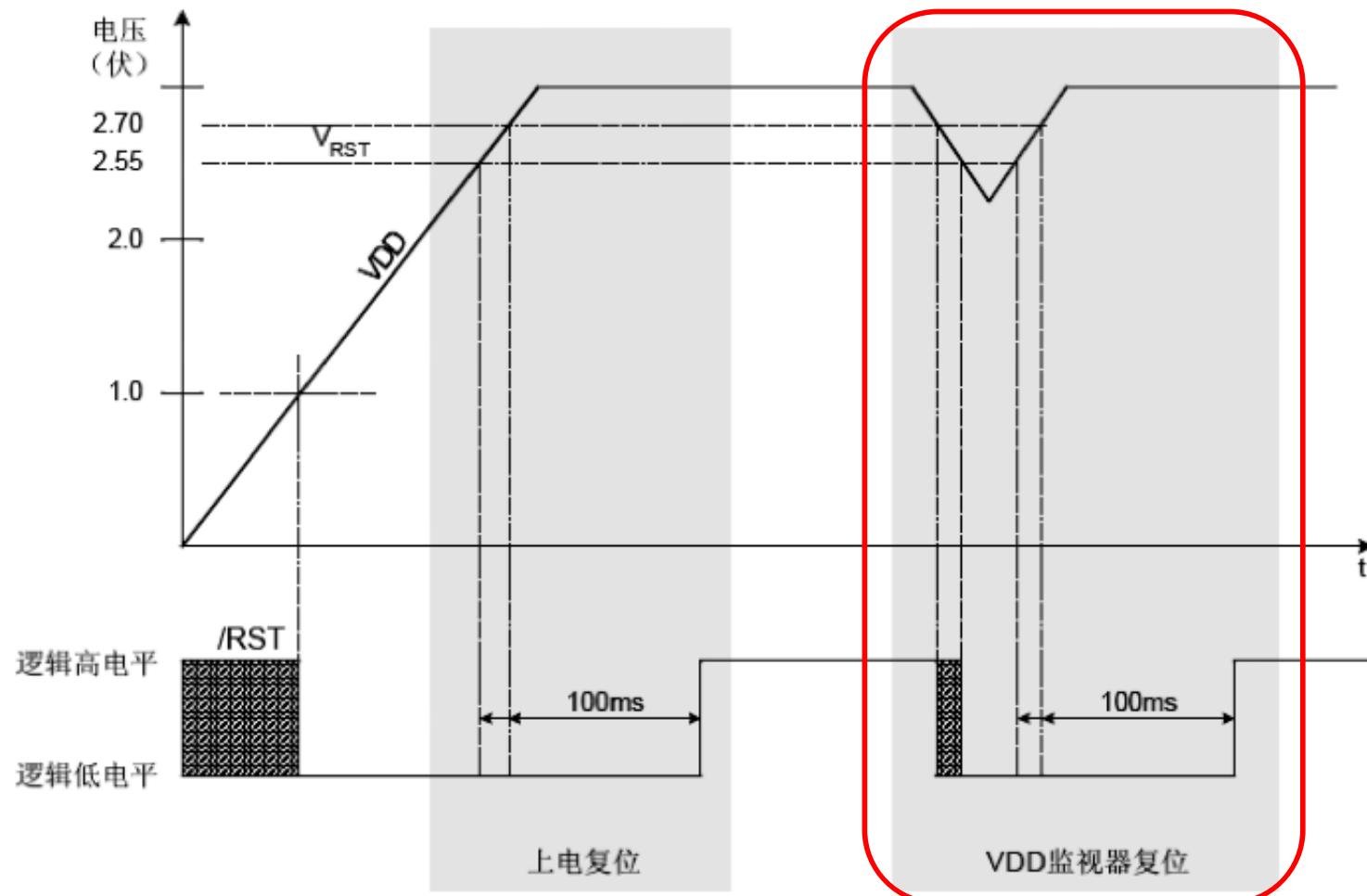


## 2.6.2 上电复位/掉电复位

**上电复位：**在上电期间，器件保持在复位状态，/RST引脚被驱动到低电平，直到 $V_{DD}$ 上升到超过 $V_{RST}$ 电平。从复位开始到退出复位状态要经过一个延时。



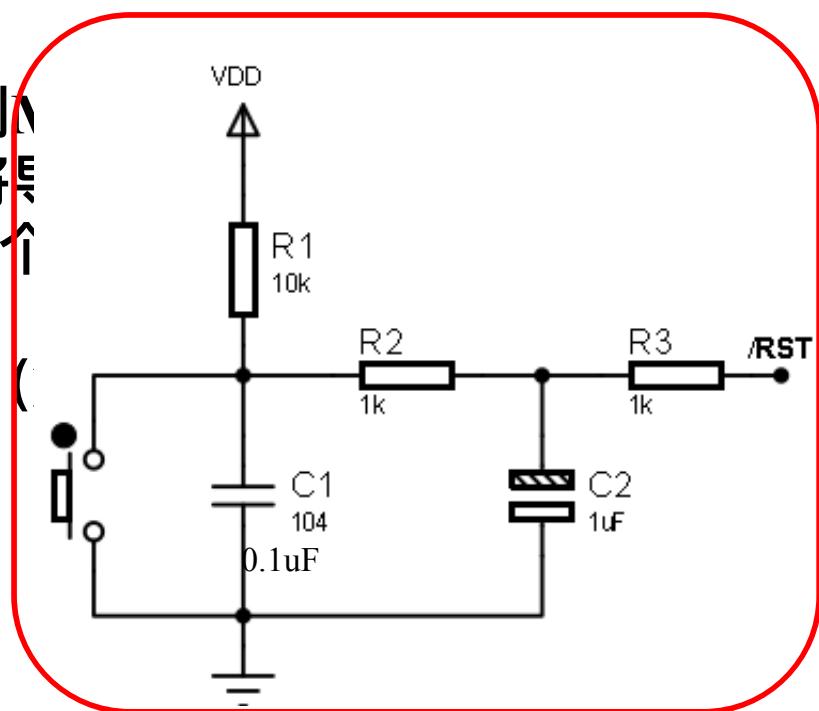
当发生掉电或因电源波动导致VDD降到VRST以下时，电源监视器将/RST引脚驱动为低电平并使CIP-51保持复位状态。当VDD又回到高于VRST的电平时，CIP-51将退出复位状态。



## 2.6.3 外部复位

### 1.外部复位引脚/RST复位：

- ✓ 外部/RST引脚提供了使用外部电路强制复位的功能。在/RST引脚上加一个低电平有效信号将使微控制器复位。
- ✓ 尽管在内部有弱上拉，但最好能提供一个去耦以防止强噪声引起复位。
- ✓ 从外部复位状态退出后，PINRSF标志被置‘1’。



### 2.外部CNVSTR引脚复位：

- ✓ 向CNVRSEF标志（RSTSRC.6）写1可将外部CNVSTR信号配置成复位源
- ✓ CNVSTR通过交叉开关配置，可配置给P0~P3任一引脚，低电平有效
- ✓ CNVSTR复位后，CNVRSEF标志被置‘1’。



## 2.6.4 内部复位

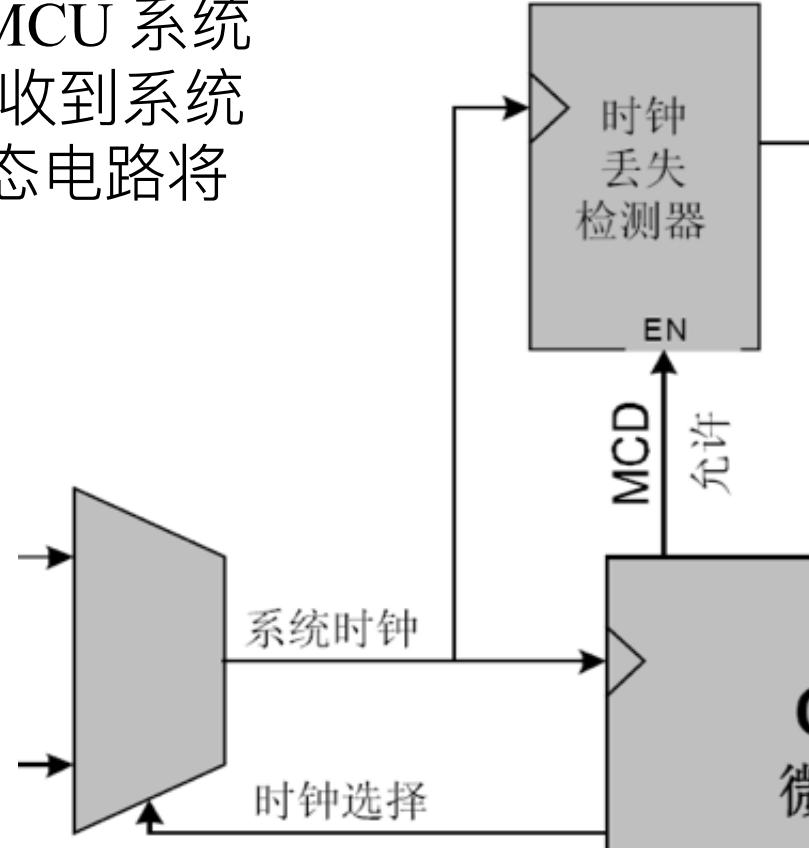
1.软件强制复位：向SWRSEF位写1 将强制产生一个上电复位。

2.时钟丢失检测器复位：

时钟丢失检测器实际上是由MCU 系统时钟触发的单稳态电路。如果未收到系统时钟的时间大于100 微秒，单稳态电路将超时并产生一个复位。

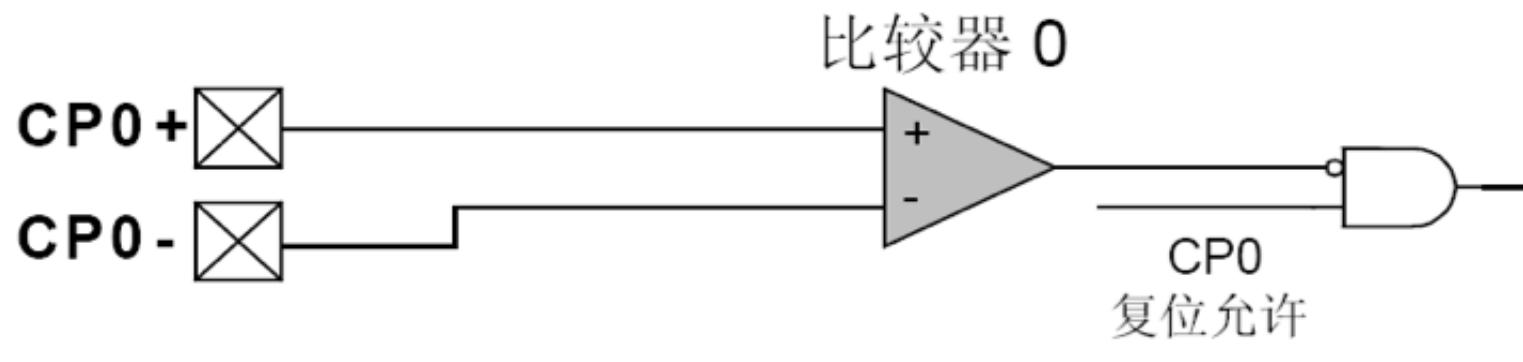
在发生时钟丢失检测器复位后，MCDRSF标志（RSTSRC.2）将被置‘1’，表示本次复位源为MSD；否则该位被清‘0’。

要使用该复位，应把内部振荡器控制特殊功能寄存器（OSCICN寄存器）中的MSCLKE位置1



## 比较器0复位：

比较器0 复位是低电平有效：如果同相端输入电压（CP0+引脚） 小于反相端输入电压（CP0-引脚），则MCU 被置于复位状态。



## 操作方法：

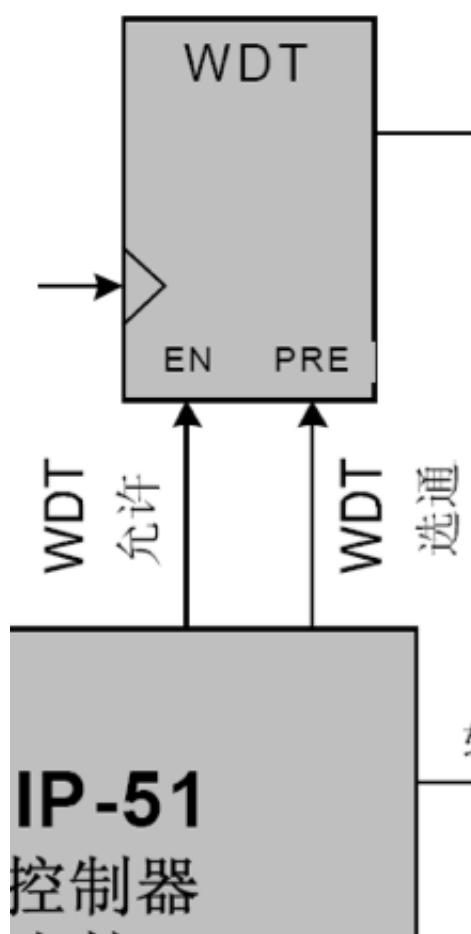
向C0RSEF 标志（RSTSRC.5）写‘1’可以将比较器0 配置为复位源。应在写C0RSEF之前用CPT0CN.7使能比较器0，以防止通电瞬间在输出端产生抖动，从而产生不希望的复位。



## 看门狗定时器复位：

MCU 内部有一个使用系统时钟的可编程看门狗定时器（WDT）。

当看门狗定时器溢出时，WDT 将强制CPU 进入复位状态。



为了防止复位，必须在溢出发生前由应用软件重新触发WDT。如果系统出现了软件/硬件错误，使应用软件不能重新触发WDT，则 WDT将溢出并产生一个复位，这可以防止系统失控。

在从任何一种复位退出时，WDT 被自动使能并使用缺省的最大时间间隔运行。系统软件可以根据需要禁止 WDT 或将其锁定为运行状态以防止意外产生的禁止操作。WDT 一旦被锁定，在下一次系统复位之前将不能被禁止。/RST 引脚的状态不受该复位的影响

## 使能/复位 WDT：

向 WDTCN 寄存器写入 0xA5 将使能并复位看门狗定时器。用户的应用软件应周期性地向WDTCN 写入 0xA5，以防止看门狗定时器溢出。每次系统复位都将使能并复位 WDT。

## 禁止 WDT：

向 WDTCN 寄存器写入 0xDE 后再写入 0xAD 将禁止 WDT。

```
EA = 0;           //禁止所有中断
WDTCN = 0xDE;    //禁止软件看门狗定时器
WDTCN = 0xAD;
EA = 1;           //重新允许中断
```

写 0xDE 和写 0xAD 必须发生在 4 个时钟周期之内，否则禁止操作将被忽略。在这个过程期间应禁止中断，以避免两次写操作之间有延时。



## 锁定WDT的禁止功能：

向 WDTCN 写入 0xFF 将使禁止功能无效。一旦锁定，在下一次复位之前禁止操作将被忽略。写 0xFF 并不使能或复位看门狗定时器。如果应用程序想一直使用看门狗，则应在初始化代码中向 WDTCN 写入 0xFF。

## 设置 WDT 定时间隔：

WDTCN.[2:0]控制看门狗超时间隔。超时间隔由下式给出：

$$4^{3+WDTCN[2:0]} \times T_{SYSCLK} ; \text{ (其中 } T_{SYSCLK} \text{ 为系统时钟周期)}$$

对于 2MHz 的系统时钟，超时间隔的范围是 0.032ms 到 524ms。在设置这个超时间隔时，WDTCN.7 必须为 0。读 WDTCN 将返回编程的超时间隔。  
**在系统复位后，WDT.[2:0]为 111b。**



# WDTCN: 看门狗定时器控制寄存器

R/W	复位值 xxxxx111 SFR地址: 0xFF							
位7	位6	位5	位4	位3	位2	位1	位0	

**位 7-0: WDT 控制**  
 写入 0xA5 将使能并重新装载 WDT。  
 写入 0xDE 后四个系统周期内写入 0xAD，将禁止 WDT。  
 写入 0xFF 将锁定禁止功能。

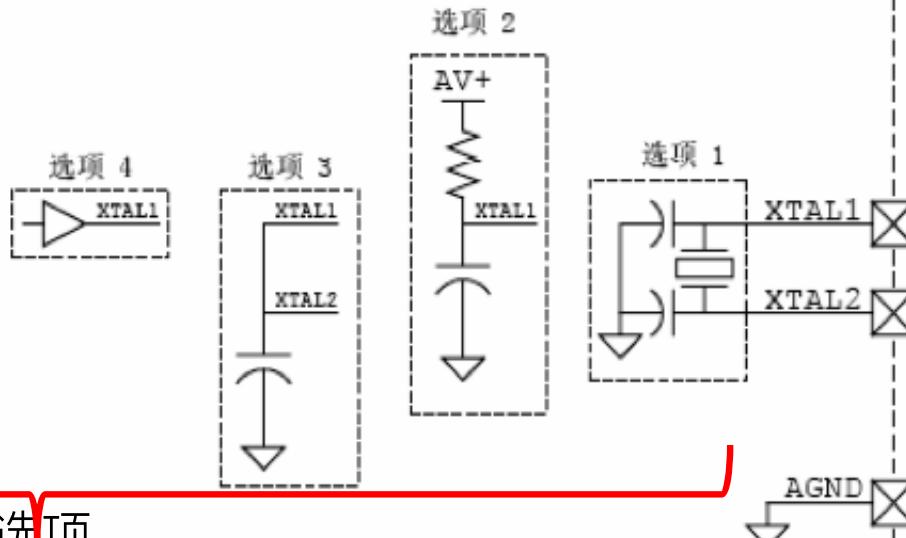
**位 4:** 看门狗状态位（读）  
 读 WDTCN.[4]得到看门狗定时器的状态。  
 0: WDT 处于不活动状态。  
 1: WDT 处于活动状态。

**位 2-0: 看门狗超时间隔位**  
 位 WDTCN.[2:0]设置看门狗的超时间隔。在写这些位时，WDTCN.7 必须被置为 ‘0’。



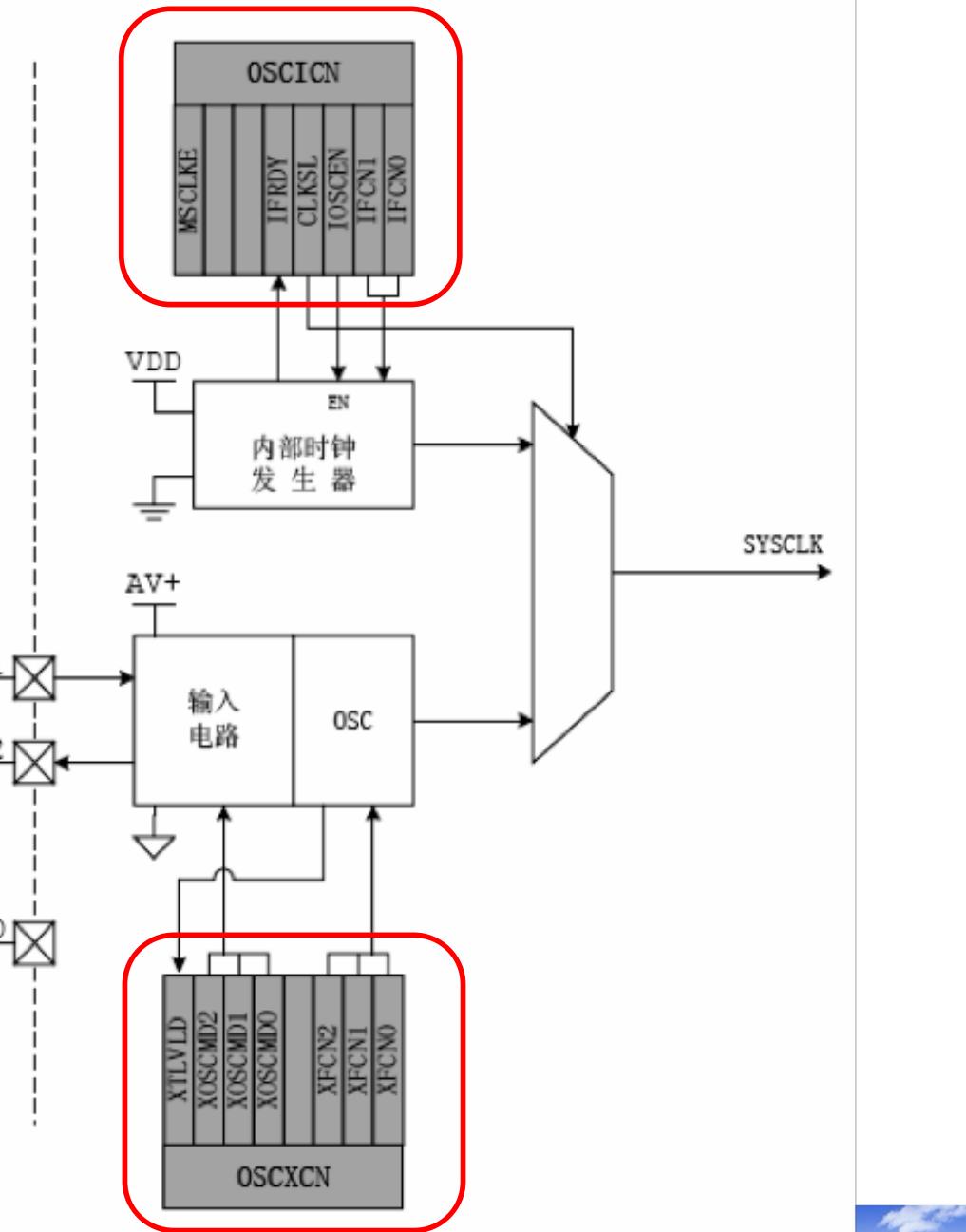
## 2.6.5 C8051F单片机的时钟系统

- 一个内部振荡器和一个外部振荡器电路，可自由切换；分别由OSCICN、OSCXCN两个SFR控制
- 选择内部振荡器时，外部振荡器可保持在允许状态，可避免系统时钟切换时的启动延时



选项

- 1: 外接晶体的电路连接
- 2: 外部RC连接
- 3: 外部电容方式连接
- 4: 外接CMOS时钟源的连接



- ◆ 复位后工作频率为2MHz
- ◆ 不使用外部振荡器时，XTAL1引脚应该外部接地（如图所示）或内部接地（通过将XOSCMOD位，即OSCXCN.[6~4]，设置为000）
- ◆ IFCN位（OSCICN.[1~0]）编程，可设置2MHz、4MHz、8MHz、16MHz四种频率
- ◆ 内部振荡器的精度较差，考虑工艺、供电电压以及温度等影响，内部振荡器的精度在±20%

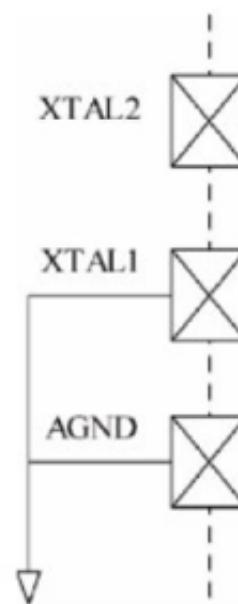


图 2-26 不用外部振荡器时  
XTAL1 引脚接地



图 14.2 OSCICN: 内部振荡器控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值 00010100 SFR地址: 0xB2
MSCLKE 位7	- 位6	- 位5	IFRDY 位4	CLKSL 位3	IOSCEN 位2	IFCN1 位1	IFCN0 位0	

- 位 7: **MSCLKE:** 时钟丢失检测器使能位  
 0: 禁止时钟丢失检测器。  
 1: 使能时钟丢失检测器; 检测到时钟丢失时间大于 100 微秒时将触发复位。
- 位 6-5: 未用。读=00b, 写=忽略。
- 位 4: **IFRDY:** 内部振荡器频率准备好标志  
 0: 内部振荡器频率不是按 IFCN 位指定的速度运行。  
 1: 内部振荡器频率按照 IFCN 位指定的速度运行。
- 位 3: **CLKSL:** 系统时钟源选择位  
 0: 选择内部振荡器作为系统时钟。  
 1: 选择外部振荡器作为系统时钟。
- 位 2: **IOSCEN:** 内部振荡器使能位  
 0: 内部振荡器禁止。  
 1: 内部振荡器使能。
- 位 1-0: **IFCN1-0:** 内部振荡器频率控制位  
 00: 内部振荡器典型频率为 2MHz。  
 01: 内部振荡器典型频率为 4MHz。  
 10: 内部振荡器典型频率为 8MHz。  
 11: 内部振荡器典型频率为 16MHz。

- ◆ 先配置外部振荡器寄存器OSCXCN
- ◆ 然后将CLKSL位（内部振荡器控制寄存器OSCICN.3）设置为逻辑1，选择外部振荡器作为时钟
- ◆ 外接晶体振荡器时，由于晶振启动需要几个毫秒，所有将CLKSL置1前需要等晶振有效标志位XTLVLD（OSCXCN.7）变成1

四种不同的配置

- 1: 外接晶体的电路连接
- 2: 外部RC连接

$$f_{osc} = (1.23 \times 1000) \left( \frac{1}{R \cdot C} \right)$$

- 3: 外部电容方式连接

$$f_{osc} = KF / (C \cdot AV_+)$$

- 4: 外接CMOS时钟源的连接

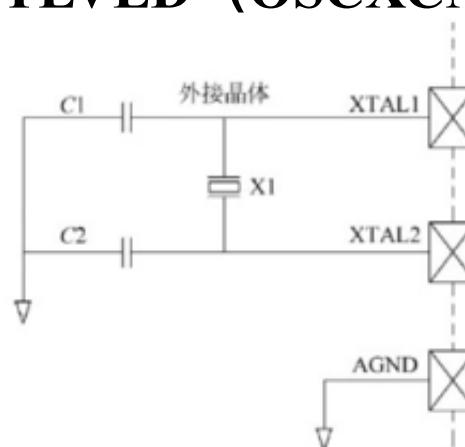


图 2-27 外接晶体的电路连接

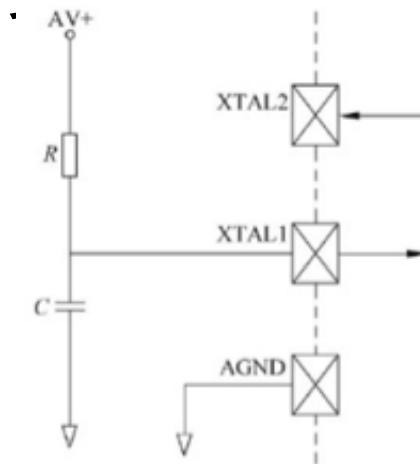


图 2-28 外部 RC 连接

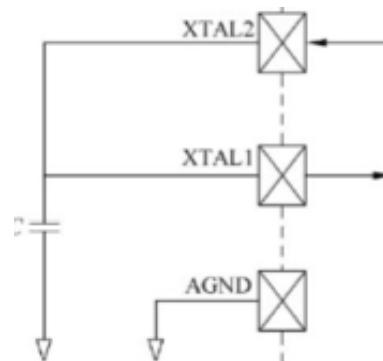


图 2-29 外部电容方式的连接

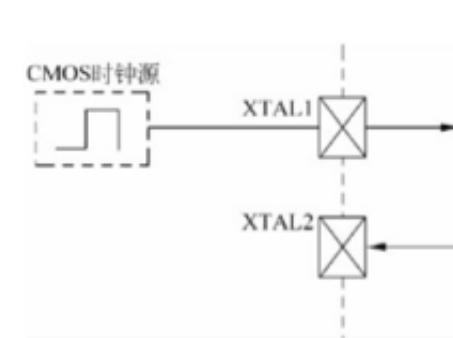


图 2-30 外接 CMOS 时钟源的连接



图 14.3 OSCXCN: 外部振荡器控制寄存器

TY

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
XTLVLD	XOSCMD2	XOSCMD1	XOSCMD0	-	XFCN2	XFCN1	XFCN0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xB1

位 7: **XTLVLD:** 晶体振荡器有效标志

(只在 XOSCMD=11x 时有效)

0: 晶体振荡器未用或未稳定。

1: 晶体振荡器正在运行并且工作稳定。

位 6-4: **XOSCMD2-0:** 外部振荡器方式位

00x: 关闭。XTAL1 引脚内部接地。

010: 系统时钟为来自 XTAL1 引脚的外部 CMOS 时钟。

011: 系统时钟为来自 XTAL1 引脚的外部 CMOS 时钟的二分频。

10x: RC/C 振荡器方式二分频。

110: 晶体振荡器方式。

111: 晶体振荡器方式二分频。

位 3: 保留。读 = 无定义, 写 = 忽略。



XFCN2-0：外部振荡器频率控制位。

000-111：见下表

XFCN	晶体 (XOSCMD=11x)	RC(XOSCMD=10x)	C(XOSCMD=10x)
000	$f \leq 12\text{kHz}$	$f \leq 25\text{kHz}$	K 因子= 0.44
001	$12\text{kHz} < f \leq 30\text{kHz}$	$25\text{kHz} < f \leq 50\text{kHz}$	K 因子= 1.4
010	$30\text{kHz} \leq f \leq 95\text{kHz}$	$50\text{kHz} \leq f \leq 100\text{kHz}$	K 因子= 4.4
011	$95\text{kHz} < f \leq 270\text{kHz}$	$100\text{kHz} < f \leq 200\text{kHz}$	K 因子= 13
100	$270\text{kHz} < f \leq 720\text{kHz}$	$200\text{kHz} < f \leq 400\text{kHz}$	K 因子= 38
101	$720\text{kHz} < f \leq 2.2\text{MHz}$	$400\text{kHz} < f \leq 800\text{kHz}$	K 因子= 100
110	$2.2\text{MHz} < f \leq 6.7\text{MHz}$	$800\text{kHz} < f \leq 1.6\text{MHz}$	K 因子= 420
111	$f > 6.7\text{MHz}$	$1.6\text{MHz} < f \leq 3.2\text{MHz}$	K 因子= 1400



当外部晶体振荡器稳定运行时，晶体振荡器有效标志（OSCXCN 寄存器中的XTLVLD）被硬件置‘1’。

XTLVLD 检测电路要求在使能振荡器工作和检测XTLVLD 之间至少有1 ms的启动时间。（在外部振荡器稳定之前就切换到外部振荡器可能导致不可预见的后果。）

建议的过程为：

1. 使能外部振荡器
2. 等待至少1ms
3. 查询XTLVLD == '1'
4. 将系统时钟切换到外部振荡器

```
//外部24MHz晶振初始化程序
void Oscillator_Init()
{
    int i = 0;
    OSCXCN = 0x67;
    for (i = 0; i < 3000; i++); //等1ms以上
    while ((OSCXCN & 0x80) == 0); //判断
    XTLVLD是否为1
    OSCICN = 0x0C; //CLKSL置位
}
```

