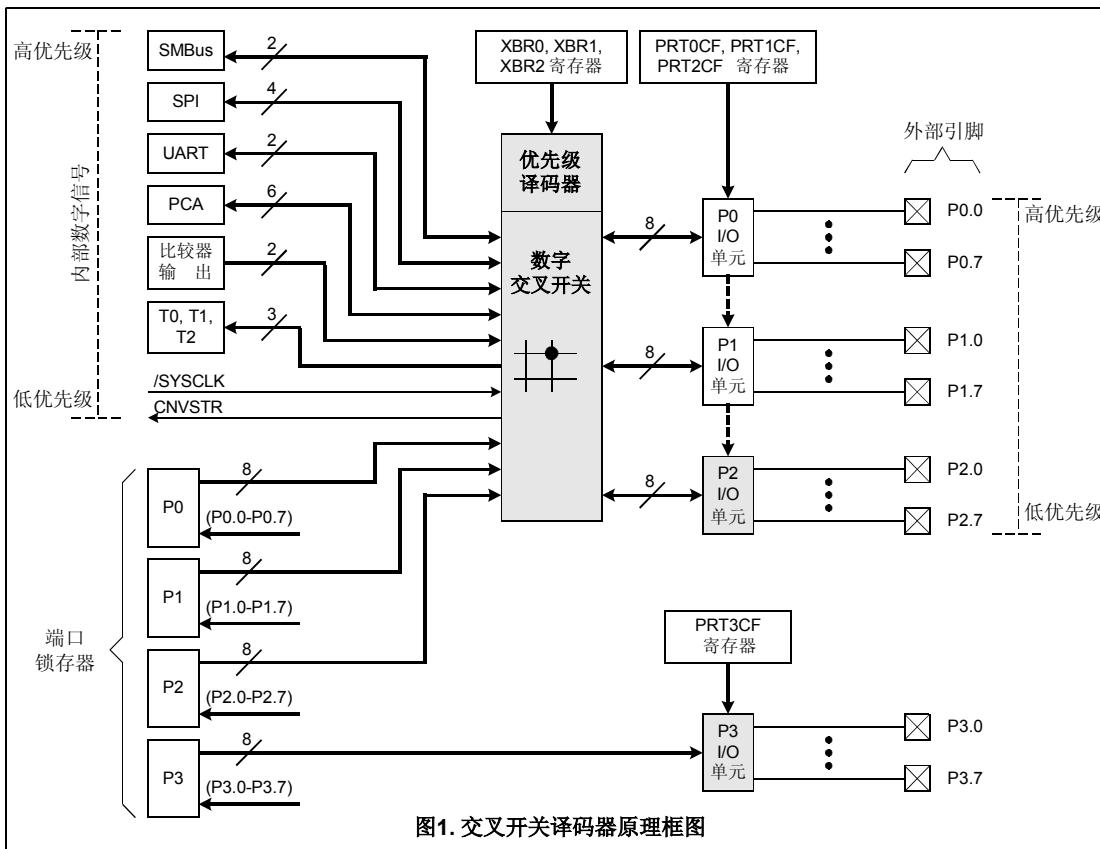


# C8051F MCU 应用笔记

## AN001 — 配置端口 I/O 交叉开关译码器



### 引言

本应用笔记的目的是说明如何配置和使用端口 I/O 交叉开关译码器。本文提供软件示例。

如图 1 所示，交叉开关是一个多路选择器，它用于为器件内部的硬件外设分配 I/O 端口。例如，它可以决定 UART 的 RXD 和 TXD 连到哪一个端口引脚。

交叉开关负责 SMBus、SPI、UART、定时器捕捉模块、外部 PCA 输入、比较器输出、定时器外部输入、/SYSCLK 以及 A/D 转换启动输入的引脚分配。交叉开关必须在访问这些外设的 I/O 之前被配置和允许。

未指定的端口引脚作为通用 I/O。

交叉开关提供了两个关键的系统特性：

# AN001 — 配置端口 I/O 交叉开关译码器

---

- 在端口 0、端口 1 和端口 2 上的所有未分配的通用 I/O 口引脚都被连续组合在一起。
- 对于引脚数量少的器件，它提供了外设选择的灵活性。外设选择只受限于可用的端口引脚数，而不受限于哪一个引脚是可用的。这就允许系统设计者在使用引脚数少的器件时能选择将哪些数字外设分配到器件上的数字 I/O 引脚。

## 关键点

- 为了使用端口 0、端口 1 或端口 2 的任何一个引脚作为输出，交叉开关必须被允许。
- 交叉开关译码器必须在任何一个数字外设被允许前配置。
- 交叉开关通常在复位时被配置一次，在复位处理过程的最开始处，以后不再进行配置。
- 交叉开关的设置改变器件的引脚分配。
- 每一种交叉开关设置导致唯一的器件引脚分配，如果你在交叉开关中允许或禁止外设，引脚分配将发生变化。
- 对于输出端口引脚，其输出方式（漏极开路或推挽）必须被显式设置，即使那些由交叉开关分配的端口引脚也是如此。例外情况是 SMBus 上的 SDA 和 SCL 及 UART 的 RXD 引脚，这些引脚被自动配置为漏极开路。
- 交叉开关分配的输入引脚（例如 NSS 或/INT0）是开路或推挽并不重要。这些引脚被配置为输入，而与相应端口配置寄存器的设置无关。为了将一个通用 I/O 引脚配置为输入，与这一引脚相关的端口配置寄存器位必须被清 0，这样选择该引脚为漏极开路输出方式。此外，与该引脚相关的端口位必须被置 1，这样使该引脚处于高阻态，或在 XBR2 中的 WEAKPUD 被清 0 时弱上拉为高电平，这是端口引脚的复位配置。
- 在任意时刻可以通过读取相应端口 SFR 得到端口引脚的电平值，而不管交叉开关寄存器的设置如何或引脚被配置为输入还是输出。
- 交叉开关寄存器中的允许位是独立的。与数字外设本身的允许位是分开的。
  - 外设部件在使用时不需要在交叉开关中被允许（例如，一个 PCA 模块即使在输出没有被接到引脚的情况下也能产生中断）。
  - 那些在交叉开关中被允许但在它们自己的 SFR 中被禁止的外设，仍然控制端口引脚。这就是说，端口引脚可以在任何时候被读取，但是输出被占用它的外设所控制，不能作为通用输出口来访问。
- 端口 1 上的四个外部中断（P1.[4..7]）是由引脚上的下降沿触发的，与下降沿触发源、交叉开关设置或端口引脚的输出方式无关。
- 与标准 8051 不同的是，C8051 提供了真正的推挽输出。如果需要 8051 的上拉功能，可以通过将相应的端口输出配置为“推挽”方式，然后再配置为“漏极开路”来进行软件仿真。

## 确定器件引脚分配

本节介绍如何使用优先权交叉开关译码表即表 4 来决定器件的引脚分配，这种分配基于交叉开关寄存器中的外设选择，这些选择在图 2—图 4 中列出。

为了确定引脚分配，首先根据所需要的外设来配置交叉开关寄存器。然后从优先权交叉开关译码表的顶部开始向下扫描，直到遇到第一个被允许的外设部件。该设备将使用 P0.0，如果需要多个引脚，就可以按顺序从 P0.1 向后分配。例如，如果 SPI 是第一个被允许的外设，则 SCK、MISO、MOSI 和 NSS 将分别被分配到 P0.0、P0.1、P0.2、P0.3。下一个被允许的外设将被分配到

---

# AN001 — 配置端口 I/O 交叉开关译码器

---

引脚 P0.4。所有未分配的引脚作为通用 I/O。

## 例 1

假设一个应用需要：

- SPI
- UART
- 两个捕捉模块
- /INT0
- T2。

参考图 2—图 4 中的 I/O 端口交叉开关寄存器说明，交叉开关寄存器配置如下：

XBR0=00010110b；允许 UART、两个捕捉模块和 SPI

XBR1=00100100b；允许 T2 和/INT0

XBR2=01000000b；允许交叉开关

从表 4 的顶部开始我们发现 SPI 引脚将占用 P0.[0..3]（因为不使用 SMBus），UART 占用引脚 P0.[4..5]，CEX0 占用 P0.6，CEX1 占用 P0.7，/INT0 占用 P1.0，T2 占用 P1.1。其它端口引脚 P1.[2..7] 和 P2.[0..7] 作为通用端口引脚。引脚分配情况列于表 1。

表 1. 例 1 的引脚分配

引脚	功能
P0.0	SCK
P0.1	MISO
P0.2	MOSI
P0.3	NSS
P0.4	TX
P0.5	RX
P0.6	CEX0
P0.7	CEX1
P1.0	/INT0
P1.1	T2
P1.[2..7]	GPIO
P2.[0..7]	GPIO

## 例 2

假设一个应用需要：

- UART
- /INT1
- /SYSCLK
- CNVSTR

交叉开关寄存器的配置如下：

# AN001 — 配置端口 I/O 交叉开关译码器

---

XBR0=00000100b; 允许 UART

XBR1=10010000b; 允许/INT1 和/SYSCLK

XBR2=01000001b; 允许 CNVSTR 和交叉开关

引脚分配情况列于表 2。

表 2. 例 2 的引脚分配

引脚	功能
P0.0	TX
P0.1	RX
P0.2	/INT1
P0.3	/SYSCLK
P0.4	CNVSTR
P0.5[5..7]	GPIO
P1[0..7]	GPIO
P2[0..7]	GPIO

## 例 3

假设一个应用需要下列外设之外的所有部件:

- 定时器捕捉模块
- SPI 端口
- T0、T1 和 T2 输入
- 比较器输出
- /SYSCLK。

也就是说需要下列部件:

- SMBus (I2C 口)
- UART
- ECI
- /INT0
- /INT1
- T2EX
- CNVSTR

为了允许以上部件，我们将交叉开关寄存器配置如下:

XBR0=01000101b; 允许 ECI、UART 和 SMBus

XBR1=01010100b; 允许/INT0、/INT1 和 T2EX

XBR2=01000001b; 允许 CNVSTR 和交叉开关

引脚分配情况列于表 3。

# AN001 — 配置端口 I/O 交叉开关译码器

---

表 3. 例 3 的引脚分配

引脚	功能
P0.0	SDA
P0.1	SCL
P0.2	TX
P0.3	RX
P0.4	ECI
P0.5	/INT0
P0.6	/INT1
P0.7	T2EX
P1.0	/CNVSTR
P1.[1..7]	GPIO
P2.[0..7]	GPIO

AN001 — 配置端口 I/O 交叉开关译码器

## 软件示例：

下面的代码段说明如何配置交叉开关寄存器。

```
; CYGNAL INTEGRATED PRODUCTS, INC.  
;  
;  
; 文件名: cross1.ASM  
; 目标MCU: C8051F000  
; 说明: 配置交叉开关的示例代码。  
;  
; 实现笔记:  
;  
;  
;  
-----  
;  
; EQUATES  
-----  
  
$MOD8F000  
;  
;  
; 全局变量及定义  
;  
;  
; 交叉开关端口引脚定义 (可选)  
SCK    EQU    P0.0  
MISO   EQU    P0.1  
MOSI   EQU    P0.2  
NSS    EQU    P0.3  
TX     EQU    P0.4  
RX     EQU    P0.5  
CEX0   EQU    P0.6  
CEX1   EQU    P0.7  
INT0   EQU    P1.0  
T2     EQU    P1.1  
;  
; 其它变量  
;  
;  
; 中断向量表  
;  
;  
; 复位向量  
    org    00h  
    ljmp   Reset  
;  
; 其它中断向量  
;  
;  
; 中断向量代码  
;  
;  
    org    0B3h
```

# AN001 — 配置端口 I/O 交叉开关译码器

```
Reset:
    Mov     XBR0, 00010110b ; 允许SPI、UART和两个外部捕捉模块
    Mov     XBR1, 00100100b ; 允许/INT0和T2
    Mov     XBR2, 01000000b ; 允许交叉开关和弱上拉
                           ; 端口引脚分配如下
                           ; P0.0 = SCK - 例：主模式
                           ; P0.1 = MISO
                           ; P0.2 = MOSI
                           ; P0.3 = NSS
                           ; P0.4 = TX
                           ; P0.5 = RX
                           ; P0.6 = CEX0 - 例：输出
                           ; P0.7 = CEX1 - 例：输入
                           ; P1.0 = /INT0
                           ; P1.1 = T2
                           ; P1.[2..7], P2.[0..7] 为GPIO
    mov     PRT0CF, #01010101b ; CEX0、TX、MOSI和SCK为推挽输出
    mov     PRT1CF, #00000000b ; T2和/INT0是由交叉开关分配的输入,
                           ; 因此与它们的端口配置值无关
    mov     PRT2CF, #00000000b ; 配置端口2全部为输入
    mov     P2, #11111111b    ; 为了将未被分配的通用I/O脚配置为输入,
                           ; 输出类型必须为漏极开路，且端口位必须被置1。

                           ; 其它配置代码
    ljmp  Main
;-----;
; 主程序代码
;-----;

Main:
    sjmp  $                      ; 原地跳转
; 此处为主程序代码
;-----;
; 结束
;-----;

END
```

表 4. 优先权交叉开关译码表

I/O 引脚	P0							P1							P2									
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
SDA	●																							
SCL		●																						
SCK	●		●																					
MISO		●		●																				

AN001 — 配置端口 I/O 交叉开关译码器

在优先权交叉开关译码表中，黑圆点（●）用于指示每个信号（行）可以被用户应用程序分配的外部端口I/O引脚（列）。

# AN001 — 配置端口 I/O 交叉开关译码器

图2. 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
CP0OEN	ECIE		PCA0ME		UARTEN	SPI0OEN	SMB0OEN	00000000 SFR 地址: <b>0xE1</b>

- 位7: CP0OEN: 比较器0的输出允许位  
0: CP0 不连到端口引脚。  
1: CP0 连到端口引脚。
- 位6: ECIE: PCA0 计数器输入允许位  
0: ECI 不连到端口引脚。  
1: ECI 连到端口引脚。
- 位5-3: PCA0ME: PCA模块I/O允许位  
000: 所有的PCA I/O都不连到端口引脚。  
001: CEX0 连到端口引脚。  
010: CEX0、CEX1 连到两个端口引脚。  
011: CEX0、CEX1、CEX2 连到三个端口引脚。  
100: CEX0、CEX1、CEX2、CEX3 连到四个端口引脚。  
101: CEX0、CEX1、CEX2、CEX3、CEX4 连到五个端口引脚。  
110: 保留  
111: 保留
- 位2: UARTEN: UART I/O 允许位  
0: UART I/O 不连到端口引脚。  
1: RX、TX 连到两个端口引脚。
- 位1: SPI0OEN: SPI Bus I/O 允许位  
0: SPI I/O 不连到端口引脚。  
1: MISO、MOSI、SCK和NSS连到四个端口引脚。
- 位0: SMB0OEN: SMBus Bus I/O允许位  
0: SMBus I/O不连到P0.0、P0.1。  
1: SDA 连到 P0.0, SCL 连到 P0.1。

# AN001 — 配置端口 I/O 交叉开关译码器

图3. 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	CP1OE	00000000

位7 位6 位5 位4 位3 位2 位1 位0

SFR 地址：  
**0xE2**

位7: SYSCKE: SYSCLK 输出允许位

0: SYSCLK不连到端口引脚。

1: SYSCLK输出连到端口引脚。

位6: T2EXE: T2EX允许位

0: T2EX不连到端口引脚。

1: T2EX连到端口引脚。

位5: T2E: T2允许位

0: T2不连到端口引脚。

1: T2连到端口引脚。

位4: INT1E: /INT1允许位

0: /INT1不连到端口引脚

1: /INT1连到端口引脚

位3: T1E: T1允许位

0: T1不连到端口引脚

1: T1连到端口引脚

位2: INT0E: /INT0允许位

0: /INT0不连到端口引脚。

1: /INT0连到端口引脚。

位1: T0E: T0允许位

0: T0不连到端口引脚。

1: T0连到引脚。

位0: CP1OE: 比较器1输出允许位

0: CP1不连到端口引脚。

1: CP1 连到端口引脚。

# AN001 — 配置端口 I/O 交叉开关译码器

图4. 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 位7	XBARE 位6	- 位5	- 位4	- 位3	- 位2	- 位1	CNVSTE 位0	00000000 SFR 地址: 0xE3

- 位7: WEAKPUD: 端口I/O弱上拉禁止位  
0: 弱上拉允许  
1: 弱上拉禁止  
位6: XBARE: 交叉开关允许位  
0: 交叉开关禁止  
1: 交叉开关允许  
位5-1: 未使用。读= 00000b, 写=无效  
位0: CNVSTE: ADC转换启动输入允许位  
0: CNVSTR不连到端口引脚。  
1: CNVSTR 连到端口引脚。

## XBR0、XBR1、XBR2用法示例:

当被选中时，数字资源按顺序（如优先权交叉开关译码表中所示，从上往下）分配端口I/O引脚，从P0.0到P0.7，然后从P1.0到P1.7，最后从P2.0到P2.7。如果数字资源没有被连到I/O端口引脚，则这些端口引脚按缺省状态对应于内部端口寄存器位。

- 例1: 如果XBR0 = 0x11, XBR1 = 0x00, XBR2 = 0x40:  
P0.0=SDA, P0.1=SCL, P0.2=CEX0, P0.3=CEX1, P0.4 – P2.7 分配给相应的端口I/O。  
例2: 如果XBR0 = 0x80, XBR1 = 0x04, XBR2 = 0x41:  
P0.0=CP0, P0.1=/INT0, P0.2 = CNVSTR, P0.3 – P2.7 分配给相应的端口 I/O。