

8-pin/6-pin 通用型单片机

◇ 1.概述

8P0801C 是一款内置 1K-Word OTP 和 48Byte SRAM 的通用型单芯片；仅 41 条等宽指令，非常容易学习和使用；除跳转等指令为 2 个机器周期外，其他的指令均单机器周期运行时间；内部集成 2 个 8 位的定时器，且带 4 路 PWM 输出，可输出溢出中断来实现多任务应用场景；同时芯片内部集成有高精度比较器，32768 实时时钟电路，灵活的端口配置上下拉及推挽或开漏输出等方式，可适应各种复杂的接口环境；该芯片采用标准的 CMOS 5V 工艺设计制造，可接入较宽的电压范围来使用；

◇ 2.特性

■ CPU

- 41 条精简指令
- 8 位的数据处理能力
- 最高 8M/2T 的运行速度

■ 存储器

- 1K OTP 存储空间
- 49 字节 SRAM

■ 时钟

- 内置的高精度高频时钟，最高可达 8MHz
- 内置的低频低功耗时钟，可直接输出给 WatchDog 单元使用
- 外接 32768 做精准计时使用

■ 复位机制

- 内置低电压复位电路 LVR
- WatchDog Reset(WDR)

■ 中断源

- PB0 设置的上升或下降沿变化触发的中断
- PB 口设置的电平变化所引起的中断
- Timer0 溢出引起的中断
- Timer1 溢出引起的中断

■ IO 端口

- 最多 6 个 IO 端口(PORTB)
- 可单独设置任意端口为输入或输出(IOSTB)
- 可单独设置任意端口输入上拉或下拉(PHCON/PDCON)
- 可单独设置任意端口推挽输出或开漏输出(ODCON)
- 可单独设置 IO 口有电平变化时产生中断(WUCON)
- 可设置驱动电流强弱

■ 8 位定时器(有两个 8 位的定时器 T0 和 T1)

- T0/T1 都可实现溢出中断
- 可设置起始计数值，并且 T1 定时器可设置自动初始装载值
- T0 定时器的时钟源可设置为高频 CPU 时钟或低频 32768 时钟(可省 2 个匹配电容)
- T1 定时器可带 4 路/5 路 PWM 输出，且 T1 的时钟最快可以和晶振频率相同 16MHz

■ 其他特性

- 内置高精度低电压检测标记(ADO)
- 可设置比较条件的比较器

内置可关闭的稳压管电路，当外接高压时可直接进行稳压
可外接 32768 晶振实现准确计时，并可省外部谐振电容
没有 PCH 寄存器，查表指令可在 1K 地址范围内查表返回
设置有低电压复位功能件

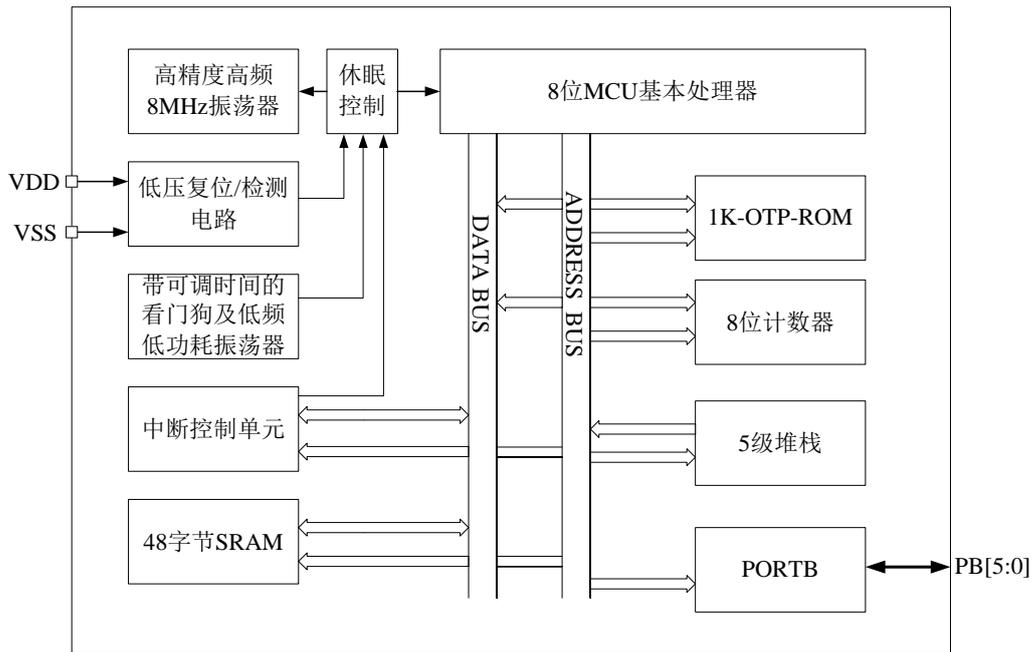
■ 工作电压范围

2M/2T 可以工作在 2.0V - 5.5V

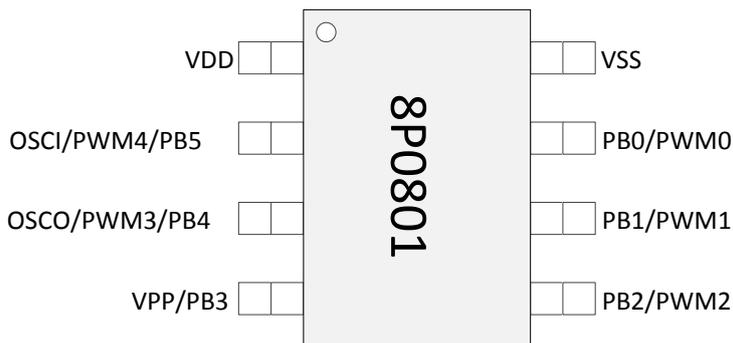
4MHz/2T 可以工作在 2.4V - 5.5V

8MHz/2T 可以工作在 2.6V - 5.5V

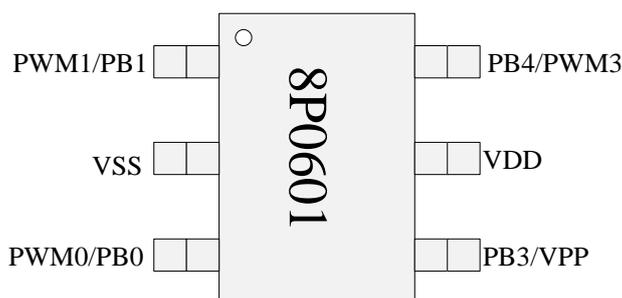
◇ 3. 内部框图



◇ 4.管脚图



SOP8 封装外形图



SOT23-6 封装外形图

◇ 5.管脚功能描述

管脚名称	输入输出功能	说明
PB0/PWM0/INT	I/O	双向 I/O 口同时具有系统唤醒功能 可软件设置为上拉/下拉或开漏输出 外部中断输入脚
PB1/ PWM1	I/O	双向 I/O 口同时具有系统唤醒功能 可软件设置为上拉/下拉或开漏输出
PB2/ PWM2/T0CKI	I/O	双向 I/O 口同时具有系统唤醒功能 可软件设置为上拉/下拉或开漏输出 外部计数输入脚
PB3/VPP	I/O	双向 I/O 口同时具有系统唤醒功能 可软件设置为上拉或开漏输出 (注意此管脚不可外接高于 VDD 电压的电平)
PB4/PWM3/OSCO	I/O	双向 I/O 口同时具有系统唤醒功能 可软件设置为上拉或开漏输出 可外接 32768 晶振做计时用
PB5/PWM4/OSCI	I/O	双向 I/O 口同时具有系统唤醒功能 可软件设置为上拉或开漏输出 可外接 32768 晶振做计时用
VDD	-	电源正端
VSS	-	电源负端

I/O: 表示输入或输出, 电源 VDD 有时候也被表述成 VCC, 电源 VSS 有时候也被表述成 GND;

◇ 6.存储器结构

8P0801 内的存储器包含了程序存储器和数据存储器

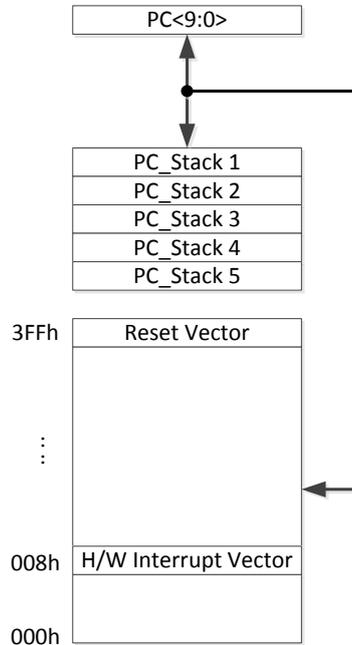
1. 程序存储器

8P0801 有一个 10 位 PC 指针能访问 1Kx13 的存储空间。

8P0801 的复位地址为 3FFh。H/W 中断向量地址 008h。

8P0801 的 CALL/GOTO 能指向在同一个程序页面（一个程序页面为 1K）的所有存储空间。

程序存储器分布图和堆栈结构



2. 数据存储器

数据存储器包含特殊功能寄存器组和通用寄存器组，所有的通用寄存器可以直接寻址或者通过 FSR 寄存器间接寻址。特殊功能寄存器用来控制 CPU 或外围功能模块的工作；

表 1.1:8P0801 寄存器列表

Address	Description	
00h	INDF	
01h	TMRO	
02h	PCL	
03h	STATUS	
04h	FSR	
05h	ADS	
06h	PORTB	
07h	General Purpose Register	
08h	PCON	Timer1_CNT
09h	WUCON	Timer1_reload
0Ah	--	PWM_CR
0Bh	PDCON	PWM0_DATA
0Ch	ODCON	PWM1_DATA
0Dh	PHCON	PWM2_DATA
0Eh	INTEN	PWM3_DATA
0Fh	INTFLAG	Timer1_CR
10h~30h	General Purpose Register	

NA

OPTION

05h

--

06h

IOSTB

表 1.2: 通过 OPTION 或 IOSTB 指令控制的寄存器(仅可写不可读)

地址	说明	B7	B6	B5	B4	B3	B2	B1	B0
02h(w)	OPTION		INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
06h(w)	IOSTB	-	-	PortB I/O 控制寄存器					

表 1.3: 寄存器表(FSR 第 6bit 设置为 0 时)

地址	说明	B7	B6	B5	B4	B3	B2	B1	B0
00h(w)	INDF	通过 FSR 访问数据寄存器(不是一个实际的物理地址)							
01h(w)	TMRO	8 位定时/计数器							
02h(w)	PCL	低 8 位 PC 指针							
03h(w)	STATUS	RST	GP1	ADO	/TO	/PD	Z	DC	C
04h(w)	FSR	-	间接地址访问指针(RAM 选择寄存器)						
05h(w)	ADS					ADS3	ADS2	ADS1	ADS0
06h(w)	PORTB	-	-	PB5	PB4	PB3	PB2	PB1	PB0
07h(w)	SRAM	通用寄存器							
08h(w)	PCON	WDTE	EIS	ADC_EN	ADC4	ADC3	ADC2	ADC1	ADC0
09h(w)	WUCON	-	-	WUB5	WUB4	WUB3	WUB2	WUB1	WUB0
0Ah(w)	PCH	-	-	-	-	-	-	-	-
0Bh(w)	PDCON	*	/PDB2	/PDB1	/PDB0	-	-	-	-
0Ch(w)	ODCON	-	-	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0
0Dh(w)	PHCON	*	*	/PHB5	/PHB4	/PHB3	/PHB2	/PHB1	/PHB0
0Eh(w)	INTEN	GIE	-	-	-	T1IE	INTIE	PBIE	TOIE
0Fh(w)	INTFLAG	-	-	-	-	T1IF	INTIF	PBIF	TOIF

上表中有标注-号的, 当读的时候被设置为 0, 带*号的, 当读的时候被设置为 1;

寄存器表(FSR 第 6bit 设置为 1 时,地址从 00-07 的寄存器不变)

地址	说明	B7	B6	B5	B4	B3	B2	B1	B0
08h(w)	Timer1_CNT	Timer1_CNT 计数器							
09h(w)	Timer1_load	Timer1 自动装载值							
0Ah(w)	PWM_CR	INV	-	PWM4	PWM3	-	PWM2	PWM1	PWM0
0Bh(w)	PWM0_DATA	PWM0 比较数值							
0Ch(w)	PWM1_DATA	PWM1 比较数值							
0Dh(w)	PWM2_DATA	PWM2 比较数值							
0Eh(w)	PWM3_DATA	PWM3 比较数值							
0Fh(w)	Timer_CR	T1_EN	-	-	-	T1_CLK	-	-	-

◇ 7. 特殊寄存器说明

1. INDF(间接寻址寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
00h(w/r)	INDF	通过 FSR 访问数据寄存器(不是一个实际的物理地址)							

INDF 不是一个实际的物理地址, 间接寻址时 INDF 通过 RAM 选择寄存器 FSR 来访问其所指向的地址。间接寻址读操作直接读地址 00h(FSR=0x00), 间接寻址不能对 INDF 直接进行写操作(尽管有些状态会发生改变)。

FSR 的 5-0 位可以用来选择 64 个寄存器 (地址: 00h~3Fh)。

例：

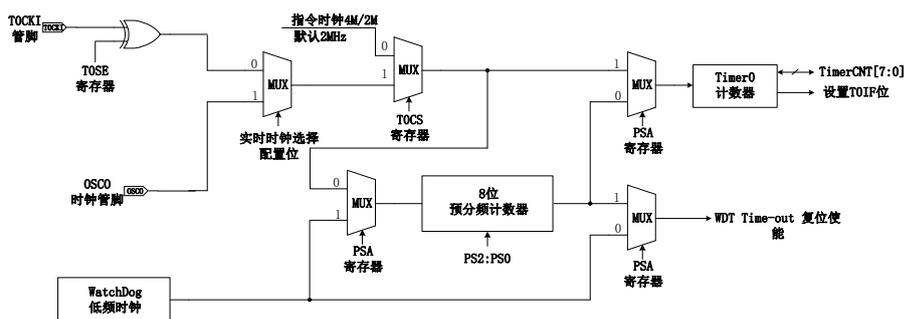
假设地址 38 中的内容是 2Dh（或任一数值）

现在需要通过间接寻址方式来读取地址 38 和 39 中的数据内容；

则先需要赋值 FSR 为 38，然后采用读取 INDF 的语句来间接获得 FSR 指向的寄存器的内容值；也就是说读取 INDF 的值实际上是读取 FSR 指向的数据寄存器的内容的值；同理写也是这样的一个过程；

2. TMRO(定时/计数器 Time clock/Counter register)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
01h(w/r)	TMRO	8 位定时/计数器							



TMRO 是一个 8 位定时/计数器，需要注意的是，TMRO 的时钟源可以设置为多种形式，这个决定着定时/计数器的变化快与慢，TMRO 的最终时钟源自 3 个不同的时钟，通过设置可以采用高频时钟源的 CPU 时钟，也可以采用低频 32768 的时钟，还可以是外部引脚 TOCKI 输入的时钟；且这两个时钟源均可以通过额外的设置让 TMRO 采用以上三个时钟源的分频 PS[2:0]作为 TMRO 的时钟，如以上示意图所示，这里特别需要注意的是当在外接 32768 晶振的时候，上图中，配置位需要在烧录的时候就配置好，寄存器可以在程序中进行设置；

3. PCL(PC 计数器的低 8 位)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
02h(w/r)	PCL	PC 指针的低 8 位							

8P0801 的 ROM 地址是 10 位，所以 PC 指针和堆栈也是 10 位的，但我们只能修改低位的 8 位数据，高位的无法修改；所以要求查表指令所有的表的部分，一定要在一个 8 位地址内，不可以跨越 8 位地址的空间到另外一个地址空间；

4. STATUS(状态寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
03h(w/r)	STATUS	RST	GP1	ADO	/TO	/PD	Z	DC	C

状态字寄存器包含运算标志，CPU 工作状态标志。比较重要的是其中 3 个工作状态位 RST，/TO，/PD 是无法写入的，只能读取；GP1 是一个通用的寄存器位，可以自由使用；ADO:是比较器的输出状态，当比较器+端电压大于比较器-端电压时，比较器的输出为 1，则 ADO 位为 1；当比较器+端电压小于比较器-端电压输出为 0，则该位被设置为 0；

5. FSR(间接寻址指针)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
04h(w/r)	FSR	0	间接寻址指针						

因 8P0801 仅有 48 个字节的通用寄存器，所以仅需要 B5:B0 共 6 位来进行寻址即可；

Bit7 始终被置为 0，所以读出的时候，这位始终为 0；

Bit6 被设置为特殊寄存器的 bank 访问标志位，需要小心使用；正常使用时，需要设置

FSR[6]=0; 当需要访问 PWM 寄存器的时候, 才需要设置 FSR[6]=1;

6. ADC 比较条件设置寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
05h(w/r)	ADS	-	-	-	-	ADS3	ADS 2	ADS 1	ADS 0

ADS 寄存器可以控制内置的比较器的各种输入条件, 具体用法可参考后面图例;

ADS 3: 选择比较器的+端输入为 bandgap=1.2V 的参考电压, 或外部端口输入电压 (默认为 0 选择内部 bandgap 电压), 当该电压大于比较器-端口的电压时, 则比较器输出 1, 数据可以通过 LVD 口来进行读取;

ADS 2: 选择比较器的-端输入为电阻分压还是外部端口电压 (默认为 0 选择内部电阻分压)

ADS 1: 选择内部电阻分压的上电源电阻大小

ADS 0: 选择内部电阻分压的下电阻大小

7. PORTB(PortB 寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
06h(w/r)	PortB			IOB5	IOB4	IOB3	IOB2	IOB1	IOB0

PortB 寄存器是作为出口的数据存储在这个寄存器中, 但当作为输入口的时候, 如果选择的是从端口读取数据, 则芯片会从这个输出数据寄存器中读取数据;

8. PCON(电源控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
08h(w/r)	PCON	WDTE	EIS	ADC_EN	ADC4	ADC3	ADC2	ADC1	ADC0

上述地址的寄存器中的 bit3:bit0 没有使用;

WDTE: 看门狗使能位

=0, 关闭看门狗

=1, 开启看门狗

EIS: 定义 PORTB0 位的功能是 PORT 功能还是 INT 外部中断功能

=0, 设置 PORTB0 为正常的 IO 口功能

=1, 设置 PORTB0 为外部中断口输入功能; (同时需要把 PORTB0 设置为输入)

ADC_EN: 比较器检测功能开启控制位

=0, 比较器检测关闭(默认关闭)

=1, 比较器检测开启 (当开启比较器后, ADO 将在 3-5ms 内开始输出比较器的值, 可以通过读取 ADO 寄存器的值, 了解当前比较器的输出状态);

ADC[4:0]高精度低电压检测设定值

共计 32 种参考检测电压值选择, 可以通过 ADC 寄存器来进行设置;

9. WUCON(PortB 输入改变/唤醒控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
09h(w/r)	WUCON	-	-	WUB5	WUB4	WUB3	WUB2	WUB1	WUB0

WUB 寄存器可以控制各个输入管脚是不是开启相应的电平变化改变/唤醒事件相应处理;

=0, 关闭相应的事件处理(默认关闭)

=1, 打开相应的事件处理功能

10. PCH 寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Ah(w/r)	PCH	固定置为 0							

这里没有任何作用的一个寄存器，地址 0A 的寄存器固定输出为 0；

11. PDCON(PortB 下拉控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Bh(w/r)	PDCON	*	/PDB2	/PDB1	/PDB0	*	*	*	*

8P0801 仅有 3 个管脚具备内部下拉电阻，分别是 PortB0, 1, 2 这三个管脚；

=0, 使能内部下拉电阻

=1, 禁止内部下拉电阻(默认)

12. ODCON(PortB 开漏输出控制器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Ch(w/r)	ODCON	-	-	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0

开漏输出的好处在于可以多个端口同时输出到同一根总线上，类似 I2C 的 SDA 总线；特别的要注意，PortB3 虽然也能设置为开漏输出，但是由于这个管脚在烧录的时候需要外接高压烧录，所以即使被设置为开漏输出，也不允许在这个管脚外接高于芯片 VDD 的电压；

=0, 关闭开漏功能(默认)

=1, 使能开漏输出功能

13. PHCON(PortB 上拉控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Dh(w/r)	PHCON	*	*	/PHB5	/PHB4	/PHB3	/PHB2	/PHB1	/PHB0

=0, 使能内部上拉电阻

=1, 禁止内部上拉电阻(默认)

14. INTEN(中断使能控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Eh(w/r)	INTEN	GIE				T1IE	INTIE	PBIE	TOIE

这里比较特殊的是，当设置了 T1IE，这里是设置定时器 TMR1 计数器开启中断相应；

15. INTF(中断标志位寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Fh(w/r)	INTF					T1IF	INTIF	PBIF	TOIF

注意以上标志位均需要软件清零；

16. W 工作暂存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
N/A(w/r)	Work	累加器(也有命名是 ACC)							

累加器是一个内部中转用的暂存器，当执行指令的时候这里会暂存部分数据，无法被用户指定访问；

17. OPTION Register(选项寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
N/A(w)	OPTION		INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0

这里主要有一个特殊的设置点，当设置 TOCS 为默认的值 1 的时候，和其他家方案不同的是，8P0801 不会自动把 PortB2 设置为输入，而是需要用户确认是不是真的需要使用 TOCS 功能，如果确认是需要使用 TOCS 的功能，那么需要软件设置 IOSTB2 为输入；否则 PORTB2 的功能还是由 IOSTB2 来决定是输入还是输出(IOSTB 的优先级最高)；

PS2/PS1/PS0: 分频率选择控制位

PS2/PS1/PS0	Timer0 分频比例	WDT 分频比例
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

PSA: 分频器选择位

- =1, WDT 使用预分频器分频
- =0, TMR0 使用预分频器分频

TOSE: 外部触发方式选择位

- =1, TOCKI 脚下降沿触发计数
- =0, TOCKI 脚上升沿触发计数

TOCS: TMR0 时钟源选择控制位

- =1, 外部 TOCKI 脚或者 32768 的 OSCO 输出作为 TMR0 的时钟；
- =0, 内部 CPU 指令时钟(例如选择 4MHz/2T, 则这里的时钟周期为 2MHz)

INTEDG: 中断触发方式控制位

- =1, 中断触发方式为 INT 脚上升沿触发；
- =0, 中断触发方式为 INT 脚下降沿触发；

16. IOSTB(PortB 输入/输出控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
N/A(w)	IOSTB			IOSTB5	IOSTB4	IOSTB3	IOSTB2	IOSTB1	IOSTB0

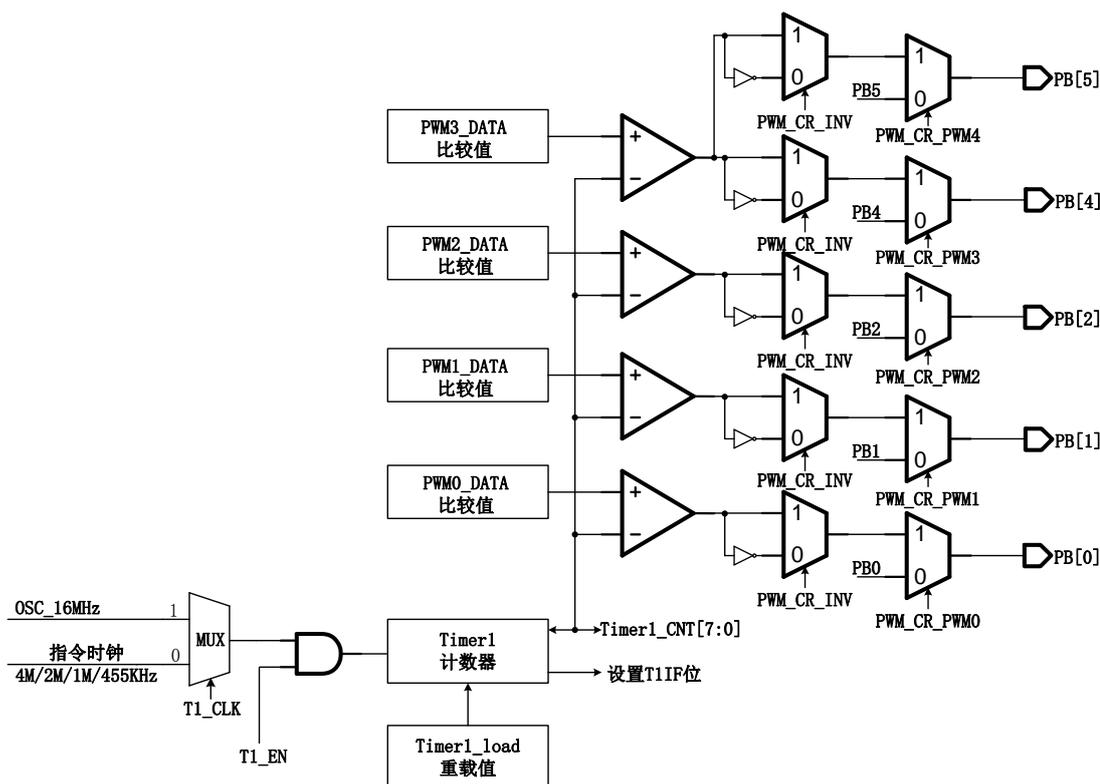
通过单独的 IOSTB 指令，可以设置 PortB 的输入或输出功能，只可通过特殊指令来写，不可读该寄存器；

- =1, 输入功能(默认)
- =0, 输出功能

17. TMR1(定时/计数器 T1)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
08h(w/r)	TMR1	8 位定时/计数器							

注意，当设置 FSR[6]=1 的时候，对地址 08 进行读写的操作，都是对 TMR1 进行操作，可以对定时器 T1 的值进行读写；



18. T1_LOAD(定时器 T1 的自动重载值)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
09(w/r)	T1_LOAD	T1_LOAD							

注意，设置该寄存器的时候，需要预先设置 FSR[6]=1；才可以对此寄存器进行访问读写操作；当 T1 计数溢出时，T1 计数器会自动装载该值为初始计数值；

19. PWM_CR(PWM 控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
09(w/r)	PWM_CR	INV	-	PWM4	PWM3	-	PWM2	PWM1	PWM0

注意，设置该寄存器的时候，需要预先设置 FSR[6]=1；才可以对此寄存器进行访问读写操作；

PWM0=0 PB0 口保持正常的输入输出功能；

PWM0=1 PB0 口切换为 PWM0_DATA 控制的输出；（注意需预先设置 PB0 口为输出）

PWM1=0 PB1 口保持正常的输入输出功能；

PWM1=1 PB1 口切换为 PWM1_DATA 控制的输出；（注意需预先设置 PB1 口为输出）

PWM2=0 PB2 口保持正常的输入输出功能；

PWM2=1 PB2 口切换为 PWM2_DATA 控制的输出；（注意需预先设置 PB2 口为输出）

PWM3=0 PB4 口保持正常的输入输出功能；

PWM3=1 PB4 口切换为 PWM3_DATA 控制的输出；（注意需预先设置 PB4 口为输出）

PWM4=0 PB5 口保持正常的输入输出功能；

PWM5=1 PB5 口切换为 PWM3_DATA 控制的输出；（注意需预先设置 PB5 口为输出）

INV=0 所有的 PWM 口的输出波形为正波形输出；

INV=1 所有的 PWM 口的输出波形为反向波形输出；

20. PWM_DATA(PWM 的占空比控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0B(w/r)	PWM0_DATA	PWM0_DATA							
0C(w/r)	PWM1_DATA	PWM1_DATA							
0D(w/r)	PWM2_DATA	PWM2_DATA							
0E(w/r)	PWM3_DATA	PWM3_DATA							

注意，设置该寄存器的时候，需要预先设置 FSR[6]=1；才可以对此寄存器进行访问读写操作；

在 PWM_CR.INV=0 控制 PWM 正向输出的情况下，当 PWM_DATA 越大，则占空比越小(高电平比较少)，反之 PWM_DATA 越小，则占空比越大；

T1_CR(定时器 T1 的控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0F(w/r)	T1_CR	T1_EN	-	-	-	T1_CLK	-	-	-

注意，设置该寄存器的时候，需要预先设置 FSR[6]=1；才可以对此寄存器进行访问读写操作；

T1_EN: =1, 使能 T1 定时开始计数

=0, 关闭 T1 计数器计数

T1_CLK:=1, 选择 T1 计数器的时钟选用高频系统时钟为 16MHz

=0, 选择 T1 计数器的时钟选用 CPU 指令时钟，根据 CPU 的速度来决定时钟；

PWM 设置使用例程说明：

预先定义：

```

TMR1      EQU      08H
T1_LOAD   EQU      09H
PWM_CR    EQU      0AH
PWM0_DATA EQU      0BH
PWM1_DATA EQU      0CH
PWM2_DATA EQU      0DH
PWM3_DATA EQU      0EH
T1_CR     EQU      0FH
#define   T1_EN    T1_CR,7

```

1. 需要设置 PWM 寄存器之前，先设置 FSR[6]为 1，当 FSR[6]为 1 时，下面的地址开始切换为对 PWM 寄存器进行设置读写；

通常语句是：MOVIA 40H

MOVAR FSR

2. 通常先设置好 PWM 寄存器的各个寄存器的值，然后再打开时钟开关 T1_EN；

比如我们需要产生一个精准的 108KHz 的 PWM 波形在 PB2 口上，这个通常应用于超声波雾化片驱动，时钟精度要求较准，并且希望可调范围较宽，这样我们根据上图说明使用主频为 16MHz 来做 PWM 计数器的主时钟，T1_CLK 应该设置为 1；由于是要产生 108KHz，则我们用 $16000\text{KHz}/108\text{KHz}=148$ ，所以我们要设置计数器的分频数为 148；通过反向计算，重载寄存器值应该设置为 $\text{FFH}-94\text{H}+2=6\text{DH}$ (其中 $148=94\text{H}$)；

MOVIA 6DH

MOVAR TMR1

MOVAR T1_LOAD

```
BSR    T1_CR,3
```

然后我们设置输出大约 50%的占空比，经过计算 $148/2=74=4AH$ 然后 $FFH-4AH=B5H$ 就是这个约为 50%的数值；我们需要设置 PB2 对应的 PWM 数据，也就是 PWM2_DATA 所以我们按以下赋值：

```
MOVIA  B5H
```

```
MOVAR  PWM2_DATA
```

接下来控制 PWM 开始运行起来，也就是 T1_EN 要设置为 1：

```
BSR    T1_EN
```

然后控制使得 PB2 口切换为 PWM 的输出波形；

```
MOVIA  04H
```

```
MOVAR  PWM_CR
```

至此，在 PB2 口输出一个 108KHz 的 PWM 波形，占空比是 50%，就已经生成并输出到 PB2 口上；如果要改变占空比，则更改 PWM2_DATA 的值就好；值越大，则 PWM 出的高电平则越少；以此类推；

在 PWM 的应用中，有两种极端的方式，需要按以下方式注意使用：

1. 当 PWM_DATA 的值设置等于或小于 T1_LOAD 的值时，这时，对应的输出波形正向(INV=0)的应该是全部高电平，但是实际输出电平会有一个大约不到 0.5uS 的低电平脉冲，具体的时间长度由 CPU 的运行速度决定；但是无论多小，有时候想要等到全关的波形，也是没有办法达到的，所以这点需要注意，我们可以设置 PB 输出为 1 来关闭该值，或者使用反向 INV=1 的情况来彻底关闭该值的输出；
2. 当 PWM_DATA 的值设置等于 FF 值时，这时，对应的输出波形正向(INV=0)的是全部低电平，请以此为基准；利用这点，我们可以使用反向 INV 输出，使得 PWM 值可以得到全高电平输出的情况，可以得到彻底关闭输出波形的效果；

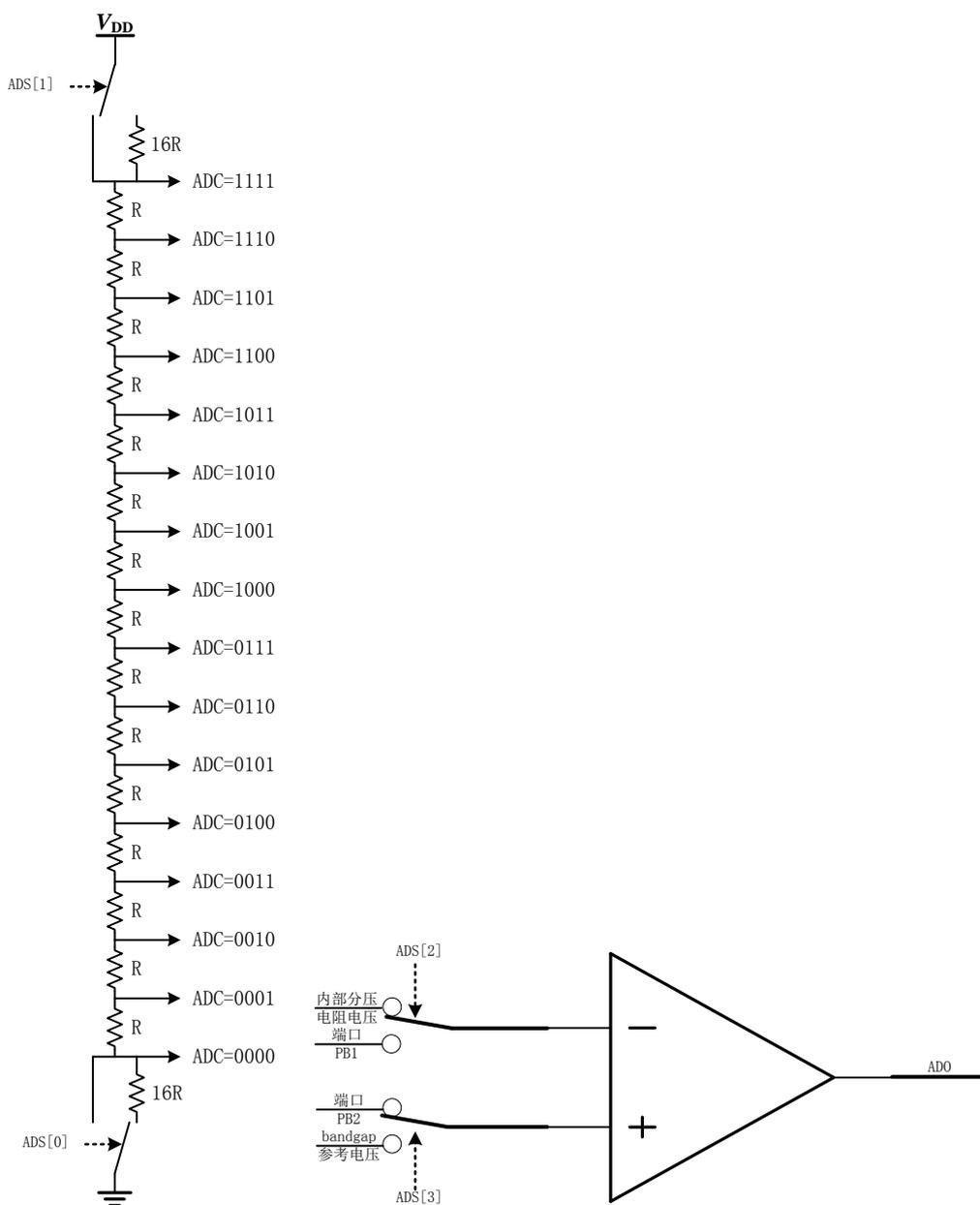
◇ 8.特殊部件说明

内置的比较器单元,由寄存器 ADC_EN,ADS[3:0],ADC[4:0]来组合控制比较输出；

其中 ADC_EN 是比较器的工作开始使能，当设置该位为 1 的时候，比较器开始工作，建议在开始工作之前，先设置好比较器的正端和负端输入条件，然后再打开比较器开始工作使能；

ADS 是 4bit，其中低位 2bit 来分别控制比较器的正端和负端的输入源；高 2 位是控制内部电阻分压的比例关系；

ADC 共计 5bit，是选择内部分压电阻的抽头，可以选择电源电压的比例形成一个比较电压；



比较器使用说明:

当 ADS[3] 设置为 1 时，比较器的正端选择输入电压为 bandgap 参考电压，一般为 1.2V 左右，上下不超过 0.1V，且该电压不会随电源电压以及温度变化而变化；是一个比较好的参考电压，当 ADS[3] 寄存器被设置为 0 的时候，则比较器的正端连接到 PB2 端口；以使用 PB2 端口的电压为比较电压；注意，由于是内置的 NMOS 管的比较器，NMOS 管的比较电压一定要大于 1V 为好，必须设置 PB2 的端口大于 1V 才可以进行比较；PB2 的最大电压值可以达到 VDD 电源，最大值可以无限制要求；比较器比较精度最高的电压点是在 1.2V 左右；

同理，当设置 ADS[2] 寄存器为 1 的时候，比较器的负端连接到端口 PB1，当设置为 0，比较器的负端连接到内部分压电阻的输出端口上，具体的输入电压还需要通过其他的寄存器选择端口；

ADS[1] 当设置为 0 时，则短路上部电阻 16R，当设置为 1 时，则上部电阻 16R 开启使用串联在分压电路中；

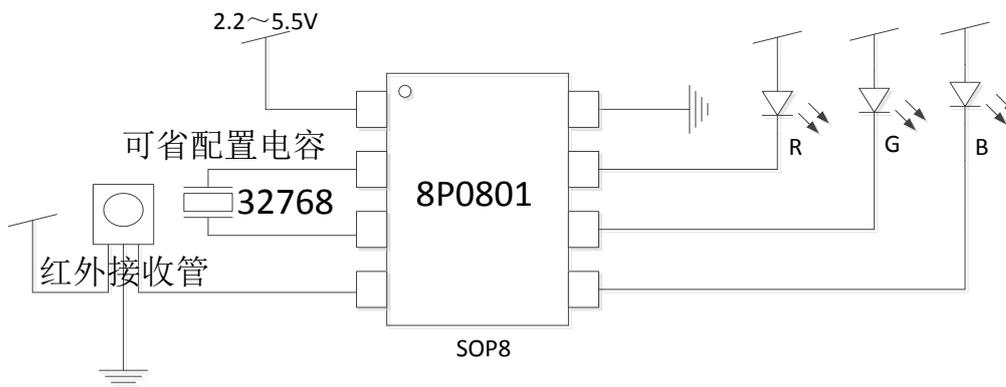
ADS[0] 当设置为 1 时，则短路下部电阻 16R，当设置为 0 时，则下部电阻 16R 开启使用

串联在分压电路中；

ADC[4:0]则分别选择哪个端口输出电压到最终的接口上，数字越大，则参考电压越高，为了精确比较，建议计算当电压为 1.2V 时为比较的翻转点；上图仅标注出 4 位分压电阻选择，实际有 5 位数字可供选择，比上图更细；

以下是电风扇检测电源电压的一个设置过程，电风扇需要设置在 2.4V 进行低电压复位；根据以上情况，我们先设置 ADS=08H，然后设置 PCON=2FH，然后设定在主程序中去判断 ADO 的值，当读取到 ADO 的值为 1 的时候，标识电源电压是高于 2.4V 的，而当读取到 ADO 的值为 0 的时候，则表示当前电压值小于 2.4V 了，系统因为跳转的低电压保护程序段；

◇ 9. 典型应用电路



以上为实际的SOP8产品应用电路图，注意，由于红外接收管在电压较高的情况下，需要额外的滤波电容104一只；8P0801本身不需要额外电容滤波，但由于红外接收管可能无法工作在高电压系统下，红外接收管可能需要额外的滤波电容104一只

◇ 9.配置选项说明

名称	说明
IOB4 功能选择[10]	=0 IOB4 功能(默认) =1 IRC_CPU_CLOCK 输出到 IOB4
实时时钟选择[9]	=0 IOB4/5 选择为普通输出(默认) =1 选择 IOB4/5 外接 32768 晶振
RDPORT[8]	=0 选择从寄存器读取数据(默认) =1 选择从端口读取数据
CPU_SPEED[7:6]	=0 4M/2T (默认) =1 8M/2T
上电延迟选择[5:4]	=00 PWRT=WDT=18mS (默认) =01 PWRT=WDT=4.5mS =10 PWRT=WDT=288mS =11 PWRT=WDT=72mS
SCHMITT[3]	=0 打开施密特读取电路(默认) =1 关闭施密特读取电路
32768 加快起振选项[2]	=0 不加快启动(默认) =1 加快启动 (加快启动可能导致振荡功耗加大)
LVR/LVD 电压选择[1:0]	=00 低电压检测 1.8V(默认)

	=01 低电压检测 1.9V =10 低电压检测 2.1V =11 低电压检测 2.3V
Protect[12]	=0 选择芯片烧录的时候不加密输出(默认) =1 选择芯片烧录的时候加密输出
内置稳压管[8]	=0 默认启动后关闭稳压管(默认) =1 始终打开稳压管
弱驱动/标准驱动/加强驱动[10:9]	=00 弱驱动(默认) =01 标准驱动 =10 标准加强驱动 =11 最强驱动

◇ 10.指令集合

操作语法	说明	操作内容	指令周期	影响标志位
BCR R,bit	清除 R 的 bit 位为 0	0→R	1	-
BSR R,bit	设置 R 的 bit 位为 1	1→R	1	-
BTRSC R,bit	检测 R 的 bit 位, 如果是 0 跳过下一条	如果 R=0 跳过下一条	1/2	-
BTRSS R,bit	检测 R 的 bit 位, 如果是 1 跳过下一条	如果 R=1 跳过下一条	1/2	-
NOP	空操作	无操作	1	-
CLRWDT	清 watchdog 寄存器	00→WDT 相关寄存器	1	/TO,/PD
OPTION	加载 ACC 数据到 OPTION 寄存器	ACC→OPTION	1	-
SLEEP	进入休眠模式	00→WDT 相关寄存器	1	/TO,/PD
RETURN	从子程序返回上一级主程序	Stack→PC	2	-
RETIFIE	从中断返回并重新设置 GIE 为 1	Stack→PC GIE=1	2	-
IOSTB	加载 ACC 数据到 IOSTB	ACC→IOSTB	1	-
CLRA	清 ACC 寄存器数据	00→ACC	1	Z
CLRR R	清 R 寄存器数据	00→R	1	Z
MOVAR R	传送 ACC 数据到 R	ACC→R	1	-
MOVR R,d	传送数据	R→dest	1	Z
DECR R,d	自减 1	R-1→dest	1	Z
DECRSZ R,d	自减 1, 如果为 0 跳过下一条	R-1→dest	1/2	-
INCR R,d	自加 1	R+1→dest	1	Z
INCRSZ R,d	自加 1, 如果为 0 跳过下一条	R+1→dest	1/2	-
ADDAR R,d	加法指令	R+ACC→dest	1	C,DC,Z
SUBAR R,d	减法指令	R-ACC→dest	1	C,DC,Z
ANDAR R,d	执行 AND 操作	ACC & R →dest	1	Z
IORAR R,d	执行 OR 操作	ACC R →dest	1	Z
XORAR R,d	执行 XOR 操作	ACC ^ R →dest	1	Z
COMR R,d	取反	!R → dest	1	Z

RLR R,d	带进位的左移 1 位操作	$R\langle 7 \rangle \rightarrow C$, $R\langle 6:0 \rangle \rightarrow \text{dest}\langle 7:1 \rangle$ $C \rightarrow \text{dest}\langle 0 \rangle$	1	C
RRR R,d	带进位的右移 1 位操作	$C \leftarrow \text{dest}\langle 7 \rangle$, $R\langle 7:1 \rangle \leftarrow \text{dest}\langle 6:0 \rangle$ $R\langle 0 \rangle \leftarrow C$	1	C
SWAPR R,d	高低位 4bit 交换操作	$R\langle 3:0 \rangle \rightarrow \text{dest}\langle 7:4 \rangle$ $R\langle 7:4 \rangle \rightarrow \text{dest}\langle 3:0 \rangle$	1	-
MOVIA I	传送立即数到 ACC	$I \rightarrow \text{ACC}$	1	-
ADDIA I	执行立即数 ADD 加法操作	$I + \text{ACC} \rightarrow \text{ACC}$	1	C,DC,Z
SUBIA I	执行立即数 SUB 减法操作	$I - \text{ACC} \rightarrow \text{ACC}$	1	C,DC,Z
ANDIA I	执行立即数 AND 操作	$I \& \text{ACC} \rightarrow \text{ACC}$	1	Z
IORIA I	执行立即数 OR 操作	$I \text{ACC} \rightarrow \text{ACC}$	1	Z
XORIA I	执行立即数 XOR 操作	$I \wedge \text{ACC} \rightarrow \text{ACC}$	1	Z
RETIA I	返回, 同时把 I 值放入 ACC	$I \rightarrow \text{ACC}, \text{Stack} \rightarrow \text{PC}$	2	-
CALL I	调用子程序	$\text{PC} + 1 \rightarrow \text{Stack}$ $I \rightarrow \text{PC}$	2	-
GOTO I	无条件跳转	$I \rightarrow \text{PC}$	2	-

红色字体指令不可使用;

◇ 11. 电气特性

最大工作范围

项目	范围	说明
DC 供电范围	2.0V~5.5V	芯片使用标准 5V 工艺设计生产
操作温度范围	-10~60℃	
存储温度范围	-40~125℃	

11.1 电气特性

项目	参数	条件	最小	典型	最大	单位
DC 供电范围	常温下	4M/2T 条件下工作	2.0	5	5.5	V
静态电流	5V	STOP 模式下, WDT 关闭			1	uA
	5V	STOP 模式下, WDT 打开			5	uA
	5V	STOP 模式下, 32768 打开			20	uA
工作电流	5V	8M/2T 下, 没有外接负载			1	mA