

# BJ8P164A

## FAQ（汇集）

如果您在使用上还有其它问题请与我们联系：[tbs002@bjxmcu.com](mailto:tbs002@bjxmcu.com)

# 目 录

Q1: 如何正确使用 WATCHDOG? .....	4
Q2: MCU 的 I/O 口作 OPEN DRAIN 输出时, 外接上拉电阻的供电电压比 IC 电源电压高, 是否允许? .....	4
Q3: 低功耗 I/O 口应如何设置? .....	4
Q4: MCU 的 I/O 上拉电阻是多少? .....	4
Q5: 独立按键与 MCU 如何连接能有效避免 I/O 口损伤? .....	5
Q6: I/O 外接下拉电阻, 没有外接信号时芯片为何读不到低电平? .....	5
Q7: 使用 RESET 脚作为输入口时 (通过按键接地), 为何读不到低电平? .....	5
Q8: 单片机 I/O 口驱动电流是多大? .....	5
Q9: IO 口从输入状态切换为输出状态注意事项? .....	5
Q10: 为什么程序中清 FP00IEN 后, 本来应该禁止外部中断 INT0, 但在程序运行时发现也可进入? .....	6
Q11: 程式不进入中断, 需要检查哪些方面? .....	6
Q12: P00 设置完全正确, 为何无法进入中断? .....	6
Q13: 单片机有没有中断优先顺序? 如何设置? .....	6
Q14: 可以在 INTERRUPT SERVICE ROUTINE 内执行 CALL 吗? .....	7
Q15: 在一个计数器在用作 PWM 输出的时候, 该计数器能否同时用作中断源? .....	8
Q16: 片内集成的 PWM 模组可以输出多种频率吗? 是否可控可调? .....	8

---

<b>Q17: PWM 功能中如何去更新占空比和周期? .....</b>	<b>8</b>
<b>Q18: 定时/计数器的时钟源来自哪里? 如何设置计时器的分频数? .....</b>	<b>9</b>
<b>Q19: 低速模式的计时器定时和普通模式有何不同? 定时时间怎么计算? .....</b>	<b>9</b>
<b>Q20: 使用内置的即时时钟 (RTC) 功能时, 需要注意哪些事项? .....</b>	<b>9</b>
<b>Q21: RTC 功能仿真与 IC 应用时, 其设定方式及注意事项为何? .....</b>	<b>10</b>
<b>Q22: 在 AD 应用时, 软件有那些注意事项? .....</b>	<b>10</b>
<b>Q23: 使用 BJ8P164A 单片机的 ADC 功能时, 在软件和硬件设计上应注意哪些事项? ....</b>	<b>11</b>
<b>Q24: 在芯片进入睡眠模式之前, 应如何设置 ADC 口? .....</b>	<b>11</b>
<b>Q25: 带 AD 的单片机 AD 引脚能否输入负电压? 如 VDD=5V 电源时, 当 AD 输入(+5V)(-5V) 时单片机的数据是多少? .....</b>	<b>11</b>
<b>Q26: 多通道 AD 转换时, 是否会产生 AD 互相干扰的情况? .....</b>	<b>11</b>
<b>Q27: 仿真 BJ8P164A 时, 仿真器上的 P0.4 脚为何没有 BUZZER 输出? .....</b>	<b>12</b>
<b>Q28: 在电路设计及 PCB LAYOUT 中, 可采用那些措施以加强芯片抗 ESD 能力? .....</b>	<b>12</b>
<b>Q29: 芯片为何无法进入 SLEEP MODE? .....</b>	<b>13</b>
<b>Q30: 如何正确设置进入低速模式? .....</b>	<b>13</b>
<b>Q31: 在产品开发过程中如何提高 EFT 特性? .....</b>	<b>13</b>
<b>Q32: BJ8P164A 有几级堆栈.....</b>	<b>14</b>

## Q1: 如何正确使用 Watchdog?

Watchdog (看门狗) 计时器的振荡源, 来源于晶片内部独立的低速 RC 振荡。看门狗计时器的溢出时间为: 256ms@5V、512ms@3V, 由于内部独立 RC 振荡频率受工作电压和环境温度影响, 因此看门狗溢出时间为一般经验值。看门狗除了在特殊场合用作计时器外, 主要用来防止程式的跑飞。但如果使用方法不当, 看门狗的作用便不能充分发挥。

用户可以通过 Code Option 选项控制看门狗计时器。在编译选项中, Watch\_Dog 有 3 个配置选项可供选择: Always\_On, Enable 和 Disable。各选项的选择可以参考 datasheet 看门狗计时器相关章节。程式中不可以打开或关闭 Watchdog, 清除 WatchDog 采用 @RST\_WDT, 即给 WDTR 赋值 5A。复位看门狗计时器的位置:

一般来说复位看门狗计时器的最好位置是将重定设置在程式的主干道上。这样如果程式跑飞造成某个程式模组的闭环回不到程式的主干道上, 看门狗计时器就会及时溢出促使晶片复位。特别要注意的是不要安置在副程式或中断处理程式中。而且复位看门狗计时器的动作最好只作一次, 从而避免在某个闭环中还是一直复位, 看门狗计时器也就失去了作用。

另外复位看门狗计时器也不能放置在 Timer 中断里, 因为 MCU 跑到死回圈时, 一样会进 TIMER 中断也会清看门狗计时器。此外, 用户可以在程式开始通过侦测 Pflag 系统寄存器中的 NT0 = 0、NPD = 0 来判定是否由看门狗计时器溢出复位, 从而作相应的处理。

## Q2: MCU 的 I/O 口作 OPEN DRAIN 输出时, 外接上拉电阻的供电电压比 IC 电源电压高, 是否允许?

open drain 是属于 pseudo open drain, 外接上拉电阻的电压 要小于等于 MCU 的 VDD, 主要目的是应用在多个 output 并接 bus 上 (例如 I2C), 或者是电压比 VDD 低的接口电压电平的转换。用户可以通过外接三极管来解决这个问题。

## Q3: 低功耗 I/O 口应如何设置?

正确设定 I/O 口状态, 可以避免 I/O 口的漏电流:

1. 对空闲口的设置, 空闲口一般设定为输入上拉或输出低电平。
2. 对 ADC I/O 通道, 应用 P4CON 的设定, 可以避免 I/O 口的漏电流。
3. 对于不同的外部硬件电路, 考虑 I/O 的状态设置, 设定不当, 会有漏电流,

特别注意上拉电阻的正确设定, 若 I/O 口外接低点平, 又将 pull-up enable 会造成漏电, 例如: VDD=5V 会有约  $5V / 100K = 50\mu A$  漏电

## Q4: MCU 的 I/O 上拉电阻是多少?

此点在芯片的规格书中 (电气特性章节) 有所描述。

Vdd=3V 时, 上拉电阻典型值为 200K 欧姆。

Vdd=5V 时, 上拉电阻典型值为 100K 欧姆。

## Q5: 独立按键与 MCU 如何连接能有效避免 I/O 口损伤?

在产品开发时, 按键是最常见的功能, 由于考虑到成本等因素, 很多客户都是将按键的两端分别与 MCU 和 GND 相连, 这种做法在大部分情况下是没有问题的。但是, 当该产品的工作环境比较差时, 比如静电、电源噪声等, 此时若按键被按下, 则相当于将 MCU 的一个 I/O 口直接与 GND 相连, 此时就很可能会有瞬间大电流或高电压

甚至负压灌入 I/O 口, 造成 I/O 损坏。减少这种问题的防范措施是: 根据实际电路的需要, 在按键与 MCU 的 I/O 之间串接 100ohm~1Kohm 电阻, 可有效避免上述问题带来的损伤。

## Q6: I/O 外接下拉电阻, 没有外接信号时芯片为何读不到低电平?

MCU 的普通 I/O 口内建上拉电阻, 其等效阻值大概是 100 Kohm@5V、200 Kohm@3V。如果客户在程序中将某 I/O 的上拉电阻使能, 同时又在该 I/O 的外围接有下拉电阻, 在没有信号输入的情况下, 相当于内置上拉电阻与外接下拉电阻串联分压, 所以会导致 I/O 口读不到低电平。例如, 在 VDD=5V 的情况下, 内置上拉电阻等效为 100K, 如果此时外部接有 100K 的下拉电阻, 则在外部输入信号时, 该 I/O 的输入电平约为 0.5VDD, 该电平已经超出了 MCU 的 VIL/VIH 范围, 肯定会导致读取状态错误。

所以, 客户在程序设计时, 如果 I/O 需要下拉状态, 可在该 I/O 外接下拉电阻, 但切记要关闭该 I/O 口的内部上拉功能。

## Q7: 使用 Reset 脚作为输入口时 (通过按键接地), 为何读不到低电平?

查看 Code Option 选项 Reset 选项是否选择 Px.x, 若选择 Reset, 则表示该引脚作外部复位引脚使用; OTP MCU 该引脚只能作为输入口使用, 且无内置上拉电阻, 用户需确认电路已外接上拉电阻; 只有当其输入电压大于 0.9VDD 时才可读取到高电平 (ViH)。另外 Reset 引脚输入电压低于 0.2VDD 时才检测到低电平 (ViL), 用户在使用时也需注意;

当 reset 脚作为复位脚时, 也是在 0.9VDD 才认到高, 在 0.2VDD 认到低, 即当电压低于 0.2Vdd 时才能有效的进入重定模式, 当电压高于 0.9Vdd 后, 才能保证进入正常工作。

## Q8: 单片机 I/O 口驱动电流是多大?

I/O 口驱动电流可分为拉电流和灌电流, 拉电流是指当 I/O 为输出口并且输出为高电平时, 对外部电路驱动能力的强弱。电流越大, 驱动能力越强。对于一般的 IO 口, 其输出最大拉电流为 12mA, 最大灌电流为 15mA。

## Q9: IO 口从输入状态切换为输出状态注意事项?

输入口切换到输出口步骤:

- (1) 设定 I/O 预定状态, 即送预定值给 Pn;
- (2) 切换为输出状态, 即送“1”给 PnM。

从 I/O 的基本结构可以看出: 上拉电阻在切换成输出模式时, 是无效的。所以可以不操作 PnUR。修改同一埠的不同 I/O 引脚状态时, 最好使用对埠操作的 MOV PX, A 指令操作, 如:

```
MOV A,#XXH
B0MOV PX,A
MOV A,#XXH
B0MOV PXM,A
```

### Q10: 为什么程序中清 FP00IEN 后, 本来应该禁止外部中断 INT0, 但在程序运行时发现也可进入?

中断机制是一旦在 P0.0 口接受到中断条件, 总会设置相应的中断标志位 FP00IRQ, 但能不能进入中断还要看 FP00IEN 是否允许 (这样做是为了客户采用查询方式的需要)。但是一旦中断标志位 FP00IRQ 被外部中断条件设定后, 又有其他中断产生, 此时若中断程序处理不当, 则可能造成错误进入 INT0 中断。解决办法:

1. 若中断程序中不需要 INT0 中断, 则中断程序中不要查 FP00IRQ 标志。

2. 若中断程序中需要 INT0 中断, 应该配合 FP00IEN 一起使用, 如下:

```
B0BTS1 FP00IEN ; P00 中断是否打开
JMP EXIT ; 没有 P00 中断, 退出
B0BTS1 P00IRQ ; 进一步查询中断请求标志
JMP NEXT
NEXT: ; P00 中断服务程序
EXIT: ; 退出中断服务程序
RETI
```

### Q11: 程式不进入中断, 需要检查哪些方面?

用户需要从以下几个方面检查:

1. 检查全局中断 FGIE 是否使能;
  2. 相应中断的中断使能是否开启;
  3. 确定程式是否可以产生中断, 如计时器初始化中的设置是否可以溢出;
- 要产生外部中断, P0、P1 相应的 I/O 口必须设置为输入模式, 且有电平变化等。

### Q12: P00 设置完全正确, 为何无法进入中断?

首先查看 P00 中断的设置是否正确, 比如 P00IEN 是否置 1, PEDGE 寄存器设置是否正确 (控制触发边缘类型)。如果以上设定都没有问题, 接着检查程序中是否在使用 TC0 或 TC1 计数或者产生中断, 如果是, 请确认 TC0 或 TC1 的计数时钟源选择为 Fcpu, 而不能将 TC0 或 TC1 的计数时钟源选择为外部 P00 口, 否则当 P00 口有触发电平时, TC0 或 TC1 的计数值会加 1, 但 P00IRQ 不会置 1, 从而导致 P00 无法进入中断。

### Q13: 单片机有没有中断优先顺序? 如何设置?

硬件没有设计中断的优先顺序。但实际应用的过程中, 可能需要应用中断优先顺序, 可以根据 MCU 的

特点用软体来实现。对各中断进行优先权的设置,并根据 IEN 和 IRQ 的状态决定系统是否回应中断请求。用户必须在中断向量里检查中断控制位元和中断请求标志位元,因此可将具有高优先顺序的中断控制位元和中断请求标志位元放在前面检测。

举例说明:

ISR:

```

;-----Save ACC-----
  b0xch A,AccBuf ;B0xch instruction do not change C,Z flag
  b0mov A,PFLAG
  B0MOV PflagBuf,A
;-----interrupt first-----
INT0CHK:
  B0BTS1 FT0IEN
  JMP INTTC0CHK
  B0BTS0 FT0IRQ
  JMP INTT0
INTTC0CHK:      ; Check TC0 interrupt request
  B0BTS1 FTC0IEN ; Check TC0IEN
  JMP ISR90 ; Jump to exit of IRQ
  B0BTS0 FTC0IRQ ; Check TC0IRQ
  JMP INTTC0 ; Jump to TC0 interrupt service routine
;-----exit interrupt-----
ISR90:
  B0MOV A, PflagBuf
  B0MOV PFLAG,A ;Restore the PFlag
  B0XCH A,AccBuf ;Restore the Reg. A
      ;B0xch instruction do not change C,Z flag
  RETI      ;Exit the interrupt routine
;-----t0-----
INTT0:
  B0BCLR FT0IRQ
INTT090:
  JMP ISR90
;-----tc0-----
INTTC0:
  B0BCLR FTC0IRQ
INTTC090:
  JMP ISR90

```

#### Q14: 可以在 Interrupt Service Routine 内执行 Call 吗?

因为 MCU Stack 数目有限,在 Interrupt 中执行 Call 要特别注意,如果在进入 Interrupt 时 Stack 已经满了,则会造成 Stack 溢出,而使得程式无法返回。所以若要在 Interrupt 内执行 Call,必需考虑在最差情况下 Stack

的使用数目, 务必使在最差状况下进入 Interrupt 仍有 Stack 空间。

### Q15: 在一个计数器在用作 PWM 输出的时候, 该计数器能否同时用作中断源?

当一个计数器在用作 PWM 输出的时候, 该计数器是可以同时用作中断源的, 不过此时中断溢出的时间是由 PWM 输出频率决定的, 该计数器的计数范围也有 PWM 的计数决定。

### Q16: 片内集成的 PWM 模组可以输出多种频率吗? 是否可控可调?

可以的, 在单片机中, 有多种方法可以改变 PWM 的输出频率。PWM 输出频率与计时器时钟频率, 计时器溢出边界以及计时器初值有关, 受 TCnM, TCnC 寄存器控制。

1. PWM 的时钟源自计时器时钟频率, 而计时器的时钟频率来自 Fcpu 的分频, 由 TCnRATE[2: 0] 决定, 因而改变 TCnRATE[2: 0]可以改变 PWM 的输出频率;
2. 计时器溢出边界受 ALOADn、TCnOUT 位的控制, 通过这两位可选择 4 种溢出边界, 其频率可在较大的范围调整;
3. 可通过在中断中改变计时器初值改变 PWM 输出频率;
4. 利用 TCnOUT 实现 PWM 频率的调整, 只是此时其占空比固定为 1/2;
5. 利用普通 I/O 口输出实现 PWM 输出及其频率的改变。

### Q17: PWM 功能中如何去更新占空比和周期?

PWM 的占空比可通过 TC0R 来改变, 周期可通过 TC0C 来改变; 如下范例所示

```
=====
```

```
; 范例 1: 中断中改变 TC0C 的值可改变 PWM 的频率/周期
```

```
=====
```

```
inttc0:
```

```
    mov a,#20h ;修改 TC0C 的值可以改变 PWM 输出频率
    mov tc0c,a
    b0bclr ftc0irq
```

```
=====
```

```
; 范例 2: 改变 TC0R 的值可改变 PWM 的占空比
```

```
=====
```

```
MOV A, #30H
B0MOV TC0R, A
INCMS BUF0
NOP
B0MOV A, BUF0
B0MOV TC0R, A
```



## Q18: 定时/计数器的时钟源来自哪里? 如何设置计时器的分频数?

定时/计数器 TC0 具有双时钟源, 可根据实际需要选择内部时钟或外部时钟作为计时标准。其中, 内部时钟来自 Fcpu, 外部时钟 INT0 由 P0.0 引脚 (下降沿触发) 输入。定时/计数器的计数频率一般有 8 种选择, 用户可通过设置 T0M/TCnM 寄存器中 T0rate0~T0rate2/TCnrate0~TCnrate2 位来设置定时/计数器的分频 (2~256 分频), 分频数的设置可参考 datasheet。

如:

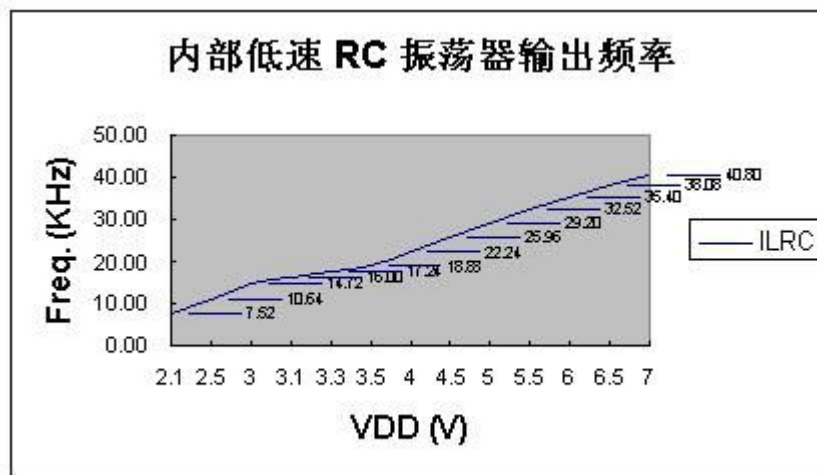
```
MOV A,#01100000B
```

```
B0MOV TC0M,A; TC0 速率=Fcpu/4。
```

对于具有 TC0X8/TC1X8 位的单片机, 可以设置计时器的计数频率信号直接来自外部振荡器频率 (Fosc) 或系统时钟 (Fcpu)。此时分频数可选 1~128。

## Q19: 低速模式的计时器定时和普通模式有何不同? 定时时间怎么计算?

◆ 当系统从普通模式转为低速模式时, (可参考 datasheet 中工作模式说明部分) 系统自带的内部低速振荡器处于运行状态, 其振荡速率可以参考 datasheet 中系统低速时钟章节描述, 一般为 5V 时 32KHz, 3V 时 16KHz。



◆ 若此时客户的程式中使用到计时器, 其系统时钟源 (Fosc) 将不再是外部振荡器 (外挂的晶振/陶振/RC 振荡器), 而变为内部低速振荡器, 计时器计时的计算方法与外接振荡器的方法是相同的。如: 系统工作在 5V 条件下, 则此时 Fosc 将为 32KHz, 由于进入低速模式时, 单片机硬体设置 Slow mode Fcpu=Fosc/4, 若计时器的分频数设置为 1 分频, 则其每计数一次的时间将变为  $1 / (32\text{KHz}/4) = 0.125\text{ms}$ 。

◆ 另外须注意: 在 RTC 模式下, 进入低速模式, T0 的时钟源仍然是外部 32.768K。

## Q20: 使用内置的即时时钟 (RTC) 功能时, 需要注意哪些事项?

1. 执行 B0BSET FT0TB 或 B0BCLR FT0TB 指令即可开启或关闭 RTC 功能;
2. 要实现精准计时需要外接 32768Hz 晶振可实现等分频;
3. 开启 RTC 功能时, T0rate 无效;
4. RTC 模式下, 不能在中断服务程式中重定 T0C;

**Q21: RTC 功能仿真与 IC 应用时, 其设定方式及注意事项为何?**

1. 对于具有 RTC 功能的 MCU, 仿真时, 用户根据所选的 CODE OPTION 选项插入对应的晶体 (选择 IHRC\_RTC, 则插入 16M; 选择 4M\_X'tal, 则插入 4M;), 由于 ICE 上已经焊接了 32768 的晶体, 因而可以直接通过 FT0TB 控制位来仿真 RTC 功能; 对于实际 IC, 2501B 需要选择 IHRC\_RTC, 并且 XIN/XOUT 挂上 32768Hz 晶振;
2. 程式中开启 RTC 功能范例如下:
 

```

B0BCLR FT0ENB ; T0 计时器
B0BCLR FT0IEN ; T0 中断功能禁止.
B0BCLR FT0IRQ ; T0 中断请求标志位元清零.
B0BSET FT0TB ; 选择 T0 RTC 时钟源.
B0BSET FT0IEN ; 使能 T0 中断功能.
B0BSET FT0ENB ; 开启 T0 计时器.

```
3. 开启 RTC 功能后 T0RATE[2:0]位无效。
4. 带 RTC 功能的单片机, 进入中断服务程式后, 清除 T0IRQ 前需要延时 16us 时间。

**Q22: 在 AD 应用时, 软件有那些注意事项?**

1. AD enable 后, 必须 delay 200us 以上, 让 AD 电路稳定后才可开始取样。
2. AD channel 切换后, 必须 delay 200us 以上, 让 AD 信号稳定后才可开始取样。
3. AD 取样获得的数据, 建议先做滤波处理后, 再以其他方式做数据处理。
4. 建议客户在做程序初始化时, enable ADC, 并定时启动 ADC 转换功能。(范例如下)

例如: 每隔 1ms 启动一次 ADC 转换功能

Sys\_Int: ;在程序初始化过程, 开启 ADC 转换

```
mov A, #00h ;设置 AD 工作模式
```

```
b0bset FADENB ;开启 AD 转换功能
```

Sys\_Init90:

```
ret
```

ADC\_Convert:

```
b0btsl f_time_1ms ;判断是否到达 1ms
```

```
jmp ADC_Convert90
```

```
b0bclr f_time_1ms ;1ms 时间到, 启动 ADC 转换功能
```

```
b0bset FADS
```

ADC\_Convert10:

```
b0btsl FEOC
```

```
jmp ADC_Convert10
```

```
b0bclr FEOC
```

```
mov A, ADB
```

```
mov ADC_buf1, A
```

```
mov A, ADM
```

```
mov ADC_buf2, A ;保存 ADC 转换后的数据
```

```
b0bclr FCHS2
```

```
b0bclr FCHS1
b0bclr FCHS0 ;切换 ADC 通道, 准备下一次 ADC 转换
ADC_Convert90:
    Ret
```

### Q23: 使用 BJ8P164A 单片机的 ADC 功能时, 在软件和硬件设计上应注意哪些事项?

#### 1. 软件方面应该注意的:

A/D 信道转换需要一定的稳定时间, 在软件设计时需要作考虑。为了保证每次 AD 转换的稳定与正确, 最好在每次 AD 转换前都重新确定 AD 转换信道、AD 转换分辨率、AD 时钟源选择位, 而且根据所应用场合对所取得资料进行合理的处理。在第一次使能 AD 时(B0BSET FADENB), 要有一个 100us 的 delay, 以保证内部 AD 电路达到稳定状态,

#### 2. 硬件方面应该注意的:

信号源要尽量与 A/D 转换输入端接近, 而且要视芯片输入阻抗添加合适的电容并入信号源输入端。此外需保证 AD 转换基准电压的稳定, 模拟地与数字地要分开或隔离

### Q24: 在芯片进入睡眠模式之前, 应如何设置 ADC 口?

因为 AD 口的输入信号为模拟量, 为了防止漏电流的产生, 可在 I/O 口初始化的时候对 P4CON 寄存器进行设置, 比如 P4.1 固定作为 AD 口使用, 则可将 P4CON.1 置 1, 而无需在睡眠前后做改变, 但要注意的是, 当 P4CON.1 为 1 的时候, P4.1 口将作为单纯的模拟输入口, 而不能作为数字 I/O 口使用。

另外, 在睡眠之前, 需将 AD 功能关闭 (即 ADENB=0) 以减小睡眠电流, 唤醒后需重新开启 AD 功能 (即 ADENB=1), 并做 100us 的延时之后才能进行 AD 采样 (不管在任何条件下, 打开 AD 功能后都要经过至少 100us 的延时之后才能进行 AD 采样)。

### Q25: 带 AD 的单片机 AD 引脚能否输入负电压? 如 vdd=5v 电源时, 当 AD 输入(+5V)(-5V) 时单片机的数据是多少?

MCU 的 AD 输入口不能输入负压。如果使用+5V 参考电压, AD 输入+5V 时, 12 位 AD 输出为 FFFH。

### Q26: 多通道 AD 转换时, 是否会产生 AD 互相干扰的情况?

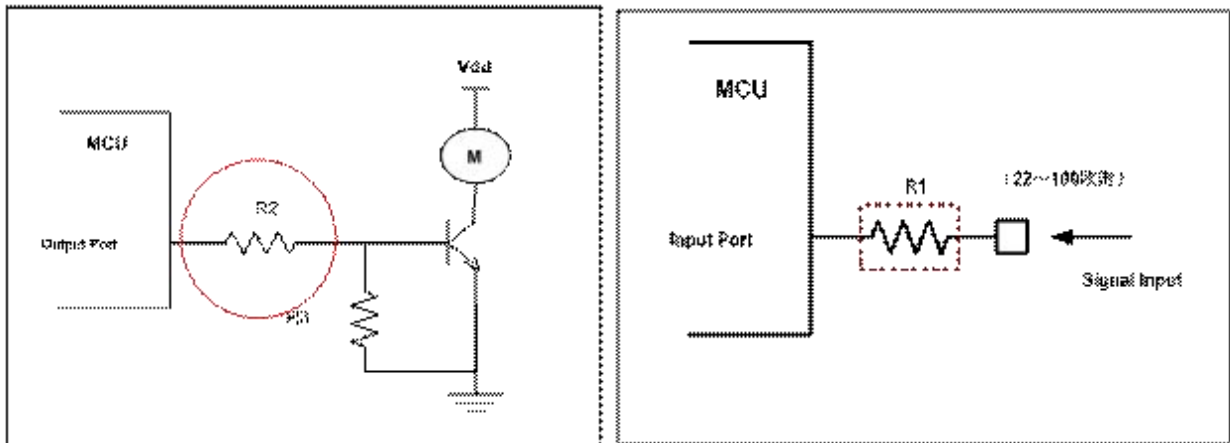
会产生相互干扰的情况。解决方法是, 从一个通道 AD 转换结束到另一个通道 A/D 开始之前, 应该加入一段延时, 以便在 AD 器件的输入口稳定传送资料, 让 AD 器件准确读取。一般 DELAY 100us 以上。

### Q27: 仿真 BJ8P164A 时, 仿真器上的 P0.4 脚为何没有 BUZZER 输出?

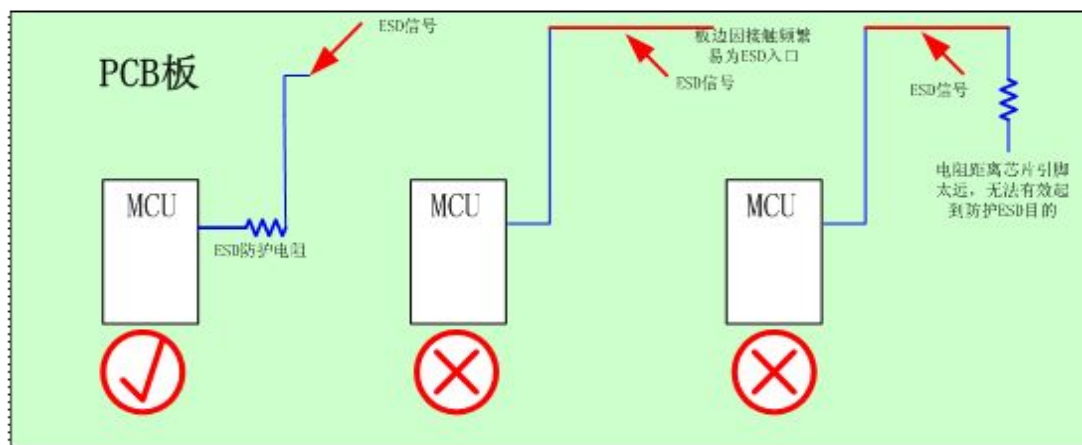
在实际芯片中, BJ8P164A 的 P0.4 和 BZ 脚是复用的, 但是在仿真器上的 P0.4 口仅能仿真普通 I/O 功能, 而 BUZZER 功能则会通过 LCD\_DAT 输出。在仿真 BUZZER 功能时, 请确保蜂鸣器与仿真器的 LCD\_DAT 相连, 如果在连接正确的情况下仍然没有输出, 则要检查程序中对 BZM 等相关寄存器的设置是否正确。

## Q28: 在电路设计及 PCB layout 中, 可采用那些措施以加强芯片抗 ESD 能力?

1. 在电路设计中, 针对数字输入输出及接头连接线的芯片接脚, 建议可串接一适当大小的电阻 (22~100 欧姆) 以达到保护 I/O 口的效果。(请参考下图)



2. 在 PCB layout 中, 应尽量避免 I/O 口直接在 PCB 板边走线。有串电阻的 I/O 口, 应尽量将电阻靠近芯片接脚, 以达到最佳的保护效果。(请参考下图)



## Q29: 芯片为何无法进入 Sleep Mode?

需要进入 Sleep Mode, 可在程序中设置: B0BSET CPUM0。如果无法进入 Sleep Mode, 请检查以下三种情况:

1. 检查 watchdog 的设置, 当选择 Always on 选项时, 系统将无法进入 sleep 模式;
2. 检查 Port0 是否有变化的电平信号输入, 用户需要根据实际情况决定 Port0 是否上拉;
3. 检查是否在程序中设置 Port1 具有唤醒功能, 如果有设置此功能, 还需检测 Port1 是否有变化的电平信号输入。

以上三种情况都可使芯片退出 Sleep Mode, 因此用户需确保外来信号的正确性, 而且作为有唤醒功能的口都要上拉。

### Q30: 如何正确设置进入低速模式?

设置进入低速模式可分为两种情况: 不停掉外部高速振荡器或停掉外部高速振荡器。

1. 用户若对功耗要求不是很高, 可以不停掉外部高速振荡器。

程式设置如下:

```
B0BSET FCLKMD
```

2. 停掉外部高速振荡器以减小功耗。

```
B0BSET FCLKMD
```

```
B0BSET FSTPHX
```

以上指令执行顺序不能颠倒, 若客户在没有设置进入低速模式前停掉外部高速振荡器, 由于此时程式还处于普通模式状态, 振荡器一旦停振, 系统将不再执行, 从而导致出错。

详细资讯可参考 **datasheet** 中系统模式切换和系统低速时钟章节。

### Q31: 在产品开发过程中如何提高 EFT 特性?

要保证芯片的正常运行, 除了自身有一定的抗干扰能力外, 硬件保护也是很重要的。一般情况下从以下几个方面加以保护:

1. 电源端: 保证电源的稳定性, 强电情况下蓄能电路、异常电压保护电路要根据情况加以考虑。芯片 Vdd 与 Vss 端一定也是必须要并入一小容量电容 (建议为 0.1uF), 而且位置要正确, 一定是电源先经过电容再到达芯片电源端。

2. 复位端: 如果采用外部复位方式, 则按照建议的复位电路设计; 如果内部复位方式, 一般情况下只需将 RST 端与 Vdd 端直接相连或复位埠选择为 Input 口使用。如果抗干扰要求较高, 可以采用 RST 端上拉一 100K 欧姆左右的电阻到 Vdd 端或埠选择为 Input 口使用, 并且 RST 端无需添加接地电容, 因为电源中的高频干扰会通过电容而直接侵入芯片内部。此外复位电路在 PCB 布线时, 尽量要靠近复位管脚, 以减小与地端、电源端的耦合强度。

3. 晶振端: 设计振荡电路时, 应该视晶振参数选择合适的匹配电容。一般来说, 芯片工作频率过高, 电容值不易过大。在要求频繁停振、起振的场合下, 可以使 XOUT 端的电容值略大于 XIN 端的电容值, 甚至可以在两端并联一大阻值电阻 R1。另外, 振荡电路尽可能接近芯片振荡脚位元, 并与地线和电源线保持足够的距离 (3mm), 以避免电源高频噪声干扰。

4. I/O 端: 具有高噪声的负载最好以光耦等组件隔离或加有吸噪声电路, 如果有危险的负载, 应加有上拉或下拉电阻以防止微控制器损坏时的误动作。另外, 某些特殊场合由于安全的需要, 具有危险性的负载也可以利用软件脉冲驱动的方式, 透过电容耦合, 以避免微控制器复位或当机时的误启动造成的危险。

### Q32: BJ8P164A 有几级堆栈

有 8 级堆栈

文件更新记录

序号	版本	更改单号	更改前内容	更改后内容
1	A/0	首次发布		