

中断标志清除注意事项

Questions: 中断标志清除有何注意事项?

Answer:

应用为了提升代码执行效率，可能会选择直接操作寄存器方式来实现软件功能，直接操作寄存器方式应用到中断标志清除上，如果清除时机设计不恰当时，就可能出现中断函数被异常多执行一次的现象。

以 EXT1 中断为例，其中断响应机制如下：

- 1) IO 口上出现期望边沿
- 2) 置位中断状态寄存器的 LINE_x 标志
- 3) 置位 NVIC Flag
- 4) 指令跳转到中断函数执行
- 5) 跳转进中断函数时立即清除 NVIC Flag
- 6) 清 LINE_x 标志及中断代码执行
- 7) 退出中断

而实际清除标志命令的执行到标志被真正清除通常需要约 2 个 cycle 的时间（视外设而定，有些外设可能会存在细微差异）。

如果应用代码设计时将清 LINE_x 标志写在中断函数末尾，则在执行清除标志命令后就会立即退出中断函数。此时就可能会出现代码退出中断函数了，但 LINE_x 标志因为未及时被清除而处于置位状态，从而又会从前述中断响应机制的 2) 步骤开始再执行一次循环。最终导致中断被异常地多执行一次。

示例：存在隐患的标志清除写法

```
void EXINT0_IRQHandler(void)
{
    /*
    user code
    */
    EXINT->intsts = EXINT_LINE_0;
}
```

示例：正常标志清除写法

```
void EXINT0_IRQHandler(void)
{
    EXINT->intsts = EXINT_LINE_0;
    /*
    user code
    */
}
```

```
*/  
}
```

当然，如果应用代码调用库函数来清除标志的话，不会存在前述问题，因为指令的跳转是需要时间的，该跳转时间可以满足标志清除的时间要求。

类型： MCU 应用

适用型号： AT32 全系列

主功能： 中断标志清除

次功能： 无

文档版本历史

日期	版本	变更
2022.2.28	2.0.0	最初版本

重要通知 - 请仔细阅读

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途(及其依据任何司法管辖区的法律的对应情况)，或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：(A) 对安全性有特别要求的应用，如：生命支持、主动植入设备或对产品功能安全有要求的系统；(B) 航空应用；(C) 汽车应用或汽车环境；(D) 航天应用或航天环境，且/或(E) 武器。因雅特力产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险由购买者单独承担，并且独力负责在此类相关使用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2022 雅特力科技 (重庆) 有限公司 保留所有权利