

厦门国科安芯科技有限公司

AS32X601 应用笔记

AN002

目录

1. Clearcache.....	3
1.1. 功能说明.....	3
1.2. 使用原则.....	3
1.3. 特殊问题.....	3
1.4. 解决办法.....	3
2. 调试.....	4
2.1. 中断调试.....	4
2.1.1. 问题.....	4
2.1.2. 解决办法.....	4
2.2. 烧录.....	4
2.2.1. 问题.....	4
2.2.2. 解决办法.....	5

1. Clearcache

1.1. 功能说明

该条指令用于同步 CACHE 和 SRAM 之间的数据同步，同时用于控制多主机包括 cpu、dma 以及 csr 定时器之间的时序同步，控制读写顺序。

1.2. 使用原则

- 当通过 DMA 进行接口数据交互过程中，每次向内存写入数据或者读取 DMA 接受数据对应的内存时，均需要在写操作之后，读操作之前添加 clearcache 操作函数。
- 当使用 csr 作为时钟的延时函数时，偶尔会出现延时不准确，此时为 cpu 和 csr 之间的乱序操作不同步，此时需要在调用延时之前添加一条此指令。

1.3. 特殊问题

AS32X601 由于此 cache 可能会导致正常程序操作过程的问题点，应用程序开发过程中，涉及到多层数据传递以及处理过程，在此过程中会出现大量的临时变量声明及定义。

在部分操作中，会涉及形参直接作为数据处理过程的判断或者赋值，由于一个 cache line 为 64bit，如果传输过程出现前后操作 8bit 数据的读写过程，那么可能会造成同一个变量的读写在同一个 cache line,所造成的后果就是变量的赋值还未完成就读出，造成数据错误。

1.4. 解决办法

遇到上述问题时，提供三种方法避免出错：

- 在问题前增加八个 __asm volatile("nop")指令或者短时间的延时，跳过本条 cache line 操作。

- 问题前增加 `__asm volatile("fence.i")` 指令或者 `clearcache` 操作，同步时序；
- 将传递变量修改为 32bit 变量，后续进行数据处理。

后续会在硬件层面解决掉此部分问题，确保开发工作正常运行。

2. 调试

2.1. 中断调试

2.1.1. 问题

在调试中断时，会出现断点在中断处理函数外，此时中断过程比较频繁时，会出现中断无法触发。

造成此现象的原因是，芯片内部中断触发信号为上升沿，如果在断点过程中，中断控制器正常运行，此时触发的中断由于断点的原因无法响应，因此会在中断控制器前阻塞为一个高信号，导致后续中断无法响应。

2.1.2. 解决办法

当前阶段提供三种解决办法：

- 在类似于定时器或者持续数据通信中断时，分步调试，将断点加在相应中断处理函数中，每个中断单独进行调试，确定完整后整体全速运行
- 控制中断频率，确保断点再次运行后产生下一次中断。
- 在相应断点前，关闭中断控制器时钟或者中断使能，调试完该处之后，重新打开时钟或者使能。

2.2. 烧录

2.2.1. 问题

在烧录或者进入 debug 模式时，经常会弹出 `cpu can not stop`，造成此问题的原因是程序操作了未开时钟的外设或者地址，那么此时 `cpu` 或者总线就会完全卡

死在该状态下，不在响应任何指令。

2.2.2. 解决办法

对于代码中操作了未开时钟的总线或者外设，那么此时需要将 **boot** 引脚切换到从其他地址启动，确保 **cpu** 能够被正确 **halt**，烧写完正常代码后，恢复 **boot** 引脚模式即可。