Cortex-M3 内核 HardFault 错误调试定位方法

1、首先更改 startup.s 的启动文件，把里面的 HardFault_Handler 代码段换成下面的代码：

```
HardFault_Handler\
                PROC
                IMPORT  hard_fault_handler_c
                TST LR, #4
                ITE EQ
                MRSEQ R0, MSP
                MRSNE R0, PSP
                B   hard_fault_handler_c
                ENDP
```

2、然后把 hard_fault_handler_c 函数放在 c 文件的代码中。代码如下：

```
void hard_fault_handler_c(unsigned int * hardfault_args)
{
    static unsigned int stacked_r0;
    static unsigned int stacked_r1;
    static unsigned int stacked_r2;
    static unsigned int stacked_r3;
    static unsigned int stacked_r12;
    static unsigned int stacked_lr;
    static unsigned int stacked_pc;
    static unsigned int stacked_psr;
    static unsigned int SHCSR;
    static unsigned char MFSR;
    static unsigned char BFSR;
    static unsigned short int UFSR;
    static unsigned int HFSR;
    static unsigned int DFSR;
    static unsigned int MMAR;
    static unsigned int BFAR;

    stacked_r0 = ((unsigned long) hardfault_args[0]);
    stacked_r1 = ((unsigned long) hardfault_args[1]);
    stacked_r2 = ((unsigned long) hardfault_args[2]);
    stacked_r3 = ((unsigned long) hardfault_args[3]);
    stacked_r12 = ((unsigned long) hardfault_args[4]);
    /*异常中断发生时，这个异常模式特定的物理 R14,即 lr 被设置成该异常模式将要返回的地址*/
    stacked_lr = ((unsigned long) hardfault_args[5]);
    stacked_pc = ((unsigned long) hardfault_args[6]);
    stacked_psr = ((unsigned long) hardfault_args[7]);
```

```
    SHCSR = (*((volatile unsigned long *)(0xE000ED24))); //系统 Handler 控制及状态寄存器
    MFSR = (*((volatile unsigned char *)(0xE000ED28)));   //存储器管理 fault 状态寄存器
    BFSR = (*((volatile unsigned char *)(0xE000ED29)));   //总线 fault 状态寄存器
    UFSR = (*((volatile unsigned short int *)(0xE000ED2A)));//用法 fault 状态寄存器
    HFSR = (*((volatile unsigned long *)(0xE000ED2C)));   //硬 fault 状态寄存器
    DFSR = (*((volatile unsigned long *)(0xE000ED30)));   //调试 fault 状态寄存器
    MMAR = (*((volatile unsigned long *)(0xE000ED34))); //存储管理地址寄存器
    BFAR = (*((volatile unsigned long *)(0xE000ED38))); //总线 fault 地址寄存器
    while (1);
}
```

3、执行程序后，若发生内核错误，则程序会运行到最后的 while(1)处。此时观察相应的堆栈和故障寄存器值， stacked_lr 即为故障发生时进入故障中断前 pc 的值，在 MDK 软件调试状态下，假如 stacked_lr 的值为 0x1A002D08，在左下方的命令窗口输入“pc = 0x1A002D08”，回车，即可定位发生错误的代码位置。

4、根据内核错误状态寄存器的值，对应下面的说明，也可以看出是发生了何种内核错误。

附录：Cortex-M3 内核错误寄存器说明

表D.17 系统Handler控制及状态寄存器SHCSR 0xE000_ED24

位段	名称	类型	复位值	描述
18	USGFAULTENA	R/W	0	用法 fault 服务例程使能位
17	BUSFAULTENA	R/W	0	总线 fault 服务例程使能位
16	MEMFAULTENA	R/W	0	存储器管理 fault 服务例程使能位
15	SVCALLPENDED	R/W	0	SVC 悬起中。本来已经要 SVC 服务例程，但是却被更高优先级异常取代
14	BUSFAULTPENDED	R/W	0	总线 fault 悬起中，细节同上。
13	MEMFAULTPENDED	R/W	0	存储器管理 fault 悬起中，细节同上
12	USGFAULTPENDED	R/W	0	用法 fault 悬起中，细节同上
11	SYSTICKACT	R/W	0	SysTick 异常活动中
10	PENDSVACT	R/W	0	PendSV 异常活动中
9	-	-	-	-
8	MONITORACT	R/W	0	Monitor 异常活动中
7	SVCALLACT	R/W	0	SVC 异常活动中
6:4	-	-	-	-
3	USGFAULTACT	R/W	0	用法 fault 异常活动中
2	-	-	-	-
1	BUSFAULTACT	R/W	0	总线 fault 异常活动中
0	MEMFAULTACT	R/W	0	存储器管理 fault 异常活动中

表 D.18 存储器管理 fault 状态寄存器(MFSR) 0xE000_ED28

位段	名称	类型	复位值	描述
7	MMARVALID	-	0	=1 时表示 MMAR 有效
6:5	-	-	-	-
4	MSTKERR	R/Wc	0	入栈时发生错误
3	MUNSTKERR	R/Wc	0	出栈时发生错误
2	-	-	-	-
1	DACCVIOL	R/Wc	0	数据访问违例
0	IACCVIOL	R/Wc	0	取指访问违例

表 D.19 总线 fault 状态寄存器(BFSR) 0xE000_ED29

位段	名称	类型	复位值	描述
7	BFARVALID	-	0	=1 时表示 BFAR 有效
6:5	-	-	-	-
4	STKERR	R/Wc	0	入栈时发生错误
3	UNSTKERR	R/Wc	0	出栈时发生错误
2	IMPRECISERR	R/Wc	0	不精确的数据访问违例（violation）
1	PRECISERR	R/Wc	0	精确的数据访问违例
0	IBUSERR	R/Wc	0	取指时的访问违例

表 D.20 用法 fault 状态寄存器(UFSR)，地址：0xE000_ED2A

位段	名称	类型	复位值	描述
9	DIVBYZERO	R/Wc	0	表示除法运算时除数为零（只有在 DIV_0_TRP 置位时才会发生）
8	UNALIGNED	R/Wc	0	未对齐访问导致的 fault
7:4	-	-	-	-
3	NOCP	R/Wc	0	试图执行协处理器相关指令
2	INVPC	R/Wc	0	在异常返回时试图非法地加载 EXC_RETURN 到 PC。包括非法的指令，非法的上下文以及非法的值。The return PC 指向的指令试图设置 PC 的值（要理解此位的含义，还需学习后面的讨论中断级异常的章节）
1	INVSTATE	R/Wc	0	试图切入 ARM 状态
0	UNDEFINSTR	R/Wc	0	执行的指令其编码是未定义的——解码不能

表 D.21 硬 fault 状态寄存器 0xE000_ED2C

位段	名称	类型	复位值	描述
31	DEBUGEVT	R/Wc	0	硬 fault 因调试事件而产生
30	FORCED	R/Wc	0	硬 fault 是总线 fault，存储器管理 fault 或是用法 fault 上访的结果
29:2	-	-	-	-
1	VECTBL	R/Wc	0	硬 fault 是在取向量时发生的
0	-	-	-	-

表 D.22 调试 fault 状态寄存器(DFSR) 0xE000_ED30

位段	名称	类型	复位值	描述
4	EXTERNAL	R/Wc	0	EDBGREQ 信号有效
3	VCATCH	R/Wc	0	发生向量加载
2	DWTTRAP	R/Wc	0	发生 DWT 匹配
1	BKPT	R/Wc	0	执行到 BKPT 指令
0	HALTED	R/Wc	0	在 NVIC 中请求 HALT

表 D.23 存储管理地址寄存器(MMAR) 0xE000_ED34

位段	名称	类型	复位值	描述
31:0	MMAR	R	-	触发存储管理 fault 的地址

表 D.24 总线 fault 地址寄存器(BFAR) 0xE000_ED38

位段	名称	类型	复位值	描述
31:0	BFAR	R	-	触发总线 fault 的地址