

# S32K1xx 上的异常和故障检查

作者: 恩智浦半导体

## 目录

## 1. 介绍

S32K1xx 产品系列进一步扩展了汽车行业中高度可扩展的 ARM® Cortex®-M0+/M4F MCU 产品组合。它建立在 KEA 系列的传承基础上, 并引入了更高的内存选项以及更丰富的外设集, 将功能扩展到各种汽车应用中。S32K 产品系列器件采用 2.70-5.5V电源并专注于汽车环境下的鲁棒/健壮性, 非常适合用于工作在电气恶劣环境中的各种应用, 并针对成本敏感的应用进行了优化, 提供低引脚数选项。S32K 产品系列提供多样的存储器、外围设备和封装选项。它们共享通用外设和引脚数, 允许开发人员在 MCU 系列内或 MCU 系列之间轻松迁移, 以充分利用更多内存或集成功能。这种可扩展性允许开发人员使用 S32K 产品系列作为他们平台的标准, 最大限度地提高硬件和软件的重复用率并缩短产品的开发周期与上市时间。

1. 介绍 .....	1
2. Cortex-M4处理器内核寄存器 .....	1
3. 堆栈的结构和选择 .....	4
4. 案例示例 .....	7
5. 结论 .....	8
6. 参考资料 .....	8

## 2. Cortex-M4处理器内核寄存器

处理器具有以下32位寄存器:

- 13个通用寄存器, r0-r12



- 堆栈指针 (SP) 的别名 - 存储区寄存器、PSP 和 MSP
- 链接寄存器 (LR), r14
- 程序计数器 (PC), r15
- 专用程序状态寄存器, (xPSR)
- 异常掩码寄存器
- 控制寄存器

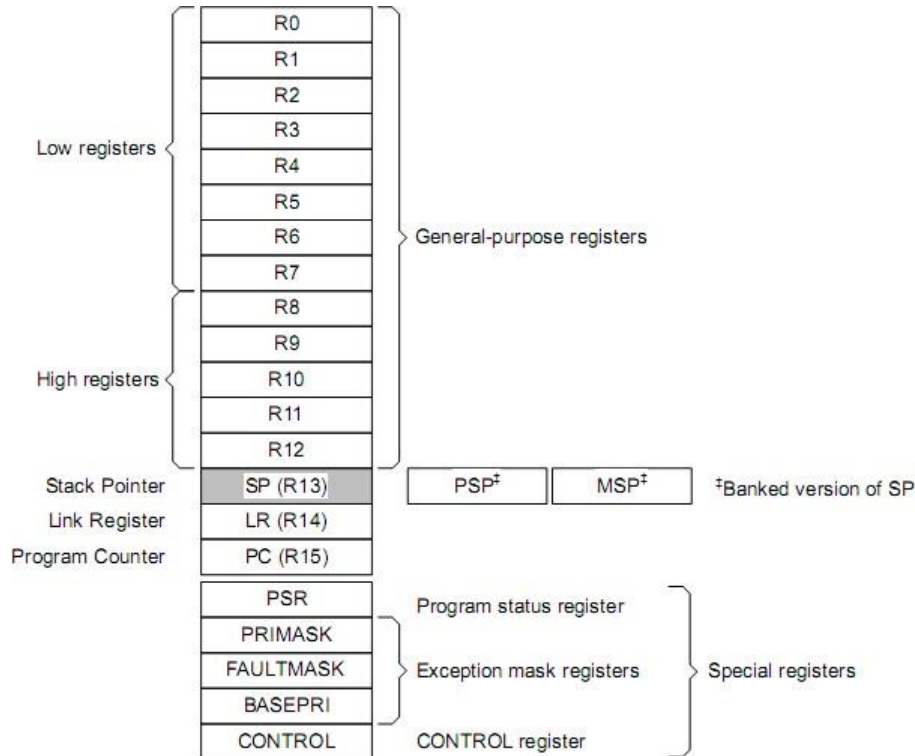


图1. 处理器寄存器集

通用寄存器 r0-r12 没有特殊的架构定义上的用途。大多数可以指定通用寄存器的指令也可以指定 r0-r12。

表1. 通用寄存器

低寄存器	所有指定通用寄存器的指令都可以访问寄存器 r0-r7。
高寄存器	寄存器 r8-r12 可由所有指定通用寄存器的 32 位指令访问。 所有 16 位指令都无法访问寄存器 r8-r12。

寄存器 r13、r14 和 r15 具有以下特殊功能:

表2. SP LR 和 PC 寄存器

堆栈指针	寄存器 r13 用作堆栈指针 (SP)。因为 SP 忽略对位 [1:0] 的写入，所以它会自动对齐到一个字、四字节边界。 处理程序模式始终使用 MSP，但您可以配置Thread模式来使用 MSP 或 PSP。
链接寄存器	寄存器 r14 是子程序链接寄存器 (LR)。 当执行分支链接 (BL) 或分支链接交换 (BLX) 指令时，LR 从 PC 接收返回地址。 LR 也用于异常返回。 在所有其他时间，您可以将 r14 视为通用寄存器。
程序计数器	寄存器 r15 是程序计数器 (PC)。 位 [0] 始终为 0，因此指令始终与字或半字边界对齐。

对于其他寄存器，详细信息如下。

表3. 特殊寄存器

程序状态寄存器	应用程序状态寄存器 (APSR) 中断程序状态寄存器 (IPSR) 执行程序状态寄存器 (EPSR)
优先屏蔽寄存器	PRIMASK 寄存器防止激活所有具有可配置优先级的异常
故障屏蔽寄存器	FAULTMASK 寄存器防止激活除不可屏蔽中断 (NMI) 之外的所有异常
基本优先级屏蔽寄存器	BASEPRI 寄存器定义了异常处理的最低优先级。当 BASEPRI 设置为非零值时，它会阻止激活与 BASEPRI 值具有相同或更低优先级级别的所有异常。
控制寄存器	当处理器处于Thread模式时，CONTROL 寄存器控制使用的堆栈和软件执行的特权级别，如果执行，则指示 FPU 状态是否处于活动状态。

处理器支持两种操作模式，Thread模式和Handler模式：

- Thread模式：用于执行应用软件。处理器退出复位后进入Thread模式。
- Handler模式：用于处理异常。处理器在完成所有异常处理后返回Thread模式。

软件执行的权限级别是：

**无特权：**软件：

- 对 msr 和 mrs 指令的访问受限，不能使用 cps 指令
- 无法访问系统定时器、NVIC或系统控制块
- 可能对内存或外围设备的访问受限

非特权软件在非特权级别执行。

在Thread模式下，控制寄存器控制软件执行是特权还是非特权。在 Handler 模式下，软件执行总是有特权的。

**特权：**

- 该软件可以使用所有指令并可以访问所有资源。
- 特权软件在特权级别执行。

处理器在复位时或作为异常返回的结果进入Thread模式。特权和非特权代码可以在Thread模式下运行。作为异常的结果，处理器进入处理程序模式。所有代码都在 Handler 模式下享有特权。代码可以以特权或非特权身份执行。非特权执行限制或排除对某些资源的访问。特权执行可以访问所有资源。处理程序模式始终具有特权。Thread模式可以是特权模式，也可以是非特权模式。

### 3. 堆栈的结构和选择

MSP是默认的堆栈指针，在复位时通过从存储器的第一个字加载值进行初始化。对于简单的应用程序，MSP会一直使用。在这种情况下，只有一个堆栈区域。对于可靠性要求较高的系统，通常会涉及嵌入式操作系统，并定义多个堆栈区域。

当处理器发生异常时，除非异常是尾链异常或迟到异常，否则处理器会将信息压入当前堆栈。八个数据字的结构称为堆栈帧。堆栈帧包括返回地址。这是被中断程序中下一条指令的地址。该值在异常返回时恢复到电脑。这样被中断的程序就可以恢复了。与堆栈操作并行，处理器执行向量提取，从向量表中读取异常处理程序的起始地址。堆叠完成后，处理器开始执行异常处理程序。同时，处理器向LR写入一个EXC\_RETURN值。这表明哪个堆栈指针对应于堆栈帧以及处理器在条目发生之前处于哪种操作模式。

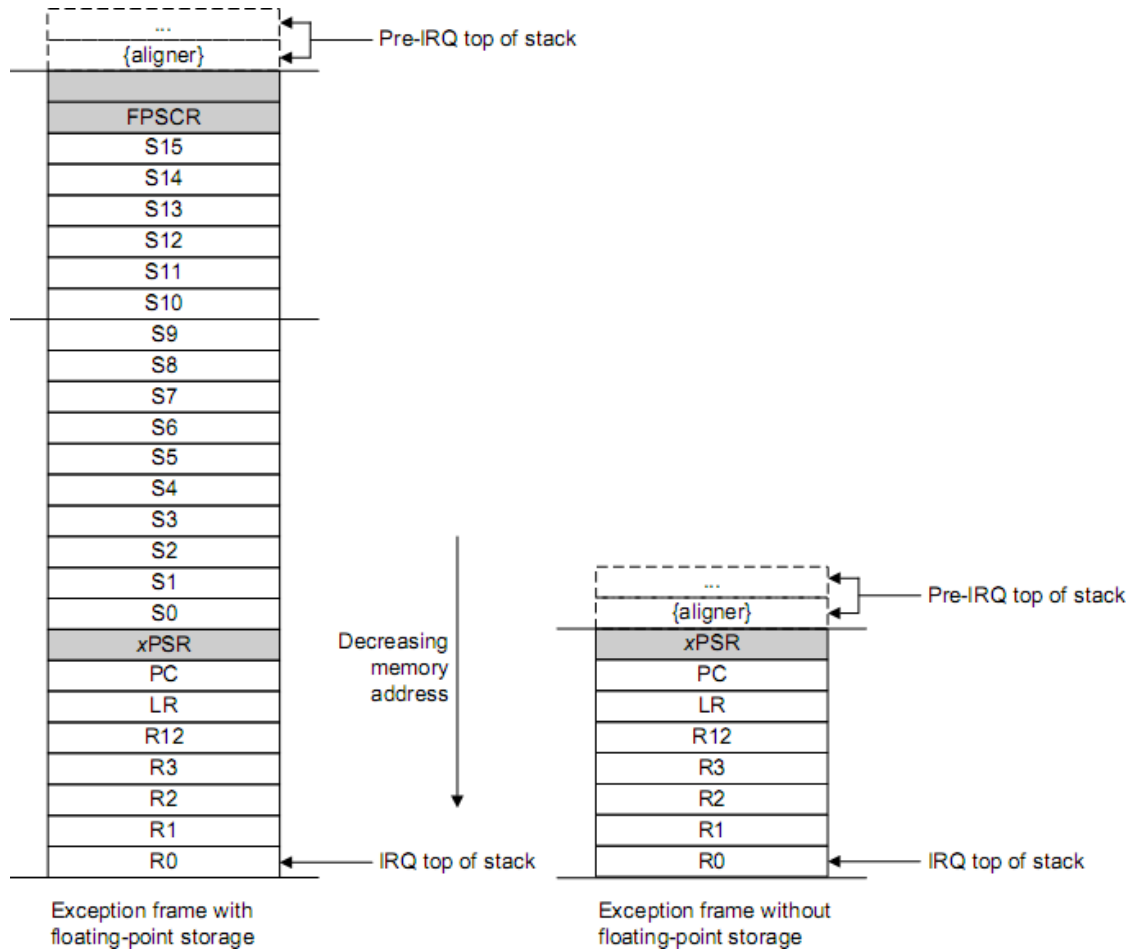


图2. 堆栈帧

EXC\_RETURN是异常条目时加载到 LR 中的值。异常机制依赖此值来检测处理器何时完成了异常处理程序。该值的最低五位提供有关返回堆栈和进程模式的信息。

EXC_RETURN[31:0]	Description
0xFFFFFFFF1	Return to Handler mode, exception return uses non-floating-point state from the MSP and execution uses MSP after return.
0xFFFFFFFF9	Return to Thread mode, exception return uses non-floating-point state from MSP and execution uses MSP after return.
0xFFFFFFFFD	Return to Thread mode, exception return uses non-floating-point state from the PSP and execution uses PSP after return.
0xFFFFFE1	Return to Handler mode, exception return uses floating-point-state from MSP and execution uses MSP after return.
0xFFFFFE9	Return to Thread mode, exception return uses floating-point state from MSP and execution uses MSP after return.
0xFFFFFED	Return to Thread mode, exception return uses floating-point state from PSP and execution uses PSP after return.

图3. 异常返回行为

堆栈指针的切换由软件或异常的进入或退出来选择。

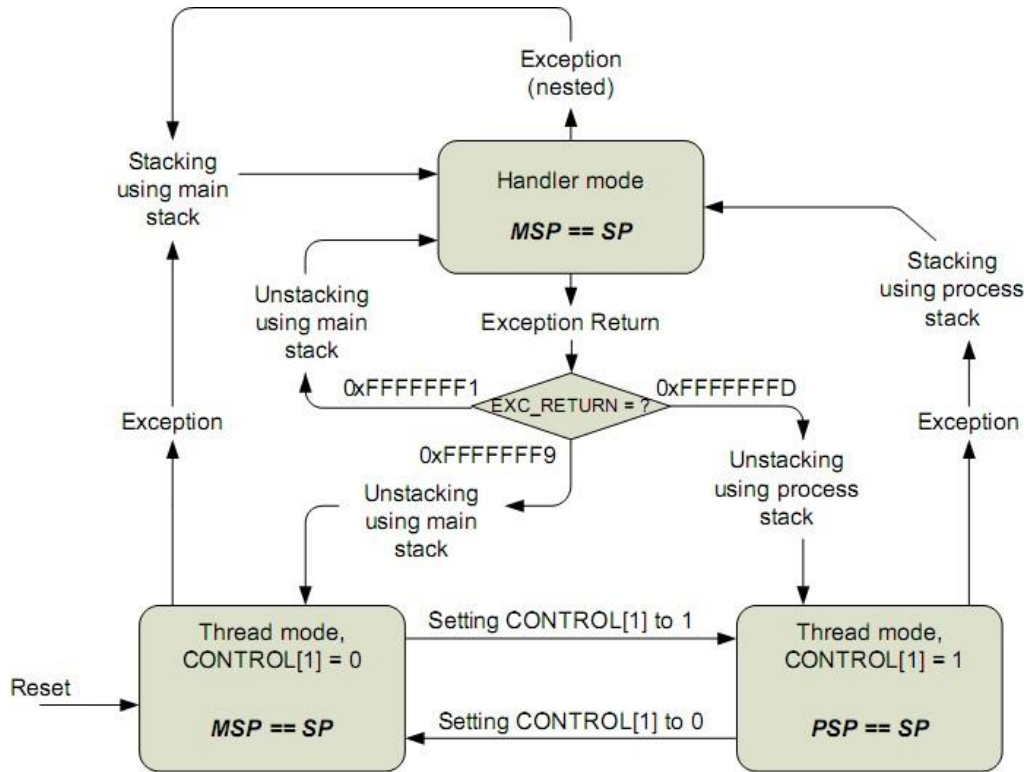


图4. 通过软件或异常进入/退出切换堆栈指针选择

## 4. 案例示例

基于以上介绍，以下假设了两种情况，并提出了解决方法。

### 案例一：记录执行中断的次数

默认中断处理程序通常实现为无限循环。如果应用程序以这样的默认处理程序结束，则需要确定实际执行的是哪个中断。

下面的代码显示了如何在默认无限循环处理程序中添加一些指令，以在进入无限循环之前将执行中断的编号加载到 r2 中。以这种方式从NVIC读取的中断号与矢量表的开始有关。

```
DefaultISR:
    ldr r3, NVIC_INT_CTRL_CONST
    ldr r2, [r3, #0]
    uxtb r2, r2

Infinite_loop:
    b Infinite_loop

NVIC_INT_CTRL_CONST: .word 0xe000ed04
```

### 案例2：记录触发故障的地址

有时应用中会产生故障，如果没有进程触发这些故障，程序就会死锁。稳健的设计将增加超时机制，以检查死锁的状态。还可以使用另一种方法，如记录故障地址以备将来检查，并覆盖生成故障的地址。

ARM Cortex-M 内核实现了一组故障异常，如下所示。

```
.long HardFault_Handler          /* Hard Fault Handler*/
.long MemManage_Handler         /* MPU Fault Handler*/
.long BusFault_Handler         /* Bus Fault Handler*/
.long UsageFault_Handler       /* Usage Fault Handler*/
```

每个异常都与一个错误条件相关。如果发生错误，ARM Cortex-M 内核将停止执行当前指令并跳转到异常处理函数。这种机制就像用于中断的机制一样，ARM Cortex-M 内核在接受中断时分支到中断处理程序。

下面的代码显示了如何向硬故障处理程序添加一些指令并修改存储在堆栈中 LR 位置的值。在硬故障处理程序执行并弹出帧后，电脑将指向产生硬故障的指令之后的下一条指令。用户可以在其中添加变量来记录产生故障的地址。

```
void HardFaultHandler(void)
{
    __asm volatile
    (
        "mov r1,r13    \n"
        "ldr r2,=0x18  \n"
        "add r3,r1,r2  \n"
        "ldr r0,[r3]   \n"
        "ldr r2,=0x4   \n"
        "add r0,r0,r2  \n"
        "str r0,[r3]   \n"
    );
}
```

## 5. 结论

S32K1xx产品系列是汽车行业中高度可扩展的 ARM Cortex-M0+/M4F MCU 产品组合。本文档通过堆栈机制解释了Cortex-M0+/M4F的寄存器，还引入了异常和故障处理程序的技能，这也可以增加设计的稳健性。

## 6. 参考资料

- [S32K1xx Series Reference Manual \(Rev.6\)](#)
- [Cortex-M4 Device: Generic User Guide](#)





**How to Reach Us:**

**Home Page:**  
[nxp.com](http://nxp.com)

**Web Support:**  
[nxp.com/support](http://nxp.com/support)

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C 5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and  $\mu$ Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2018 NXP B.V.

Document Number: AN12201  
Rev. 0  
07/2018

