

3 Cortex-M3 内核级外设

本章介绍 Stellaris®系列微控制器中 Cortex-M3 处理器的内核级外设，包括：

- 系统定时器 SysTick（见[第 118 页](#)）
提供一个简单易用、配置灵活的 24 位单调递减计数器。该计数器具有写入即清零、过零自动重载等特性。
- 嵌套式向量化中断控制器（Nested Vectored Interrupt Controller，简称为 NVIC）
 - 可实现异常及中断的快速响应处理
 - 可控制电源管理
 - 实现系统控制寄存器
- 系统控制模块（System Control Block，简称为 SCB）（见[第 121 页](#)）
可提供系统构成信息，并进行系统控制，包括系统异常的配置、控制以及上报。
- 存储器保护单元（MPU）（见[第 121 页](#)）
支持标准的 ARMv7 受保护存储器系统架构（PMSA）模型。MPU 为保护区、重叠保护区、访问权限提供了完善的支持，并且支持将存储器属性导出到系统。

第 118 页的[表 3-1](#)列出了私有外设总线（Private Peripheral Bus，简称为 PPB）的地址映射。某些外设寄存器空间还会进一步划分为两个区，在表中分别列为两组地址。

表 3-1. 内核级外设寄存器分布

地址	内核级外设	详见
0xE000 E010~0xE000 E01F	系统定时器	第 118 页
0xE000 E100~0xE000 E4EF 0xE000 EF00~0xE000 EF03	嵌套式向量化中断控制器	第 119 页
0xE000 E008~0xE000 E00F 0xE000 ED00~0xE000 ED3F	系统控制模块	第 121 页
0xE000 ED90~0xE000 EDB8	存储器保护单元	第 121 页

3.1 功能描述

本章介绍部分 Cortex-M3 处理器的内核级外设，在 Stellaris®系列微控制器中的实现，包括 SysTick、NVIC、SCB 以及 MPU。

3.1.1 系统定时器（SysTick）

Cortex-M3 内核集成有一个系统定时器 SysTick，提供简单易用、配置灵活的 24 位单调递减计数器，还具有写入即清零、过零自动重载等特性。该计数器的用途广泛，举例来说，可以：

- 用作 RTOS 的节拍定时器，按照可编程的频率（例如 100Hz）定时触发，调用系统定

定时器服务子程序：

- 用作高速报警定时器，采用系统时钟作为时钟源；
- 用作频率可变的报警或信号定时器——其周期取决于所采用的参考时钟源以及计数器的动态范围；
- 用作简单计数器，测量任务的完成时刻、总体耗时等等；
- 用于实现基于失配/匹配周期的内部时钟源控制。此时通过查询 **STCTRL** 控制及状态寄存器的 **COUNT** 标志位，可以判定某个动作是否在指定的时间内完成；以此作为动态时钟管理控制环的一部分。

系统定时器包含以下 3 个寄存器：

- **系统定时器控制及状态寄存器 (STCTRL)**：该寄存器用于配置系统定时器的时钟、使能计数器、使能 SysTick 中断、判定计数器状态；
- **系统定时器重载值寄存器 (STRELOAD)**：该寄存器包含计数器重载值，每当计数器过零时自动重载；
- **系统定时器当前值寄存器 (STCURRENT)**：该寄存器包含计数器的当前值。

使能系统定时器后，计数器将在每个时钟递减一次，从重载值逐个递减到 0，之后在下一个时钟沿翻转（重载 **STRELOAD** 寄存器的值），之后继续每个时钟递减一次，如此周而复始。如果将 **STRELOAD** 寄存器清零，则会在下次重载时终止计数器的运行。当计数器递减到 0 时，**COUNT** 标志位将置位。读取 **COUNT** 标志位后其自动清零。

对 **STCURRENT** 寄存器进行写操作，即可将此寄存器清零，同时还将清零 **COUNT** 标志位。这个写操作并不会触发 **SysTick** 异常逻辑。读取该寄存器时，返回值是该寄存器被访问时刻的内容。

系统定时器的计数器是按照处理器时钟运行的。假如在某些低功耗模式下停止提供该时钟信号，则系统定时器的计数器也将停止运行。软件在访问系统定时器的寄存器时，应确保始终采用字对齐操作予以访问。

注：在调试过程中若处理器暂停，那么此计数器也不再递减。

3.1.2 嵌套式向量化中断控制器 (NVIC)

本节介绍嵌套式向量化中断控制器 (NVIC) 及其寄存器。NVIC 支持：

- 53 个中断；
- 每个中断的优先级均可编程，取值范围 0~7。优先级数字越大则其优先级越低，也就是说 0 代表最高优先级；
- 可实现异常及中断的快速响应处理；
- 中断信号可以是电平检测或脉冲检测；
- 动态重设中断优先级；
- 优先级可分组，划分为分组优先级域以及子优先级域；
- 支持咬尾中断；

- 提供一个外部的不可屏蔽中断（NMI）。

处理器在异常入口处能够自动将状态入栈，在退出异常时能够自动将状态出栈。这个过程是处理器自行完成的、无需多余的指令开销，因此可快速响应异常并进行处理。

3.1.2.1 电平式中断及脉冲式中断

处理器支持电平式中断及脉冲式中断。脉冲式中断通常又称为边沿触发中断。

对于电平式中断而言，只要外设产生中断信号就会始终保持处于触发状态，直到外设中断信号复原后才不再触发。一般来说这需要 ISR（中断服务子程序）对外设进行操作，使得外设不再产生中断请求信号。脉冲式中断则在处理器时钟的上升沿同步采样中断信号，因此为了确保 NVIC 能够成功检测到中断，外设所产生的中断信号必须保持至少一个时钟周期，在此期间 NVIC 可检测到脉冲并锁存中断。

当处理器进入 ISR 后，将自动清除该中断的挂起状态（参见第 120 页的[“中断的硬件控制及软件控制”](#)一节）。对于电平式中断，假如处理器从 ISR 返回后、中断信号仍未复原，则中断将再次被判定为挂起，于是处理器将再次运行其 ISR。外设可以像这样保持中断信号持续享用服务，直到其不再需要服务为止。

3.1.2.2 中断的硬件控制及软件控制

Cortex-M3 内核能够锁存所有中断。当满足以下条件之一，外设中断即变为挂起状态：

- NVIC 检测到某个中断信号为高电平，并且该中断为未激活状态；
- NVIC 检测到某个中断信号的上升沿；
- 软件对中断设置挂起寄存器的相应位写 1，或向软件触发中断寄存器（SWTRIG）的相应位写 1，于是形成软件产生中断的挂起。详见第 138 页关于 PEND0 寄存器中 INT 标志位的介绍，以及第 145 页关于 SWTRIG 寄存器的介绍。

中断挂起后将保持挂起状态，直到满足以下条件之一：

- 处理器进入该中断的 ISR，将中断状态由挂起改为已激活，于是：
 - 对于电平式中断，当处理器从 ISR 返回后，NVIC 将再次采样中断信号。假如仍然检测到中断信号，那么中断状态将再次变为挂起，使得处理器立即重新进入该 ISR。否则，中断状态将变为未激活。
 - 对于脉冲式中断，处理器时刻监视着中断信号的状态。只要检测到中断脉冲信号就会将中断状态改为挂起并激活。在此情况下，当处理器从 ISR 返回后，中断状态将再次变为挂起，使得处理器立即重新进入该 ISR。

如果当处理器在 ISR 内期间并未产生中断脉冲信号，那么当处理器从 ISR 返回后，中断状态将变为未激活。

- 软件对中断清除挂起寄存器中的相应标志位执行写操作
 - 对于电平式中断，若仍旧产生中断信号，那么中断状态将保持不变。否则，中断状态将变为未激活。

- 对于脉冲式中断，假如中断状态是挂起并激活，则将变为未激活。

3.1.3 系统控制模块 (SCB)

系统控制模块提供系统构成信息，并能实现系统控制功能，包括系统异常的配置、控制以及上报。

3.1.4 存储器保护单元 (MPU)

本节介绍存储器保护单元 (MPU)。MPU 将存储器映射空间划分为若干个存储区，并分别规定每一个存储区的起始地址、大小、访问权限以及存储属性。MPU 支持为每一个存储区分别定义其属性设置，支持重叠区的设置，此外还能将存储属性导出到系统。

存储属性影响对该存储区进行访问时的表现。Cortex-M3 的 MPU 定义了 8 个相互独立的存储区，分别编号为 0~7，此外还定义了一个背景区。

当存储区出现重叠时，重叠部分的访问将由编号最大的存储区属性决定。例如，第 7 存储区始终优先于其它存储区，因此一旦与其它区发生重叠时，都将以第 7 区的存储属性为准。

背景区的存储器访问属性与默认的存储器映射相同，但只允许特权级软件依此进行访问。

Cortex-M3 的 MPU 存储器映射是统一的，也就是说指令访问和数据访问都是共用相同的存储区设置。

假如程序试图访问某个地址，而该地址被 MPU 设置为禁止访问，那么处理器将产生一个存储管理故障，并由此触发故障异常。这个异常可能会导致操作系统环境下某个进程终止运行。在操作系统环境下，操作系统内核能够按照进程的实际需求来动态更新 MPU 区的设置。一般来说，嵌入式操作系统都需要通过 MPU 实现存储器保护。

MPU 区的配置是基于存储器类型的，参见第 96 页的“存储区、类型及属性”一节。

第 121 页的表 3-2 列出了 MPU 区的可能属性。至于如何配置微控制器的指南，可参阅第 125 页的“Stellaris® 微控制器的 MPU 配置”一节。

表 3-2. 存储器属性摘要

存储器类型	描述
严格顺序	对严格顺序存储器的所有访问必须按照程序顺序进行。
设备	存储器映射外部设备
普通	普通存储器

为避免出现无法预料的执行结果，如果某些中断的处理函数可能访问某存储区，那么在更新该存储区的属性之前，应当先关闭这些中断。

在访问 MPU 寄存器时，应确保软件按照正确的宽度进行对齐访问：

- 除 MPU 区属性及大小寄存器 (MPUATTR) 外，所有 MPU 寄存器必须按字对齐进行访问；
- MPUATTR 寄存器可按字节对齐、半字对齐或字对齐进行访问。

处理器不支持对 MPU 寄存器进行未对齐访问。

在配置 MPU 时，由于 MPU 可能曾经被编程过，因此所有当前不用的存储区都应当禁用，防止存储区的旧设置对当前更新后的 MPU 设置产生不良影响。

3.1.4.1 更新单个 MPU 区

要更新某个 MPU 区的属性，必须更新 MPU 区编号寄存器 (MPUNUMBER)、MPU 区基地址寄存器 (MPUBASE) 以及 MPUATTR 寄存器。用户可以分别编程各个寄存器，也可以利用多字写操作来同时编程所有寄存器。利用 MPUBASEx、MPUATTRx 别名寄存器，可以通过一条 STM 指令来同时编程最多 4 个区的配置。

采用单字写操作，更新单个 MPU 区

下面的例程可以完成单个 MPU 区的配置：

```
; R1 = 存储区编号
; R2 = 大小/是否使能
; R3 = 属性
; R4 = 地址
LDR R0, =MPUNUMBER ; 0xE00ED98, MPU区编号寄存器
STR R1, [R0, #0x0] ; 区编号
STR R4, [R0, #0x4] ; 区基地址
STRH R2, [R0, #0x8] ; 区大小以及是否使能
STRH R3, [R0, #0xA] ; 区属性
```

假如之前使能了某个存储区，又需要更改其设置，那么在向 MPU 写入新的设置之前应当先将此存储区禁用。例如：

```
; R1 = 存储区编号
; R2 = 大小/是否使能
; R3 = 属性
; R4 = 地址
LDR R0, =MPUNUMBER ; 0xE00ED98, MPU区编号寄存器
STR R1, [R0, #0x0] ; 区编号
BIC R2, R2, #1 ; 禁用该区
STRH R2, [R0, #0x8] ; 区大小以及是否使能
STR R4, [R0, #0x4] ; 区基地址
STRH R3, [R0, #0xA] ; 区属性
ORR R2, #1 ; 使能该区
STRH R2, [R0, #0x8] ; 区大小以及是否使能
```

在以下情况下，软件中必须采用存储器隔离指令：

- 假如在修改 MPU 配置时，可能存在某些仍在进行中的存储器传输过程（例如缓冲式写操作），则该操作将可能受到 MPU 配置改变的影响。此时在配置 MPU 之前必须采用存储器隔离指令；
- 假如在配置 MPU 之后的某个存储器传输操作必须运用更新后的 MPU 配置，那么在配置 MPU 之后必须采用存储器隔离指令。

假如进入异常处理程序后立即开始配置 MPU，或配置完 MPU 后立即从异常中返回，那么可以不采用存储器隔离指令，因为异常入口和异常出口已经能够实现相当于存储器隔离指令的保护机制。

在配置 MPU 期间，并不需要软件再增加存储器隔离指令，这是因为访问 MPU 时必须通过私有外设总线（Private Peripheral Bus，简称为 PPB），而 PPB 是严格顺序存储区。

举例来说，假如配置 MPU 之后的存储器访问都需要按照 MPU 的新配置工作，则必须添加一条 DSB 指令和一条 ISB 指令。更改 MPU 配置后应当立即接续一条 DSB 指令，例如上下文切换的结尾。如果对 MPU 存储区编程的代码是通过分支语句或调用语句进入的，那么应当添加一条 ISB 指令。假如编程序列是从异常返回之前或进入异常之后执行的，那么并不需要添加 ISB 指令。

采用多字写操作，更新单个 MPU 区

取决于写入信息如何分割，MPU 也可以直接采用多字写操作进行编程。例如：

```
; R1 = 存储区编号
; R2 = 地址
; R3 = 大小、属性合并
LDR R0, =MPUNUMBER ; 0xE000ED98, MPU区编号寄存器
STR R1, [R0, #0x0] ; 区编号
STR R2, [R0, #0x4] ; 区基地址
STR R3, [R0, #0x8] ; 区属性、大小以及是否使能
```

以上这段程序完全可以利用一条 STM 指令予以优化：

```
; R1 = 存储区编号
; R2 = 地址
; R3 = 大小/是否使能
LDR R0, =MPUNUMBER ; 0xE000ED98, MPU区编号寄存器
STM R0, {R1-R3} ; 区编号、基地址、属性、大小以及是否使能
```

更新单个 MPU 区时，若对配置信息预先整合，也可以只占用两个字的寄存器。这就是说，**MPU 区基地址寄存器（MPUBASE）**（见 [第 181 页](#)）应已包含所需的区编号，并且其 VALID 标志位置位。这种方式比较适合静态配置数据的应用场合，例如对于 Bootloader，就能够进一步精简其大小。

```
; R1 = 地址与区编号合并
; R2 = 大小、属性合并
LDR R0, =MPUBASE ; 0xE000ED9C, MPU区基地址寄存器
STR R1, [R0, #0x0] ; 区基地址及编号合并, 且VALID(第4位)置位
STR R2, [R0, #0x4] ; 区属性、大小以及是否使能
```

如前所述，以上这段程序可以用一条 STM 指令予以优化：

```
; R1 = 地址与区编号合一
; R2 = 大小、属性合一
LDR R0, =MPUBASE ; 0xE000ED9C, MPU区基地址寄存器
STM R0, {R1-R2} ; 区基地址、编号且VALID置位, 属性、大小以及是否使能
```

存储子区

任何不小于 256 字节的存储区还能进一步均分为 8 个存储子区。通过将 **MPU 区属性及大小寄存器（MPUATTR）**（见 [第 183 页](#)）SRD 位域中某个位置位，即可禁用相应的子区。SRD 位域的最低位代表首个子区，最高位代表末个子区。所谓禁用某个子区，就是保障与之重叠的另一个区的设置生效，彻底取代这段被禁用子区的设置。如果禁用某个子区后，也没

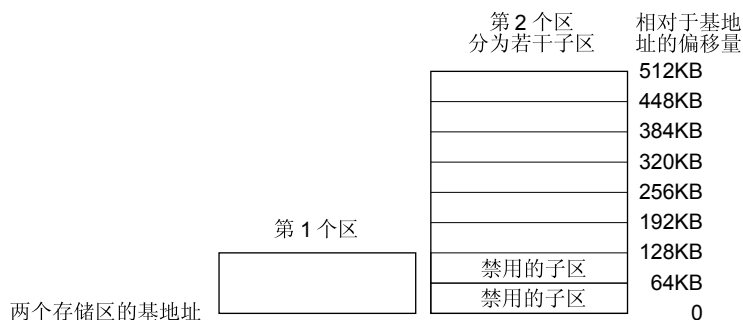
有其它已使能的存储区重叠在该区域上，那么MPU会产生一个故障。

如果存储区的大小为 32、64 或 128 字节，是无法再划分为子区的。因此若存储区的大小是这些数值的话，其 SRD 位域必须配置为 0x00，否则 MPU 的表现将无法预料。

SRD 的用法示例

在下面的例子中，两个存储区具有相同的基地址，因此它们相互重叠。第 1 个区的大小为 128kB，第 2 个区的大小为 512kB。为确保从基地址开始的 128kB 存储器具有第 1 个区所设置的属性，应将第 2 个区的 SRD 位域配置为 0x03，即禁用第 2 个区的前两个子区（共计 128kB），如图 3-1 所示（见第 124 页）。

图 3-1. SRD 的用法示例



3.1.4.2 MPU 访问权限属性

MPUATTR 寄存器中的访问权限位（TEX、S、C、B、AP、XN）负责控制相应存储区的访问权限。假如企图访问存储器中的某个区域，而该区域并未开放相应的权限，则 MPU 会产生一个访问权限故障。

第 124 页的表 3-3 列出了 TEX、C、B、S 访问权限位的编码组合。出于完整方面的考虑，表中列出了所有可能的编码组合，不过目前的 Cortex-M3 内核尚不支持可高速缓冲性以及可共享性。关于对于 Stellaris® 如何编程 MPU，请参阅第 125 页的“Stellaris® 微控制器的 MPU 配置”一节。

表 3-3. TEX、S、C、B 位域的编码

TEX	S	C	B	存储器类型	可共享性	其它属性
000b	x ^a	0	0	严格顺序	可共享	-
000	x ^a	0	1	设备	可共享	-
000	0	1	0	普通	不可共享	外部及内部均为完全写入式。无写分配。
000	1	1	0	普通	可共享	
000	0	1	1	普通	不可共享	
000	1	1	1	普通	可共享	外部及内部均不可高速缓冲。
001	0	0	0	普通	不可共享	
001	1	0	0	普通	可共享	外部及内部均为回写式。写及读均分配。
001	x ^a	0	1	保留	-	
001	x ^a	1	0	保留	-	-
001	0	1	1	普通	不可共享	外部及内部均为回写式。写及读均分配。
001	1	1	1	普通	可共享	

TEX	S	C	B	存储器类型	可共享性	其它属性
010	x ^a	0	0	设备	不可共享	不可共享的外设
010	x ^a	0	1	保留	-	-
010	x ^a	1	x ^a	保留	-	-
1BB	0	A	A	普通	不可共享	可高速缓冲的存储器（BB代表外部策略，A代表内部策略）。关于AA和BB的编码值，参见表3-4。
1BB	1	A	A	普通	可共享	

a: MPU 将忽略此位的值。

第 125 页的表 3-4 列出了当 TEX 位域的值在 0x4~0x7 范围内时，不同存储器属性编码所对应的高速缓冲策略。

表 3-4. 存储器属性编码对应的高速缓冲策略

AA 或 BB 的编码值	对应的高速缓冲策略
00	不可高速缓冲
01	回写式，写及读均分配
10	完全写入式，无写分配
11	回写式，无写分配

第 125 页的表 3-5 列出了 MPUATTR 寄存器 AP 位域的不同编码值为特权级软件和非特权级软件所定义的访问权限。

表 3-5. AP 位域编码值

AP 位域	特权级软件的访问权限	非特权级软件的访问权限	功能说明
000	禁止访问	禁止访问	任何访问操作都会导致产生一个访问权限故障。
001	R/W	禁止访问	仅特权级软件才拥有访问权限。
010	R/W	RO	非特权级软件的写操作会导致产生一个访问权限故障。
011	R/W	R/W	任何软件均可任意访问。
100	无法预料	无法预料	保留
101	RO	禁止访问	仅特权级软件才拥有只读权限。
110	RO	RO	特权级软件或非特权级软件均拥有只读权限。
111	RO	RO	特权级软件或非特权级软件均拥有只读权限。

Stellaris®系列微控制器的 MPU 配置

Stellaris®系列微控制器只有一个处理器内核，并且不带高速缓存。因此，应按第 125 页的表 3-6 中的参数配置 MPU。

表 3-6. Stellaris®系列微控制器的存储区属性配置

存储器区	TEX	S	C	B	存储器类型及属性
Flash 存储器	000b	0	1	0	普通存储器，不可共享，完全写入式

存储器区	TEX	S	C	B	存储器类型及属性
片内 SRAM	000b	1	1	0	普通存储器, 可共享, 完全写入式
片外 SRAM	000b	1	1	1	普通存储器, 可共享, 回写式, 写分配
外设	000b	1	0	1	设备存储器, 可共享

目前在 **Stellaris®** 系列微控制器的构成中, 可共享性和高速缓冲策略这两类属性对系统性能并无任何实际影响。不过, 如果正确配置这些 MPU 区属性, 有益于充分保障代码的可移植性。上表给出的都是常见配置的参考值。

3.1.4.3 MPU 失配

若某个访问操作违反了 MPU 权限, 处理器将会产生一个存储器管理故障 (详见第 94 页的“异常及中断”一节)。通过 **MPFAULTSTAT** 寄存器可查询到产生故障的原因, 详见第 167 页。

3.2 寄存器映射

第 126 页的表 3-7 列出了 Cortex-M3 内核级外设 SysTick、NVIC、SCB 以及 MPU 的相关寄存器。下表中的“偏移量”一列代表该寄存器相对于内核级外设基地址 0xE000 E000 的十六进制地址增量。

注: 未使用的寄存器空间均保留用于将来扩展、或为处理器内部使用, 因此软件切勿修改任何保留地址的内容。

表 3-7. 内核级外设寄存器映射

偏移量	寄存器名称	类型	复位值	寄存器描述	参见页码
系统定时器 (SysTick) 寄存器					
0x010	STCTRL	R/W	0x0000 0004	系统定时器控制及状态寄存器	129
0x014	STRELOAD	R/W	0x0000 0000	系统定时器重载值寄存器	131
0x018	STCURRENT	R/WC	0x0000 0000	系统定时器当前值寄存器	132
嵌套式向量化中断控制器 (NVIC) 寄存器					
0x100	EN0	R/W	0x0000 0000	中断 0~31 设置使能寄存器	133
0x104	EN1	R/W	0x0000 0000	中断 32~54 设置使能寄存器	134
0x180	DIS0	R/W	0x0000 0000	中断 0~31 清除使能寄存器	135
0x184	DIS1	R/W	0x0000 0000	中断 32~54 清除使能寄存器	136
0x200	PEND0	R/W	0x0000 0000	中断 0~31 设置挂起寄存器	137
0x204	PEND1	R/W	0x0000 0000	中断 32~54 设置挂起寄存器	138
0x280	UNPEND0	R/W	0x0000 0000	中断 0~31 清除挂起寄存器	139
0x284	UNPEND1	R/W	0x0000 0000	中断 32~54 清除挂起寄存器	140
0x300	ACTIVE0	RO	0x0000 0000	中断 0~31 激活标志寄存器	141
0x304	ACTIVE1	RO	0x0000 0000	中断 32~54 激活标志寄存器	142
0x400	PRI0	R/W	0x0000 0000	中断 0~3 优先级寄存器	143

偏移量	寄存器名称	类型	复位值	寄存器描述	参见页码
0x404	PRI1	R/W	0x0000 0000	中断 4~7 优先级寄存器	143
0x408	PRI2	R/W	0x0000 0000	中断 8~11 优先级寄存器	143
0x40C	PRI3	R/W	0x0000 0000	中断 12~15 优先级寄存器	143
0x410	PRI4	R/W	0x0000 0000	中断 16~19 优先级寄存器	143
0x414	PRI5	R/W	0x0000 0000	中断 20~23 优先级寄存器	143
0x418	PRI6	R/W	0x0000 0000	中断 24~27 优先级寄存器	143
0x41C	PRI7	R/W	0x0000 0000	中断 28~31 优先级寄存器	143
0x420	PRI8	R/W	0x0000 0000	中断 32~35 优先级寄存器	143
0x424	PRI9	R/W	0x0000 0000	中断 36~39 优先级寄存器	143
0x428	PRI10	R/W	0x0000 0000	中断 40~43 优先级寄存器	143
0x42C	PRI11	R/W	0x0000 0000	中断 44~47 优先级寄存器	143
0x430	PRI12	R/W	0x0000 0000	中断 48~51 优先级寄存器	143
0x434	PRI13	R/W	0x0000 0000	中断 52~53 优先级寄存器	143
0xF00	SWTRIG	WO	0x0000 0000	软件触发中断寄存器	145
系统控制模块 (SCB) 寄存器					
0x008	ACTLR	R/W	0x0000 0000	辅助控制寄存器	146
0xD00	CPUID	RO	0x412F C230	CPU ID 寄存器	148
0xD04	INTCTRL	R/W	0x0000 0000	中断控制及状态寄存器	149
0xD08	VTABLE	R/W	0x0000 0000	向量表偏移量寄存器	153
0xD0C	APINT	R/W	0xFA05 0000	应用程序中断及复位控制寄存器	154
0xD10	SYSCTRL	R/W	0x0000 0000	系统控制寄存器	156
0xD14	CFGCTRL	R/W	0x0000 0200	配置及控制寄存器	158
0xD18	SYSPRI1	R/W	0x0000 0000	系统处理程序优先级寄存器 1	160
0xD1C	SYSPRI2	R/W	0x0000 0000	系统处理程序优先级寄存器 2	161
0xD20	SYSPRI3	R/W	0x0000 0000	系统处理程序优先级寄存器 3	162
0xD24	SYSHNDCTRL	R/W	0x0000 0000	系统处理程序控制及状态寄存器	163
0xD28	FAULTSTAT	R/W1C	0x0000 0000	可配置故障状态寄存器	167
0xD2C	HFAULTSTAT	R/W1C	0x0000 0000	硬故障状态寄存器	173
0xD34	MMADDR	R/W	-	存储器管理故障地址寄存器	175
0xD38	FAULTADDR	R/W	-	总线故障地址寄存器	176
存储器保护单元 (MPU) 寄存器					
0xD90	MPUTYPE	RO	0x0000 0800	MPU 类型寄存器	177
0xD94	MPUCTRL	R/W	0x0000 0000	MPU 控制寄存器	178
0xD98	MPUNUMBER	R/W	0x0000 0000	MPU 区编号寄存器	180
0xD9C	MPUBASE	R/W	0x0000 0000	MPU 区基地址寄存器	181
0xDA0	MPUATTR	R/W	0x0000 0000	MPU 区属性及大小寄存器	183
0xDA4	MPUBASE1	R/W	0x0000 0000	MPU 区基地址别名寄存器 1	181
0xDA8	MPUATTR1	R/W	0x0000 0000	MPU 区属性及大小别名寄存器 1	183
0xDAC	MPUBASE2	R/W	0x0000 0000	MPU 区基地址别名寄存器 2	181

偏移量	寄存器名称	类型	复位值	寄存器描述	参见页码
0xDB0	MPUATTR2	R/W	0x0000 0000	MPU 区属性及大小别名寄存器 2	183
0xDB4	MPUBASE3	R/W	0x0000 0000	MPU 区基地址别名寄存器 3	181
0xDB8	MPUATTR3	R/W	0x0000 0000	MPU 区属性及大小别名寄存器 3	183

3.3 系统定时器 (SysTick) 寄存器描述

本节按照地址偏移量由小到大的顺序依次详细介绍系统定时器的各寄存器。

寄存器 1：系统定时器控制及状态寄存器（STCTRL），偏移量 0x010

注：本寄存器只能在特权模式下访问。

STCTRL 寄存器用于控制系统定时器模块功能的开关。

系统定时器控制及状态寄存器（STCTRL）

基地址 0xE000 E000

偏移量 0x010

类型 R/W，复位值 0x0000 0004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															COUNT
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR													CLK_SRC	INTEN	ENABLE
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

位/位域	名称	类型	复位值	描述
31:17	保留	RO	0x000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
16	COUNT	RO	0	计数标志 取值 描述 0 自上一次读取本标志位之后，系统定时器未曾计数到 0。 1 自上一次读取本标志位之后，系统定时器曾经计数到 0。
15:3	保留	RO	0x000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。

读取本寄存器后，本标志位将自动清零；此外，若对 **STCURRENT** 寄存器写入任意值，本标志位也将自动清零。

如果调试器通过 DAP 读取本寄存器，那么只有当 **AHB-AP 控制寄存器** 的 MasterType 位清零时，本标志位才会自动清零。否则调试器的读操作将不会改变 COUNT 位的状态。关于 MasterType 标志位的详细信息，请参阅《ARM®调试接口 V5 架构规范》。

位/位域	名称	类型	复位值	描述
2	CLK_SRC	R/W	1	<p>时钟源</p> <p>取值 描述</p> <p>0 采用外部参考时钟。(Stellaris®系列微控制器尚未实现此功能)</p> <p>1 采用系统时钟。</p> <p>由于尚未实现以外部参考时钟作为系统定时器的时钟源，因此本标志位必须置位。</p>
1	INTEN	R/W	0	<p>中断使能</p> <p>取值 描述</p> <p>0 禁止产生中断。此时软件可通过 COUNT 标志位来判断计数器是否曾经计数到 0。</p> <p>1 当系统定时器计数到 0 时，向 NVIC 产生一个中断。</p>
0	ENABLE	R/W	0	<p>计数器使能</p> <p>取值 描述</p> <p>0 禁用计数器。</p> <p>1 使能计数器。此后系统定时器将以自动重载的方式开始工作。详细来说就是：计数器首先载入 RELOAD 中的重载值，并开始递减计数。当计数到 0 时，计数器将 COUNT 标志位置位；如果 INTEN 置位，还会产生一个中断。之后计数器再次载入 RELOAD 中的重载值并继续递减计数。如是周而复始、不停工作、。</p>

寄存器 2: 系统定时器重载值寄存器 (STRELOAD), 偏移量 0x014

注: 本寄存器只能在特权模式下访问。

STRELOAD 寄存器用于设置自动重载的计数值。每当计数器计数到 0 后, 会自动将本寄存器的值载入到**系统定时器当前值寄存器 (STCURRENT)**中。起始值的有效范围是 0x0000 0001~0x00FF FFFF。虽然取值为 0 也是合法的, 不过由于系统定时器只在发生从 1 到 0 的跳变时才将 COUNT 位置位并产生中断, 因此起始值为 0 将没有任何实际效果。

系统定时器按照自动重载方式工作, 每 N+1 个时钟动作一次 (N 的取值范围为 0x0000 0001~0x00FF FFFF), 如是周而复始、永不停歇。举例来说, 如果需要每 100 个时钟周期产生一次系统定时中断, 则应向 RELOAD 位域写入 99。

系统定时器重载值寄存器 (STRELOAD)

基地址 0xE000 E000

偏移量 0x014

类型 R/W, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								RELOAD							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RELOAD															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:24	保留	RO	0x00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
23:0	RELOAD	R/W	0x00 0000	重载值 当计数器计数到 0 时, 本位域的内容将自动载入到 系统定时器当前值寄存器 (STCURRENT) 中。

寄存器 3: 系统定时器当前值寄存器 (STCURRENT), 偏移量 0x018

注: 本寄存器只能在特权模式下访问。

STCURRENT 寄存器中包含系统定时器的计数器当前值。

系统定时器当前值寄存器 (STCURRENT)

基地址 0xE000 E000

偏移量 0x018

类型 R/WC, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								CURRENT							
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CURRENT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:24	保留	RO	0x00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
23:0	CURRENT	R/WC	0x00 0000	当前值 此位域包含本寄存器被访问时刻的计数器值。此寄存器本身并不具有读-修改-写的保护机制, 因此应慎重考虑读出值的可能变化。 本寄存器具有写入清零属性, 亦即对本寄存器写入任何值都会将其清零。与此同时 STCTRL 寄存器的 COUNT 标志位也会一同清零。

3.4 NVIC 寄存器描述

本节按照地址偏移量由小到大的顺序依次详细介绍 NVIC 的各寄存器。

NVIC 寄存器在特权模式下全部可以访问的。在非特权模式下, 只允许通过**配置及控制寄存器 (CFGCTRL)**屏蔽中断; 对其它 NVIC 寄存器的访问都会导致产生总线故障。

软件访问寄存器时应确保其对齐正确。处理器不支持对 NVIC 寄存器执行非对齐访问。

即使某个中断已被禁用, 该中断也能够进入挂起状态。

在对**VTABLE**寄存器编程(重定位向量表)之前, 应确保新向量表中已经包含正确的故障处理程序入口、NMI处理程序入口以及所有已经使能的异常(例如中断)入口。详见[第153页](#)。

寄存器 4: 中断 0~31 设置使能寄存器 (EN0), 偏移量 0x100

注: 本寄存器只能在特权模式下访问。

EN0 寄存器用于使能中断, 此外也可以返回各中断的使能状态。第 0 位代表中断 0, 第 31 位代表中断 31, 依此类推。关于中断的分配可参见第 106 页的表 2-9。

假如某个已挂起的中断被使能, **NVIC** 将按照其优先级激活中断。假如某个中断未使能, 那么当中断源产生中断信号后, 虽然该中断状态变为挂起, 但 **NVIC** 永远不会按照其优先级将其激活。

中断 0~31 设置使能寄存器 (EN0)

基地址 0xE000 E000

偏移量 0x100

类型 R/W, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:0	INT	R/W	0x0000 0000	中断使能

取值	描述
0	读出为 0, 代表该中断已被禁用。 写入 0 时, 无意义。
1	读出为 1, 代表该中断已经使能。 写入 1 时, 将使能对应中断。

要想将本寄存器中的某个位清零, 需将 **DIS0** 寄存器的相应 INT[n] 位置位。

寄存器 5: 中断 32~54 设置使能寄存器 (EN1), 偏移量 0x104

注: 本寄存器只能在特权模式下访问。

EN1 寄存器用于使能中断, 此外也可以返回各中断的使能状态。第 0 位代表中断 32, 第 22 位代表中断 54, 依此类推。依此类推。关于中断的分配可参见第 106 页的表 2-9。

假如某个已挂起的中断被使能, **NVIC** 将按照其优先级激活中断。假如某个中断未使能, 那么当中断源产生中断信号后, 虽然该中断状态变为挂起, 但 **NVIC** 永远不会按照其优先级将其激活。

中断 32~54 设置使能寄存器 (EN1)

基地址 0xE000 E000

偏移量 0x104

类型 R/W, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								INT							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:23	保留	RO	0x00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
22:0	INT	R/W	0x00 0000	中断使能 取值 描述 0 读出为 0, 代表该中断已被禁用。 写入 0 时, 无意义。 1 读出为 1, 代表该中断已经使能。 写入 1 时, 将使能对应中断。

要想将本寄存器中的某个位清零, 需将 **DIS1** 寄存器的相应 INT[n] 位置位。

寄存器 6: 中断 0~31 清除使能寄存器 (DIS0), 偏移量 0x180

注: 本寄存器只能在特权模式下访问。

DIS0 寄存器用于禁用中断。第 0 位代表中断 0, 第 31 位代表中断 31, 依此类推。关于中断的分配可参见第 106 页的表 2-9。

中断 0~31 清除使能寄存器 (DIS0)

基地址 0xE000 E000

偏移量 0x180

类型 R/W, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:0	INT	R/W	0x0000 0000	中断禁用 取值 描述 0 读数为 0, 代表该中断已被禁用。 写入 0 时, 无意义。 1 读数为 1, 代表该中断已经使能。 写入 1 时, 将禁用对应中断[n], 并将 ENO 寄存器中的 INT[n]位清零。

寄存器 7：中断 32~54 清除使能寄存器 (DIS1)，偏移量 0x184

注：本寄存器只能在特权模式下访问。

DIS1 寄存器用于禁用中断。第 0 位代表中断 32，第 22 位代表中断 54，依此类推。关于中断的分配可参见第 106 页的表 2-9。

中断 32~54 清除使能寄存器 (DIS1)

基地址 0xE000 E000

偏移量 0x184

类型 R/W，复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留									INT						
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述						
31:23	保留	RO	0x00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。						
22:0	INT	R/W	0x00 0000	中断禁用 <table border="0"> <tr> <td>取值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>读出为 0，代表该中断已被禁用。 写入 0 时，无意义。</td> </tr> <tr> <td>1</td> <td>读出为 1，代表该中断已经使能。 写入 1 时，将禁用对应中断[n]，并将 EN1 寄存器中的 INT[n]位清零。</td> </tr> </table>	取值	描述	0	读出为 0，代表该中断已被禁用。 写入 0 时，无意义。	1	读出为 1，代表该中断已经使能。 写入 1 时，将禁用对应中断[n]，并将 EN1 寄存器中的 INT[n]位清零。
取值	描述									
0	读出为 0，代表该中断已被禁用。 写入 0 时，无意义。									
1	读出为 1，代表该中断已经使能。 写入 1 时，将禁用对应中断[n]，并将 EN1 寄存器中的 INT[n]位清零。									

寄存器 8: 中断 0~31 设置挂起寄存器 (PEND0), 偏移量 0x200

注: 本寄存器只能在特权模式下访问。

PEND0 寄存器用于强制将中断置为挂起状态, 此外也可以返回各中断的挂起状态。第 0 位代表中断 0, 第 31 位代表中断 31, 依此类推。关于中断的分配可参见第 106 页的表 2-9。

中断 0~31 设置挂起寄存器 (PEND0)

基地址 0xE000 E000

偏移量 0x200

类型 R/W, 复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:0	INT	R/W	0x0000 0000	中断设置挂起
				取值 描述
				0 读出为 0, 代表该中断未挂起。 写入 0 时, 无意义。
				1 读出为 1, 代表该中断已挂起。 写入 1 时, 对应中断的状态将变为挂起, 即使该中断已被禁用也是如此。

假如某个中断已经挂起, 那么对该位写 1 将没有实际意义。

要想将本寄存器中的某个位清零, 需将 **UNPEND0** 寄存器的相应 INT[n] 位置位。

寄存器 9: 中断 32~54 设置挂起寄存器 (PEND1), 偏移量 0x204

注: 本寄存器只能在特权模式下访问。

PEND1 寄存器用于强制将中断置为挂起状态, 此外也可以返回各中断的挂起状态。第 0 位代表中断 32, 第 22 位代表中断 54, 依此类推。关于中断的分配可参见第 106 页的表 2-9。

中断 32~54 设置挂起寄存器 (PEND1)

基地址 0xE000 E000

偏移量 0x204

类型 R/W, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留									INT						
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述						
31:23	保留	RO	0x00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。						
22:0	INT	R/W	0x00 0000	中断设置挂起 <table border="0"> <tr> <td>取值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>读出为 0, 代表该中断未挂起。 写入 0 时, 无意义。</td> </tr> <tr> <td>1</td> <td>读出为 1, 代表该中断已挂起。 写入 1 时, 对应中断的状态将变为挂起, 即使该中断已被禁用也是如此。</td> </tr> </table>	取值	描述	0	读出为 0, 代表该中断未挂起。 写入 0 时, 无意义。	1	读出为 1, 代表该中断已挂起。 写入 1 时, 对应中断的状态将变为挂起, 即使该中断已被禁用也是如此。
取值	描述									
0	读出为 0, 代表该中断未挂起。 写入 0 时, 无意义。									
1	读出为 1, 代表该中断已挂起。 写入 1 时, 对应中断的状态将变为挂起, 即使该中断已被禁用也是如此。									

假如某个中断已经挂起, 那么对该位写 1 将没有实际意义。

要想将本寄存器中的某个位清零, 需将 **UNPEND1** 寄存器的相应 **INT[n]** 位置位。

寄存器 10: 中断 0~31 清除挂起寄存器 (UNPEND0), 偏移量 0x280

注: 本寄存器只能在特权模式下访问。

UNPEND0 寄存器用于清除中断的挂起状态, 此外也可以返回各中断的挂起状态。第 0 位代表中断 0, 第 31 位代表中断 31, 依此类推。关于中断的分配可参见第 106 页的表 2-9。

中断 0~31 清除挂起寄存器 (UNPEND0)

基地址 0xE000 E000

偏移量 0x280

类型 R/W, 复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:0	INT	R/W	0x0000 0000	中断清除挂起
				取值 描述
				0 读数为 0, 代表该中断未挂起。 写入 0 时, 无意义。
				1 读数为 1, 代表该中断已挂起。 写入 1 时, 对应中断[n]的状态将不再挂起, 并且 PEND0 寄存器的对应 INT[n]位清零。 如果某个中断已激活, 那么写 1 将失去实际意义。

寄存器 11：中断 32~54 清除挂起寄存器 (UNPEND1)，偏移量 0x284

注：本寄存器只能在特权模式下访问。

UNPEND1 寄存器用于清除中断的挂起状态，此外也可以返回各中断的挂起状态。第 0 位代表中断 32，第 22 位代表中断 54，依此类推。关于中断的分配可参见第 106 页的表 2-9。

中断 32~54 清除挂起寄存器 (UNPEND1)

基地址 0xE000 E000

偏移量 0x284

类型 R/W，复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留									INT						
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:23	保留	RO	0x00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
22:0	INT	R/W	0x00 0000	中断清除挂起 取值 描述 0 读出为 0，代表该中断未挂起。 写入 0 时，无意义。 1 读出为 1，代表该中断已挂起。 写入 1 时，对应中断[n]的状态将不再挂起，并且 PEND1 寄存器的对应 INT[n] 位清零。 如果某个中断已激活，那么写 1 将失去实际意义。

寄存器 12: 中断 0~31 激活标志寄存器 (ACTIVE0), 偏移量 0x300

注: 本寄存器只能在特权模式下访问。

ACTIVE0 寄存器用于指示哪些中断处于激活状态。第 0 位代表中断 0, 第 31 位代表中断 31, 依此类推。关于中断的分配可参见第 106 页的表 2-9。

重要提示: 切勿手动修改本寄存器的内容!

中断 0~31 激活标志寄存器 (ACTIVE0)

基地址 0xE000 E000

偏移量 0x300

类型 RO, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INT															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:0	INT	RO	0x0000 0000	中断激活状态
				取值 描述
				0 对应中断未激活。
				1 对应中断已激活, 或处于激活并等待状态。

寄存器 13: 中断 32~54 激活标志寄存器 (ACTIVE1), 偏移量 0x304

注: 本寄存器只能在特权模式下访问。

ACTIVE1 寄存器用于指示哪些中断处于激活状态。第 0 位代表中断 32, 第 22 位代表中断 54, 依此类推。关于中断的分配可参见第 106 页的表 2-9。

重要提示: 切勿手动修改本寄存器的内容!

中断 32~54 激活标志寄存器 (ACTIVE1)

基地址 0xE000 E000

偏移量 0x304

类型 RO, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								INT							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:23	保留	RO	0x00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
22:0	INT	R/W	0x00 0000	中断激活状态 取值 描述 0 对应中断未激活。 1 对应中断已激活, 或处于激活并等待状态。

- 寄存器 14: 中断 0~3 优先级寄存器 (PRI0), 偏移量 0x400
- 寄存器 15: 中断 4~7 优先级寄存器 (PRI1), 偏移量 0x404
- 寄存器 16: 中断 8~11 优先级寄存器 (PRI2), 偏移量 0x408
- 寄存器 17: 中断 12~15 优先级寄存器 (PRI3), 偏移量 0x40C
- 寄存器 18: 中断 16~19 优先级寄存器 (PRI4), 偏移量 0x410
- 寄存器 19: 中断 20~23 优先级寄存器 (PRI5), 偏移量 0x414
- 寄存器 20: 中断 24~27 优先级寄存器 (PRI6), 偏移量 0x418
- 寄存器 21: 中断 28~31 优先级寄存器 (PRI7), 偏移量 0x41C
- 寄存器 22: 中断 32~35 优先级寄存器 (PRI8), 偏移量 0x420
- 寄存器 23: 中断 36~39 优先级寄存器 (PRI9), 偏移量 0x424
- 寄存器 24: 中断 40~43 优先级寄存器 (PRI10), 偏移量 0x428
- 寄存器 25: 中断 44~47 优先级寄存器 (PRI11), 偏移量 0x42C
- 寄存器 26: 中断 48~51 优先级寄存器 (PRI12), 偏移量 0x430
- 寄存器 27: 中断 52~53 优先级寄存器 (PRI13), 偏移量 0x434

注: 以上寄存器只能在特权模式下访问。

PRIn 寄存器为每个中断提供 3 位的优先级位域。所有 **PRIn** 寄存器都是可字节访问的。每个寄存器中包含 4 个中断的优先级位域, 如下所示:

PRIn 寄存器位域	中断
[31:29]位	中断[4n+3]
[23:21]位	中断[4n+2]
[15:13]位	中断[4n+1]
[7:5]位	中断[4n]

关于中断的分配可参见第 106 页的表 2-9。

优先级位域还可以进一步划分为组优先级位域以及子优先级位域。应用程序中断及复位控制寄存器 (APINT) (见第 154 页) 的 PRIGROUP 位域可定义本寄存器 INTx 位域中二进制小数点的位置, 依此划分组优先级位域和子优先级位域。

Cortex-M3 内核级外设

中断 0~3 优先级寄存器 (PRIO)

基地址 0xE000 E000

偏移量 0x400

类型 R/W, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTD			保留				INTC			保留					
类型	R/W	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTB			保留				INTA			保留					
类型	R/W	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:29	INTD	R/W	0x0	中断[4n+3]的中断优先级 此位域包含序号为[4n+3]的中断的优先级，其中 n 是 中断优先级寄存器 的序号（即 PRIO 寄存器的 n=0， PR1 寄存器的 n=1，依此类推）。优先级为 3 位，取值范围 0~7，数值越小则优先级越高。
29:24	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
23:21	INTC	R/W	0x0	中断[4n+2]的中断优先级 此位域包含序号为[4n+2]的中断的优先级，其中 n 是 中断优先级寄存器 的序号（即 PRIO 寄存器的 n=0， PR1 寄存器的 n=1，依此类推）。优先级为 3 位，取值范围 0~7，数值越小则优先级越高。
20:16	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
15:13	INTB	R/W	0x0	中断[4n+1]的中断优先级 此位域包含序号为[4n+1]的中断的优先级，其中 n 是 中断优先级寄存器 的序号（即 PRIO 寄存器的 n=0， PR1 寄存器的 n=1，依此类推）。优先级为 3 位，取值范围 0~7，数值越小则优先级越高。
12:8	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
7:5	INTA	R/W	0x0	中断[4n]的中断优先级 此位域包含序号为[4n]的中断的优先级，其中 n 是 中断优先级寄存器 的序号（即 PRIO 寄存器的 n=0， PR1 寄存器的 n=1，依此类推）。优先级为 3 位，取值范围 0~7，数值越小则优先级越高。
4:0	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。

寄存器 28：软件触发中断寄存器（SWTRIG），偏移量 0xF00

注：本寄存器只能在特权模式下访问。

向 **SWTRIG** 寄存器写入某个中断的序号，就会产生软件中断（Software Generated Interrupt，简称为 SGI）。关于中断的分配可参见第 106 页的表 2-9。

若配置及控制寄存器（CFGCTRL）（见第 158 页）的MAINPEND位置位，则非特权级软件也可以访问**SWTRIG**寄存器。

软件触发中断寄存器（SWTRIG）

基地址 0xE000 E000

偏移量 0xF00

类型 WO，复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留										INTID					
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	WO	WO	WO	WO	WO	WO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:6	保留	RO	0x0000 00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
5:0	INTID	WO	0x00	中断 ID 此位域包含请求的 SGI 中断的 ID。例如，此位域写入 0x3 时，将会触发 IRQ3 产生中断。

3.5 系统控制模块（SCB）寄存器描述

本节按照地址偏移量由小到大的顺序依次详细介绍系统控制模块（SCB）的各寄存器。SCB 寄存器只能在特权模式下访问。

除了 **FAULTSTAT** 和 **SYSPRI1~SYSPRI3** 寄存器能够按字节对齐、按半字对齐或按字对齐进行访问外，其余寄存器必须按照字对齐格式进行访问。处理器不支持对 SCB 寄存器执行非对齐访问。

寄存器 29: 辅助控制寄存器 (ACTLR), 偏移量 0x008

注: 本寄存器只能在特权模式下访问。

ACTLR 寄存器为禁用以下功能提供控制位: IT 堆叠; 访问默认的存储器映射时的写缓冲; 对多周期指令的中断。本寄存器的默认配置已经为 Cortex-M3 处理器内核实现了优化的性能, 一般来说无需修改。

辅助控制寄存器 (ACTLR)

基地址 0xE000 E000

偏移量 0x008

类型 R/W, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													DISFOLD	DISWBUF	DISMCYC
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:3	保留	RO	0x0000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
2	DISFOLD	R/W	0	禁用 IT 堆叠 取值 描述 0 无影响。 1 禁用 IT 堆叠。
1	DISWBUF	R/W	0	禁用写入缓冲 取值 描述 0 无影响。 1 在访问默认的存储空间时禁用写入缓冲。在此情况下, 一旦产生总线故障, 则必然是精确总线故障; 但处理器性能将会有所下降, 因为处理器要等待对内存的存储操作完全结束后才会执行下一条指令。 注: 此标志位只影响 Cortex-M3 处理器内核中的写入缓冲, 与外设写入缓冲没有任何关系。

位/位域	名称	类型	复位值	描述
0	DISMCYC	R/W	0	禁用多周期指令的中断
				取值 描述
				0 无影响。
				1 禁止在多周期装载指令以及多周期存贮指令执行期间产生中断。若本标志位置位，由于处理器必须等待 LDM 或 STM 指令执行完成后才将状态入栈并进入中断处理程序，因此处理器的中断延时将会增加。

寄存器 30: CPU ID 寄存器 (CPUID), 偏移量 0xD00

注: 本寄存器只能在特权模式下访问。

CPUID 寄存器中包含 ARM® Cortex™-M3 处理器的芯片编号、版本以及其它构成信息。

CPU ID 寄存器 (CPUID)

基地址 0xE000 E000

偏移量 0xD00

类型 RO, 复位值 0x412F C230

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IMP							VAR					CON			
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	1	0	0	0	0	0	1	0	0	1	0	1	1	1	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PARTNO											REV				
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	1	1	0	0	0	0	1	0	0	0	1	1	0	0	0	0

位/位域	名称	类型	复位值	描述
31:24	IMP	RO	0x41	标准制订商代码 取值 描述 0x41 ARM
23:20	VAR	RO	0x2	修订版本代码 取值 描述 0x2 此位域的值即为内核版本 rmpn 中的 m 值。例如, 对于 r2p0 版内核, 此位域的值为 2。
19:16	CON	RO	0xF	常量 取值 描述 0xF 此位域始终为 0x0F。
15:4	PARTNO	RO	0xC23	芯片型号 取值 描述 0xC23 表示 Cortex-M3 处理器。
3:0	REV	RO	0x0	修订版本代码 取值 描述 0x0 此位域的值即为内核版本 rmpn 中的 n 值。例如, 对于 r2p0 版内核, 此位域的值为 0。

寄存器 31：中断控制及状态寄存器 (INTCTRL)，偏移量 0xD04

注：本寄存器只能在特权模式下访问。

INTCTRL 寄存器为 NMI 异常提供了设置挂起位，为 PendSV 以及 SysTick 异常提供了设置挂起位以及清除挂起位。此外，本寄存器还能指示出正在处理的异常序号、是否存在抢占式激活的异常、挂起异常中优先级最高的异常序号，以及是否有中断正在挂起。

若对 INTCTRL 寄存器执行写操作，并且 PENDSV 和 UNPENDSV 同时写 1 或 PENDSTSET 和 PENDSTCLR 同时写 1，则可能产生无法预料的后果。

中断控制及状态寄存器 (INTCTRL)

基地址 0xE000 E000

偏移量 0xD04

类型 R/W，复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NMISSET	保留		PENDSV	UNPENDSV	PENDSTSET	PENDSTCLR	保留	ISRPRE	ISRPEND	保留			VECPEND		
类型	R/W	RO	RO	R/W	WO	R/W	WO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VECPEND			RETBASE		保留				VECACT						
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31	NMISSET	R/W	0	NMI 设置挂起 取值 描述 0 读数为 0，代表 NMI 异常并未挂起。 写入 0 时，无实际意义。 1 读数为 1，代表 NMI 异常正在挂起。 写入 1 时，NMI 异常的状态将变为挂起。

由于 NMI 是最高优先级的异常，因此一般来说，处理器只要检测到此标志置位就会立即进入 NMI 异常处理程序，并在进入中断处理程序后即刻清除此标志位。只有当处理器执行 NMI 异常处理程序期间又再度产生了 NMI 信号，才可能在异常处理程序中读取此标志位为 1。

30:29	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
-------	----	----	-----	---

位/位域	名称	类型	复位值	描述
28	PENDSV	R/W	0	<p>PendSV 设置挂起</p> <p>取值 描述</p> <p>0 读出为 0, 代表 PendSV 异常并未挂起。 写入 0 时, 无实际意义。</p> <p>1 读出为 1, 代表 PendSV 异常正在挂起。 写入 1 时, PendSV 异常的状态将变为挂起。</p> <p>要想将 PendSV 异常的状态变为挂起, 只能通过对本标志位写 1 来实现。对 UNPENDSV 位写 1 即可将本标志位清零。</p>
27	UNPENDSV	WO	0	<p>PendSV 清除挂起</p> <p>取值 描述</p> <p>0 写入 0 时, 无实际意义。</p> <p>1 写入 1 时, 将清除 PendSV 异常的挂起状态。</p> <p>此标志位为只写。读本寄存器时本标志位的状态不定。</p>
26	PENDSTSET	R/W	0	<p>SysTick 设置挂起</p> <p>取值 描述</p> <p>0 读出为 0, 代表 SysTick 异常并未挂起。 写入 0 时, 无实际意义。</p> <p>1 读出为 1, 代表 SysTick 异常正在挂起。 写入 1 时, SysTick 异常的状态将变为挂起。</p> <p>对 PENDSTCLR 位写 1 即可将本标志位清零。</p>
25	PENDSTCLR	WO	0	<p>SysTick 清除挂起</p> <p>取值 描述</p> <p>0 写入 0 时, 无实际意义。</p> <p>1 写入 1 时, 将清除 SysTick 异常的挂起状态。</p> <p>此标志位为只写。读本寄存器时本标志位的状态不定。</p>
24	保留	RO	0	<p>软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。</p>
23	ISRPRE	RO	0	<p>调试中断处理</p> <p>取值 描述</p> <p>0 处理器从暂停状态恢复运行并非基于中断。</p> <p>1 处理器从暂停状态恢复运行是基于中断。</p> <p>此标志位为只写。读本寄存器时本标志位的状态不定。</p>

位/位域	名称	类型	复位值	描述																																						
22	ISRPEND	RO	0	<p>中断挂起</p> <table border="1"> <thead> <tr> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>无中断正在挂起。</td> </tr> <tr> <td>1</td> <td>有中断正在挂起。</td> </tr> </tbody> </table> <p>此标志位反映除 NMI 及故障之外的所有中断状态。</p>	取值	描述	0	无中断正在挂起。	1	有中断正在挂起。																																
取值	描述																																									
0	无中断正在挂起。																																									
1	有中断正在挂起。																																									
21:19	保留	RO	0x0	<p>软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。</p>																																						
18:12	VECPEND	RO	0x00	<p>中断挂起向量序号</p> <p>此位域包含已使能、已挂起且优先级最高的异常序号。此位域的内容已经包含了 BASEPRI 及 FAULTMASK 寄存器的影响，但并不包含 PRIMASK 寄存器的影响。</p> <table border="1"> <thead> <tr> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>无正在挂起的异常</td> </tr> <tr> <td>0x01</td> <td>保留</td> </tr> <tr> <td>0x02</td> <td>NMI</td> </tr> <tr> <td>0x03</td> <td>硬故障</td> </tr> <tr> <td>0x04</td> <td>存储器管理故障</td> </tr> <tr> <td>0x05</td> <td>总线故障</td> </tr> <tr> <td>0x06</td> <td>用法故障</td> </tr> <tr> <td>0x07~0x0A</td> <td>保留</td> </tr> <tr> <td>0x0B</td> <td>SVCALL</td> </tr> <tr> <td>0x0C</td> <td>保留用于调试</td> </tr> <tr> <td>0x0D</td> <td>保留</td> </tr> <tr> <td>0x0E</td> <td>PendSV</td> </tr> <tr> <td>0x0F</td> <td>SysTick</td> </tr> <tr> <td>0x10</td> <td>中断向量 0</td> </tr> <tr> <td>0x11</td> <td>中断向量 1</td> </tr> <tr> <td>.....</td> <td>.....</td> </tr> <tr> <td>0x46</td> <td>中断向量 54</td> </tr> <tr> <td>0x47~0x7F</td> <td>保留</td> </tr> </tbody> </table>	取值	描述	0x00	无正在挂起的异常	0x01	保留	0x02	NMI	0x03	硬故障	0x04	存储器管理故障	0x05	总线故障	0x06	用法故障	0x07~0x0A	保留	0x0B	SVCALL	0x0C	保留用于调试	0x0D	保留	0x0E	PendSV	0x0F	SysTick	0x10	中断向量 0	0x11	中断向量 1	0x46	中断向量 54	0x47~0x7F	保留
取值	描述																																									
0x00	无正在挂起的异常																																									
0x01	保留																																									
0x02	NMI																																									
0x03	硬故障																																									
0x04	存储器管理故障																																									
0x05	总线故障																																									
0x06	用法故障																																									
0x07~0x0A	保留																																									
0x0B	SVCALL																																									
0x0C	保留用于调试																																									
0x0D	保留																																									
0x0E	PendSV																																									
0x0F	SysTick																																									
0x10	中断向量 0																																									
0x11	中断向量 1																																									
.....																																									
0x46	中断向量 54																																									
0x47~0x7F	保留																																									
11	RETBASE	RO	0	<p>返回基地址</p> <table border="1"> <thead> <tr> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>有其它抢占式激活异常等待处理。</td> </tr> <tr> <td>1</td> <td>当前无激活的异常，或当前正在处理的异常是唯一激活的异常。</td> </tr> </tbody> </table> <p>此标志位反映除 NMI 及故障之外的所有中断状态。只有当处理器正在执行某个 ISR 时（即中断程序状态寄存器 IPSR 非 0），本标志位才有意义。</p>	取值	描述	0	有其它抢占式激活异常等待处理。	1	当前无激活的异常，或当前正在处理的异常是唯一激活的异常。																																
取值	描述																																									
0	有其它抢占式激活异常等待处理。																																									
1	当前无激活的异常，或当前正在处理的异常是唯一激活的异常。																																									

位/位域	名称	类型	复位值	描述
10:7	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
6:0	VECACT	RO	0x00	<p>挂起中断的向量序号</p> <p>此位域包含当前激活的异常的序号。各异常的序号参见 VECPEND 位域的介绍。假如此位域为 0，则处理器将进入线程模式。此位域的值与 IPSR 寄存器的 ISRNUM 位域相同。</p> <p>此位域的值减去 16 即可得到 IRQ 号，可用于索引中断置位使能寄存器 (ENn)、中断清除使能寄存器 (DISn)、中断置位挂起寄存器 (PENDn)、中断清除挂起寄存器 (UNPENDn)、中断优先级寄存器 (PRIn) (见第 86 页)。</p>

寄存器 32: 向量表偏移量寄存器 (VTABLE), 偏移量 0xD08

注: 本寄存器只能在特权模式下访问。

VTABLE 寄存器中包含向量表基地址相对于存储器地址 0x0000 0000 的偏移量。

向量表偏移量寄存器 (VTABLE)

基地址 0xE000 E000

偏移量 0xD08

类型 R/W, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留		BASE	OFFSET												
类型	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OFFSET							保留								
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:30	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
29	BASE	R/W	0	向量表基地址 取值 描述 0 向量表位于代码存储器区。 1 向量表位于 SRAM 存储器区。
28:9	OFFSET	R/W	0x000 00	向量表偏移量 配置 OFFSET 位域时, 偏移量必须按照向量表中的异常入口数目对齐。由于总共有 54 个中断, 因此最小对齐单位为 128 字节。
8:0	保留	RO	0x00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。

寄存器 33：应用程序中断及复位控制寄存器（APINT），偏移量 0xD0C

注：本寄存器只能在特权模式下访问。

APINT 寄存器为异常模型提供分组优先级控制，提供数据访问的端状态，并提供系统的复位控制。要想对本寄存器执行写操作，必须对 VECTKEY 位域写入 0x05FA，否则写操作将被忽略。

PRIGROUP位域用于设置二进制小数点的位置，该小数点将**中断优先级寄存器（IPx）**的 INTx位域划分为组优先级位域以及子优先级位域。第 154 页的表 3-8列出了PRIGROUP位域与划分的关系。表中“组优先级位域”以及“子优先级位域”两列的内容都是指针对于INTA位域中的位偏移。对于INTB位域来说，相应的位域是[15:13]；对于INTC位域来说，相应的位域是[23:21]；对于INTD位域来说，相应的位域是[31:29]。

注：在判定异常的抢占优先级时，只考虑组优先级位域。

表 3-8. 中断优先级划分

PRIGROUP 位域	二进制小数点 ^a	组优先级位域	子优先级位域	组优先级数目	子优先级数目
0x0~0x4	bxxx.	[7:5]	无	8	1
0x5	bxx.y	[7:6]	[5]	4	2
0x6	bx.yy	[7]	[6:5]	2	4
0x7	b.yyy	无	[7:5]	1	8

a: 从 INTx 位域可查询二进制小数点的位置。表中 x 代表分组优先级位，y 代表子优先级位。

应用程序中断及复位控制寄存器（APINT）

基地址 0xE000 E000

偏移量 0xD0C

类型 R/W，复位值 0xFA05 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VECTKEY															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	0	1	0	0	0	0	0	0	1	0	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENDIANESS	保留				PRIGROUP			保留				SYSRESETREQ	VECTCLRRACT	VECTRESET	
类型	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO	WO	WO	WO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:16	VECTKEY	R/W	0xFA05	注册密钥 此位域用于保护本寄存器，防止发生意外写入的状况。要想更改本寄存器的内容，在写入时必须对本位域填写 0x05FA。在读取本寄存器时，本位域返回 0xFA05。
15	ENDIANESS	RO	0	数据端模式 Stellaris®系列微控制器仅支持小端模式，因此本标志位回读时始终为 0。

位/位域	名称	类型	复位值	描述
14:11	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
10:8	PRIGROUP	R/W	0x0	中断优先级分组 本位域确定分组优先级与子优先级的分隔界限。详见第154页的表3-8。
7:3	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
2	SYSRESETREQ	WO	0	系统复位请求 取值 描述 0 无意义。 1 复位处理器内核以及所有片上外设（调试接口除外）。 此标志位在处理器内核复位期间自动清零。 此标志位的回读值始终为0。
1	VECTCLRACT	WO	0	清除已激活的NMI/故障 此标志位保留用于调试，其回读值始终为0。此标志位必须写为0，否则可能产生无法预料的后果。
0	VECTRESET	WO	0	系统复位 此标志位保留用于调试，其回读值始终为0。此标志位必须写为0，否则可能产生无法预料的后果。

寄存器 34：系统控制寄存器 (SYSCTRL)，偏移量 0xD10

注：本寄存器只能在特权模式下访问。

SYSCTRL 寄存器负责控制进入/退出低功耗模式。

系统控制寄存器 (SYSCTRL)

基地址 0xE000 E000

偏移量 0xD10

类型 R/W，复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留											SEVONPEND	保留	SLEEPDEEP	SLEEPEXIT	保留
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	RO	R/W	R/W	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述						
31:5	保留	RO	0x0000 00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。						
4	SEVONPEND	R/W	0	挂起时唤醒 <table border="0"> <tr> <td>取值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>只有已使能的中断或事件才能唤醒处理器；如果某中断已禁用，则不能唤醒处理器。</td> </tr> <tr> <td>1</td> <td>已使能的事件以及所有中断（包括已禁用的中断）都能唤醒处理器。</td> </tr> </table> <p>当一个事件或中断进入挂起状态后，事件信号能够将处理器从 WFE 指令中唤醒。假如处理器并未等待事件，那么该事件仍会被记录下来，并将影响下一次 WFE 的运行。</p> <p>处理器也可以在执行 SEV 指令时唤醒，或通过外部事件唤醒。</p>	取值	描述	0	只有已使能的中断或事件才能唤醒处理器；如果某中断已禁用，则不能唤醒处理器。	1	已使能的事件以及所有中断（包括已禁用的中断）都能唤醒处理器。
取值	描述									
0	只有已使能的中断或事件才能唤醒处理器；如果某中断已禁用，则不能唤醒处理器。									
1	已使能的事件以及所有中断（包括已禁用的中断）都能唤醒处理器。									
3	保留	RO	0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。						
2	SLEEPDEEP	R/W	0	深度睡眠使能 <table border="0"> <tr> <td>取值</td> <td>描述</td> </tr> <tr> <td>0</td> <td>以睡眠模式作为低功耗模式。</td> </tr> <tr> <td>1</td> <td>以深度睡眠模式作为低功耗模式。</td> </tr> </table>	取值	描述	0	以睡眠模式作为低功耗模式。	1	以深度睡眠模式作为低功耗模式。
取值	描述									
0	以睡眠模式作为低功耗模式。									
1	以深度睡眠模式作为低功耗模式。									

位/位域	名称	类型	复位值	描述						
1	SLEEPEXIT	R/W	0	<p>在退出 ISR 后睡眠</p> <table border="1"> <thead> <tr> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>当从处理模式返回到线程模式后，并不进入睡眠状态。</td> </tr> <tr> <td>1</td> <td>当从处理模式返回到线程模式后，一退出 ISR 则立即进入睡眠或深度睡眠状态。</td> </tr> </tbody> </table> <p>此功能适用于中断驱动型的应用程序。若此标志位置位，可避免处理器返回到无内容的主函数。</p>	取值	描述	0	当从处理模式返回到线程模式后，并不进入睡眠状态。	1	当从处理模式返回到线程模式后，一退出 ISR 则立即进入睡眠或深度睡眠状态。
取值	描述									
0	当从处理模式返回到线程模式后，并不进入睡眠状态。									
1	当从处理模式返回到线程模式后，一退出 ISR 则立即进入睡眠或深度睡眠状态。									
0	保留	RO	0	<p>软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。</p>						

寄存器 35：配置及控制寄存器 (CFGCTRL)，偏移量 0xD14

注：本寄存器只能在特权模式下访问。

CFGCTRL 寄存器负责控制线程模式的入口，此外还能够使能：**NMI**处理程序、硬故障处理程序以及由**FAULTMASK**寄存器升级后的故障处理程序忽略总线故障；被零除以及未对齐访问的额外处理；非特权级软件对**SWTRIG**寄存器的访问（见第 145 页）。

配置及控制寄存器 (CFGCTRL)

基地址 0xE000 E000

偏移量 0xD14

类型 R/W，复位值 0x0000 0200

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	保留																
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	保留						STKALIGN	BHFHNMIGN		保留			DIV0	UNALIGNED	保留	MAINPEND	BASETHR
类型	RO	RO	RO	RO	RO	RO	R/W	R/W	RO	RO	RO	R/W	R/W	RO	R/W	R/W	
复位值	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	

位/位域	名称	类型	复位值	描述
31:10	保留	RO	0x0000 0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
9	STKALIGN	R/W	1	异常入口的栈对齐 取值 描述 0 栈按 4 字节对齐。 1 栈按 8 字节对齐。
8	BHFHNMIGN	R/W	0	在 NMI 及故障中忽略总线故障 通过此标志位可以使得优先级为-1 和-2 的异常处理程序忽略由装载指令和存贮指令产生的数据总线故障。此标志位的设置适用于 NMI、硬故障以及经 FAULTMASK 寄存器升级后的处理程序。 取值 描述 0 由装载指令及存贮指令产生的数据总线故障都会导致错误锁定。 1 优先级为-1 和-2 的处理程序均忽略装载指令及存贮指令产生的数据总线故障。
				只有当处理程序及其数据均位于绝对安全的存储器中时，才允许将此标志位置位。此标志位通常用于检测系统设备及总线桥有无控制通道隐患，并予以修复。

位/位域	名称	类型	复位值	描述
7:5	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
4	DIV0	R/W	0	被零除的额外处理 若处理器执行 SDIV 或 UDIV 指令时遭遇除数为零的状况，本标志位将决定是否产生故障或暂停运行。 取值 描述 0 当遇到被零除的状况时不进行额外处理，商直接赋 0。 1 当遇到被零除的状况时，进行额外处理。
3	UNALIGNED	R/W	0	未对齐访问的额外处理 取值 描述 0 当遇到未对齐的半字访问或字访问时，不进行额外处理。 1 当遇到未对齐的半字访问或字访问时，进行额外处理：此时将产生一个用法故障。 未对齐的 LDM、STM、LDRD、STRD 指令始终产生故障，与本标志位的设置无关。
2	保留	RO	0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
1	MAINPEND	R/W	0	允许主中断触发 取值 描述 0 禁止非特权级软件访问 SWTRIG 寄存器。 1 允许非特权级软件访问 SWTRIG 寄存器（见第 145 页）。
0	BASETHR	R/W	0	线程状态控制 取值 描述 0 仅当异常激活后才允许处理器进入线程模式。 1 取决于对 EXC_RETURN 值的控制，处理器随时都可进入线程模式，详见第 111 页的“异常返回”一节。

寄存器 36: 系统处理程序优先级寄存器 1 (SYSPRI1), 偏移量 0xD18

注: 本寄存器只能在特权模式下访问。

SYSPRI1 寄存器用于配置用法故障、总线故障、存储器管理故障的异常处理程序的优先级, 取值范围为 0~7。本寄存器可以按字节访问。

系统处理程序优先级寄存器 1 (SYSPRI1)

基地址 0xE000 E000

偏移量 0xD18

类型 R/W, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								USAGE			保留				
类型	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BUS			保留					MEM			保留				
类型	R/W	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:24	保留	RO	0x00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
23:21	USAGE	R/W	0x0	用法故障优先级 本位域用于配置用法故障的优先级。优先级的取值范围为 0~7, 其中数值越小、所代表的优先级越高。
20:16	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
15:13	BUS	R/W	0x0	总线故障优先级 本位域用于配置总线故障的优先级。优先级的取值范围为 0~7, 其中数值越小、所代表的优先级越高。
12:8	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
7:5	MEM	R/W	0x0	存储器管理故障优先级 本位域用于配置存储器管理故障的优先级。优先级的取值范围为 0~7, 其中数值越小、所代表的优先级越高。
4:0	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。

寄存器 37：系统处理程序优先级寄存器 2 (SYSPRI2)，偏移量 0xD1C

注：本寄存器只能在特权模式下访问。

SYSPRI2 寄存器用于配置 SVCcall 处理程序的优先级，取值范围为 0~7。本寄存器可以按字节访问。

系统处理程序优先级寄存器 2 (SYSPRI2)

基地址 0xE000 E000

偏移量 0xD1C

类型 R/W，复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SVC			保留												
类型	R/W	R/W	R/W	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:29	SVC	R/W	0x0	SVCcall 优先级 本位域用于配置 SVCcall 的优先级。优先级的取值范围为 0~7，其中数值越小、所代表的优先级越高。
28:0	保留	RO	0x0000 0000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。

寄存器 38：系统处理程序优先级寄存器 3 (SYSPRI3)，偏移量 0xD20

注：本寄存器只能在特权模式下访问。

SYSPRI3 寄存器用于配置 SysTick 异常处理程序、PendSV 处理程序的优先级，取值范围为 0~7。本寄存器可以按字节访问。

系统处理程序优先级寄存器 3 (SYSPRI3)

基地址 0xE000 E000

偏移量 0xD20

类型 R/W，复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TICK			保留					PENDSV			保留				
类型	R/W	R/W	R/W	RO	RO	RO	RO	RO	R/W	R/W	R/W	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:29	TICK	R/W	0x0	SysTick 异常优先级 本位域用于配置 SysTick 异常的优先级。优先级的取值范围为 0~7，其中数值越小、所代表的优先级越高。
28:24	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
23:21	PENDSV	R/W	0x0	PendSV 优先级 本位域用于配置 PendSV 的优先级。优先级的取值范围为 0~7，其中数值越小、所代表的优先级越高。
20:0	保留	RO	0x0 0000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。

寄存器 39：系统处理程序控制及状态寄存器（SYSHNDCTRL），偏移量 0xD24

注：本寄存器只能在特权模式下访问。

SYSHNDCTRL 寄存器用于使能系统处理程序，并且能够指示用法故障、总线故障、存储器管理故障、SVC 异常的状态，以及系统处理程序的激活状态。

假如禁用了某个系统处理程序，又产生了相应故障，那么处理器会按照硬故障对其进行处理。

通过修改本寄存器可以改变系统异常的挂起状态或激活状态。操作系统内核可以通过写激活标志位来进行上下文切换，以这种方式更改当前的异常类型。

重要提示： 软件在更改本寄存器中某个激活标志位时，如果并未对已入栈内容进行正确的修正，则可能导致处理器产生故障异常。应确保写本寄存器的软件保存当前激活状态，并在之后复原。

使能系统处理程序后，假如必须修改本寄存器中的某个标志位，则必须通过读-修改-写流程予以操作，确保只修改所需的位。

系统处理程序控制及状态寄存器（SYSHNDCTRL）

基地址 0xE000 E000

偏移量 0xD24

类型 R/W，复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留													USAGE	BUS	MEM
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SVC	BUSP	MEMP	USAGEP	TICK	PNDSV	保留	MON	SVCA	保留			USGA	保留	BUSA	MEMA
类型	R/W	R/W	R/W	R/W	R/W	R/W	RO	R/W	R/W	RO	RO	RO	R/W	RO	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:19	保留	RO	0x000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
18	USAGE	R/W	0	用法故障使能 取值 描述 0 禁用用法故障异常。 1 使能用法故障异常。
17	BUS	R/W	0	总线故障使能 取值 描述 0 禁用总线故障异常。 1 使能总线故障异常。

位/位域	名称	类型	复位值	描述
16	MEM	R/W	0	存储器管理故障使能 取值 描述 0 禁用存储器管理故障异常。 1 使能存储器管理故障异常。
15	SVC	R/W	0	SVC 调用挂起 取值 描述 0 当前无正在挂起的 SVC 调用异常。 1 当前有正在挂起的 SVC 调用异常。 通过本标志位可修改 SVC 调用异常的挂起状态。
14	BUSP	R/W	0	总线故障挂起 取值 描述 0 当前无正在挂起的总线故障异常。 1 当前有正在挂起的总线故障异常。 通过本标志位可修改总线故障异常的挂起状态。
13	MEMP	R/W	0	存储器管理故障挂起 取值 描述 0 当前无正在挂起的存储器管理故障异常。 1 当前有正在挂起的存储器管理故障异常。 通过本标志位可修改存储器管理故障异常的挂起状态。
12	USAGEP	R/W	0	用法故障挂起 取值 描述 0 当前无正在挂起的用法故障异常。 1 当前有正在挂起的用法故障异常。 通过本标志位可修改用法故障异常的挂起状态。
11	TICK	R/W	0	SysTick 异常激活 取值 描述 0 SysTick 异常未激活。 1 SysTick 异常已激活。 通过本标志位可修改 SysTick 异常的激活状态，不过在修改本标志位之前，务必仔细阅读本寄存器的“重要提示”部分内容。

位/位域	名称	类型	复位值	描述
10	PND SV	R/W	0	<p>PendSV 异常激活</p> <p>取值 描述</p> <p>0 PendSV 异常未激活。</p> <p>1 PendSV 异常已激活。</p> <p>通过本标志位可修改 PendSV 异常的激活状态，不过在修改本标志位之前，务必仔细阅读本寄存器的“重要提示”部分内容。</p>
9	保留	RO	0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
8	MON	R/W	0	<p>调试监视器激活</p> <p>取值 描述</p> <p>0 调试监视器未激活。</p> <p>1 调试监视器已激活。</p>
7	SVCA	R/W	0	<p>SVC 调用激活</p> <p>取值 描述</p> <p>0 SVC 调用未激活。</p> <p>1 SVC 调用已激活。</p> <p>通过本标志位可修改 SVC 调用异常的激活状态，不过在修改本标志位之前，务必仔细阅读本寄存器的“重要提示”部分内容。</p>
6:4	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
3	USGA	R/W	0	<p>用法故障激活</p> <p>取值 描述</p> <p>0 用法故障未激活。</p> <p>1 用法故障已激活。</p> <p>通过本标志位可修改用法故障异常的激活状态，不过在修改本标志位之前，务必仔细阅读本寄存器的“重要提示”部分内容。</p>
2	保留	RO	0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。

位/位域	名称	类型	复位值	描述
1	BUSA	R/W	0	<p>总线故障激活</p> <p>取值 描述</p> <p>0 总线故障未激活。</p> <p>1 总线故障已激活。</p> <p>通过本标志位可修改总线故障异常的激活状态，不过在修改本标志位之前，务必仔细阅读本寄存器的“重要提示”部分内容。</p>
0	MEMA	R/W	0	<p>存储器管理故障激活</p> <p>取值 描述</p> <p>0 存储器管理故障未激活。</p> <p>1 存储器管理故障已激活。</p> <p>通过本标志位可修改存储器管理故障异常的激活状态，不过在修改本标志位之前，务必仔细阅读本寄存器的“重要提示”部分内容。</p>

寄存器 40: 可配置故障状态寄存器 (FAULTSTAT), 偏移量 0xD28

注: 本寄存器只能在特权模式下访问。

FAULTSTAT 寄存器能够指示产生存储器管理故障、总线故障、用法故障的原因。这些功能可以进一步划分为子寄存器, 如下所示:

- 用法故障状态子寄存器 (**UFAULTSTAT**), 即本寄存器的[31:16]位;
- 总线故障状态子寄存器 (**BFAULTSTAT**), 即本寄存器的[15:8]位;
- 存储器管理故障状态子寄存器 (**MFAULTSTAT**), 即本寄存器的[7:0]位。

FAULTSTAT 寄存器可按字节访问。**FAULTSTAT** 寄存器及其子寄存器可按如下方式访问:

- 对偏移量 0xD28 按字访问, 即可访问完整的 **FAULTSTAT** 寄存器;
- 对偏移量 0xD28 按字节访问, 即可访问 **MFAULTSTAT** 子寄存器;
- 对偏移量 0xD28 按半字访问, 即可访问 **MFAULTSTAT** 与 **BFAULTSTAT** 子寄存器;
- 对偏移量 0xD29 按字节访问, 即可访问 **BFAULTSTAT** 子寄存器;
- 对偏移量 0xD2A 按半字访问, 即可访问 **UFAULTSTAT** 子寄存器。

本寄存器中的标志位写 1 即可清零。

在故障处理程序中, 发生故障的实际地址可按如下方法确定:

1. 读取并保存存储器管理故障地址寄存器 (**MMADDR**) 或总线故障地址寄存器 (**FAULTADDR**) 的值;
2. 读取 **MFAULTSTAT** 子寄存器的 **MMARV** 标志位, 或 **BFAULTSTAT** 子寄存器的 **BFARV** 标志位, 以检查 **MMADDR** 或 **FAULTADDR** 寄存器的内容是否有效。

软件必须按照此序列操作, 因为另一个优先级更高的异常有可能会更改 **MMADDR** 或 **FAULTADDR** 寄存器的内容。例如, 如果某个优先级更高的处理程序抢占了当前的处理程序, 那么另一个故障可能会改写 **MMADDR** 或 **FAULTADDR** 寄存器的内容。

可配置故障状态寄存器 (FAULTSTAT)

基地址 0xE000 E000

偏移量 0xD28

类型 R/W1C, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留						DIV0	UNALIGN	保留				NOCP	INVPC	INVSTAT	UNDEF
类型	RO	RO	RO	RO	RO	RO	R/W1C	R/W1C	RO	RO	RO	RO	R/W1C	R/W1C	R/W1C	R/W1C
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BFARV	保留		BSTKE	BUSTKE	IMPRE	PRECISE	IBUS	MMARV	保留		MSTKE	MUSTKE	保留	DERR	IERR
类型	R/W1C	RO	RO	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	RO	RO	R/W1C	R/W1C	RO	R/W1C	R/W1C
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:26	保留	RO	0x00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。

位/位域	名称	类型	复位值	描述						
25	DIV0	R/W1C	0	<p>被零除用法故障</p> <table border="1"> <thead> <tr> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>未发生被零除故障，或并未使能被零除的额外处理。</td> </tr> <tr> <td>1</td> <td>处理器执行 SDIV 或 UDIV 指令时，除数为 0。</td> </tr> </tbody> </table> <p>若此标志位置位，原本用于异常返回的已入栈 PC 值将指向出现被零除现象的那条指令。</p> <p>如果需要对被零除进行额外处理，应将配置及控制寄存器 (CFGCTRL)（见第 158 页）的 DIV0 标志位置位。</p> <p>对本标志位写 1，即可将其清零。</p>	取值	描述	0	未发生被零除故障，或并未使能被零除的额外处理。	1	处理器执行 SDIV 或 UDIV 指令时，除数为 0。
取值	描述									
0	未发生被零除故障，或并未使能被零除的额外处理。									
1	处理器执行 SDIV 或 UDIV 指令时，除数为 0。									
24	UNALIGN	R/W1C	0	<p>未对齐访问用法故障</p> <table border="1"> <thead> <tr> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>未发生未对齐访问故障，或并未使能未对齐访问的额外处理。</td> </tr> <tr> <td>1</td> <td>处理器企图执行未对齐访问操作。</td> </tr> </tbody> </table> <p>未对齐的 LDM、STM、LDRD、STRD 指令总会产生故障，与本标志位的配置无关。</p> <p>如果需要对未对齐访问进行额外处理，应将 CFGCTRL 寄存器（见第 158 页）的 UNALIGNED 标志位置位。</p> <p>对本标志位写 1，即可将其清零。</p>	取值	描述	0	未发生未对齐访问故障，或并未使能未对齐访问的额外处理。	1	处理器企图执行未对齐访问操作。
取值	描述									
0	未发生未对齐访问故障，或并未使能未对齐访问的额外处理。									
1	处理器企图执行未对齐访问操作。									
23:20	保留	RO	0x0	<p>软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。</p>						
19	NOCP	R/W1C	0	<p>无协处理器用法故障</p> <table border="1"> <thead> <tr> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>未发生企图访问协处理器的用法故障。</td> </tr> <tr> <td>1</td> <td>处理器企图访问协处理器。</td> </tr> </tbody> </table> <p>对本标志位写 1，即可将其清零。</p>	取值	描述	0	未发生企图访问协处理器的用法故障。	1	处理器企图访问协处理器。
取值	描述									
0	未发生企图访问协处理器的用法故障。									
1	处理器企图访问协处理器。									
18	INVPC	R/W1C	0	<p>装载无效 PC 值用法故障</p> <table border="1"> <thead> <tr> <th>取值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>未发生企图装载无效 PC 值的用法故障。</td> </tr> <tr> <td>1</td> <td>处理器企图执行非法的将 EXC_RETURN 装载到 PC 的操作。这可能是由于上下文无效，或由于 EXC_RETURN 值无效。</td> </tr> </tbody> </table> <p>若此标志位置位，原本用于异常返回的已入栈 PC 值将指向企图装载无效 PC 值的那条指令。</p> <p>对本标志位写 1，即可将其清零。</p>	取值	描述	0	未发生企图装载无效 PC 值的用法故障。	1	处理器企图执行非法的将 EXC_RETURN 装载到 PC 的操作。这可能是由于上下文无效，或由于 EXC_RETURN 值无效。
取值	描述									
0	未发生企图装载无效 PC 值的用法故障。									
1	处理器企图执行非法的将 EXC_RETURN 装载到 PC 的操作。这可能是由于上下文无效，或由于 EXC_RETURN 值无效。									

位/位域	名称	类型	复位值	描述
17	INVSTAT	R/W1C	0	<p>无效状态用法故障</p> <p>取值 描述</p> <p>0 未发生无效状态的用法故障。</p> <p>1 处理器执行了某条指令，该指令企图非法使用 EPSR 寄存器。</p> <p>若此标志位置位，原本用于异常返回的已入栈 PC 值将指向企图非法使用 EPSR 寄存器（执行程序状态寄存器）的那条指令。</p> <p>如果某条未定义的指令使用了 EPSR 寄存器，本标志位不会置位。</p> <p>对本标志位写 1，即可将其清零。</p>
16	UNDEF	R/W1C	0	<p>未定义指令用法故障</p> <p>取值 描述</p> <p>0 未发生未定义指令的用法故障。</p> <p>1 处理器企图执行一条未定义指令。</p> <p>若此标志位置位，原本用于异常返回的已入栈 PC 值将指向那条未定义指令。</p> <p>所谓未定义指令，就是指处理器内核无法对其正确解码的指令。</p> <p>对本标志位写 1，即可将其清零。</p>
15	BFARV	R/W1C	0	<p>总线故障地址寄存器有效</p> <p>取值 描述</p> <p>0 总线故障地址寄存器（FAULTADDR） 中未包含有效的故障地址。</p> <p>1 FAULTADDR 寄存器中包含有效的故障地址。</p> <p>当出现总线故障后，若故障地址已知，则本标志位将置位。其它故障有可能清零本标志位，例如之后若发生存储器管理故障，即会清零本标志位。</p> <p>若产生总线故障，并且由于优先级而升级为硬故障，那么硬故障处理程序必须清零本标志位。否则，当程序返回到已入栈的总线故障处理程序时，处理程序以为 FAULTADDR 寄存器仍然有效，但实际上其内容已经被覆盖。</p> <p>对本标志位写 1，即可将其清零。</p>
14:13	保留	RO	0x0	<p>软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。</p>

位/位域	名称	类型	复位值	描述
12	BSTKE	R/W1C	0	<p>入栈总线故障</p> <p>取值 描述</p> <p>0 未发生入栈操作时的用法故障。</p> <p>1 当在异常入口处进行入栈操作时产生一个或多个总线故障。</p> <p>若此标志位置位，SP 寄存器的值仍然会正常变更，但是栈内上下文的内容可能包含错误内容。此类总线故障并不向 FAULTADDR 寄存器写入故障地址。</p> <p>对本标志位写 1，即可将其清零。</p>
11	BUSTKE	R/W1C	0	<p>出栈总线故障</p> <p>取值 描述</p> <p>0 未发生出栈操作时的用法故障。</p> <p>1 当在异常出口处进行出栈操作时产生一个或多个总线故障。</p> <p>此故障将使得处理器再次返回到异常处理程序。因此，若此标志位置位，原始的返回栈并不会被破坏。由于前一次返回失败，因此 SP 寄存器的值不变，也不会进行入栈操作。此类总线故障并不向 FAULTADDR 寄存器写入故障地址。</p> <p>对本标志位写 1，即可将其清零。</p>
10	IMPRE	R/W1C	0	<p>不精确数据总线错误</p> <p>取值 描述</p> <p>0 未产生不精确数据总线错误。</p> <p>1 产生了数据总线错误，但栈中的返回地址未必指向导致错误产生的那条指令。</p> <p>若此标志位置位，不会向 FAULTADDR 寄存器写入故障地址。</p> <p>此故障是异步故障。因此，当检测到故障发生时，当前进程的优先级高于总线故障优先级，总线故障将保持挂起状态，只有等到处理器完成所有高优先级的进程后，才会转入激活状态。假如某个精确故障发生，在处理器进入处理程序之前，不精确总线故障，那么处理程序可以查询到 IMPRE 位与某个精确故障状态位同时置位。</p> <p>对本标志位写 1，即可将其清零。</p>

位/位域	名称	类型	复位值	描述
9	PRECISE	R/W1C	0	<p>精确数据总线错误</p> <p>取值 描述</p> <p>0 未产生精确数据总线错误。</p> <p>1 产生了数据总线错误，并且原本用于异常返回的已入栈 PC 值指向导致错误发生的那条指令。</p> <p>若此标志位置位，将会向 FAULTADDR 寄存器写入故障地址。</p> <p>对本标志位写 1，即可将其清零。</p>
8	IBUS	R/W1C	0	<p>指令总线错误</p> <p>取值 描述</p> <p>0 未发生指令总线错误。</p> <p>1 发生了指令总线错误。</p> <p>处理器在预取指时就已经检测指令总线是否出错，不过只有在试图执行错误指令时才会置位本标志位。</p> <p>若此标志位置位，不会向 FAULTADDR 寄存器写入故障地址。</p> <p>对本标志位写 1，即可将其清零。</p>
7	MMARV	R/W1C	0	<p>存储器管理故障地址寄存器有效</p> <p>取值 描述</p> <p>0 存储器管理故障地址寄存器 (MMADDR) 中未包含有效的故障地址。</p> <p>1 MMADDR 寄存器中包含有效的故障地址。</p> <p>若产生存储器管理故障，并且由于优先级而升级为硬故障，那么硬故障处理程序必须清零本标志位。否则，当程序返回到已入栈的存储器管理故障处理程序时，处理程序以为 FAULTADDR 寄存器仍然有效，但实际上其内容已经被覆盖。</p> <p>对本标志位写 1，即可将其清零。</p>
6:5	保留	RO	0x0	<p>软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。</p>
4	MSTKE	R/W1C	0	<p>入栈访问违规</p> <p>取值 描述</p> <p>0 未发生入栈操作时的访问违规。</p> <p>1 当在异常入口处进行入栈操作时产生一个或多个访问违规。</p> <p>若此标志位置位，SP 寄存器的值仍然会正常变更，但是栈内上下文的内容可能包含错误内容。此类故障并不向 MMADDR 寄存器写入故障地址。</p> <p>对本标志位写 1，即可将其清零。</p>

位/位域	名称	类型	复位值	描述
3	MUSTKE	R/W1C	0	<p>出栈访问违规</p> <p>取值 描述</p> <p>0 未发生出栈操作时的访问违规。</p> <p>1 当在异常出口处进行出栈操作时产生一个或多个访问违规。</p> <p>此故障将使得处理器再次返回到异常处理程序。因此，若此标志位置位，原始的返回栈并不会被破坏。由于前一次返回失败，因此 SP 寄存器的值不变，也不会进行入栈操作。此类故障并不向 MMADDR 寄存器写入故障地址。</p> <p>对本标志位写 1，即可将其清零。</p>
2	保留	RO	0	<p>软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。</p>
1	DERR	R/W1C	0	<p>数据访问违规</p> <p>取值 描述</p> <p>0 未发生数据访问违规。</p> <p>1 处理器试图对某个地址进行装载或存贮操作，但该地址并不允许执行该操作。</p> <p>若此标志位置位，原本用于异常返回的已入栈 PC 值将指向出错的那条指令，并且企图访问的地址将写入 MMADDR 寄存器中。</p> <p>对本标志位写 1，即可将其清零。</p>
0	IERR	R/W1C	0	<p>指令访问违规</p> <p>取值 描述</p> <p>0 未发生指令访问违规。</p> <p>1 处理器试图从某个地址取指，但该地址并不允许执行指令。</p> <p>当对任何 XN 存储区访问时，均会产生此故障，即使 MPU 被禁用、甚至未集成 MPU 也是如此。</p> <p>若此标志位置位，原本用于异常返回的已入栈 PC 值将指向出错的那条指令，并且企图访问的地址将写入 MMADDR 寄存器中。</p> <p>对本标志位写 1，即可将其清零。</p>

寄存器 41：硬故障状态寄存器 (HFAULTSTAT)，偏移量 0xD2C

注：本寄存器只能在特权模式下访问。

HFAULTSTAT 寄存器提供导致触发硬故障处理程序的事件的相关信息。

本寄存器中的标志位写 1 即可清零。

硬故障状态寄存器 (HFAULTSTAT)

基地址 0xE000 E000

偏移量 0xD2C

类型 R/W1C，复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DBG	FORCED	保留				保留									
类型	R/W1C	R/W1C	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留														VECT	保留
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W1C	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31	DBG	R/W1C	0	调试事件 此标志位保留用于调试。平时此标志位必须写为 0，否则可能产生无法预料的后果。
30	FORCED	R/W1C	0	强制硬故障 取值 描述 0 未发生强制硬故障。 1 发生了强制硬故障。其原因可能是某个优先级可配置的故障无法得到及时处理（可能是因为优先级较低或被禁用）导致其升级。 若此标志位置位，则硬故障处理程序必须读取其它故障状态寄存器，找出故障原因。
29:2	保留	RO	0x000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
1	VECT	R/W1C	0	向量表读取故障 取值 描述 0 在读取向量表时未发生总线故障。 1 在读取向量表时发生总线故障。 此错误始终由硬故障处理程序处理。 若此标志位置位，原本用于异常返回的已入栈 PC 值将指向被异常抢占使用权的那条指令。 对本标志位写 1，即可将其清零。

位/位域	名称	类型	复位值	描述
0	保留	RO	0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。

寄存器 42: 存储器管理故障地址寄存器 (MMADDR), 偏移量 0xD34

注: 本寄存器只能在特权模式下访问。

MMADDR 寄存器中包含造成存储器管理故障的地址。当发生未对齐访问故障时, **MMADDR** 寄存器中包含的是产生故障的实际地址。对于其它类型故障, 由于单条读写指令还可能进一步分割为多个对齐操作, 因此故障地址可能是被请求访问范围内的任一地址。**存储器管理故障状态寄存器 (MFAULTSTAT)** 中的状态位能够指示出故障原因以及 **MMADDR** 寄存器中的值是否有效, 见第 167 页。

存储器管理故障地址寄存器 (MMADDR)

基地址 0xE000 E000

偏移量 0xD34

类型 R/W, 复位值-

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位/位域	名称	类型	复位值	描述
31:0	ADDR	R/W	-	故障地址

若 **MFAULTSTAT** 寄存器的 **MMARV** 置位, 则本位域包含产生存储器管理故障的地址。

寄存器 43: 总线故障地址寄存器 (FAULTADDR), 偏移量 0xD38

注: 本寄存器只能在特权模式下访问。

FAULTADDR 寄存器包含造成总线故障的地址。当发生未对齐访问故障时, **FAULTADDR** 寄存器中包含的是错误指令所请求的地址 (即使其并非产生故障的实际地址)。**总线故障状态寄存器 (BFaultStat)** 中的状态位能够指示出故障原因 **FAULTADDR** 寄存器中的值是否有效, 见 [第 167 页](#)。

总线故障地址寄存器 (FAULTADDR)

基地址 0xE000 E000

偏移量 0xD38

类型 R/W, 复位值-

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

位/位域	名称	类型	复位值	描述
31:0	ADDR	R/W	-	故障地址

若 **BFaultStat** 寄存器的 **FAULTADDRV** 置位, 则本位域包含产生总线故障的地址。

3.6 存储器保护单元 (MPU) 寄存器描述

本节按照地址偏移量由小到大的顺序依次详细介绍存储器保护单元 (MPU) 的各寄存器。

MPU 寄存器只能在特权模式下访问。

寄存器 44: MPU 类型寄存器 (MPUTYPE), 偏移量 0xD90

注: 本寄存器只能在特权模式下访问。

MPUTYPE 寄存器用于指示本芯片是否包含 MPU, 若包含 MPU 还能指示出其支持多少个存储区。

MPU 类型寄存器 (MPUTYPE)

基地址 0xE000 E000

偏移量 0xD90

类型 RO, 复位值 0x0000 0800

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留								IREGION							
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DREGION								保留						SEPARATE	
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:24	保留	RO	0x00	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
23:16	IREGION	RO	0x00	I 区数目 此位域可显示出芯片所支持的 MPU 指令区数目。此位域始终为 0。MPU 存储器映射是统一的, 因此可参照 DREGION 位域的设置。
15:8	DREGION	RO	0x08	D 区数目 取值 描述 8 表示总共支持 8 个 MPU 数据区。
7:1	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
0	SEPARATE	RO	0	MPU 分立还是统一 取值 描述 0 表示 MPU 是统一管理的。

寄存器 45: MPU 控制寄存器 (MPUCTRL), 偏移量 0xD94

注: 本寄存器只能在特权模式下访问。

MPUCTRL 寄存器能够使能 MPU、使能默认存储器映射背景区, 此外还能选择在硬故障、不可屏蔽中断 (NMI), 以及**故障掩码寄存器 (FAULTMASK)** 升级后的处理程序中是否允许使用 MPU。

当 **ENABLE** 与 **PRIVDEFEN** 同时置位后:

- 对于特权级访问, 默认的存储器映射参见第 94 页的“存储器模型”一节。若某存储区并未使能, 则特权级软件的任何访问特性将与默认的存储器映射相同。
- 若某存储区并未使能, 则非特权级软件的任何访问都将产生存储器管理故障。

不论 **ENABLE** 位的值如何, 从未执行 (XN) 以及严格顺序规则总适用于系统控制空间。

若 **ENABLE** 置位, 一般来说存储器映射中必须至少使能一个存储区, 这样才能保障系统工作正常。若没有使能的存储区, 如果 **PRIVDEFEN** 也置位, 那么特权级软件还能正常工作。

若 **ENABLE** 清零, 系统将采用默认的存储器映射, 其存储器属性与未采用 MPU 之前的属性相同 (参见第 97 页的表 2-5)。此时无论特权级软件还是非特权级软件均按照默认的存储器映射进行访问。

当使能 MPU 后, 将始终允许对系统控制空间以及向量表的访问。对其它区域的访问权限则取决于区属性设置以及 **PRIVDEFEN** 是否置位。

除非 **HFNMENA** 置位, 否则当处理器执行优先级为-1 或-2 的异常服务程序时, MPU 将不会使能。这两个优先级仅对硬故障、NMI 或使能 **FAULTMASK** 后的异常处理有效。若 **HFNMENA** 置位, 则在这两个优先级下工作时, 也将使能 MPU。

MPU 控制寄存器 (MPUCTRL)

基地址 0xE000 E000

偏移量 0xD94

类型 R/W, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													PRIVDEFEN	HFNMENA	ENABLE
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:3	保留	RO	0x0000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。

位/位域	名称	类型	复位值	描述
2	PRIVDEFEN	R/W	0	<p>MPU 默认区</p> <p>此标志位用于选择是否允许特权级软件按照默认的存储器映射进行访问。</p> <p>取值 描述</p> <p>0 若 MPU 已经使能，此标志位为 0 代表禁用默认存储器映射。对某个地址进行访问时，若该地址不处于任一使能的区中，则会产生故障。</p> <p>1 若 MPU 已经使能，此标志位为 1 代表允许特权级软件按照默认存储器映射作为背景区进行访问。</p> <p>若此标志位置位，背景区将如同编号为-1 的存储区一样正常工作。任何已定义、已使能的区都将优于背景区。</p> <p>若 MPU 未使能，处理器将忽略本标志位。</p>
1	HFNMIENA	R/W	0	<p>故障期间使能 MPU</p> <p>此标志位控制 MPU 在硬故障、NMI 以及 FAULTMASK 处理程序期间的表现。</p> <p>取值 描述</p> <p>0 不管 ENABLE 标志位的状态如何，在硬故障、NMI 以及 FAULTMASK 处理程序期间都将自动禁用 MPU。</p> <p>1 在硬故障、NMI 以及 FAULTMASK 处理程序期间，仍然使能 MPU。</p> <p>若此标志位置位，而 MPU 又没有使能，那么其后果将无法预料。</p>
0	ENABLE	R/W	0	<p>MPU 使能</p> <p>取值 描述</p> <p>0 禁用 MPU。</p> <p>1 使能 MPU。</p> <p>若 HFNMIENA 标志位置位，而 MPU 又没有使能，那么其后果将无法预料。</p>

寄存器 46: MPU 区编号寄存器 (MPUNUMBER), 偏移量 0xD98

注: 本寄存器只能在特权模式下访问。

MPUNUMBER 寄存器用于选择 **MPU 区基地址寄存器 (MPUBASE)** 以及 **MPU 区属性及大小寄存器 (MPUATTR)** 所针对的存储区编号。一般在访问 **MPUBASE** 及 **MPUATTR** 寄存器之前, 应将需要操作的存储区编号写入本寄存器。不过, 若在写 **MPUBASE** 寄存器的同时将 **VALID** 置位 (见第 181 页), 也能够按照 **REGION** 位域改变相应区的内容。

MPU 区编号寄存器 (MPUNUMBER)

基地址 0xE000 E000

偏移量 0xD98

类型 R/W, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	保留															
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	保留													NUMBER		
类型	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:3	保留	RO	0x0000 000	软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。
2:0	NUMBER	R/W	0x0	待操作的 MPU 区 此位域用于指定当前由 MPUBASE 及 MPUATTR 寄存器所代表的 MPU 区。MPU 支持 8 个存储区。

寄存器 47: MPU 区基地址寄存器 (MPUBASE), 偏移量 0xD9C

寄存器 48: MPU 区基地址别名寄存器 1 (MPUBASE1), 偏移量 0xDA4

寄存器 49: MPU 区基地址别名寄存器 2 (MPUBASE2), 偏移量 0xDAC

寄存器 50: MPU 区基地址别名寄存器 3 (MPUBASE3), 偏移量 0xDB4

注: 本寄存器只能在特权模式下访问。

MPUBASE 寄存器用于定义由 **MPU 区编号寄存器 (MPUNUMBER)** 所指定 MPU 区的基地址, 此外也能够直接更新 **MPUNUMBER** 寄存器的内容。如果想修改当前的存储区编号并连带更新 **MPUNUMBER** 寄存器内容, 应在写 **MPUBASE** 寄存器时将 **VALID** 位置位。

MPUBASE 寄存器的有效 **ADDR** 位域是 [31:N], 剩余的 [N-1:5] 位域将保留。N 的值由区大小所决定, 而区大小由 **MPU 区属性及大小寄存器 (MPUATTR)** 的 **SIZE** 位域决定, 见 [第 183 页](#)。N 值的定义如下:

$$N = \log_2(\text{区大小}) \quad \text{其中区大小以字节数表示}$$

若在 **MPUATTR** 寄存器中配置区大小为 **4GB**, 那就没有可用的 **ADDR** 位域了。此时, 该存储区将占据处理器的全部寻址空间, 其基地址为 **0x0000 0000**。

基地址必须按照存储区的大小对齐。例如, 定义某个区大小为 **64kB**, 则其基地址必须按照 **64kB** 的倍数对齐, 例如可以是 **0x0001 0000** 或 **0x0002 0000**。

MPU 区基地址寄存器 (MPUBASE)

基地址 0xE000 E000

偏移量 0xD9C

类型 R/W, 复位值 0x0000 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR															
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR											VALID	保留	REGION		
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	WO	RO	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:5	ADDR	R/W	0x0000 000	基地址掩码

[31:N]位域包含存储器区的基地址。N 的值取决于区大小, 如前所述。剩余的[N-1:5]位域应保留。

软件不得依赖保留位的值。为保障软件与将来产品的兼容性, 对寄存器进行“读-修改-写”操作时, 不得变更保留位的值。

位/位域	名称	类型	复位值	描述
4	VALID	WO	0	<p>区编号有效</p> <p>取值 描述</p> <p>0 MPUNUMBER 寄存器不变，处理器按照 MPUNUMBER 寄存器的设置更新存储区的基地址，忽略 REGION 位域的设置。</p> <p>1 MPUNUMBER 寄存器将更新为 REGION 位域的值，并依此更新相应存储区的基地址。</p> <p>此标志位的回读值始终为 0。</p>
3	保留	RO	0	<p>软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。</p>
2:0	REGION	R/W	0x0	<p>区编号</p> <p>写入本位域时，其包含将要写入 MPUNUMBER 寄存器的值。</p> <p>读取本位域时，将返回当前 MPUNUMBER 寄存器的内容。</p>

寄存器 51: MPU 区属性及大小寄存器 (MPUATTR), 偏移量 0xDA0

寄存器 52: MPU 区属性及大小别名寄存器 1 (MPUATTR1), 偏移量 0xDA8

寄存器 53: MPU 区属性及大小别名寄存器 2 (MPUATTR2), 偏移量 0xDB0

寄存器 54: MPU 区属性及大小别名寄存器 3 (MPUATTR3), 偏移量 0xDB8

注: 本寄存器只能在特权模式下访问。

MPUATTR 寄存器用于定义由 **MPU 区编号寄存器 (MPUNUMBER)** 所指定 MPU 区的大小以及存储器属性, 并且能够使能该 MPU 区及其子区。

MPUATTR 寄存器可按字或半字对齐访问, 其高半字保存区属性, 低半字保存区大小、区使能标志以及各子区的使能标志。

MPU 访问权限属性位 (XN、AP、TEX、S、C、B) 用于控制对相应存储区的访问。假如试图访问某个区域, 而该区域并未开放相关权限, 那么 MPU 将会产生一个访问权限故障。

SIZE 位域用于定义由 **MPUNUMBER** 寄存器所指定的存储区的大小, 如下所示:

区大小 = $2^{(SIZE+1)}$ 其中区大小以字节数表示

存储区最小允许为 32 字节, 即 SIZE 位域最小允许为 4。第 183 页的表 3-9 列出了 SIZE 位域取值与区大小关系的一些示例, 同时列出了 **MPU 区基地址寄存器 (MPUBASE)** 中对应的 N 值。

表 3-9. SIZE 位域取值示例

SIZE 位域编码	区大小	N 值 ^a	说明
00100b (0x4)	32B	5	允许的最小区大小
01001b (0x9)	1kB	10	-
10011b (0x13)	1MB	20	-
11101b (0x1D)	1GB	30	-
11111b (0x1F)	4GB	MPUBASE 寄存器中不包含 ADDR 位域; 该区已经占据全部寻址空间。	允许的最大区大小

a: 指 **MPUBASE** 寄存器中的 N 参数, 详见第 181 页。

MPU 区属性及大小寄存器 (MPUATTR)

基地址 0xE000 E000

偏移量 0xDA0

类型 R/W, 复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			XN	保留	AP			保留			TEX		S	C	B
类型	RO	RO	RO	R/W	RO	R/W	R/W	R/W	RO	RO	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRD				保留					SIZE				ENABLE		
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RO	RO	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位/位域	名称	类型	复位值	描述
31:29	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
28	XN	R/W	0	禁止访问指令 取值 描述 0 使能取值操作。 1 禁止取值操作。
27	保留	RO	0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
26:24	AP	R/W	0x0	访问权限 关于此位域的详细信息，可参阅第 125 页的表 3-5。
23:22	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
21:19	TEX	R/W	0x0	类型扩展掩码 关于此位域的详细信息，可参阅第 124 页的表 3-3。
18	S	R/W	0	可共享性 关于此位域的详细信息，可参阅第 124 页的表 3-3。
17	C	R/W	0	可高速缓冲性 关于此位域的详细信息，可参阅第 124 页的表 3-3。
16	B	R/W	0	可缓冲性 关于此位域的详细信息，可参阅第 124 页的表 3-3。
15:8	SRD	R/W	0x00	子区禁用位 取值 描述 0 使能相应子区。 1 禁用相应子区。 存储区小于等于 128 字节时，将无法再分割为子区。此时写入存储区属性时，应将 SRD 位域始终配置为 0x00。参见第 123 页的“存储子区”一节。
7:6	保留	RO	0x0	软件不得依赖保留位的值。为保障软件与将来产品的兼容性，对寄存器进行“读-修改-写”操作时，不得变更保留位的值。
5:1	SIZE	R/W	0x0	区大小掩码 SIZE 位域用于定义由 MPUNUMBER 寄存器指定的 MPU 存储区的大小。关于此方面的详细信息，请参阅第 183 页的表 3-9。

位/位域	名称	类型	复位值	描述
0	ENABLE	R/W	0	区使能
				取值 描述
				0 禁用此存储器区。
				1 使能此存储器区。

北京锐鑫同创公司相关信息

技术支持

如果您对文档有所疑问，您可以在办公时间（星期一至星期五上午 8:30~11:50；下午 1:30~5:30）拨打技术支持电话或 E-mail 联系。

北京锐鑫同创是 TI 第三方合作伙伴，专注于 TI Stellaris M3 产品的市场推广、方案设计和技术服务，同时提供开发板、仿真器、编程器等开发工具，公司以“把握市场脉搏，专注技术创新，提供诚信服务，实现共赢发展！”为核心价值理念，为客户提供实时、高效的技术和服务。

电话：010-82418301

传真：010-82418302

Email: support@realsense.com.cn

网站: www.realsense.com.cn

技术论坛: www.hellom3.cn