附录B

# 16位Thumb指令及架构版本

### 表B.1 不同ARM架构版本下对16位指令的改动

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令 | v4T | v5T | v6 | Cortex-M3(v7-M) |
| BKPT |  | Y | Y | Y |
| BLX |  | Y | Y | 只能使用BLX <reg> |
| CBZ,CBNZ |  |  |  | Y |
| CPS |  |  | Y | CPSIE <i/f>, CPSID <i/f> |
| CPY |  |  | Y | Y |
| NOP |  |  |  | Y |
| IT |  |  |  | Y |
| REV(多种形式) |  |  | Y | Y |
| SEV |  |  |  | Y |
| SETEND |  |  | Y |  |
| SWI | Y | Y | Y | 改为SVC |
| SXTB,SXTH |  |  | Y | Y |
| UXTB,UXTH |  |  | Y | Y |
| WFI,WFE |  |  |  | Y |

附录C

# Cortex-M3异常快速参考

### 表C.1 异常一览表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 类型 | 优先级 | 简介 |  |
| 0 | N/A | N/A | 没有异常在运行 |  |
| 1 | 复位 | -3（最高） | 复位 | 总是 |
| 2 | NMI | -2 | 不可屏蔽中断（来自外部NMI输入脚） | 总是 |
| 3 | 硬fault | -1 | 所有被除能的fault，都将“上访”(escalation)成硬fault。只要FAULTMASK没有置位，硬fault服务例程就被强制执行。Fault被除能的原因包括被禁用，或者被PRIMASK/BASEPRI掩蔽 | 总是 |
| 4 | Mem Manage fault | 可编程  E000\_ED18 | 存储器管理fault，MPU访问犯规以及访问非法位置均可引发。企图在“非执行区”取指也会引发此fault | NVIC SHCSR.16  E000\_ED24 |
| 5 | 总线fault | 可编程  E000\_ED19 | 从总线系统收到了错误响应，原因可以是预取流产（Abort）或数据流产，或者企图访问协处理器 | NVIC SHCSR.17  E000\_ED24 |
| 6 | 用法  Fault | 可编程  E000\_ED1A | 由于程序错误导致的异常。通常是使用了一条无效指令，或者是非法的状态转换，例如尝试切换到ARM状态 | NVIC SHCSR.18  E000\_ED24 |
| 7-10 | 保留 | N/A | N/A |  |
| 11 | SVCall | 可编程  E000\_ED1F | 执行系统服务调用指令（SVC）引发的异常 | 总是 |
| 12 | 调试监视器 | 可编程  E000\_ED20 | 调试监视器（断点，数据观察点，或者是外部调试请求 | NVIC DEMCR.16  E000\_EDFC |
| 13 | 保留 | N/A | N/A |  |
| 14 | PendSV | 可编程  E000\_ED22 | 为系统设备而设的“可悬挂请求”（pendable request） | 总是 |
| 15 | SysTick | 可编程  E000\_ED23 | 系统滴答定时器（也就是周期性溢出的时基定时器——译注） | SysTick CTRLSTAT.1  E000\_E010 |
| 16-  255 | IRQ | E000\_E400 | 多达240条外部中断 | NVIC SETENA寄存器阵列 |

### 表C.2 自动入栈后堆栈中的内容以及SP的调整

|  |  |  |
| --- | --- | --- |
| 地址 | 寄存器 | 被保存的顺序 |
| 旧SP (N-0) | 原先已压入的内容 | - |
| (N-4) | xPSR | 2 |
| (N-8) | PC | 1 |
| (N-12) | LR | 8 |
| (N-16) | R12 | 7 |
| (N-20) | R3 | 6 |
| (N-24) | R2 | 5 |
| (N-28) | R1 | 4 |
| 新SP (N-32) | R0 | 3 |
| 注意：如果启用了堆栈的双字对齐特性，但是SP却没能对齐到双字，则堆栈帧的顶部有可能从((OLD\_SP-4) AND 0xFFFF\_FFF8)处开始，且其余的内容被向下错位一个字 | | |

附录D

# NVIC寄存器小结

### 表D.1 中断控制器类型寄存器ICTR 0xE000\_E004

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 4:0 | INTLINESUM | R | - | 中断输入的数量，以32为粒度，如  0=1至32  1=33至64  2=65至96  … |

### 表D.2 SysTick控制及状态寄存器（地址：0xE000\_E010）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 16 | COUNTFLAG | R | 0 | 如果在上次读取本寄存器后，SysTick已经数到了0，则该位为1。如果读取该位，该位将自动清零 |
| 2 | CLKSOURCE | R/W | 0 | 0=外部时钟源(STCLK)  1=内核时钟(FCLK) |
| 1 | TICKINT | R/W | 0 | 1=SysTick倒数到0时产生SysTick异常请求  0=数到0时无动作 |
| 0 | ENABLE | R/W | 0 | SysTick定时器的使能位 |

### 表D.3 SysTick重装载数值寄存器（地址：0xE000\_E014）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 23:0 | RELOAD | R/W | 0 | 当倒数至零时，将被重装载的值 |

### 表D.4 SysTick当前数值寄存器（地址：0xE000\_E018）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 23:0 | CURRENT | R/Wc | 0 | 读取时返回当前倒计数的值，写它则使之清零，同时还会清除在SysTick控制及状态寄存器中的COUNTFLAG标志 |

### 表D.5 SysTick校准数值寄存器（地址：0xE000\_E01C）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 31 | NOREF | R | - | 1=没有外部参考时钟（STCLK不可用）  0=外部参考时钟可用 |
| 30 | SKEW | R | - | 1=校准值不是准确的10ms  0=校准值是准确的10ms |
| 23:0 | TENMS | R/W | 0 | 10ms的时间内倒计数的格数。芯片设计者应该通过Cortex-M3的输入信号提供该数值。若该值读回零，则表示无法使用校准功能 |

### 表D.6 SETENA/CLRENA寄存器族

SETENAs: xE000\_E100 – 0xE000\_E11C ; CLRENAs:0xE000E180 - 0xE000\_E19C

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 地址 | 复位值 | 描述 |
| SETENA0 | R/W | 0xE000\_E100 | 0 | 中断0-31的使能寄存器，共32个使能位  位[n]，中断#n使能（异常号16+n） |
| SETENA1 | R/W | 0xE000\_E104 | 0 | 中断32-63的使能寄存器，共32个使能位 |
| … | … | … | … | … |
| SETENA7 | R/W | 0xE000\_E11C | 0 | 中断224-239的使能寄存器，共16个使能位 |
|  |  |  |  |  |
|  |  |  |  |  |
| CLRENA0 | R/W | 0xE000\_E180 | 0 | 中断0-31的除能寄存器，共32个除能位  位[n]，中断#n除能（异常号16+n） |
| CLRENA1 | R/W | 0xE000\_E184 | 0 | 中断32-63的除能寄存器，共32个除能位 |
| … | … | … | … | … |
| CLRENA7 | R/W | 0xE000\_E19C | 0 | 中断224-239的除能寄存器，共16个除能位 |

### 表D.7 SETPEND/CLRPEND寄存器族

SETPENDs:0xE000\_E200 – 0xE000\_E21C ; CLRPENDs:0xE000E280 - 0xE000\_E29C

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 地址 | 复位值 | 描述 |
| SETPEND0 | R/W | 0xE000\_E200 | 0 | 中断0-31的悬起寄存器，共32个悬起位  位[n]，中断#n悬起（异常号16+n） |
| SETPEND1 | R/W | 0xE000\_E204 | 0 | 中断32-63的悬起寄存器，共32个悬起位 |
| … | … | … | … | … |
| SETPEND7 | R/W | 0xE000\_E21C | 0 | 中断224-239的悬起寄存器，共16个悬起位 |
|  |  |  |  |  |
|  |  |  |  |  |
| CLRPEND0 | R/W | 0xE000\_E280 | 0 | 中断0-31的解悬寄存器，共32个解悬位  位[n]，中断#n解悬（异常号16+n） |
| CLRPEND1 | R/W | 0xE000\_E284 | 0 | 中断32-63的解悬寄存器，共32个解悬位 |
| … | … | … | … | … |
| CLRPEND7 | R/W | 0xE000\_E29C | 0 | 中断224-239的解悬寄存器，共16个解悬位 |

### 表D.8 ACTIVE寄存器族 0xE000\_E300\_0xE000\_E31C

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 地址 | 复位值 | 描述 |
| ACTIVE0 | RO | 0xE000\_E300 | 0 | 中断0-31的活动状态寄存器，共32个状态位  位[n]，中断#n活动状态（异常号16+n） |
| ACTIVE1 | RO | 0xE000\_E304 | 0 | 中断32-63的活动状态寄存器，共32个状态位 |
| … | … | … | … | … |
| ACTIVE7 | RO | 0xE000\_E31C | 0 | 中断224-239的活动状态寄存器，共16个状态位 |

### 表D.9 中断优先级寄存器阵列 0xE000\_E400 – 0xE000\_E4EF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 地址 | 复位值 | 描述 |
| PRI\_0 | R/W | 0xE000\_E400 | 0（8位） | 外中断#0的优先级 |
| PRI\_1 | R/W | 0xE000\_E401 | 0（8位） | 外中断#1的优先级 |
| … | … | … | … | … |
| PRI\_239 | R/W | 0xE000\_E4EF | 0（8位） | 外中断#239的优先级 |

### 表D.10 CPUID寄存器 0xE000\_ED00

|  |  |  |  |
| --- | --- | --- | --- |
| 位段 | 名称 | 复位值 | 描述 |
| 31:24 | R | 0x41 | 实现者代码，ARM=0x41 |
| 23:20 | R | 0x0/0x1/0x02 | 实现定义的变种号 |
| 19:16 | R | 0xF | 常量 |
| 15:4 | R | 0xC23 | Part编号 |
| 3:0 | R | 0x0/0x1 | 修订号 |

### 表D.11 中断控制及状态寄存器ICSR 0xE000\_ED04

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 31 | NMIPENDSET | R/W | 0 | 写1以悬起NMI。因为NMI的优先级最高且从不掩蔽，在置位此位后将立即进入NMI服务例程。 |
| 28 | PENDSVSET | R/W | 0 | 写1以悬起PendSV。读取它则返回PendSV的状态 |
| 27 | PENDSVCLR | W | 0 | 写1以清除PendSV悬起状态 |
| 26 | PENDSTSET | R/W | 0 | 写1以悬起SysTick。读取它则返回PendSV的状态 |
| 25 | PENDSTCLR | W | 0 | 写1以清除SysTick悬起状态 |
| 23 | ISRPREEMPT | R | 0 | 为1时，则表示一个悬起的中断将在下一步时进入活动状态（用于单步执行时的调试目的） |
| 22 | ISRPENDING | R | 0 | 1=当前正有外部中断被悬起（不包括NMI） |
| 21:12 | VECTPENDING | R | 0 | 悬起的ISR的编号。如果不止一个中断悬起，则它的值是这引动中断中，优先级最高的那一个。 |
| 11 | RETTOBASE | R | 0 | 当从异常返回后将回到基级(base level)，且没有其它异常悬起时，此位为1。若是在线程模式下，在某个服务例程中，有不止一级的异常处于活动状态，或者在异常没有活动时执行了异常服务例程（此时执行返回指令将产生fault。此乃高危行为，大虾专用），则此位为0 |
| 9:0 | VECTACTIVE | R | 0 | 当前活动的ISR编号，该位段指出当前运行中的ISR是哪个中断的（提供异常序号），包括NMI和硬fault。如果多个异常共享一个服务例程，该例程可根据本位段的值来判定是哪一个异常的响应导致它的执行。把本位段的值减去16,就得到了外中断的编号，并可以用此编号来操作外中断相关的使能/除能等寄存器。 |

### 表D.12 向量表偏移量寄存器(VTOR) 0xE000\_ED08

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 7-28 | TBLOFF | RW | 0 | 向量表起始地址 |
| 29 | TBLBASE | R | - | 向量表是在Code区（0），还是在RAM区（1） |

### 表D.13 应用程序中断及复位控制寄存器(AIRCR) 0xE000\_ED0C

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 31:16 | VECTKEY | RW | - | 访问钥匙：任何对该寄存器的写操作，都必须同时把0x05FA写入此段，否则写操作被忽略。若读取此半字，则0xFA05 |
| 15 | ENDIANESS | R | - | 指示端设置。1＝大端(BE8)，0＝小端。此值是在复位时确定的，不能更改。 |
| 10:8 | PRIGROUP | R/W | 0 | 优先级分组 |
| 2 | SYSRESETREQ | W | - | 请求芯片控制逻辑产生一次复位 |
| 1 | VECTCLRACTIVE | W | - | 清零所有异常的活动状态信息。通常只在调试时用，或者在OS从错误中恢复时用。 |
| 0 | VECTRESET | W | - | 复位CM3处理器内核（调试逻辑除外），但是此复位不影响芯片上在内核以外的电路 |

### 表D.14 系统控制寄存器 0xE000\_ED10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 4 | SEVONPEND | RW | - | 发生异常悬起时请发送事件，用于在一个新的中断悬起时从WFE指令处唤醒。不管这个中断的优先级是否比当前的高，都唤醒。如果没有WFE导致睡眠，则下次使用WFE时将立即唤醒 |
| 3 | 保留 | - | - | - |
| 2 | SLEEPDEEP | R/W | 0 | 当进入睡眠模式时，使能外部的SLEEPDEEP信号，以允许停止系统时钟 |
| 1 | SLEEPONEXIT | R/W | - | 激活“SleepOnExit”功能 |
| 0 | 保留 | - | - | - |

### 表D.15 配置与控制寄存器 0xE000\_ED14

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 9 | STKALIGN | RW | 0(r1开始)  1(r2开始) | 在响应异常的自动入栈操作时，强制SP对齐到双字地址上。修订版0无此功能 |
| 8 | BFHFNMIGN | RW | 0 | 在硬fault与NMI服务例程中忽略数据总线fault |
| 7:5 | 保留 | - | - | - |
| 4 | DIV\_0\_TRP | RW | 0 | 除数为零时陷入用法fault |
| 3 | UNALIGN\_TRP | RW | 0 | 访问未对齐时陷入用法fault |
| 2 | 保留 | - | - | - |
| 1 | USERSETMPEND | RW | 0 | 如果为1,则允许用户代码设置STIR |
| 0 | NONBASETHRDENA | RW | 0 | 非基于线程模式使能位。如果为1,则允许异常服务例程通过修改EXC\_RETURN，使其在线程模式下执行 |

表D.16 系统异常优先级寄存器 0xE000\_ED18 － 0xE000\_ED23

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 地址 | 名称 | 类型 | 复位值 | 描述 |
| 0xE000\_ED18 | PRI\_4 |  |  | 存储器管理fault的优先级 |
| 0xE000\_ED19 | PRI\_5 |  |  | 总线fault的优先级 |
| 0xE000\_ED1A | PRI\_6 |  |  | 用法fault的优先级 |
| 0xE000\_ED1B | - | - | - | - |
| 0xE000\_ED1C | - | - | - | - |
| 0xE000\_ED1D | - | - | - | - |
| 0xE000\_ED1E | - | - | - | - |
| 0xE000\_ED1F | PRI\_11 |  |  | SVC优先级 |
| 0xE000\_ED20 | PRI\_12 |  |  | 调试监视器的优先级 |
| 0xE000\_ED21 | - | - | - | - |
| 0xE000\_ED22 | PRI\_14 |  |  | PendSV的优先级 |
| 0xE000\_ED23 | PRI\_15 |  |  | SysTick的优先级 |

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 |

表D.25 辅助fault地址寄存器(AFAR) 0xE000\_ED3C

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 31:0 | AFAR | R | - | 由芯片制造商决定（可选） |

表D.26 MPU类型寄存器MPUTR 0xE０00\_ED90

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 23:16 | IREGION | R | 0 | MPU支持的指令region数量。因为ARMv7-M只使用单个统一的MPU，此位段永远为零 |
| 15:8 | DREGION | R | 0 | MPU支持的数量。若系统中配了MPU则为8,否则为零 |
| 0 | SEPARATE | R | 0 | 固定为零 |

表D.27 MPU控制寄存器MPUCR （地址：0xE０00\_ED94）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 2 | PRIVDEFENA | RW | 0 | 是否为特权级打开缺省存储器映射（即背景region）。  1=特权级下打开背景region  0=不打开背景region。任何访问违例以及对region外地址区的访问都将引起fault |
| 1 | HFNMIENA | RW | 0 | 1=在NMI和硬fault服务例程中不强制除能MPU  0=在NMI和硬fault服务例程中强制除能MPU |
| 0 | ENABLE | RW | 0 | 使能MPU |

表D.28 MPU region号寄存器MPURNR （地址：0xE０00\_ED98）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 7:0 | REGION | RW | - | 选择下一个要配置的region。因为只支持8个region，所以事实上只有[2:0]有意义 |

表D.29 MPU region号寄存器MPURNR （地址：0xE０00\_ED9C）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 31:N | ADDR | RW | - | Region基址字段。N取决于region容量，以使**基址在数值上能被容量整除**。在MPU region属性及容量寄存器中有个SZENABLE位段，它决定ADDR中有多少个位被采用。 |
| 4 | VALID | RW | - | 决定是否理会写入REGION字段的值  1=MPU region号寄存器被REGION覆盖  0=MPU region号寄存器的值保持不变 |
| 3:0 | REGION | RW | - | MPU region覆写位段 |

表D.30 MPU region属性及容量寄存器MPURASR（地址：0xE000\_EDA0）

|  |  |  |  |
| --- | --- | --- | --- |
| 位段 | 长度 | 名称 | 功能 |
| 31:29 | 3 | - | 保留 |
| 28 | 1 | XN | 1=此区禁止取指  2=此区允许取指 |
| 27 | 1 | - | 保留 |
| 26:24 | 3 | AP | 访问许可，如下表所示   |  |  |  |  | | --- | --- | --- | --- | | 值 | 特权级下的许可 | 用户级下的许可 | 典型用法 | | 0b000 | 禁地 | 禁地 | 该区没有存储器，是空地址 | | 0b001 | RW | 禁地 | OS和系统软件使用的数据区 | | 0b010 | RW | RO | 禁止在用户级下更改的高危地带 | | 0b011 | RW | RW | 共享内存，或彻底开放的设备 | | 0b100 | n/a | n/a | n/a | | 0b101 | RO | 禁地 | OS使用的常量数据 | | 0b110 | RO | RO | 常量数据或只读存储器的地址区 | | 0b111 | RO | RO | 常量数据或只读存储器的地址区 | |
| 23:22 | 2 | － | 保留 |
| 21:19 | 3 | TEX | 类型扩展 |
| 18 | 1 | S | Sharable（可否共享）  1=共享可  0=共享不可 |
| 17 | 1 | C | Cachable（可否缓存）  1=缓存可  0=缓存不可 |
| 16 | 1 | B | Buffable（可否缓冲）  1=缓冲可  0=缓冲不可 |
| 15:8 | 8 | SRD | 子region除能位段。每设置SRD的一个位，就会除能与之对应的一个子region。容量大于128字节的region都被划分成8个容量相同的子region。容量小于等于128字节的region不能再分。更多信息，请参见对子Region的论述。 |
| 7:6 | 2 | - | 保留 |
| 5:1 | 5 | REGIONSIZE | Region容量，单位是字节。容量为1<<(REGIONSIZE+1)，但是最小容量为**32**字节 |
| 0 | 1 | SZENABLE | 1=使能此region 0=除能此region |

表D.31 调试停机控制及状态寄存器DHCSR （地址：0xE０00\_EDF0）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 31:15 | KEY | W | - | 调试钥匙。必须在任何写操作中把该位段写入A05F，否则忽略写操作 |
| 25 | S\_RESET\_ST | R | - | 内核已经或即将复位，读后清零 |
| 24 | S\_RETIRE\_ST | R | - | 在上次读取以后指令已执行完成，读后清零 |
| 19 | S\_LOCKUP | R | - | 1=内核进入锁定状态 |
| 18 | S\_SLEEP | R | - | 1=内核睡眠中 |
| 17 | S\_HALT | R | - | 1=内核已停机 |
| 16 | S\_REGRDY | R | - | 1=寄存器的访问已经完成 |
| 15:6 | 保留 | - | - |  |
| 5 | C\_SNAPSTALL | RW | 0\* | 打断一个stalled存储器访问 |
| 4 | 保留 | - | - |  |
| 3 | C\_MASKINTS | RW | 0\* | 调试期间关中断，只有在停机后方可设置 |
| 2 | C\_STEP | RW | 0\* | 让处理器单步执行，在C\_DEBUGEN=1时有效 |
| 1 | C\_HALT | RW | 0\* | 喊停处理器，在C\_DEBUGEN=1时有效 |
| 0 | C\_DEBUGEN | RW | 0\* | 使能停机模式的调试 |

表D.32 调试内核寄存器选择者寄存器DCRSR（地址：0xE０00\_EDF4）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 16 | REGWnR | W | - | 1=写寄存器  0=读寄存器 |
| 15:5 | 保留 | - | - | - |
| 4:0 | REGSEL | W | - | 00000= R0  00001=R1  …  01111=R15  10000=xPSR  10001=MSP  10010=PSP  10100=特殊功能寄存器组  [31:24]: CONTROL  [23:16]: FAULTMASK  [15:8]: BASEPRI  [7:0]: PRIMASK |

表D.33 调试内核寄存器数据寄存器DCRDR（地址：0xE０00\_EDF8）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 31:0 | DATA | R/W | - | 读回来的寄存器的值，或欲写入寄存器的值，寄存器由DCRSR选择 |

表D.34 调试乃及监视器控制寄存器DEMCR （地址：0xE０00\_EDFC）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 24 | TRCENA | RW | 0\* | 跟踪系统的使能位。在使用DWT,ETM,ITM和TPIU前，必须先设置此位 |
| 23:20 | 保留 |  |  |  |
| 19 | MON\_REQ | RW | 0 | 1=调试监视器异常不是由硬件调试事件触发，而是由软件手工悬起的 |
| 18 | MON\_STEP | RW | 0 | 让处理器单步执行，在MON\_EN=1时有效 |
| 17 | MON\_PEND | RW | 0 | 悬起监视器异常请求，内核将在优先级允许时响应 |
| 16 | MON\_EN | RW | 0 | 使能调试监视器异常 |
| 15:11 | 保留 |  |  |  |
| 10 | VC\_HARDERR | RW | 0\* | 发生硬fault时停机调试 |
| 9 | VC\_INTERR | RW | 0\* | 指令/异常服务错误时停机调试 |
| 8 | VC\_BUSERR | RW | 0\* | 发生总线fault时停机调试 |
| 7 | VC\_STATERR | RW | 0\* | 发生用法fault时停机调试 |
| 6 | VC\_CHKERR | RW | 0\* | 发生用法fault使能的检查错误时停机调试（如未对齐，除数为零） |
| 5 | VC\_NOCPERR | RW | 0\* | 发生用法fault之无处理器错误时停机调试 |
| 4 | VC\_MMERR | RW | 0\* | 发生存储器管理fault时停机调试 |
| 3:1 | 保留 |  |  |  |
| 0 | VC\_CORERESET | RW | 0\* | 发生内核复位时停机调试 |

\*：DEMCR中的控制位是在上电复位时得到复位的。系统复位（例如，往NVIC应用程序中断及复位寄存器中写命令）不会影响到它们

表D.35 软件触发中断寄存器STIR 0xE000\_EF00

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位段 | 名称 | 类型 | 复位值 | 描述 |
| 8:0 | INTID | W | - | 影响编号为INTID的外部中断，其悬起位被置位。例如，写入8，则悬起IRQ #8 |

表D36 中断优先级寄存器阵列 0xE000\_E400 – 0xE000\_E4EF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 地址 | 复位值 | 描述 |
| PERIPHID4 | R | 0xE000\_EFD0 | 0x04 | 外设ID寄存器4 |
| PERIPHID5 | R | 0xE000\_EFD4 | 0 | 外设ID寄存器5 |
| PERIPHID6 | R | 0xE000\_EFD8 | 0 | 外设ID寄存器6 |
| PERIPHID7 | R | 0xE000\_EFDC | 0 | 外设ID寄存器7 |
| PERIPHID0 | R | 0xE000\_EFE0 | 0 | 外设ID寄存器0 |
| PERIPHID1 | R | 0xE000\_EFE4 | 0 | 外设ID寄存器1 |
| PERIPHID2 | R | 0xE000\_EFE8 | 0x0B/0x1B | 外设ID寄存器2 |
| PERIPHID3 | R | 0xE000\_EFEC | 0 | 外设ID寄存器3 |
| PCELLID0 | R | 0xE000\_EFF0 | 0x0D | 组件ID寄存器0 |
| PCELLID1 | R | 0xE000\_EFF4 | 0xE0 | 组件ID寄存器1 |
| PCELLID2 | R | 0xE000\_EFF8 | 0x05 | 组件ID寄存器2 |
| PCELLID3 | R | 0xE000\_EFFC | 0xB1 | 组件ID寄存器3 |