

嵌入式系统

北京邮电大学
计算机学院

戴志涛



北京邮电大学

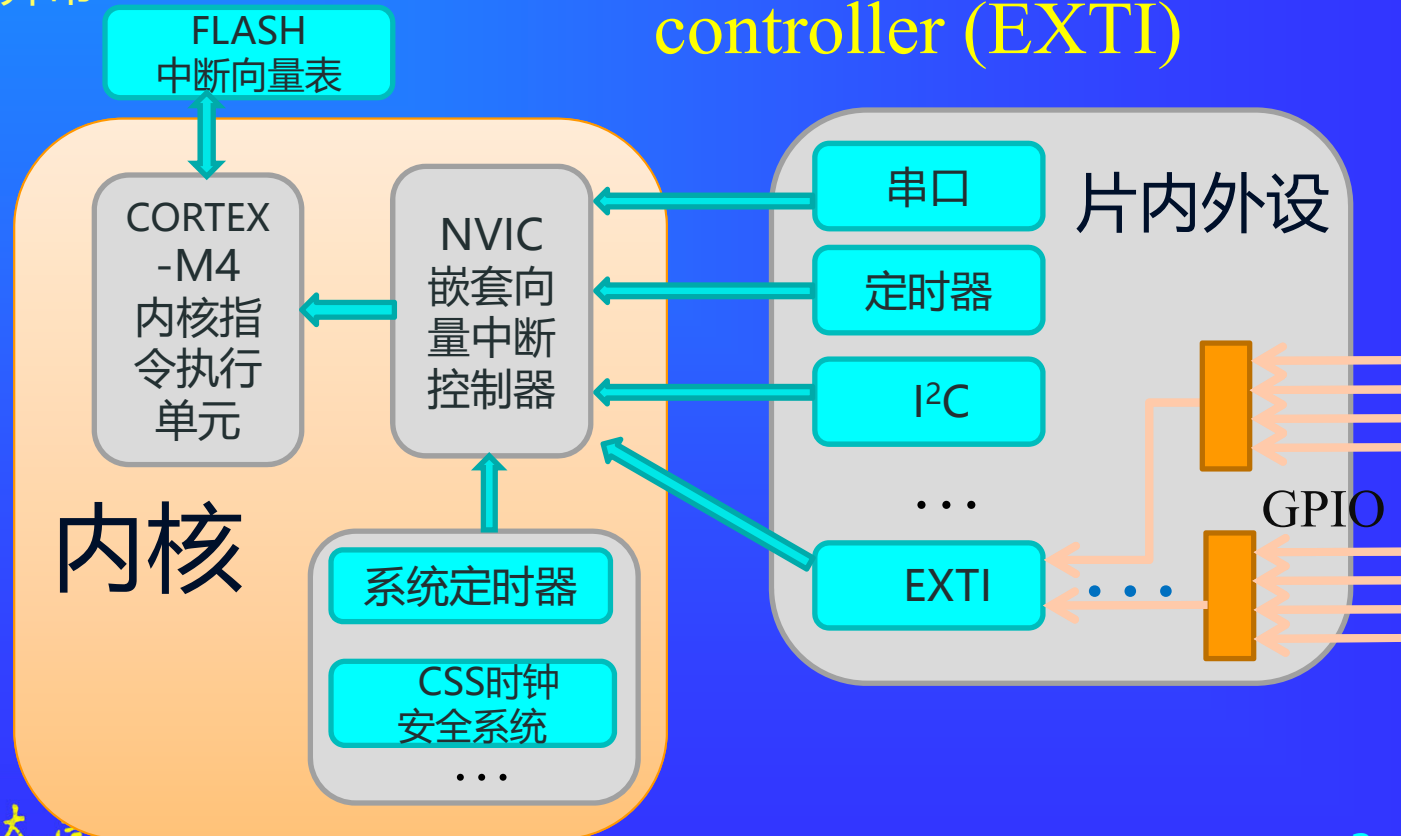
STM32F4的异常与中断



STM32F4的中断体系结构

- 中断：微处理器外部产生，通过中断通道送入处理器内部，一般由硬件引起
- 异常：通常在微处理器内部发生，大多由软件引起，如：
 - ❑ 除法出错异常
 - ❑ 特权调用异常

External interrupt/event controller (EXTI)



中断挂起



- Interrupt Pending: 中断挂起/中断悬挂/中断登记
- 挂起状态: 中断被置于一种等待处理器处理的状态
 - 有些情况下, 处理器在中断挂起时就会进行处理
 - 若中断发生但无法被立即处理, 则在其他中断处理结束前或中断屏蔽被清除前, 挂起请求会一直保持
 - ☒ 处理器已经在处理另一个更高或同等优先级的中断
 - ☒ 中断被屏蔽



NVIC 中断管理寄存器

- ISER: 中断使能寄存器 Interrupt Set Enable Register
 - ICER: 中断禁止 (清除使能) 寄存器
Interrupt Clear Enable Register
 - ISPR: 中断挂起寄存器 Interrupt Set Pending Register
 - ICPR: 清除中断挂起寄存器 Interrupt Clear Pending Register
- ❑ 中断处理完成后应该清除挂起, 表示已处理

表 7.9 用于中断控制的 NVIC 寄存器列表

地址	寄存器	CMSIS-Core 符号	功 能
0xE000E100~ 0xE000E11C	中断设置使能寄存器	NVIC->ISER [0]~ NVIC->ISER [7]	写 1 设置使能
0xE000E180~ 0xE000E19C	中断清除使能寄存器	NVIC->ICER [0]~ NVIC->ICER [7]	写 1 清除使能
0xE000E200~ 0xE000E21C	中断设置挂起寄存器	NVIC->ISPR [0]~ NVIC->ISPR [7]	写 1 设置挂起状态
0xE000E280~ 0xE000E29C	中断清除挂起寄存器	NVIC->ICPR [0]~ NVIC->ICPR [7]	写 1 清除挂起状态
0xE000E300~ 0xE000E31C	中断活跃位寄存器	NVIC->IABR [0]~ NVIC->IABR [7]	活跃状态位, 只读
0xE000E400~ 0xE000E4EF	中断优先级寄存器	NVIC->IP [0]~ NVIC->IR [239]	每个中断的中断优先级 (8 位宽)
0xE000EF00	软件触发中断寄存器	NVIC->STIR	写中断编号设置相应中断的挂起状态



中断管理寄存器



➤ IABR: 中断活跃位寄存器

- ❑ 每个外部中断都有一个活跃状态位，当处理器正在处理该中断时，该位会被置1
- ❑ 只读

➤ IPR: 中断优先级寄存器

➤ STIR: 软件触发中断寄存器

地址	寄存器	CMSIS-Core 符号	功 能
0xE000E280~ 0xE000E29C	中断清除挂起寄存器	NVIC->ICPR [0] ~ NVIC->ICPR [7]	写 1 清除挂起状态
0xE000E300~ 0xE000E31C	中断活跃位寄存器	NVIC->IABR [0]~ NVIC->IABR [7]	活跃状态位,只读
0xE000E400~ 0xE000E4EF	中断优先级寄存器	NVIC->IP [0]~ NVIC->IR [239]	每个中断的中断优先级(8 位宽)
0xE000EF00	软件触发中断寄存器	NVIC->STIR	写中断编号设置相应中断的挂起状态



STM32F407 中断和异常向量



➤ CM4内核:

- ❑ 支持256个中断，含16个内核中断和240个外部中断

➤ STM32F4:

- ❑ 只使用了10个内部异常和82个外部中断
- ❑ 只实现了Cortex-M4内核中断的部分功能



中断和异常向量表



编号	优先级	优先级类型	名称	说明	地址
	-	-	-	保留（实际存的是 MSP 地址）	0X0000 0000
	-3	固定	Reset	复位	0X0000 0004
	-2	固定	NMI	不可屏蔽中断。RCC 时钟安全系统 (CSS) 连接到 NMI 向量	0X0000 0008
	-1	固定	HardFault	所有类型的错误	0X0000 000C
	0	可编程	MemManage	存储器管理	0X0000 0010
	1	可编程	BusFault	预取指失败，存储器访问失败	0X0000 0014
	2	可编程	UsageFault	未定义的指令或非法状态	0X0000 0018
	-	-	-	保留	0X0000 001C- 0X0000 002B
	3	可编程	SVCall	通过 SWI 指令调用的系统服务	0X0000 002C
	4	可编程	Debug Monitor	调试监控器	0X0000 0030
	-	-	-	保留	0X0000 0034
	5	可编程	PendSV	可挂起的系统服务	0X0000 0038
	6	可编程	SysTick	系统嘀嗒定时器	0X0000 003C
0	7	可编程	-	窗口看门狗中断	0X0000 0040
1	8	可编程	PVD	连接 EXTI 线的可编程电压检测中断	0X0000 0044
2	9	可编程	TAMP_STAMP	连接 EXTI 线的入侵和时间戳中断	0X0000 0048
中间部分省略，详情请参考 STM32F4xx 中文参考手册》第十章-中断和事件-向量表部分					
79	86	可编程	CRYP	CRYP 加密全局中断	0X 0000 017C
80	87	可编程	HASH_RNG	哈希和随机数发生器全局中断	0X 0000 0180
81	88	可编程	FPU	FPU 全局中断	0X 0000 0184



中断的优先级



➤ STM32F4中断的优先级:

- ❑ 优先级越小优先级越高
- ❑ 3个固定优先级，都是负值，不能改变
- ❑ 16个可编程优先级，用4个bit表示

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
用于表达优先级				未使用，读回为 0			

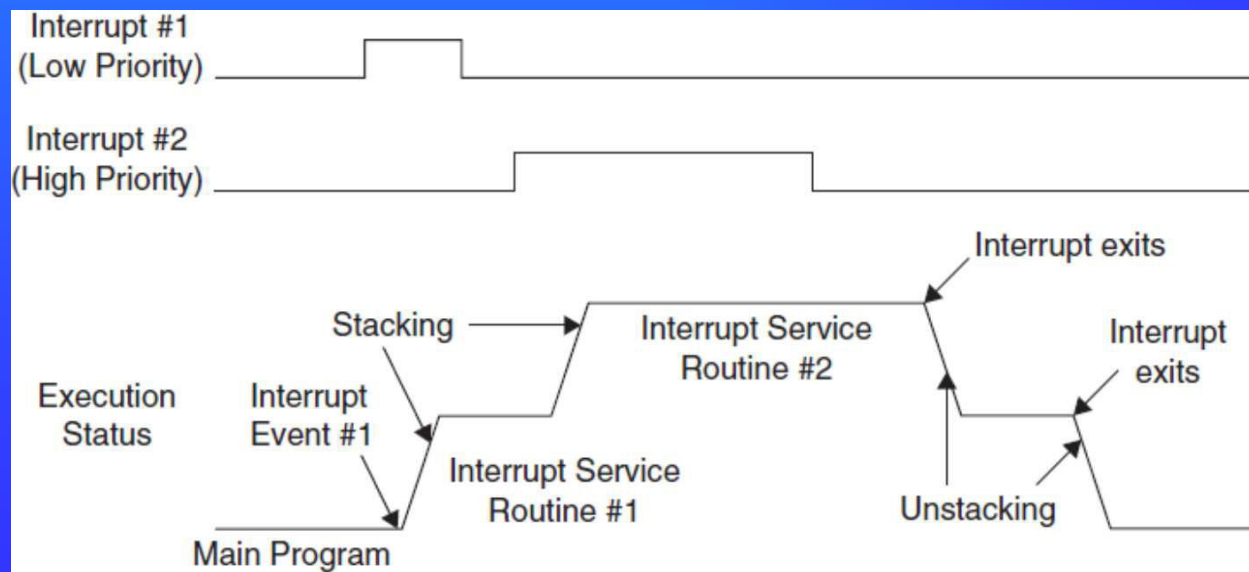
➤ NVIC中断优先级寄存器NVIC_IPRx

- ❑ F407: $x=0\dots81$
- ❑ 用于配置外部中断的优先级
- ❑ 宽度为8bit，原则上每个外部中断可配置的优先级为0~255，数值越小，优先级越高
 - ✉ F407: 只使用高4bit



中断优先级响应规则

- 抢占行为由抢占优先级决定：只有抢占优先级更高的中断源可以抢占当前正在响应的中断
- 当抢占优先级相同的中断有不只一个挂起时，最先响应子优先级最高的中断
- 如果优先级完全相同的多个中断同时挂起，则先响应中断编号最小的中断源



中断的优先级



➤ 中断优先级寄存器NVIC_IPRx中的4位，分为抢占优先级和响应优先级

- ❑ 抢占优先级在前，响应优先级（子优先级）在后
- ❑ 两个优先级各占几位根据SCB->AIRCRC寄存器的bit10~bit8中断分组PRIGROUP设置决定
 - ✉ STM32F4使用4bit标识每个中断的优先级

AIRCRC[10: 8]	bit[7: 4]分配情况	分配结果
111	0: 4	0 位抢占优先级, 4 位响应优先级
110	1: 3	1 位抢占优先级, 3 位响应优先级
101	2: 2	2 位抢占优先级, 2 位响应优先级
100	3: 1	3 位抢占优先级, 1 位响应优先级
011	4: 0	4 位抢占优先级, 0 位响应优先级



STM32实现的中断优先级寄存器

应用程序中断及复位控制器SCB_AIRCR

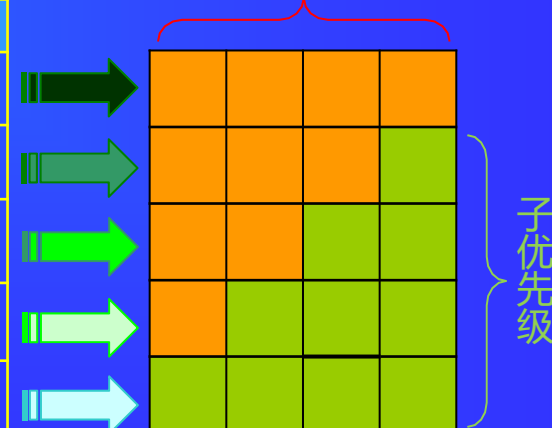
地址: 0xE000_ED0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VECTKEYSTAT[15:0](read)/ VECTKEY[15:0](write)															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENDIANESS	Reserved				PRIGROUP			Reserved					SYS RESET REQ	VECT CLR ACTIVE	VECT RESET
													w	w	w

STM32只用
中断优先级
寄存器高4位

PRIGROUP (3 Bits)	Binary Point (group. sub)		Preempting Priority (Group Priority)		Sub-Priority	
			Bits	Levels	Bits	Levels
011	4.0	gggg	4	16	0	0
100	3.1	gggs	3	8	1	2
101	2.2	ggss	2	4	2	4
110	1.3	gsss	1	2	3	8
111	0.4	ssss	0	0	4	16

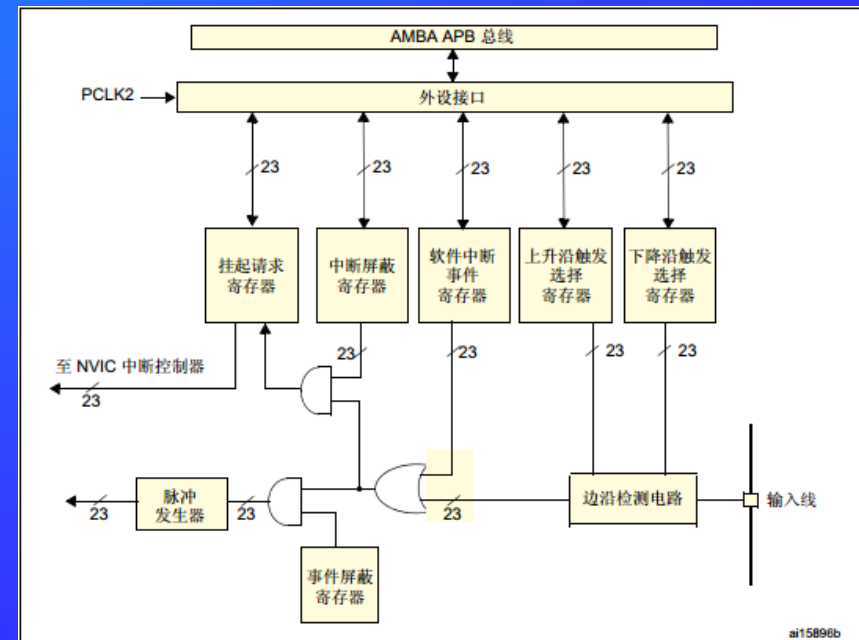
抢占优先级



External interrupt/event controller (EXTI)

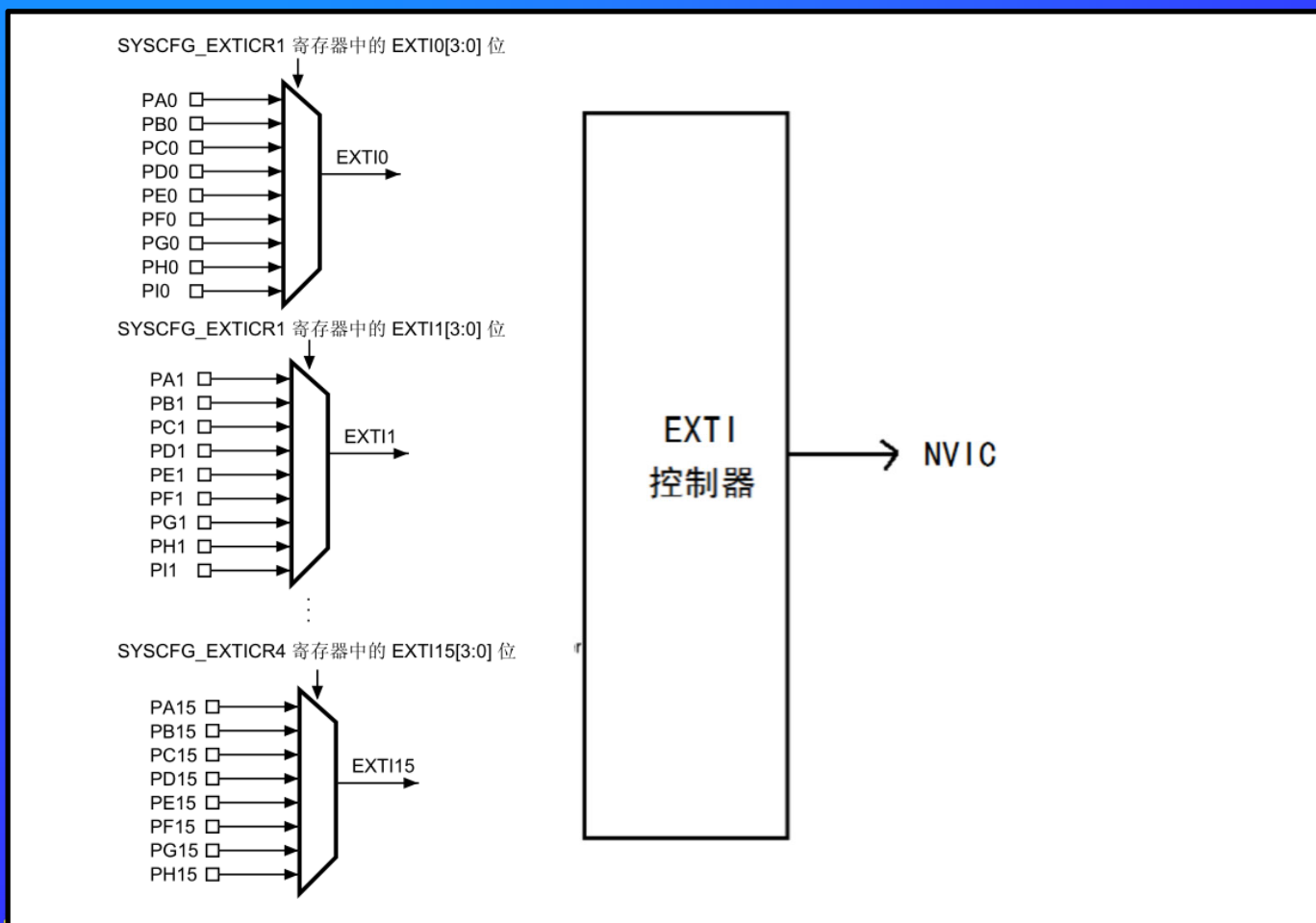
- 支持多达23个边沿检测器用于产生事件/中断请求
- 每根中断/事件线都可单独配置其类型（interrupt or event）和触发事件（rising or falling or both）
 - ❑ 中断：输入信号输入到NVIC中触发中断服务程序
 - ❑ 事件：输入一个脉冲信号给其他外设使用
- ✉ 例：给定时器TIMER或ADC使用

- 每根中断/事件线都可单独屏蔽
- 挂起寄存器可以记录中断请求线的状态
- GPIO口连接到16根外部中断/事件线



STM32 中断系统

- STM32F4xx将多达140个GPIO通过以下方式连接到16个外部中断/事件线:



系统配置控制器 (SYSCFG)

➤ 一组系统配置控制器，主要用途：

- ❑ 重映射存储器到代码起始区域
- ❑ 管理连接到GPIO口的外部中断
- ❑ 管理系统的可靠性特性

➤ SYSCFG外部中断配置寄存器1-4 (SYSCFG_EXTICR1—4)

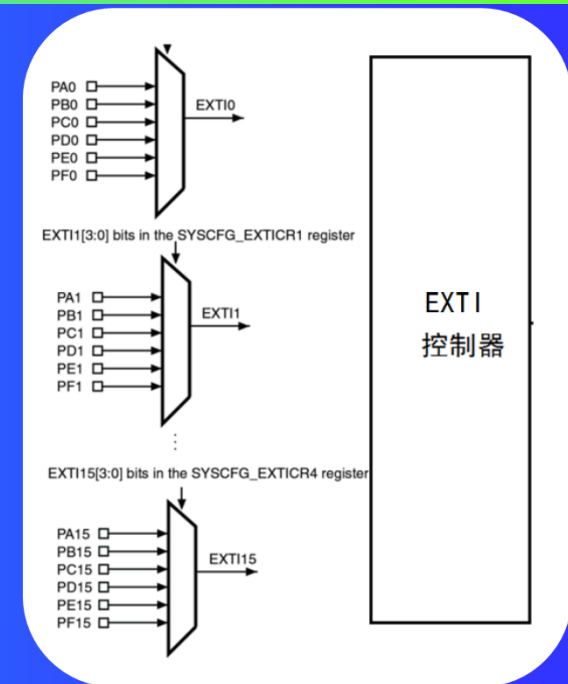
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **EXTIx[3:0]**: EXTI x 配置 (x = 0 到 3) (EXTI x configuration (x = 0 to 3))

这些位通过软件写入，以选择 **EXTIx** 外部中断的源输入。

- 0000: PA[x] 引脚
- 0001: PB[x] 引脚
- 0010: PC[x] 引脚
- 0011: PD[x] 引脚
- 0100: PE[x] 引脚
- 0101: PF[C] 引脚
- 0110: PG[x] 引脚
- 0111: PH[x] 引脚
- 1000: PI[x] 引脚



STM32 中断系统



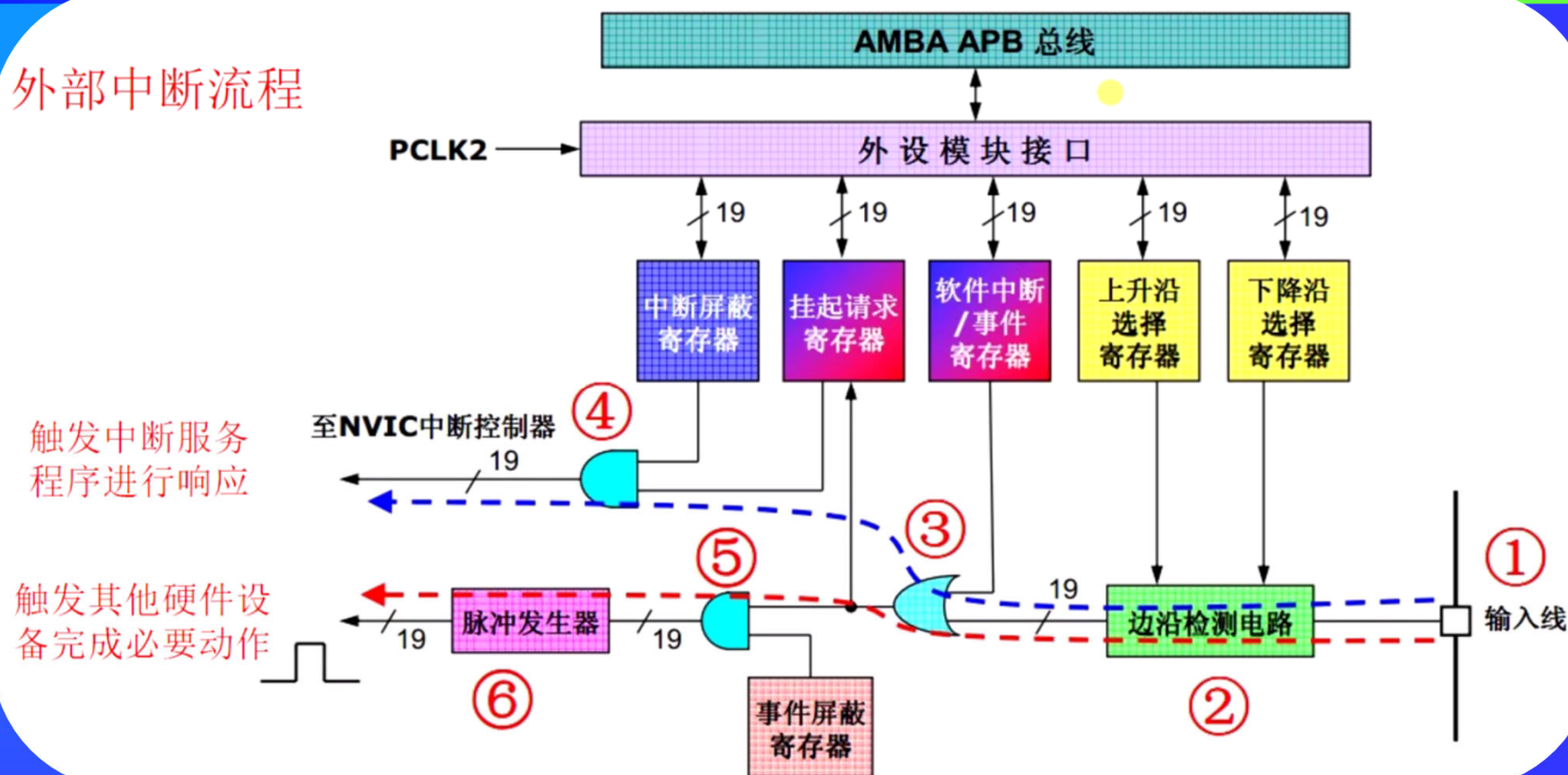
➤ EXIT支持多达23个事件/中断请求，除16跟GPIO中断外，另外七根EXTI线连接方式如下：

- ❑ EXTI线16连接到PVD输出
- ❑ EXTI线17连接到RTC闹钟事件
- ❑ EXTI线18连接到USB OTG FS唤醒事件
- ❑ EXTI线19连接到以太网唤醒事件
- ❑ EXTI线20连接到USB OTG HS唤醒事件
- ❑ EXTI线21连接到RTC入侵和时间戳事件
- ❑ EXTI线22连接到RTC唤醒事件



外部中断 / 事件框图

外部中断流程



- 软件向软件中断/事件寄存器 (software interrupt/event register) 写“1”也可以产生中断/事件请求



本章结束

