

فصل هشتم : Interrupts

Why Interrupt?!

- با یک مثال نحوه عملکرد وقفه را توضیح می‌دهیم :
- فرض کنید تلفن قابلیت تولید صدا هنگام دریافت تماس را نداشت. در این صورت شما بایستی به منظور از دست ندادن تماس دریافتی، پشت سر هم گوشی تلفن را بردارید و چک نمایید که آیا تماسی دریافت کرده‌اید یا خیر! (Polling mode) اما اگر تلفن قابلیت تولید صدا یا هرگونه هشدار را به هنگام دریافت تماس دارا بود، شما می‌توانستید به سایر کارهای خود رسیدگی نمایید و تنها زمانی که تماس دریافتی داشتید با آن رسیدگی نمایید.
- در واقع وقفه، یک عملیات نرم‌افزاری یا سخت‌افزاری است که پردازنده را مجبور به توقف فعالیتش می‌کند تا یک کد مشخص به نام Interrupt Service Routine (ISR) را اجرا نماید. پس از اجرای ISR، پردازنده مجدد فعالیت قبلی خود را از سر می‌گیرد.

Preemptive scenario and Non-preemptive scenario

Preemptive scenario ▪

- در این حالت اگر وقفه‌ای رخ داده باشد و پردازنده در حال اجرای آن وقفه باشد که وقفه دیگری با اولویت بالاتر رخ دهد، پردازنده وقفه قبلی را متوقف نموده و ابتدا وقفه با اولویت بالاتر را اجرا می‌کند. سپس عملیات مربوط به وقفه اولیه را به اتمام رسانده و در انتها به فعالیت عادی خود ادامه می‌دهد.

Non-Preemptive scenario ▪

- در این حالت وقفه جدید نمی‌تواند عملیات وقفه قبل را متوقف نماید مگر اینکه خود وقفه، کنترل پردازنده را واگذار نماید. معمولاً این سناریو در تایمرها استفاده می‌شود که چندین تایمر با وقفه‌های مشخص قصد انجام متناوب یکسری عملیات را دارند.

Interrupt numbers?

- در پردازنده‌های Cortex-M تا 256 گونه وقفه وجود دارد. هر وقفه دارای یک عدد اختصاصی است که بازه آن بین 15- تا 240 می‌باشد این اعداد معمولاً ثابت و توسط شرکت سازنده ارائه می‌شود و در حالت کلی به دو دسته تقسیم می‌شوند :
- 16 مورد اول که اعداد آن‌ها نیز منفی می‌باشند وقفه‌های سیستمی می‌باشند که به آن‌ها System exceptions نیز می‌گویند
- بقیه 240 مورد دیگر مربوط به وقفه‌های Peripheral می‌باشند که به آن‌ها non-system exceptions نیز می‌گویند. این دسته از وقفه‌ها با عدد 0 شروع می‌شوند.
- توضیحات فوق استاندارد کلی وقفه‌ها برای پردازنده‌های Cortex-M را بیان می‌کند ولی ممکن است در برخی چیپ‌ها، شرکت سازنده از اعداد متفاوتی استفاده کرده باشد که در دیتاشیت آن بیان شده است.

Interrupt Service Routines

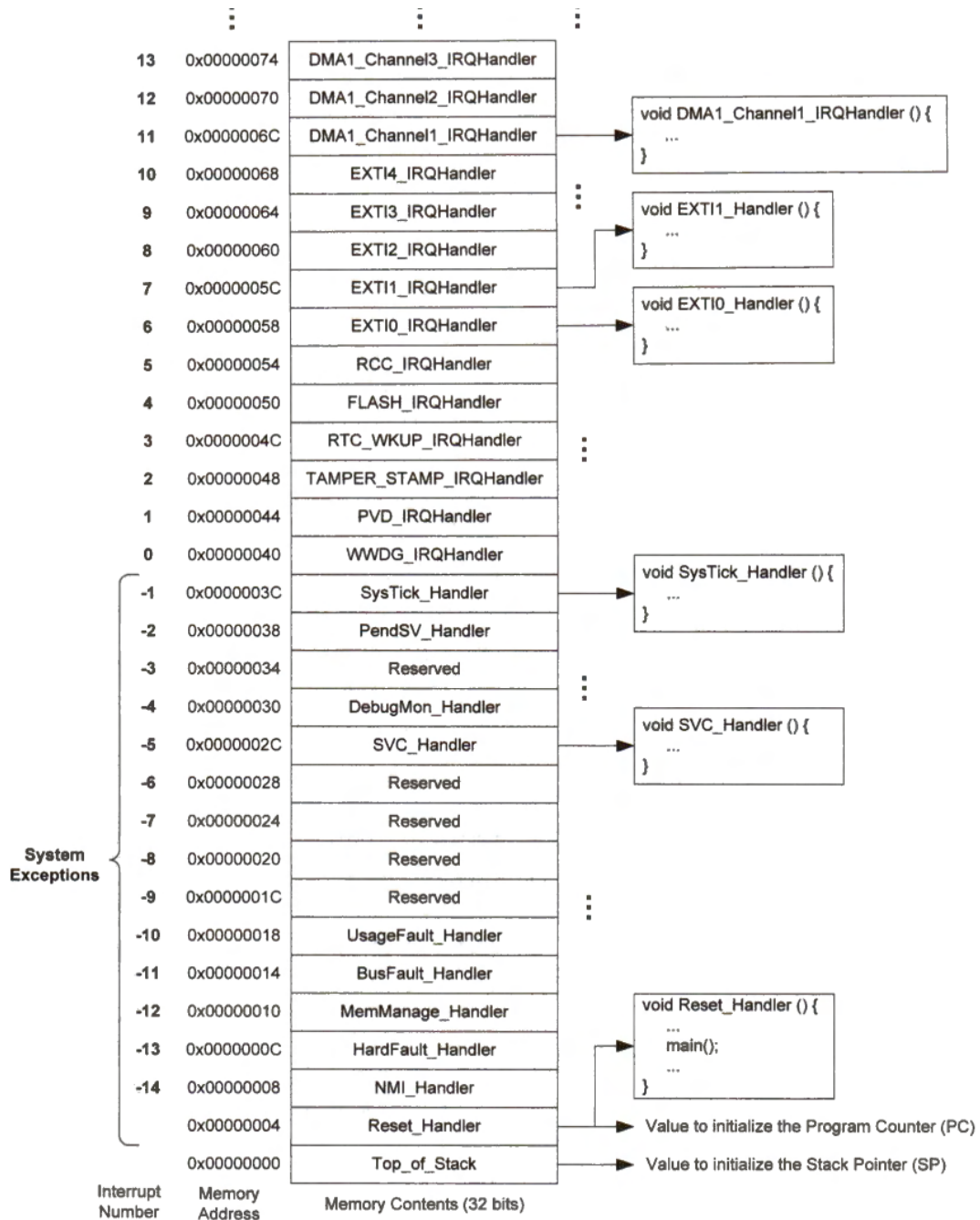
- همان برنامه‌ای است که هنگام وقوع وقفه توسط سخت‌افزار فراخوانی می‌شود. هر ISR ساختار پیش‌فرضی دارد که در فایل Startup بیان شده است. برای مثال شکل زیر ساختار ISR مربوط SysTick را نشان می‌دهد.

```
void SysTick_Handler (void) {  
    ...  
}
```

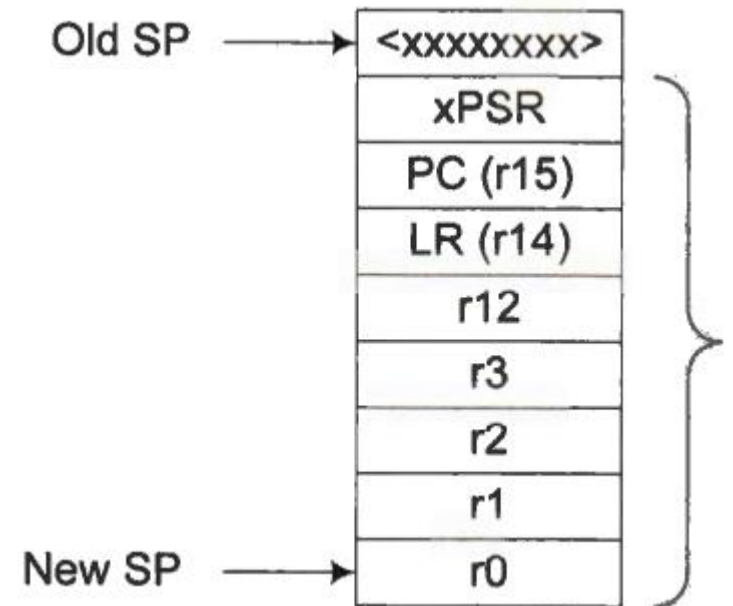
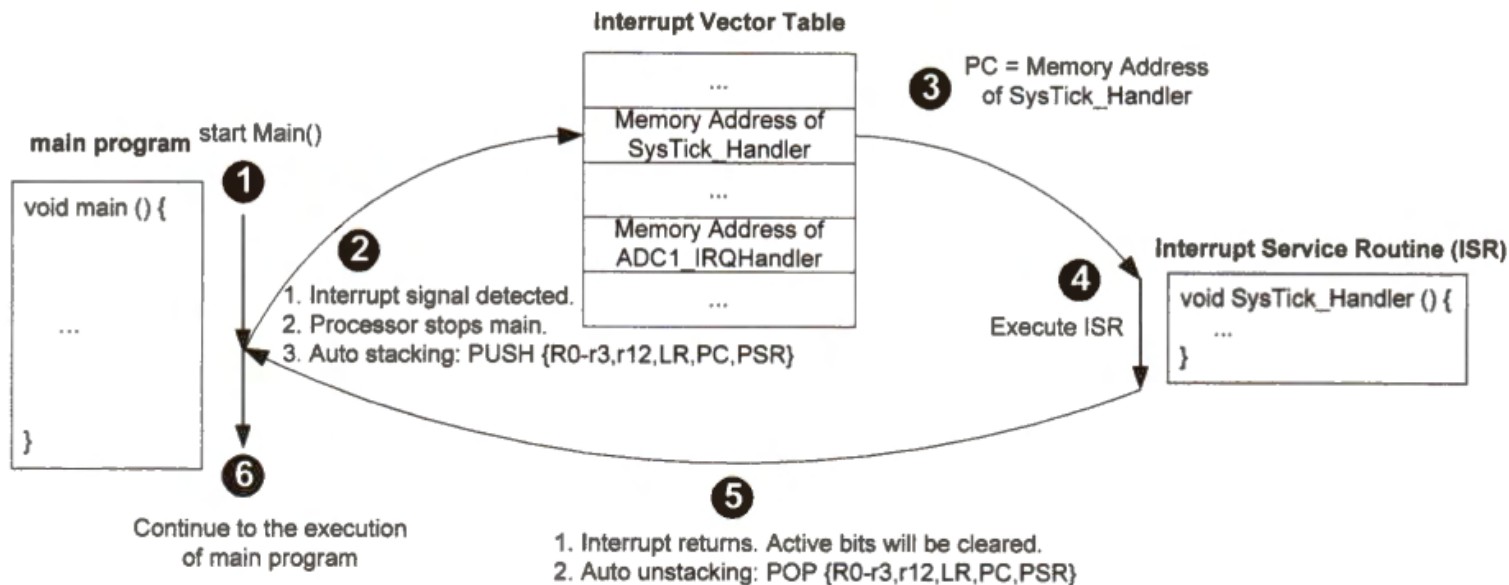
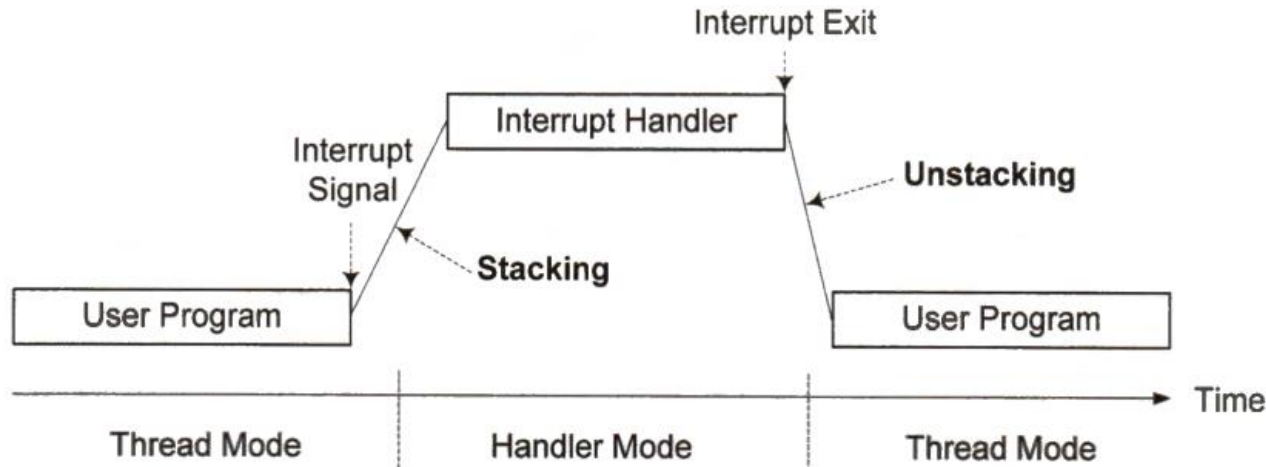
- وقفه Reset که از نوع Non-Maskable است تابع Main را فراخوانی می‌کند.

Interrupt Vector Table

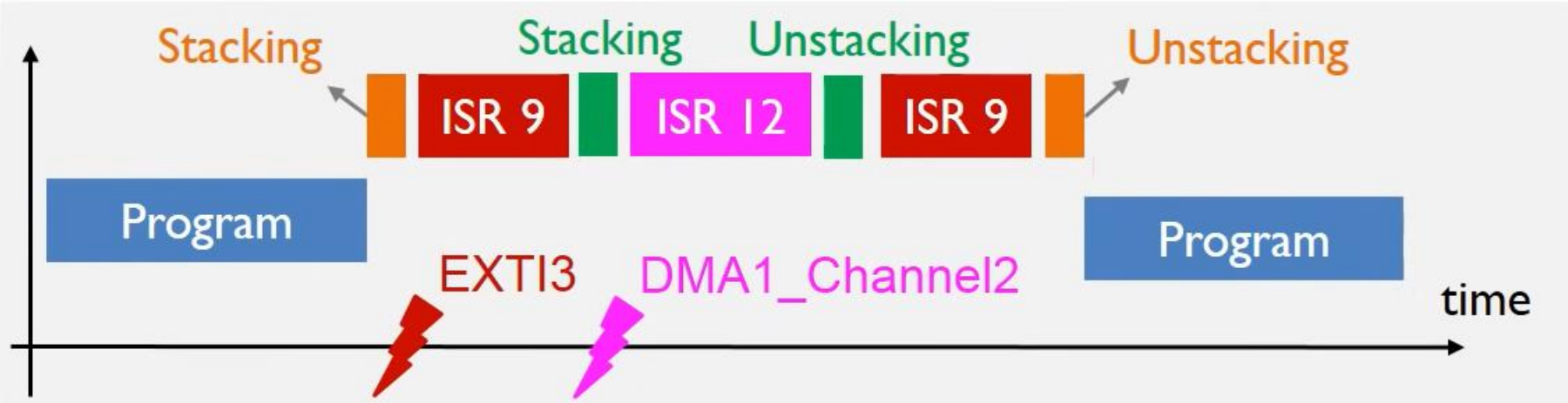
- توجه شود که در خانه‌های فوق، صرفاً آدرس شروع تابع ISR متناظر با هر تابع نوشته شده است و به آدرسی اشاره می‌کند که قرار ISR متناظر با آن قرار دارد.



Interrupt Stacking and Unstacking



Nested Vectored Interrupt Controller (NVIC)



	DMA1_Channel2			EXTI3	
Interrupt Number	12	11	10	9	8
Enable Register	1	0	0	1	0
Active Register	0	0	0	0	0
Pending Register	0	0	0	0	0
Priority Register	3	4	7	5	3

XXXXXXXX
xPSR
PC(r15)
LR(r14)
r12
r3
r2
r1
r0
xPSR
PC(r15)
LR(r14)
r12
r3
r2
r1
r0

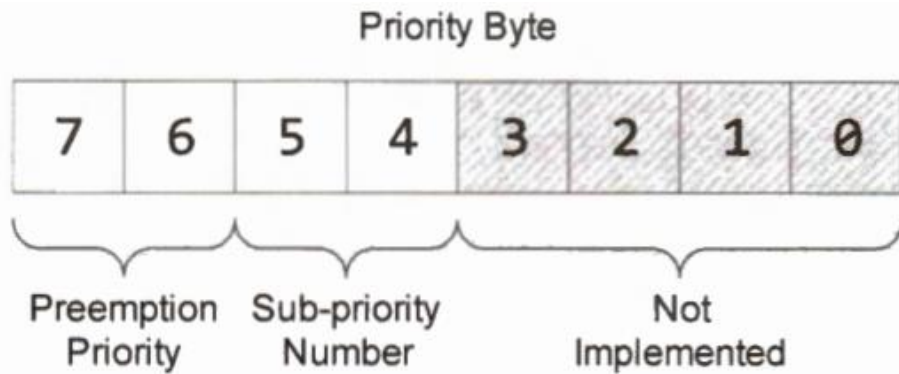
SP →

Interrupt Priority

- با توجه به تعریف 256 وقفه نیاز به 8 بیت برای نمایش اولویت وقفه وجود دارد.
- با توجه به توانایی‌های پردازنده، تعداد وقفه‌های قابل تعریف برای آن مشخص می‌شود (برای مثال در STM32F10x این تعداد عدد 16 می‌باشد که برای نمایش به 4 بیت نیاز دارد.
- قابلیت تقسیم وقفه‌ها به دو دسته زیر نیز وجود دارد :

Preemption Priority ▪

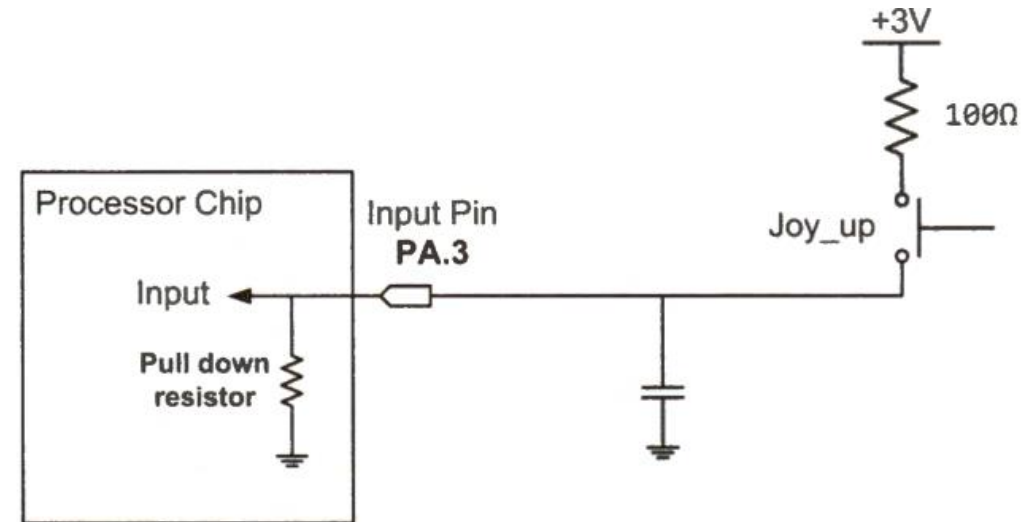
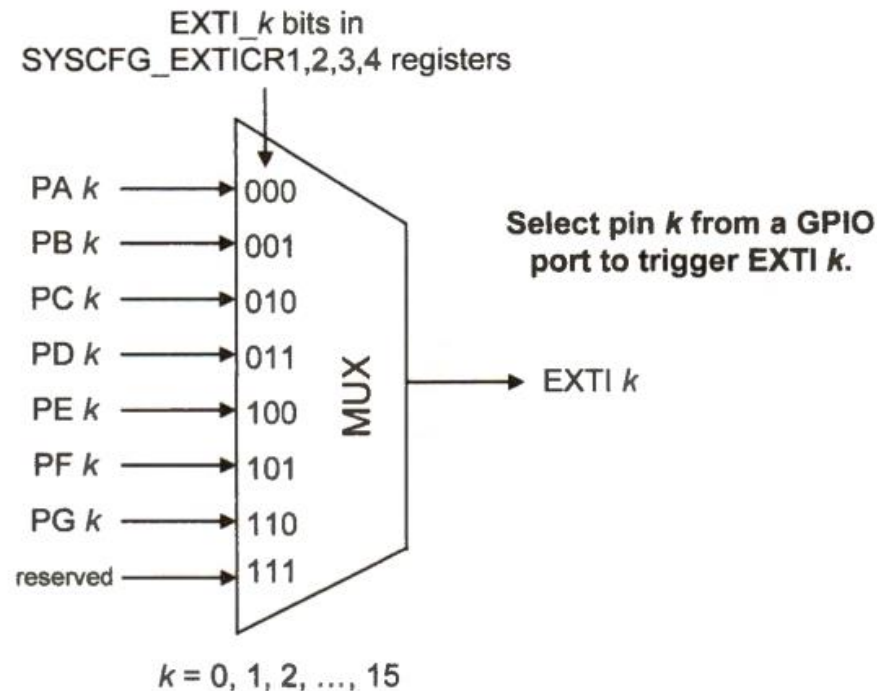
Sub-priority ▪



- توصیه می‌شود وقفه‌ها در حالت Preemptive قرار دهید. چرا!!؟

External Interrupts

- علاوه بر وقفه‌های داخلی، قابلیت تعریف وقفه به صورت خارجی نیز وجود دارد.
- در بسیاری از کاربردها، نیاز است تا پس از آماده شدن دیتای یک سنسور، سیگنالی از سمت آن صادر گردد تا پردازنده به وضعیت آن رسیدگی نماید. در این شرایط می‌توان آن پایه را به یکی از پایه‌های میکرو و به صورت وقفه خارجی متصل نمود. در این حالت با تغییر وضعیت پایه، وقفه‌ای برای پردازنده ارسال می‌گردد.



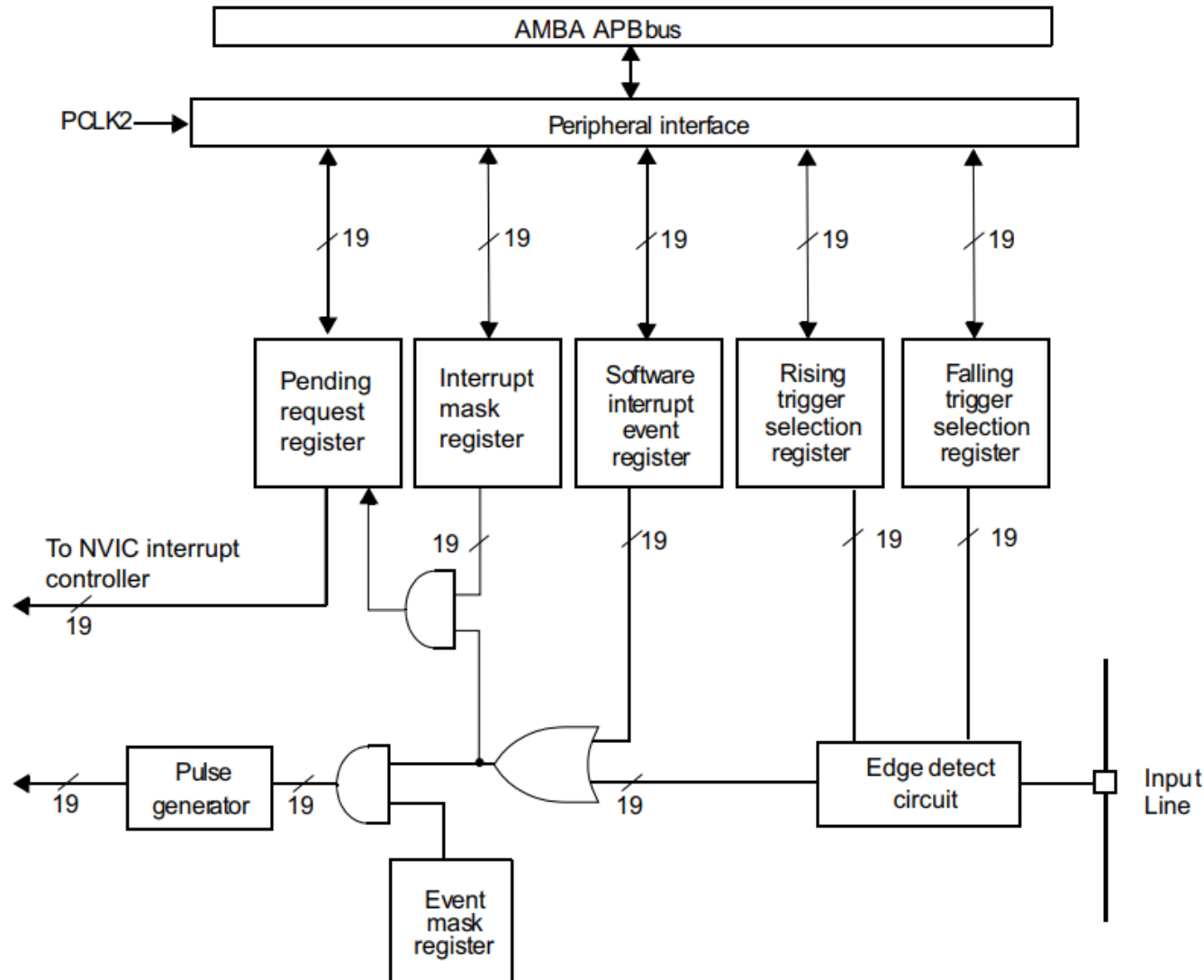
STM32F10x NVIC

Position	Priority	Type of priority	Acronym	Description	Address
-	-	-	-	Reserved	0x0000_0000
-	-3	fixed	Reset	Reset	0x0000_0004
-	-2	fixed	NMI	Nonmaskable interrupt. The RCC Clock Security System (CSS) is linked to the NMI vector.	0x0000_0008
-	-1	fixed	HardFault	All class of fault	0x0000_000C
-	0	settable	MemManage	Memory management	0x0000_0010
-	1	settable	BusFault	Prefetch fault, memory access fault	0x0000_0014
-	2	settable	UsageFault	Undefined instruction or illegal state	0x0000_0018
-	-	-	-	Reserved	0x0000_001C-0x0000_002B
-	3	settable	SVCall	System service call via SWI instruction	0x0000_002C
-	4	settable	Debug Monitor	Debug monitor	0x0000_0030
-	-	-	-	Reserved	0x0000_0034
-	5	settable	PendSV	Pendable request for system service	0x0000_0038
-	6	settable	SysTick	Systick timer	0x0000_003C
0	7	settable	WWDG	Window watchdog interrupt	0x0000_0040
1	8	settable	PVD	PVD through EXTI Line detection interrupt	0x0000_0044
2	9	settable	TAMPER	Tamper interrupt	0x0000_0048
3	10	settable	RTC	RTC global interrupt	0x0000_004C
4	11	settable	FLASH	Flash global interrupt	0x0000_0050

63	70	settable	CAN2_TX	CAN2 TX interrupts	0x0000_013C
64	71	settable	CAN2_RX0	CAN2 RX0 interrupts	0x0000_0140
65	72	settable	CAN2_RX1	CAN2 RX1 interrupt	0x0000_0144
66	73	settable	CAN2_SCE	CAN2 SCE interrupt	0x0000_0148
67	74	settable	OTG_FS	USB On The Go FS global interrupt	0x0000_014C

- 68 مدل وقفه را پشتیبانی می‌کند.
- دارای 4 بیت برای تعریف اولویت می‌باشد.

STMF10x External Interrupt Controller



■ تفاوت Interrupt و Event؟!