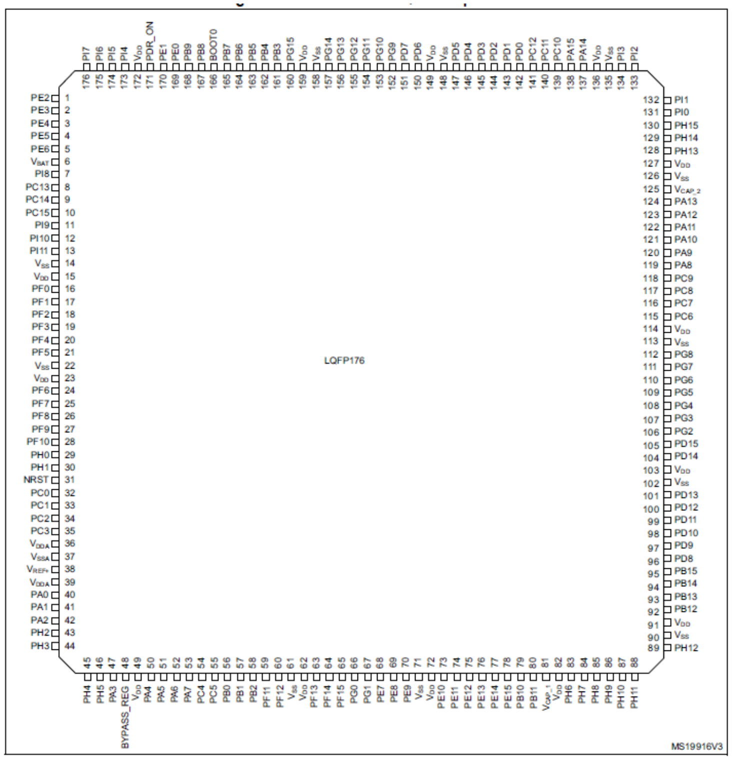
STM32F407 EXCEPTION



한국산업기술대학교 메카트로닉스공학과
마이크로컴퓨터구조
담당교수: 남윤석

인터럽트 및 예외

●구조: 내부에 **NVIC(Nested Vectored Interrupt Controller : 중첩 벡터형 인터럽트 제어기)**가 각종 인터럽트의 효율적인 제어 수행

●예외(**Exceptions)**

: 프로세서가 동작하는 도중 예상하지 못한 상황이나, 예외적인 상황 또는 외부 인터럽트가 발생하는 경우를 지칭

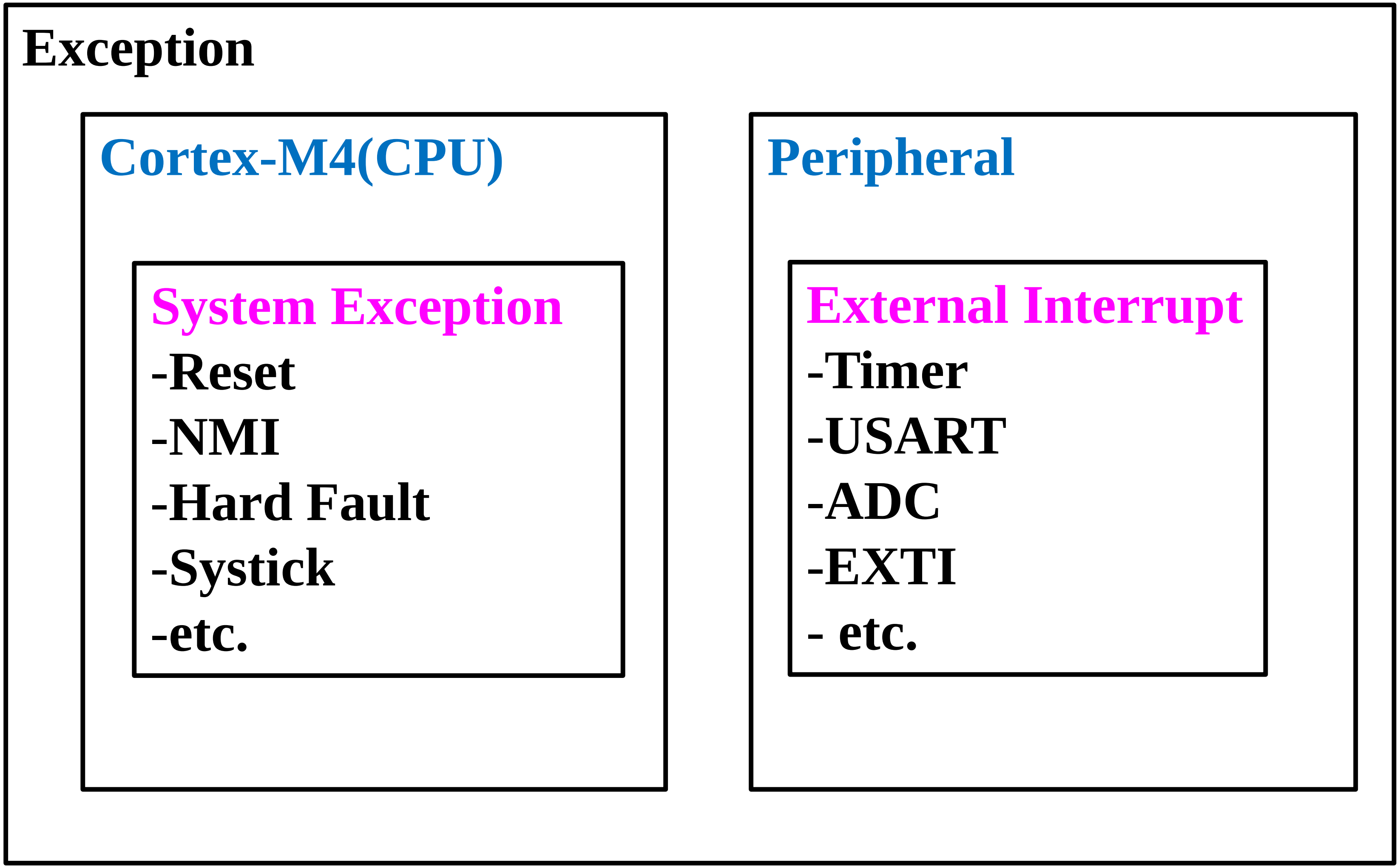
•시스템 예외(**System Exceptions): **프로세서의 내부에서 발생

•외부 인터럽트(**External Interrupts)**: 그 외의 주변장치에서 발생

"예외 = 시스템 예외 + 외부 인터럽트"

• **Exception**이 발생하면 프로세서는 **Thread mode**에서 **Handler mode**로 진입하고, **ISR(Interrupt Service Routine)**이 종료되면 다시 **Thread mode**로 복귀함

인터럽트 및 예외



Exception

Cortex-M4(CPU)

System Exception
-Reset
-NMI
-Hard Fault
-Systick
-etc.

Peripheral

External Interrupt
-Timer
-USART
-ADC
-EXTI
- etc.

●NVIC의 인터럽트 처리

: NVIC는 모든 인터럽트(예외)에 대한 우선 순위를 결정하고 처리

•NVIC의 인터럽트 처리 순서

① 스택킹(**Stacking**) : 인터럽트 처리에 사용될 레지스터의 현재 값을 스택에 저장 (**xPSR, PC, LR, R12, R3,R2,R1,R0** 차례로 자동저장)

② 벡터 인출(**Vector Fetch**) : 벡터테이블에서 인터럽트 핸들러의 시작 주소 **Fetch**

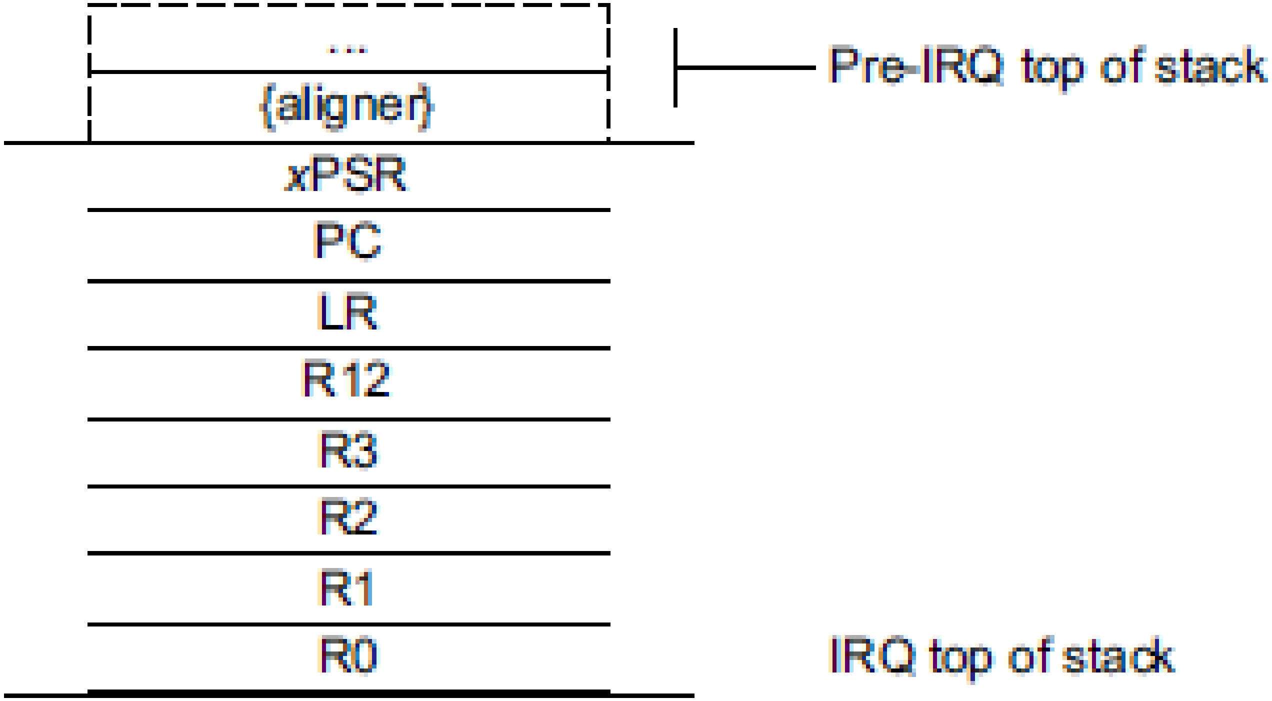
③ 레지스터 업데이트 : 인터럽트 핸들러로 진입할 때 관련된 레지스터가 업데이트

④ 인터럽트 서비스 루틴 실행 : 발생된 인터럽트에 해당되는 인터럽트 서비스 루틴을 실행

⑤ 인터럽트 종료/빠져나오기 : 인터럽트 서비스 루틴의 실행이 완료되면 인터럽트를 종료하기 위해 ①단계에서 스택에 넣었던 내용을 다시 읽어와 레지스터를 원래의 값으로 복구(**UnStacking**)

✓인터럽트 핸들러 (**Interrupt Handler**) : 어떤 인터럽트가 발생했을 때 이에 대응하는 인터럽트 서비스 루틴(**ISR**)으로 연결해주는 역할을 하는 명령. 여기에는 발생된 특정 인터럽트에 대응하는 **ISR**의 주소가 저장되어 있음

- **Stacking후 stack 상태**



•중첩 인터럽트 (Nested Interrupt) 동작

-정의: 현재 실행중인 인터럽트보다 우선 순위가 높은 인터럽트가 발생할 경우, 현재의 동작을 멈추고 높은 순위의 인터럽트를 먼저 처리하는 기능 (이 경우 먼저 실행 중이던 인터럽트는 우선 순위가 높은 인터럽트가 끝나면 다시 실행)

-NVIC의 역할: 미리 정해진 인터럽트의 우선 순위에 따라 자동적으로 중첩 인터럽트 동작을 수행 (이 경우 레지스터의 스택킹과 언스택킹은 자동으로 처리되어 이전의 데이터를 잃어버릴 위험없이 프로그램이 실행되도록 해줌)

•벡터형 인터럽트 (Vectored Interrupt) 동작

-정의: 벡터 테이블(ISR의 주소 저장)을 이용하면 인터럽트의 발생시에 별도의 소프트웨어가 필요없이 바로 대응되는 ISR의 시작 주소를 알 수 있으며 이를 벡터형 인터럽트 동작

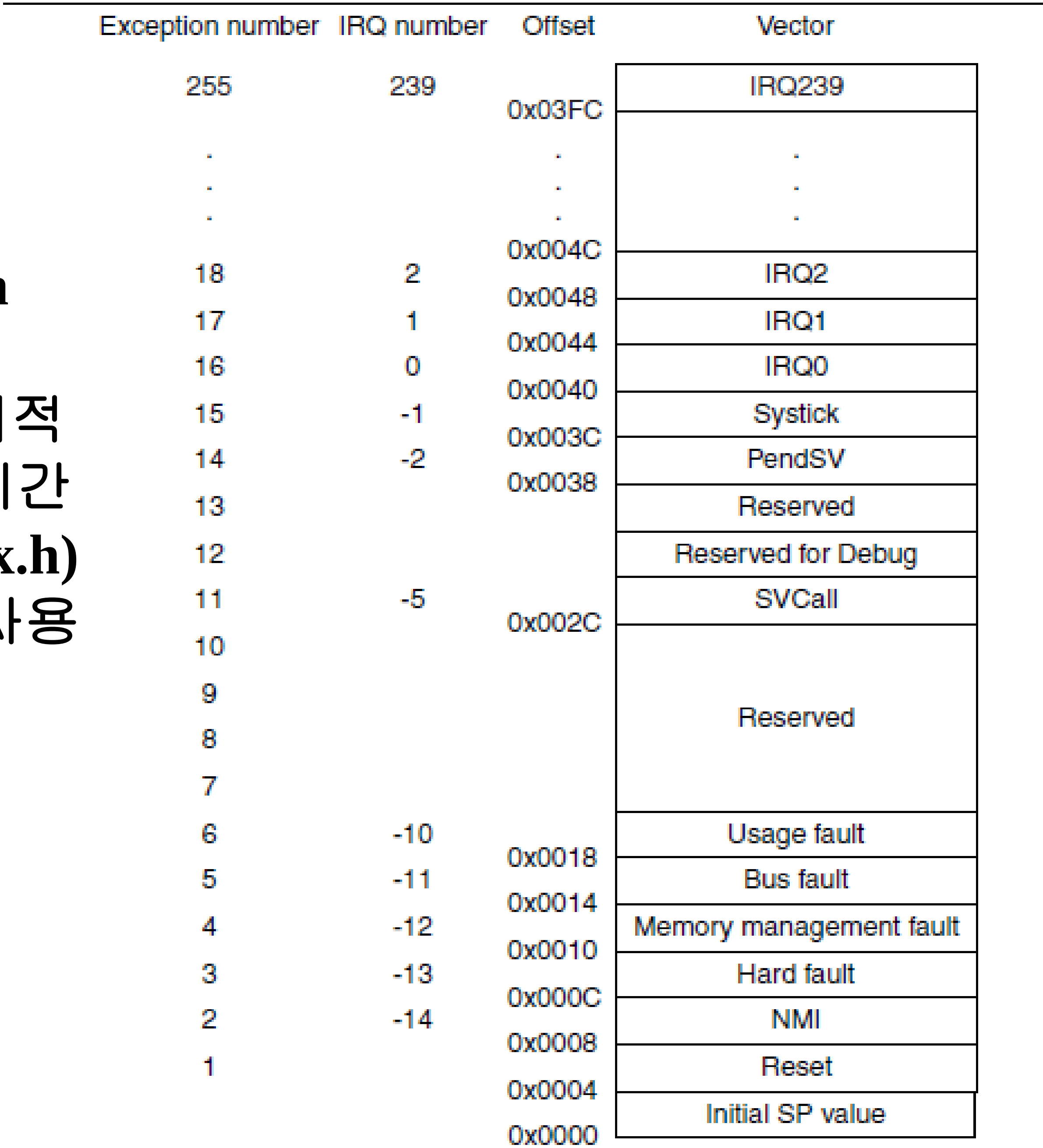
-효과: 인터럽트 처리 속도 증가

- 벡터테이블 위치

-Flash(0x0800.0000)
or SRAM(0x2000.0000)

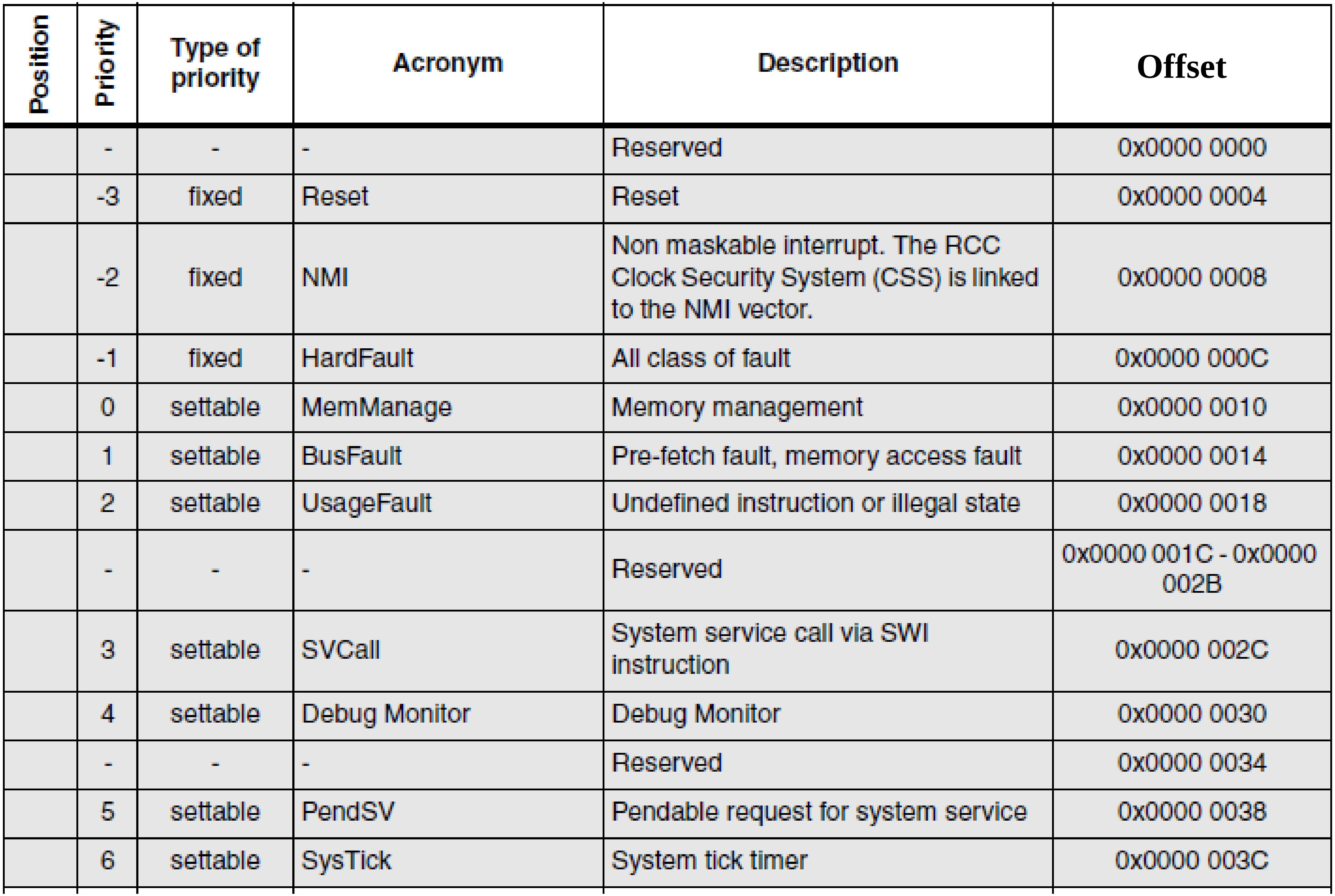
- 벡터테이블 의 **Exception numbering(priority)**는 사용자에 의해 소프트웨어적으로 변경가능(단, 실습시간에는 헤더파일**(stm32f4xxx.h)**에서 정한대로 변경없이 사용함)

Figure 11. Vector table



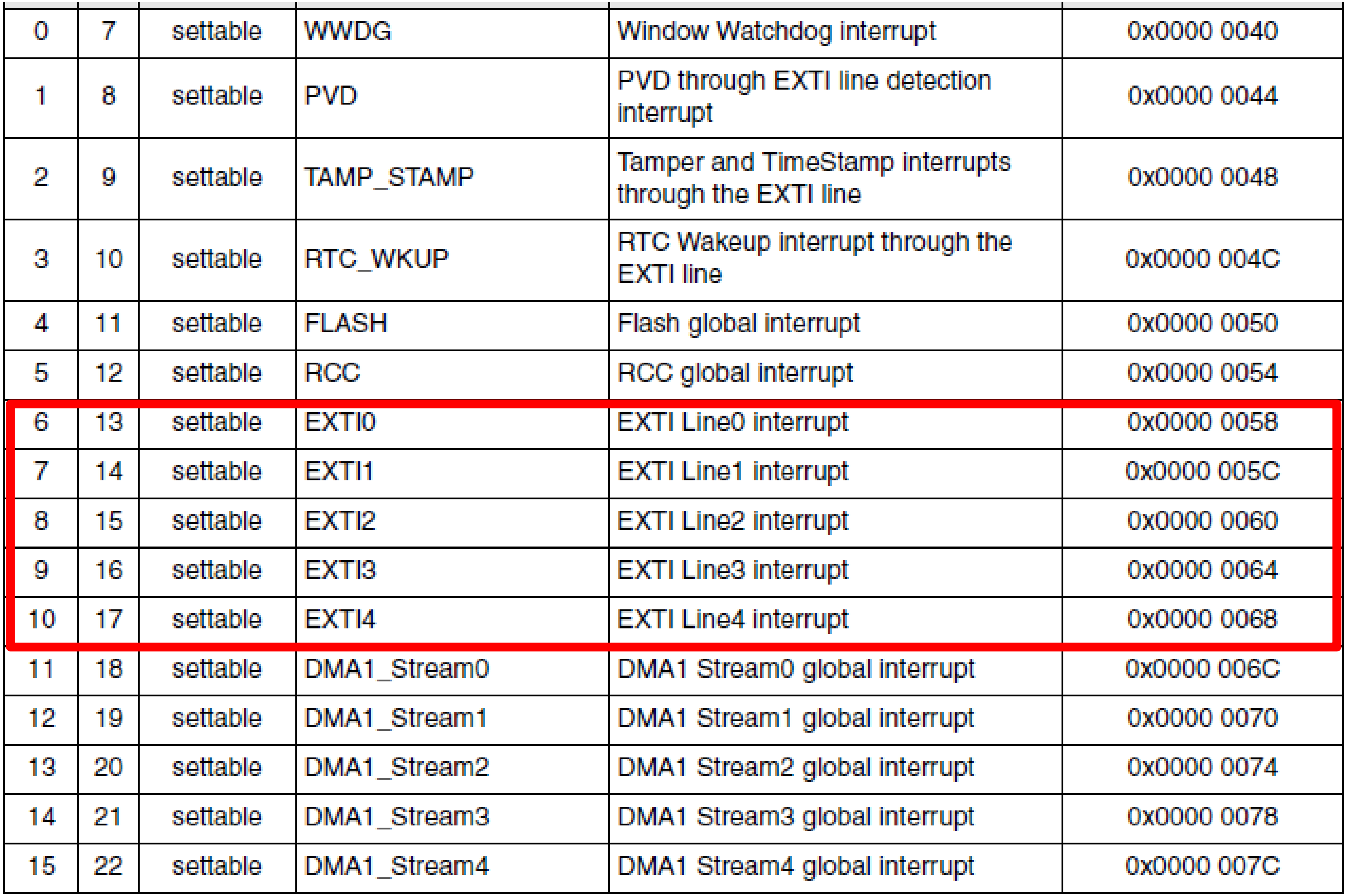
Exception number	IRQ number	Offset	Vector
255	239	0x03FC	IRQ239
. . .		. . . 0x004C	. . .
18	2	0x0048	IRQ2
17	1	0x0044	IRQ1
16	0	0x0040	IRQ0
15	-1	0x003C	Systick
14	-2	0x0038	PendSV
13			Reserved
12			Reserved for Debug
11	-5	0x002C	SVCall
10			Reserved
9			
8			
7			
6	-10	0x0018	Usage fault
5	-11	0x0014	Bus fault
4	-12	0x0010	Memory management fault
3	-13	0x000C	Hard fault
2	-14	0x0008	NMI
1		0x0004	Reset
		0x0000	Initial SP value

●인터럽트 벡터 테이블- System Exceptions(10개)



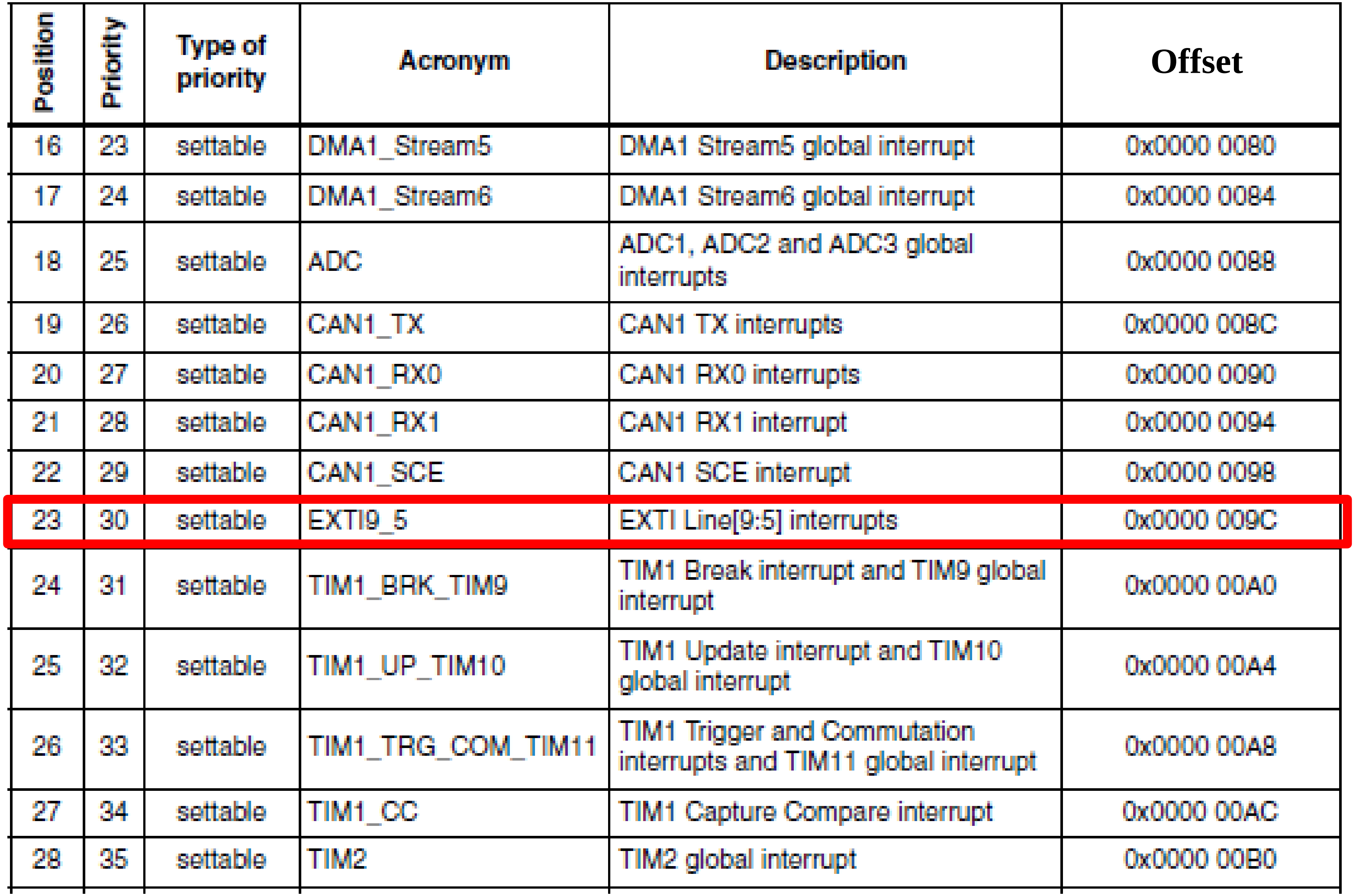
Position	Priority	Type of priority	Acronym	Description	Offset
	-	-	-	Reserved	0x0000 0000
	-3	fixed	Reset	Reset	0x0000 0004
	-2	fixed	NMI	Non maskable interrupt. The RCC Clock Security System (CSS) is linked to the NMI vector.	0x0000 0008
	-1	fixed	HardFault	All class of fault	0x0000 000C
	0	settable	MemManage	Memory management	0x0000 0010
	1	settable	BusFault	Pre-fetch 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	System tick timer	0x0000 003C

●인터럽트 벡터 테이블(총82개)- Interrupts (0~15)



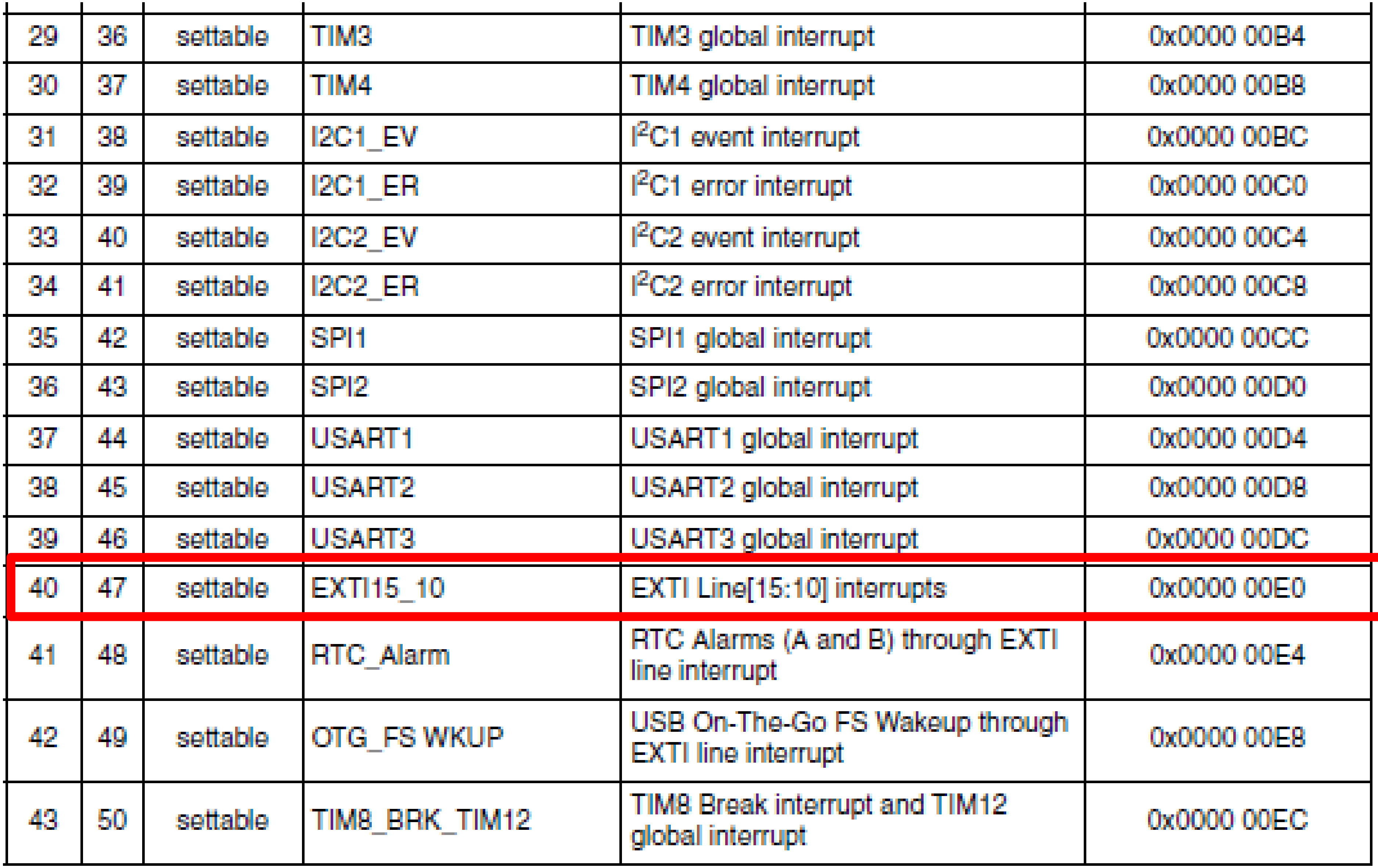
0	7	settable	WWDG	Window Watchdog interrupt	0x0000 0040
1	8	settable	PVD	PVD through EXTI line detection interrupt	0x0000 0044
2	9	settable	TAMP_STAMP	Tamper and TimeStamp interrupts through the EXTI line	0x0000 0048
3	10	settable	RTC_WKUP	RTC Wakeup interrupt through the EXTI line	0x0000 004C
4	11	settable	FLASH	Flash global interrupt	0x0000 0050
5	12	settable	RCC	RCC global interrupt	0x0000 0054
6	13	settable	EXTI0	EXTI Line0 interrupt	0x0000 0058
7	14	settable	EXTI1	EXTI Line1 interrupt	0x0000 005C
8	15	settable	EXTI2	EXTI Line2 interrupt	0x0000 0060
9	16	settable	EXTI3	EXTI Line3 interrupt	0x0000 0064
10	17	settable	EXTI4	EXTI Line4 interrupt	0x0000 0068
11	18	settable	DMA1_Stream0	DMA1 Stream0 global interrupt	0x0000 006C
12	19	settable	DMA1_Stream1	DMA1 Stream1 global interrupt	0x0000 0070
13	20	settable	DMA1_Stream2	DMA1 Stream2 global interrupt	0x0000 0074
14	21	settable	DMA1_Stream3	DMA1 Stream3 global interrupt	0x0000 0078
15	22	settable	DMA1_Stream4	DMA1 Stream4 global interrupt	0x0000 007C

●인터럽트 벡터 테이블- Interrupts (16~28)



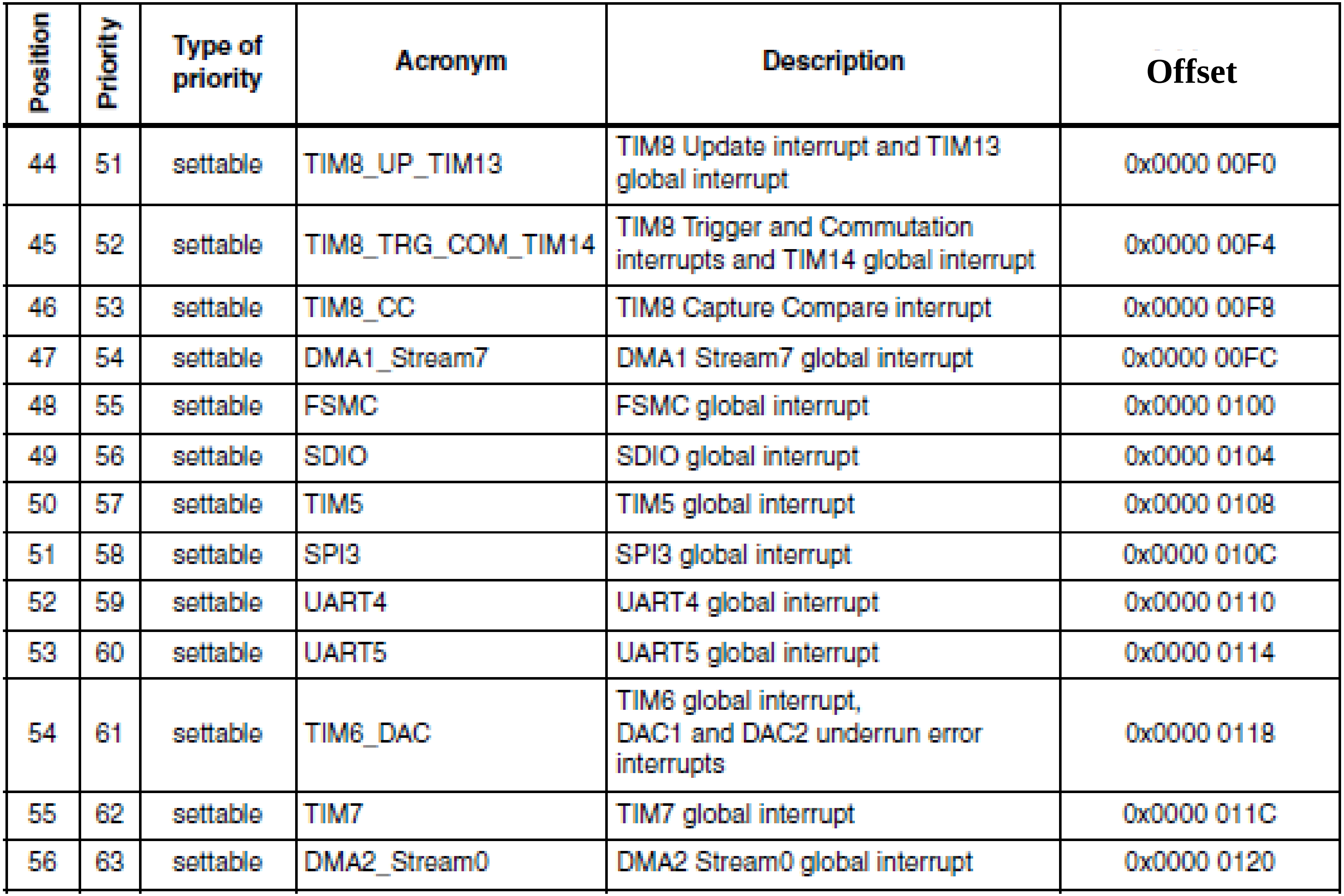
Position	Priority	Type of priority	Acronym	Description	Offset
16	23	settable	DMA1_Stream5	DMA1 Stream5 global interrupt	0x0000 0080
17	24	settable	DMA1_Stream6	DMA1 Stream6 global interrupt	0x0000 0084
18	25	settable	ADC	ADC1, ADC2 and ADC3 global interrupts	0x0000 0088
19	26	settable	CAN1_TX	CAN1 TX interrupts	0x0000 008C
20	27	settable	CAN1_RX0	CAN1 RX0 interrupts	0x0000 0090
21	28	settable	CAN1_RX1	CAN1 RX1 interrupt	0x0000 0094
22	29	settable	CAN1_SCE	CAN1 SCE interrupt	0x0000 0098
23	30	settable	EXTI9_5	EXTI Line[9:5] interrupts	0x0000 009C
24	31	settable	TIM1_BRK_TIM9	TIM1 Break interrupt and TIM9 global interrupt	0x0000 00A0
25	32	settable	TIM1_UP_TIM10	TIM1 Update interrupt and TIM10 global interrupt	0x0000 00A4
26	33	settable	TIM1_TRG_COM_TIM11	TIM1 Trigger and Commutation interrupts and TIM11 global interrupt	0x0000 00A8
27	34	settable	TIM1_CC	TIM1 Capture Compare interrupt	0x0000 00AC
28	35	settable	TIM2	TIM2 global interrupt	0x0000 00B0

●인터럽트 벡터 테이블- Interrupts (29~43)



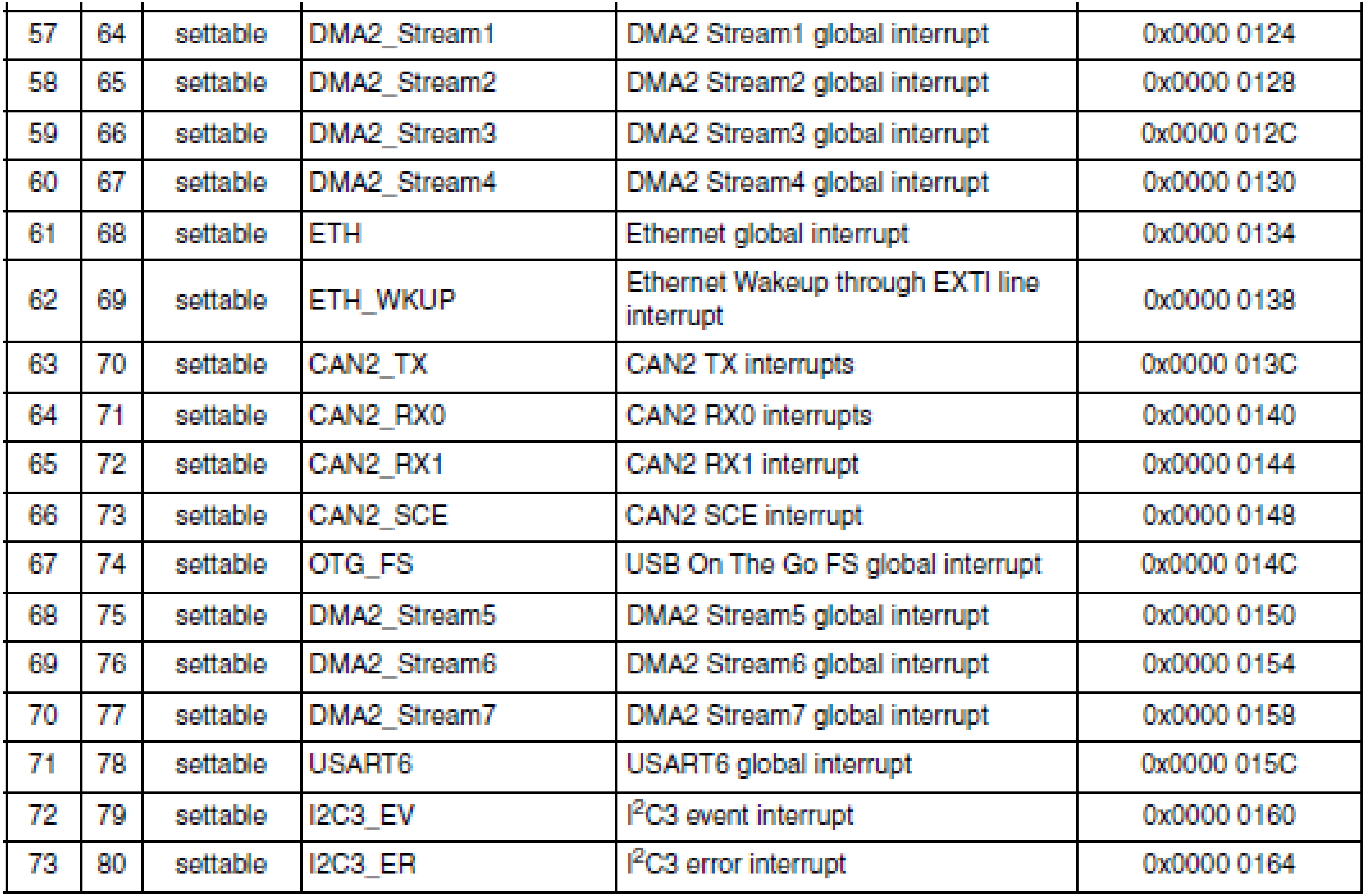
29	36	settable	TIM3	TIM3 global interrupt	0x0000 00B4
30	37	settable	TIM4	TIM4 global interrupt	0x0000 00B8
31	38	settable	I2C1_EV	I^{2}C1 event interrupt	0x0000 00BC
32	39	settable	I2C1_ER	I^{2}C1 error interrupt	0x0000 00C0
33	40	settable	I2C2_EV	I^{2}C2 event interrupt	0x0000 00C4
34	41	settable	I2C2_ER	I^{2}C2 error interrupt	0x0000 00C8
35	42	settable	SPI1	SPI1 global interrupt	0x0000 00CC
36	43	settable	SPI2	SPI2 global interrupt	0x0000 00D0
37	44	settable	USART1	USART1 global interrupt	0x0000 00D4
38	45	settable	USART2	USART2 global interrupt	0x0000 00D8
39	46	settable	USART3	USART3 global interrupt	0x0000 00DC
40	47	settable	EXTI15_10	EXTI Line[15:10] interrupts	0x0000 00E0
41	48	settable	RTC_Alarm	RTC Alarms (A and B) through EXTI line interrupt	0x0000 00E4
42	49	settable	OTG_FS WKUP	USB On-The-Go FS Wakeup through EXTI line interrupt	0x0000 00E8
43	50	settable	TIM8_BRK_TIM12	TIM8 Break interrupt and TIM12 global interrupt	0x0000 00EC

●인터럽트 벡터 테이블- Interrupts (44~56)



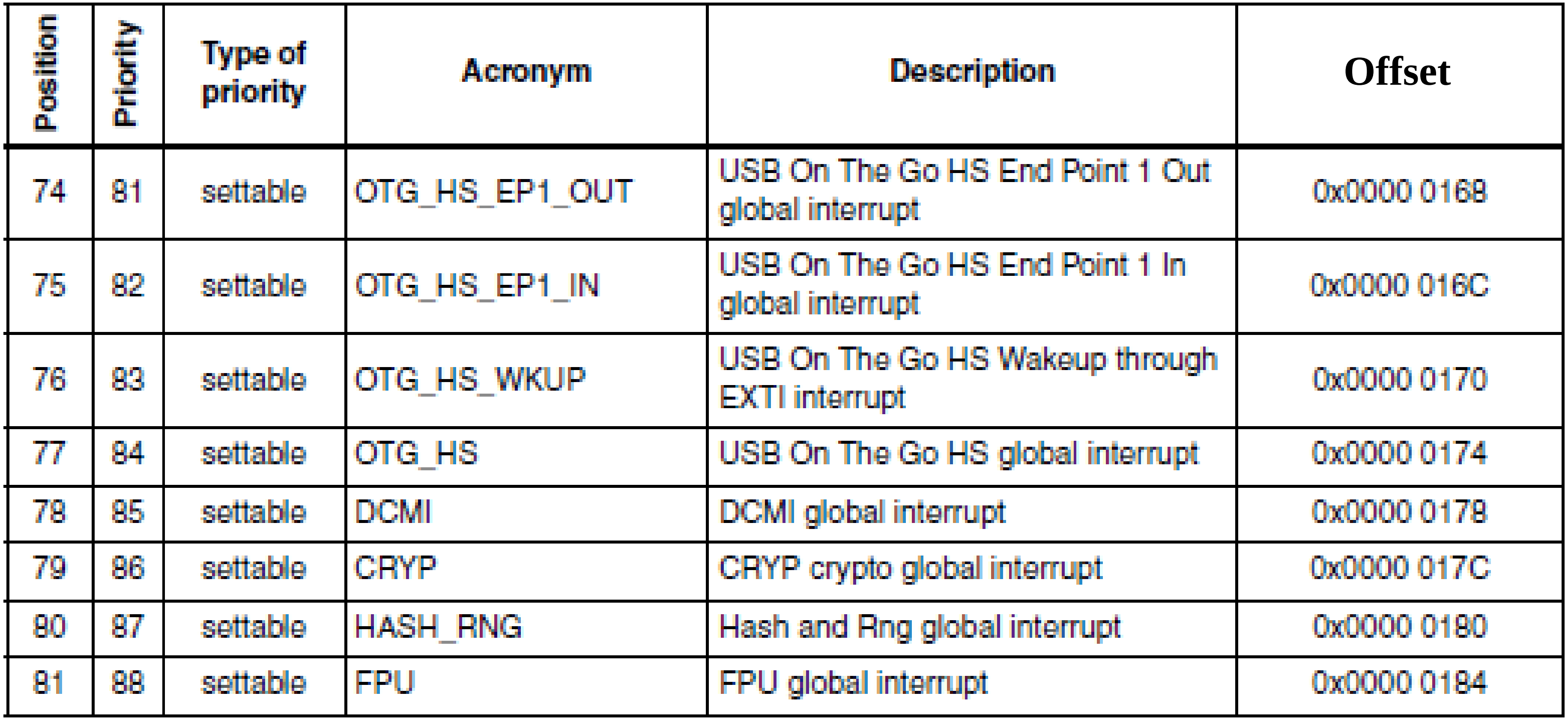
Position	Priority	Type of priority	Acronym	Description	Offset
44	51	settable	TIM8_UP_TIM13	TIM8 Update interrupt and TIM13 global interrupt	0x0000 00F0
45	52	settable	TIM8_TRG_COM_TIM14	TIM8 Trigger and Commutation interrupts and TIM14 global interrupt	0x0000 00F4
46	53	settable	TIM8_CC	TIM8 Capture Compare interrupt	0x0000 00F8
47	54	settable	DMA1_Stream7	DMA1 Stream7 global interrupt	0x0000 00FC
48	55	settable	FSMC	FSMC global interrupt	0x0000 0100
49	56	settable	SDIO	SDIO global interrupt	0x0000 0104
50	57	settable	TIM5	TIM5 global interrupt	0x0000 0108
51	58	settable	SPI3	SPI3 global interrupt	0x0000 010C
52	59	settable	UART4	UART4 global interrupt	0x0000 0110
53	60	settable	UART5	UART5 global interrupt	0x0000 0114
54	61	settable	TIM6_DAC	TIM6 global interrupt, DAC1 and DAC2 underrun error interrupts	0x0000 0118
55	62	settable	TIM7	TIM7 global interrupt	0x0000 011C
56	63	settable	DMA2_Stream0	DMA2 Stream0 global interrupt	0x0000 0120

●인터럽트 벡터 테이블- Interrupts (57~73)



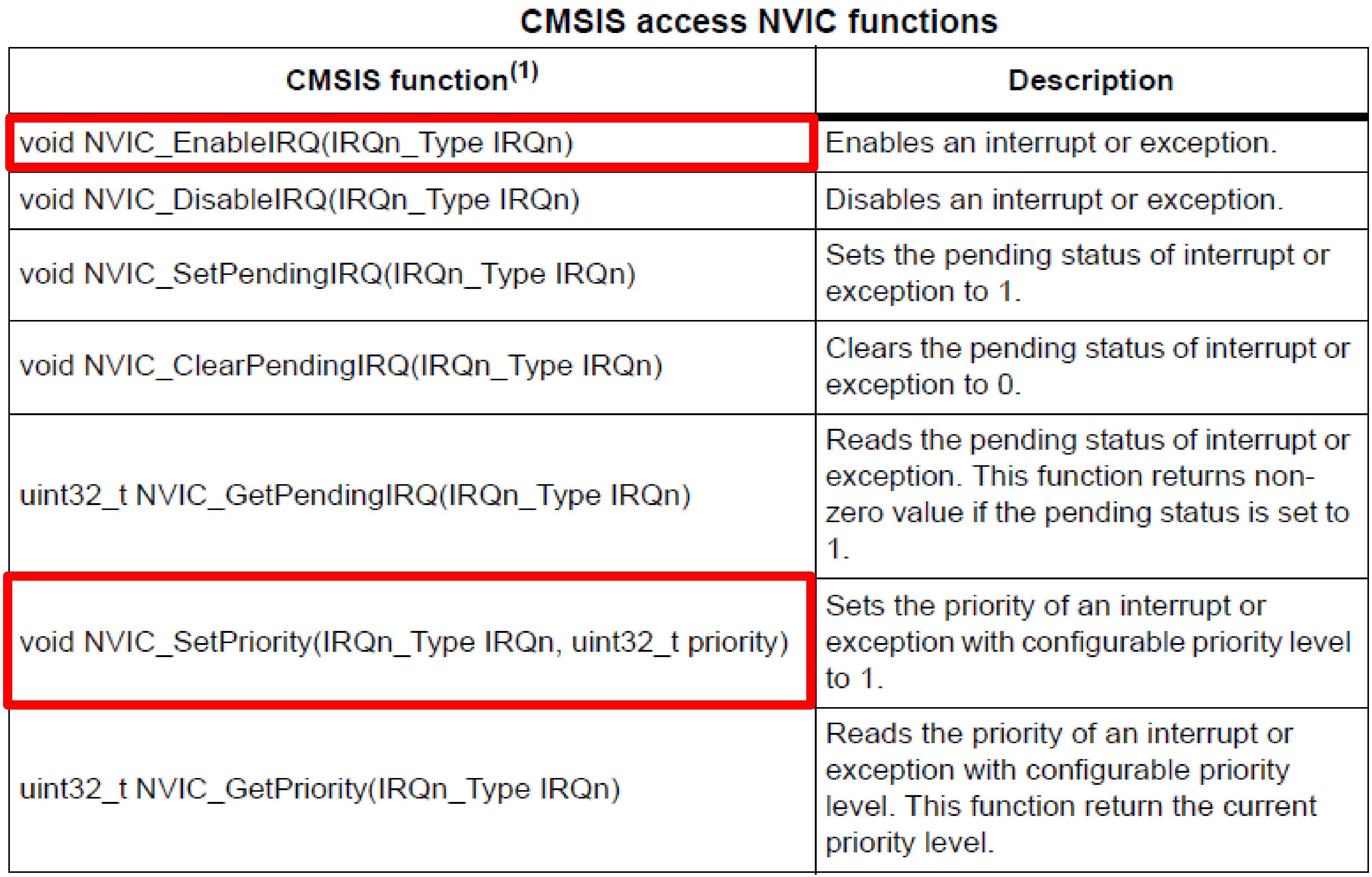
57	64	settable	DMA2_Stream1	DMA2 Stream1 global interrupt	0x0000 0124
58	65	settable	DMA2_Stream2	DMA2 Stream2 global interrupt	0x0000 0128
59	66	settable	DMA2_Stream3	DMA2 Stream3 global interrupt	0x0000 012C
60	67	settable	DMA2_Stream4	DMA2 Stream4 global interrupt	0x0000 0130
61	68	settable	ETH	Ethernet global interrupt	0x0000 0134
62	69	settable	ETH_WKUP	Ethernet Wakeup through EXTI line interrupt	0x0000 0138
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	75	settable	DMA2_Stream5	DMA2 Stream5 global interrupt	0x0000 0150
69	76	settable	DMA2_Stream6	DMA2 Stream6 global interrupt	0x0000 0154
70	77	settable	DMA2_Stream7	DMA2 Stream7 global interrupt	0x0000 0158
71	78	settable	USART6	USART6 global interrupt	0x0000 015C
72	79	settable	I2C3_EV	I^{2}C3 event interrupt	0x0000 0160
73	80	settable	I2C3_ER	I^{2}C3 error interrupt	0x0000 0164

●인터럽트 벡터 테이블- Interrupts (74~81)



Position	Priority	Type of priority	Acronym	Description	Offset
74	81	settable	OTG_HS_EP1_OUT	USB On The Go HS End Point 1 Out global interrupt	0x0000 0168
75	82	settable	OTG_HS_EP1_IN	USB On The Go HS End Point 1 In global interrupt	0x0000 016C
76	83	settable	OTG_HS_WKUP	USB On The Go HS Wakeup through EXTI interrupt	0x0000 0170
77	84	settable	OTG_HS	USB On The Go HS global interrupt	0x0000 0174
78	85	settable	DCMI	DCMI global interrupt	0x0000 0178
79	86	settable	CRYP	CRYP crypto global interrupt	0x0000 017C
80	87	settable	HASH_RNG	Hash and Rng global interrupt	0x0000 0180
81	88	settable	FPU	FPU global interrupt	0x0000 0184

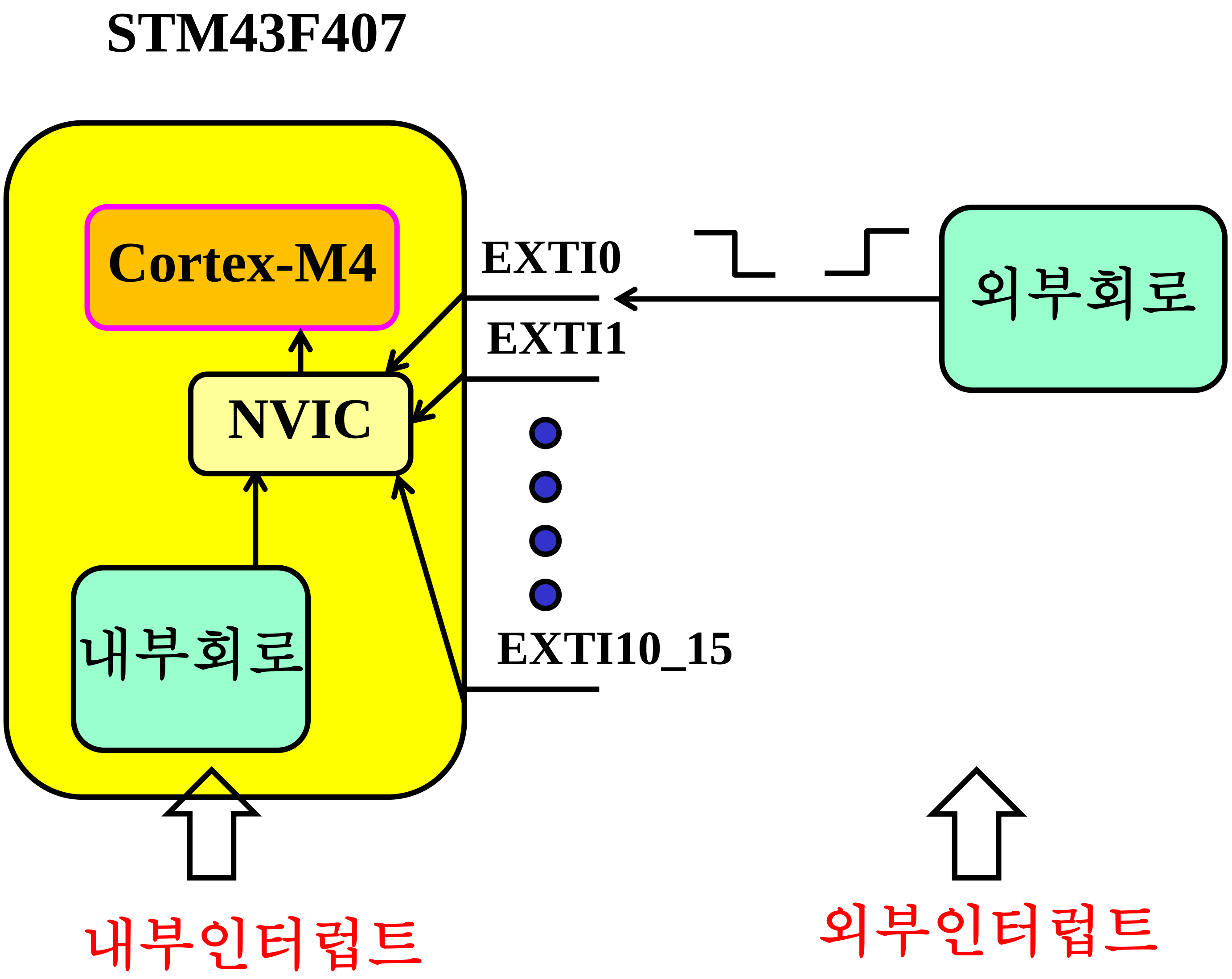
● NVIC 설정 API



CMSIS access NVIC functions

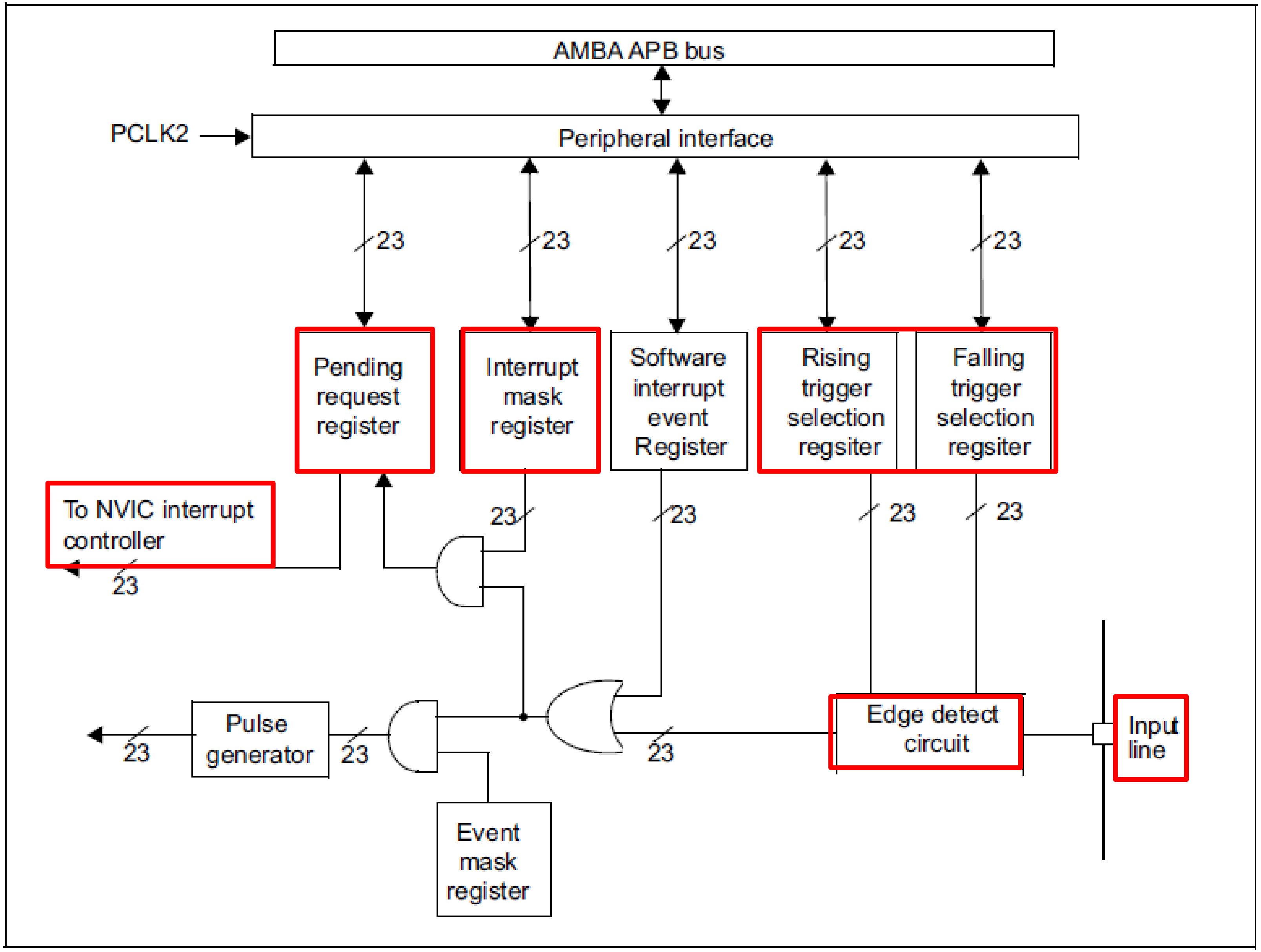
CMSIS function(1)	Description
void NVIC_EnableIRQ(IRQn_Type IRQn)	Enables an interrupt or exception.
void NVIC_DisableIRQ(IRQn_Type IRQn)	Disables an interrupt or exception.
void NVIC_SetPendingIRQ(IRQn_Type IRQn)	Sets the pending status of interrupt or exception to 1.
void NVIC_ClearPendingIRQ(IRQn_Type IRQn)	Clears the pending status of interrupt or exception to 0.
uint32_t NVIC_GetPendingIRQ(IRQn_Type IRQn)	Reads the pending status of interrupt or exception. This function returns non-zero value if the pending status is set to 1.
void NVIC_SetPriority(IRQn_Type IRQn, uint32_t priority)	Sets the priority of an interrupt or exception with configurable priority level to 1.
uint32_t NVIC_GetPriority(IRQn_Type IRQn)	Reads the priority of an interrupt or exception with configurable priority level. This function return the current priority level.

● External Interrupt 개념도



● EXTI 구조와 동작

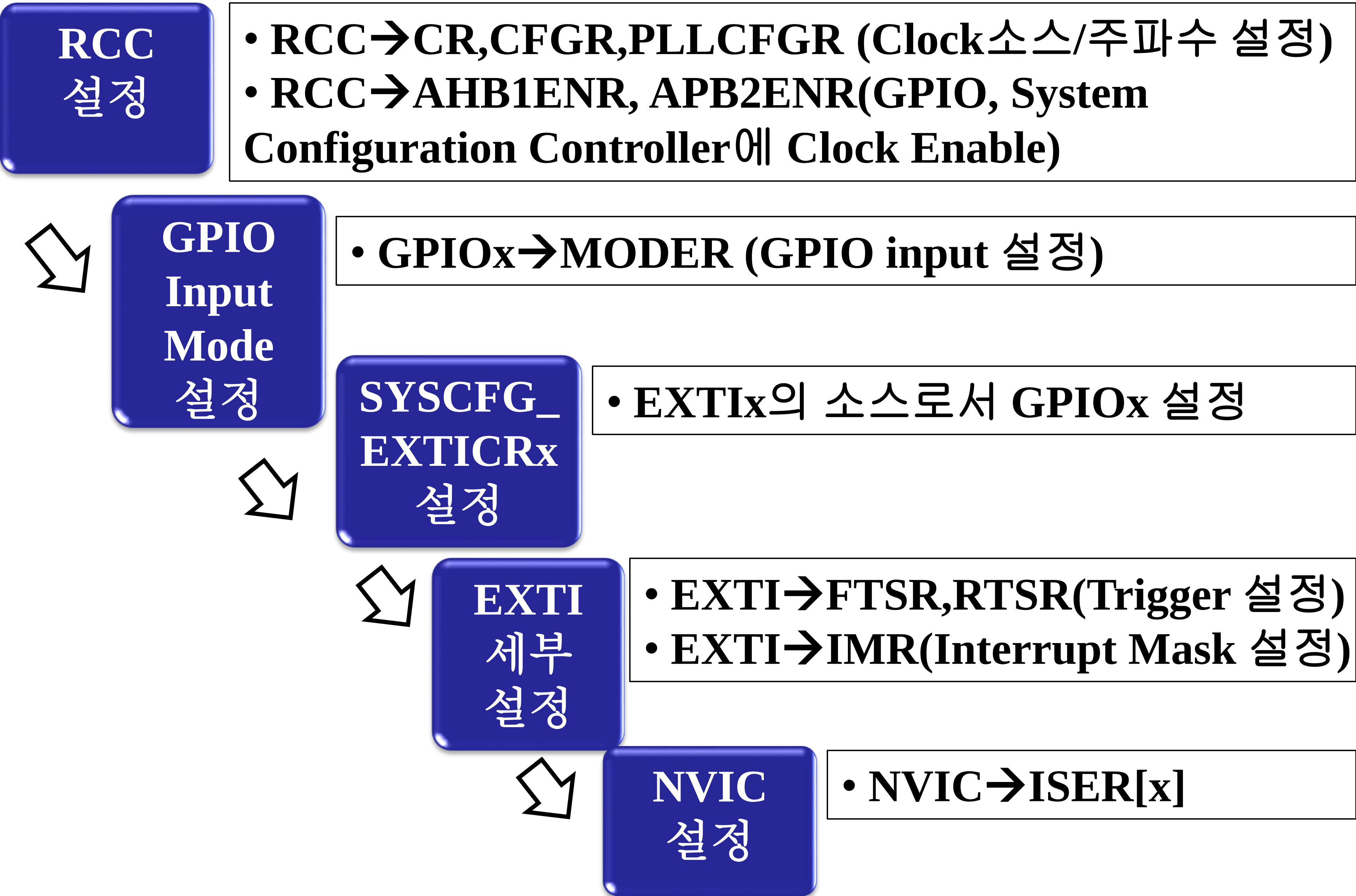
External interrupt/event controller block diagram



● EXTI 구조와 동작 (그림(블록도) 설명)

1. **Input line에 Edge 신호 발생**
2. **Trigger Mode에 따라 설정된 Edge detect 회로에서 신호 생성**
3. **OR 게이트를 통과한 신호가 Interrupt Mask의 신호와 AND 연산을 해서 Masking 안된 인터럽트(enable된 인터럽트)신호만 통과**
4. **통과된 신호에 해당하는 Pending 레지스터의 해당bit를 set, Pending 신호는 NVIC 보내 Interrupt 발생**

● 프로그램에서의 EXTI set-up 과정 및 레지스터 설정



RCC 설정

- RCC→CR,CFGR,PLLCFGR (Clock소스/주파수 설정)
- RCC→AHB1ENR, APB2ENR(GPIO, System Configuration Controller에 Clock Enable)

GPIO Input Mode 설정

- GPIOx→MODER (GPIO input 설정)

SYSCFG_ EXTICRx 설정

- EXTIx의 소스로서 GPIOx 설정

EXTI 세부 설정

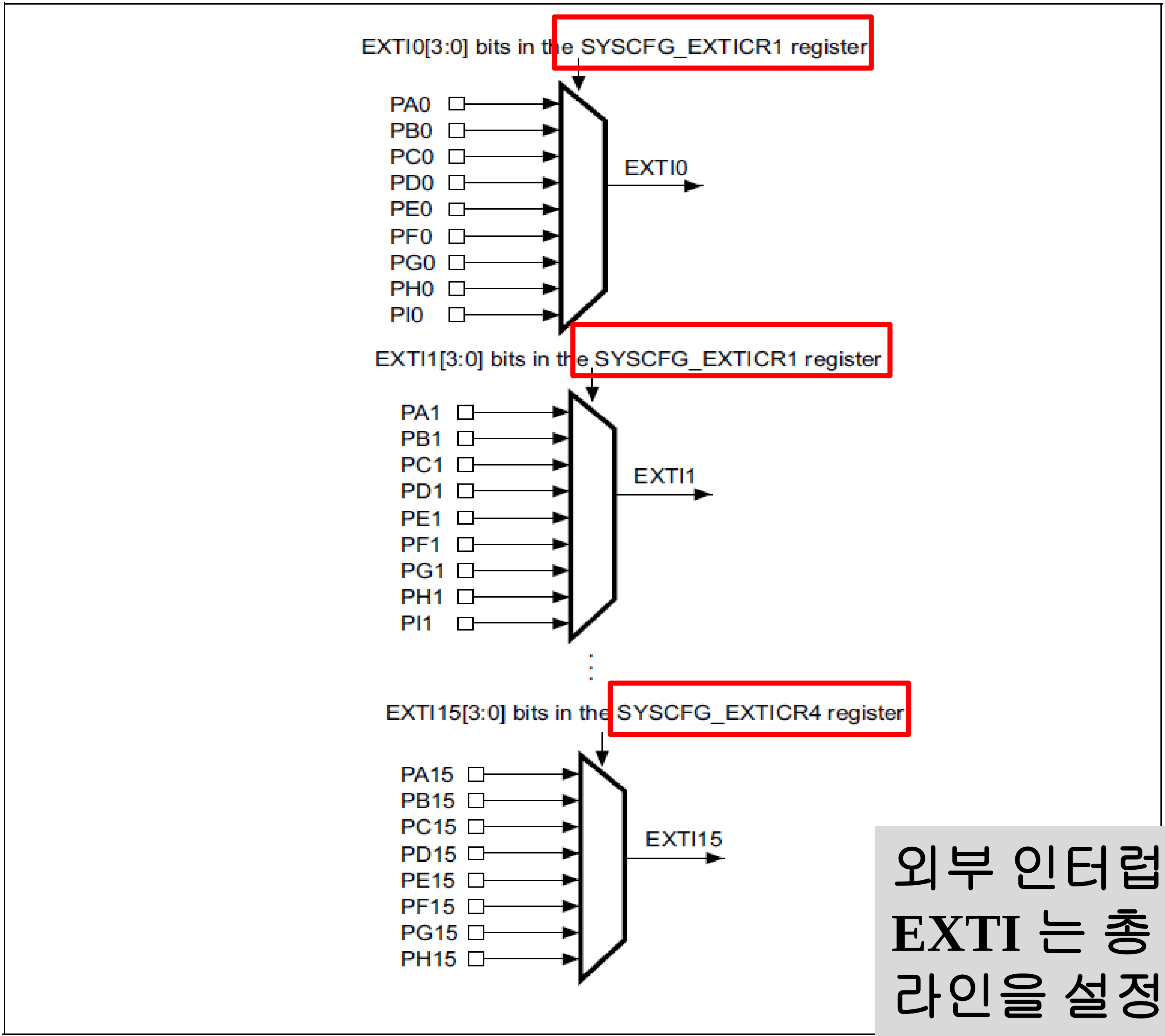
- EXTI→FTSR,RTSR(Trigger 설정)
- EXTI→IMR(Interrupt Mask 설정)

NVIC 설정

- NVIC→ISER[x]

● EXTI 구조와 동작

External interrupt/event GPIO mapping


EXTI0[3:0] bits in the SYSCFG_EXTICR1 register
PA0
PB0
PC0
PD0
PE0
PF0
PG0
PH0
PI0
EXTI0
EXTI1[3:0] bits in the SYSCFG_EXTICR1 register
PA1
PB1
PC1
PD1
PE1
PF1
PG1
PH1
PI1
EXTI1
EXTI15[3:0] bits in the SYSCFG_EXTICR4 register
PA15
PB15
PC15
PD15
PE15
PF15
PG15
PH15
EXTI15

외부 인터럽트를 사용 할 경우 **EXTI** 는 총 **16**개의 인터럽트 라인을 설정 가능

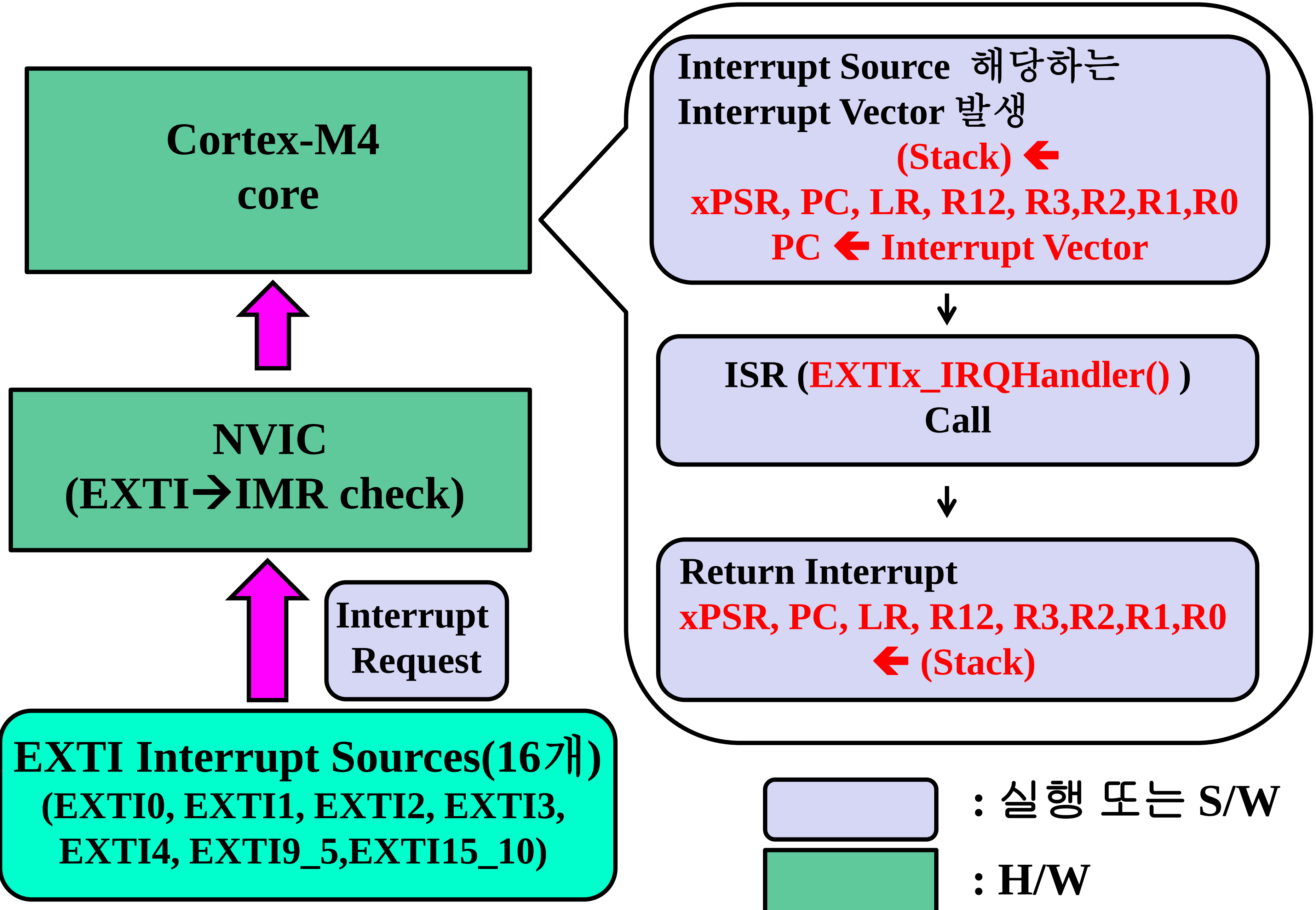
●NVIC 설정 API (EXTI 관련)

- **NVIC_SetPriority의 priority parameter는 0~15까지** 설정 가능, 낮은 숫자가 높은 우선순위

- **NVIC_EnableIRQ의 parameter는** 아래의 항목 중 선택
  EXTI0_IRQn, EXTI1_IRQn,
  EXTI2_IRQn, EXTI3_IRQn,
  EXTI4_IRQn, EXTI9_5_IRQn
  EXTI15_10_IRQn

- **ISR title:**
  EXTI0_IRQHandler, EXTI1_IRQHandler
  EXTI2_IRQHandler, EXTI3_IRQHandler
  EXTI4_IRQHandler, EXTI9_5_IRQHandler
  EXTI15_10_IRQHandler

STM32F407IG Interrupt(EXTI) flow



Cortex-M4 core

NVIC (EXTI→IMR check)

Interrupt Request

EXTI Interrupt Sources(16개) (EXTI0, EXTI1, EXTI2, EXTI3, EXTI4, EXTI9_5,EXTI15_10)

Interrupt Source 해당하는 Interrupt Vector 발생
(Stack) ← xPSR, PC, LR, R12, R3,R2,R1,R0
PC ← Interrupt Vector

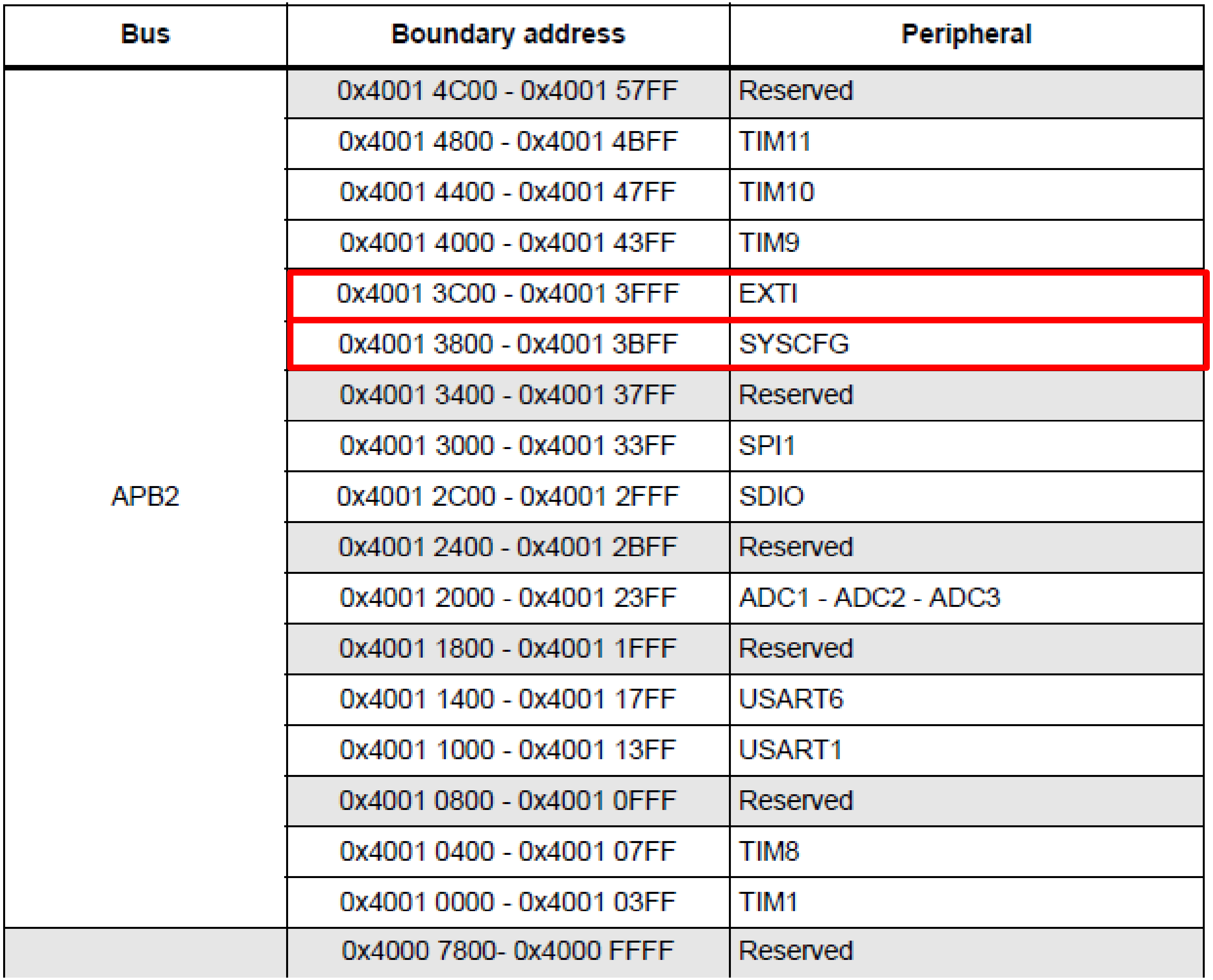
ISR (EXTIx_IRQHandler()) Call

Return Interrupt
xPSR, PC, LR, R12, R3,R2,R1,R0 ← (Stack)

🟪 : 실행 또는 S/W

🟩 : H/W

● EXTI 관련 Memory Map



Bus	Boundary address	Peripheral
APB2	0x4001 4C00 - 0x4001 57FF	Reserved
	0x4001 4800 - 0x4001 4BFF	TIM11
	0x4001 4400 - 0x4001 47FF	TIM10
	0x4001 4000 - 0x4001 43FF	TIM9
	0x4001 3C00 - 0x4001 3FFF	EXTI
	0x4001 3800 - 0x4001 3BFF	SYSCFG
	0x4001 3400 - 0x4001 37FF	Reserved
	0x4001 3000 - 0x4001 33FF	SPI1
	0x4001 2C00 - 0x4001 2FFF	SDIO
	0x4001 2400 - 0x4001 2BFF	Reserved
	0x4001 2000 - 0x4001 23FF	ADC1 - ADC2 - ADC3
	0x4001 1800 - 0x4001 1FFF	Reserved
	0x4001 1400 - 0x4001 17FF	USART6
	0x4001 1000 - 0x4001 13FF	USART1
	0x4001 0800 - 0x4001 0FFF	Reserved
	0x4001 0400 - 0x4001 07FF	TIM8
	0x4001 0000 - 0x4001 03FF	TIM1
	0x4000 7800- 0x4000 FFFF	Reserved

● External Interrupt/Event Controller(EXTI) 관련 registers (structure 구조, stm32f4xx.h)

```
/**   * @brief External Interrupt/Event Controller  */

typedef struct
{
  __IO uint32_t IMR;    /*!< EXTI Interrupt mask register, Address offset: 0x00 */
  __IO uint32_t EMR;    /*!< EXTI Event mask register, Address offset: 0x04 */
  __IO uint32_t RTSR;   /*!< EXTI Rising trigger selection register,  Address offset: 0x08 */
  __IO uint32_t FTSR;   /*!< EXTI Falling trigger selection register, Address offset: 0x0C */
  __IO uint32_t SWIER;  /*!< EXTI Software interrupt event register,  Address offset: 0x10 */
  __IO uint32_t PR;     /*!< EXTI Pending register, Address offset: 0x14 */
} EXTI_TypeDef;
```

● System configuration controller(SYSCFG) 관련 registers (structure 구조 , stm32f4xx.h)

```
typedef struct
{
  __IO uint32_t MEMRMP;   /*!< SYSCFG memory remap register, Address offset:
                                                 0x00 */
  __IO uint32_t PMC;         /*!< SYSCFG peripheral mode configuration register,
                                                 Address offset: 0x04      */
  __IO uint32_t EXTICR[4];    /*!< SYSCFG external interrupt configuration
                                                 registers,  Address offset: 0x08-0x14 */
  uint32_t      RESERVED[2];  /*!< Reserved, 0x18-0x1C    */
  __IO uint32_t CMPCR;        /*!< SYSCFG Compensation cell control register,
                                                 Address offset: 0x20      */

} SYSCFG_TypeDef;
```

● EXTI, SYSCFG 선언(stm32f4xx.h)

```
#define PERIPH_BASE       ((uint32_t)0x40000000) /*!< Peripheral base address
in the alias region      */

/*!< Peripheral memory map */
#define APB1PERIPH_BASE       PERIPH_BASE
#define APB2PERIPH_BASE       (PERIPH_BASE + 0x00010000)
#define AHB1PERIPH_BASE       (PERIPH_BASE + 0x00020000)
#define AHB2PERIPH_BASE       (PERIPH_BASE + 0x10000000)

/*!< APB2 peripherals */
#define SYSCFG_BASE       (APB2PERIPH_BASE + 0x3800)
#define EXTI_BASE         (APB2PERIPH_BASE + 0x3C00)

/** @addtogroup Peripheral_declaration  * @{  */
#define SYSCFG        ((SYSCFG_TypeDef *) SYSCFG_BASE)
#define EXTI          ((EXTI_TypeDef *) EXTI_BASE)
```

● 참고용 선언부(core_cm4.h)

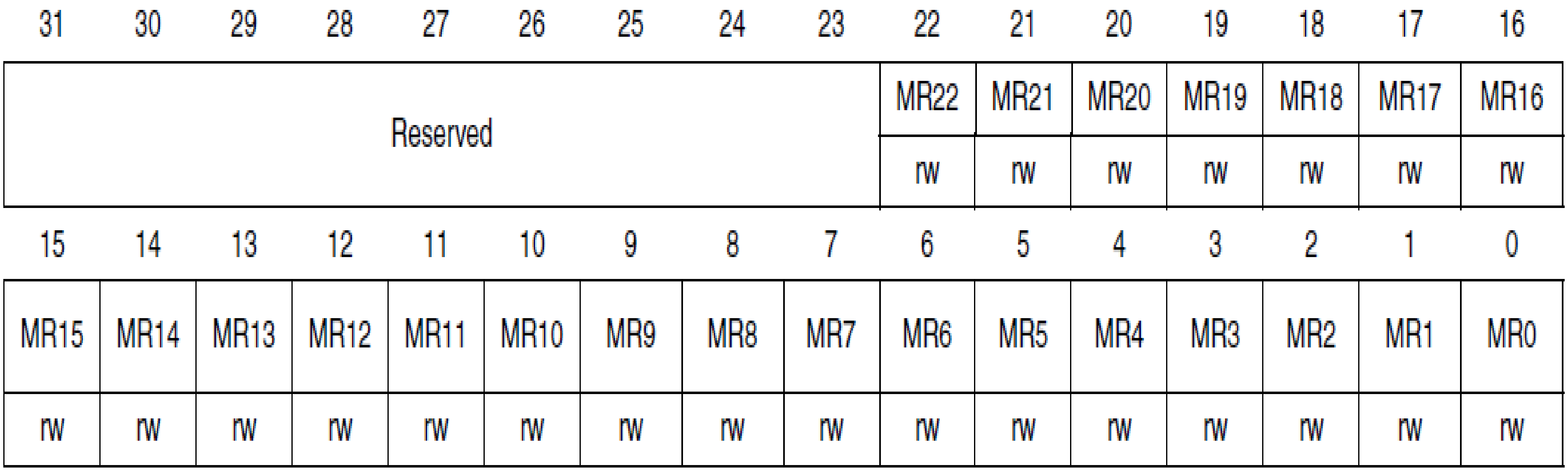
```
/* IO definitions (access restrictions to peripheral registers) */
#ifdef __cplusplus
  #define     __I     volatile           /*!< defines 'read only' permissions                */
#else
  #define     __I     volatile const   /*!< defines 'read only' permissions               */
#endif
#define     __O     volatile              /*!< defines 'write only' permissions               */
#define     __IO    volatile             /*!< defines 'read / write' permissions            */
/*@} end of group CMSIS_core_definitions */

/** \defgroup CMSIS_core_register CMSIS Core Register
  Core Register contain:
  - Core Register
  - Core NVIC Register
  - Core SCB Register
  - Core SysTick Register
  - Core Debug Register
  - Core MPU Register
  - Core FPU Register
*/
```

● NVIC 관련 registers (structure 구조, core_cm4.h) 및 선언

```
typedef struct {
  __IO uint32_t ISER[8];  /*!< Offset: 0x000 Interrupt Set Enable Register     */
       uint32_t RESERVED0[24];
  __IO uint32_t ICER[8];  /*!< Offset: 0x080 Interrupt Clear Enable Register  */
       uint32_t RSERVED1[24];
  __IO uint32_t ISPR[8];  /*!< Offset: 0x100 Interrupt Set Pending Register   */
       uint32_t RESERVED2[24];
  __IO uint32_t ICPR[8];  /*!< Offset: 0x180 Interrupt Clear Pending Register*/
       uint32_t RESERVED3[24];
  __IO uint32_t IABR[8];  /*!< Offset: 0x200 Interrupt Active bit Register     */
       uint32_t RESERVED4[56];
  __IO uint8_t  IP[240];  /*!< Offset: 0x300 Interrupt Priority Register (8Bit) */
       uint32_t RESERVED5[644];
  __O  uint32_t STIR;  /*!< Offset: 0xE00 Software Trigger Interrupt Register    */
}  NVIC_Type;
/* Memory mapping of Cortex-M4 Hardware */
#define SCS_BASE     (0xE000E000)    /*!< System Control Space Base Address */
#define NVIC_BASE    (SCS_BASE +  0x0100)  /*!< NVIC Base Address  */
#define NVIC    ((NVIC_Type *)  NVIC_BASE) /*!< NVIC configuration struct  */
```

● External Interrupt 관련 Register(EXTI)

(1) Interrupt mask register (EXTI_IMR)
Address offset: 0x00
Reset value: 0x0000 0000



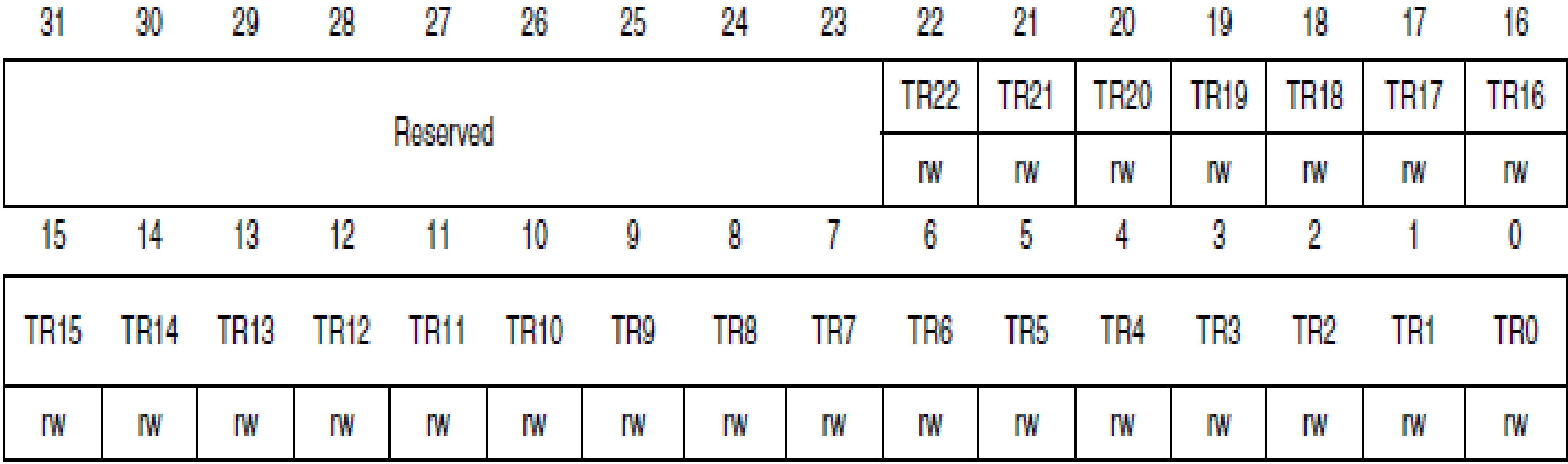
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									MR22	MR21	MR20	MR19	MR18	MR17	MR16
									rw	rw	rw	rw	rw	rw	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 22:0 MRx: Interrupt mask on line x
0: Event request from line x is masked
1: Event request from line x is not masked

(2) Rising trigger selection register (EXTI_RTSR)

Address offset: 0x08

Reset value: 0x0000 0000



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									TR22	TR21	TR20	TR19	TR18	TR17	TR16
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 22:0 TRx: Rising trigger event configuration bit of line x

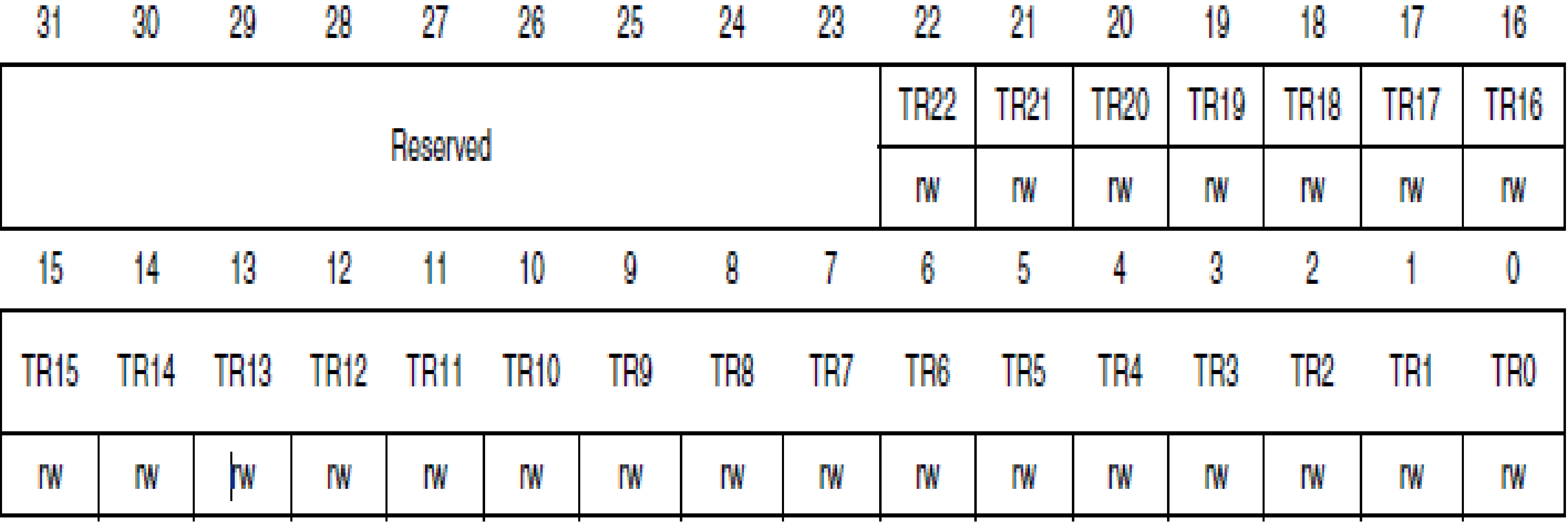
0: Rising trigger disabled (for Event and Interrupt) for input line

1: Rising trigger enabled (for Event and Interrupt) for input

(3) Falling trigger selection register (EXTI_FTSR)

Address offset: 0x0C

Reset value: 0x0000 0000



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									TR22	TR21	TR20	TR19	TR18	TR17	TR16
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 22:0 TRx: Falling trigger event configuration bit of line x

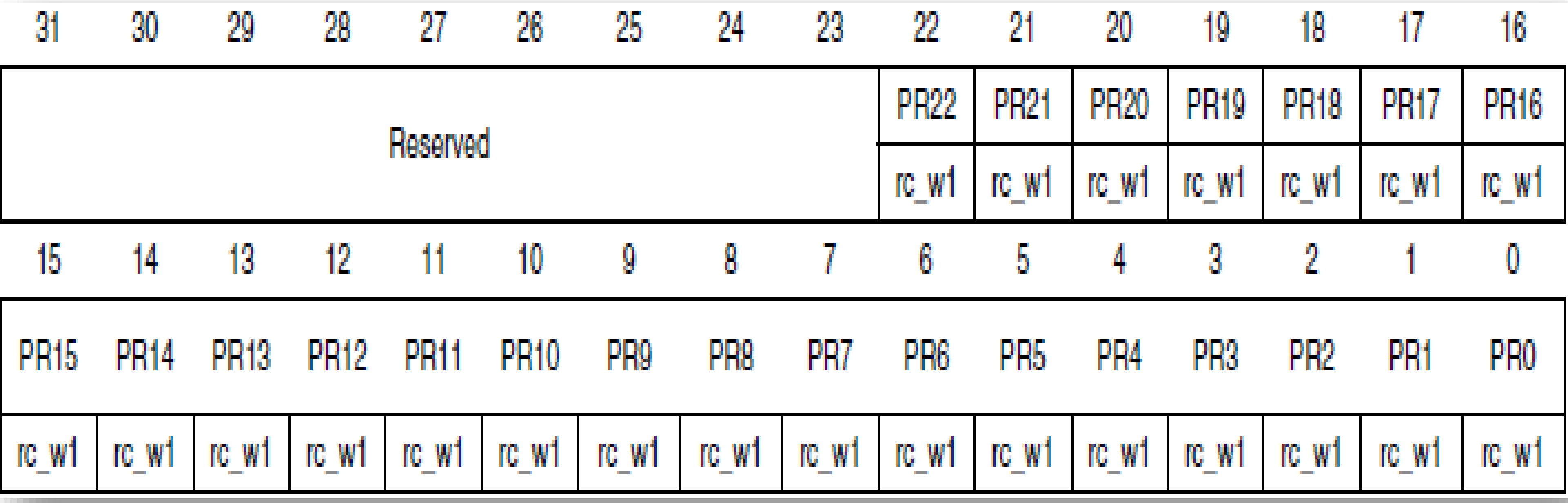
0: Falling trigger disabled (for Event and Interrupt) for input line

1: Falling trigger enabled (for Event and Interrupt) for input line.

(4) Pending register (EXTI_PR)

Address offset: 0x14

Reset value: undefined



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PR22	PR21	PR20	PR19	PR18	PR17	PR16
									rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

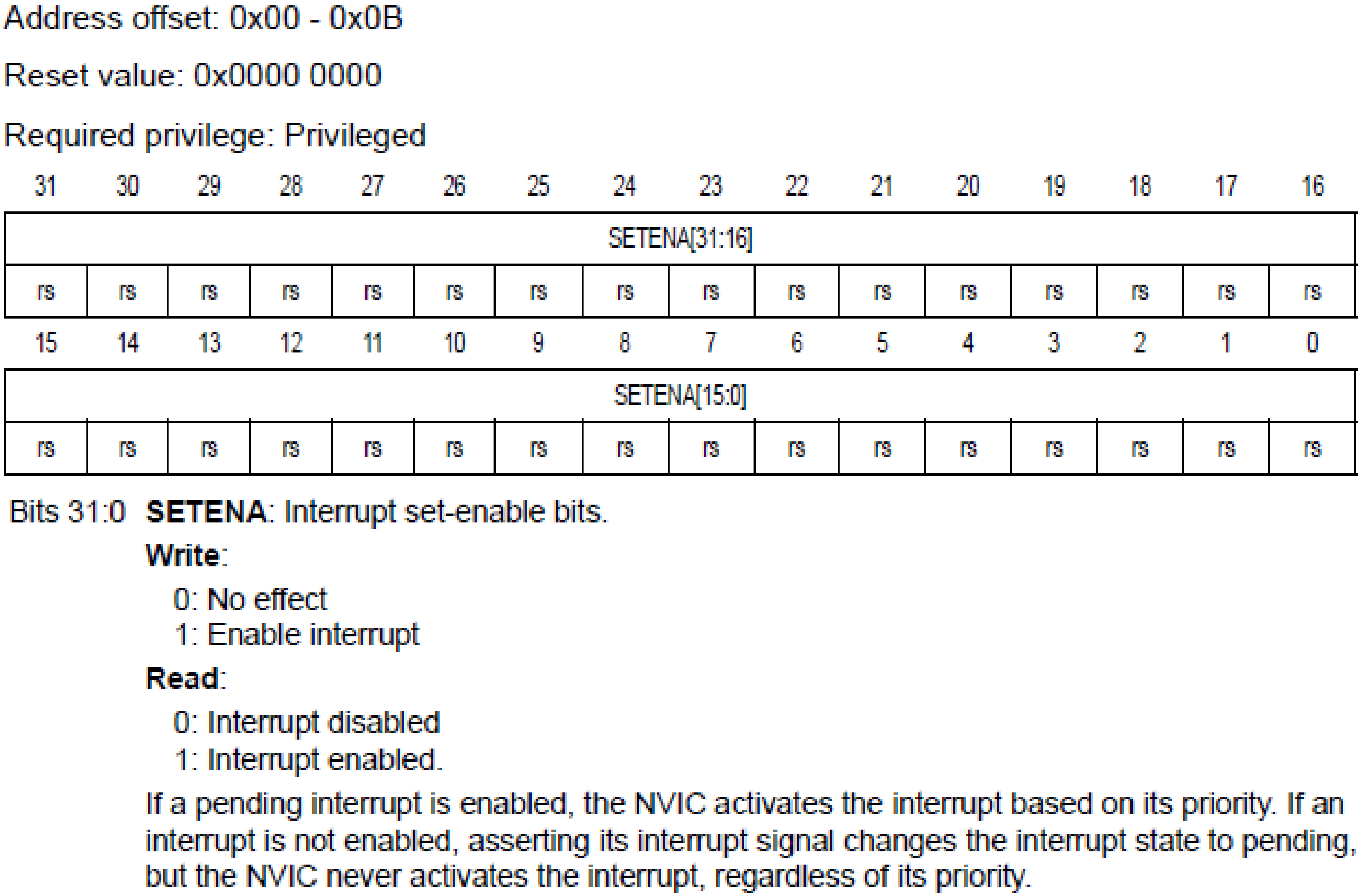
Bits 22:0 PRx: Pending bit

0: No trigger request occurred

1: selected trigger request occurred

This bit is set when the selected edge event arrives on the external interrupt line. This bit is cleared by writing a 1 to the bit or by changing the sensitivity of the edge detector

(5) Interrupt set-enable registers (NVIC_ISERx)



Address offset: 0x00 - 0x0B

Reset value: 0x0000 0000

Required privilege: Privileged

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

Bits 31:0 **SETENA**: Interrupt set-enable bits.

Write:

0: No effect
1: Enable interrupt

Read:

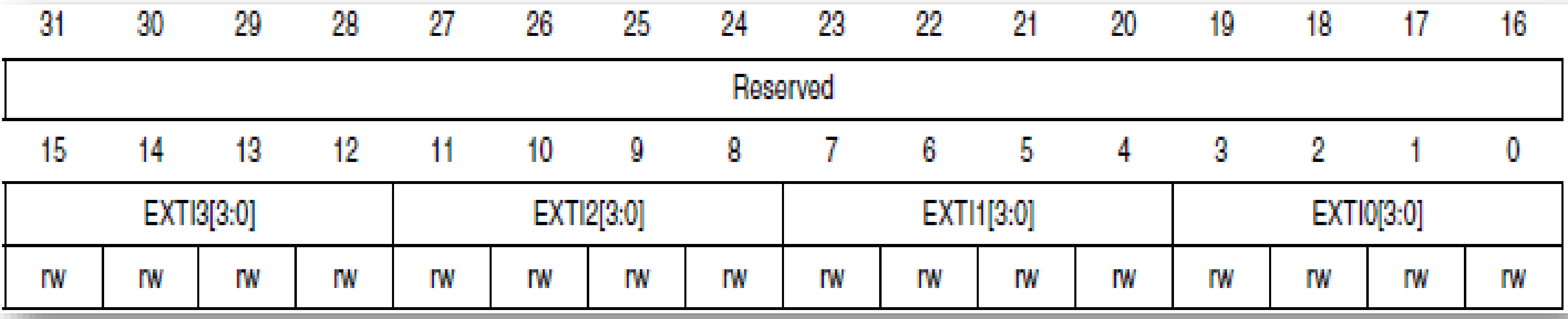
0: Interrupt disabled
1: Interrupt enabled.

If a pending interrupt is enabled, the NVIC activates the interrupt based on its priority. If an interrupt is not enabled, asserting its interrupt signal changes the interrupt state to pending, but the NVIC never activates the interrupt, regardless of its priority.

(6-1) SYSCFG external interrupt configuration register 1 (SYSCFG_EXTICR1)

Address offset: 0x08

Reset value: 0x0000



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

Bits 15:0 EXTIx[3:0]: EXTI x configuration (x = 0 to 3)
These bits are written by software to select the source input for the EXTIx external interrupt.

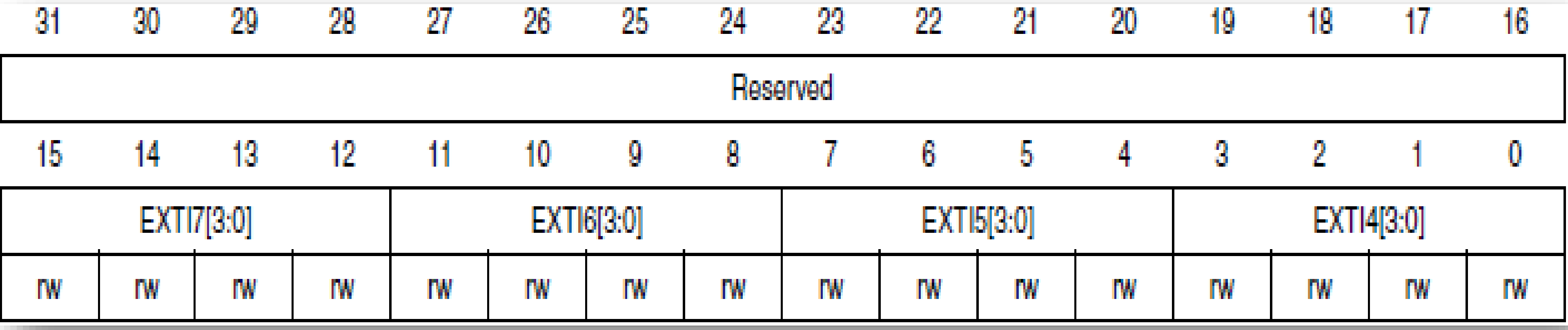
0000: PA[x] pin
0001: PB[x] pin
0010: PC[x] pin
0011: PD[x] pin
0100: PE[x] pin
0101: PF[x] pin
0110: PG[x] pin
0111: PH[x] pin
1000: PI[x] pin

SYSCFG_EXTICR1 : 0 ~ 3 Pin
SYSCFG_EXTICR2 : 4 ~ 7 Pin
SYSCFG_EXTICR3 : 8 ~ 11 Pin
SYSCFG_EXTICR4 : 12 ~ 15 Pin

(6-2) SYSCFG external interrupt configuration register 2 (SYSCFG_EXTICR2)

Address offset: 0x0C

Reset value: 0x0000



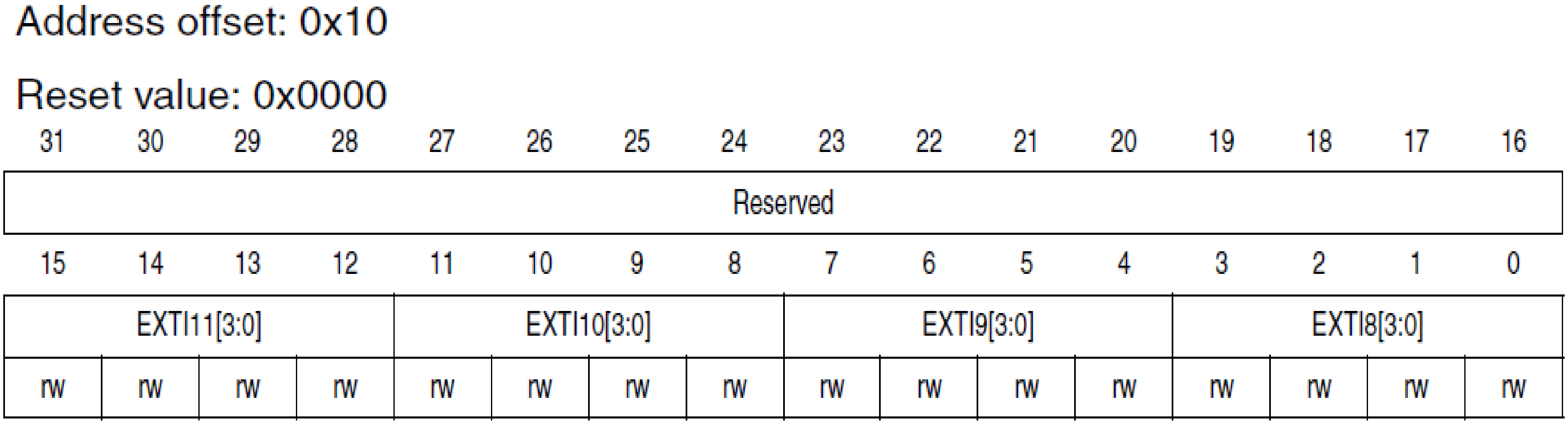
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 EXTIx[3:0]: EXTI x configuration (x = 0 to 3)

These bits are written by software to select the source input for the EXTIx external interrupt.

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

(6-3) SYSCFG external interrupt configuration register 3 (SYSCFG_EXTICR3)



Address offset: 0x10

Reset value: 0x0000

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
EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 EXTIx[3:0]: EXTI x configuration (x = 0 to 3)

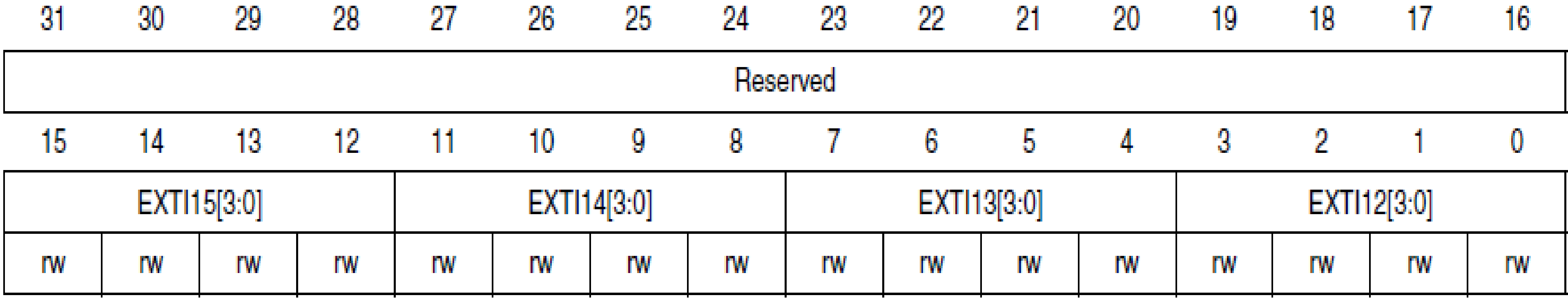
These bits are written by software to select the source input for the EXTIx external interrupt.

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

(6-4) SYSCFG external interrupt configuration register 4 (SYSCFG_EXTICR4)

Address offset: 0x14

Reset value: 0x0000



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**
EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 EXTIx[3:0]: EXTI x configuration (x = 0 to 3)

These bits are written by software to select the source input for the EXTIx external interrupt.

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

●SYSCFG registers

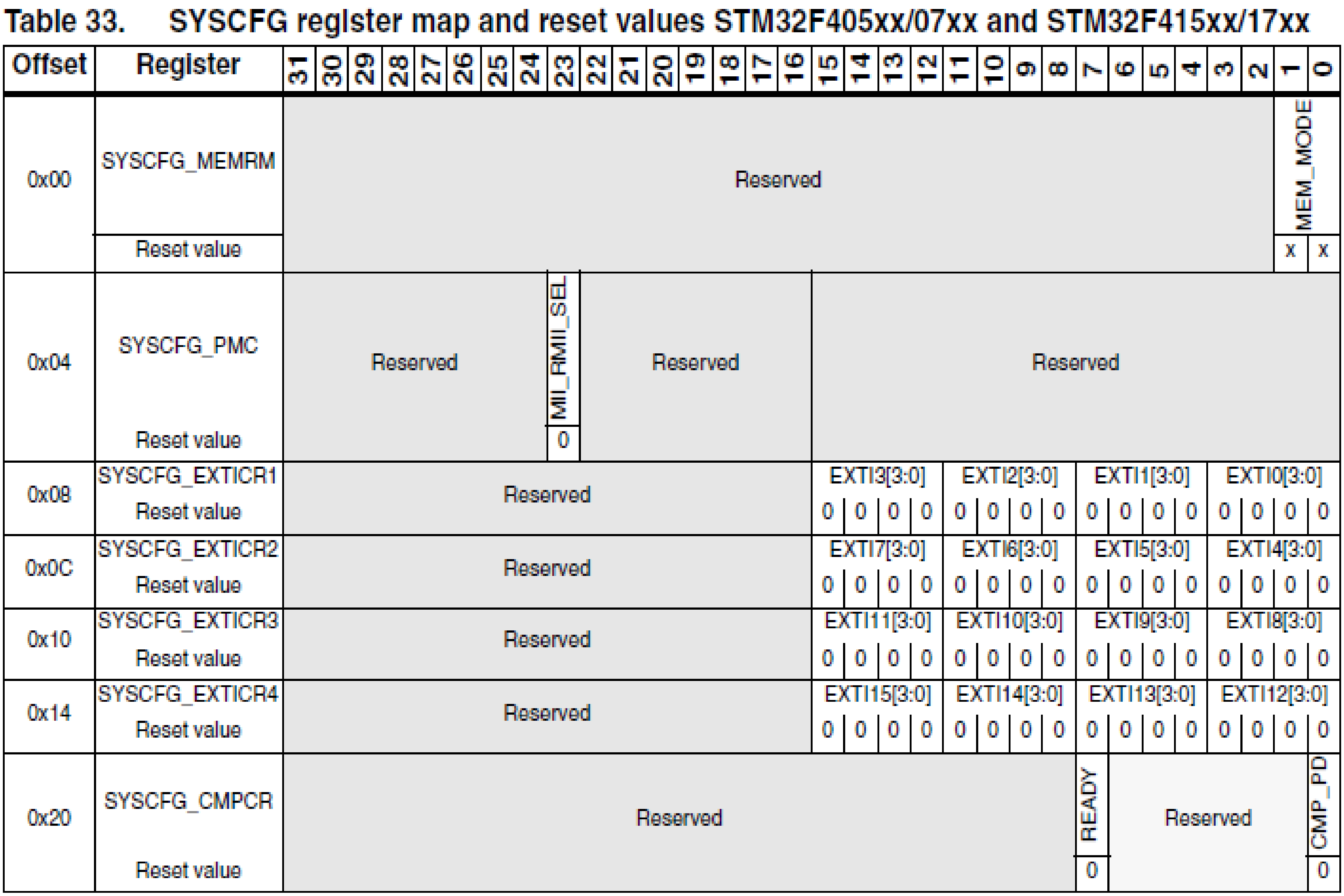


Table 33. SYSCFG register map and reset values STM32F405xx/07xx and STM32F415xx/17xx

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SYSCFG_MEMRM	Reserved																														MEM_MODE	
	Reset value																															x	x
0x04	SYSCFG_PMC	Reserved								MII_RMII_SEL	Reserved							Reserved															
	Reset value									0																							
0x08	SYSCFG_EXTICR1	Reserved																EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	SYSCFG_EXTICR2	Reserved																EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	SYSCFG_EXTICR3	Reserved																EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	SYSCFG_EXTICR4	Reserved																EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	SYSCFG_CMPCR	Reserved																								READY	Reserved						CMP_PD
	Reset value																									0							0

• Vector table Exception number(Priority)지정(stm32f4xx.h)

```
typedef enum IRQn
{
/******  Cortex-M4 Processor Exceptions Numbers ******************/
  NonMaskableInt_IRQn         = -14,    /*!< 2 Non Maskable Interrupt          */
  MemoryManagement_IRQn  = -12,    /*!< 4 Cortex-M4 Memory Management Interrupt */
  BusFault_IRQn               = -11,    /*!< 5 Cortex-M4 Bus Fault Interrupt     */
  UsageFault_IRQn             = -10,    /*!< 6 Cortex-M4 Usage Fault Interrupt     */
  SVCall_IRQn                 = -5,     /*!< 11 Cortex-M4 SV Call Interrupt     */
  DebugMonitor_IRQn    = -4,     /*!< 12 Cortex-M4 Debug Monitor Interrupt    */
  PendSV_IRQn                 = -2,     /*!< 14 Cortex-M4 Pend SV Interrupt      */
  SysTick_IRQn                = -1,     /*!< 15 Cortex-M4 System Tick Interrupt      */

/******  STM32 specific Interrupt Numbers ******************************/
  WWDG_IRQn              = 0,      /*!< Window WatchDog Interrupt      */
  PVD_IRQn                  = 1,      /*!< PVD through EXTI Line detection Interrupt   */
  TAMP_STAMP_IRQn   = 2,    /*!< Tamper and TimeStamp interrupts through the EXTI
line            */
  RTC_WKUP_IRQn    = 3,      /*!< RTC Wakeup interrupt through the EXTI line   */
 FLASH_IRQn              = 4,      /*!< FLASH global Interrupt                              */
  RCC_IRQn                 = 5,      /*!< RCC global Interrupt                                */
```

• Vector table Exception number(Priority)지정(stm32f4xxx.h)

```
 EXTI0_IRQn              = 6,     /*!< EXTI Line0 Interrupt                          */
 EXTI1_IRQn              = 7,     /*!< EXTI Line1 Interrupt                          */
 EXTI2_IRQn              = 8,     /*!< EXTI Line2 Interrupt                          */
 EXTI3_IRQn              = 9,     /*!< EXTI Line3 Interrupt                          */
 EXTI4_IRQn              = 10,     /*!< EXTI Line4 Interrupt                          */
 DMA1_Stream0_IRQn            = 11,    /*!< DMA1 Stream 0 global Interrupt     */
 DMA1_Stream1_IRQn            = 12,    /*!< DMA1 Stream 1 global Interrupt      */
 DMA1_Stream2_IRQn            = 13,    /*!< DMA1 Stream 2 global Interrupt    */
 DMA1_Stream3_IRQn            = 14,    /*!< DMA1 Stream 3 global Interrupt    */
 DMA1_Stream4_IRQn            = 15,    /*!< DMA1 Stream 4 global Interrupt      */
 DMA1_Stream5_IRQn            = 16,    /*!< DMA1 Stream 5 global Interrupt         */
 DMA1_Stream6_IRQn            = 17,    /*!< DMA1 Stream 6 global Interrupt        */
 ADC_IRQn                = 18,    /*!< ADC1, ADC2 and ADC3 global Interrupts     */

#if defined(STM32F40_41xxx)
 CAN1_TX_IRQn              = 19,    /*!< CAN1 TX Interrupt           */
 CAN1_RX0_IRQn              = 20,    /*!< CAN1 RX0 Interrupt      */
 CAN1_RX1_IRQn              = 21,    /*!< CAN1 RX1 Interrupt        */
 CAN1_SCE_IRQn              = 22,    /*!< CAN1 SCE Interrupt     */
 EXTI9_5_IRQn             = 23,    /*!< External Line[9:5] Interrupts    */
```

• Vector table Exception number(Priority)지정(stm32f4xxx.h)

```
 TIM1_BRK_TIM9_IRQn = 24,   /*!< TIM1 Break interrupt & TIM9 global interrupt  */
 TIM1_UP_TIM10_IRQn  = 25,   /*!< TIM1 Update Interrupt & TIM10 global interrupt */
 TIM1_TRG_COM_TIM11_IRQn     = 26,     /*!< TIM1 Trigger and Commutation
Interrupt and TIM11 global interrupt */
 TIM1_CC_IRQn                = 27,     /*!< TIM1 Capture Compare Interrupt       */
 TIM2_IRQn                 = 28,     /*!< TIM2 global Interrupt          */
 TIM3_IRQn                 = 29,     /*!< TIM3 global Interrupt          */
 TIM4_IRQn                 = 30,     /*!< TIM4 global Interrupt          */
 I2C1_EV_IRQn                = 31,     /*!< I2C1 Event Interrupt       */
 I2C1_ER_IRQn                = 32,     /*!< I2C1 Error Interrupt           */
 I2C2_EV_IRQn                = 33,     /*!< I2C2 Event Interrupt          */
 I2C2_ER_IRQn                = 34,     /*!< I2C2 Error Interrupt           */
 SPI1_IRQn                 = 35,     /*!< SPI1 global Interrupt            */
 SPI2_IRQn                 = 36,     /*!< SPI2 global Interrupt          */
 USART1_IRQn               = 37,     /*!< USART1 global Interrupt          */
 USART2_IRQn              = 38,     /*!< USART2 global Interrupt        */
 USART3_IRQn              = 39,     /*!< USART3 global Interrupt        */
 EXTI15_10_IRQn        = 40,     /*!< External Line[15:10] Interrupts          */
 RTC_Alarm_IRQn        = 41,     /*!< RTC Alarm (A and B) through EXTI Line Interrupt */
 OTG_FS_WKUP_IRQn  = 42,  /*!< USB OTG FS Wakeup through EXTI line interrupt */
 TIM8_BRK_TIM12_IRQn = 43,  /*!< TIM8 Break Interrupt & TIM12 global interrupt */
```

• Vector table Exception number(Priority)지정(stm32f4xxx.h)

```
 TIM8_UP_TIM13_IRQn = 44,    /*!< TIM8 Update Interrupt & TIM13 global interrupt */
 TIM8_TRG_COM_TIM14_IRQn  = 45,     /*!< TIM8 Trigger and Commutation Interrupt
and TIM14 global interrupt */
 TIM8_CC_IRQn           = 46,     /*!< TIM8 Capture Compare Interrupt   */
DMA1_Stream7_IRQn   = 47,     /*!< DMA1 Stream7 Interrupt   */
 FSMC_IRQn                 = 48,     /*!< FSMC global Interrupt          */
 SDIO_IRQn                 = 49,     /*!< SDIO global Interrupt              */
 TIM5_IRQn                 = 50,     /*!< TIM5 global Interrupt              */
 SPI3_IRQn                 = 51,     /*!< SPI3 global Interrupt             */
 UART4_IRQn              = 52,     /*!< UART4 global Interrupt                */
 UART5_IRQn             = 53,     /*!< UART5 global Interrupt            */
 TIM6_DAC_IRQn       = 54,     /*!< TIM6 global & DAC1&2 underrun error  interrupts */
 TIM7_IRQn                 = 55,     /*!< TIM7 global interrupt                          */
 DMA2_Stream0_IRQn   = 56,     /*!< DMA2 Stream 0 global Interrupt  */
 DMA2_Stream1_IRQn   = 57,     /*!< DMA2 Stream 1 global Interrupt          */
 DMA2_Stream2_IRQn   = 58,     /*!< DMA2 Stream 2 global Interrupt           */
 DMA2_Stream3_IRQn   = 59,     /*!< DMA2 Stream 3 global Interrupt            */
 DMA2_Stream4_IRQn   = 60,     /*!< DMA2 Stream 4 global Interrupt           */
 ETH_IRQn                   = 61,     /*!< Ethernet global Interrupt              */
 ETH_WKUP_IRQn        = 62,     /*!< Ethernet Wakeup through EXTI line Interrupt */
 CAN2_TX_IRQn          = 63,     /*!< CAN2 TX Interrupt                            */
```

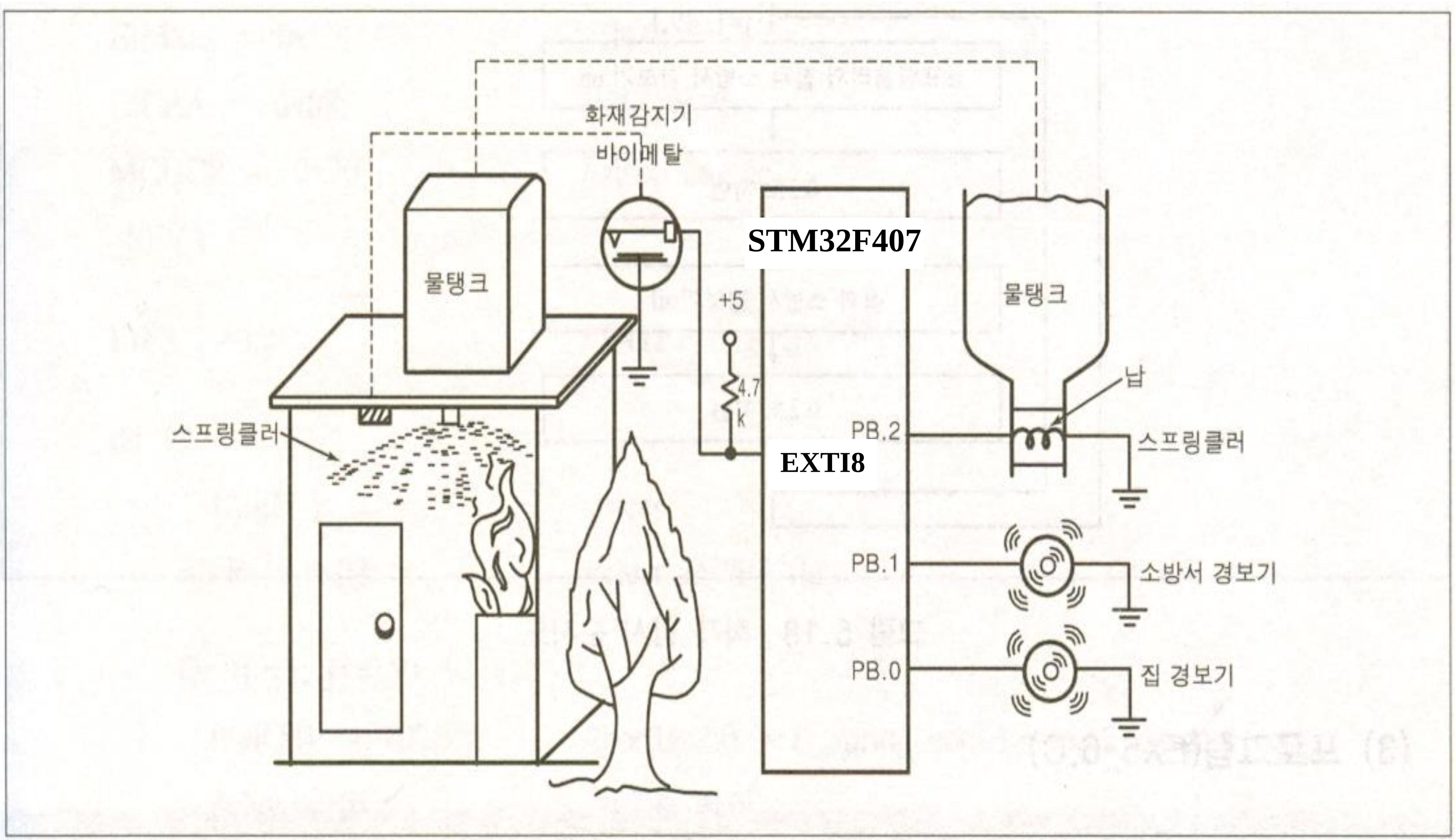
• Vector table Exception number(Priority)지정(stm32f4xxx.h)

```
  CAN2_RX0_IRQn          = 64,    /*!< CAN2 RX0 Interrupt      */
  CAN2_RX1_IRQn          = 65,    /*!< CAN2 RX1 Interrupt     */
  CAN2_SCE_IRQn          = 66,    /*!< CAN2 SCE Interrupt      */
  OTG_FS_IRQn            = 67,    /*!< USB OTG FS global Interrupt    */
  DMA2_Stream5_IRQn   = 68,    /*!< DMA2 Stream 5 global interrupt       */
  DMA2_Stream6_IRQn   = 69,    /*!< DMA2 Stream 6 global interrupt       */
  DMA2_Stream7_IRQn   = 70,    /*!< DMA2 Stream 7 global interrupt        */
  USART6_IRQn            = 71,    /*!< USART6 global interrupt          */
  I2C3_EV_IRQn           = 72,    /*!< I2C3 event interrupt               */
  I2C3_ER_IRQn           = 73,    /*!< I2C3 error interrupt              */
  OTG_HS_EP1_OUT_IRQn= 74,   /*!< USB OTG HS End Point 1 Out global interrupt */
  OTG_HS_EP1_IN_IRQn  = 75,   /*!< USB OTG HS End Point 1 In global interrupt   */
  OTG_HS_WKUP_IRQn   = 76,   /*!< USB OTG HS Wakeup through EXTI interrupt   */
  OTG_HS_IRQn             = 77,    /*!< USB OTG HS global interrupt                  */
  DCMI_IRQn             = 78,    /*!< DCMI global interrupt             */
  CRYP_IRQn             = 79,    /*!< CRYP crypto global interrupt    */
  HASH_RNG_IRQn       = 80,    /*!< Hash and Rng global interrupt         */
  FPU_IRQn              = 81     /*!< FPU global interrupt              */
#endif /* STM32F40_41xxx */
```

• 프로그램 예

```
void EXTI_Init(void)
{RCC->AHB1ENR   |= 0x80; // RCC_AHB1ENR GPIOH Enable
  RCC->APB2ENR   |= 0x4000;// Enable System Configuration Controller
Clock
 GPIOH->MODER   |= 0x0; // GPIOH PIN8~PIN15 Input mode (reset state)
 SYSCFG->EXTICR[2] |= 0x77; // EXTI8,9에 대한 소스 입력은 GPIOH로
설정

 EXTI->FTSR |= 0x100;       // Falling Trigger Enable
 EXTI->RTSR |= 0x200;       // Rising Trigger  Enable
 EXTI->IMR |= 0x300;        // EXTI8,9 인터럽트 mask

 NVIC->ISER[0] |= ( 1 << 23 );   // Enable Interrupt EXTI8,9 Vector table
Position 참조
}
void EXTI9_5_IRQHandler(void)  // EXTI 5~9 인터럽트 핸들러
{   }
```

* 외부 인터럽트(EXTI8) 응용예제 : 화재감시장치



화재 감시 시스템