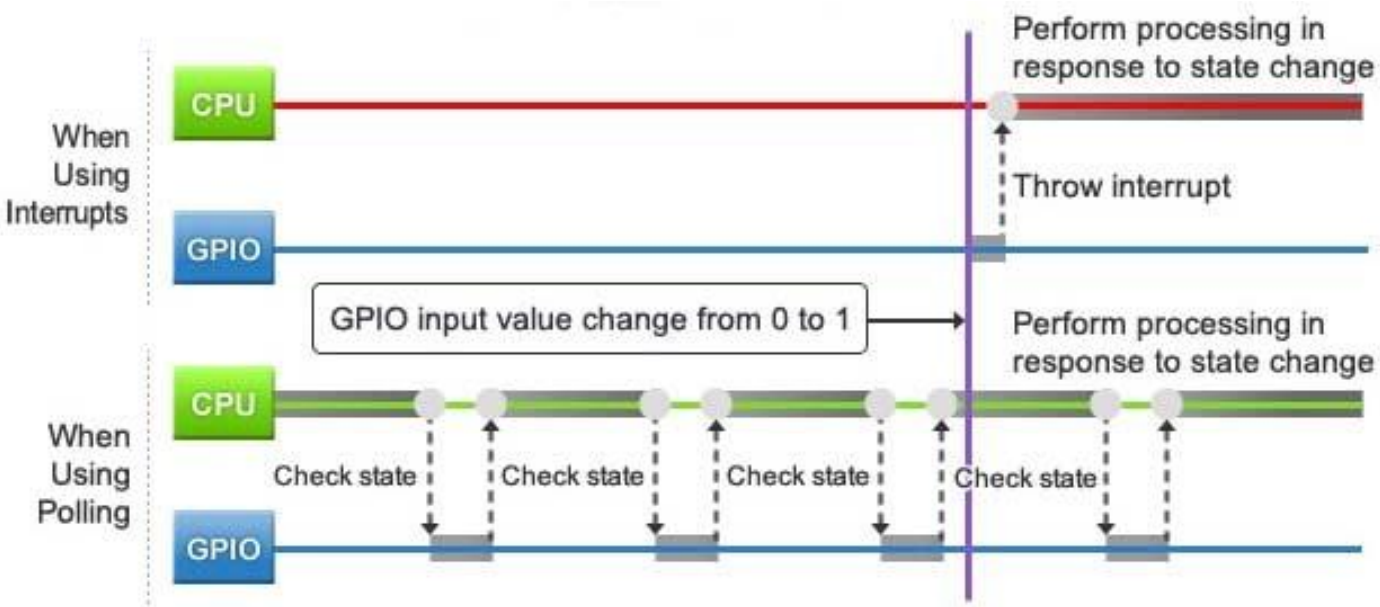
Iran University of
Science and Technology

Microcontroller

Department of Electrical Engineering
Iran University of Science and Technology

Interrupt/Polling



Interrupt

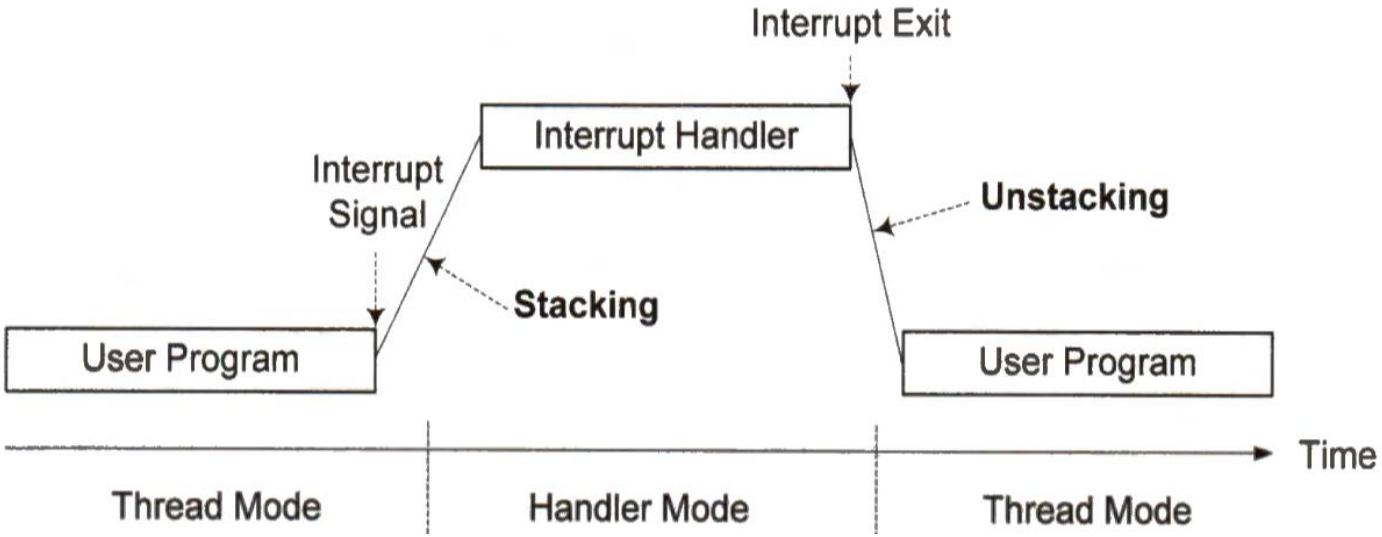


Fig 1. Automatic stacking and unstacking for interrupt handler

Fig 1. Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C

Interrupt Mechanism

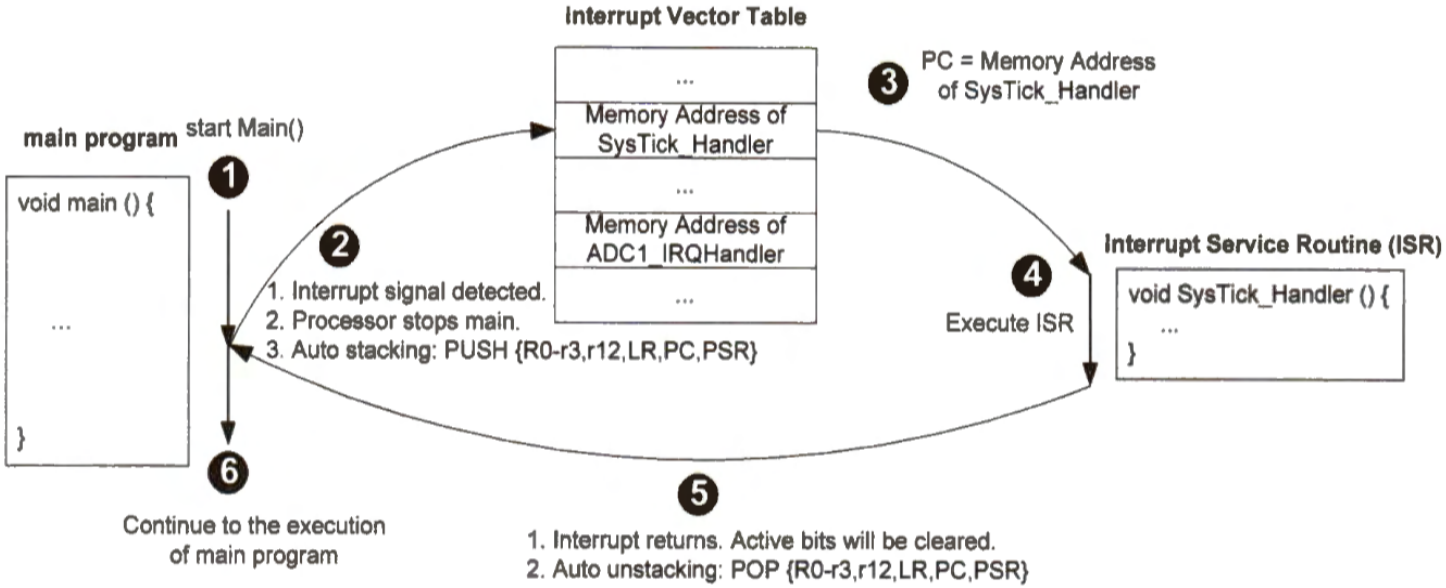
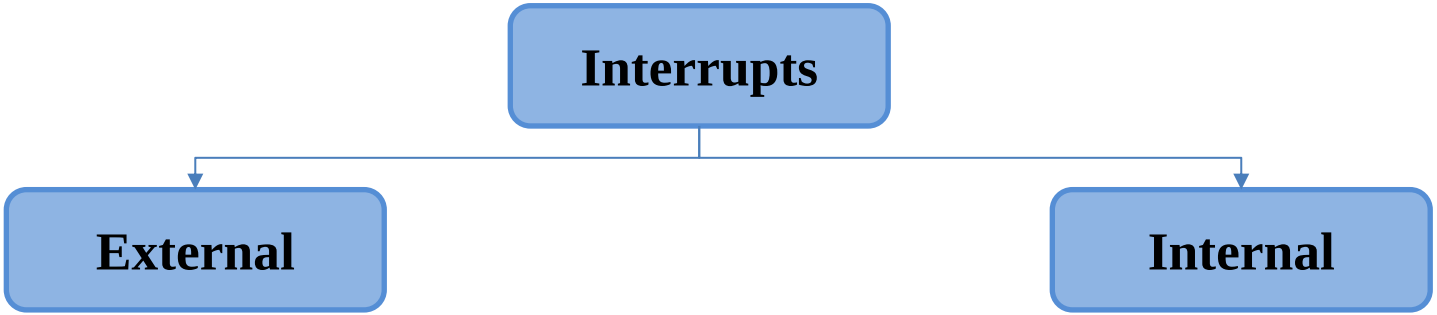


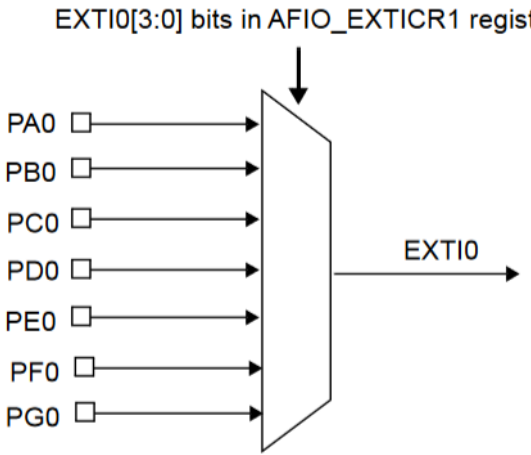
Fig 2. Steps of stacking and unstacking for interrupt handler

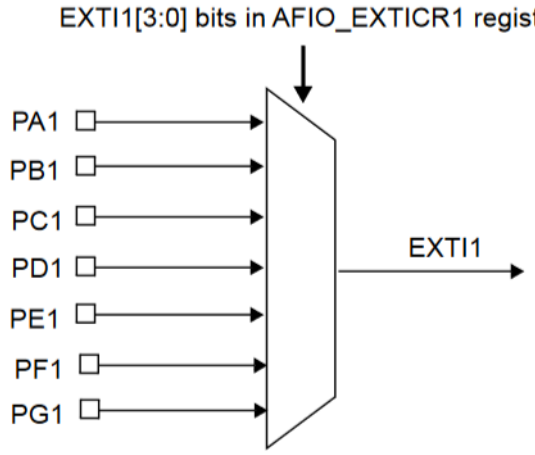
Fig 2. Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C

Interrupt

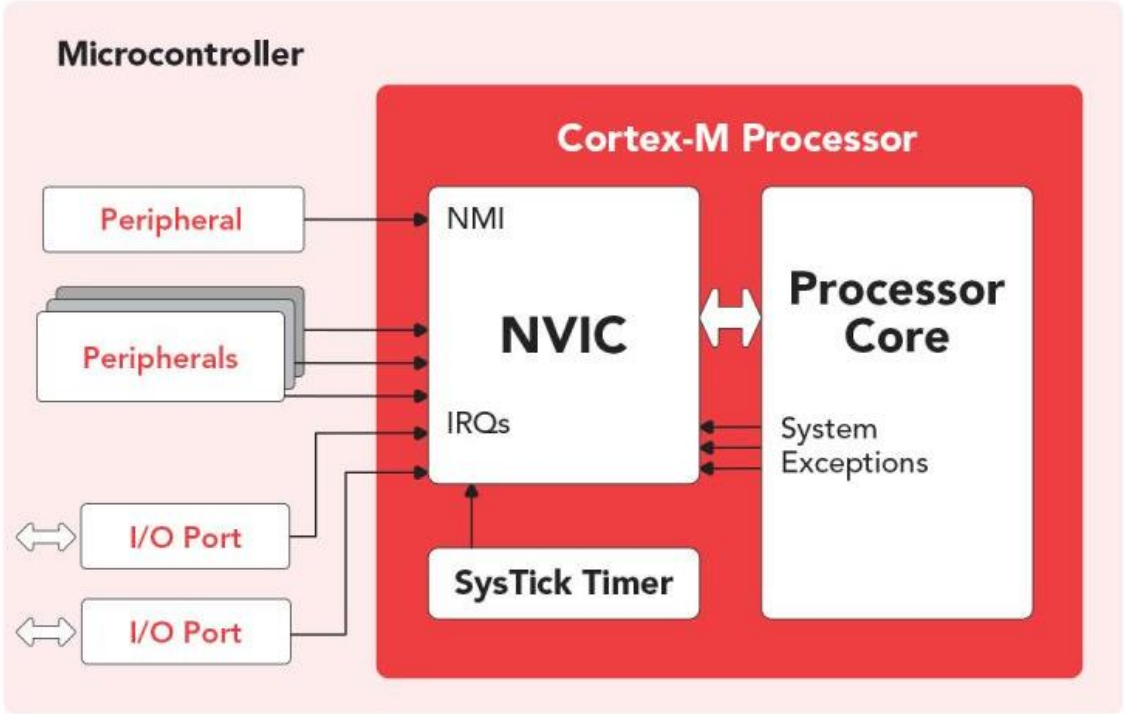


External interrupt/event GPIO mapping





NVIC


Microcontroller
Cortex-M Processor
Peripheral
Peripherals
I/O Port
I/O Port
NMI
NVIC
IRQs
SysTick Timer
Processor Core
System Exceptions

RCC_APB2ENR :IOPCEN

IO port C clock enable
0: IO port C clock disabled
1: IO port C clock enabled

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										TIM11 EN	TIM10 EN	TIM9 EN	Reserved		
										rw	rw	rw			

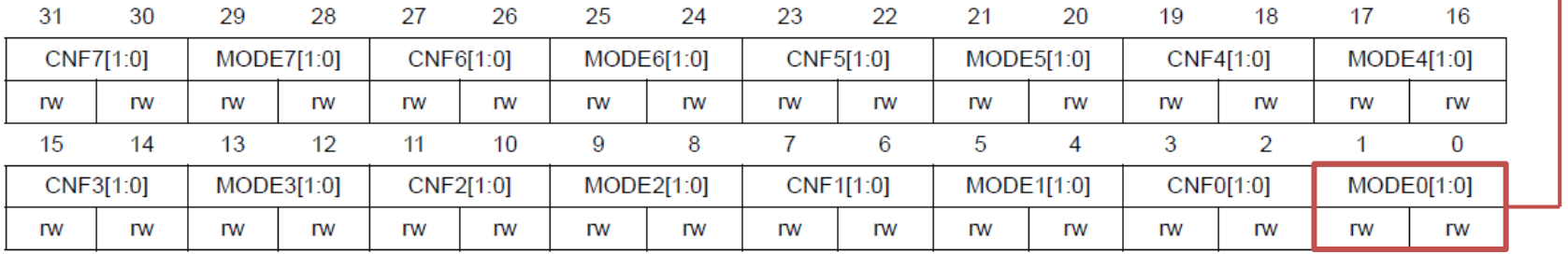
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3 EN	USART 1EN	TIM8 EN	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	IOPG EN	IOPF EN	IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	Res.	AFIO EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

RCC_APB2ENR :IOPCEN

```
RCC->APB2ENR |= RCC_APB2ENR_IOPCEN;
```

GPIOx_CRL :MODEy[1:0]

Port x mode bits (y= 0 .. 7)
00: **Input mode (reset state)**
01: Output mode, max speed 10 MHz.
10: Output mode, max speed 2 MHz.
11: Output mode, max speed 50 MHz.



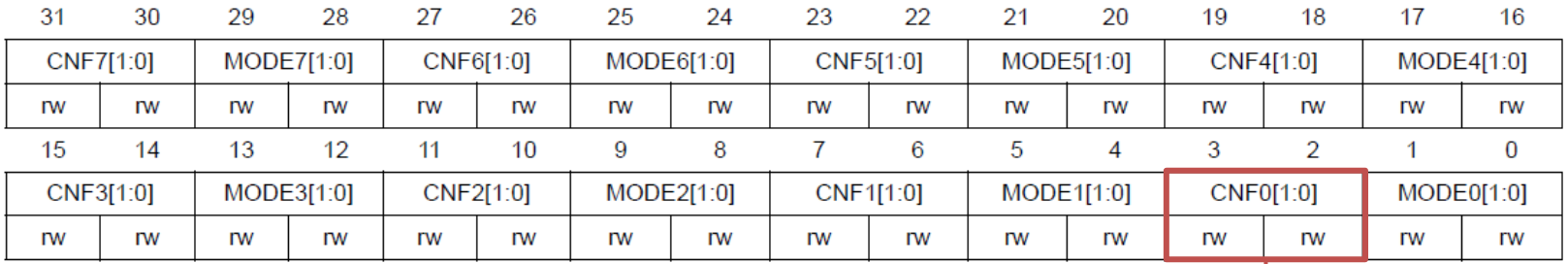
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]		MODE7[1:0]		CNF6[1:0]		MODE6[1:0]		CNF5[1:0]		MODE5[1:0]		CNF4[1:0]		MODE4[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3[1:0]		MODE3[1:0]		CNF2[1:0]		MODE2[1:0]		CNF1[1:0]		MODE1[1:0]		CNF0[1:0]		MODE0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

GPIOx_CRL :MODEy[1:0]

```
RCC->APB2ENR |= RCC_APB2ENR_IOPCEN;
GPIOC->CRL &= ~GPIO_CRL_MODE0;
```

GPIOx_CRL :CNFy[1:0]

In input mode (MODE[1:0]=00)
00: Analog mode
01: Floating input (reset state)
10: **Input with pull-up / pull-down**
11: Reserved



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]		MODE7[1:0]		CNF6[1:0]		MODE6[1:0]		CNF5[1:0]		MODE5[1:0]		CNF4[1:0]		MODE4[1:0]	
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
CNF3[1:0]		MODE3[1:0]		CNF2[1:0]		MODE2[1:0]		CNF1[1:0]		MODE1[1:0]		CNF0[1:0]		MODE0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

GPIOx_CRL :CNFy[1:0]

```
RCC->APB2ENR |= RCC_APB2ENR_IOPCEN;
GPIOC->CRL |= GPIO_CRL_MODE0;
GPIOC->CRL &= ~GPIO_CRL_CNF0;
GPIOC->CRL |= ~GPIO_CRL_CNF0_1;
```

Pull-down/Pull-up

IUST
Iran University of Science and Technology

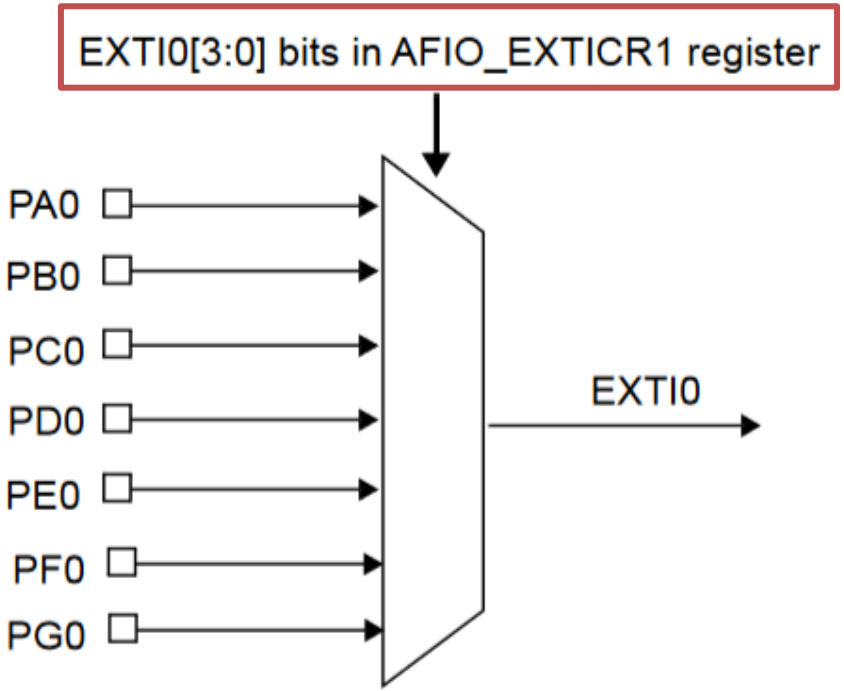
Table 20. Port bit configuration table

Configuration mode		CNF1	CNF0	MODE1	MODE0	PxODR register
General purpose output	Push-pull	0	0	01 10 11 see *Table 21*		0 or 1
	Open-drain		1			0 or 1
Alternate Function output	Push-pull	1	0			Don't care
	Open-drain		1			Don't care
Input	Analog	0	0	00		Don't care
	Input floating		1			Don't care
	Input pull-down	1	0			0
	Input pull-up					1

GPIOx_ODR

```
GPIOC->CRL &= ~GPIO_CRL_CNF0;
GPIOC->CRL |= ~GPIO_CRL_CNF0_1;
GPIOC->ODR |=  GPIO_ODR_ODR0;
```

AFIO_EXTICR1



IUST
Iran University of
Science and Technology

RCC_APB2ENR :AFION

```
RCC->APB2ENR |= RCC_APB2ENR_AFIOEN;
```

AFIO_EXTICR1

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

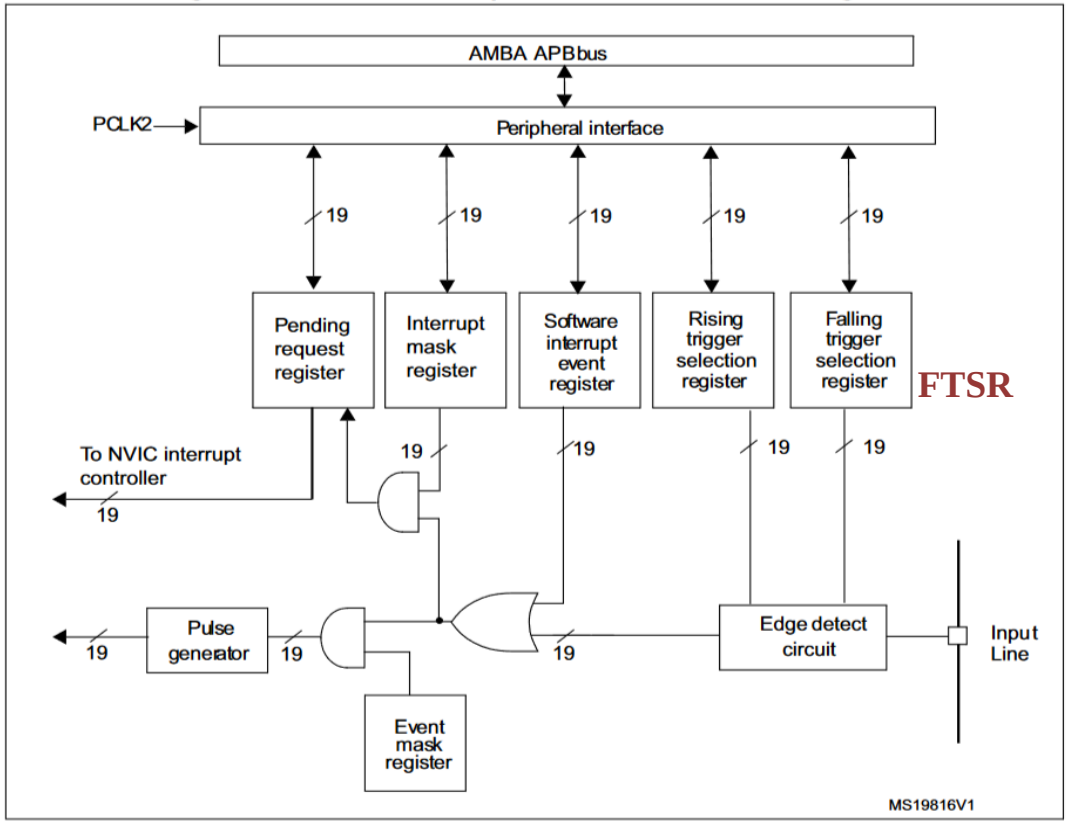
These bits are written by software to select the source input for EXTIx external interrupt.
Refer to *Section 10.2.5: External interrupt/event line mapping*

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

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

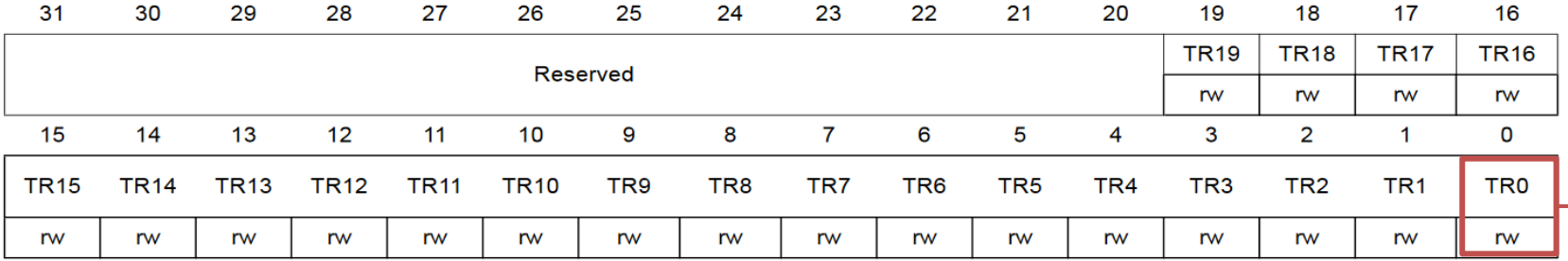
EXT Interrupt controller

Figure 20. External interrupt/event controller block diagram



Falling trigger selection register (EXTI_FTSR)

Falling trigger event configuration bit of line x
0: Falling trigger **disabled** for input line
1: Falling trigger **enabled** for input line.



EXTI_FTSR

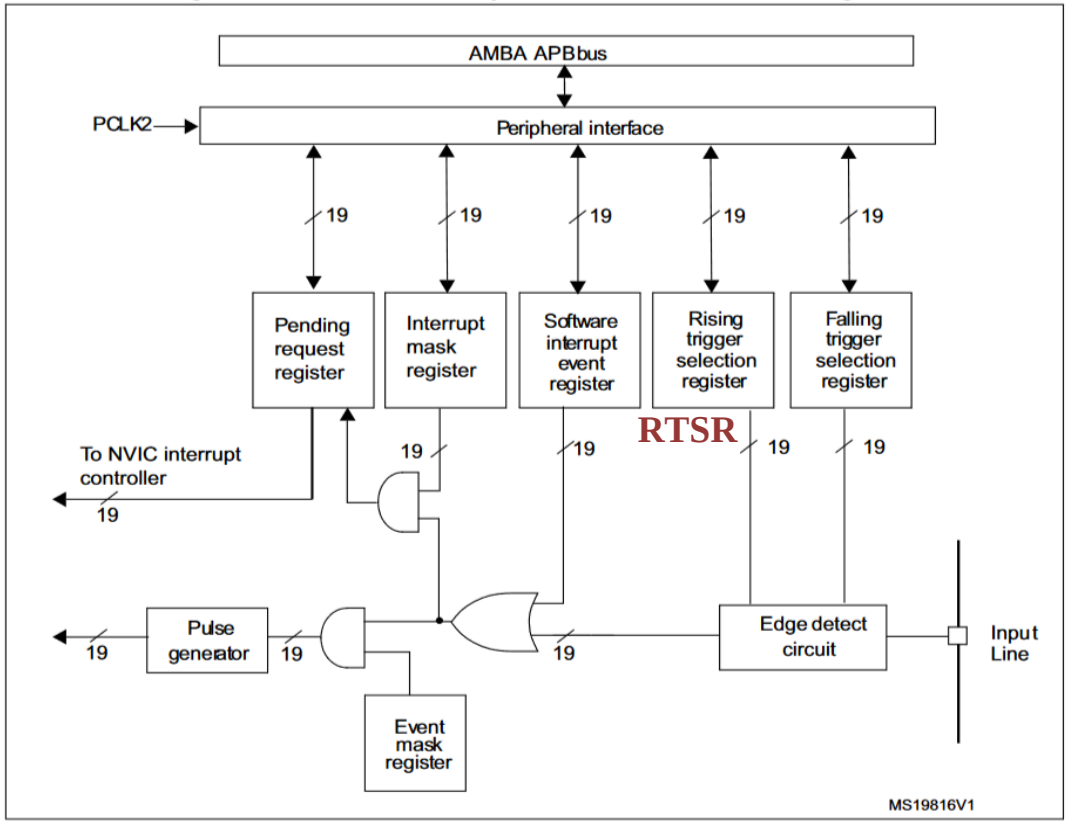
```
EXTI->FTSR |=  EXTI_FTSR_TR0;
```

OR

```
EXTI->FTSR &=  ~EXTI_FTSR_TR0;
```

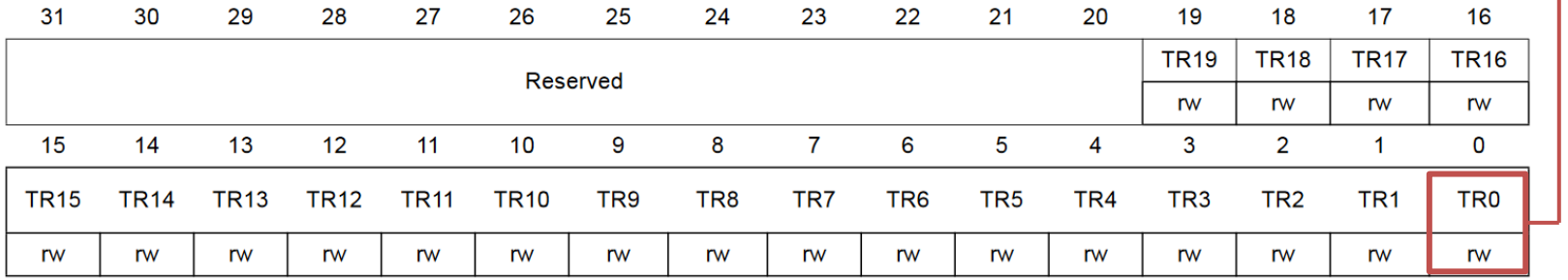
EXT Interrupt controller

Figure 20. External interrupt/event controller block diagram



Rising trigger selection register (EXTI_RTSR)

Rising trigger event configuration bit of line x
0: Rising trigger **disabled** for input line
1: Rising trigger **enabled** for input line.



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

EXTI_RTSR

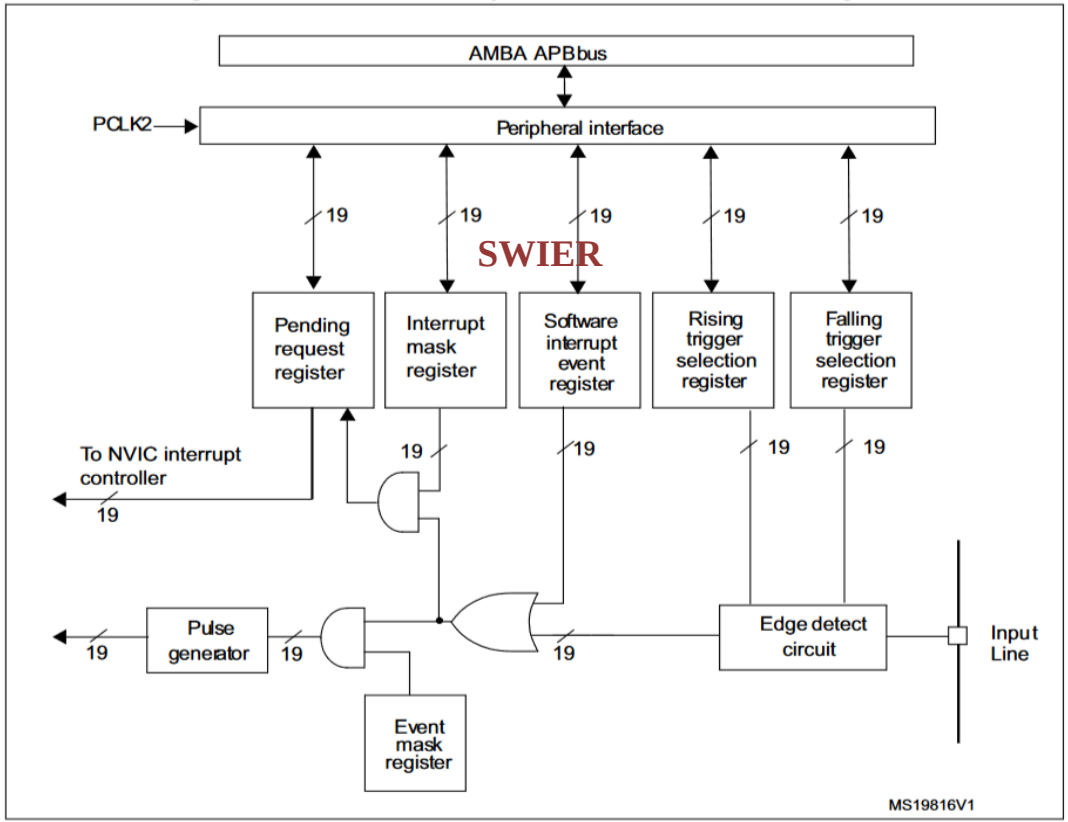
```
EXTI->RTSR &=  ~EXTI_RTSR_TR0;
```

OR

```
EXTI->RTSR |=  EXTI_RTSR_TR0;
```

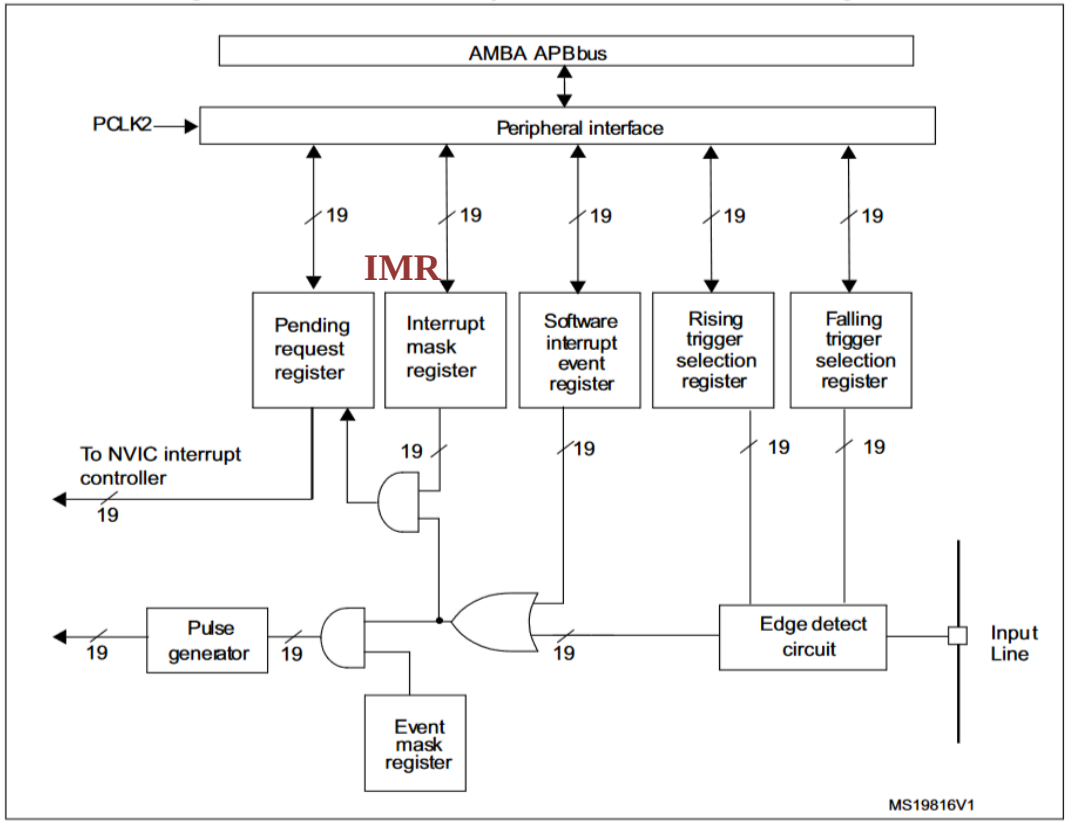
EXT Interrupt controller

Figure 20. External interrupt/event controller block diagram



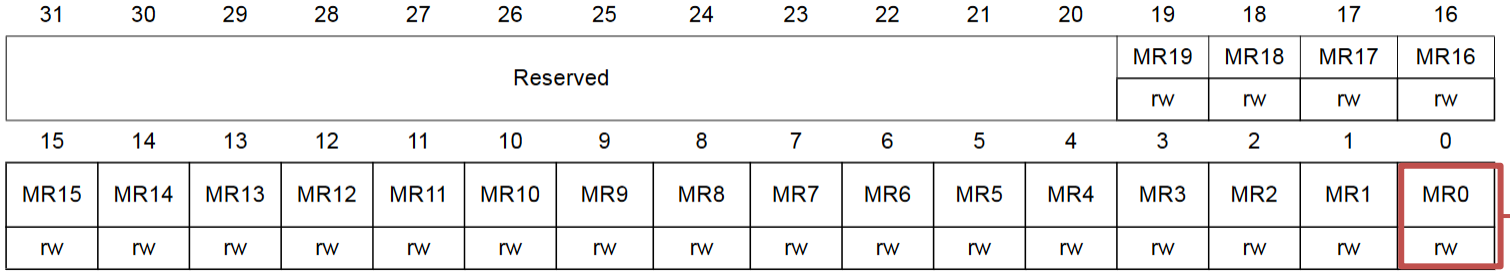
EXT Interrupt controller

Figure 20. External interrupt/event controller block diagram



Interrupt mask register (EXTI_IMR)

Interrupt Mask on line x
0: Interrupt request from Line x is masked
1: Interrupt request from Line x is **not masked**



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												MR19	MR18	MR17	MR16
												rw	rw	rw	rw

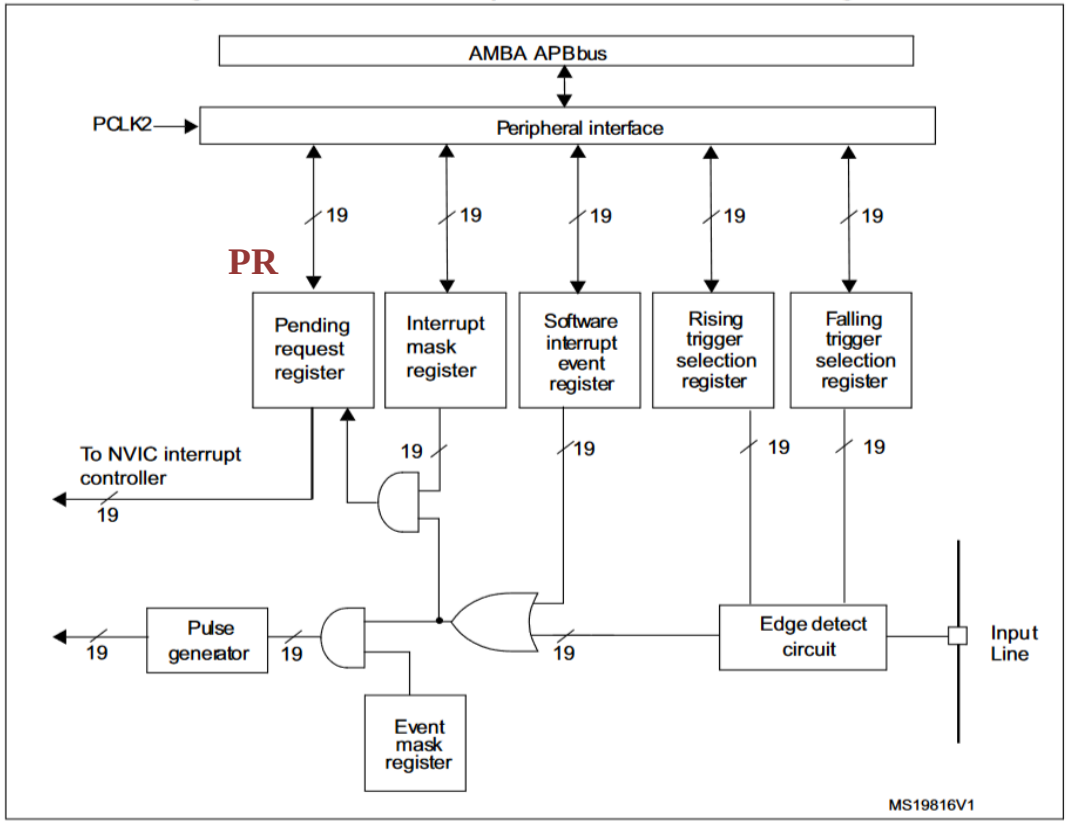
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

EXTI_IMR

```
EXTI->IMR |=  EXTI_IMR_MR0;
```

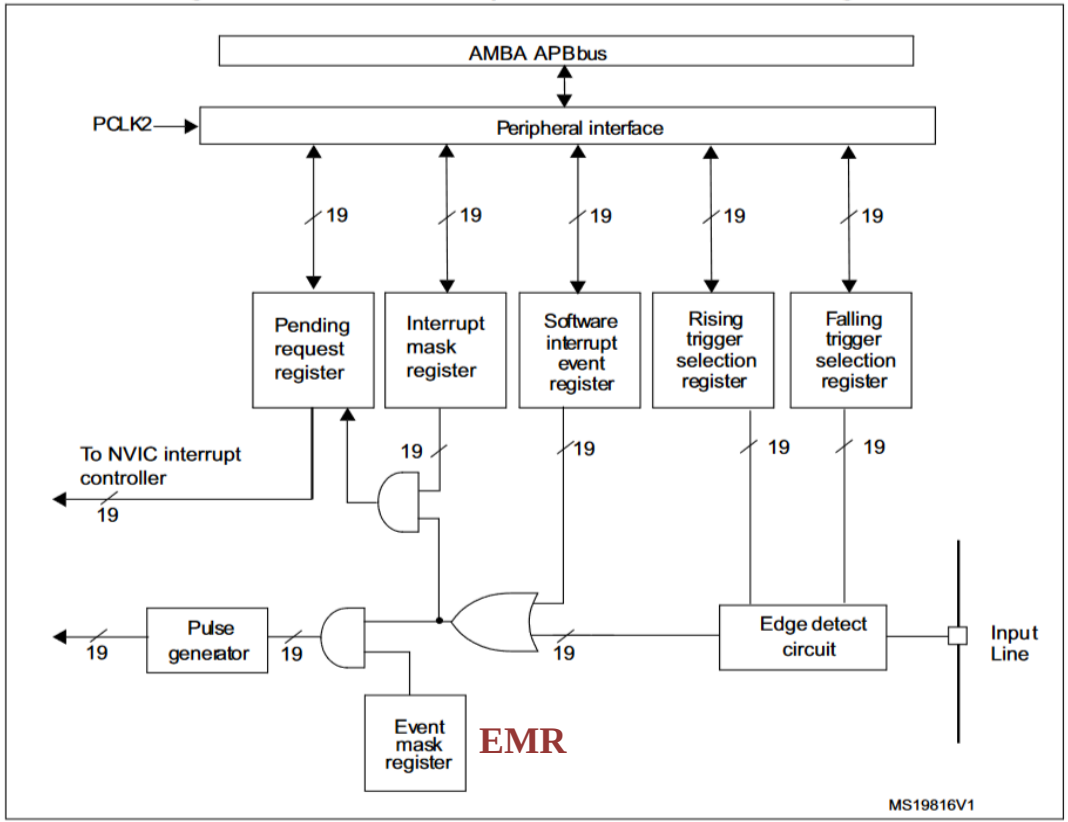
EXT Interrupt controller

Figure 20. External interrupt/event controller block diagram



EXT Interrupt controller

Figure 20. External interrupt/event controller block diagram



Contact us

tarasarpoolaki@gmail.com

amin.feizi751381@gmail.com

www.linkedin.com/in/Tara-Sarpoolaki

www.linkedin.com/in/aminfeizishahri