External Interrupt

Lecture 6

External interrupts

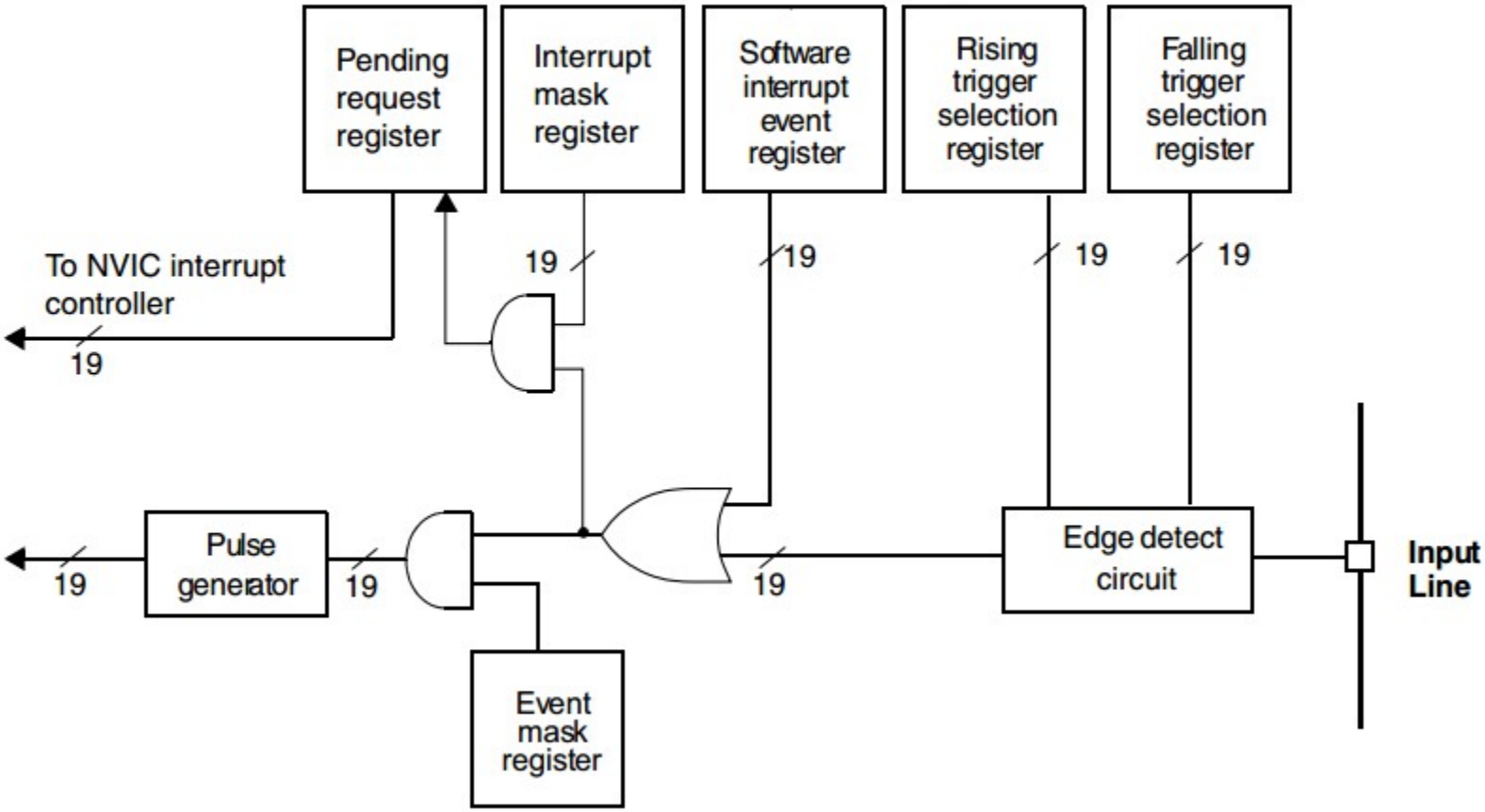
Is an electronic alerting signal sent to the processor from an external device outside the processor, like an external peripheral, ADC peripheral, UART peripheral. For example, when we press a button, they trigger hardware interrupts which cause the processor to read the button value.

Interrupt trigger

In general, when an input pin is connected to an external device to be used for interrupt, we have 5 choices for trigger point. They are:

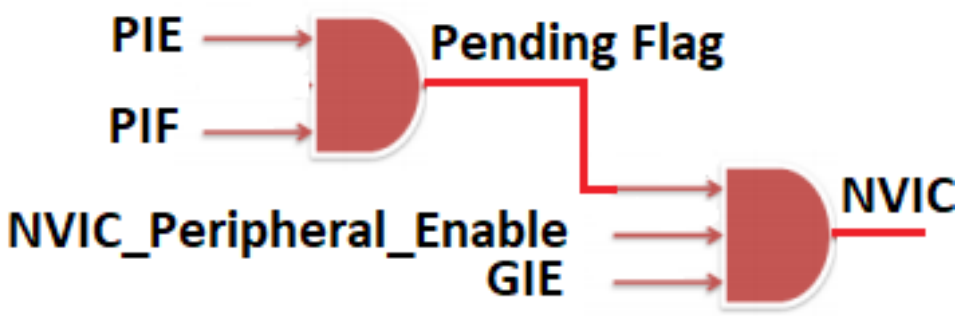
- low-level trigger (active Low level),
- high-level trigger (active High level),
- rising-edge trigger (positive-edge going from Low to High),
- falling-Edge trigger (negative-edge going from High to Low),
- Both edge (rising and falling) trigger.

External Interrupt Controller



External Interrupt Controller

There are 4 conditions to serve the external interrupt for example:



External interrupt peripheral

1- Global interrupt enable (GIE).

2- Peripheral interrupt enable (PIE).

3- Peripheral interrupt Flag (PIF).

4- NVIC Peripheral Enable.

Register Description

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												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

Bits 31:20 Reserved, must be kept at reset value (0).

Bits 19:0 **MRx:** Interrupt Mask on line x

0: Interrupt request from Line x is masked
1: Interrupt request from Line x is not masked

Note: Bit 19 is used in connectivity line devices only and is reserved otherwise.

Register Description

Event mask register (EXTI_EMR)

Address offset: 0x04
Reset value: 0x0000 0000

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

Bits 31:20 Reserved, must be kept at reset value (0).

Bits 19:0 **MRx:** Event mask on line x

0: Event request from Line x is masked
1: Event request from Line x is not masked

Note: Bit 19 is used in connectivity line devices only and is reserved otherwise.

Register Description

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												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

Bits 31:20 Reserved, must be kept at reset value (0).

Bits 19:0 **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 line.

Note: Bit 19 is used in connectivity line devices only and is reserved otherwise.

Register Description

Software interrupt event register (EXTI_SWIER)

Address offset: 0x10
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												SWIER 19	SWIER 18	SWIER 17	SWIER 16
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIER 15	SWIER 14	SWIER 13	SWIER 12	SWIER 11	SWIER 10	SWIER 9	SWIER 8	SWIER 7	SWIER 6	SWIER 5	SWIER 4	SWIER 3	SWIER 2	SWIER 1	SWIER 0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:20 Reserved, must be kept at reset value (0).

Bits 19:0 **SWIERx:** Software interrupt on line x

If the interrupt is enabled on this line in the EXTI_IMR, writing a '1' to this bit when it is set to '0' sets the corresponding pending bit in EXTI_PR resulting in an interrupt request generation.

This bit is cleared by clearing the corresponding bit of EXTI_PR (by writing a 1 into the bit).

Note: Bit 19 used in connectivity line devices and is reserved otherwise.

External Interrupt Programming Steps

Step 1 : EXTI_VidInit()

RISING_EDGE:	SET_BIT(RTSR)
FALLING_EDGE:	SET_BIT(FTSR)
ON_CHANGE:	SET_BIT(EXTI → RTSR , Copy_EXTILine)
	SET_BIT(EXTI → FTSR , Copy_EXTILine)

Step 2 : EXTI_VidEnableEXTI()

SET_BIT(IMR)

Step 3 : void EXTI_VidSoftWareTrigger()

SET_BIT(SWIER)

Step 4 : EXTI0_VidSetCallBack()

EXTI0_CallBack = Pointer To Function

Step 5 : EXTI0_IRQHandler()

EXTI0_CallBack()
SET_BIT(PR)

Step 6 : EXTI_VidDisableEXTI()

CLR_BIT(IMR)

IMT SCHOOL

www.imtschool.com

ww.facebook.com/imaketechnologyschool/