Nested Interrupt Vector Controller

Lecture 3

Introduction to ARM Processor

Definition of Exception

Exceptions are events which are generated asynchronously either from inside the core itself or from the external peripherals. In general any even that disturb the normal behavior of the processor is an exception.

Interrupts are used with exceptions from the external peripherals. In other words interrupt is nothing but an exception external to the processor core.

ARM Cortex M3/M4 Supports 255 exceptions

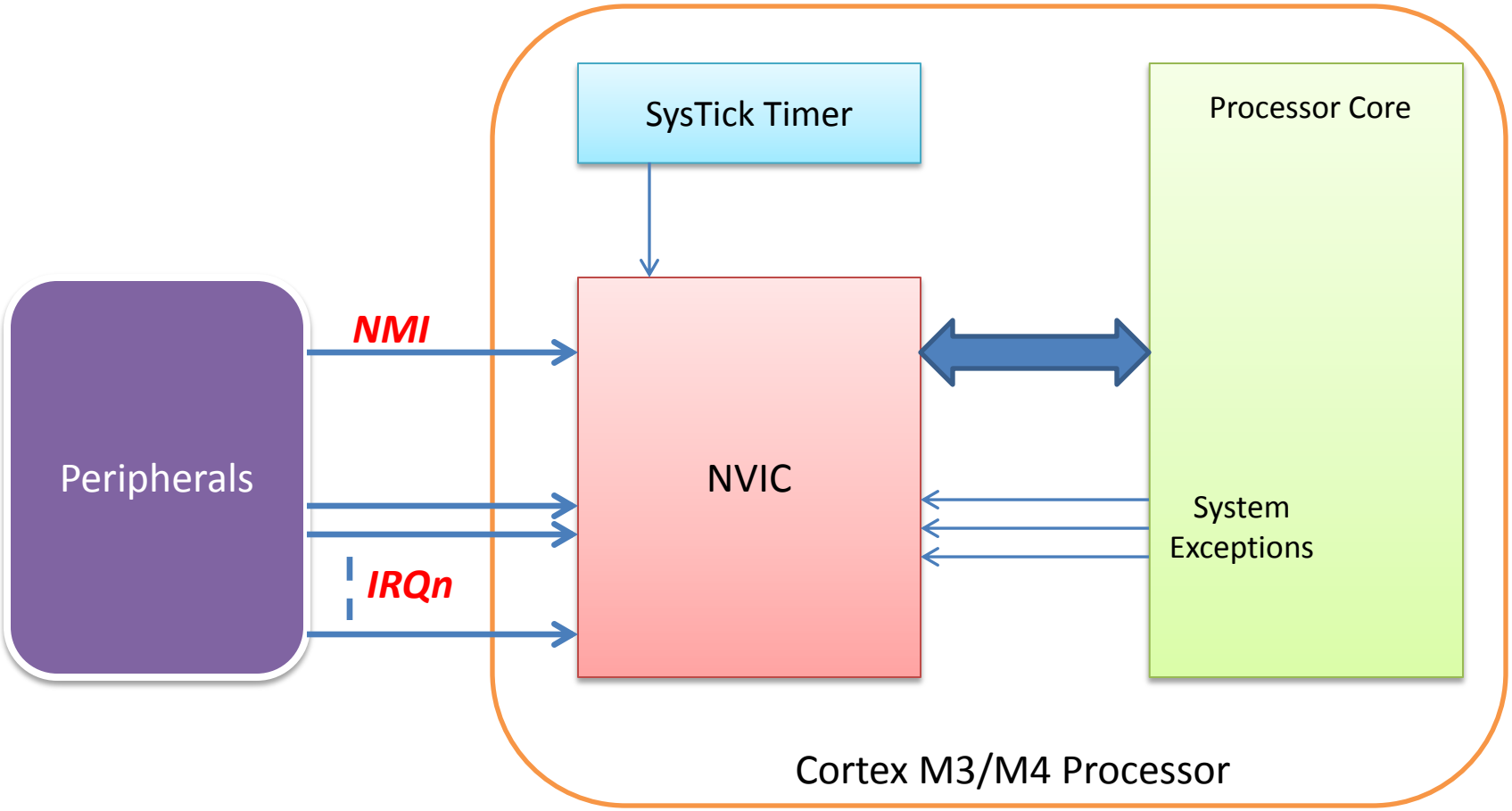
Nested Vectored Interrupt Controller

The Nested Vectored Interrupt Controller ***NVIC*** is an integrated part of the Cortex M3/M4 processor. It is closely linked to the CPU core logic and Its control registers are accessible as ***memory mapped***. It provides the following features:

1. Nested interrupt support
2. Vectored interrupt support
3. Dynamic priority changes support
4. Reduction of interrupt latency
5. Interrupt masking

The NVIC controls the interrupts as well as the system exceptions. The NVIC can be accessed in the System Control Space (SCS) address range, which is memory location ***0xE000E000***. Most of the interrupt control/status registers are accessible only in ***privileged mode***, except the Software Trigger Interrupt register (STIR), which can be set up to be accessible in user mode.

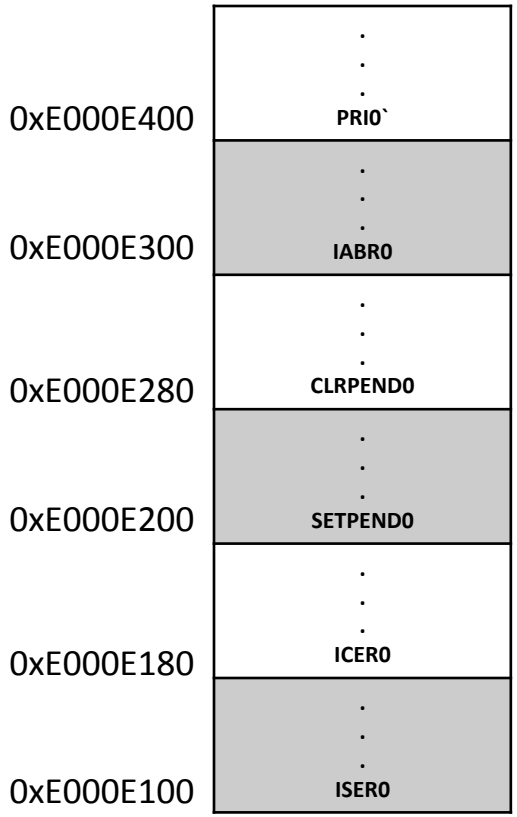
Nested Vectored Interrupt Controller



Interrupt Configuration

Each interrupt has a set of basic configurations through NVIC which are:

1. Enable
2. Clear Enable
3. Set Pending
4. Clear Pending
5. Active status
6. Priority level



Enable Interrupt Register

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[31:0]**: Interrupt set-enable bits.

Write:

0: No effect
1: Enable interrupt

Read:

0: Interrupt disabled
1: Interrupt enabled.

Note

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.

Enable Clear Enable Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRENA[31:16]															
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
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRENA[15:0]															
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 31:0 **CLRENA[31:0]**: Interrupt clear-enable bits.

Write:

0: No effect

1: Disable interrupt

Read:

0: Interrupt disabled

1: Interrupt enabled.

Note

Writing 1 to an ICPR bit does not affect the active state of the corresponding interrupt.

Set Pending Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SETPEND[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
SETPEND[15:0]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

Bits 31:0 **SETPEND[31:0]**: Interrupt set-pending bits

Write:

0: No effect

1: Changes interrupt state to pending

Read:

0: Interrupt is not pending

1: Interrupt is pending

Note

Writing 1 to the ISPR bit corresponding to an interrupt that is pending:

– has no effect.

Writing 1 to the ISPR bit corresponding to a disabled interrupt:

– sets the state of that interrupt to pending.

Clear Pending Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRPEND[31:16]															
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
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRPEND[15:0]															
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 31:0 **CLRPEND[31:0]**: Interrupt clear-pending bits

Write:

0: No effect

1: Removes the pending state of an interrupt

Read:

0: Interrupt is not pending

1: Interrupt is pending

Note

A bit reads as 1 if the status of the corresponding interrupt is active or active and pending.

Interrupt Active Bit Register

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

Bits 31:0 **ACTIVE[31:0]**: Interrupt active flags

0: Interrupt not active

1: Interrupt active

Note

A bit reads as 1 if the status of the corresponding interrupt is active or active and pending.

Interrupt Priority

1- Define Number of ***Groups*** and Number of ***Sub Priorities***

The AIRCR provides priority grouping control for the exception model, endian status for data accesses, and reset control of the system.

To write to this register, you must write 0x5FA to the VECTKEY field, otherwise the processor ignores the write.

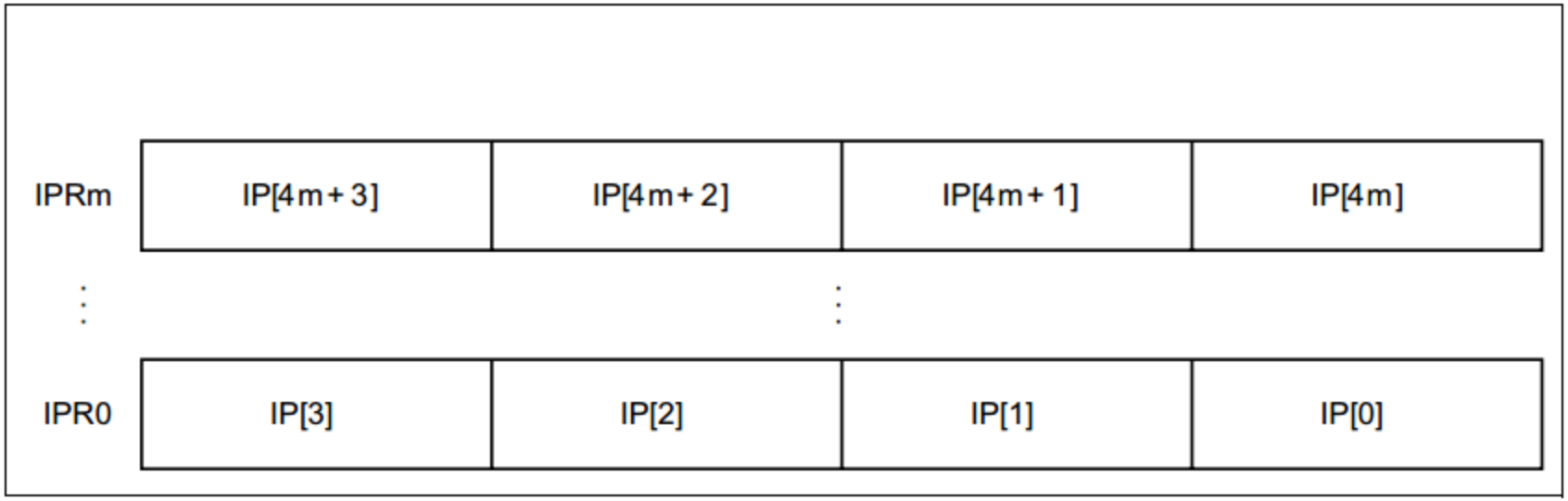
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VECTKEYSTAT[15:0](read)/ VECTKEY[15:0](write)															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENDIANESS	Reserved				PRIGROUP			Reserved					SYSRESETREQ	VECTCLRACTIVE	VECTRESET
r					rw	rw	rw						w	w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

PRIGROUP [2:0]	Interrupt priority level value, PRI_*N*[7:4]			Number of	
	Binary point[1]	Group priority bits	Subpriority bits	Group priorities	Sub priorities
0b011	0bxxxx	[7:4]	None	16	None
0b100	0bxxx.y	[7:5]	[4]	8	2
0b101	0bxx.yy	[7:6]	[5:4]	4	4
0b110	0bx.yyy	[7]	[6:4]	2	8
0b111	0b.yyyy	None	[7:4]	None	16

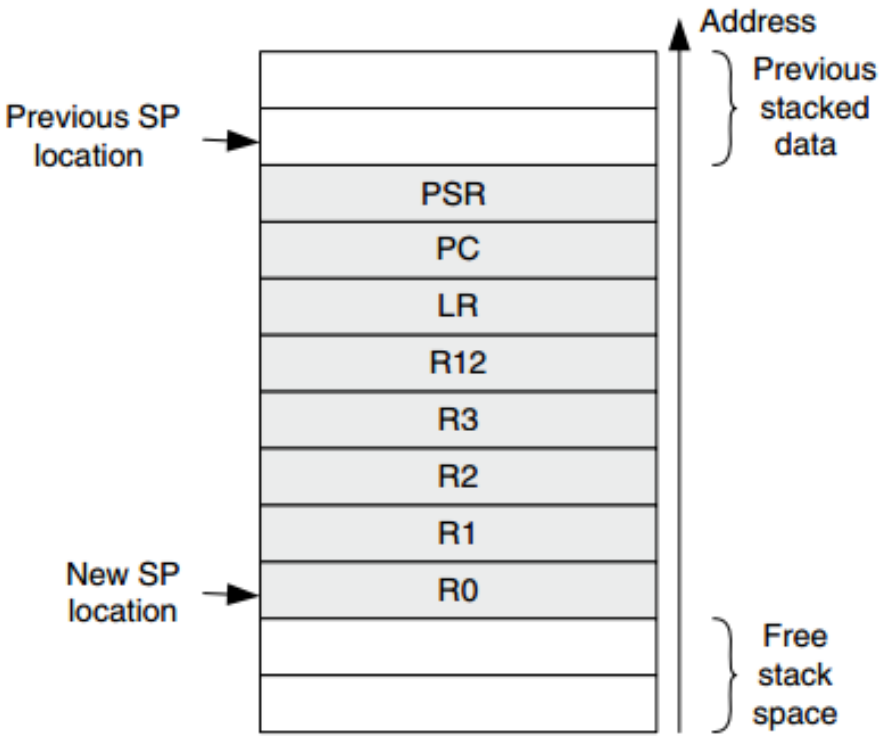
Interrupt Priority

1- Assign a priority for each interrupt

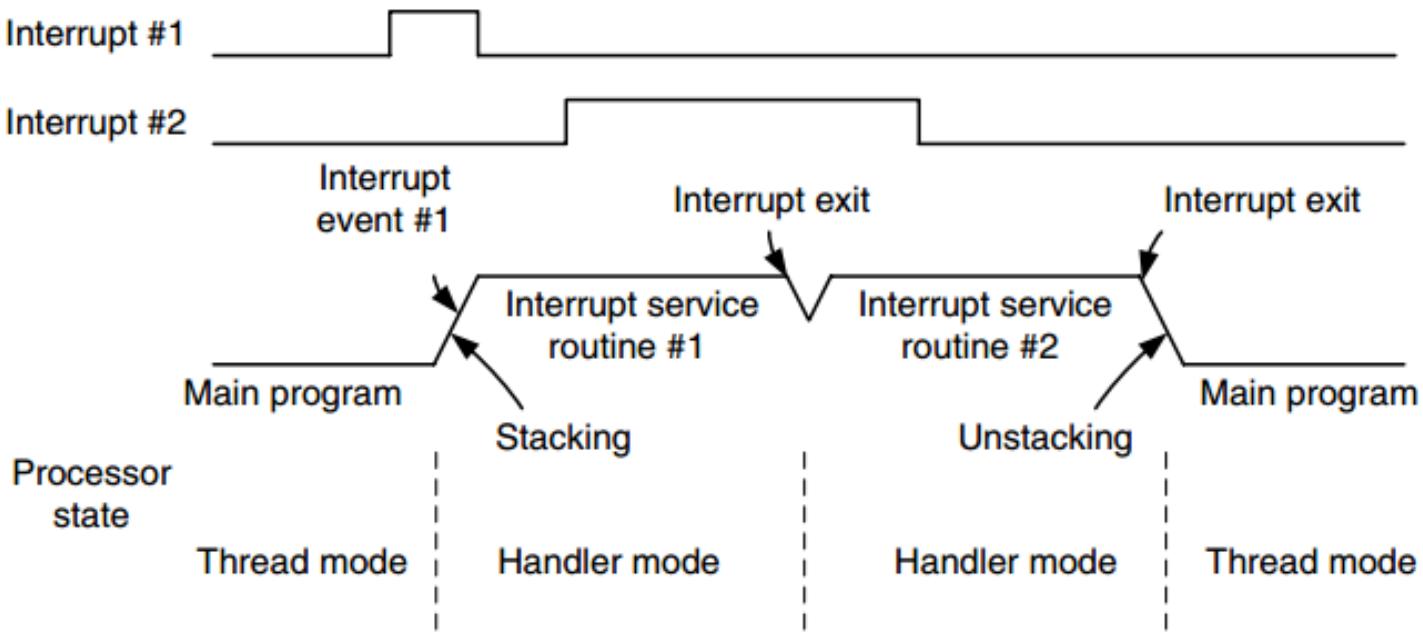
The IPR0-IPR16 registers provide a 4-bit priority field for each interrupt. These registers are byte-accessible. Each register holds four priority fields, that map to four elements in the interrupt priority array IP[0] to IP[67]



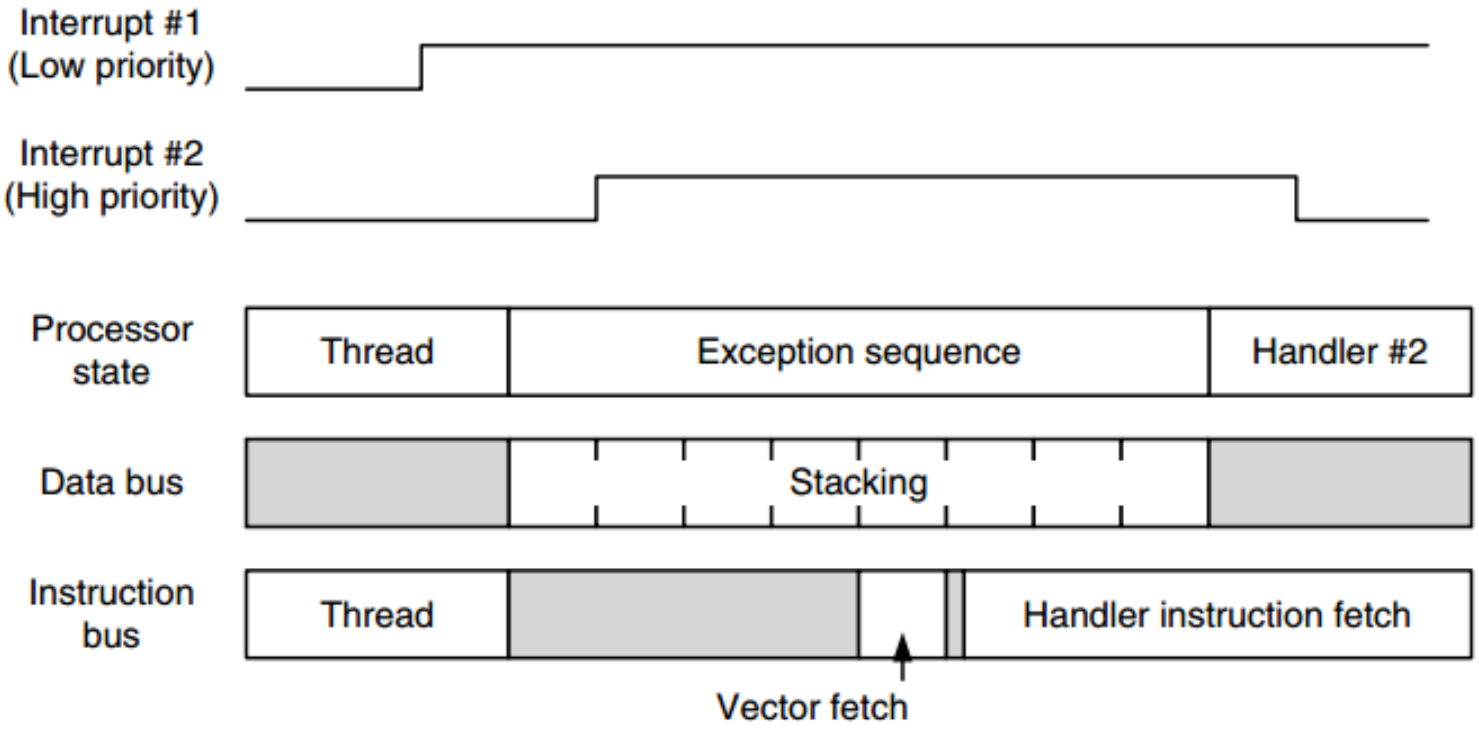
Stacking and Unstacking



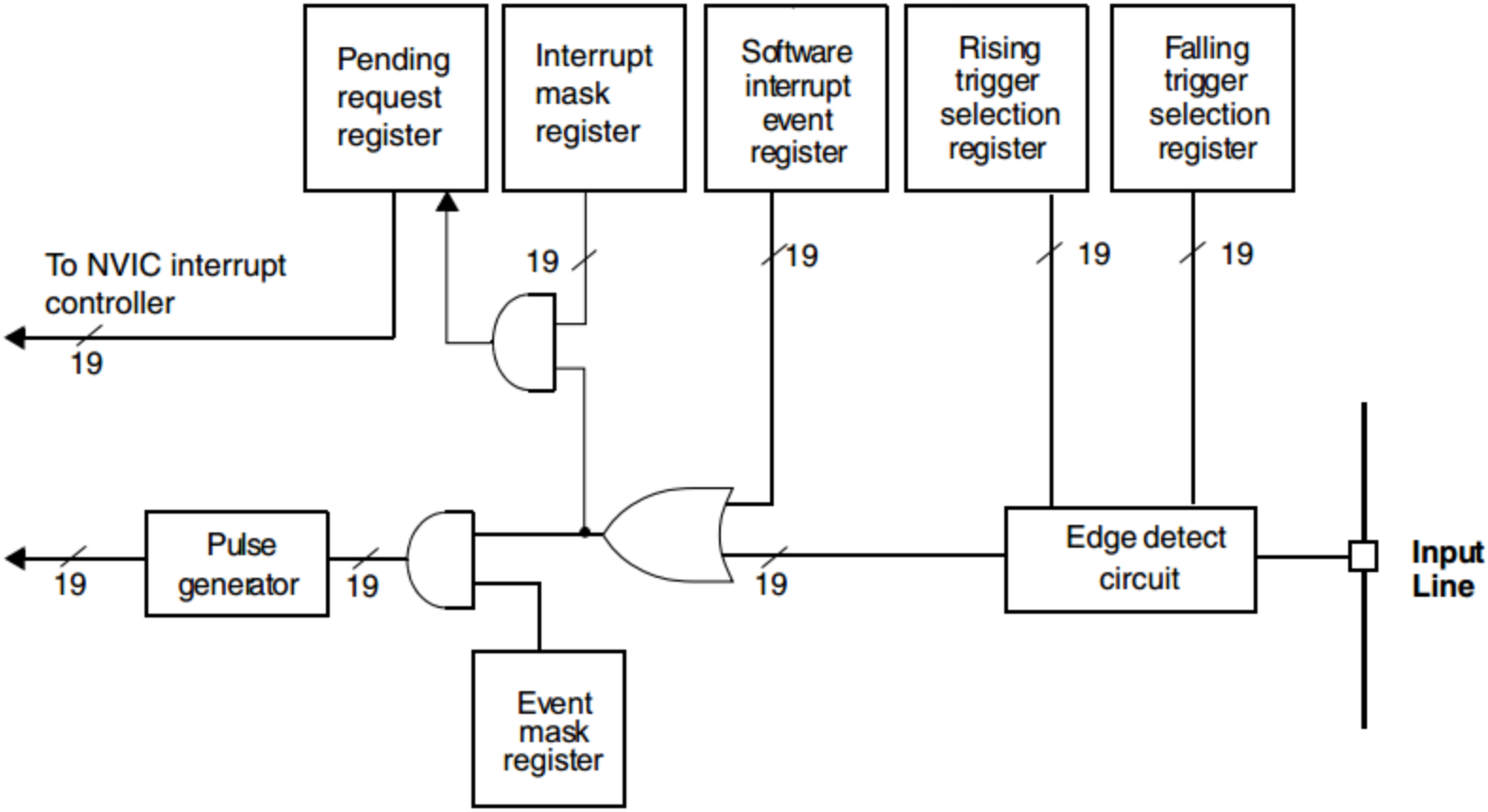
TAIL-CHAINING



LATE ARRIVALS



External Interrupt Controller



IMT SCHOOL

www.imtschool.com

ww.facebook.com/imaketechnologyschool/