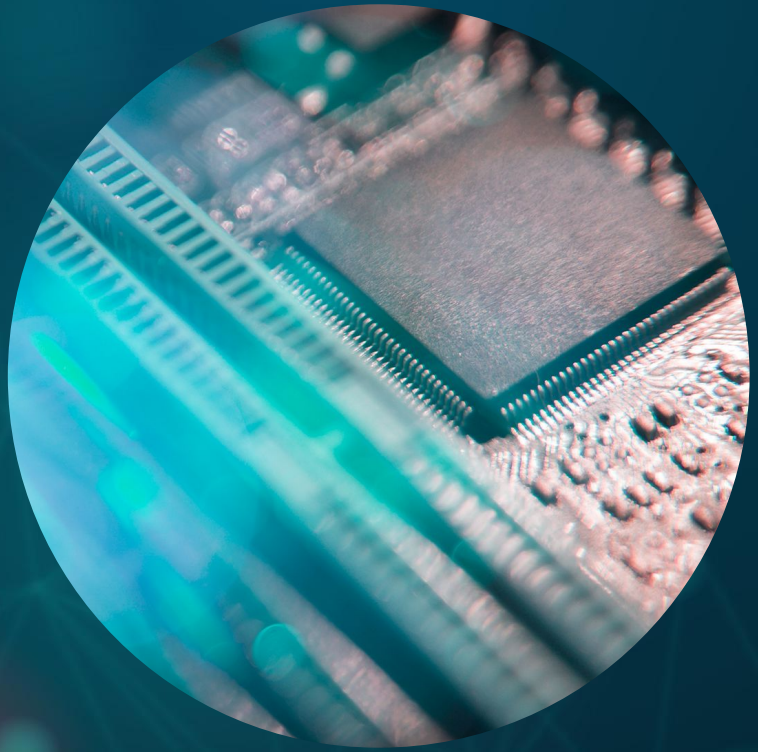
IMT SCHOOL

ARM Based Microcontroller

NVIC

Lecture 4



Advanced RISC Machines

IMT SCHOOL

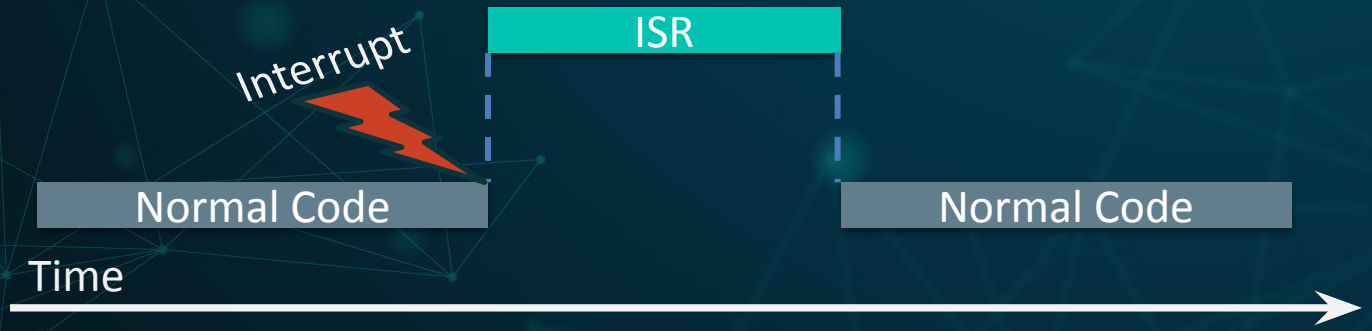
Introduction 01

Nested Vector Interrupt Controller

IMT SCHOOL

Interrupt

Interrupt: Is a combination of software and hardware to force the processor to stop its current activity and begin to execute a particular piece of code called an interrupt service routine (ISR). Which is a response to a specific event generated by hardware change (internally or externally) or even software.



INTERRUPT VS POLLING

We use a telephone as an example to compare polling and interrupt efficiency. Suppose you are expecting a call. In polling, you pick up your telephone every 10 seconds to check whether there is anyone on the line calling you (without ring). In the interrupt you continue to perform whatever tasks you are supposed to complete while waiting for the telephone to ring.

IMT SCHOOL

INTERRUPT VS POLLING

Parameters of Comparison	Polling	Interrupt
Meaning	The concurrent process in which the device is surveyed to ensure any need for servicing is known as polling.	The hardware mechanism that notifies the device whenever it requires servicing and needs to be attended by software is an interrupt.
Type of	Protocol	Hardware mechanism
Occurrence	At regular intervals of time.	At any instance of time.
Device	The device is mended by the CPU.	The device is overhauled by interrupt handlers.
Form of indication	Command ready bit is used to inform the device.	The request line is used to inform the device.

POLLING

The process in which the device is surveyed to ensure if there is any need for servicing is known as polling. It is a coeval procedure. It is also considered a polled I/O or software-driven I/O. Low-level hardware is involved in this process. It can take a lot of time if multiple devices are to be checked and surveyed.

The process of polling takes place in two steps, namely host actions and controller actions. In host actions, the busy-bit is completely read by the host. Once the busy bit is cleared, the command-ready bit is set to 1 by the host. Before setting the command-ready bit to 1, the command is written to the command register by the host.

IMT SCHOOL

POLLING

The second step is the controller actions. In this step, the command-ready bit is already set to 1, and this is noted by the controller. The controller then puts the busy bit to 1. It is the controller who reads and performs the I/O operations after reading the command register and ensuring the presence of a write bit in it. The operations are cleared and are shown to be successful.

Polling can be used to control the information sequence of elements involved in measuring contexts and also their execution. It can be used to administer the time of the processor along with other resources in multitasking operating systems. Although the probability that the data can be wasted is much higher in polling.

IMT SCHOOL

Interrupt

The hardware mechanism which notifies the device whenever it requires servicing and needs to be attended by software is known as an interrupt. It notifies the computer whenever it requires to be attended by software. It makes sure the ongoing task is completed on time. It is a type of hardware mechanism.

There are two types of interrupts, namely, hardware interrupt and software interrupt. In hardware interrupt, external hardware notifies the condition of the hardware. In software interrupt, a processor requests the interrupt when a certain condition takes place. A particular interrupt handler ought to be associated with a software interrupt.

IMT SCHOOL

Interrupt

The software interrupts can be either intentional or unexpected. Special instruction is induced to intentionally cause a software interrupt. Program execution errors that take place can cause unexpected software interruptions. Based on the triggering methods, there are two types of interrupts, namely level-triggered interrupt and edge-triggered interrupt.

The occurrence of interrupts result in increased efficiency of the CPU, and the waiting time of the CPU is decreased. The wastage of the instruction cycle is also stopped because of interrupts. Some disadvantages of interrupts are that the CPU has to carry out a lot of tasks as it has to return to its previous program.

IMT SCHOOL

INTERRUPT VS POLLING

- In polling, the performance of the microcontroller is poor. On the contrary, in interrupt, the performance of the microcontroller is great.
- In polling, all the processor cycles are wasted while checking the device. On the other hand, in interrupt when a certain device interrupts a processor, the processor is disturbed.
- In polling, the CPU is put on hold. On the other hand, in interrupt, the CPU is called if required.
- Polling takes place at regular intervals of time. On the other hand, interrupt takes place at any instance of time.
- Polling is a type of protocol. On the other hand, interrupt is a type of hardware mechanism.

IMT SCHOOL

Interrupt Types

02

Nested Vector Interrupt Controller

IMT SCHOOL

Interrupt Types

Interrupts can be categorized according to many things

Hardware Interrupts:
An electronic signal sent from an external device or hardware to communicate with the processor indicating that it requires immediate attention. For example, strokes from a keyboard or an action from a mouse invoke hardware interrupts causing the CPU to read and process it. So it arrives asynchronously and during any point of time while executing an instruction.

Hardware interrupts are classified into two types

- Maskable Interrupts
- Non-Maskable Interrupts

IMT SCHOOL

Interrupt Types

Maskable Interrupts – Processors have to interrupt mask register that allows enabling and disabling of hardware interrupts. Every signal has a bit placed in the mask register. If this bit is set, an interrupt is enabled & disabled when a bit is not set, or vice versa. Signals that interrupt the processors through these masks are referred to as masked interrupts.

Non-maskable Interrupts (NMI) – The NMIs are the highest priority activities that need to be processed immediately and under any situation, such as a timeout signal generated from a watchdog timer.

IMT SCHOOL

Interrupt Types

Software Interrupts

The processor itself requests a software interrupt after executing certain instructions or if particular conditions are met. These can be a specific instruction that triggers an interrupt such as subroutine calls and can be triggered unexpectedly because of program execution errors, known as exceptions or traps.

IMT SCHOOL

Interrupt Types

Interrupts can also be categorized according to interrupt source position:

The interrupt source could be outside the processor which is called external interrupt, and could be inside the processor which is called internal interrupt.

- **External Interrupt**

Is an electronic alerting signal sent to the processor from an external device outside the processor.

IMT SCHOOL

Interrupt Types

Internal Interrupt:
Is inside the processor divided to

Trap like:
Division by Zero
Stack Overflow
Incorrect Op-Code

Software interrupt:
which is a specific assembly instruction called (SWI) that fires an interrupt.
Is caused either by an exceptional condition or a special instruction in the instruction set which causes an interrupt when it is executed by the processor.

IMT SCHOOL

Interrupt Handling

03

Nested Vector Interrupt Controller

IMT SCHOOL

Vectored Interrupt

Vectored Interrupts

- Devices that use vectored interrupts are assigned an interrupt vector. This is a number that identifies a particular interrupt handler. The ISR address of this interrupts is fixed and is known to CPU.
- When the device interrupts the CPU branches to the particular ISR.
- The microprocessor jumps to the specific service routine.
- When the microprocessor executes the call instruction, it saves the address of the next instruction on the stack.
- At the end of the service routine, the RET instruction returns the execution to where the program was interrupted.
- All 8051 interrupts are vectored interrupts.

IMT SCHOOL

Non-Vectored Interrupt

Non-vectored Interrupt

- Non-vectored interrupt is an interrupt that has a common ISR, which is common to all non-vectored interrupts in the system. Address of this common ISR is known to the CPU.
- The interrupts which don't have fixed memory location for transfer of control from normal execution.
- The address of the memory is sent along with the interrupt.
- The CPU crucially does not know which device caused the interrupt without polling each I/O interface in a loop.
- Once the interrupt occurs, the system must determine which device, of all the devices associated actually interrupted.

IMT SCHOOL

Difference between Them

A vectored interrupt is where the CPU actually knows the address of the interrupt service routine in advance. All it needs is that the interrupting device sends its unique vector through a data bus and through its I/O interface to the CPU. The CPU takes this vector, checks an interrupt table in memory and then carries out the correct ISR for that device. Therefore, the vectored interrupt allows the CPU to be able to know that ISR to carry out in software (memory)

A non-vectored interrupt is where the interrupting device never sends an interrupt vector. An interrupt is received by the CPU and it jumps the program counter to a fixed address in hardware. This is typically a hard coded ISR which is device agnostic. The CPU crucially does not know which device caused the interrupt without polling each O/I interface in a loop and checking the status register of each I/O interface to find the one with status “interrupt created”.

IMT SCHOOL

Difference between Them

Vectored interrupts	Non Vectored interrupts

ISR (external interrupt)
{
// some codes
}
ISR (ADC interrupt)
{
// some codes
}
	ISR ( )
	{
	if (external interrupt flag==1)
	{
	// some codes
	}
	if (ADC interrupt flag==1)
	{
	// some codes
	}
	}

Interrupt vs Function call

04

Nested Vector Interrupt Controller

IMT SCHOOL

Interrupt vs Function call

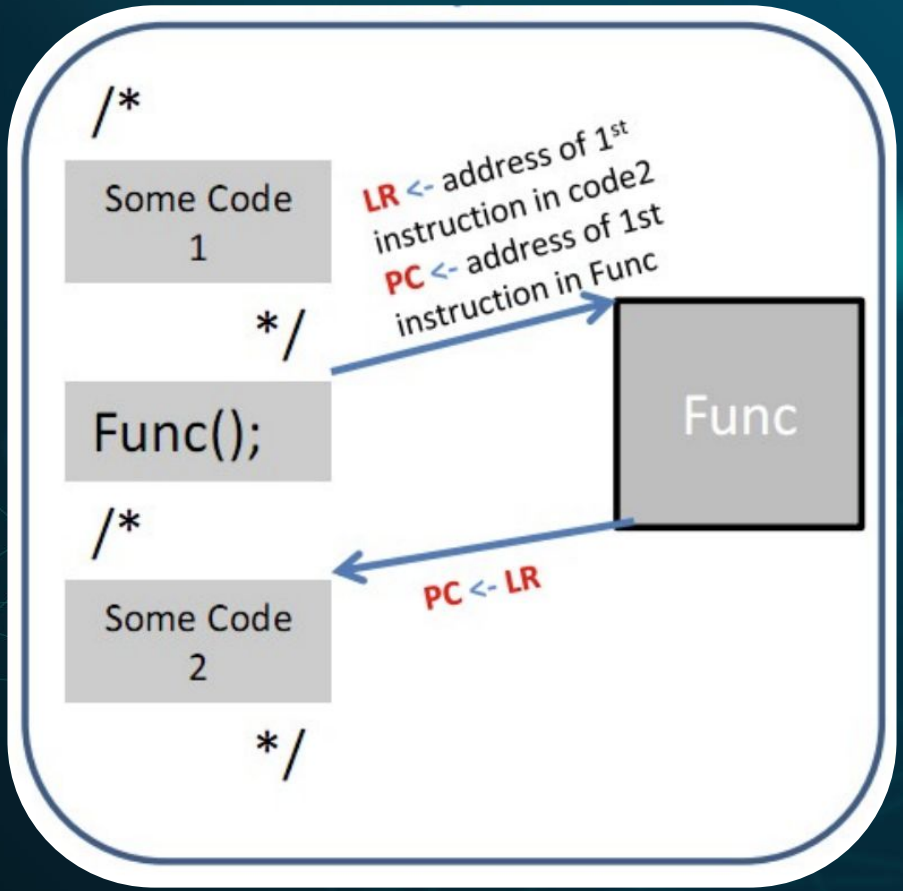
Difference between interrupt and a function call is:

The context switching in the Function call is known before so the compiler is responsible for the context switching but in the Interrupt we do not have any idea when the interrupt is coming?

So in function call the context switching is fully supported by software (compiler) but in the interrupt is fully supported by hardware (processor).

IMT SCHOOL

Interrupt vs Function call



Function call & Link Register

void *main* (void)

```
; main assembly program
;*********************
__main  PROC

; Initialize R0
MOV   R0, #0x01

; Call the function
BL    __func

;Loop forever
B     __main

ENDP
```

void *func* (void)

```
; func assembly program
;*********************
__func  PROC

;Check if R0 equals 1 or not
;and save 0 or 1 to R1
TST       R0, #0x01
MOVEQ     R1, #0
MOVNE     R1, #1

; Return from the function
BX    LR

ENDP
```

STM32 Is AWESOME

Session LAb

Create NVIC Driver

Time To Code

THANKS!

Do you have any question?

www.imtschool.com

www.facebook.com/imaketechnologyschool/

This material is developed by IMTSchool for educational use only

All copyrights are reserved