

# Types of Interrupts and How to Handle Interrupts

## ***Interrupts:***

In early years of computing processor has to wait for the signal for processing, so processor has to check each and every hardware and software program in the system if it has any signal to process. This method of checking the signal in the system for processing is called polling method. In this method the processor has to waste number of clock cycles just for checking the signal in the system, by this processor will become busy unnecessarily. If any signal came for the process, processor will take some time to process the signal due to the polling process in action. So system performance also will be degraded and response time of the system will also decrease.

So to overcome this problem engineers introduced a new mechanism, in this mechanism processor will not check for any signal from hardware or software but instead hardware/software will only send the signal to the processor for processing. The signal from hardware or software should have highest priority because processor should leave the current process and process the signal of hardware or software. This mechanism of processing the signal is called interrupt of the system.

## ***What is an Interrupt?***

Interrupt is a signal which has highest priority from hardware or software which processor should process its signal immediately.

## ***Types of Interrupts:***

Although interrupts have highest priority than other signals, there are many types of interrupts but basic types of interrupts are

1. **Hardware Interrupts:** If the signal for the processor is from external device or hardware is called hardware interrupts. Example: from keyboard we will press the key to do some action this pressing of key in keyboard will generate a signal which is given to the processor to do action, such interrupts are called hardware interrupts. Hardware interrupts can be classified into two types they are

- **Maskable Interrupt:** The hardware interrupts which can be delayed when a much highest priority interrupt has occurred to the processor.
  - **Non Maskable Interrupt:** The hardware which cannot be delayed and should process by the processor immediately.
2. **Software Interrupts:** Software interrupt can also divided in to two types. They are
- **Normal Interrupts:** the interrupts which are caused by the software instructions are called software instructions.
  - **Exception:** unplanned interrupts while executing a program is called Exception. For example: while executing a program if we got a value which should be divided by zero is called a exception.

#### **Classification of Interrupts According to Periodicity of Occurrence:**

1. **Periodic Interrupt:** If the interrupts occurred at fixed interval in timeline then that interrupts are called periodic interrupts
2. **Aperiodic Interrupt:** If the occurrence of interrupt cannot be predicted then that interrupt is called aperiodic interrupt.

#### **Classification of Interrupts According to the Temporal Relationship with System Clock:**

1. **Synchronous Interrupt:** The source of interrupt is in phase to the system clock is called synchronous interrupt. In other words interrupts which are dependent on the system clock. Example: timer service that uses the system clock.
2. **Asynchronous Interrupts:** If the interrupts are independent or not in phase to the system clock is called asynchronous interrupt.

## ***Interrupt Handling:***

We know that instruction cycle consists of fetch, decode, execute and read/write functions. After every instruction cycle the processor will check for interrupts to be processed if there is no interrupt is present in the system it will go for the next instruction cycle which is given by the instruction register.

If there is an interrupt present then it will trigger the interrupt handler, the handler will stop the present instruction which is processing and save its configuration in a register and load the program counter of the interrupt from a location which is given by the interrupt vector table. After

processing the interrupt by the processor interrupt handler will load the instruction and its configuration from the saved register, process will start its processing where it's left. This saving the old instruction processing configuration and loading the new interrupt configuration is also called as context switching.

The interrupt handler is also called as Interrupt service routine (ISR). There are different types of interrupt handler which will handle different interrupts. For example for the clock in a system will have its interrupt handler, keyboard it will have its interrupt handler for every device it will have its interrupt handler.

The main features of the ISR are

- Interrupts can occur at any time they are asynchronous. ISR's can call for asynchronous interrupts.
- Interrupt service mechanism can call the ISR's from multiple sources.
- ISR's can handle both maskable and non maskable interrupts. An instruction in a program can disable or enable an interrupt handler call.
- ISR on beginning of execution it will disable other devices interrupt services. After completion of the ISR execution it will re initialize the interrupt services.
- The nested interrupts are allowed in ISR for diversion to other ISR.

## ***Type of Interrupt Handlers:***

1. First Level Interrupt Handler (FLIH) is hard interrupt handler or fast interrupt handler. These interrupt handlers have more jitter while process execution and they are mainly maskable interrupts
2. Second Level Interrupt Handler (SLIH) is soft interrupt handler and slow interrupt handler. These interrupt handlers are having less jitter.

## ***Interrupt Latency:***

When an interrupt occur, the service of the interrupt by executing the ISR may not start immediately by context switching. The time interval between the occurrence of interrupt and start of execution of the ISR is called interrupt latency.

- **Tswitch** = Time taken for context switch
- **$\Sigma T_{exec}$**  = The sum of time interval for executing the ISR
- **Interrupt Latency** =  $T_{switch} + \Sigma T_{exec}$

**Source:** <http://www.electronicshub.org/types-of-interrupts-and-how-to-handle-interrupts/>