**Index-**

1. Types of Interrupts.
2. Steps during Interrupt occur
3. How to set Interrupt in STM32F103/F407

Interrupts can be classified into various categories based on different parameters:

1. **Hardware and Software Interrupts :**  
   When microprocessors receive interrupt signals through pins (hardware) of microprocessor, they are known as *Hardware Interrupts*.
2. **Vectored and Non-Vectored Interrupts :**  
   *Vectored Interrupts* are those which have fixed vector address (starting address of sub-routine) and after executing these, program control is transferred to that address.

Non-Vectored Interrupts are those in which vector address is not predefined. The interrupting device gives the address of sub-routine for these interrupts.

1. **Maskable and Non-Maskable Interrupts :**  
   ***Maskable Interrupts*** are those which can be disabled or ignored by the microprocessor. These interrupts are either edge-triggered or level-triggered, so they can be disabled. *INTR, RST 7.5, RST 6.5, RST 5.5*are maskable interrupts in 8085 microprocessor.

**Non-Maskable Interrupts** are those which cannot be disabled or ignored by microprocessor. *TRAP* is a non-maskable interrupt. It consists of both level as well as edge triggering and is used in critical power failure conditions.

Steps to Execute an Interrupt

When an interrupt gets active, the microcontroller goes through the following steps −

* The microcontroller closes the currently executing instruction and saves the address of the next instruction (PC) on the stack.
* It also saves the current status of all the interrupts internally (i.e., not on the stack).
* It jumps to the memory location of the interrupt vector table that holds the address of the interrupts service routine.
* The microcontroller gets the address of the ISR from the interrupt vector table and jumps to it. It starts to execute the interrupt service subroutine, which is RETI (return from interrupt).
* Upon executing the RETI instruction, the microcontroller returns to the location where it was interrupted. First, it gets the program counter (PC) address from the stack by popping the top bytes of the stack into the PC. Then, it start to execute from that address.

Edge Triggering vs. Level Triggering

Interrupt modules are of two types − level-triggered or edge-triggered.

|  |  |
| --- | --- |
| **Level Triggered** | **Edge Triggered** |
| A level-triggered interrupt module always generates an interrupt whenever the level of the interrupt source is asserted. | An edge-triggered interrupt module generates an interrupt only when it detects an asserting edge of the interrupt source. The edge gets detected when the interrupt source level actually changes. It can also be detected by periodic sampling and detecting an asserted level when the previous sample was de-asserted. |
| If the interrupt source is still asserted when the firmware interrupt handler handles the interrupt, the interrupt module will regenerate the interrupt, causing the interrupt handler to be invoked again. | Edge-triggered interrupt modules can be acted immediately, no matter how the interrupt source behaves. |
| Level-triggered interrupts are cumbersome for firmware. | Edge-triggered interrupts keep the firmware's code complexity low, reduce the number of conditions for firmware, and provide more flexibility when interrupts are handled. |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |