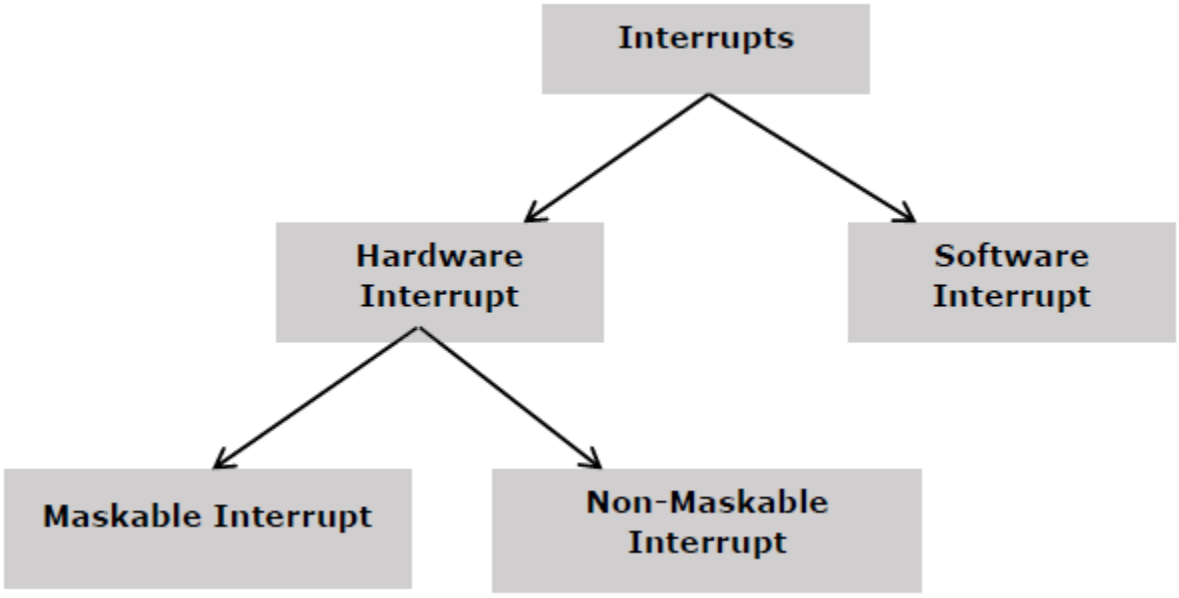
Interrupts in 8086

Interrupt is the method of creating a temporary halt during program execution and allows peripheral devices to access the microprocessor. The microprocessor responds to that interrupt with an **ISR** (Interrupt Service Routine), which is a short program to instruct the microprocessor on how to handle the interrupt.

The following image shows the types of interrupts we have in a 8086 microprocessor –



Hardware Interrupts

Hardware interrupt is caused by any peripheral device by sending a signal through a specified pin to the microprocessor.

The 8086 has two hardware interrupt pins, i.e. NMI and INTR. NMI is a non-maskable interrupt and INTR is a maskable interrupt having lower priority. One more interrupt pin associated is INTA called interrupt acknowledge.

NMI

It is a single non-maskable interrupt pin (NMI) having higher priority than the maskable interrupt request pin (INTR)and it is of type 2 interrupt.

When this interrupt is activated, these actions take place –

- Completes the current instruction that is in progress.

- Pushes the Flag register values on to the stack.
- Pushes the CS (code segment) value and IP (instruction pointer) value of the return address on to the stack.
- IP is loaded from the contents of the word location 00008H.
- CS is loaded from the contents of the next word location 0000AH.
- Interrupt flag and trap flag are reset to 0.

INTR

The INTR is a maskable interrupt because the microprocessor will be interrupted only if interrupts are enabled using set interrupt flag instruction. It should not be enabled using clear interrupt Flag instruction.

The INTR interrupt is activated by an I/O port. If the interrupt is enabled and NMI is disabled, then the microprocessor first completes the current execution and sends '0' on INTA pin twice. The first '0' means INTA informs the external device to get ready and during the second '0' the microprocessor receives the 8 bit, say X, from the programmable interrupt controller.

These actions are taken by the microprocessor –

- First completes the current instruction.
- Activates INTA output and receives the interrupt type, say X.
- Flag register value, CS value of the return address and IP value of the return address are pushed on to the stack.
- IP value is loaded from the contents of word location X × 4
- CS is loaded from the contents of the next word location.
- Interrupt flag and trap flag is reset to 0

Software Interrupts

Some instructions are inserted at the desired position into the program to create interrupts. These interrupt instructions can be used to test the working of various interrupt handlers. It includes –

INT- Interrupt instruction with type number

It is 2-byte instruction. First byte provides the op-code and the second byte provides the interrupt type number. There are 256 interrupt types under this group.

Its execution includes the following steps –

- Flag register value is pushed on to the stack.
- CS value of the return address and IP value of the return address are pushed on to the stack.
- IP is loaded from the contents of the word location 'type number' × 4
- CS is loaded from the contents of the next word location.

- Interrupt Flag and Trap Flag are reset to 0

The starting address for:

type0 interrupt is 000000H,

type1 interrupt is 00004H,

similarly for type2 is 00008H and .....so on.

The first five pointers are dedicated interrupt pointers. i.e. –

Type-0 Interrupt :

•Type-0 interrupt is Divide - by – Zero Interrupt

•When the result (Quotient) of division operation is too large to be stored in the destination register then 8086 perform Type-0 interrupt

•NMI is edge-triggered on a low to high transition

•It is required to remain in high state for more than two clock cycles

4. Type-1 Interrupt :

•Type-1 interrupt is a single step interrupt

•Single step interrupt is used for debugging Program (ALP)

•In single step control, the processor executes one instruction and then stops

•Thereafter the contents of registers and memory can be examined to see the result If the results are correct the user can give a command to execute the next instruction

•In 8086 Trap flag(TF) is set to 1, to implement single step control

•The ISR for Type-1 interrupt is written to save the contents of all registers on the stack so that results can be examined

5. Type-2 Interrupt:

• Type-2 is Non-maskable emergency conditions interrupt used for some emergency conditions.

•ISR for Type-2 interrupt is written to save program which processor is executing

• For example, it can be used to save program and data when power fails

•The program along with the data is restored when power returns

6. Type-3 Interrupt :

•Type-3 interrupt is Break-point Interrupt A break point is inserted in a program (ALP) for debugging

•The program is executed up to the break point and an interrupt occurs at this point of the program

•The ISR saves the contents of the registers on the stack

•The register contents may be displayed on the CRT terminal (or) the control is returned to the user

7. Type-4 Interrupt :

•Type-4 interrupt is an Overflow Interrupt

•When signed result of an arithmetic operation on two signed numbers is too large to be stored in the destination register (or) memory location, an overflow error occurs and the 8086 sets overflow flag(OF) to 1

•To implement Type-4 interrupt INTO instruction is inserted in the program immediately after the arithmetic instruction

•If OF = 0; INTO instruction performance no operation OF = 1; INTO instruction performance Type-4 interrupt