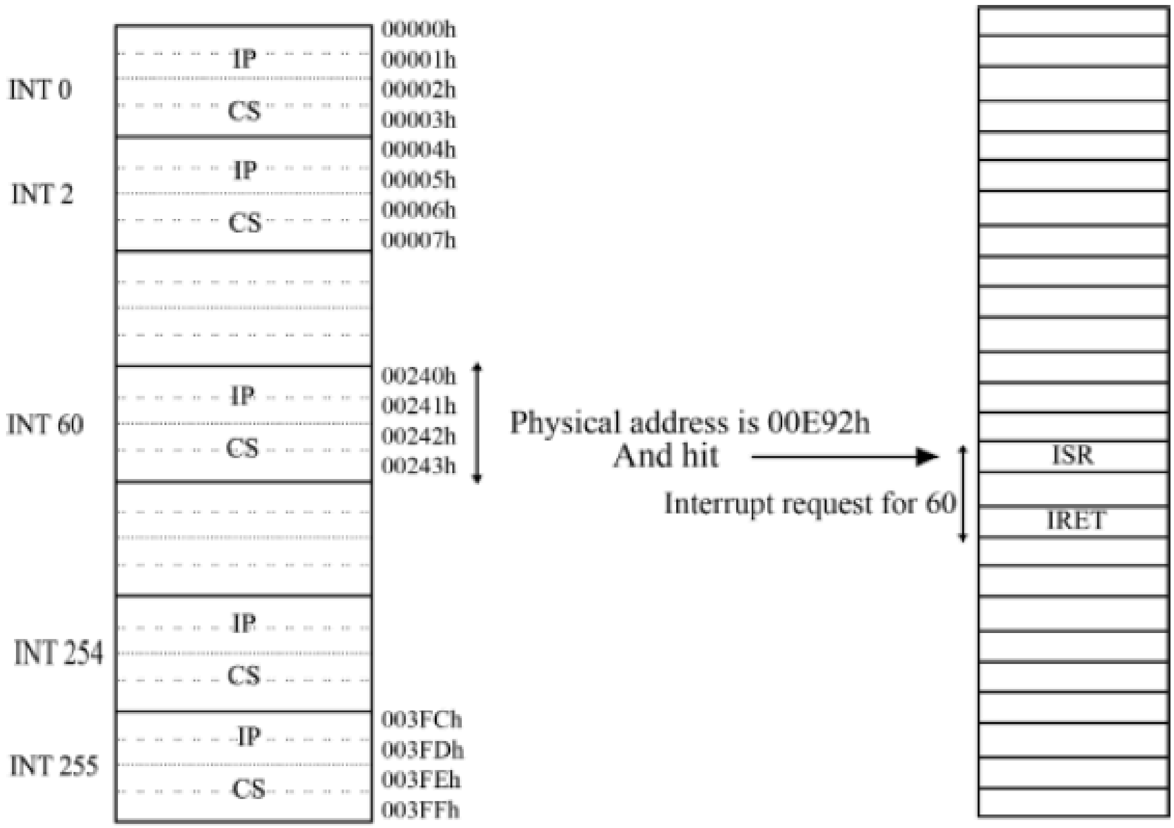
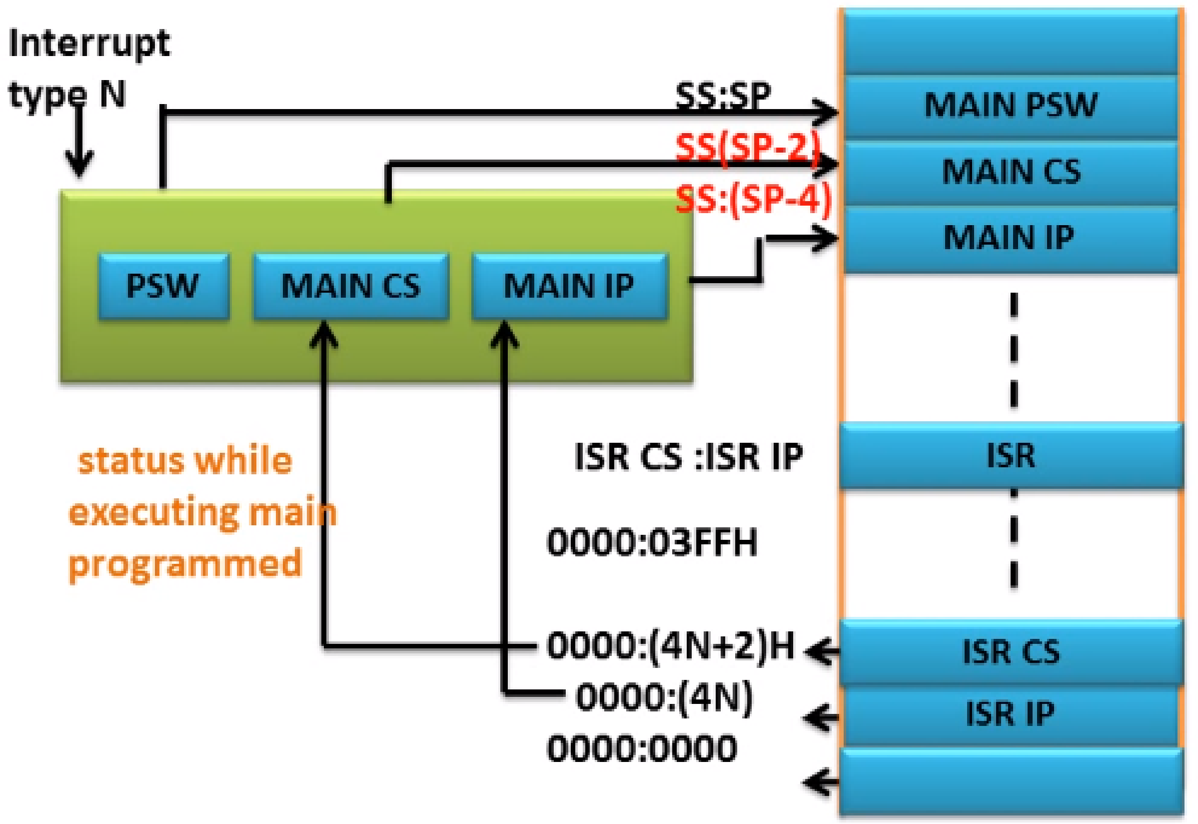
Software interrupts

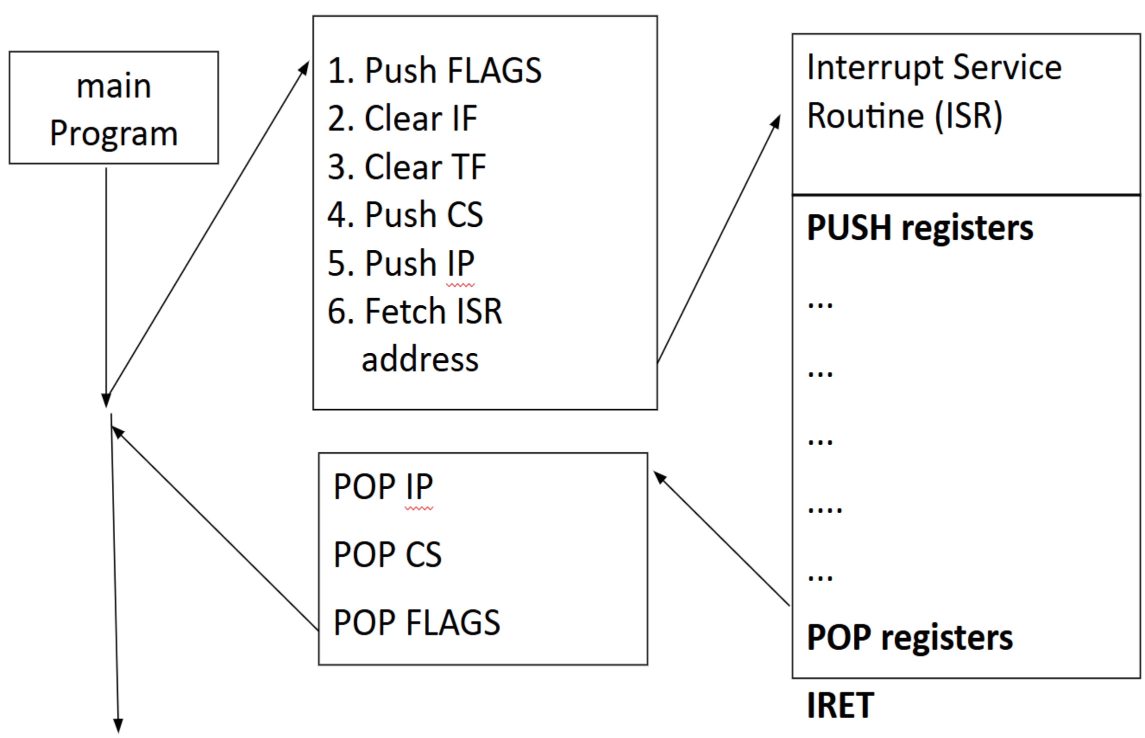
- There are 256 types of software interrupts.
- The software interrupts are program instructions. These instructions are inserted at desired locations in a program. While running a program, if software interrupt instruction is encountered then the processor initiates an interrupt.
- Types : 0-255 / 00-FF
- The software interrupt instruction is INT n
  Here, n = 0-255
- n= Type [by INTEL]
- Execution of interrupt instruction INT - user-defined interrupt/ software interrupt
- Some error condition produced by execution - Pre-defined interrupt/ Software interrupt
- There is a **interrupt handler - ISR**
- All interrupts (vectored or otherwise) are mapped onto a memory area called the **Interrupt Vector Table (IVT)**
- An **interrupt vector is a pointer** to **where the ISR is stored in memory**
- The **starting address of an ISR** is often called - **interrupt vector/ interrupt pointer.**
- **Table** is referred to as **interrupt-vector table/ interrupt-pointer table**



INT 0
IP
CS
00000h
00001h
00002h
00003h
INT 2
IP
CS
00004h
00005h
00006h
00007h
INT 60
IP
CS
00240h
00241h
00242h
00243h
INT 254
IP
CS
INT 255
IP
CS
003FCh
003FDh
003FEh
003FFh
Physical address is 00E92h
And hit
Interrupt request for 60
ISR
IRET



Interrupt type N
SS:SP
SS(SP-2)
SS:(SP-4)
MAIN PSW
MAIN CS
MAIN IP
PSW
MAIN CS
MAIN IP
status while executing main programmed
ISR CS :ISR IP
ISR
0000:03FFH
0000:(4N+2)H
ISR CS
0000:(4N)
ISR IP
0000:0000



main Program
1. Push FLAGS
2. Clear IF
3. Clear TF
4. Push CS
5. Push IP
6. Fetch ISR address
Interrupt Service Routine (ISR)
PUSH registers
...
...
...
....
...
POP registers
IRET
POP IP
POP CS
POP FLAGS

<u>Explaining Stack through example:</u>

SS = 2000H
SP = FFF0H
CS = 0F00H
IP = 1234H
FLAGS = 020AH (just an example)

Now,
(SP - 2) = FFEF;
(FEEE - 2) = FFEC;
(FFEC -2) = FFEA;

Stack push order:

Address	Value	Description
→ 2000:FFF0		
2000:FFEF	02H	FLAGS High Byte
2000:FFEE	0AH	FLAGS Low Byte
2000:FFED	0FH	CS High Byte
2000:FFEC	00H	CS Low Byte
2000:FFEB	12H	IP High Byte
2000:FFEA	34H	IP Low Byte

Stack pop order:
When the interrupt service routine ends, the values are popped off the stack in reverse order:

1. **IP ← top of stack (from FFEAH)**
2. **CS ← next (from FFECH)**
3. **FLAGS ← last (from FFEEH)**

Since the stack is LIFO, the last thing pushed (IP) is the first thing popped.

This ensures the CPU returns exactly to the same instruction and state it was in before the interrupt occurred.

Example

1. Assume the table is a portion of the current memory address space of an Intel 8086. The microprocessor currently has the following values in its registers: SS = 2000h, SP = 1124h, CS = 3000h, IP = 1450h. Now a signal arrives at the INTR pin of this 8086.

Addr	00117h	00116h	00115h	00114h	00276h	00277h	00248h	00279h
Data	45h	86h	22h	14h	12h	34h	56h	78h

a. If the 8086 decides to service the interrupt, then do the bits of the flag register change?
   ⇒ Only TF and IF can change (cleared). The rest of the values will remain the same and will be popped after the interrupt service routine finishes execution.

b. If the signal is of Interrupt type 69, then deduce the values of CS and IP as the 8086 starts the service routine.
   ⇒ Type 69 interrupt: Interrupt vector starts at = 69x4 = 276, 277 = 00114h, 00115h
   Thus, new value of IP after interrupt = (Low byte at 0114h, High byte at 0115h) = 2214h

   New value of CS after interrupt = (69*4)+2 = 278,279 (low byte at 00116h, high byte at 00117h) = 4586h