

# Interrupts in 8086

Most microprocessor allows normal program execution to be terminated by some external signal or by a special instruction in the program. In response to this the processor stops the execution of current program and calls the procedure which services the interrupt.

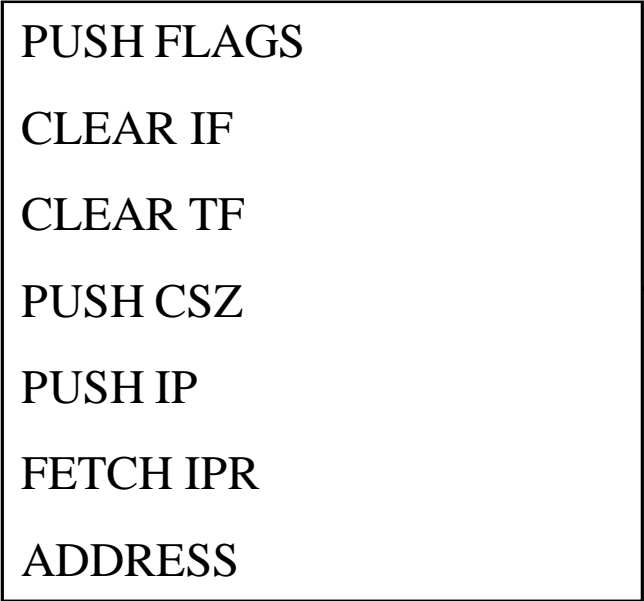
An interrupt can come from any three sources:

1. an external signal applied to the nonmaskable interrupt (NMI) input pin or to the interrupt (INTR) input pin. It is also known as hardware interrupt.
2. Execution of the interrupt instruction(INT). Also known as Software interrupt.
3. Some error condition produced by the execution of an instruction. Ex. Divide by zero.

At the end of each instruction cycle, the 8086 checks to see if any interrupts have been requested, the 8086 responds to the interrupt by stepping through the following series of major actions.

1. It decrements the stack pointer by 2 and pushes the flag register on the stack.
2. It disables the 8086 INTR interrupt input by clearing the interrupt flag(IF) in the flag register.
3. It resets the trap flag (TF) in the flag register.
4. It decrements the stack pointer by 2 and pushes the current code segment register contents on the stack..
5. It decrements the stack pointer by 2 and pushes the current instruction pointer contents on the stack.
6. It does an indirect far jump to the start of the procedure the user has written to response to the interrupt.

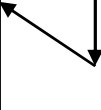
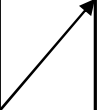
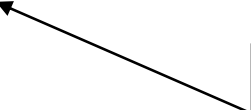
MAINLINE  
PROGRAM

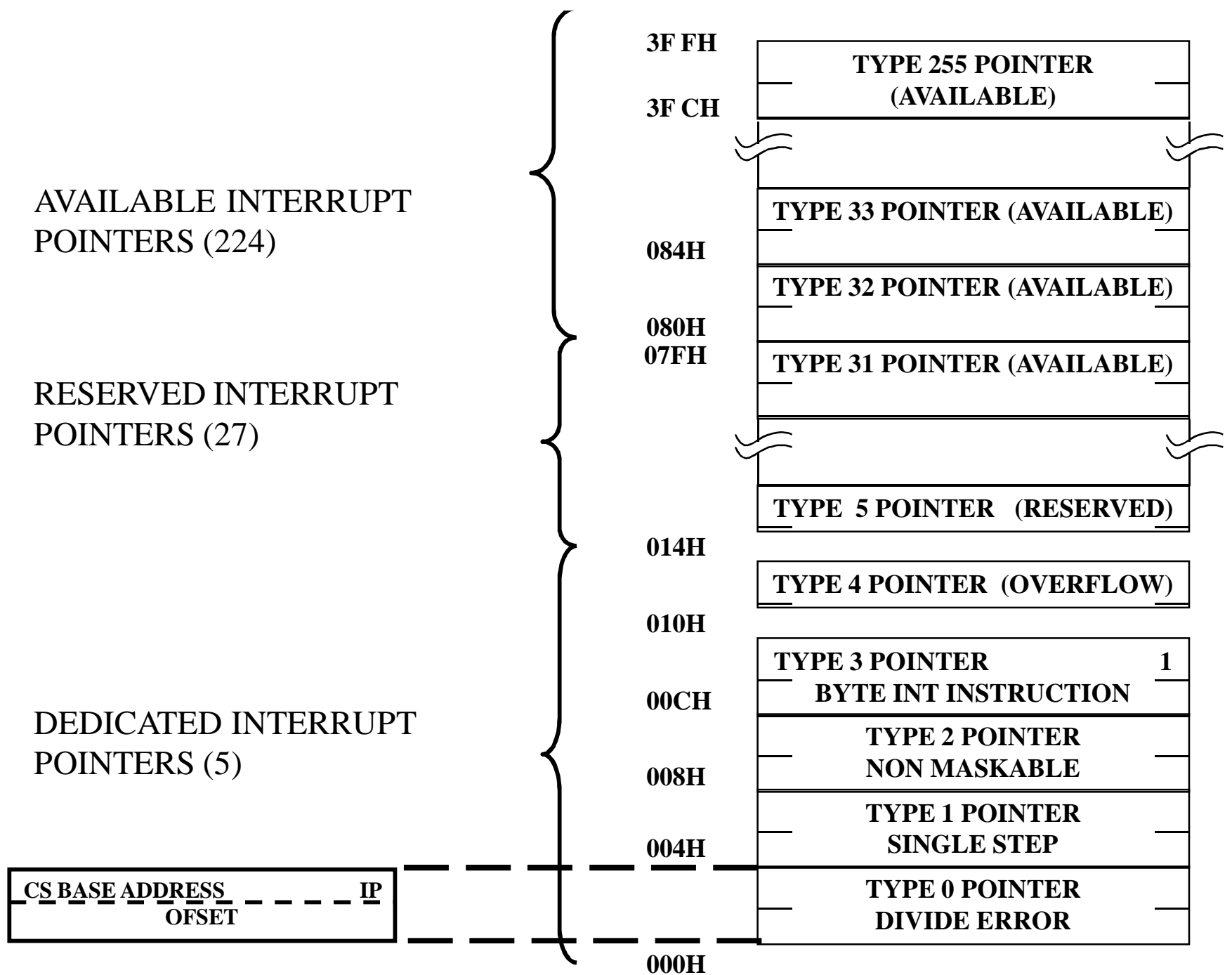


INTERRUPT  
SERVICE  
PROCEDURE



POP  
REGISTERS  
IRET





**VECTOR INTERRUPT TABLE O 8086**