Interrupt

Interrupt:

An interrupt is a condition that causes the microprocessor to temporarily work on a different task, and then later return to its previous task.

Interrupts can be internal or external.

- **Hardware Interrupt**
  - **Non-maskable Interrupt (NMI):** It is a single pin non-maskable hardware interrupt which cannot be disabled. It is the highest priority interrupt in 8086 microprocessor. After its execution, this interrupt generates a TYPE 2 interrupt. IP is loaded from word location 00008 H and CS is loaded from the word location 0000A H.
  - Maskable Interrupt (INTR): It provides a single interrupt request and is activated by I/O port. This interrupt can be masked or delayed. It is a level triggered interrupt. It can receive any interrupt type, so the value of IP and CS will change on the interrupt type received.
- **Software Interrupt (INT):** These are instructions that are inserted within the program to generate interrupts. There are 256 software interrupts in 8086 microprocessor. The instructions are of the format INT type where type ranges from 00 to FF. The starting address ranges from 00000 H to 003FF H. These are 2 byte instructions.
  Some important software interrupts are:
  (A) TYPE 0: corresponds to division by zero(0).
  (B) TYPE 1: is used for single step execution for debugging of program.
  (C) TYPE 2: represents NMI and is used in power failure conditions.
  (D) TYPE 3: represents a break-point interrupt.
  (E) TYPE 4: is the overflow interrupt.

  Type 0: The 8086 will automatically do a type 0 interrupt if the result of a DIV operation or an IDIV operation is too large to fit in the destination register. For a type 0 interrupt, the 8086 pushes the flag register on the stack, resets IF and TF and pushes the return addresses on the stack.

(line by line execution) **Type 1:** The use of single step feature found in some monitor programs and debugger programs. When a system is told to single step, it will execute one instruction and stop. If they are correct we can tell a system to single step, it will execute one instruction and stop. We can then examine the contents of registers and memory locations. In other words, when in single step mode a system will stop after it executes each instruction and wait for further direction from you. The 8086 trap flag and type 1 interrupt response make it quite easy to implement a single step feature direction.

  Type 2: The 8086 will automatically do a type 2 interrupt response when it receives a low to high transition on its NMI pin. When it does a type 2 interrupt, the 8086 will push the flags on the stack, reset TF and IF, and push the CS value and the IP value for the next instruction on the stack. It will then get the CS value for the start of the type 2

interrupt service procedure from address 0000AH and the IP value for the start of the procedure from address 00008H.

Type 3: The main use of the type 3 interrupt is to implement a breakpoint function in a system. When we insert a breakpoint, the system executes the instructions up to the breakpoint and then goes to the breakpoint procedure. Unlike the single step which stops execution after each instruction, the breakpoint feature executes all the instructions up to the inserted breakpoint and then stops execution.

Type 4: The 8086 overflow flag will be set if the signed result of an arithmetic operation on two signed numbers is too large to be represented in the destination register or memory location. For example, if you add the 8 bit signed number 01101100 and the 8 bit signed number 010111101, the result will be 10111101. This would be the correct result if we were adding unsigned binary numbers, but it is not the correct signed result.

Response to an interrupt:

i. PUSH FLAGS

ii. Disables the 8086 INTR interrupt input by clearing the interrupt (IF) in the flag register

iii. It resets the trap flag (TF) in the flag register

iv. PUSH CS

v. PUSH IP

vi. It does an indirect far jump to the start of the procedure written to respond to the interrupt