8086 Interrupts

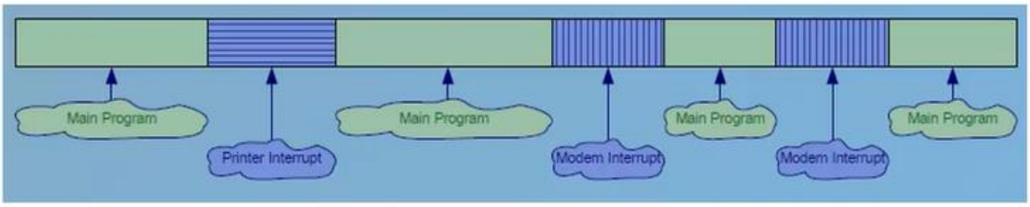
Dr. Ashish Girdhar
CSED

Interrupts

- An interrupt is used to cause a temporary halt in the execution of program.
- The microprocessors allow normal program execution to be interrupted in order to carry out a specific task/work.
- Microprocessor responds to the interrupt with ISR, which is short program that instructs the microprocessor on how to handle the interrupt.

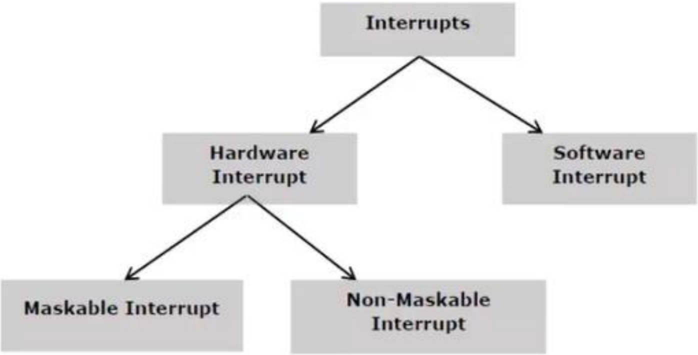
Purpose of Interrupts

- Interrupts are useful when interfacing I/O devices with low data-transfer rates, like a keyboard or a mouse.
- The peripheral interrupts the normal application execution, requesting to send or receive data.
- The processor jumps to a special program called *Interrupt Service Routine* to service the peripheral
- After the processor services the peripheral, the execution of the interrupted program continues.



Classification of Interrupts

The interrupts can be classified as:

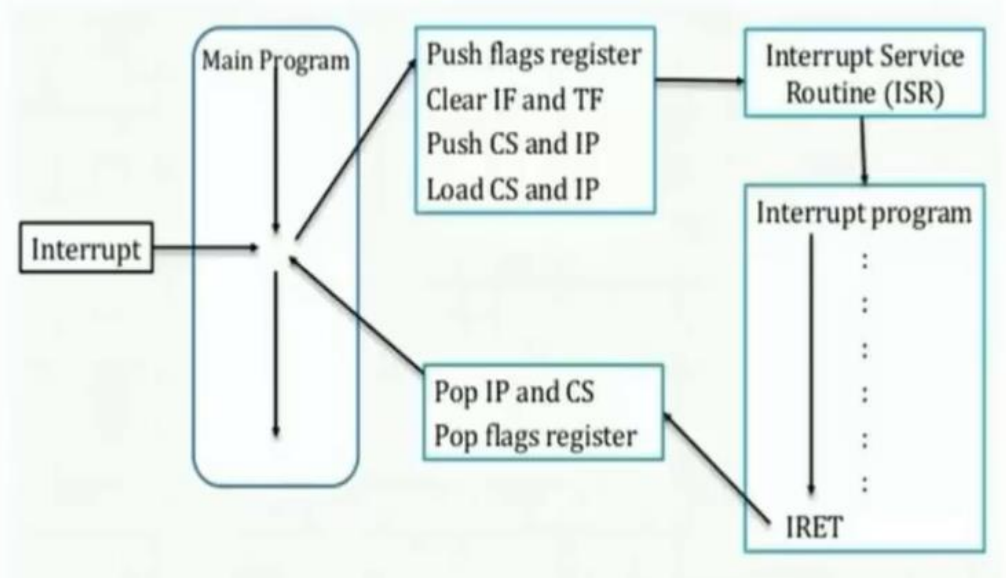


Sources of Interrupt

In 8086, interrupt can come from any one of three sources.

- Execution of the interrupt instruction (SOFTWARE INTERRUPT) e.g. *Int n* instruction
- An external signal applied to the non-maskable interrupt (NMI) input pin or to the interrupt input pin (HARDWARE INTERRUPT).
- Some error condition produced in the 8086 by the execution of an instruction. E.g. *divide by zero*

What happen when Interrupt is occurred in 8086?



Interrupt Response

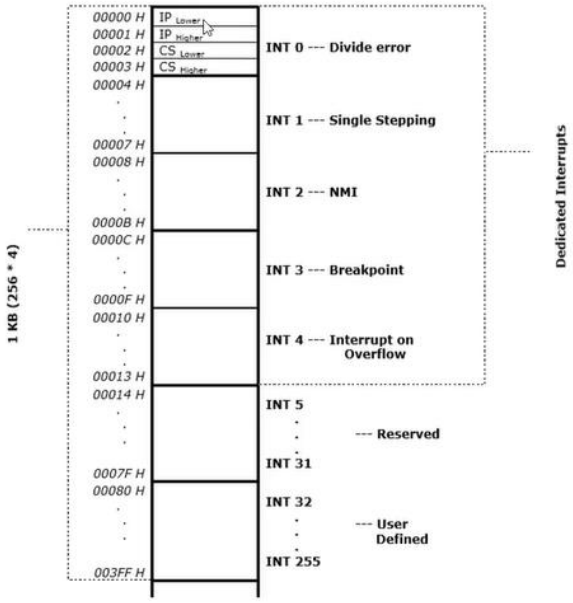
- It decrements stack pointer by 2 and pushes the flag register on the stack.
- It disables the INTR interrupt input by clearing the interrupt flag in the flag.
- It resets the trap flag in the flag register.
- It decrements stack pointer by 2 and pushes the current code segment register contents on the stack.
- It decrements stack pointer by 2 and pushes the current instruction pointer contents on the stack.
- It does an indirect far jump at the start of the procedure by loading the CS and IP values for the start of the interrupt service routine (ISR).

Interrupt Vectors

- For every interrupt, there must be a ISR associated with it.
- When an interrupt occurs, CPU runs the Interrupt handler but where is the handler?
  - In the IVT(Interrupt Vector Table)
- IVT consist of 256 entries each containing 4 bytes.
- Each entry contains the *offset and the segment address* of the interrupt vector each 2 bytes long.
- Table starts from memory address 00000h

Interrupt Vector Table

- The first 1Kbyte of memory of 8086 (00000 to 003FF) is set aside as a table for storing the starting addresses of Interrupt Service Procedures(ISP).
- Since 4-bytes are required for storing starting addresses of ISPs, the table can hold 256 Interrupt procedures.
- The starting address of an ISP is often called the Interrupt Vector or Interrupt Pointer. Therefore the table is referred as Interrupt Vector Table.
- In this table, IP value is put in as low word of the vector & CS is put in high vector



Interrupt Vector Table – Example

□To service an interrupt, the 8086 calculates the two addresses in the pointer table where IP and CS and stored for a type:

□For INTnn (where nn Type code)

□Table address for IP = 4 * nn

□Table address for CS= 4 * nn(or IP address) + 2

□Example: INT2: IP address= 4 * 2=00008

□CS address = 00008 + 2 = 0000AH

□The values of IP and Cs are loaded from locations 00008 H & 0000A H in the pointer table

Interrupt Vector Table – Example

□To service an interrupt, the 8086 calculates the two addresses in the pointer table where IP and CS and stored for a type:

□For INTnn (where nn Type code)

□Table address for IP = 4 * nn

□Table address for CS= 4 * nn(or IP address) + 2

□Example: INT2: IP address= 4 * 2=00008

□CS address = 00008 + 2 = 0000AH

□The values of IP and Cs are loaded from locations 00008 H & 0000A H in the pointer table

INT 12h (or vector 12)

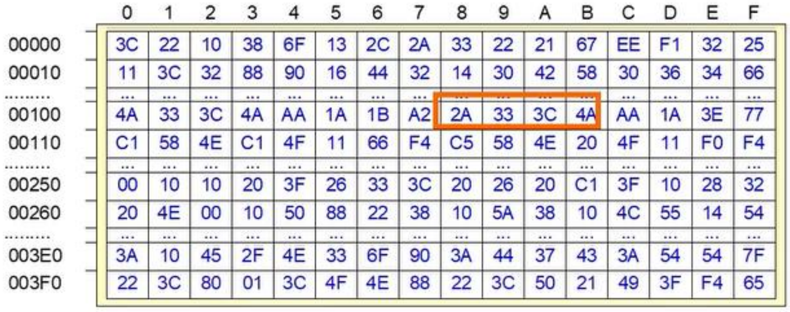
The physical address 30h (4 x 12 = 48 = 30h) contains

0030h and 0031h contain IP of the ISR

0032h and 0033h contain CS of the ISR

Interrupt Vector Table – Example

- Using the Interrupt Vector Table shown below, determine the address of the ISR of a device with interrupt vector 42H.
- Answer: Address in table = 4 X 42H = 108H
- (Multiply by 4 since each entry is 4 bytes)
- Offset Low = [108] = 2A, Offset High = [109] = 33
- Segment Low = [10A] = 3C, Segment High = [10B] = 4A
- Address = 4A3C:332A = 4A3C0 + 332A = 4D6EAH

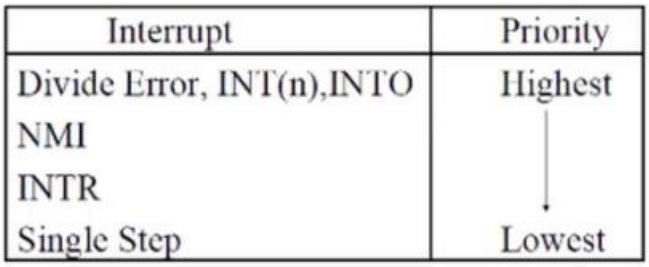


	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000	3C	22	10	38	6F	13	2C	2A	33	22	21	67	EE	F1	32	25
00010	11	3C	32	88	90	16	44	32	14	30	42	58	30	36	34	66
....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
00100	4A	33	3C	4A	AA	1A	1B	A2	2A	33	3C	4A	AA	1A	3E	77
00110	C1	58	4E	C1	4F	11	66	F4	C5	58	4E	20	4F	11	F0	F4
....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
00250	00	10	10	20	3F	26	33	3C	20	26	20	C1	3F	10	28	32
00260	20	4E	00	10	50	88	22	38	10	5A	38	10	4C	55	14	54
....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
003E0	3A	10	45	2F	4E	33	6F	90	3A	44	37	43	3A	54	54	7F
003F0	22	3C	80	01	3C	4F	4E	88	22	3C	50	21	49	3F	F4	65

Other Types of Interrupt

- TYPE 5 to TYPE 31(27 interrupts) :
  - kept reserved for other advanced microprocessors of Intel Corp.
  - However, can use by computer designers for some special interrupts in 8086 based system.
- TYPE 32 to TYPE 255(244 interrupts) : available to users for h/w and s/w interrupts.

Interrupt Priority



Interrupt	Priority
Divide Error, INT(n),INTO	Highest
NMI	↓
INTR	
Single Step	Lowest

H/w Interrupts

- NMI : to handle emergency conditions.

-- eg. Used to save prog. and data in case of power failure.

-- in case of a.c. power failure, D.C. supply can maintained for short period,(50 ms using suitable capacitors in filter circuits).

H/w Interrupts

- INTR : maskable. Enabled ,disabled using IF.
- With receive of INTR from external device, 8086 acknowledge through INTA signal.
- It executes two consecutive interrupt acknowledgement bus cycles.

-- first INTA signal to inform the external device to get ready.

Ext. device sends interrupt type through the low order data lines.

- INTR : Intr. Type is * by 4 to get staring add. of IVT.

- LOCK(low active) goes low from T2 of the first ack. cycle until T2 of the 2nd ack. cycle.
- HOLD req. is not granted till the end of 2nd bus cycle.

Hardware Interrupts – Interrupt pins

- x86 Interrupt Pins
  - INTR: Interrupt Request. Activated by a peripheral device to interrupt the processor.
    - Level triggered. Activated with a logic 1.
  - INTA': Interrupt Acknowledge. Activated by the processor to inform the interrupting device the interrupt request (INTR) is accepted.
    - Level triggered. Activated with a logic 0.
  - NMI: Non-Maskable Interrupt. Used for major system faults such as parity errors and power failures.
    - Edge triggered. Activated with a positive edge (0 to 1) transition.
    - Must remain at logic 1, until it is accepted by the processor.
    - Before the 0 to 1 transition, NMI must be at logic 0 for at least 2 clock cycles.
    - No need for interrupt acknowledgement.

8085 vs 8086 Interrupts

Microprocessor	8085	8086
Hardware Interrupts	5	2
Non-maskable Interrupt	TRAP	NMI
Software Interrupts	8	256
Vectored Address Calculation	Is done by multiplying n*8	Is done by multiplying n*4

Thank You