NEUB CSE 321 Lecture 6: Interrupts in 8086

Types of data transfer

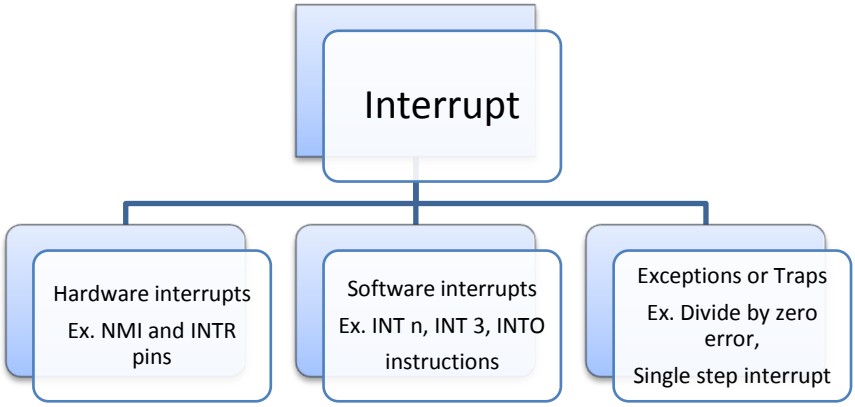
- Simple I/O – used when timings of I/O device is known(Ex. Collect newspaper leisurely any time after 8 a.m.)
- Status check I/O – Data transfer done anytime after I/O device says it is ready (Ex. Check newspaper box and collect newspaper if it is in the box)
- Interrupt driven I/O – data transfer done immediately after I/O device interrupts (Ex. Collect newspaper when the door bell rings indicating delivery of newspaper)

An interrupt is a procedure that interrupts whatever program is currently executing.

What Happens during interrupt

I. Completes the current instruction
II. Jumps to a subroutine called *Interrupt Service* procedure
III. Executes the instruction
IV. Returns back to the interrupted program

Types of interrupt



Interrupt Vector table (IVT)

RAM locations 0 to 003FFH are used to store IVT. It contains 256 Interrupt Vectors (IV) each of 4 bytes, thus totaling 1024 bytes (1KB) of memory. Figure 1 shows the interrupt vector table that can hold up to 256 interrupt vectors. Each vector is saved in CS:IP format in four consecutive bytes in memory and the vector is identified by the type number (Type 0 to type 255). For example type 0 interrupt is stored in locations 0000H:0000H - 0000H:0003H.

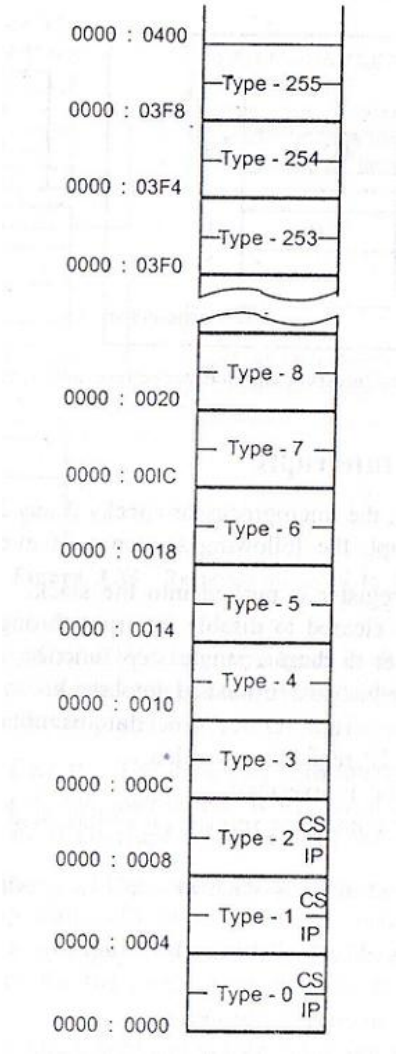


Figure 1 Interrupt vector table

8086 Response to interrupt

At the end of each instruction cycle checks if any interrupt is active. In case of active interrupt the sequence happens

a. The content of the flag register is pushed to stack
b. The interrupt flag (IF) is cleared to disable INTR input interrupt
c. Trap flag (TF) is reset to disable single step function
d. Content of CS and IP is pushed to stack
e. Interrupt vector is obtained from interrupt vector table and CS and IP registers are filled
f. Interrupt service procedure is executed
g. The last instruction in the ISP will be IRET which in turn

PREPARED BY
SHAHADAT HUSSAIN PARVEZ

i. POP CS and IP
ii. POP Flags

h. Execution returns to interruoted program

Example 3.8

Show schematically the locations of interrupt service procedure and interrupt vector for the type-8 interrupt.

Solution

Figure 3.37 shows the locations of interrupt service procedure and interrupt vector for type-8 interrupt.

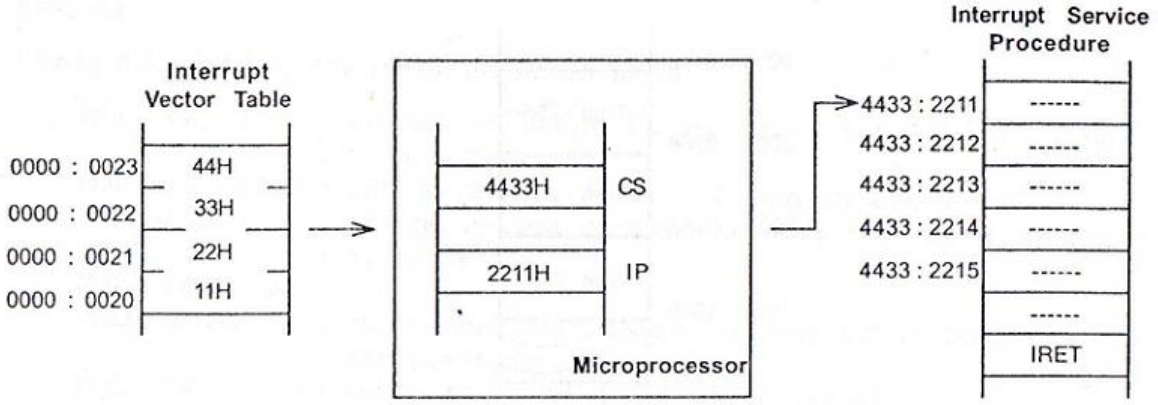


Figure 3.37 Locations of interrupt service procedure and interrupt vector.

8086 Interrupt types

a. Predefined interrupt
b. Software interrupt
c. Hardware interrupt

First 32 interrupts are the predefined interrupts, which Intel uses for some predefined function. Among these 32 interrupts, first 5 (Type 0 – Type 4) are used in 8086 and rest are reserved for advanced microprocessors. Every interrupt type in 8086 has an 8-bit Interrupt type number (ITN) or Interrupt vector number.

The details of each of the type of interrupts are as follows:

Type 0 Divide Error—Occurs whenever the result of a division overflows or whenever an attempt is made to divide by zero.

Type 0 interrupt is a nonmaskable interrupt.

Type 1 Single-Step or Trap—Occurs after the execution of each instruction if the trap (TF) flag bit is set. Upon accepting this interrupt, the TF-bit is cleared so that the interrupt service procedure executes at full speed. More detail is provided about this interrupt later in this section of the chapter.

Type 2 Non-maskable Hardware Interrupt—A result of placing a logic 1 on the NMI input pin to the microprocessor. This input is non-maskable, which means that it cannot be disabled.

Type 2 interrupt happens when a low to high transition occurs in NMI (Non Maskable Interrupt) pin on 8086

Type 3 One-Byte Interrupt—A special 1-byte instruction (INT 3) that uses this vector to access its interrupt service procedure. The INT 3 instruction is often used to store a breakpoint in a program for debugging.

PREPARED BY
SHAHADAT HUSSAIN PARVEZ

Type 4 Overflow—A special vector used with the INTO instruction. The INTO instruction interrupts the program if an overflow condition exists, as reflected by the overflow flag (OF).

Type 5 BOUND—An instruction that compares a register with boundaries stored in the memory. If the contents of the register is greater than or equal to the first word in memory and less than or equal to the second word, no interrupt occurs because the contents of the register is within bounds. If the contents of the register is out of bounds, a type 5 interrupt ensues.

Type 6 Invalid Opcode—Occurs whenever an undefined opcode is encountered in a program.

Type 7 Coprocessor Not Available—Occurs when a coprocessor is not found in the system, as dictated by the machine status word (MSW) coprocessor control bits. If an ESC or WAIT instruction executes and the coprocessor is not found, a type 7 exception or interrupt occurs.

Type 8 Double Fault—Activated whenever two separate interrupts occur during the same instruction.

Type 9 Coprocessor Segment Overrun—Occurs if the ESC instruction (coprocessor opcode) memory operand extends beyond offset address FFFFH.

Type 10 Invalid Task State Segment—Occurs if the TSS is invalid because the segment limit field is not 002BH or higher. In most cases this is caused because the TSS is not initialized.

Type 11 Segment Not Present—Occurs when the P-bit (P = 0) in a descriptor indicates that the segment is not present or not valid.

Type 12 Stack Segment Overrun—Occurs if the stack segment is not present (P = 0) or if the limit of the stack segment is exceeded.

Type 13 General Protection—Occurs for most protection violation in the 80286–Pentium Pro protected mode system. (These errors occur in Windows as *general protection faults*). A list of these protection violations follows:

a. Descriptor table limit exceeded
b. Privilege rules violated
c. Invalid descriptor segment type loaded
d. Write to code segment that is protected
e. Read from execute-only code segment
f. Write to read-only data segment
g. Segment limit exceeded
h. CPL = IOPL when executing CTS, HLT, LGDT, LIDT, LLDT, LMSW, or LTR
i. CPL > IOPL when executing CLI, IN, INS, LOCK, OUT, OUTS, and STI

Type 14 Page Fault—Occurs for any page fault memory or code access in the 80386, 80486, and Pentium/Pentium Pro microprocessors.

Type 16 Coprocessor Error—Takes effect whenever a coprocessor error ($\overline{\text{ERROR}}$ = 0) occurs for the ESCape or WAIT instructions for the 80386, 80486, and Pentium/Pentium Pro microprocessors only.

Type 17 Alignment Check—Indicates that word, doubleword data are addressed at an odd memory location, or incorrect location in the case of a doubleword. This interrupt is active in the 80486 and Pentium/Pentium Pro microprocessors.

Type 18 Machine Check—Activates a system memory management mode interrupt in the Pentium and Pentium Pro microprocessors.

PREPARED BY
SHAHADAT HUSSAIN PARVEZ

Table 1 summarizes the interrupts used in Intel X86 processor family.

Table 1Summary of the interrupt vector for X86 processor family

Vector No.	Mnemonic	Description	Source
0	#DE	Divide Error	DIV and IDIV instructions.
1	#DB	Debug	Any code or data reference.
2		NMI Interrupt	Non-maskable external interrupt.
3	#BP	Breakpoint	INT 3 instruction.
4	#OF	Overflow	INTO instruction.
5	#BR	BOUND Range Exceeded	BOUND instruction.
6	#UD	Invalid Opcode (UnDefined Opcode)	UD2 instruction or reserved opcode.[1]
7	#NM	Device Not Available (No Math Coprocessor)	Floating-point or WAIT/FWAIT instruction.
8	#DF	Double Fault	Any instruction that can generate an exception, an NMI, or an INTR.
9		CoProcessor Segment Overrun (reserved)	Floating-point instruction.[2]
10	#TS	Invalid TSS	Task switch or TSS access.
11	#NP	Segment Not Present	Loading segment registers or accessing system segments.
12	#SS	Stack Segment Fault	Stack operations and SS register loads.
13	#GP	General Protection	Any memory reference and other protection checks.
14	#PF	Page Fault	Any memory reference.
15		(Intel reserved. Do not use.)	
16	#MF	Floating-Point Error (Math Fault)	Floating-point or WAIT/FWAIT instruction.
17	#AC	Alignment Check	Any data reference in memory.[3]
18	#MC	Machine Check	Error codes (if any) and source are model dependent.[4]
19-31		(Intel reserved. Do not use.)	
32-255		Maskable Interrupts	External interrupt from INTR pin or INT *n* instruction.

1. The UD2 instruction was introduced in the Pentium® Pro processor.
2. Intel Architecture processors after the Intel386™ processor do not generate this exception.
3. This exception was introduced in the Intel486™ processor.
4. This exception was introduced in the Pentium processor and enhanced in the Pentium Pro processor.

What is divide by 0 error?

Ex. DIV BL

	Before	After	
AH	40H	00H	This is an example for divide by 0 error. It only
AL	60H	2030H	means quotient is too large for the register!
BL	02H		

PREPARED BY
SHAHADAT HUSSAIN PARVEZ

Action for divide by 0 error (Type 1)

For divide by 0 error the ITN is 0

1. Push Flags on the stack
2. Reset IE flag (to ensure no further interrupts)
3. Reset T flag (so that ISS is not executed in single step)
4. PUSH CS
5. PUSH IP
6. IP loaded from word location 0 x 4 = 00000H
7. CS loaded from next word location 00002H

Processor makes a branch to the subroutine at location 5678:1234H if the contents of IVT is as shown in the table above.

Single step interrupt (Type1)

- Single step interrupt happens when TF in the program status word (PSW) is set
- Works one instructions at a time and saves PSW in the stack
- When return restores PSW (with TF set)

Action for Single step interrupt (Type1)

For Single step interrupt the ITN is 1

1. Push Flags on the stack
2. Reset IE flag (to ensure no further interrupts)
3. Reset T flag (so that ISS is not executed in single step)
4. PUSH CS
5. PUSH IP
6. IP loaded from word location1 x 4 = 00004H
7. CS loaded from next word location 00006H

Processor makes a branch to the subroutine at location 5566:3344H as per the IVT

Action when NMI is activated (Type 2)

NMI is positive edge triggered input

1. Complete the instruction in progress
2. Push Flags on the stack
3. Reset IE flag (to ensure no further interrupts)
4. Reset T flag (so that interrupt service subroutine, ISS, is not executed in single step)
5. PUSH CS
6. PUSH IP
7. IP loaded from word location 2 x 4 = 8 (2 is ITN)
8. CS loaded from next word location (0000AH)

Action when INT 3 is executed (Type 3)

1. Push Flags on the stack
2. Reset IE flag (to ensure no further interrupts)
3. Reset T flag (so that ISS is not executed in single step)
4. PUSH CS
5. PUSH IP
6. IP loaded from word location 3 x 4 = 0000CH
7. CS loaded from next word location 0000EH

INT 3 is a 1-byte instruction with opcode of CCH

PREPARED BY
SHAHADAT HUSSAIN PARVEZ

Overflow interrupt

When OF=1 during operation on two signed numbers, INTO (Interrupt on Overflow) is executed.

Eg: ADD AX,BX
INTO

Action for INTO instruction (Type 4)

INTO is a 1-byte instruction. For interrupt on overflow the ITN is 4.

1. Do steps 2 to 7 only if Overflow flag is set
2. Push Flags on the stack
3. Reset IE flag and T flag
4. PUSH CS
5. PUSH IP
6. IP loaded from word location 4 x 4 = 00010H
7. CS loaded from next word location 00012H

INTO is equivalent to: JNO Next
INT 4
Next: ….

Listing 3.4: Programs to set/reset the TF flag—(i) to set and (ii) to reset.

```
(i) Program to set TF
    PUSHF                       ; push flags into stack
    MOV BP, SP                  ; copy SP into BP for use as index
    OR WORD PTR 0[BP], 0100H    ; set TF bit
    POPF                        ; restore flag register
(ii)Program to reset TF
    PUSHF                       ; push flags into stack
    MOV BP, SP                  ; copy SP into BP for use as index
    AND WORD PTR 0[BP], 0FEFFH  ; reset TF bit
    POPF                        ; restore flag register
```

Software interrupts

Software interrupt is a 2 byte INT 'n' instruction. The n in the instruction specifies the interrupt type and takes values from 0 to 255.

BIOS (Basic Input and Output Services) interrupts in a PC are popular applications of software interrupt. BIOS is stored in ROM and directly controls hardware components of PC.

Some of other software interrupts are as below

Component Name	Interrupt
Video Display	INT 10H
Keyboard	INT 16H
Printer	INT 17H
Mouse	INT 33H

Execution of INT n (n=0 to FF)

1. Push Flags on the stack
2. Reset IE flag (to ensure no further interrupts)
3. Reset T flag (so that ISS is not executed in single step)
4. PUSH CS
5. PUSH IP
6. IP loaded from word location n x 4 = say, W
7. CS loaded from next word location W+2

In INT n, which is a 2-byte instruction, n is the ITN. INT n has the opcode CDH

PREPARED BY
SHAHADAT HUSSAIN PARVEZ

Hardware Interrupts

Hardware interrupts are of two types

1. NMI (Non Maskable Interrupt)
2. INTR (Interrupt Request)

NMI (Non Maskable Interrupt)

Whenever NMI pin is activated, a type 2 interrupt occurs. A common application of NMI is to save critical information

a. Power failure
b. Capacitor gives some backup
c. With a type 2 interrupt Information from RAM is stored in Memory

Example 3.9

Give an application for NMI interrupt.

Solution

NMI interrupt is used to get the immediate attention of microprocessor to perform emergency operations. Interrupt through NMI input is demonstrated by connecting a smoke detector to the microprocessor.

Smoke detectors are used in industries to monitor industrial safety. When the smoke detector detects smoke, it generates an interrupt pulse and sends to NMI input. Since the NMI interrupt is non-maskable and has top most priority, the microprocessor immediately attends to the request and performs type 2 interrupt response. The interrupt service procedure will have instructions to send control word to switch OFF the process causing the smoke. Figure 3.39 illustrates the technique. The interrupt service procedure for handling the NMI interrupt is given in Listing 3.5.

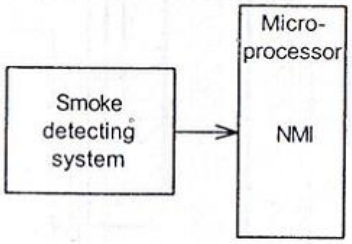


Figure 3.39 NMI interrupt.

Listing 3.5: Interrupt service procedure to handle NMI interrupt.

```
   PORT         EQU     00H    ; 8-bit port address
    CRTL_BYTE   DB      --     ; control byte to switch-off the
process
; Save the interrupt vector in vector location for the type 02H
    MOV    ES, AX
    MOV    WORD PTR ES:000A, SEG ISP
    MOV    WORD PTR ES:0008, OFFSET ISP
; Mainline program
HERE:JMP   HERE                ; remain in the loop and on interrupt
                                 execute ISP

    INT    03H                 ; interrupt service procedure
ISP:        PUSH          AX
    MOV    AL, CRTL_BYTE       ; get the control byte
    OUT    PORT, AL            ; send to output port to rotate one step
    POP    AX
    IRET                       ; return after servicing interrupt request
```


PREPARED BY
SHAHADAT HUSSAIN PARVEZ

INTR

- External devices requiring immediate attention interrupts microprocessor through this interrupt
- Logic high on INTR pin activates the interrupt
- 8086 gets interrupt type from external device (8259A priority interrupt controller)
- This interrupt is useful to interface slow device with microprocessor
- Set or reset IF (Interrupt Flag) bit in PSW
- IF is set or reset by STI and CLI instruction

Handling INTR interrupt

- 8259A can receive interrupt from upto 8 devices (**IR0-IR7**)
- Pass the signal onto microprocessor INTR pin
- Microprocessor acknowledge with 2 interrupt-acknowledge operation (Send 2 $\overline{INTA}$ pulse)
- The first pulse prepares 8259A to send the type for the interrupt
- In the second pulse microprocessor receives the type from 8259A
- The microprocessor then performs respective interrupt operation

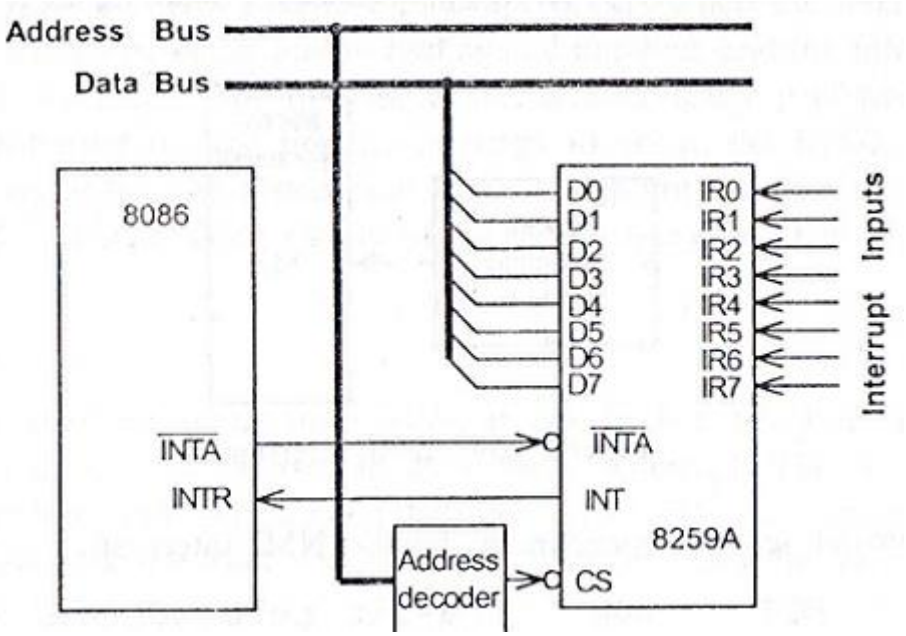


Figure 1 Handling INTR interrupt

Action when INTR is activated

INTR is level triggered input.

1. Complete the instruction in progress
2. Activate INTA o/p twice. In response 8086 receives ITN n instruction from an external device like 8259 PIC
3. Push Flags on the stack. Reset IE and T flags
4. PUSH CS
5. PUSH IP
6. IP loaded from word location n x 4 = say, W
7. CS loaded from next word location W+2

Processor makes a branch to the subroutine!

PREPARED BY
SHAHADAT HUSSAIN PARVEZ

Priority of 8086 interrupts

The priority of interrupts from highest to lowest is as follows

1. Divide by 0 error,
2. INT n
3. INTO
4. NMI
5. INTR
6. Single step interrupt

This means that a simultaneous divide by zero, and NMI/INT/Single-step interrupts cause the microprocessor to serve divide by zero interrupt first followed by other type of interrupt

1. What are the basic types of interrupt?
2. What is the difference between NMI and INTR?
3. Explain the execution of
    a. Type 0 interrupt
    b. Type 1 interrupt
    c. Type 2 interrupt
    d. Type 3 interrupt
    e. Type 4 interrupt
4. During an operation an 8086 got 3 interrupts at a time, NMI, INTR and INT n. What will the order in which 8086 will give attention?
5. What is the purpose of 8259A?
6. Explain a situation when NMI is useful.
7. What is the total size of interrupt vector table in 8086?

PREPARED BY
SHAHADAT HUSSAIN PARVEZ