Computer Interrupt

a *signal* indicating that an event needing immediate attention has occurred

2 Types of Interrupts:

External - generated outside CPU by other hardware

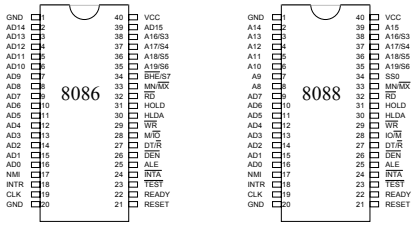
Internal - generated within CPU as a result of an instruction or operation

- x86 has internal interrupts: **`int`**, **`into`**, Divide Error and Single Step
- Trap generally means any processor generated interrupt
- in x86, Trap usually means the Single Step interrupt

x86 Interrupts:

1) Hardware Interrupt - External Uses INTR and NMI
2) Software Interrupt - Internal - from **`int`** or **`into`**
3) Processor Interrupt - Traps and 10 Software Interrupts (12 total)

8086/8088 Pinout Diagrams



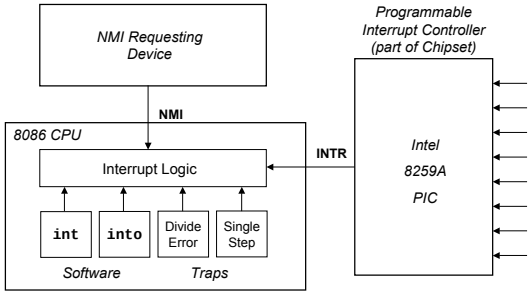
8086		
Pin	Mode	
	Minimum	Maximum
31	HOLD	RQ/GT0
30	HLDA	RQ/GT1
29	WR	LOCK
28	M/IO	S2
27	DT/R	S1
26	DEN	S0
25	ALE	QS0
24	INTA	QS1

8088		
Pin	Mode	
	Minimum	Maximum
31	HOLD	RQ/GT0
30	HLDA	RQ/GT1
29	WR	LOCK
28	IO/M	S2
27	DT/R	S1
26	DEN	S0
25	ALE	QS0
24	INTA	QS1
34	SS0	High State

**Minimum/Maximum Mode Refers to the Bus Handshaking*

8086 Interrupt Connections

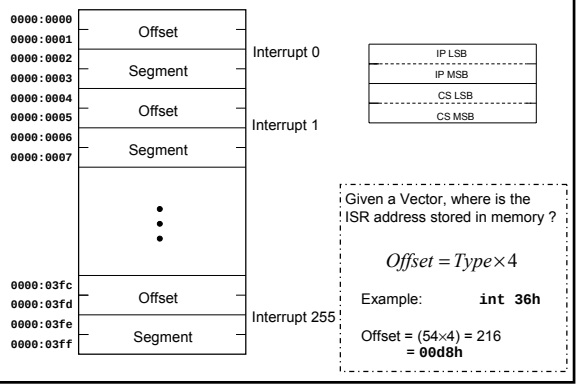
NMI - *Non-Maskable Interrupt* **INTR** - *Interrupt Request*



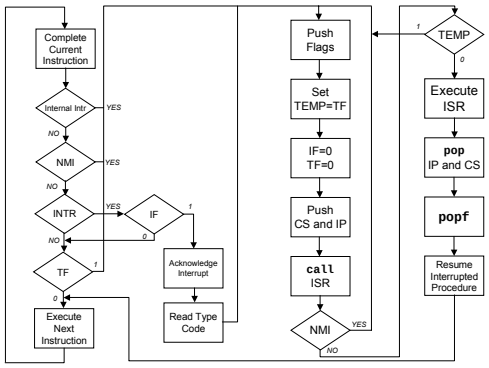
Interrupt Vector Table - IVT

- x86 Recognizes 256 Different Interrupts
  - Specified by *Type Number* or *Vector*
- 1 Byte of Data Must Accompany Each Interrupt Specifies *Type*
- *Vector* is Pointer into *Interrupt Vector Table*, *IVT*
  - Stored in Memory from **`0000:0000`** to **`0000:03ffh`**
- *IVT* Contains 256 Far Pointer Values
  - Far Pointer is **`CS:IP`** Values
- Each Far Pointer is Address of *Interrupt Service Routine*, *ISR*
  - Also Referred to as *Interrupt Handler*
- Table 4-6 of Uffenbeck (3rd edition, pg 197) has the interrupt vector numbers for the 80x86 family

IVT Format



What Happens During an Interrupt ?



Similarity to Subroutine Procedure

`call` ⇔ `int`

`ret` ⇔ `iret`

- **`call`** pushes **`CS`**, **`IP`** and loads **`CS:IP`** with address of subroutine
- **`int`** does what call does and more
- **`ret`** pops **`IP`**, **`CS`**
- **`iret`** pops FLAGS, **`IP`**, and **`CS`**

This is why ALL programs MUST have a stack segment, so that interrupts can be handled

Interrupt Acknowledge Cycles

The interrupt vector number for NMI is 2, so the location in the IVT for the NMI ISR address is 4*2 = 0x00008h.

Examination of Table 4-6 in Uffenbeck (3rd edition, pg 196) does not give an interrupt vector number for the INTR interrupt! What is the vector number for INTR?

Upon receipt of an INTR interrrupt, the 80x86 executes a two special bus cycles called an *Interrupt Acknowledge Cycle*.

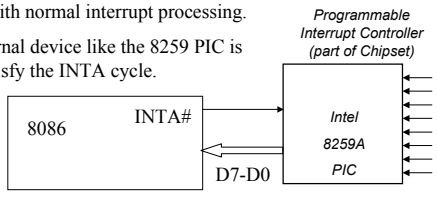
The purpose of an Interrupt Acknowledge Cycle is to fetch the interrupt vector number from the interrupting device via the D7-D0 lines.

Interrupt Acknowledge Cycles (cont)

The first INTA cycle (2 clks) asserts the INTA# line and alerts the interrupting device that it must be ready to provide the vector number.

The 2nd INTA cycle (2 clks) asserts the INTA# line and the interrupting device must provide the vector number via the lower 8 data lines (D7-D0). The 80x86 inputs the vector number, and then provides with normal interrupt processing.

Some external device like the 8259 PIC is used to satisfy the INTA cycle.



`halt` Instruction

- This instruction causes processor to enter a HALT state
- HALT state is one where no further instructions are fetched nor executed until one of the following events occurs:
  1) System is Reset - Rising Edge on RESET pin
  2) External Interrupt Occurs

CPU Asset	Content
FLAGS Register	**0000h**
IP	**0000h**
CS	**ffffh**
DS	**0000h**
SS	**0000h**
ES	**0000h**
Instruction Queue	Empty

Interrupt Vector Assignments

Type	Function	Comment
0	Divide Error	Processor - zero or overflow
1	Single Step (DEBUG)	Processor - TF=1
2	Nonmaskable Interrupt Pin	Processor - NMI Signal
3	Breakpoint	Processor - Similar to Sing Step
4	Arithmetic Overflow	Processor - into
5	Print Screen Key	BIOS - Key Depressed
6	Invalid Opcode	Processor - Invalid Opcode
7	Coprocessor Not Present	Processor - no FPU
8	Time Signal	BIOS - From RT Chip (AT - IRQ0)
9	Keyboard Service	BIOS - Gen Service (AT - IRQ1)
A - F	Originally Bus Ops (IBM PC)	BIOS - (AT - IRQ2-7)
10	Video Service Request	BIOS - Accesses Video Driver
11	Equipment Check	BIOS - Diagnostic
12	Memory Size	BIOS - DOS Memory
13	Disk Service Request	BIOS - Accesses Disk Driver
14	Serial Port Service Request	BIOS - Accesses Serial Port Drvr
15	Miscellaneous	BIOS - Cassette, etc.
16	Keyboard Service Request	BIOS - Accesses KB Driver

Interrupt Vector Assignments (cont)

Type	Function	Comment
17	Parallel Port LPT Service	BIOS - Printer Driver
18	ROM BASIC	BIOS - BASIC Interpreter in ROM
19	Reboot	BIOS - Bootstrap
1A	Clock Service	BIOS - Time of Day from BIOS
1B	Control-Break Handler	BIOS - Keyboard Break
1C	User Timer Service	BIOS - Timer Tick
1D	Pointer to Video Parm Table	BIOS - Video Initialization
1E	Pointer to Disk Parm Table	BIOS - Disk Subsystem Init.
1F	Pointer to Graphics Fonts	BIOS - CGA Graphics Fonts
20	Program Terminate	DOS - Clear Memory, etc.
21	Function Call	DOS - Transfer Control
22	Terminate Address	DOS - program Terminate handler
23	Control-C Handler	DOS - For OS Use
24	Fatal Error Handler	DOS - Critical Error
25	Absolute Disk Read	DOS - Disk Read
26	Absolute Disk Write	DOS - Disk Write
27	Terminate	DOS - TSR Usage
28	Idle Signal	DOS - Idle
2F	Print Spool	DOS - Cassette, etc.
70-77	Hardware Interrupts in AT Bios	DOS - (AT - IRQs 8-15)

AT – IRQ Definitions

IBM-AT (Advanced Technology) - Intel 80286)

Name	Interrupt Vector	Priority	Description
NMI	02	1	Memory Parity Error
IRQ0	08	2	Timer (Intel 8253 Chip 55 ms intervals)
IRQ1	09	3	Keyboard
IRQ2	0A	4	8259 PIC Slave or EGA/VGA Vert. Retrace
IRQ3	0B	13	Serial Port (COM2 or COM4)
IRQ4	0C	14	Serial Port (COM1 or COM3)
IRQ5	0D	15	Fixed Disk or LPT2 Request
IRQ6	0E	16	Floppy Disk Driver
IRQ7	0F	17	LPT1 Request
IRQ8	70	5	CMOS Real-Time Clock (RT Chip)
IRQ9	71	6	Re-directed to IRQ2
IRQ10	72	7	RESERVED
IRQ11	73	8	RESERVED
IRQ12	74	9	Mouse or other
IRQ13	75	10	Math Coprocessor (NPX)
IRQ14	76	11	Hard Disk
IRQ15	77	12	RESERVED