

## Intel Defined CPU Exception Table (see notes)

Interrupt	Function
0	Divide by zero
1	Single step
2	Non-maskable (NMI)
3	Breakpoint
4	Overflow trap
5	BOUND range exceeded (186,286,386)
6	Invalid opcode (186,286,386)
7	Coprocessor not available (286,386)
8	Double fault exception (286,386)
9	Coprocessor segment overrun (286,386)
A	Invalid task state segment (286,386)
B	Segment not present (286,386)
C	Stack exception (286,386)
D	General protection exception (286,386)
E	Page fault (286,386)
F	Reserved
10	Coprocessor error (286,386)

## IBM PC Hardware Interrupt Table (in order of priority)

IRQ#	Interrupt	Function
IRQ0	8	<a href="#">timer</a> (55ms intervals, 18.2 per second)
IRQ1	9	keyboard service required
IRQ2	A	slave <a href="#">8259</a> or EGA/VGA vertical retrace
IRQ8	70	real time clock (AT,XT286,PS50+)
IRQ9	71	software redirected to IRQ2 (AT,XT286,PS50+)
IRQ10	72	reserved (AT,XT286,PS50+)
IRQ11	73	reserved (AT,XT286,PS50+)
IRQ12	74	mouse interrupt (PS50+)
IRQ13	75	numeric coprocessor error (AT,XT286,PS50+)
IRQ14	76	fixed disk controller (AT,XT286,PS50+)
IRQ15	77	reserved (AT,XT286,PS50+)
IRQ3	B	COM2 or COM4 service required, (COM3-COM8 on MCA PS/2)
IRQ4	C	COM1 or COM3 service required
IRQ5	D	fixed disk or data request from LPT2
IRQ6	E	floppy disk service required
IRQ7	F	data request from LPT1 (unreliable on IBM mono)

## Interrupt Table as Implemented by System BIOS/DOS

INT #	Locus	Function
0	CPU	divide by zero
1	CPU	single step

2	CPU	non-maskable
3	CPU	breakpoint
4	CPU	overflow trap
5	BIOS	print screen
6	CPU	Invalid opcode (186,286,386)
7	CPU	coprocessor not available (286,386)
8	IRQ0	<a href="#">timer</a> (55ms intervals, 18.21590 per second)
9	IRQ1	keyboard service required (see <a href="#">INT 9</a> )
A	IRQ2	slave <a href="#">8259</a> or EGA/VGA vertical retrace
B	IRQ3	COM2 service required (PS/2 MCA COM3-COM8)
C	IRQ4	COM1 service required
D	IRQ5	fixed disk or data request from LPT2
E	IRQ6	floppy disk service required
F	IRQ7	data request from LPT1 (unreliable on IBM mono)
10	BIOS	video (see <a href="#">INT 10</a> )
11	BIOS	Equipment determination (see <a href="#">INT 11</a> )
12	BIOS	memory size (see <a href="#">INT 12</a> )
13	BIOS	disk I/O service (see <a href="#">INT 13</a> )
14	BIOS	serial communications (see <a href="#">INT 14</a> )
15	BIOS	system services, cassette (see <a href="#">INT 15</a> )
16	BIOS	keyboard services (see <a href="#">INT 16</a> )
17	BIOS	parallel printer (see <a href="#">INT 17</a> )
18	BIOS	ROM BASIC loader
19	BIOS	bootstrap loader (unreliable, see <a href="#">INT 19</a> )
1A	BIOS	time of day (see <a href="#">INT 1A</a> )
1B	BIOS	user defined ctrl-break handler (see <a href="#">INT 1B</a> )
1C	BIOS	user defined clock tick handler (see <a href="#">INT 1C</a> )
1D	BIOS	<a href="#">6845</a> video parameter pointer
1E	BIOS	diskette parameter pointer (base table)
1F	BIOS	graphics character table
20	DOS	general program termination
21	DOS	function request services (see <a href="#">INT 21</a> )
22	DOS	terminate address (see <a href="#">INT 22</a> )
23	DOS	control break termination address (see <a href="#">INT 23</a> )
24	DOS	critical error handler (see <a href="#">INT 24</a> )
25	DOS	absolute disk read (see <a href="#">INT 25</a> )
26	DOS	absolute disk write (see <a href="#">INT 26</a> )
27	DOS	terminate and stay resident (see <a href="#">INT 27</a> )
28	DOS	idle loop, issued by DOS when idle (see <a href="#">INT 28</a> )
29	DOS	fast TTY console I/O (see <a href="#">INT 29</a> )
2A	DOS	critical section and NETBIOS (see <a href="#">INT 2A</a> )
2B	DOS	internal, simple <a href="#">IRET</a> in DOS 2.0-5.0
2C	DOS	internal, simple IRET in DOS 2.0-5.0
2D	DOS	internal, simple IRET in DOS 2.0-5.0
2E	DOS	exec command from base level command interpreter (see <a href="#">INT 2E</a> )
2F	DOS	multiplexer (see <a href="#">INT 2F</a> )
30-31	CPM	far jump vector for CPM (not an interrupt)
31	DPMI	DOS Protected Mode Interface (for DOS extenders)
32		reserved
33		mouse support (see <a href="#">INT 33</a> )
34-3E		Microsoft/Borland floating point emulation
3F		overlay manager
40	BIOS	hard disk
41	BIOS	fixed disk 0 parameters pointer (see <a href="#">INT 13,9</a> )
42	BIOS	relocated video handler (EGA/VGA/PS)
43	BIOS	user font table (EGA/VGA/PS)
44	BIOS	first 128 graphics characters (also Netware)

45	BIOS	reserved for BIOS
46	BIOS	fixed disk 1 parameters ptr (see <a href="#">INT 13,9</a> /INT 41)
47	BIOS	reserved for BIOS
48	BIOS	PCjr cordless keyboard translation
49	BIOS	PCjr non-keyboard scancode translation table
4A	BIOS	user alarm (AT,CONV,PS/2) (see <a href="#">INT 4A</a> )
4B-4F	BIOS	reserved
50	BIOS	periodic alarm from timer (PS/2)
51-58	BIOS	reserved
59	BIOS	GSS Computer Graphics Interface
5A	BIOS	cluster adapter BIOS entry point
5B	BIOS	cluster adapter boot
5C	NETBIOS	NETBIOS interface, TOPS interface
5D-5F	BIOS	reserved for BIOS
60-67		reserved for user software interrupts
67	EMS	LIM/EMS specification (see <a href="#">INT 67</a> )
68		APPC
69-6B		reserved by IBM
6C	DOS	DOS 3.2 real time clock update
	BIOS	system resume vector
6D-6F		reserved
70	IRQ8	real time clock (AT,XT286,PS50+, see <a href="#">INT 15</a> )
71	IRQ9	software redirected to IRQ2 (AT,XT286,PS50+)
72	IRQ10	reserved (AT,XT286,PS50+)
73	IRQ11	reserved (AT,XT286,PS50+)
74	IRQ12	mouse interrupt (PS50+)
75	IRQ13	numeric coprocessor NMI error (AT,XT286,PS50+)
76	IRQ14	fixed disk controller (AT,XT286,PS50+)
77	IRQ15	reserved (AT,XT286,PS50+)
78-79		unused
80-85		ROM BASIC
86-F0	DOS	reserved for BASIC interpreter use
86	NETBIOS	NETBIOS relocated INT 18
E0	CPM	CP/M 86 function calls
F1-FF		reserved by IBM
FE-FF		may be destroyed by return from protected mode using VDISK on 286 machines (Apr 86, DDJ)

- Intel defined 0 through 20h for use for internal CPU; IBM redefined interrupts 0 through 1Fh for its own use, hence the duplicate definitions in the tables
- all interrupts except the internal CPU exceptions push the flags and the CS:IP of the next instruction onto the stack. CPU exception interrupts are similar but push the CS:IP of the causal instruction. 8086/88 divide exceptions are different, they return to the instruction following the division
- interrupts are disabled upon entry into any interrupt routine and should be enabled by the user or by an [IRET](#)
- in DOS 3.2+ hardware IRQ interrupts are re-vectored through DOS to provide standard stack frames