8086 Interrupts

Introduction

- An interrupt is used to cause a temporary halt in the execution of program.
- The meaning of ‘interrupts’ is to break the sequence of operation.
- An Interrupt is a special condition that arises during the working of a Microprocessor

Introduction

- **INTERRUPTS:** 2 main types of interrupt in the 8086 microprocessor,

i. **Internal & External Hardware Interrupts**

ii. **Edge or Level sensitive Interrupts**

iii. **Maskable Interrupts & Non Maskable Interrupts**

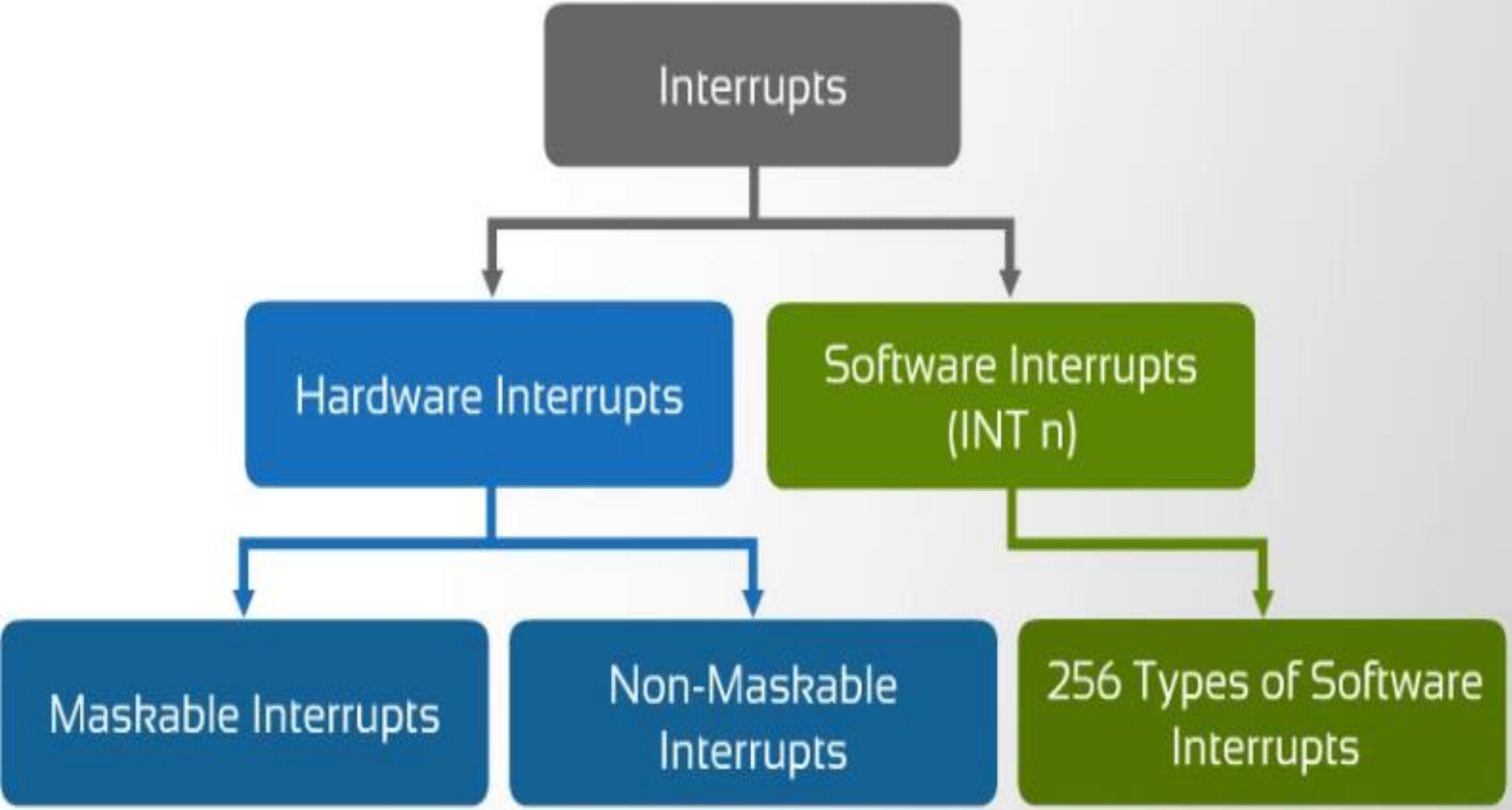
Need for Interrupt:

- Interrupts are particularly useful when interfacing I/O devices, that provide or require data at relatively low data transfer rate.

Sources of Interrupts

Three types of interrupts sources:

1. An **external signal** applied to NMI or INTR input pin **(Hardware Interrupt)**
2. Execution of **Special Interrupt Instruction** **(Software Interrupt)**
3. Interrupt raised due to some **Error Condition** produced in 8086 instruction execution process. **(Divide By Zero, Overflow Errors etc)**



Interrupts

Hardware Interrupts

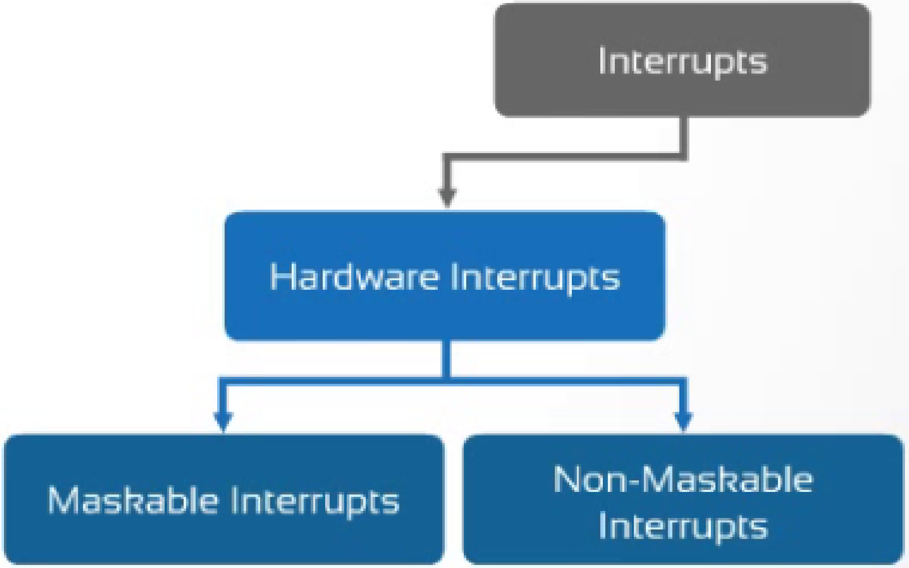
Software Interrupts (INT n)

Maskable Interrupts

Non-Maskable Interrupts

256 Types of Software Interrupts

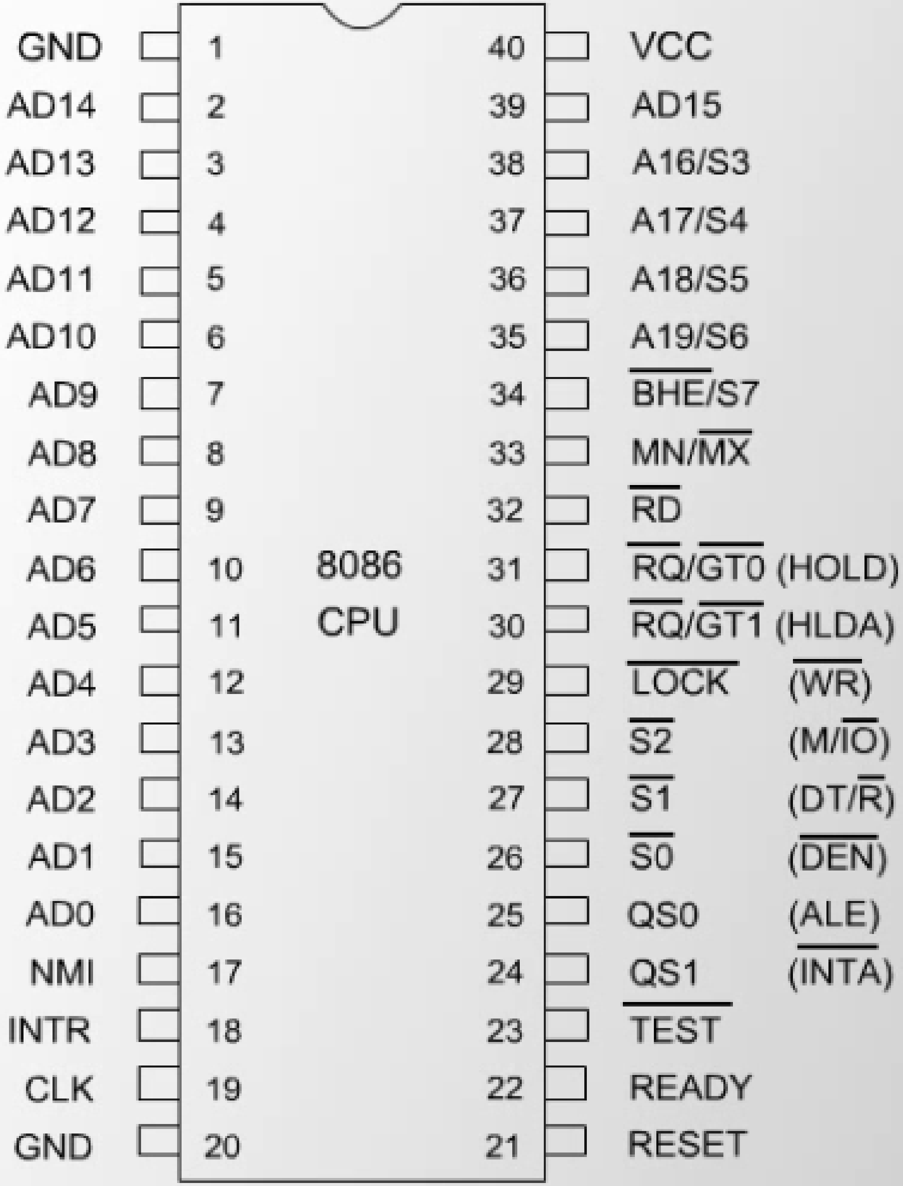
8086 CPU


Interrupts

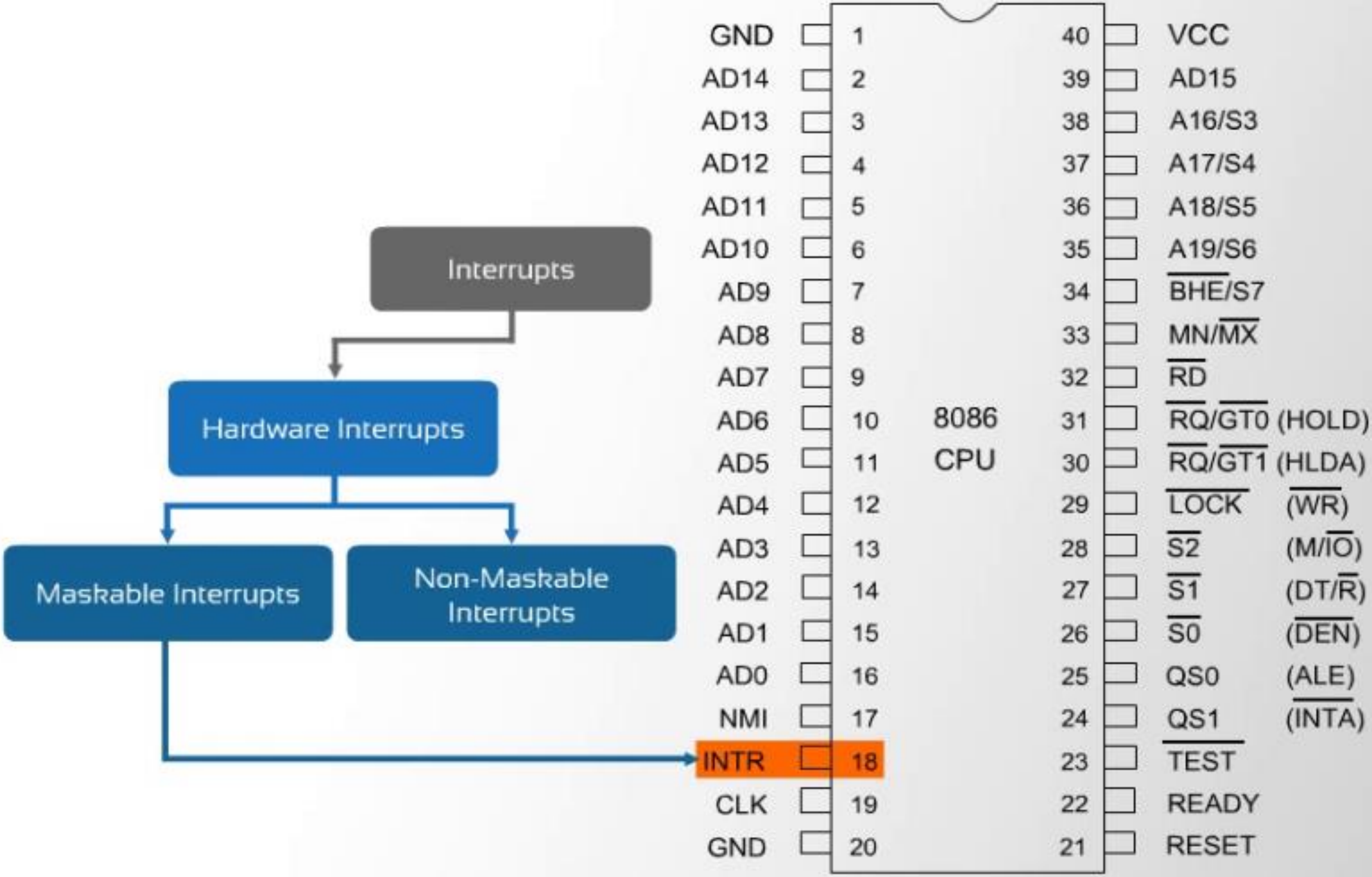
Hardware Interrupts

Maskable Interrupts

Non-Maskable Interrupts



8086 CPU



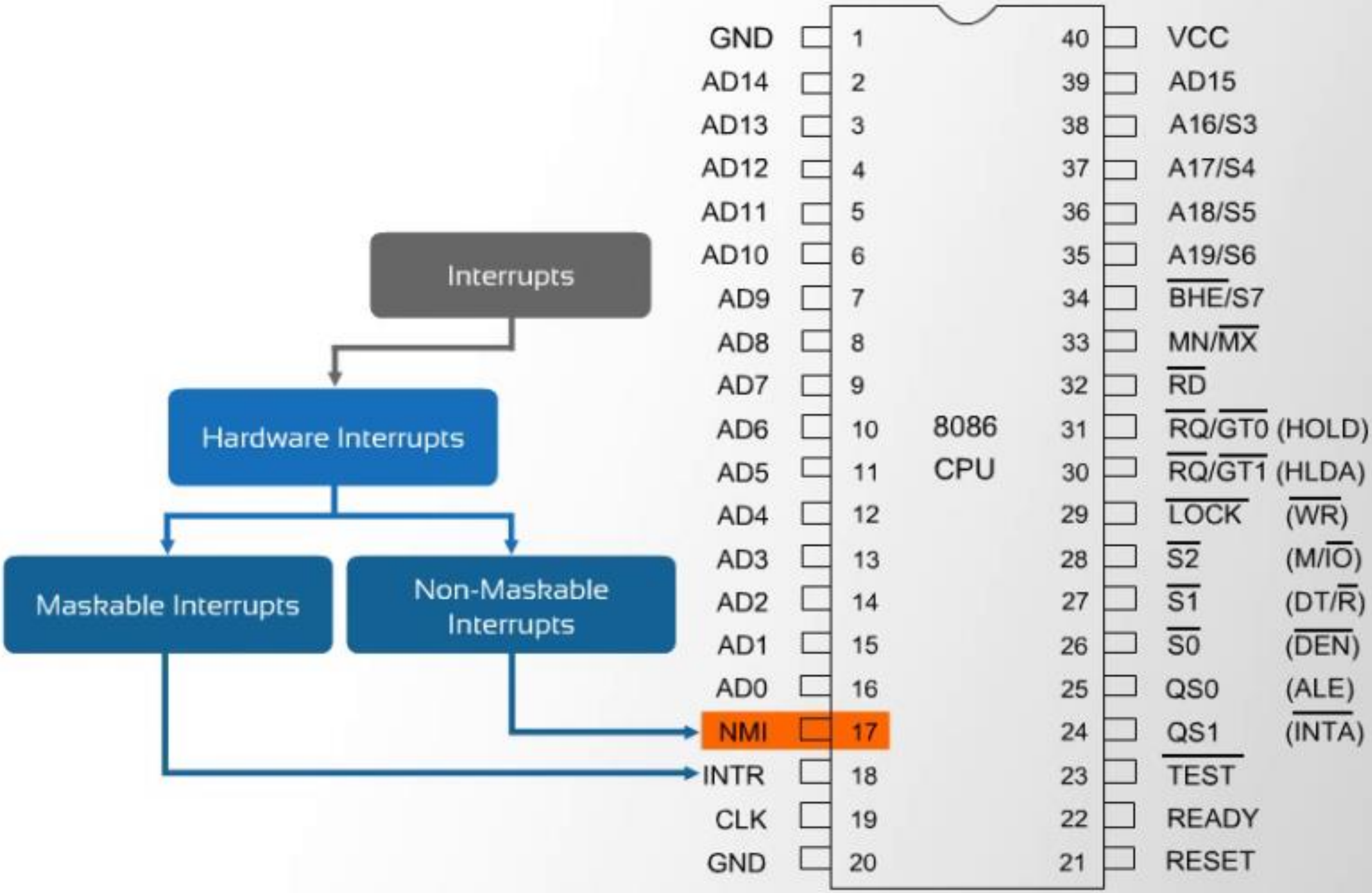
Interrupts

Hardware Interrupts

Maskable Interrupts

Non-Maskable Interrupts

8086 CPU



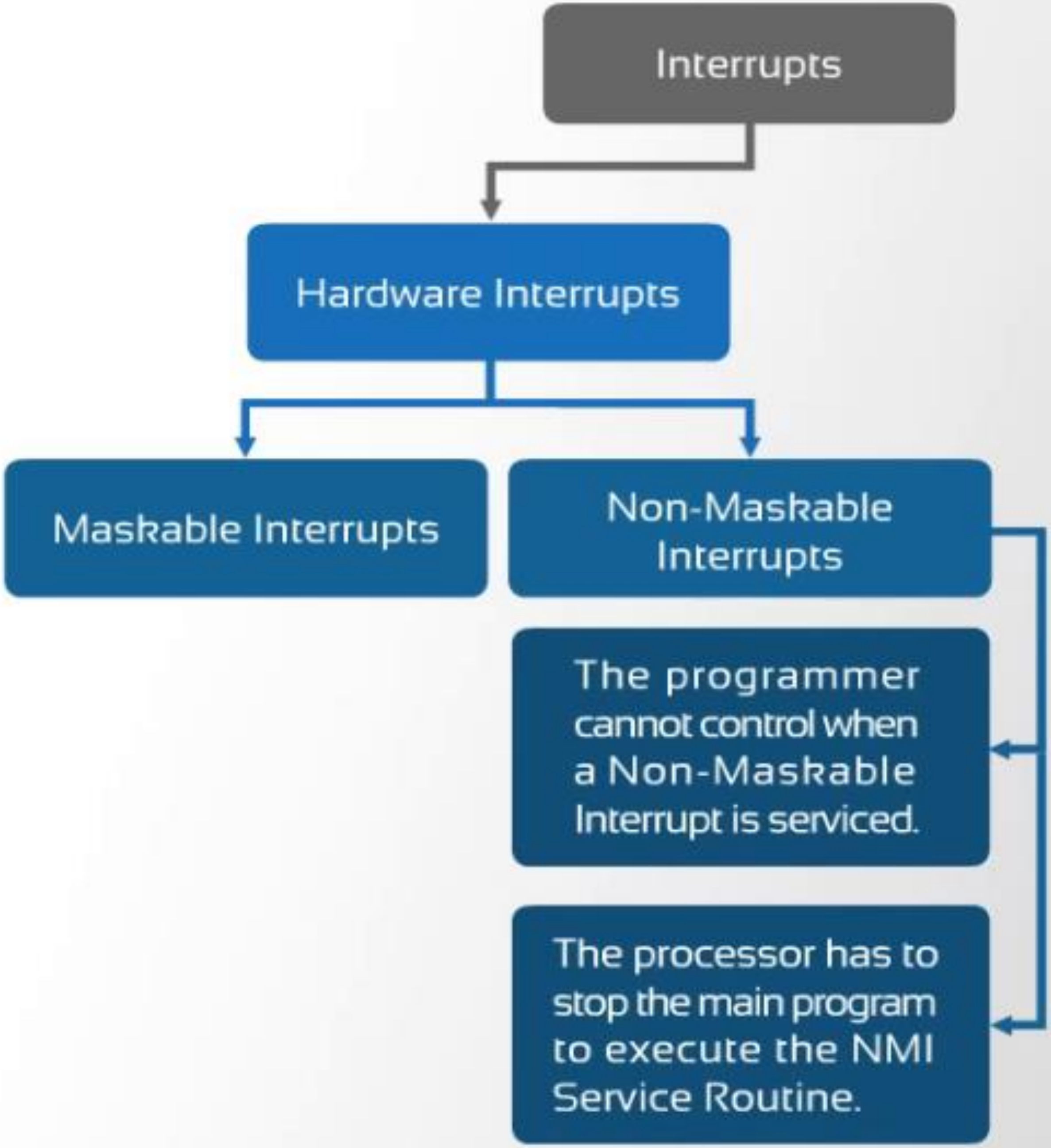
Interrupts

Hardware Interrupts

Maskable Interrupts

Non-Maskable Interrupts

Maskable Versus Non-Maskable Interrupts



Interrupts

Hardware Interrupts

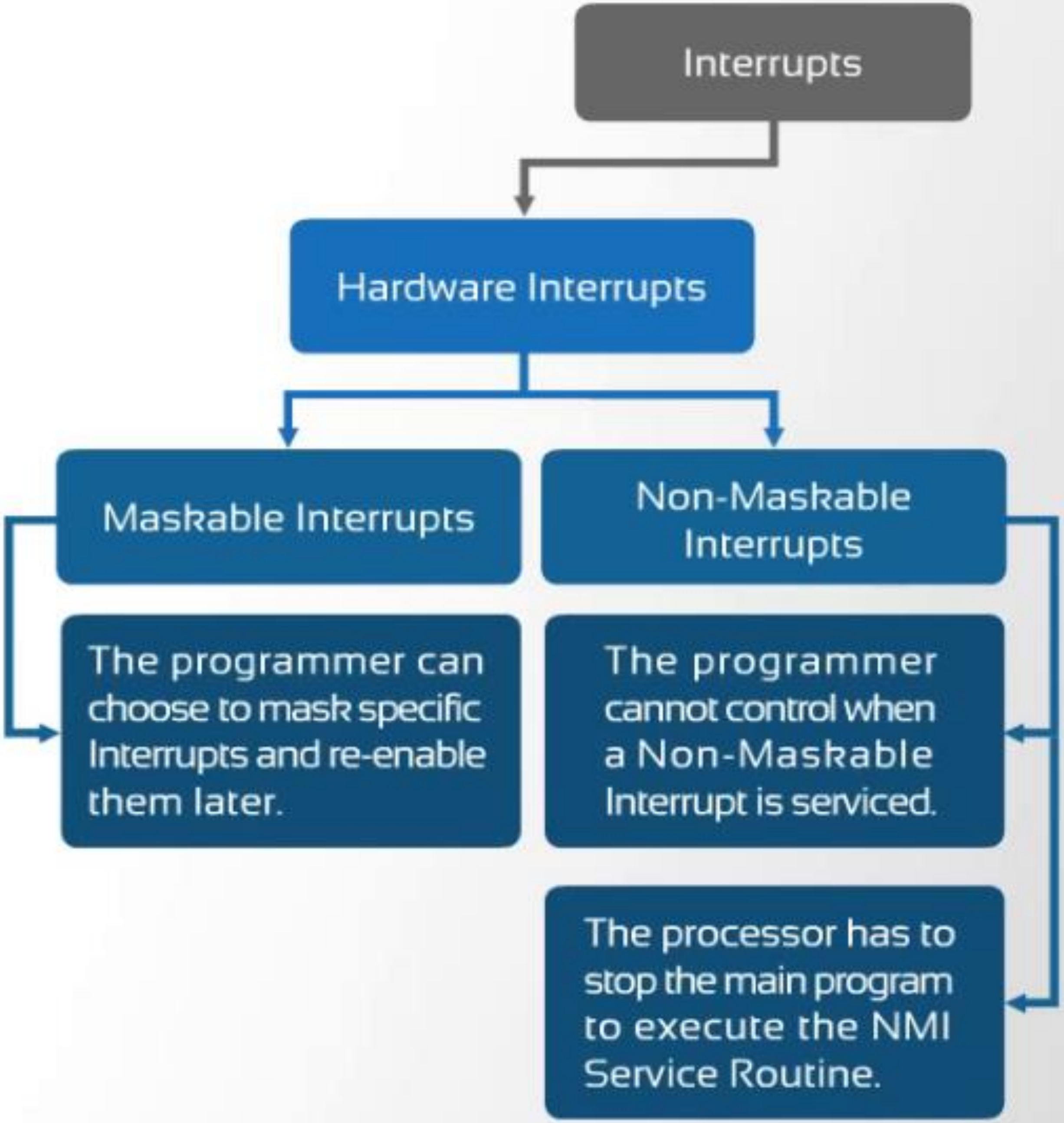
Maskable Interrupts

Non-Maskable Interrupts

The programmer cannot control when a Non-Maskable Interrupt is serviced.

The processor has to stop the main program to execute the NMI Service Routine.

Maskable Versus Non-Maskable Interrupts



Interrupts

Hardware Interrupts

Maskable Interrupts

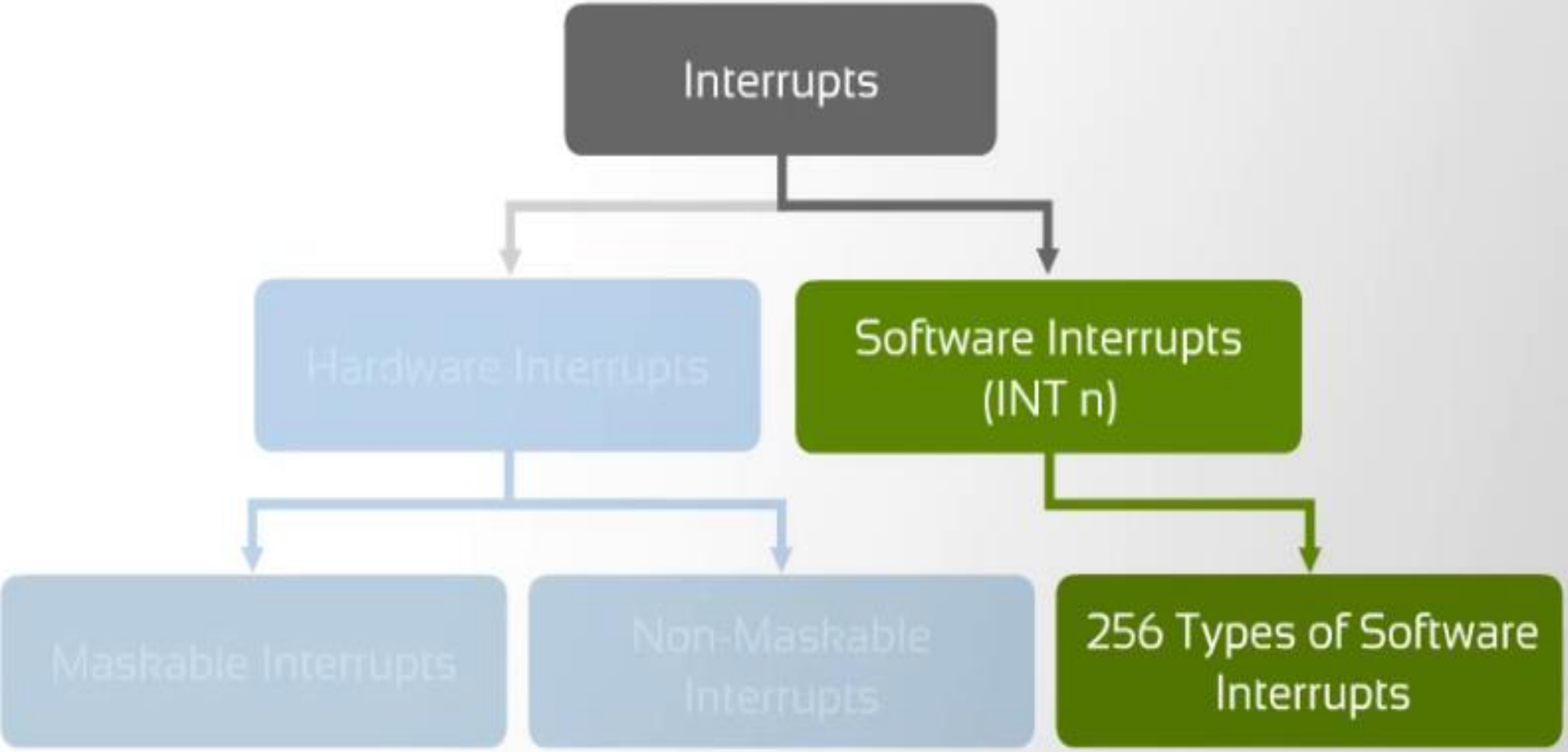
Non-Maskable Interrupts

The programmer can choose to mask specific Interrupts and re-enable them later.

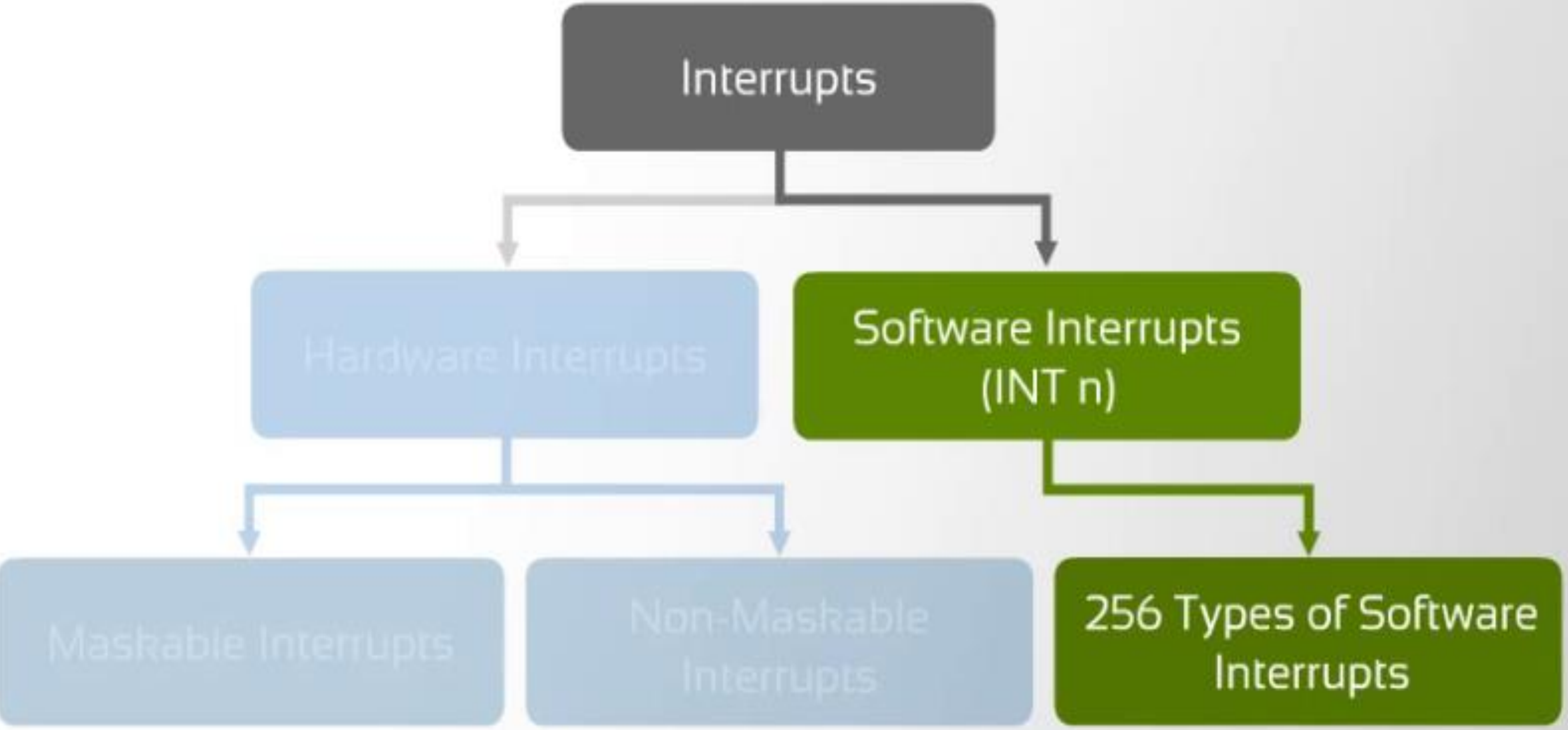
The programmer cannot control when a Non-Maskable Interrupt is serviced.

The processor has to stop the main program to execute the NMI Service Routine.

INT stands for Interrupt.



INT 00h – INT 0FFh comprises 256 kinds of Interrupts.



Interrupts

Hardware Interrupts

Software Interrupts (INT n)

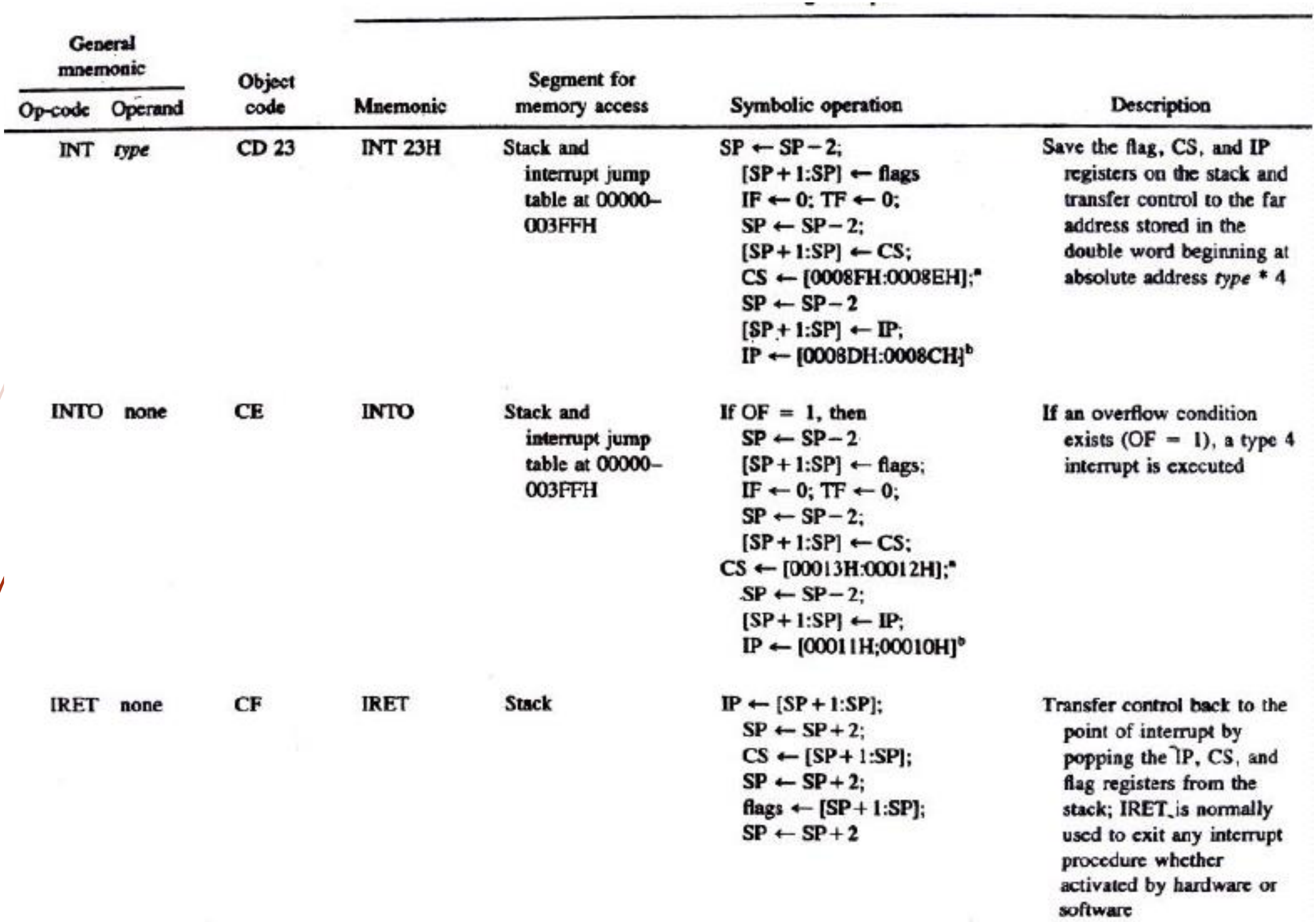
Maskable Interrupts

Non-Maskable Interrupts

256 Types of Software Interrupts

8086 - Instruction Set – Transfer Control of Instructions

- Software Interrupts



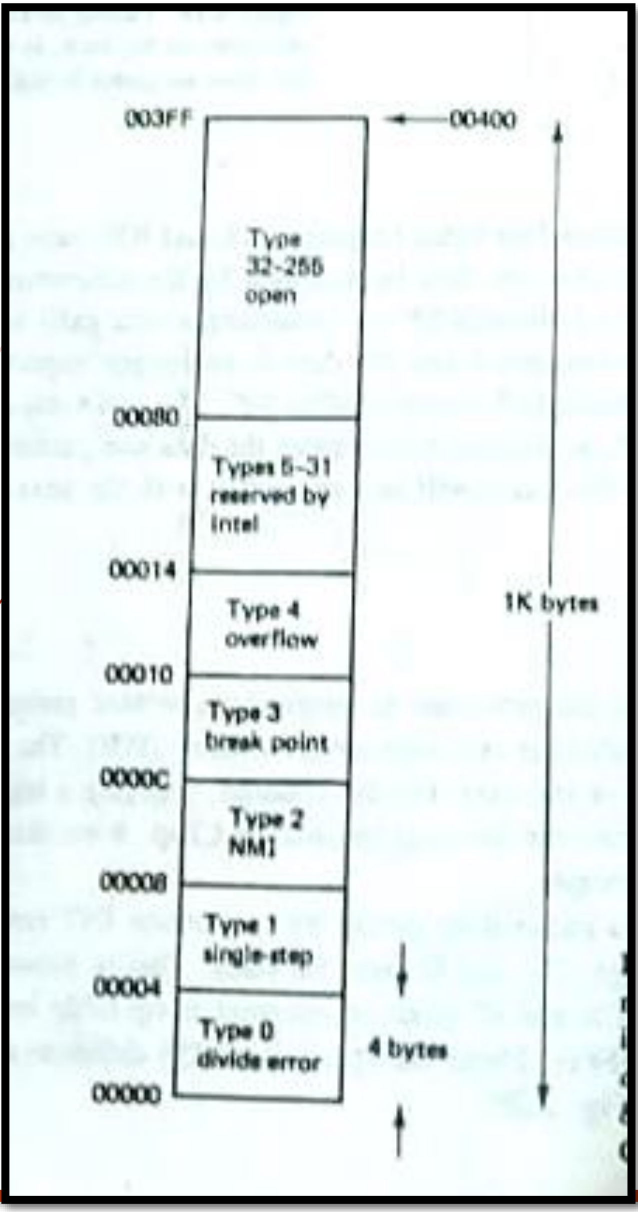
General mnemonic		Object code	Mnemonic	Segment for memory access	Symbolic operation	Description
Op-code	Operand					

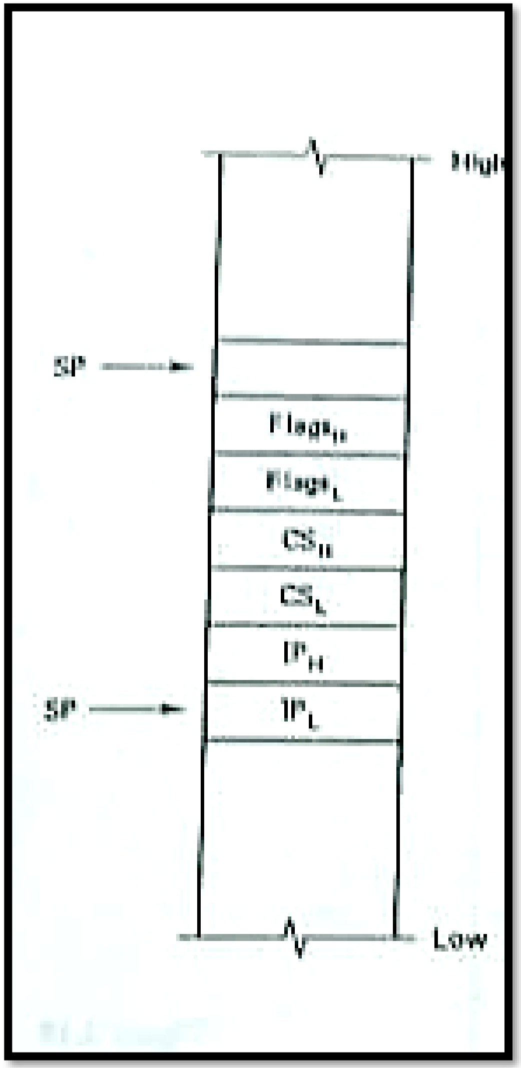
INT	*type*	CD 23	INT 23H	Stack and interrupt jump table at 00000–003FFH	SP ← SP – 2;
					[SP + 1:SP] ← flags
					IF ← 0; TF ← 0;
					SP ← SP – 2;
					[SP + 1:SP] ← CS;
					CS ← [0008FH:0008EH];[a]
					SP ← SP – 2
					[SP + 1:SP] ← IP;
					IP ← [0008DH:0008CH][b]
						Save the flag, CS, and IP registers on the stack and transfer control to the far address stored in the double word beginning at absolute address *type* * 4

INTO	none	CE	INTO	Stack and interrupt jump table at 00000–003FFH	If OF = 1, then
					SP ← SP – 2
					[SP + 1:SP] ← flags;
					IF ← 0; TF ← 0;
					SP ← SP – 2;
					[SP + 1:SP] ← CS;
					CS ← [00013H:00012H];[a]
					SP ← SP – 2;
					[SP + 1:SP] ← IP;
					IP ← [00011H;00010H][b]
						If an overflow condition exists (OF = 1), a type 4 interrupt is executed

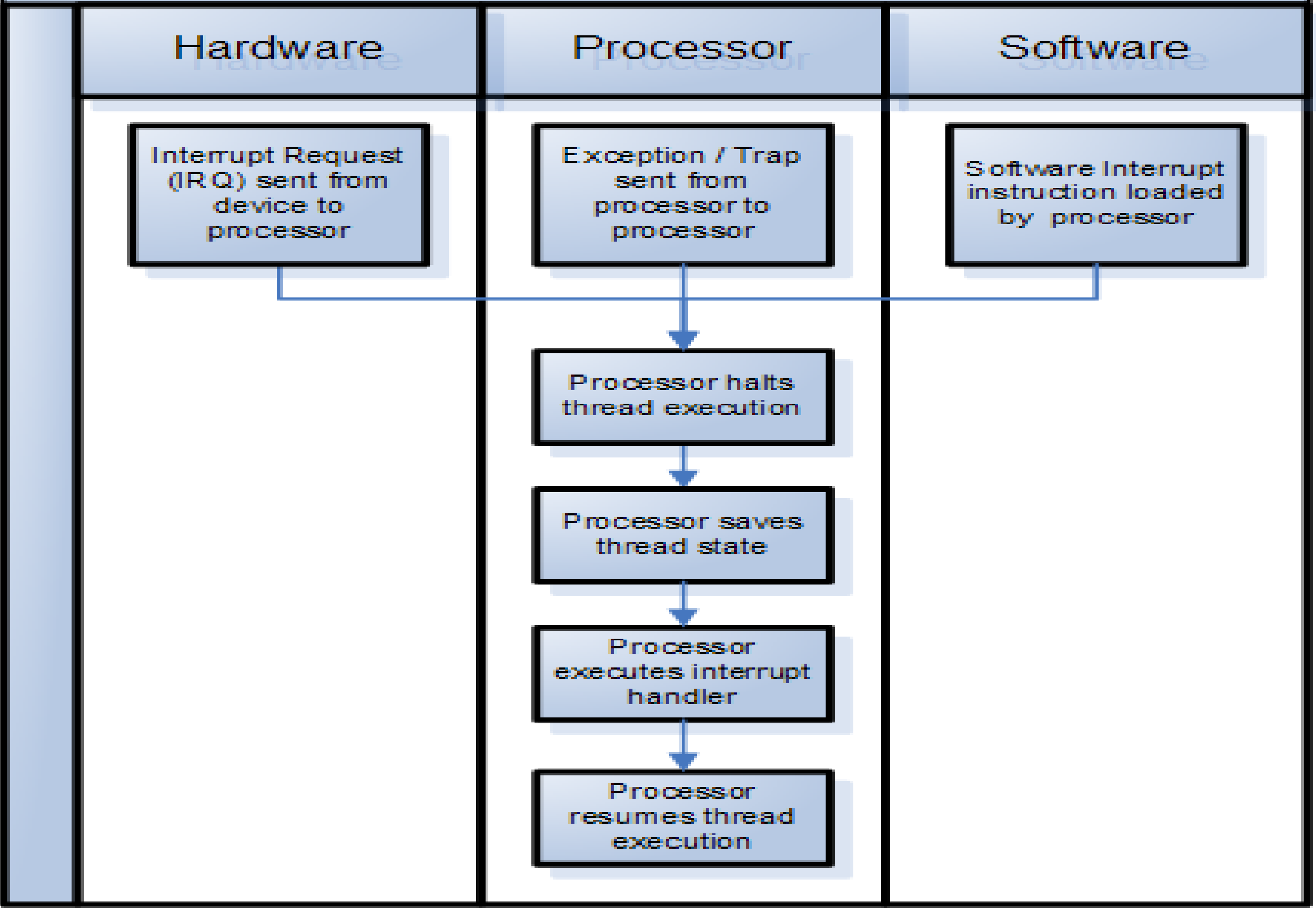
IRET	none	CF	IRET	Stack	IP ← [SP + 1:SP];
					SP ← SP + 2;
					CS ← [SP + 1:SP];
					SP ← SP + 2;
					flags ← [SP + 1:SP];
					SP ← SP + 2
						Transfer control back to the point of interrupt by popping the IP, CS, and flag registers from the stack; IRET is normally used to exit any interrupt procedure whether activated by hardware or software

Software Interrupts





Interrupt Process (from three potential sources)



Hardware

Processor

Software

Interrupt Request (IRQ) sent from device to processor

Exception / Trap sent from processor to processor

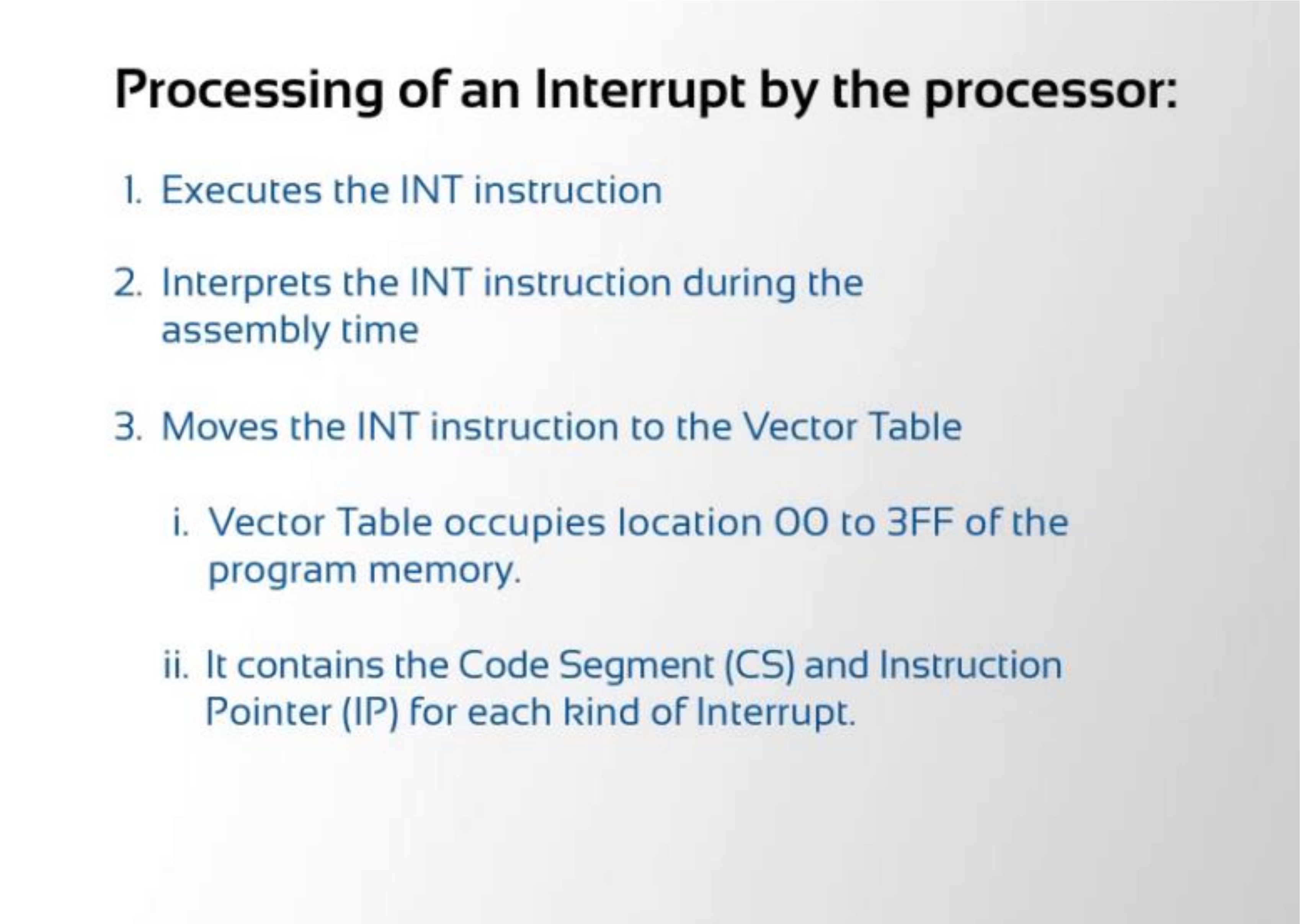
Software Interrupt instruction loaded by processor

Processor halts thread execution

Processor saves thread state

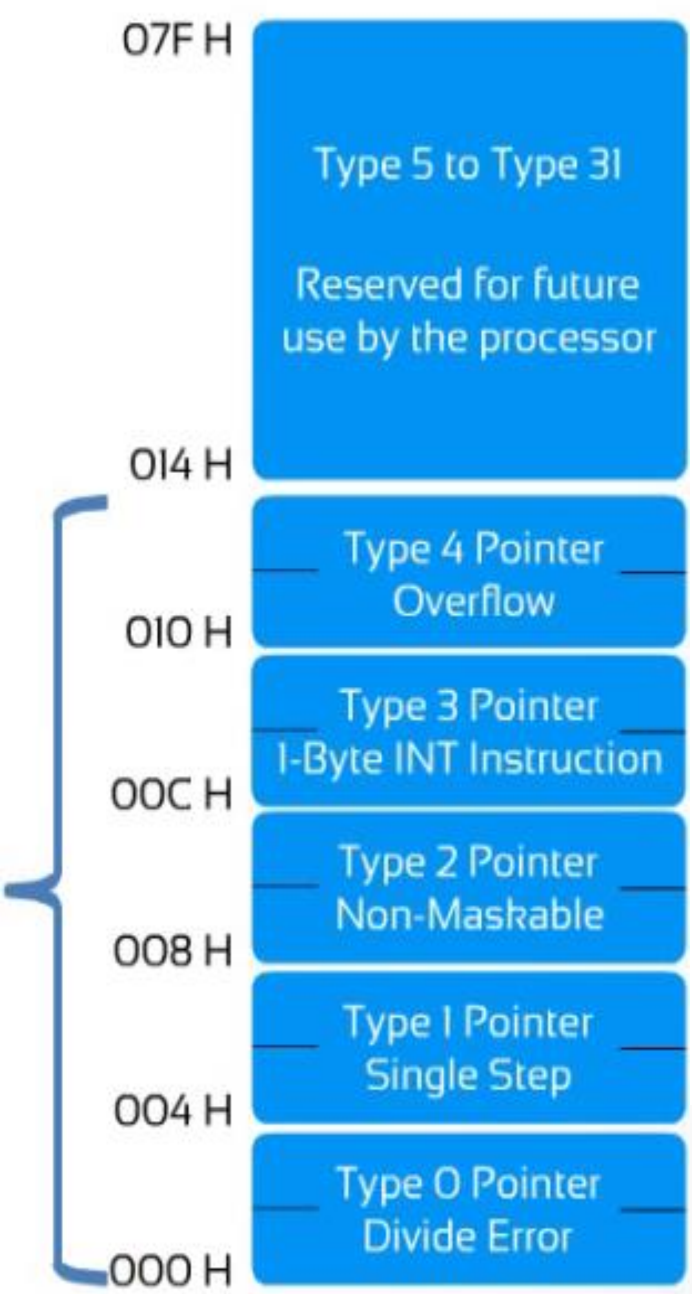
Processor executes interrupt handler

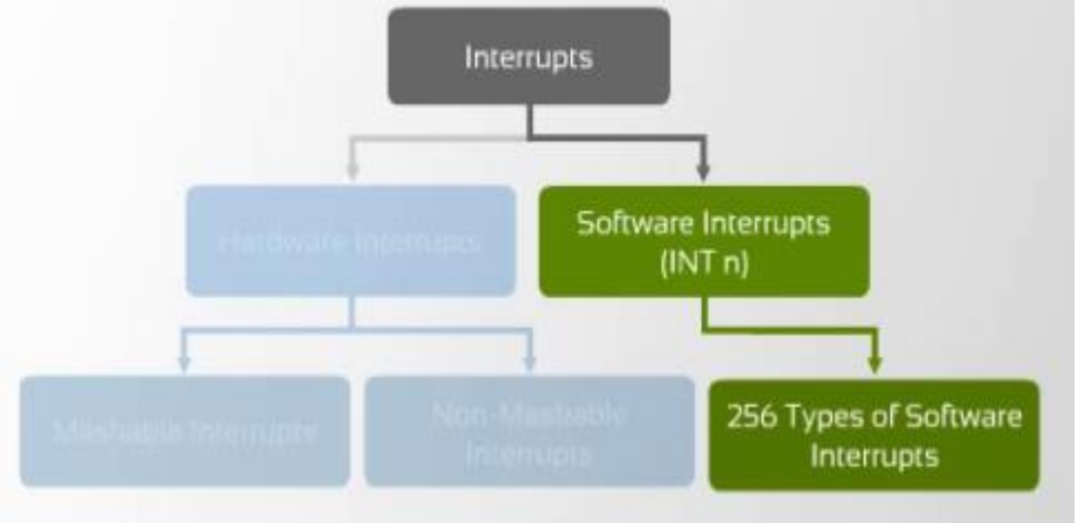
Processor resumes thread execution



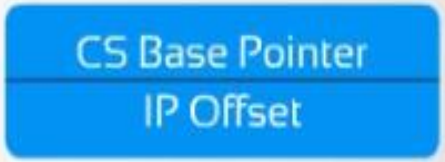
Processing of an Interrupt by the processor:

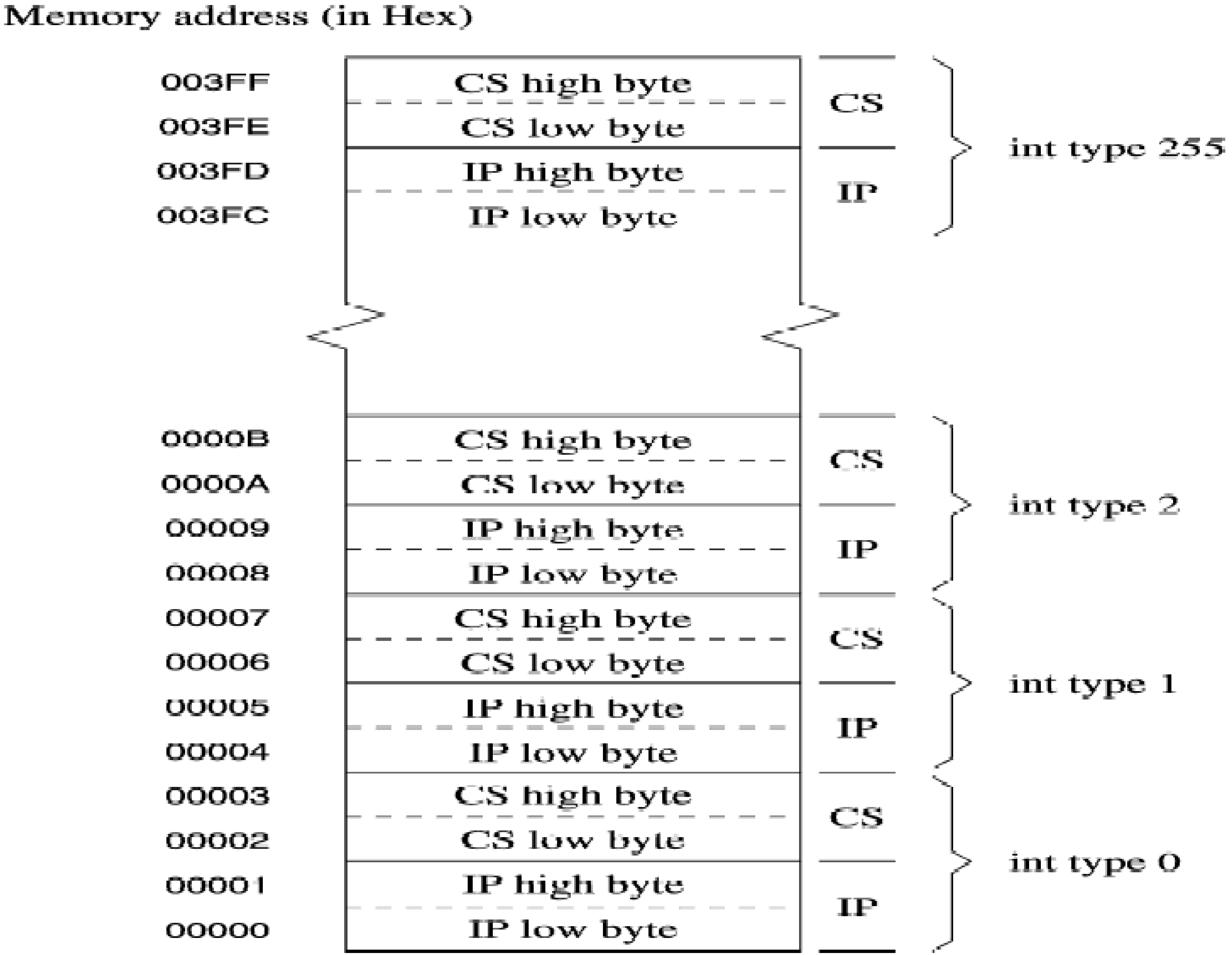
1. Executes the INT instruction
2. Interprets the INT instruction during the assembly time
3. Moves the INT instruction to the Vector Table
   1. Vector Table occupies location 00 to 3FF of the program memory.
   2. It contains the Code Segment (CS) and Instruction Pointer (IP) for each kind of Interrupt.

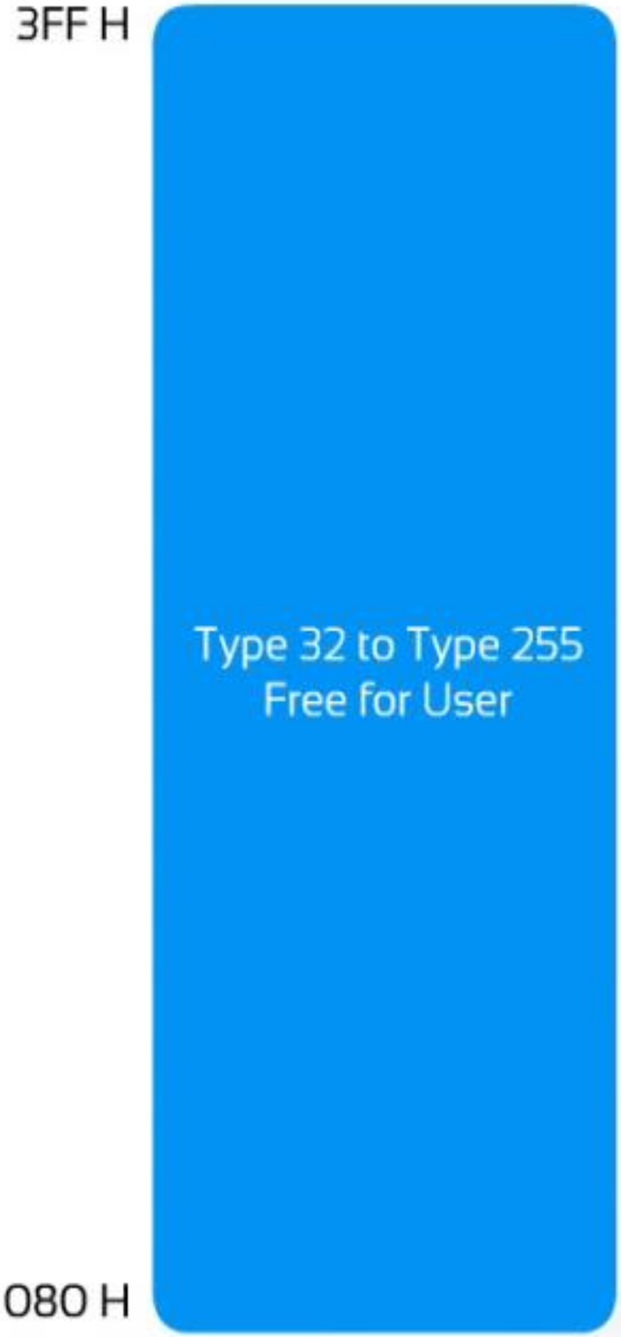




- IVT table contains ISR address for the 256 interrupts
- Each ISR address is stored as CS & IP
- ISR address is of 4 bytes (2 CS & 2 IP) & hence requires 4 locations to save
- Thus for 256 interrupts: total size of IVT table = 256x4 = 1 KB
- 1st 1KB of memory 00000h------003FFH reserved for IVT
- INT N ----→µP does Nx4 to get the value of IP & CS of the ISR



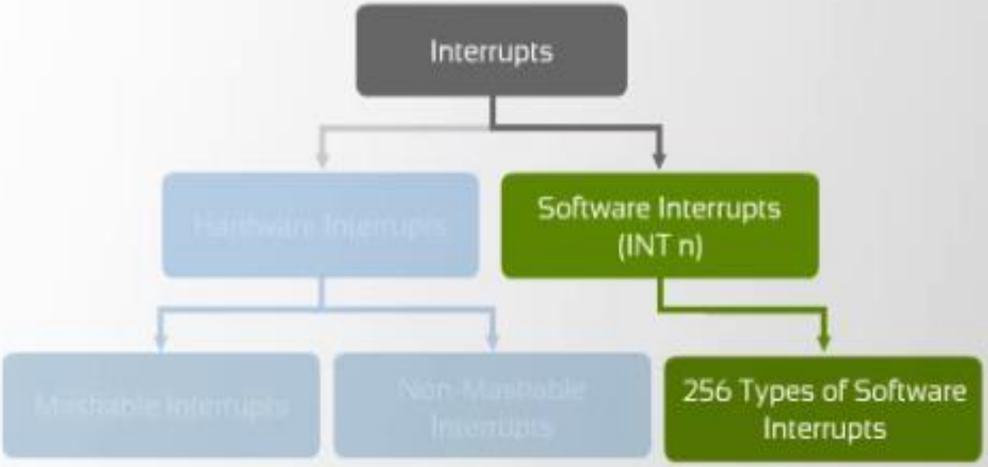

Memory address (in Hex)
003FF
003FE
003FD
003FC
CS high byte
CS low byte
IP high byte
IP low byte
CS
IP
int type 255
0000B
0000A
00009
00008
CS high byte
CS low byte
IP high byte
IP low byte
CS
IP
int type 2
00007
00006
00005
00004
CS high byte
CS low byte
IP high byte
IP low byte
CS
IP
int type 1
00003
00002
00001
00000
CS high byte
CS low byte
IP high byte
IP low byte
CS
IP
int type 0



3FF H

Type 32 to Type 255
Free for User

080 H



Interrupts

Hardware Interrupts

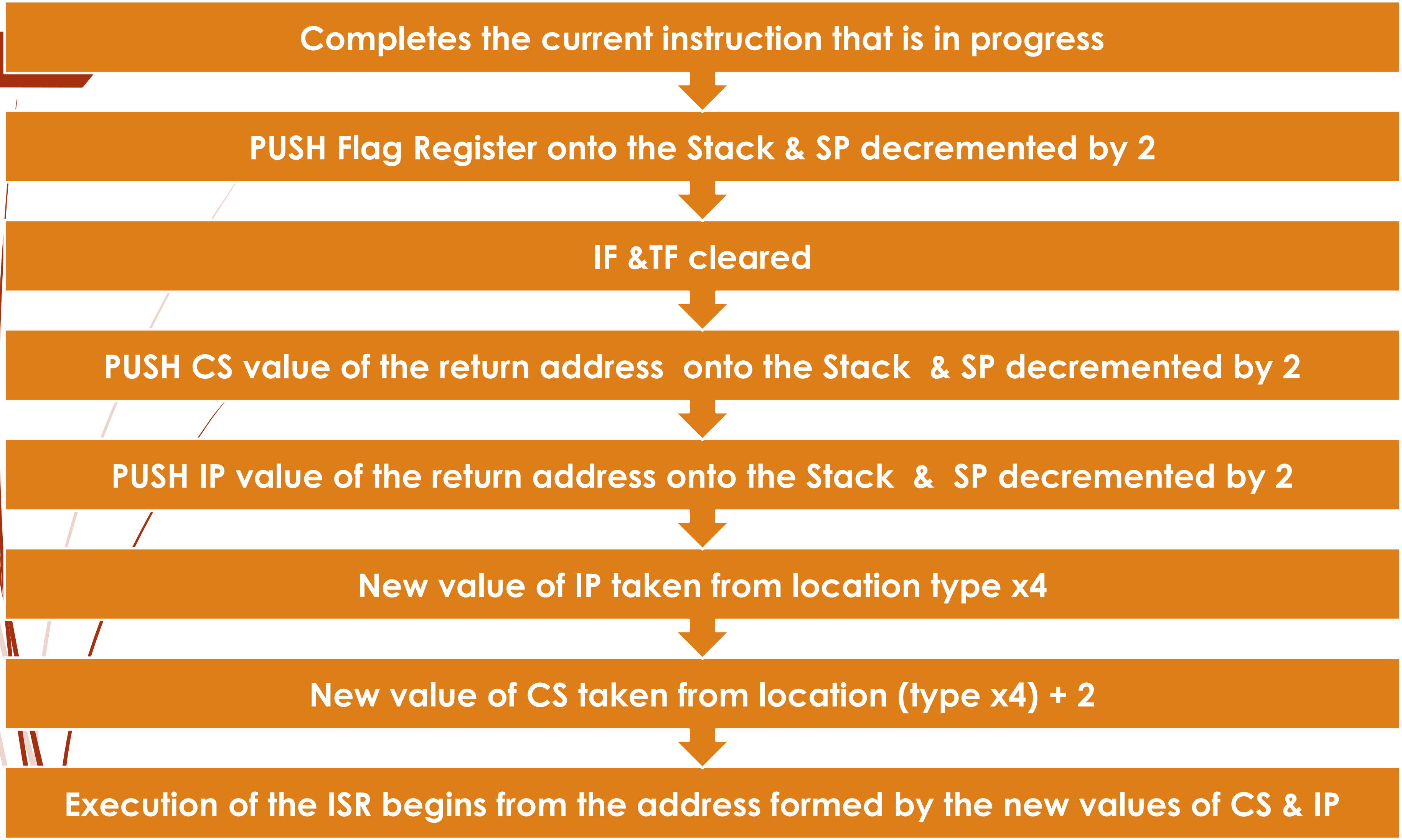
Software Interrupts
(INT n)

Maskable Interrupts

Non-Maskable
Interrupts

256 Types of Software
Interrupts

Response to any interrupt



Completes the current instruction that is in progress

PUSH Flag Register onto the Stack & SP decremented by 2

IF &TF cleared

PUSH CS value of the return address onto the Stack & SP decremented by 2

PUSH IP value of the return address onto the Stack & SP decremented by 2

New value of IP taken from location type x4

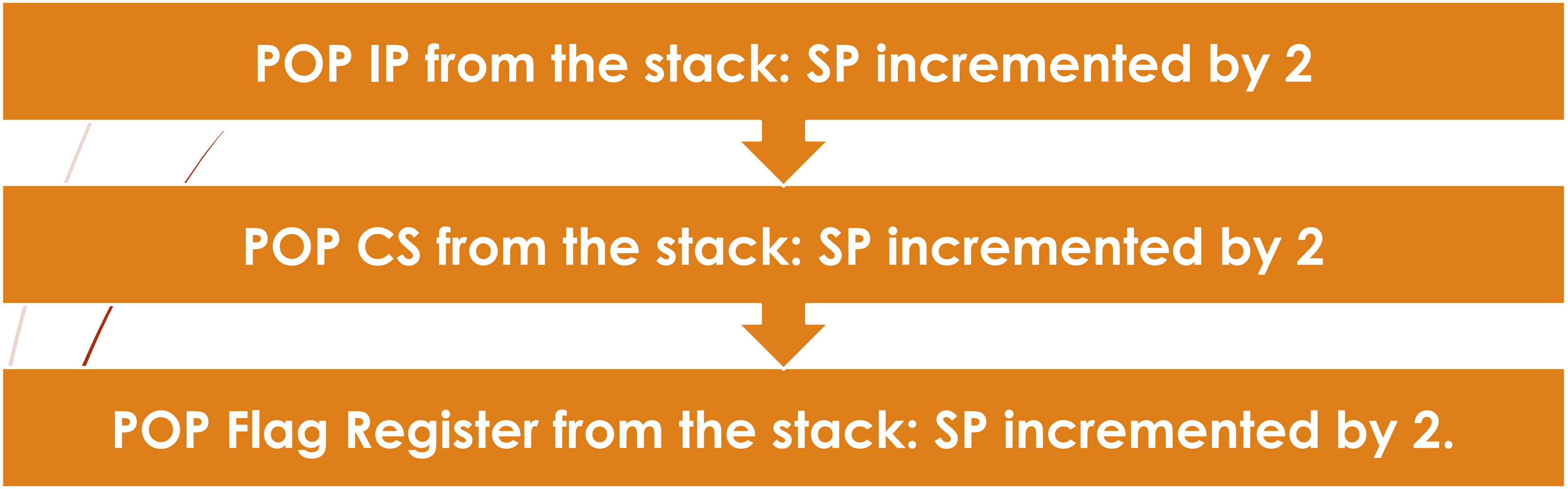
New value of CS taken from location (type x4) + 2

Execution of the ISR begins from the address formed by the new values of CS & IP

- Eg: INT 1 ;IP = {[00004] & [00005]}; CS = {[00006] & [00007]};
  as 1x4=00004H

Response to any IRET instruction

- This instruction causes 8086 to return to the main program.
- Used at the end of the ISR

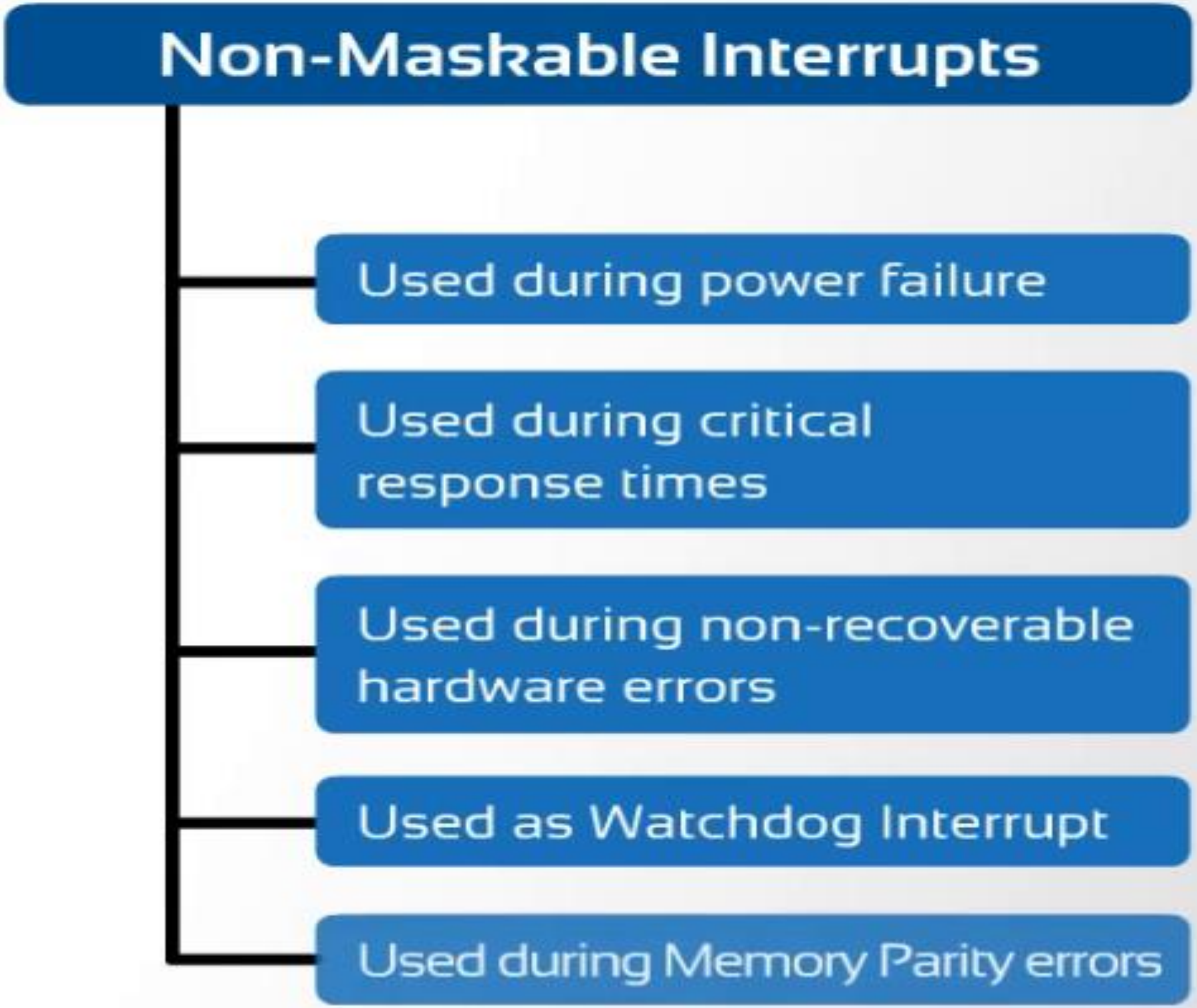


POP IP from the stack: SP incremented by 2

POP CS from the stack: SP incremented by 2

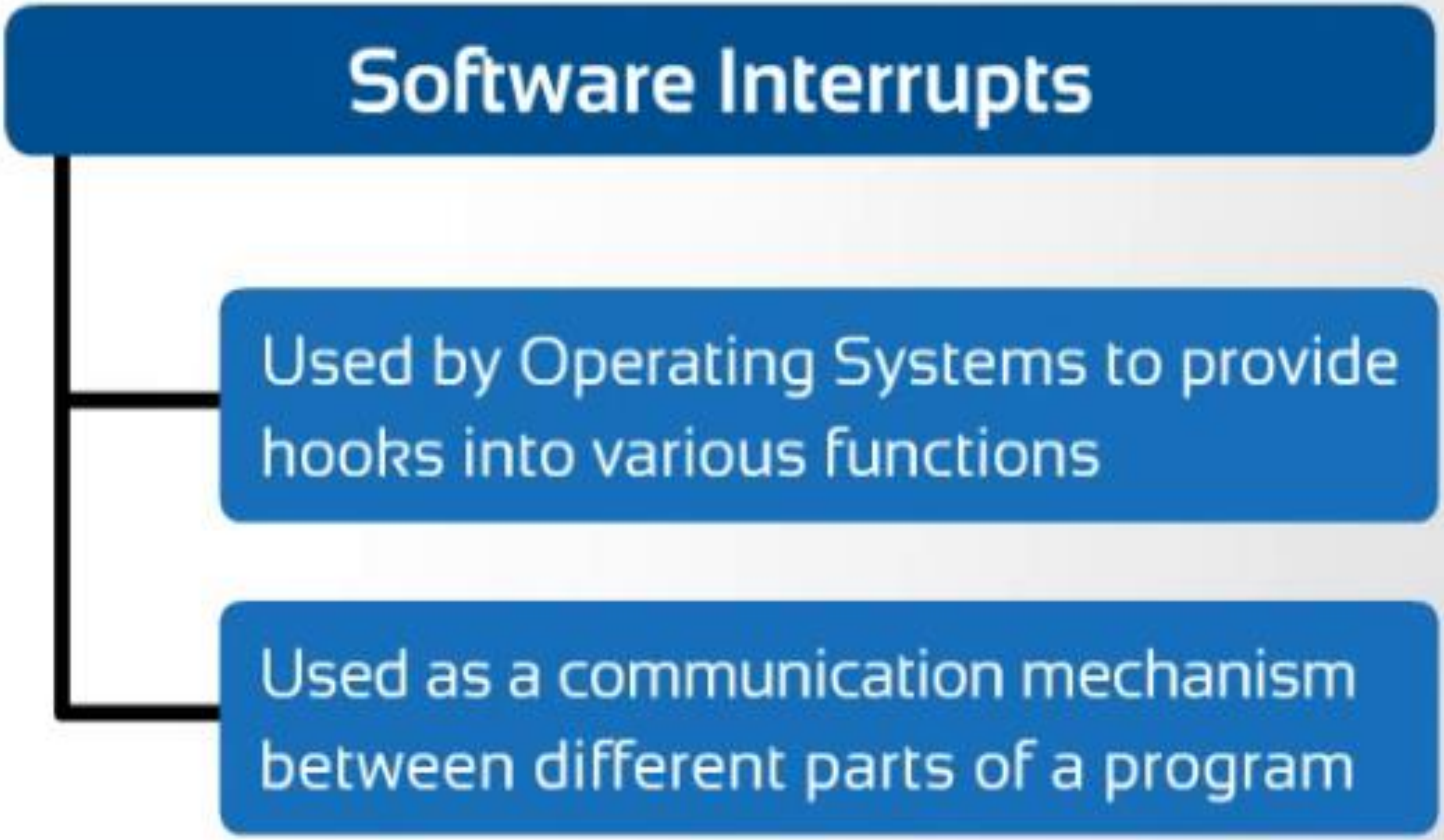
POP Flag Register from the stack: SP incremented by 2.

- Execution of the Main Program continues from the address formed by CS & IP



Non-Maskable Interrupts

- Used during power failure
- Used during critical response times
- Used during non-recoverable hardware errors
- Used as Watchdog Interrupt
- Used during Memory Parity errors



Software Interrupts

Used by Operating Systems to provide hooks into various functions

Used as a communication mechanism between different parts of a program

Hardware Interrupts

Used to handle external hardware peripherals, such as keyboards, mouse, hard disks, floppy disks, DVD drives, and printers

Keyboard

Mouse

Hard disk

Floppy disk

DVD drive

PRIORITY OF INTERRUPTS

Interrupt Type	Priority
INT0, INT3-INT 255,	Highest
NMI(INT2)	↓
INTR	↓
Single Step	Lowest

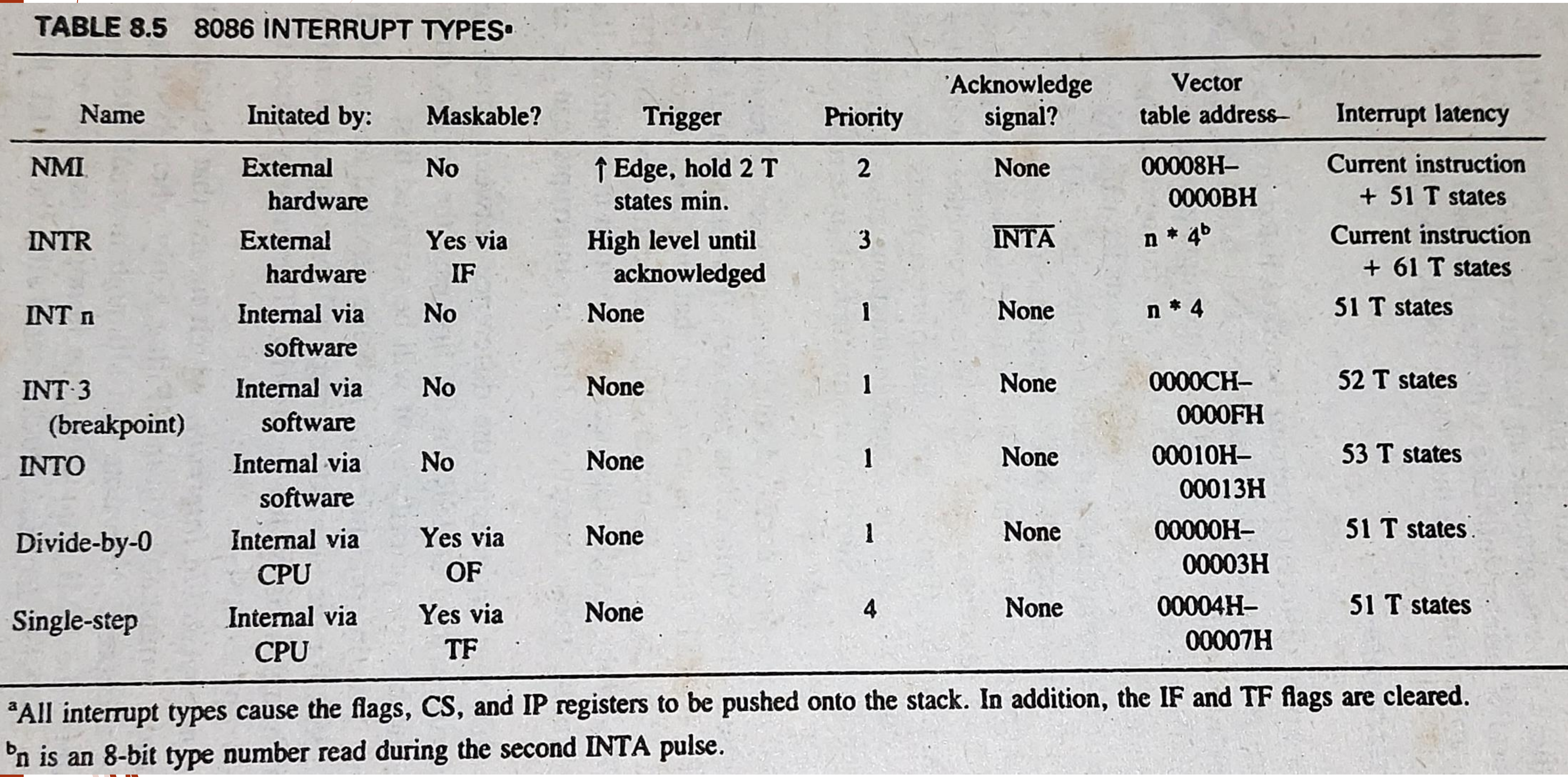


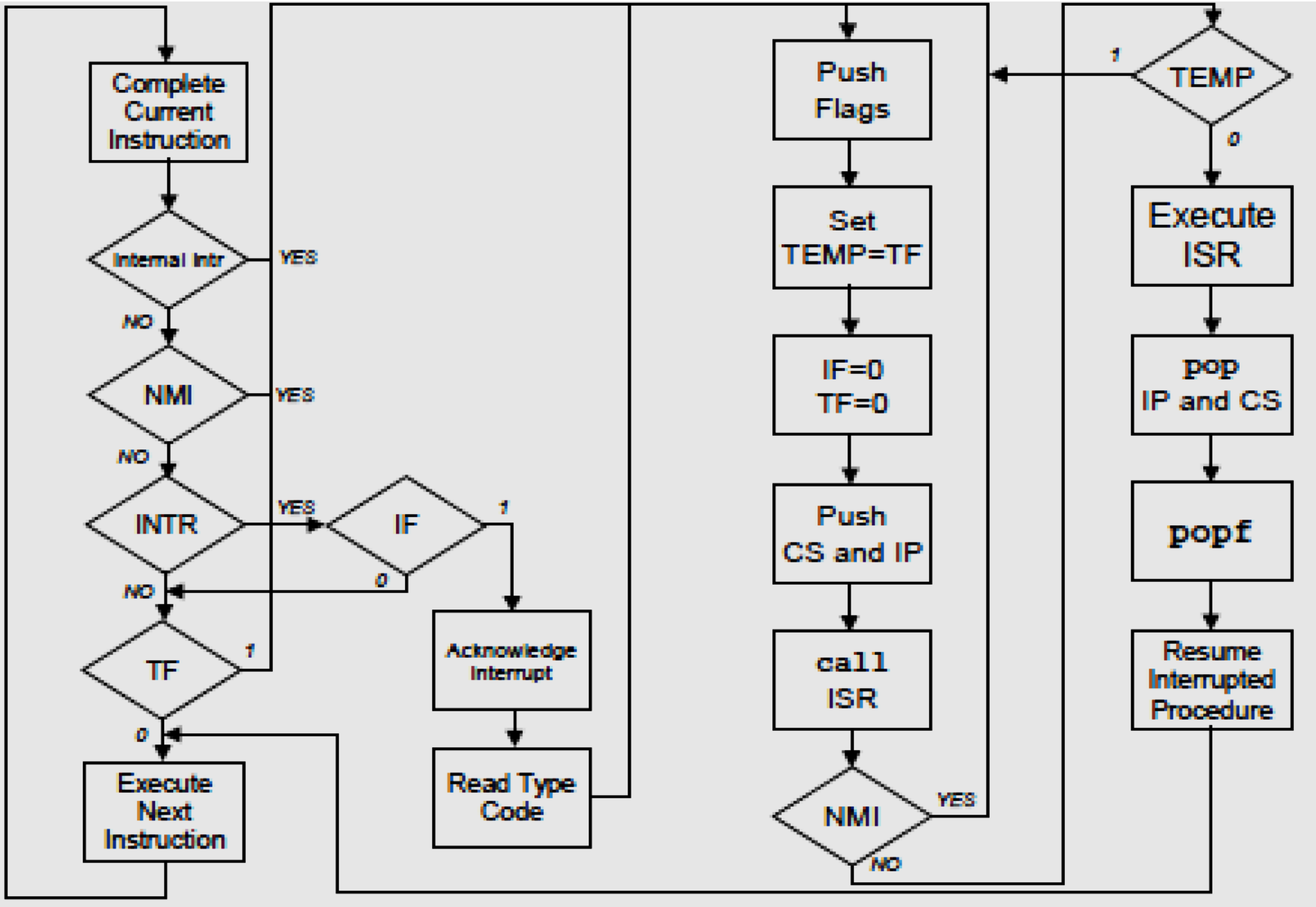
TABLE 8.5 8086 INTERRUPT TYPES[a]

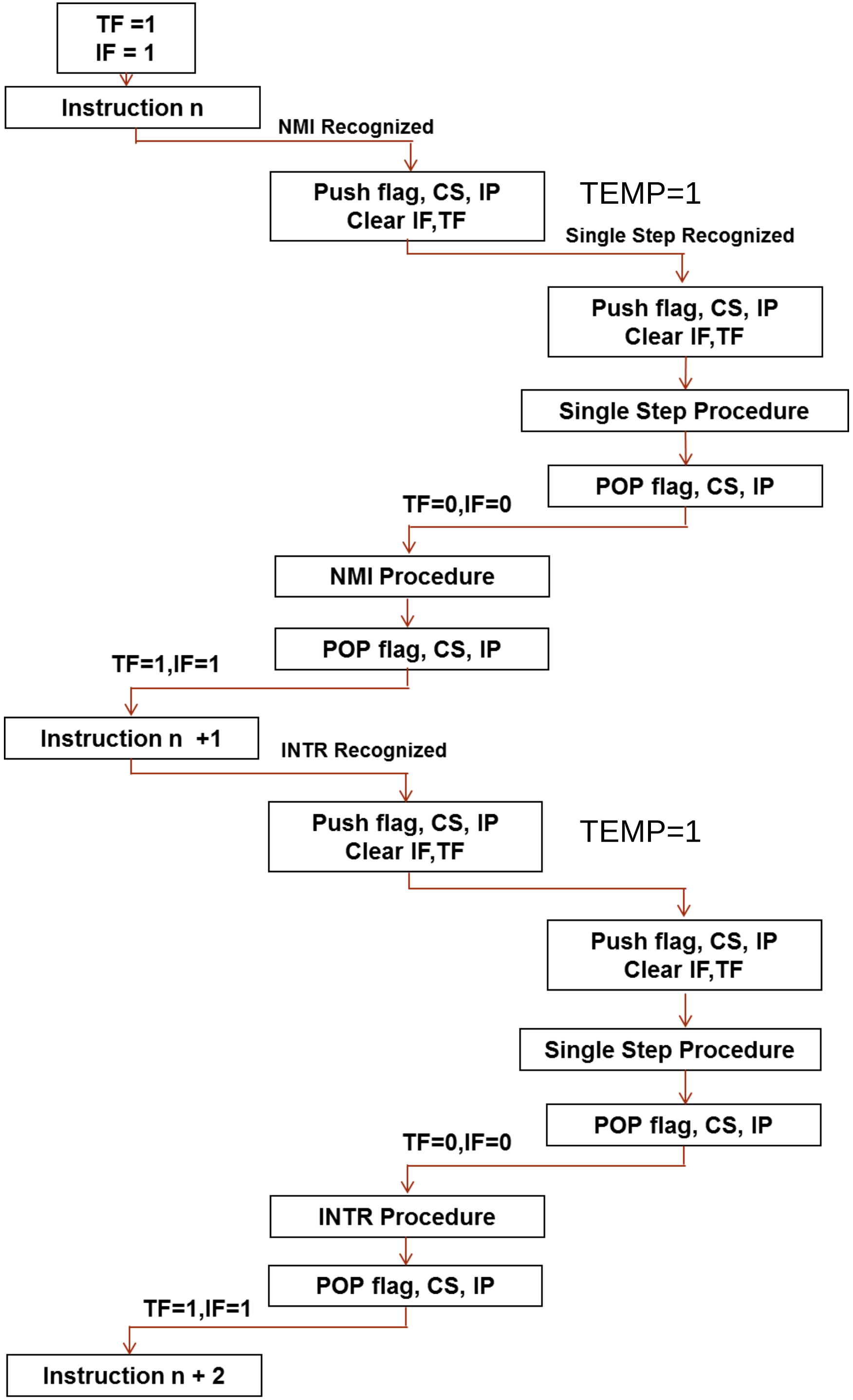
Name	Initated by:	Maskable?	Trigger	Priority	Acknowledge signal?	Vector table address–	Interrupt latency
NMI	External hardware	No	↑ Edge, hold 2 T states min.	2	None	00008H–0000BH	Current instruction + 51 T states
INTR	External hardware	Yes via IF	High level until acknowledged	3	$\overline{\text{INTA}}$	n * 4[b]	Current instruction + 61 T states
INT n	Internal via software	No	None	1	None	n * 4	51 T states
INT 3 (breakpoint)	Internal via software	No	None	1	None	0000CH–0000FH	52 T states
INTO	Internal via software	No	None	1	None	00010H–00013H	53 T states
Divide-by-0	Internal via CPU	Yes via OF	None	1	None	00000H–00003H	51 T states
Single-step	Internal via CPU	Yes via TF	None	4	None	00004H–00007H	51 T states

[a] All interrupt types cause the flags, CS, and IP registers to be pushed onto the stack. In addition, the IF and TF flags are cleared.

[b] n is an 8-bit type number read during the second INTA pulse.

FLOWCAHRT FOR INTERRUPT PROCESSING SEQUENCE





TF =1
IF = 1

Instruction n

NMI Recognized

Push flag, CS, IP
Clear IF,TF

TEMP=1

Single Step Recognized

Push flag, CS, IP
Clear IF,TF

Single Step Procedure

POP flag, CS, IP

TF=0,IF=0

NMI Procedure

POP flag, CS, IP

TF=1,IF=1

Instruction n +1

INTR Recognized

Push flag, CS, IP
Clear IF,TF

TEMP=1

Push flag, CS, IP
Clear IF,TF

Single Step Procedure

POP flag, CS, IP

TF=0,IF=0

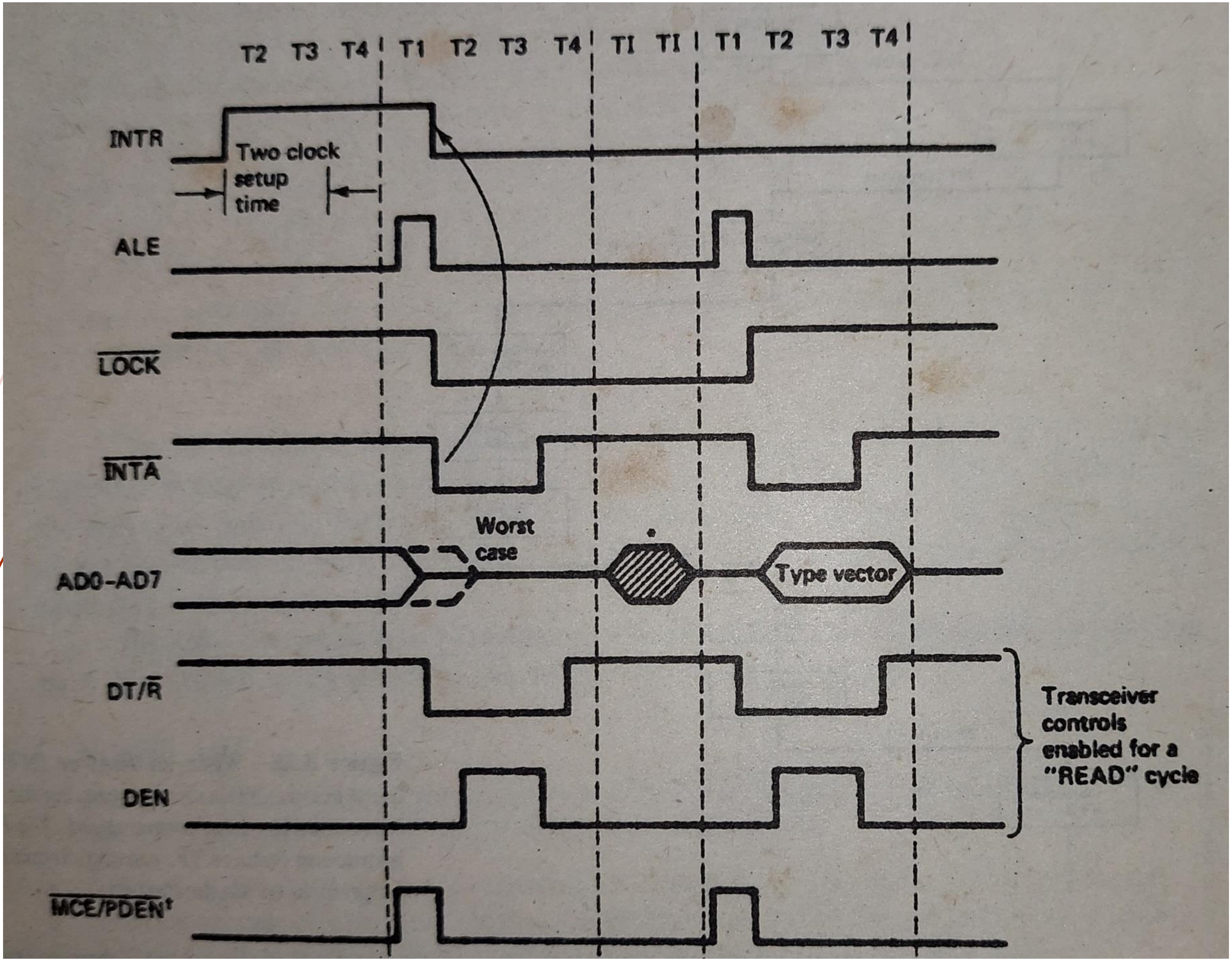
INTR Procedure

POP flag, CS, IP

TF=1,IF=1

Instruction n + 2

Interrupt Acknowledge Bus Cycle


T2
T3
T4
T1
T2
T3
T4
TI
TI
T1
T2
T3
T4
INTR
Two clock setup time
ALE
LOCK
INTA
Worst case
AD0-AD7
Type vector
DT/R
DEN
MCE/PDEN†
Transceiver controls enabled for a "READ" cycle