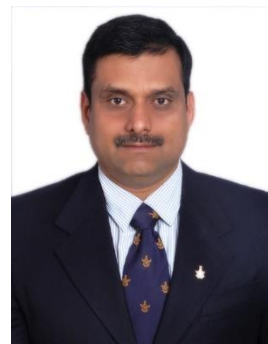




Microprocessors and Interfaces: 2021-22

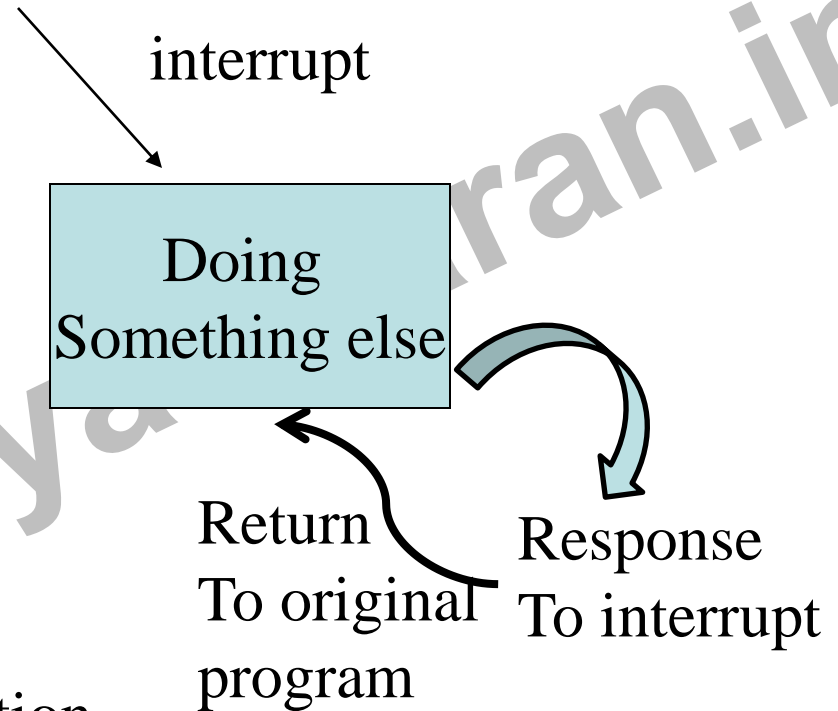
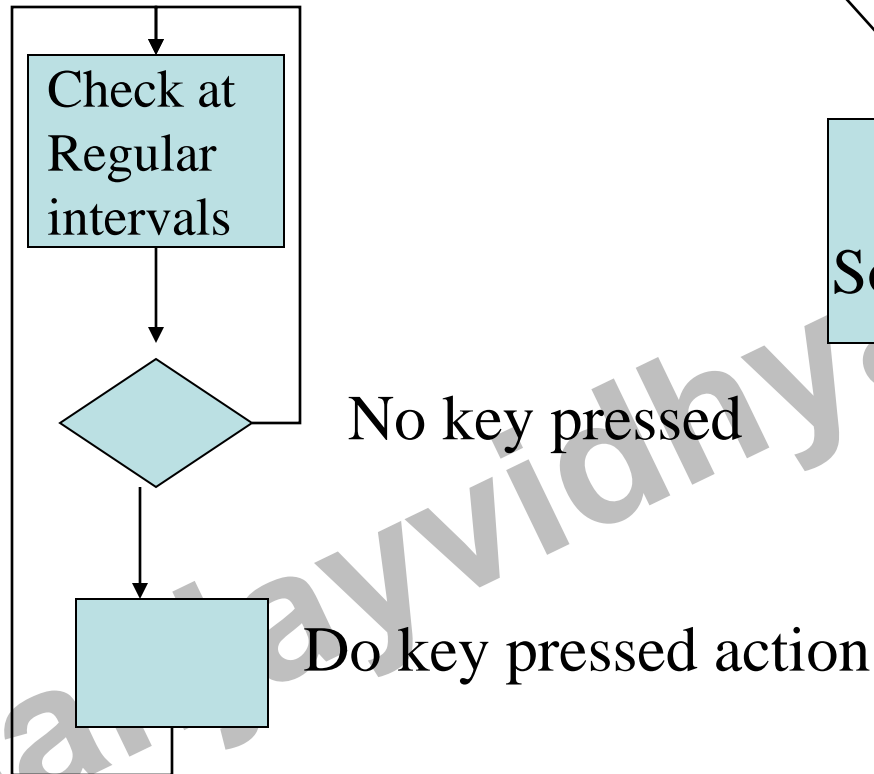
Lecture 21 : Interrupts

By Dr. Sanjay Vidhyadharan

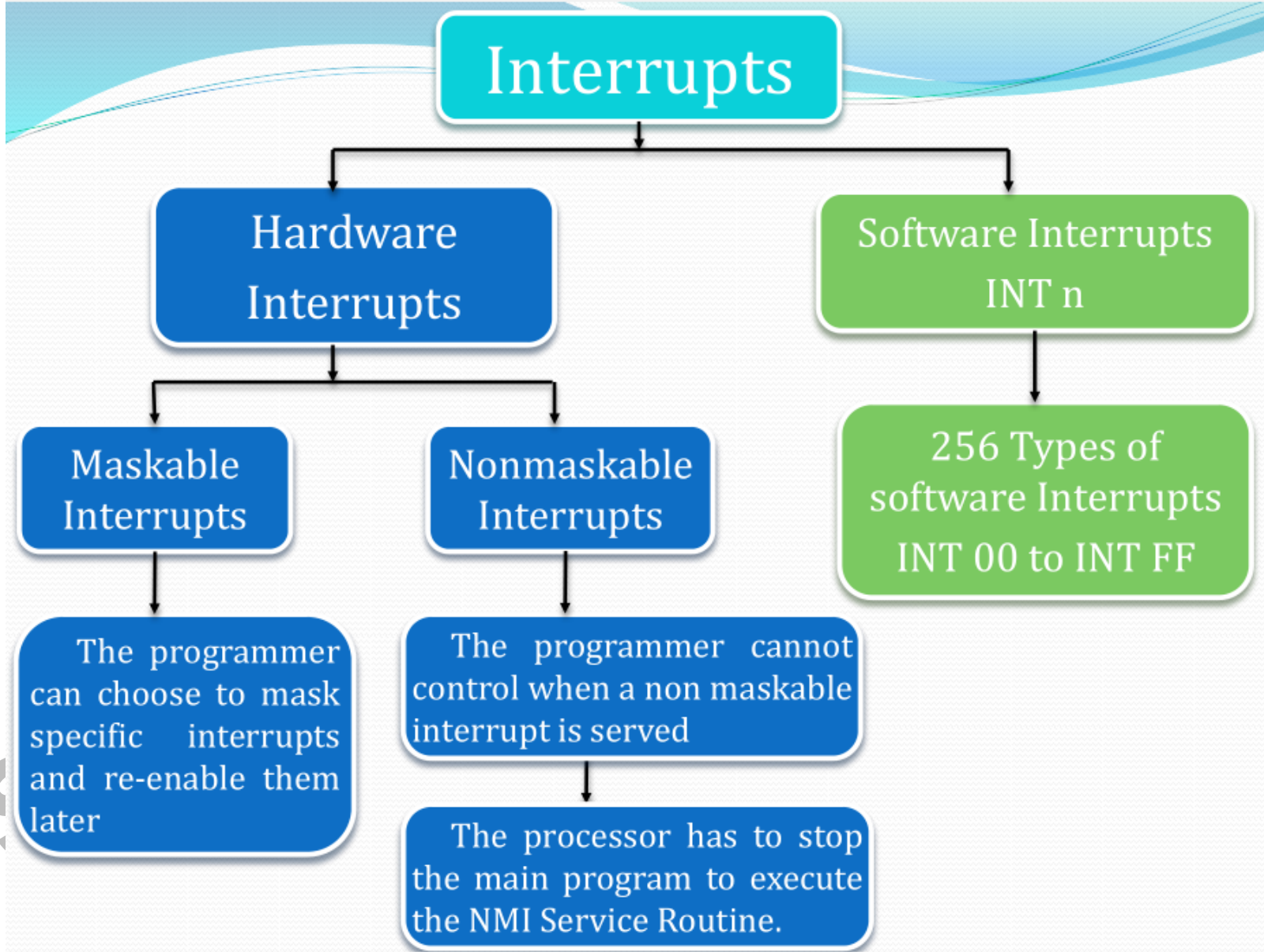


Interrupts vs. Polling

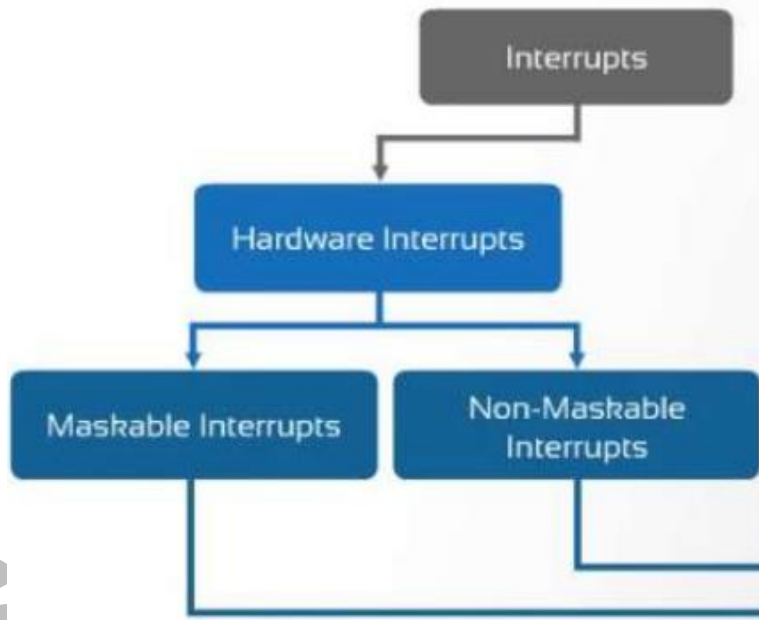
Polling



Types of 8086 Interrupts



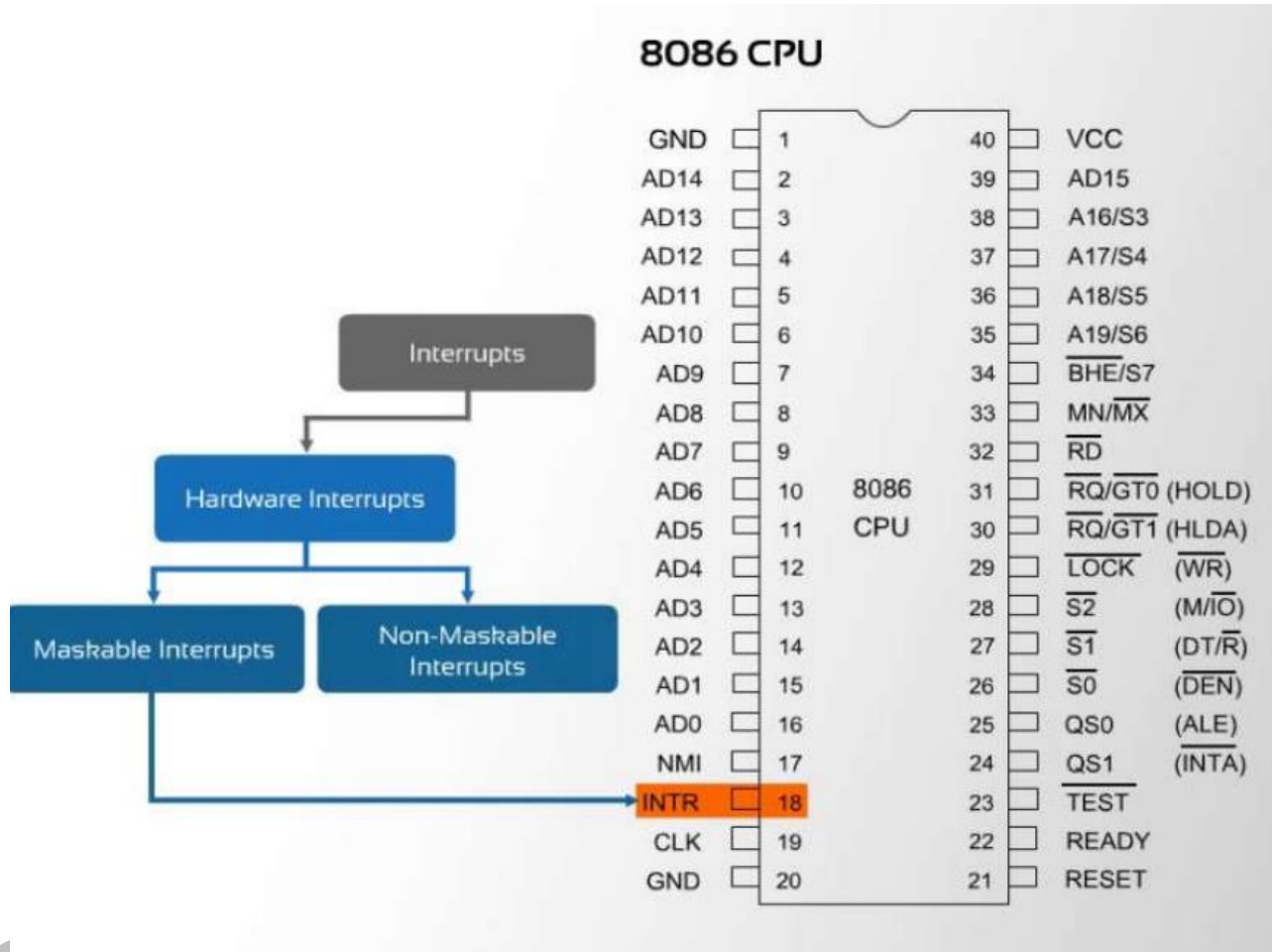
8086 Interrupts



8086 CPU

GND	1	40	VCC
AD14	2	39	AD15
AD13	3	38	A16/S3
AD12	4	37	A17/S4
AD11	5	36	A18/S5
AD10	6	35	A19/S6
AD9	7	34	$\overline{\text{BHE}}/\text{S7}$
AD8	8	33	$\text{MN}/\overline{\text{MX}}$
AD7	9	32	$\overline{\text{RD}}$
AD6	10	31	$\overline{\text{RQ}}/\overline{\text{GT0}}$ (HOLD)
AD5	11	30	$\overline{\text{RQ}}/\overline{\text{GT1}}$ (HLDA)
AD4	12	29	$\overline{\text{LOCK}}$ $\overline{\text{WR}}$
AD3	13	28	$\overline{\text{S2}}$ (M/ $\overline{\text{IO}}$)
AD2	14	27	$\overline{\text{S1}}$ (DT/ $\overline{\text{R}}$)
AD1	15	26	$\overline{\text{S0}}$ ($\overline{\text{DEN}}$)
AD0	16	25	QS0 (ALE)
NMI	17	24	QS1 ($\overline{\text{INTA}}$)
INTR	18	23	TEST
CLK	19	22	READY
GND	20	21	RESET

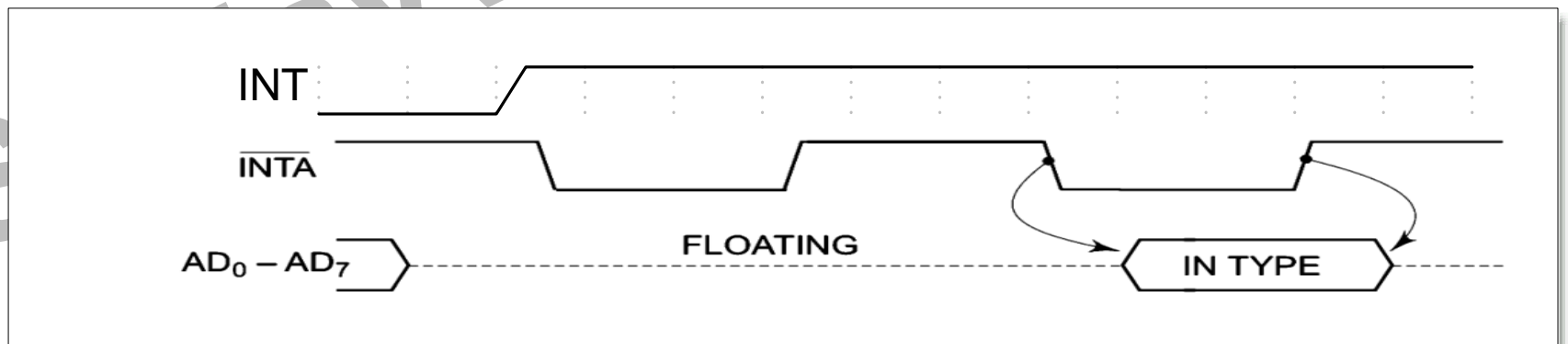
8086 Interrupts



Hardware Interrupts : Interrupt pins and timing

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 that the interrupt request (INTR) is accepted. **(IF Checked)**
 - 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.
 - **INT 02h**



8086 Vector Table

Interrupt Type	Content (16-bit)	Address	Comments	
Type 0	ISR IP	0000:0000	Reserved for divide by Zero interrupt	
	ISR CS	0000:0002		
Type 1	ISR IP	0000:0004	Reserved for single step interrupt	
	ISR CS	0000:0006		
Type 2	ISR IP	0000:0008	Reserved for NMI	
	ISR CS	0000:000A		
Type 3	ISR IP	0000:000C	Reserved for INT single byte instruction	
	ISR CS	0000:000E		
Type 4	ISR IP	0000:0010	Reserved for INTO instruction	
	ISR CS	0000:0012		
Type N		0000:0014	Reserved for two byte instruction INT TYPE	
		0000:0016		
	ISR IP	0000:004N		
	ISR CS	0000:(004N+2)		
		0000:03FC		
	Type FFH	ISR IP		0000:03FE
		ISR CS		0000:03FF

ISR: Interrupt Service Routine

haran.in

INT 0

The screenshot shows the emu8086 emulator interface. The main window displays an assembly program with the following code:

```
01  
02 ; You may custo  
03 ; The locatio  
04  
05 org 100h  
06  
07 Mov BL, 00h  
08 Mov AL, 01h  
09 Div B1  
10 Hlt  
11  
12 ret  
13  
14  
15  
16  
17
```

A message dialog box is open in the foreground, displaying the following text:

message

divide error - overflow.
to manually process this error,
change address of INT 0 in interrupt vector table.

OK

The background window also shows a menu bar with options: file, edit, bookmarks, assembler, emulator, math, ascii codes, help. Below the menu bar, there are buttons for 'new', 'open', and 'examples'. The main window title is 'emu8086 - assembler and microprocessor emulator 4.08'. The background window also shows a menu bar with options: file, math, debug, view, external, virtual devices, virtual dri.

Interrupt Sub-Routine (ISR)

256 Interrupts Of 8086 are Divided in To 3 Groups

1. Type 00 to Type 04 interrupts-

These are used for fixed operations and hence are called dedicated interrupts

2. Type 05 to Type 31 interrupts

Preserved for Higher processors

3. Type 32 to Type 255 interrupts

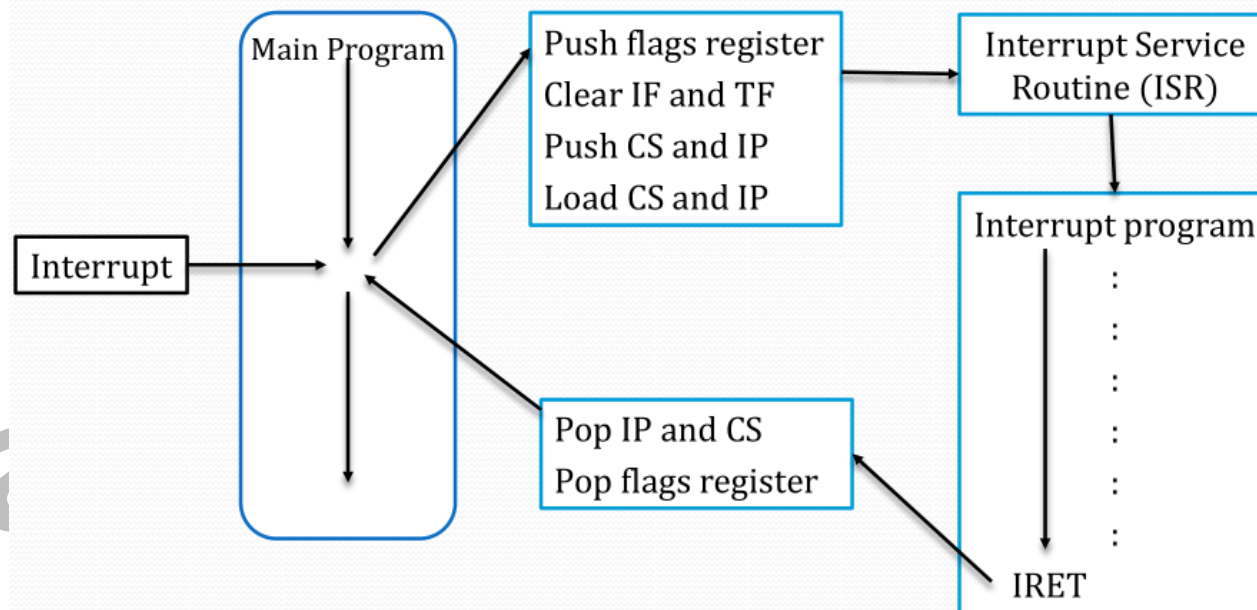
Available for user, called user defined interrupts these can be H/W interrupts and activated through INTR line or can be S/W interrupts.

[E.g. int 21 : 34th interrupt in vector Table or Type 33]

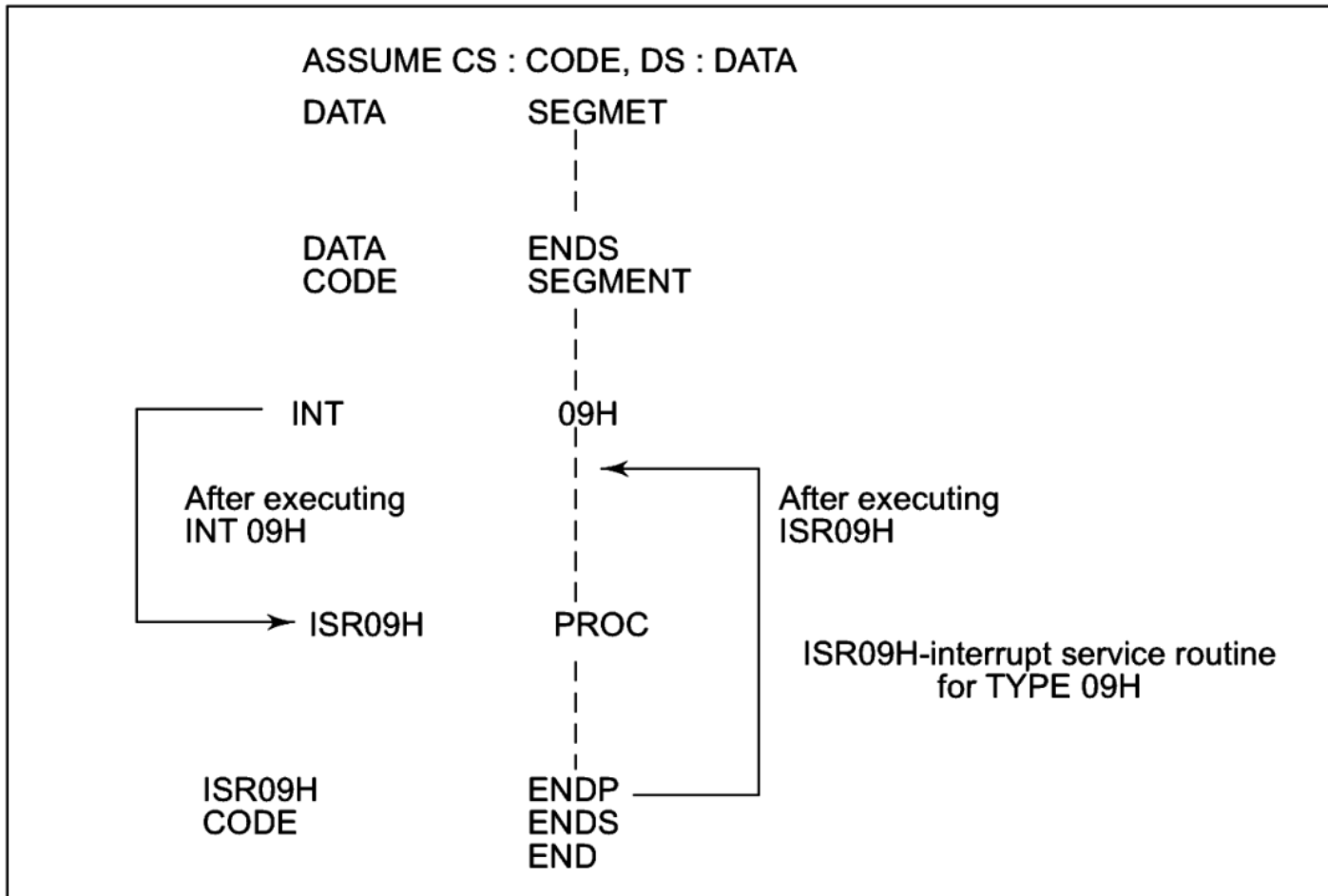
Sanjayvishnyadharan.in

Processing of an Interrupt by the 8086

1. It decrements the stack pointer by 2 and pushes the flag register on the stack.
2. It disables the 8086 INTR interrupt input by clearing the interrupt flag in the flag register.
3. It resets the trap flag in the flag register.
4. It decrements the stack pointer by 2 and pushes the current code segment register contents on the stack.
5. It decrements the stack pointer again by 2 and pushes the current instruction pointer contents on the stack.



Soft Interrupts 8086



S

.in

Bus Cycle

Instruction Cycle

Machine Cycle

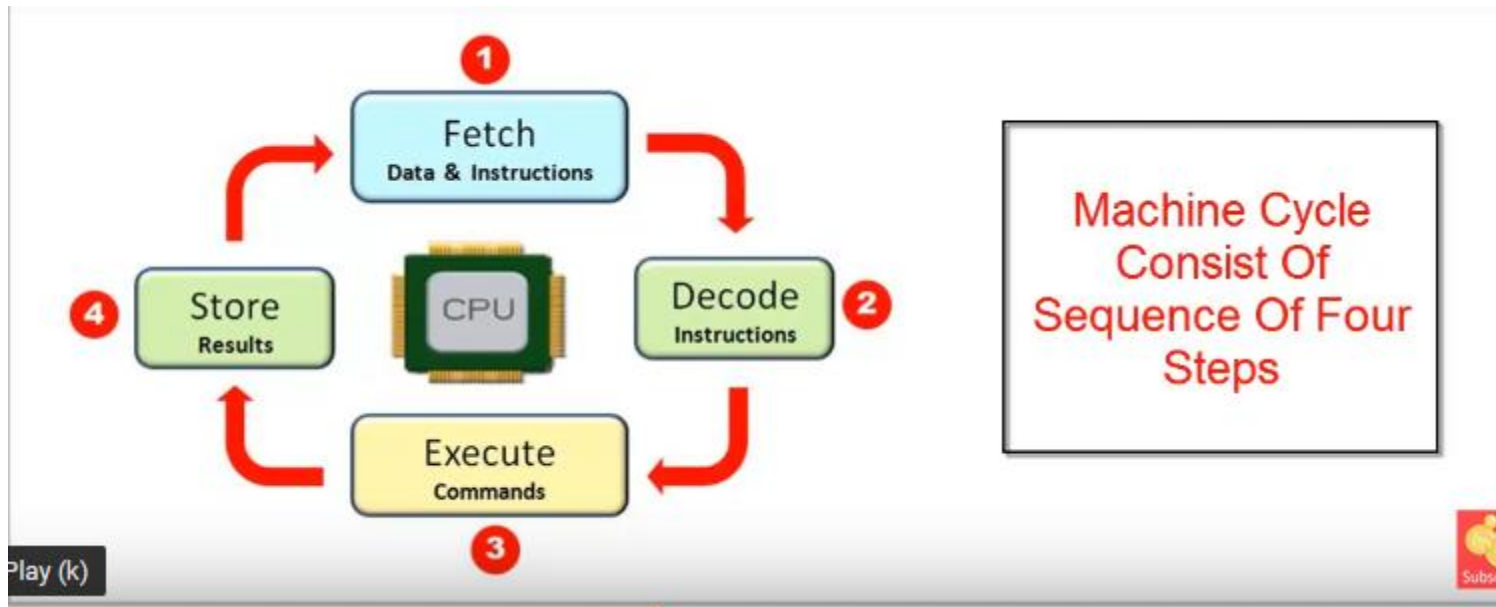
T states

Sanjayvi

an.in

Bus Cycle

Machine Cycle



Sanjay

Lin

Bus Cycle

Instruction Cycle

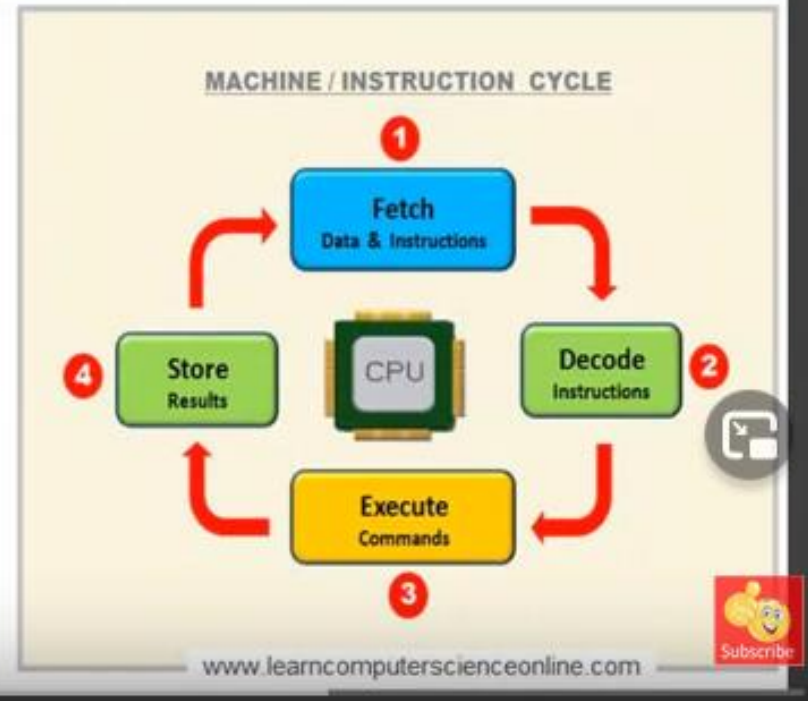
Has more than 1 Machine Cycles

CPU - Instruction Cycle / Machine Cycle .

The CPU's instruction cycle is executed sequentially and each instruction is processed by CPU which consist of following steps :

- Read an Instruction from memory .
- Decode the instruction as per OPCODE
- Find the address of operand .
- Retrieve an operand .
- Perform the desired operation
- Find the address of destination memory .
- Store the result into the destination memory

www.learncomputerscienceonline.com



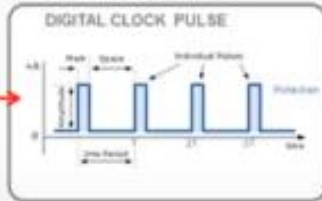
Bus Cycle

T states

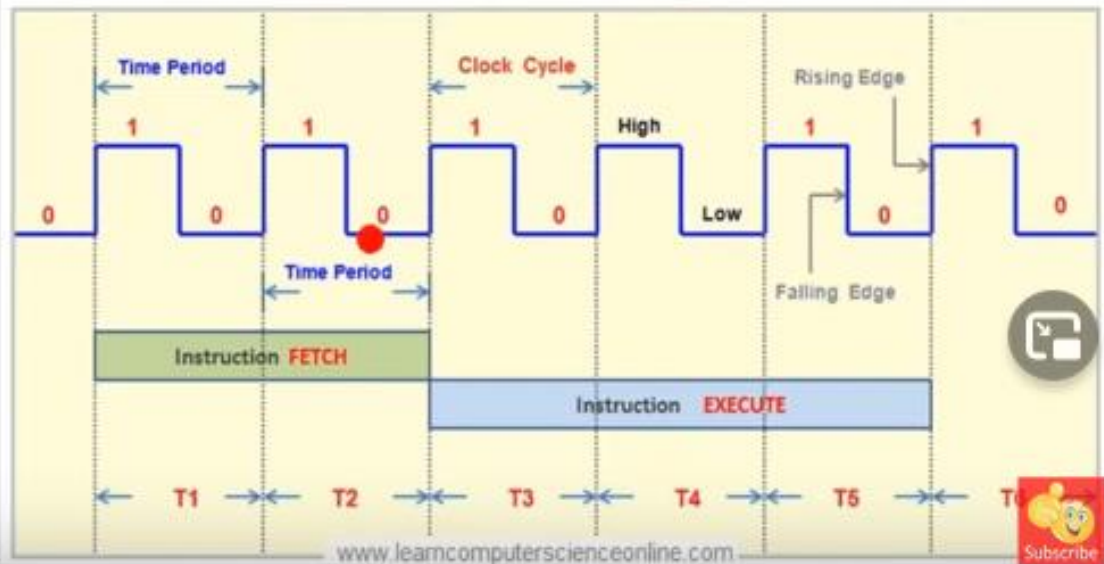
CPU – The Is Driven By Stream Of Clock Pulses

CPU AND DIGITAL CLOCK PULSE

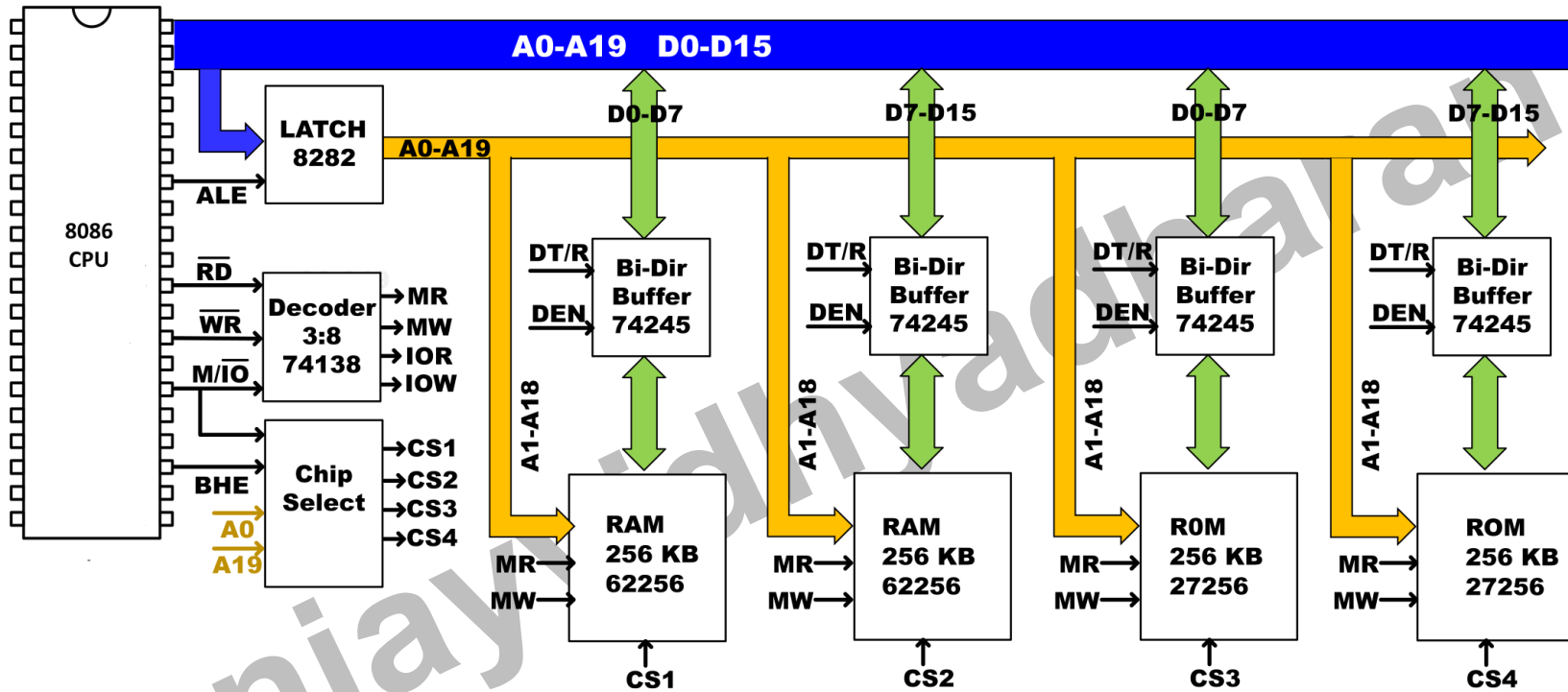
DIGITAL CLOCK PULSES DRIVE CPU FOR EACH PULSE CPU EXECUTES A PART OF THE INSTRUCTION



The System Clock Speed And Instruction Cycle



8086 Memory Interface



Thankyou

Sanjayvidhyadharan.in