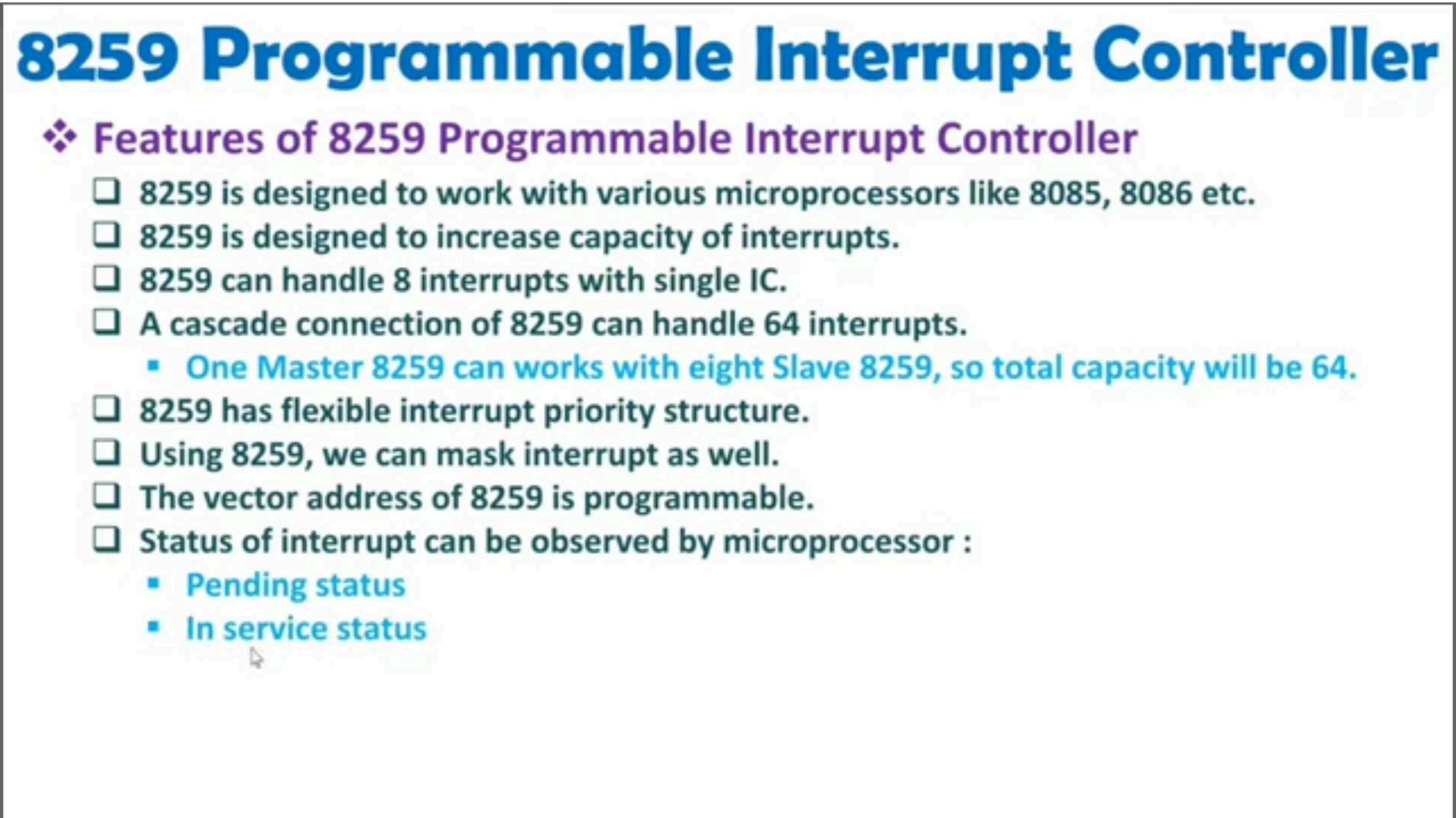
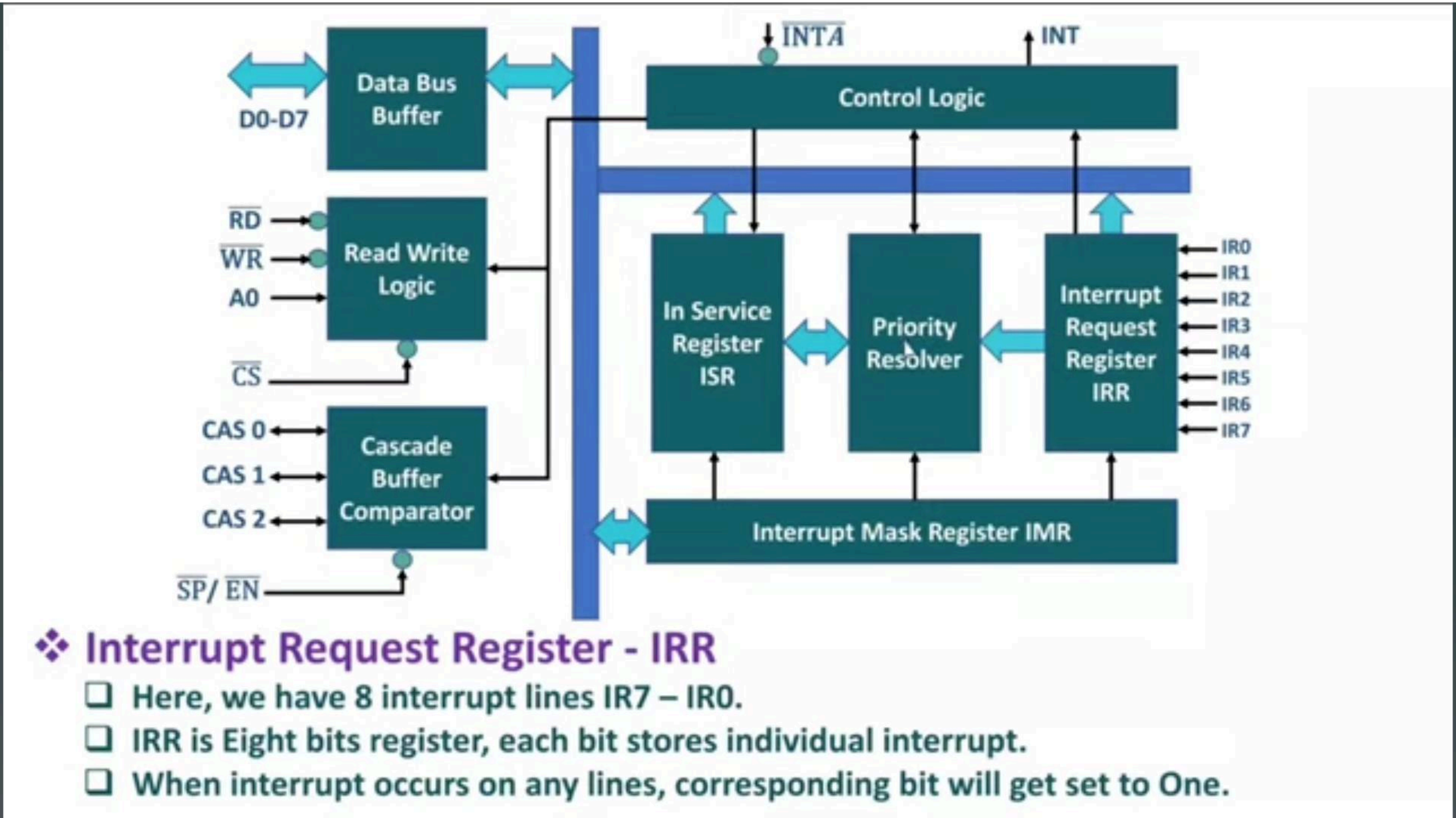


8259 Programmable Interrupt Controller

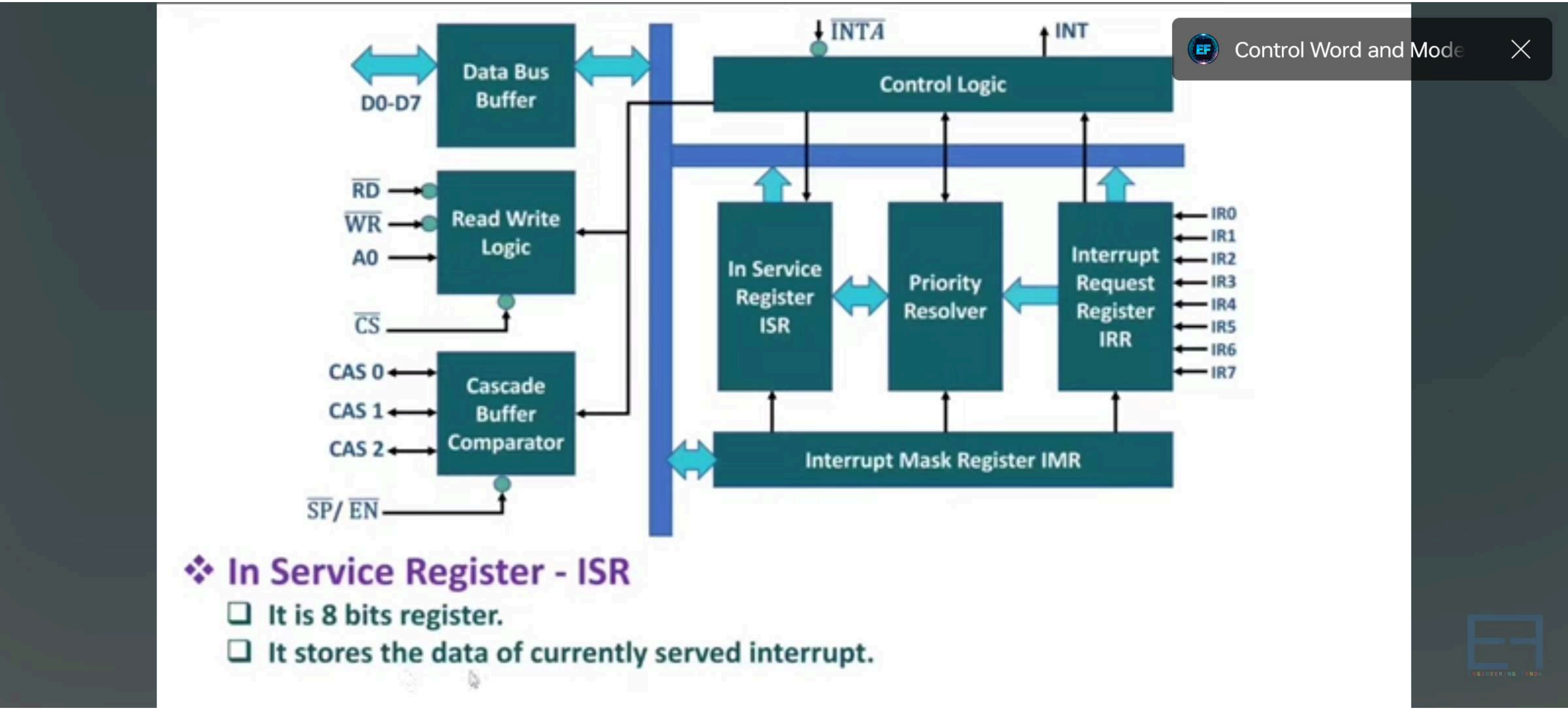
❖ Features of 8259 Programmable Interrupt Controller

- ❑ 8259 is designed to work with various microprocessors like 8085, 8086 etc.
- ❑ 8259 is designed to increase capacity of interrupts.
- ❑ 8259 can handle 8 interrupts with single IC.
- ❑ A cascade connection of 8259 can handle 64 interrupts.
  - One Master 8259 can works with eight Slave 8259, so total capacity will be 64.
- ❑ 8259 has flexible interrupt priority structure.
- ❑ Using 8259, we can mask interrupt as well.
- ❑ The vector address of 8259 is programmable.
- ❑ Status of interrupt can be observed by microprocessor :
  - Pending status
  - In service status



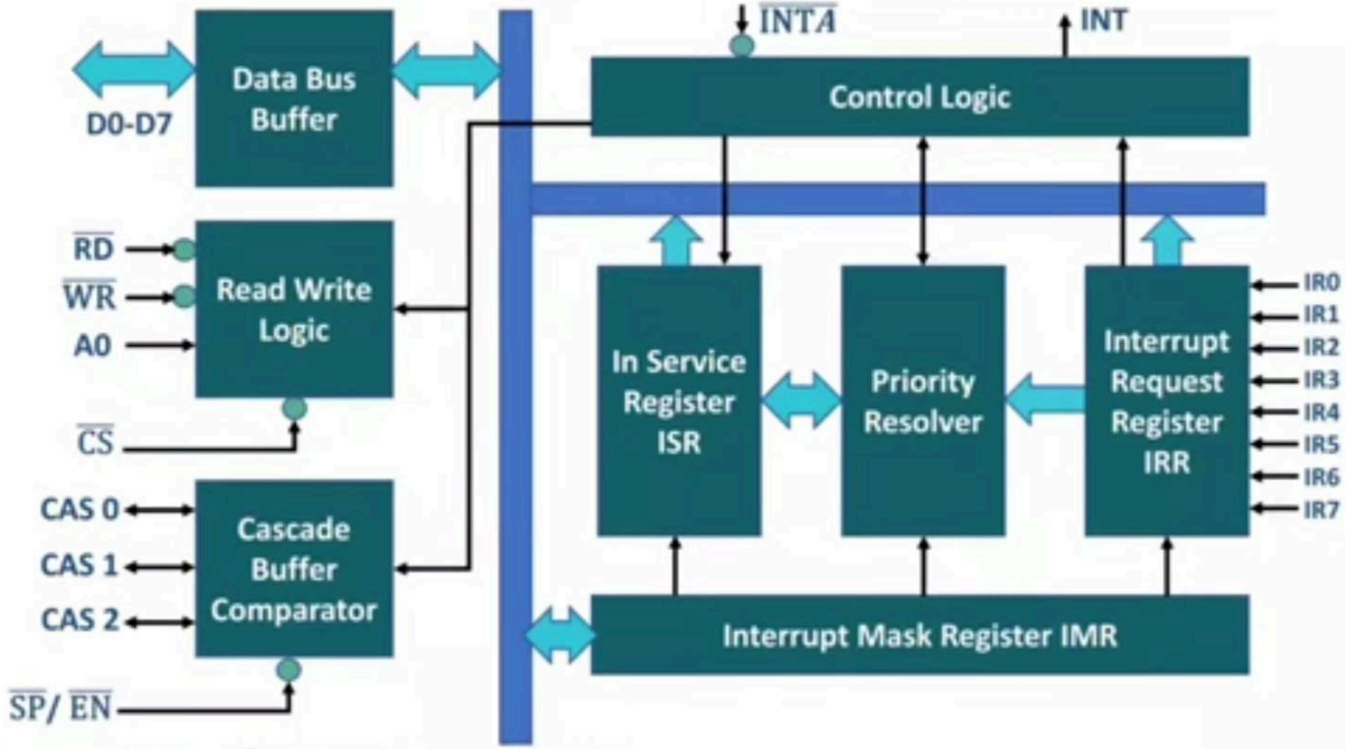
❖ Interrupt Request Register - IRR

- ❑ Here, we have 8 interrupt lines IR7 – IR0.
- ❑ IRR is Eight bits register, each bit stores individual interrupt.
- ❑ When interrupt occurs on any lines, corresponding bit will get set to One.



❖ In Service Register - ISR

- ❑ It is 8 bits register.
- ❑ It stores the data of currently served interrupt.

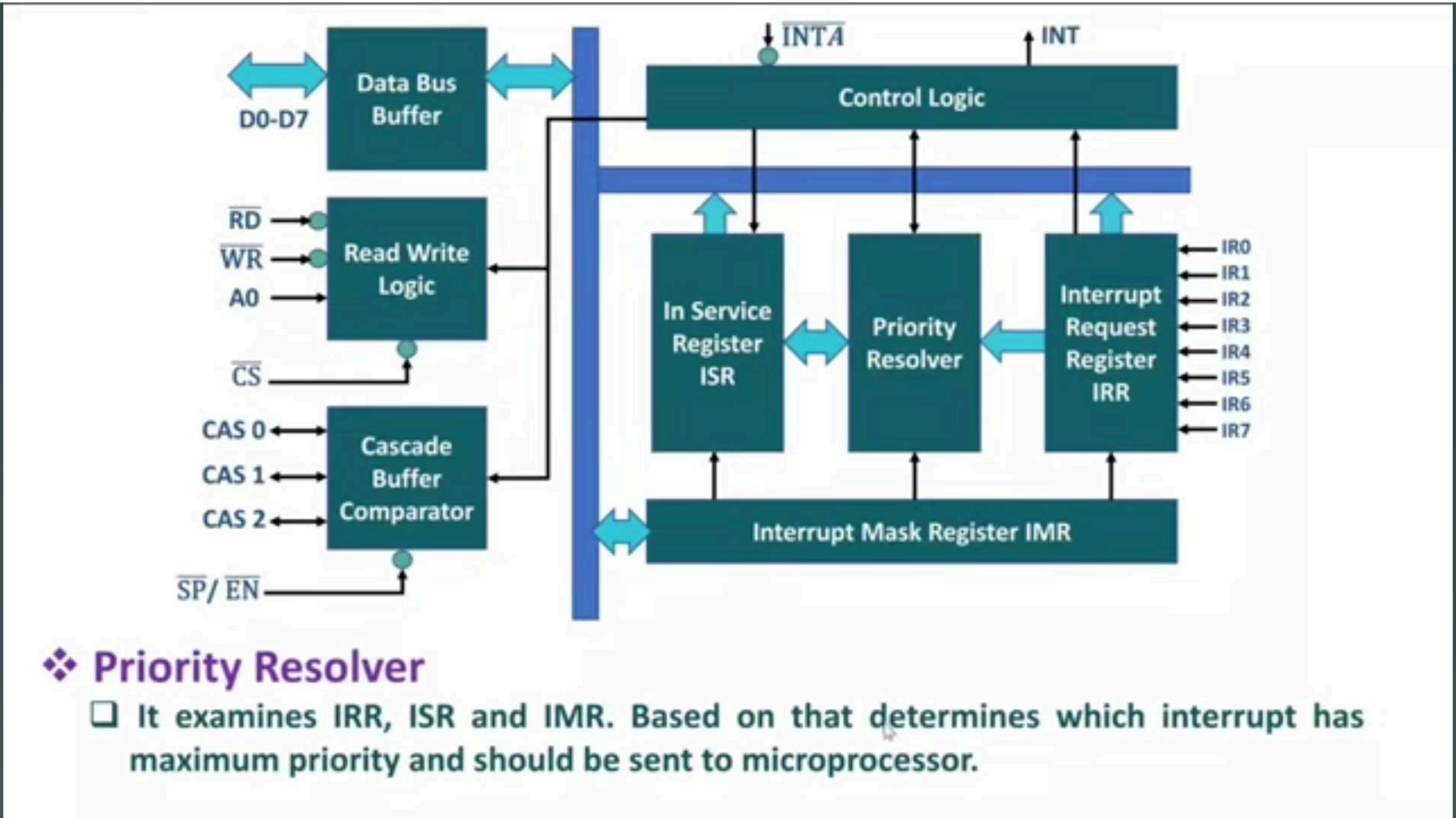


❖ Interrupt Mask Register - IMR

❑ It is 8 bits register.

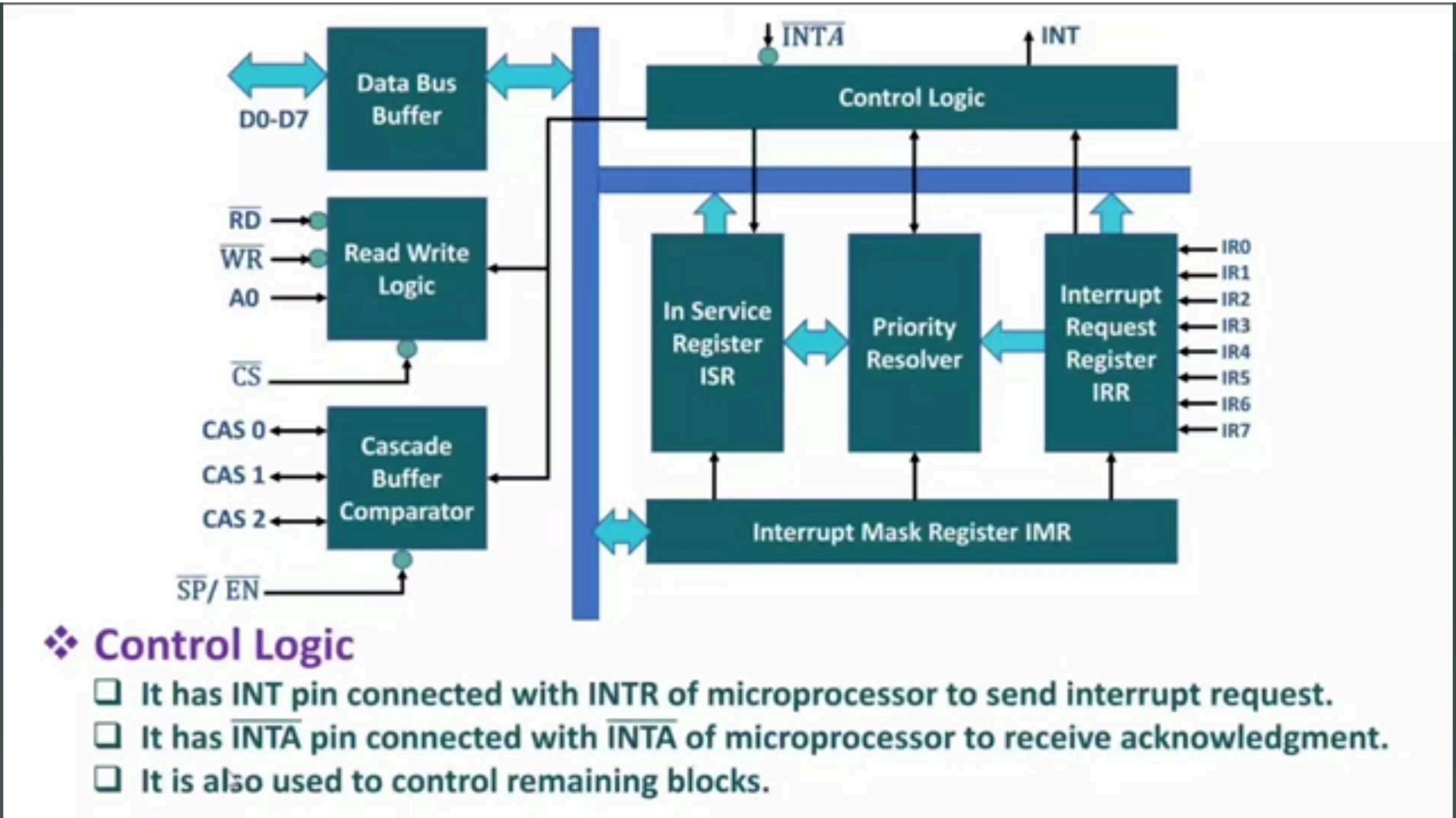
❑ It stores the masking pattern of 8259.

❑ Each bits holds masking of individual interrupt.



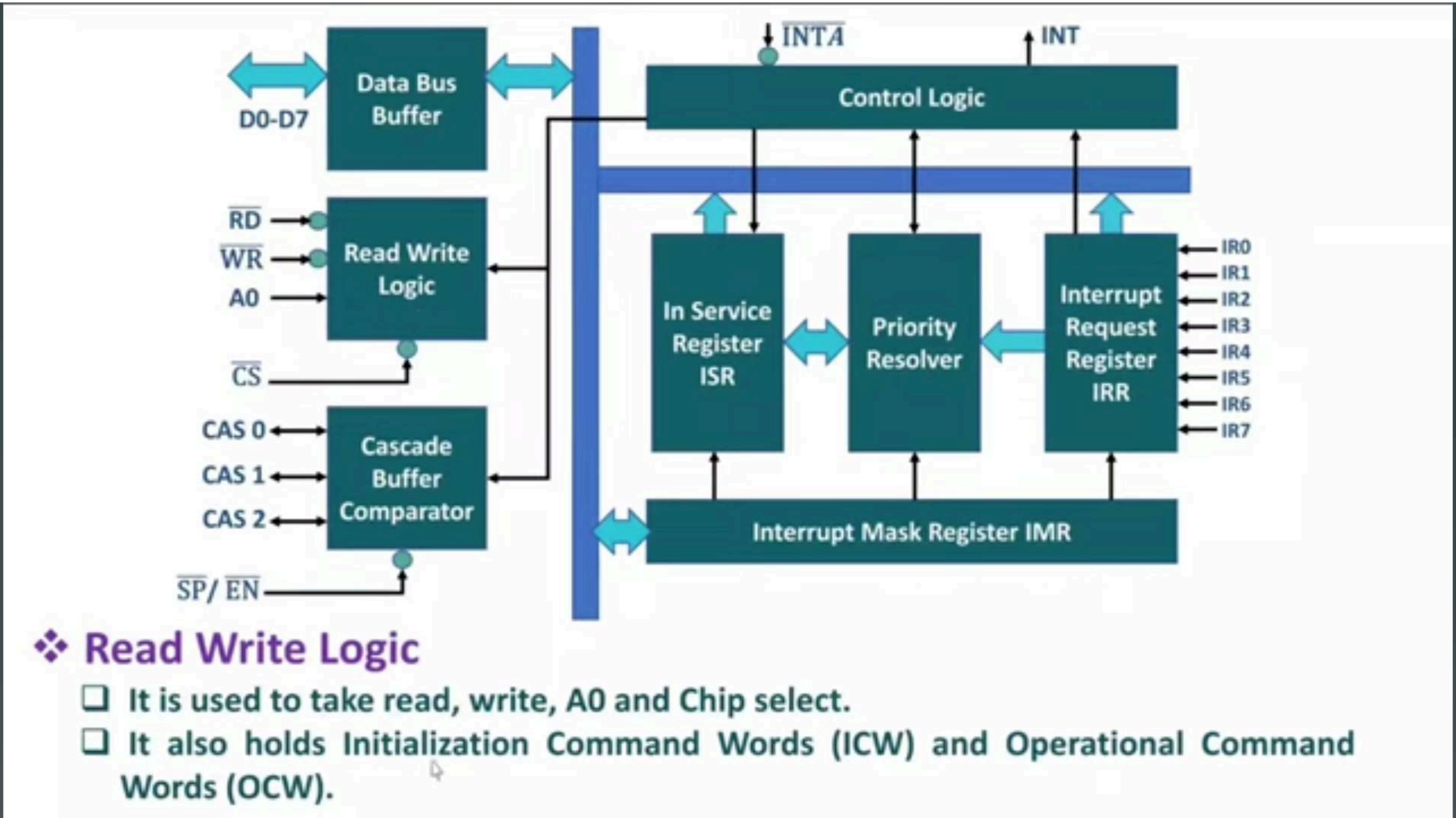
❖ Priority Resolver

- ❑ It examines IRR, ISR and IMR. Based on that determines which interrupt has maximum priority and should be sent to microprocessor.



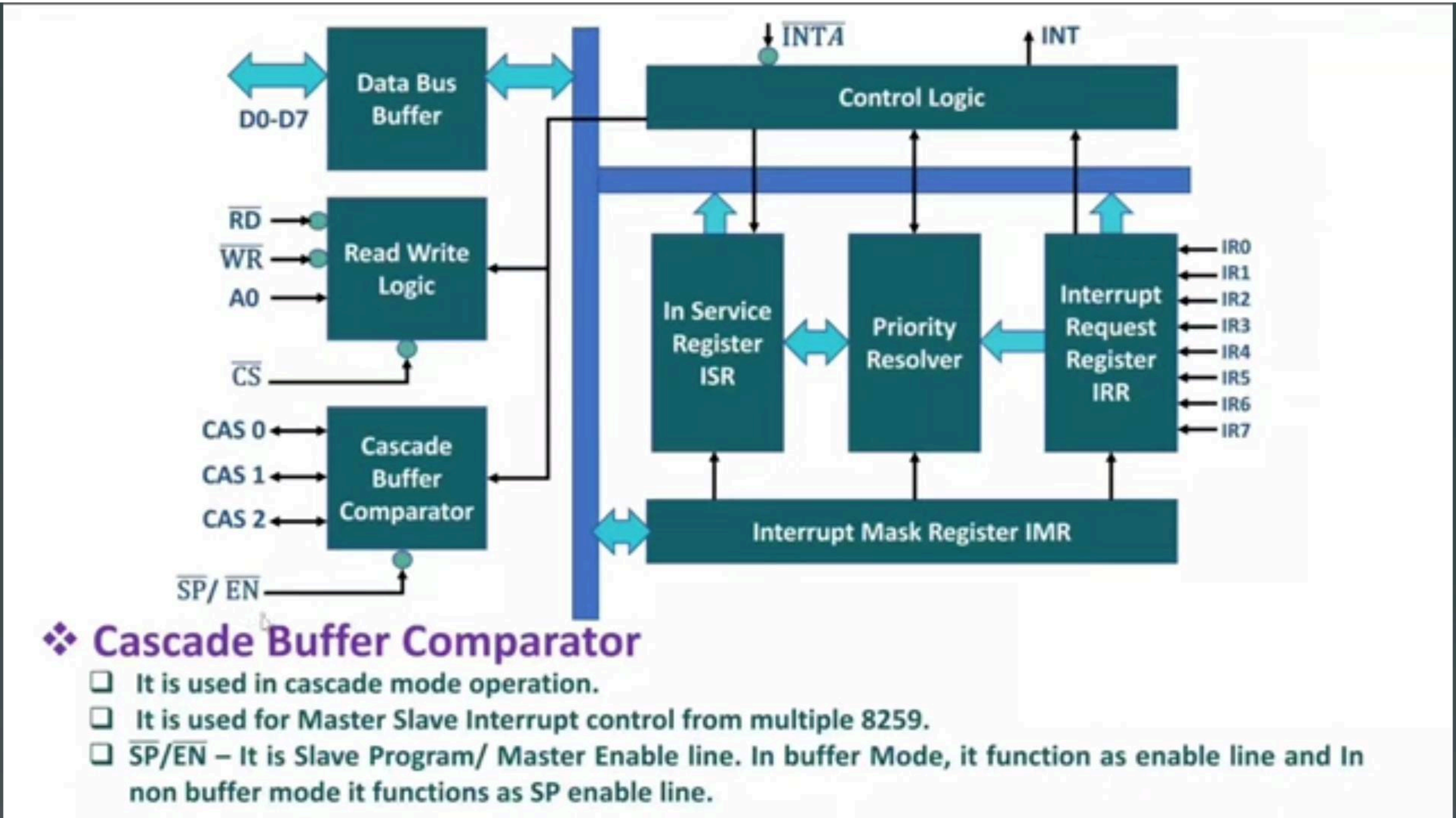
❖ Control Logic

- ❑ It has INT pin connected with INTR of microprocessor to send interrupt request.
- ❑ It has $\overline{INTA}$ pin connected with $\overline{INTA}$ of microprocessor to receive acknowledgment.
- ❑ It is also used to control remaining blocks.



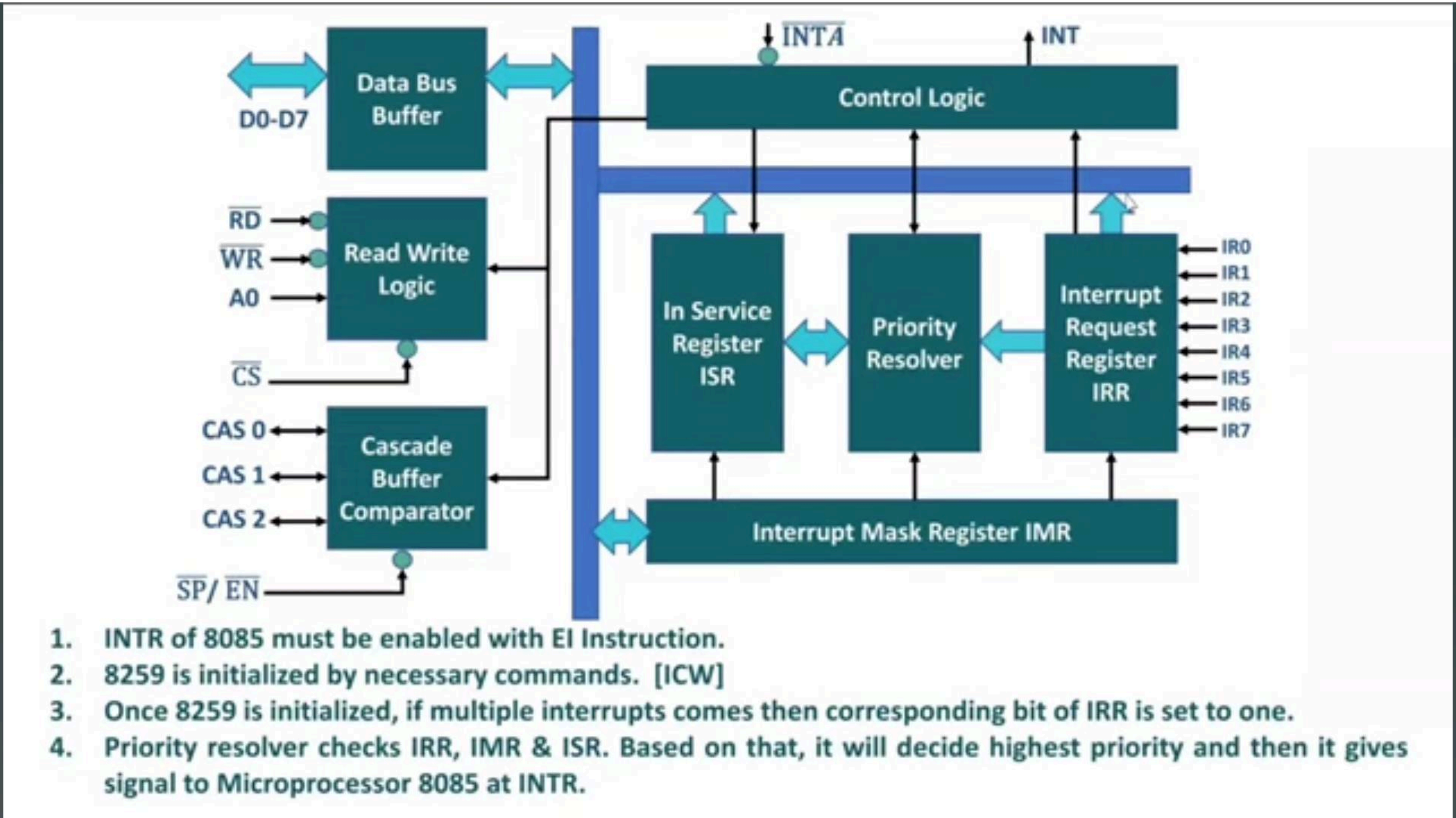
❖ Read Write Logic

- ❑ It is used to take read, write, A0 and Chip select.
- ❑ It also holds Initialization Command Words (ICW) and Operational Command Words (OCW).

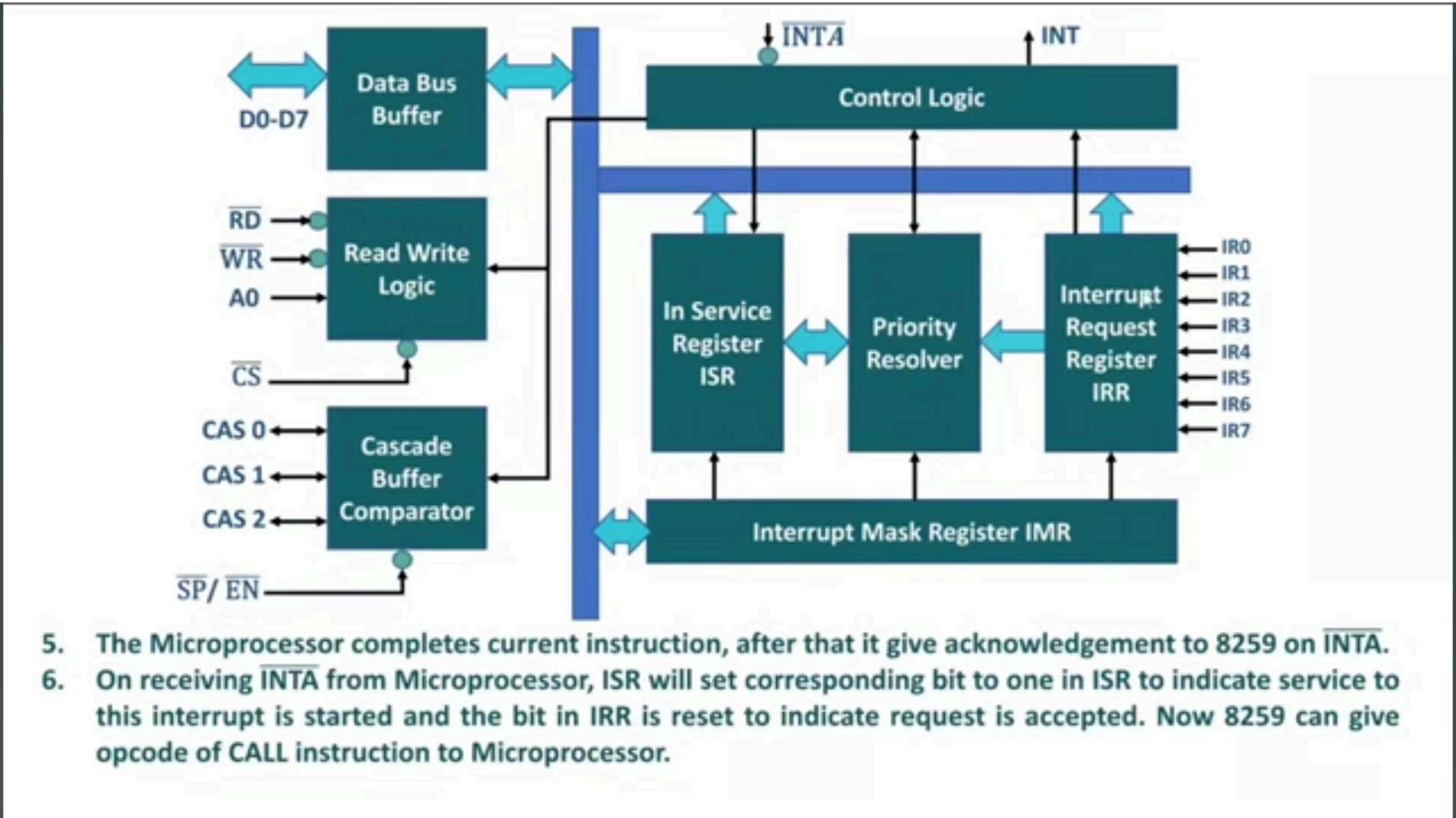


❖ Cascade Buffer Comparator

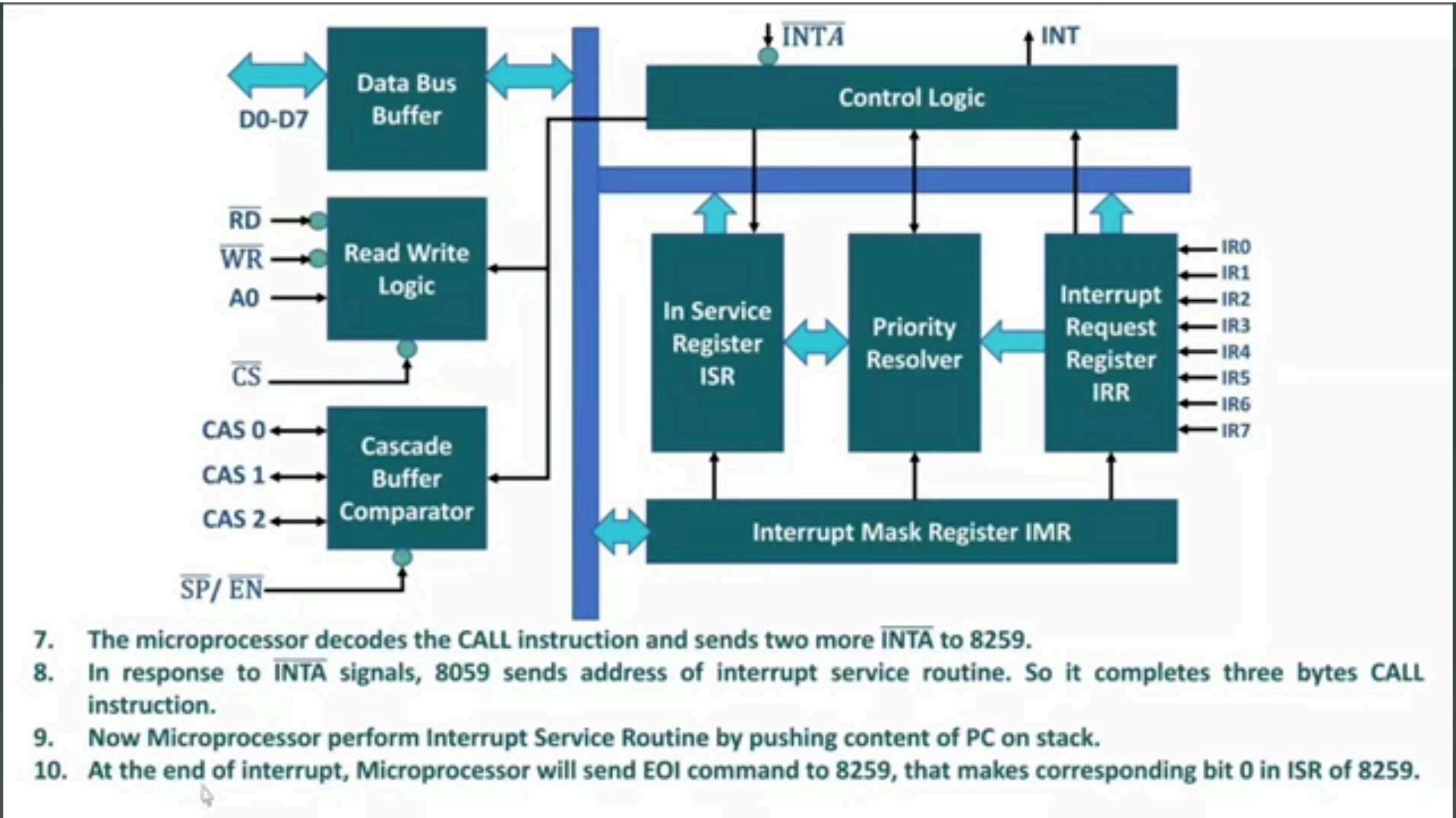
- ❑ It is used in cascade mode operation.
- ❑ It is used for Master Slave Interrupt control from multiple 8259.
- ❑ $\overline{SP}/\overline{EN}$ – It is Slave Program/ Master Enable line. In buffer Mode, it function as enable line and In non buffer mode it functions as SP enable line.



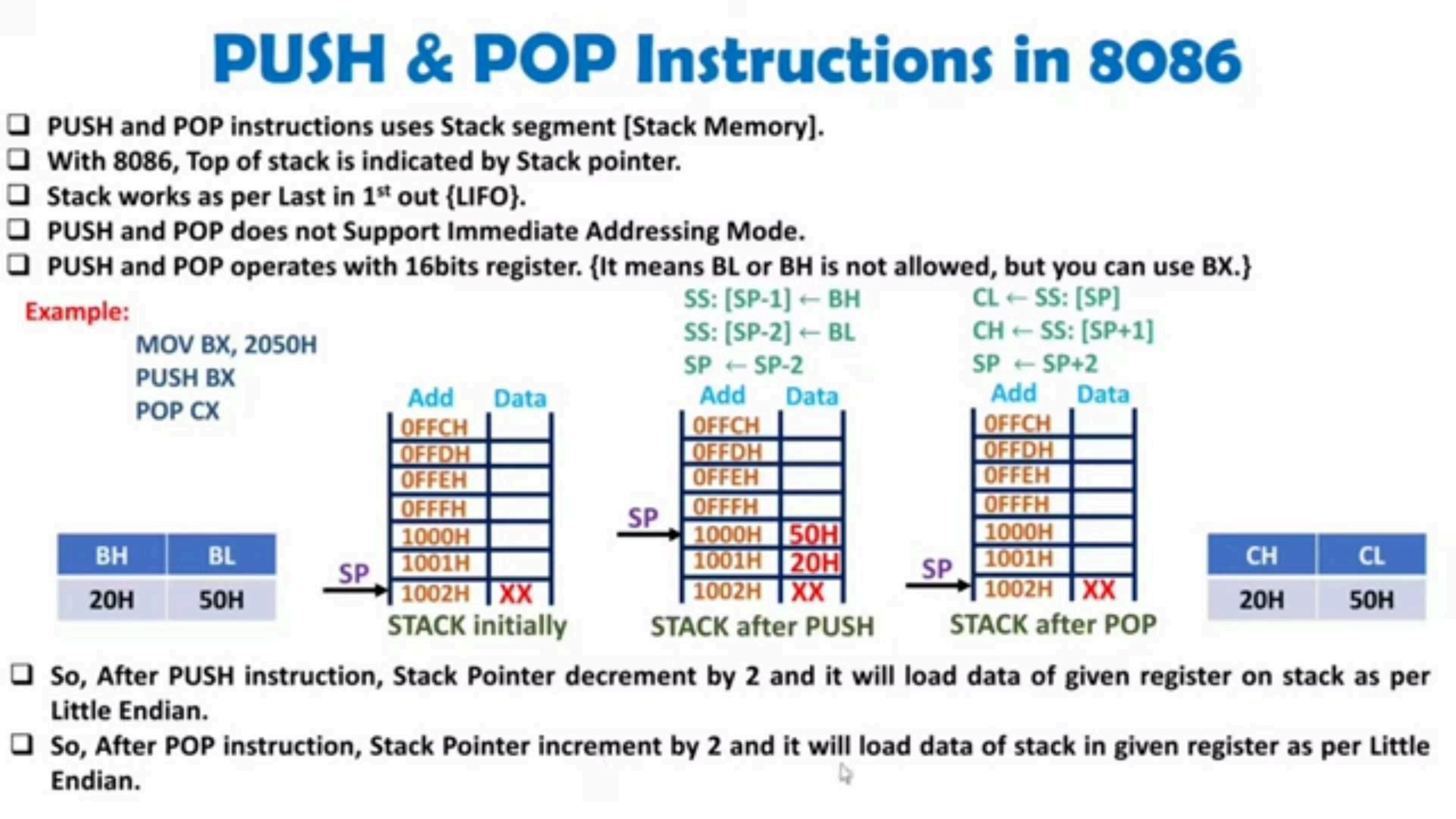
1. INTR of 8085 must be enabled with EI Instruction.
2. 8259 is initialized by necessary commands. [ICW]
3. Once 8259 is initialized, if multiple interrupts comes then corresponding bit of IRR is set to one.
4. Priority resolver checks IRR, IMR & ISR. Based on that, it will decide highest priority and then it gives signal to Microprocessor 8085 at INTR.



5. The Microprocessor completes current instruction, after that it give acknowledgement to 8259 on $\overline{INTA}$.
6. On receiving $\overline{INTA}$ from Microprocessor, ISR will set corresponding bit to one in ISR to indicate service to this interrupt is started and the bit in IRR is reset to indicate request is accepted. Now 8259 can give opcode of CALL instruction to Microprocessor.



7. The microprocessor decodes the CALL instruction and sends two more $\overline{\text{INTA}}$ to 8259.
8. In response to $\overline{\text{INTA}}$ signals, 8059 sends address of interrupt service routine. So it completes three bytes CALL instruction.
9. Now Microprocessor perform Interrupt Service Routine by pushing content of PC on stack.
10. At the end of interrupt, Microprocessor will send EOI command to 8259, that makes corresponding bit 0 in ISR of 8259.



PUSH & POP Instructions in 8086

- ❑ PUSH and POP instructions uses Stack segment [Stack Memory].
- ❑ With 8086, Top of stack is indicated by Stack pointer.
- ❑ Stack works as per Last in 1st out {LIFO}.
- ❑ PUSH and POP does not Support Immediate Addressing Mode.
- ❑ PUSH and POP operates with 16bits register. {It means BL or BH is not allowed, but you can use BX.}

Example:

```
MOV BX, 2050H
PUSH BX
POP CX
```

BH	BL
20H	50H

STACK initially

	Add	Data
	0FFCH	
	0FFDH	
	0FFEH	
	0FFFH	
	1000H	
	1001H	
SP →	1002H	XX

SS: [SP-1] ← BH

SS: [SP-2] ← BL

SP ← SP-2

STACK after PUSH

	Add	Data
	0FFCH	
	0FFDH	
	0FFEH	
	0FFFH	
SP →	1000H	50H
	1001H	20H
	1002H	XX

CL ← SS: [SP]

CH ← SS: [SP+1]

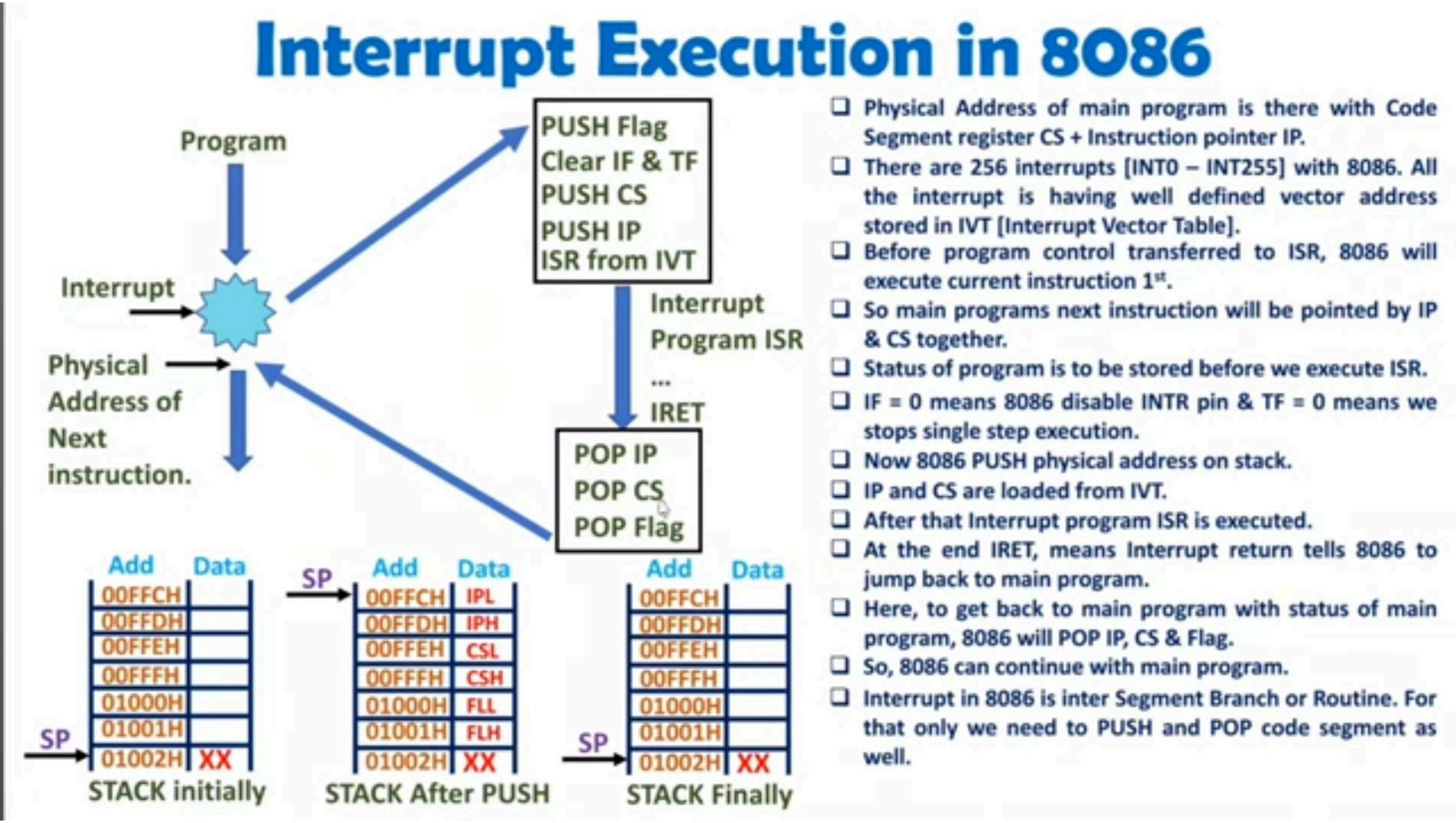
SP ← SP+2

STACK after POP

	Add	Data
	0FFCH	
	0FFDH	
	0FFEH	
	0FFFH	
	1000H	
	1001H	
SP →	1002H	XX

CH	CL
20H	50H

- ❑ So, After PUSH instruction, Stack Pointer decrement by 2 and it will load data of given register on stack as per Little Endian.
- ❑ So, After POP instruction, Stack Pointer increment by 2 and it will load data of stack in given register as per Little Endian.



Interrupt Execution in 8086

STACK initially

	Add	Data
	00FFCH	
	00FFDH	
	00FFEH	
	00FFFH	
	01000H	
	01001H	
SP →	01002H	XX

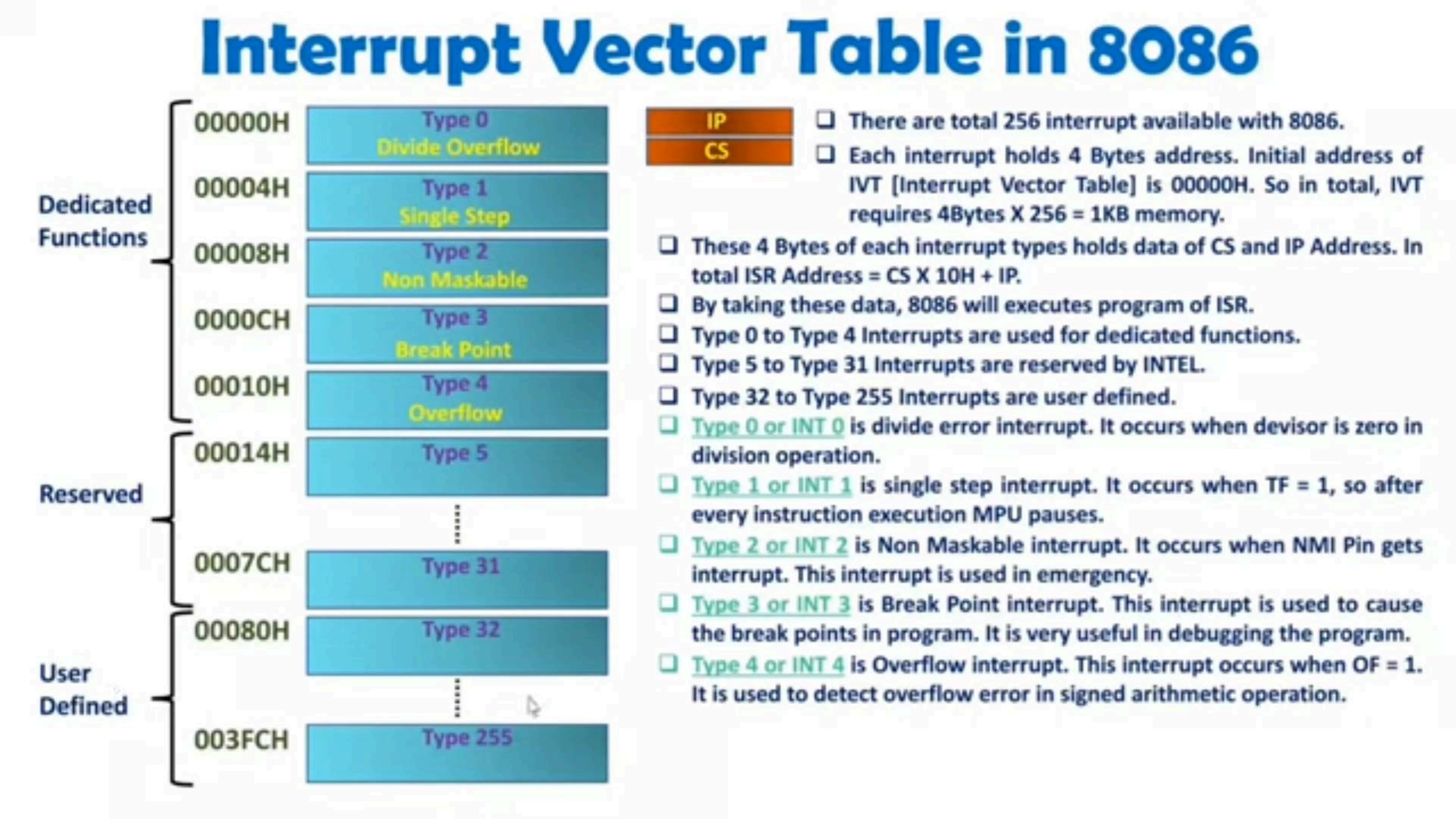
STACK After PUSH

	Add	Data
SP →	00FFCH	IPL
	00FFDH	IPH
	00FFEH	CSL
	00FFFH	CSH
	01000H	FLL
	01001H	FLH
	01002H	XX

STACK Finally

	Add	Data
	00FFCH	
	00FFDH	
	00FFEH	
	00FFFH	
	01000H	
	01001H	
SP →	01002H	XX

- ❑ Physical Address of main program is there with Code Segment register CS + Instruction pointer IP.
- ❑ There are 256 interrupts [INT0 – INT255] with 8086. All the interrupt is having well defined vector address stored in IVT [Interrupt Vector Table].
- ❑ Before program control transferred to ISR, 8086 will execute current instruction 1st.
- ❑ So main programs next instruction will be pointed by IP & CS together.
- ❑ Status of program is to be stored before we execute ISR.
- ❑ IF = 0 means 8086 disable INTR pin & TF = 0 means we stops single step execution.
- ❑ Now 8086 PUSH physical address on stack.
- ❑ IP and CS are loaded from IVT.
- ❑ After that Interrupt program ISR is executed.
- ❑ At the end IRET, means Interrupt return tells 8086 to jump back to main program.
- ❑ Here, to get back to main program with status of main program, 8086 will POP IP, CS & Flag.
- ❑ So, 8086 can continue with main program.
- ❑ Interrupt in 8086 is inter Segment Branch or Routine. For that only we need to PUSH and POP code segment as well.



Interrupt Vector Table in 8086

Dedicated Functions

00000H Type 0 Divide Overflow

00004H Type 1 Single Step

00008H Type 2 Non Maskable

0000CH Type 3 Break Point

00010H Type 4 Overflow

Reserved

00014H Type 5

0007CH Type 31

User Defined

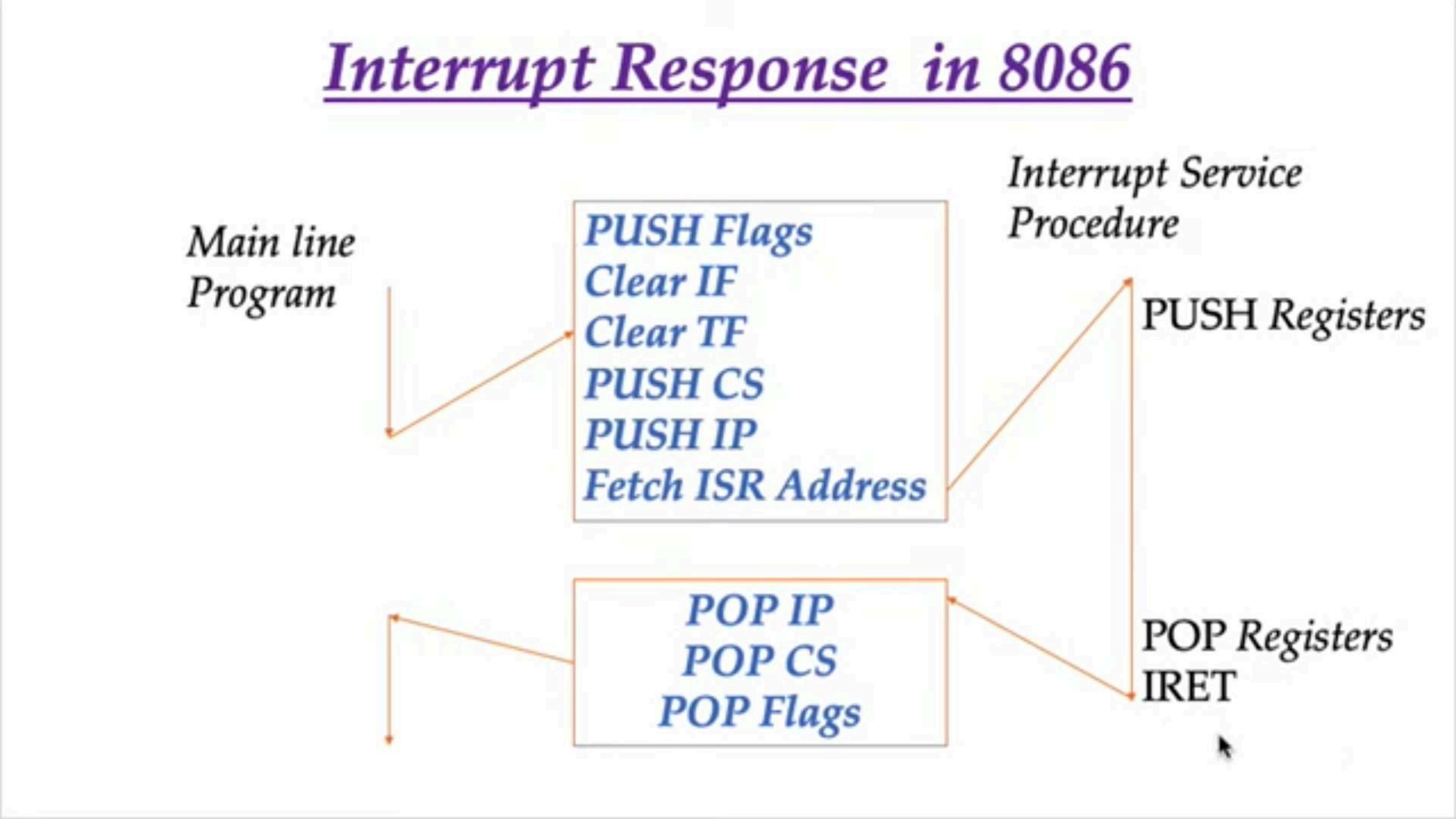
00080H Type 32

003FCH Type 255

IP

CS

- ❑ There are total 256 interrupt available with 8086.
- ❑ Each interrupt holds 4 Bytes address. Initial address of IVT [Interrupt Vector Table] is 00000H. So in total, IVT requires 4Bytes X 256 = 1KB memory.
- ❑ These 4 Bytes of each interrupt types holds data of CS and IP Address. In total ISR Address = CS X 10H + IP.
- ❑ By taking these data, 8086 will executes program of ISR.
- ❑ Type 0 to Type 4 Interrupts are used for dedicated functions.
- ❑ Type 5 to Type 31 Interrupts are reserved by INTEL.
- ❑ Type 32 to Type 255 Interrupts are user defined.
- ❑ Type 0 or INT 0 is divide error interrupt. It occurs when devisor is zero in division operation.
- ❑ Type 1 or INT 1 is single step interrupt. It occurs when TF = 1, so after every instruction execution MPU pauses.
- ❑ Type 2 or INT 2 is Non Maskable interrupt. It occurs when NMI Pin gets interrupt. This interrupt is used in emergency.
- ❑ Type 3 or INT 3 is Break Point interrupt. This interrupt is used to cause the break points in program. It is very useful in debugging the program.
- ❑ Type 4 or INT 4 is Overflow interrupt. This interrupt occurs when OF = 1. It is used to detect overflow error in signed arithmetic operation.



Interrupt Response in 8086

Main line Program

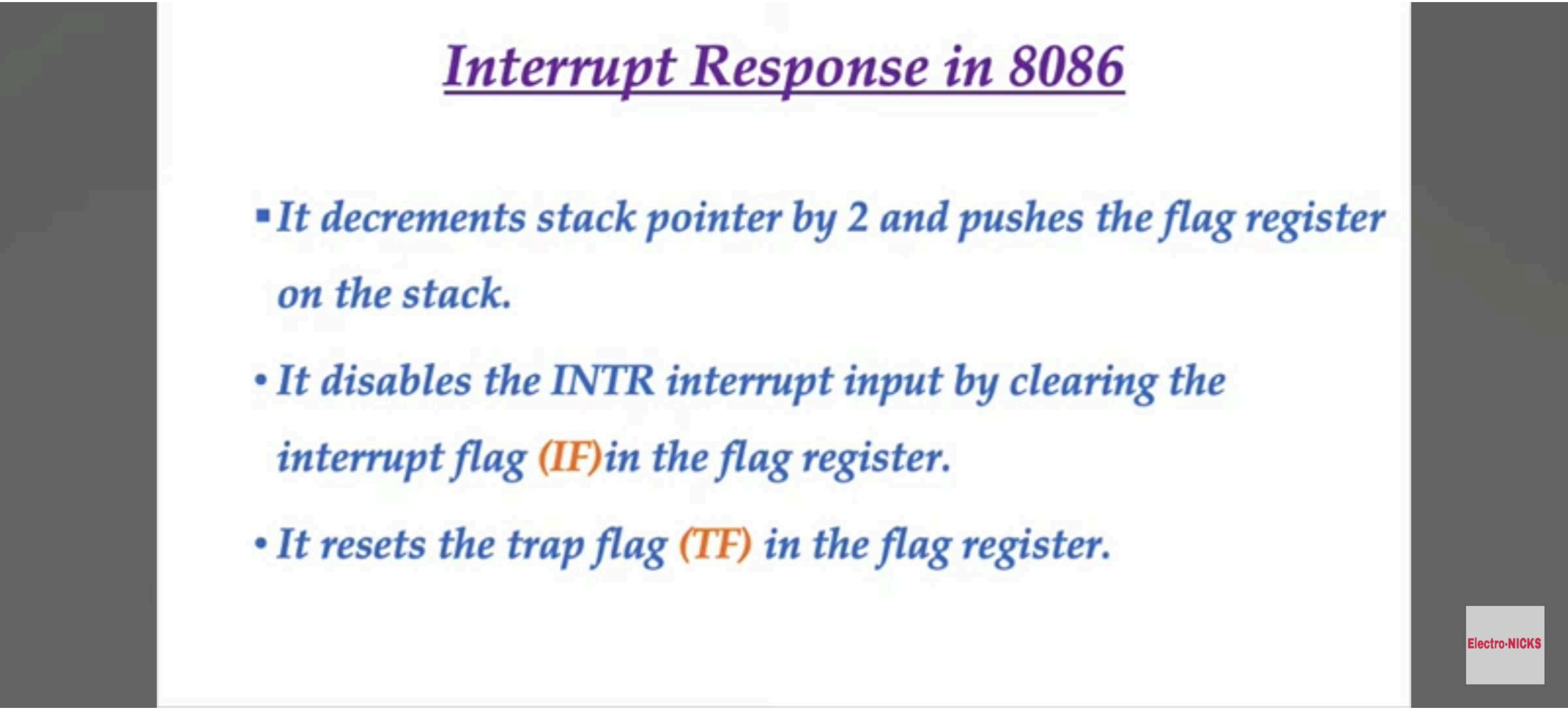
PUSH Flags
Clear IF
Clear TF
PUSH CS
PUSH IP
Fetch ISR Address

Interrupt Service Procedure

PUSH Registers

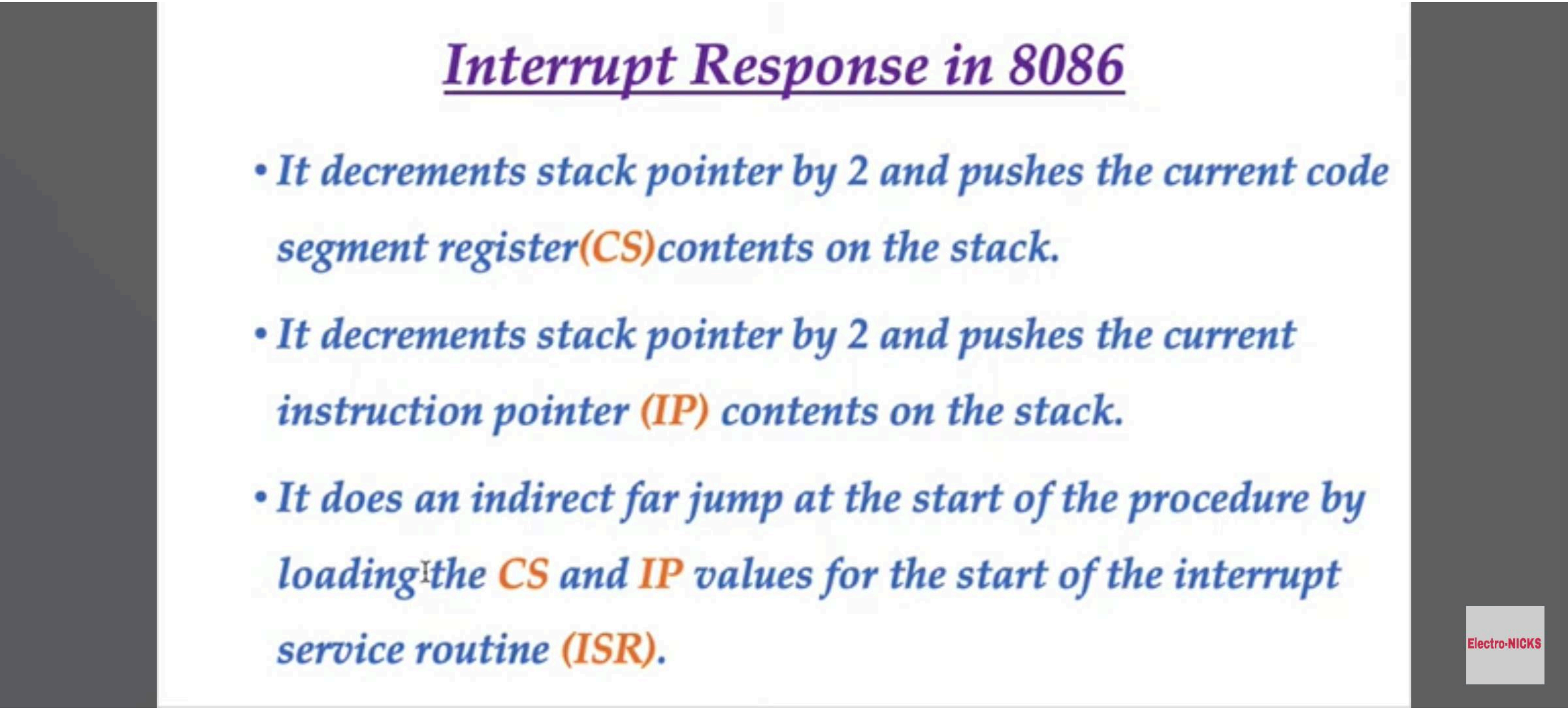
POP Registers
IRET

POP IP
POP CS
POP Flags



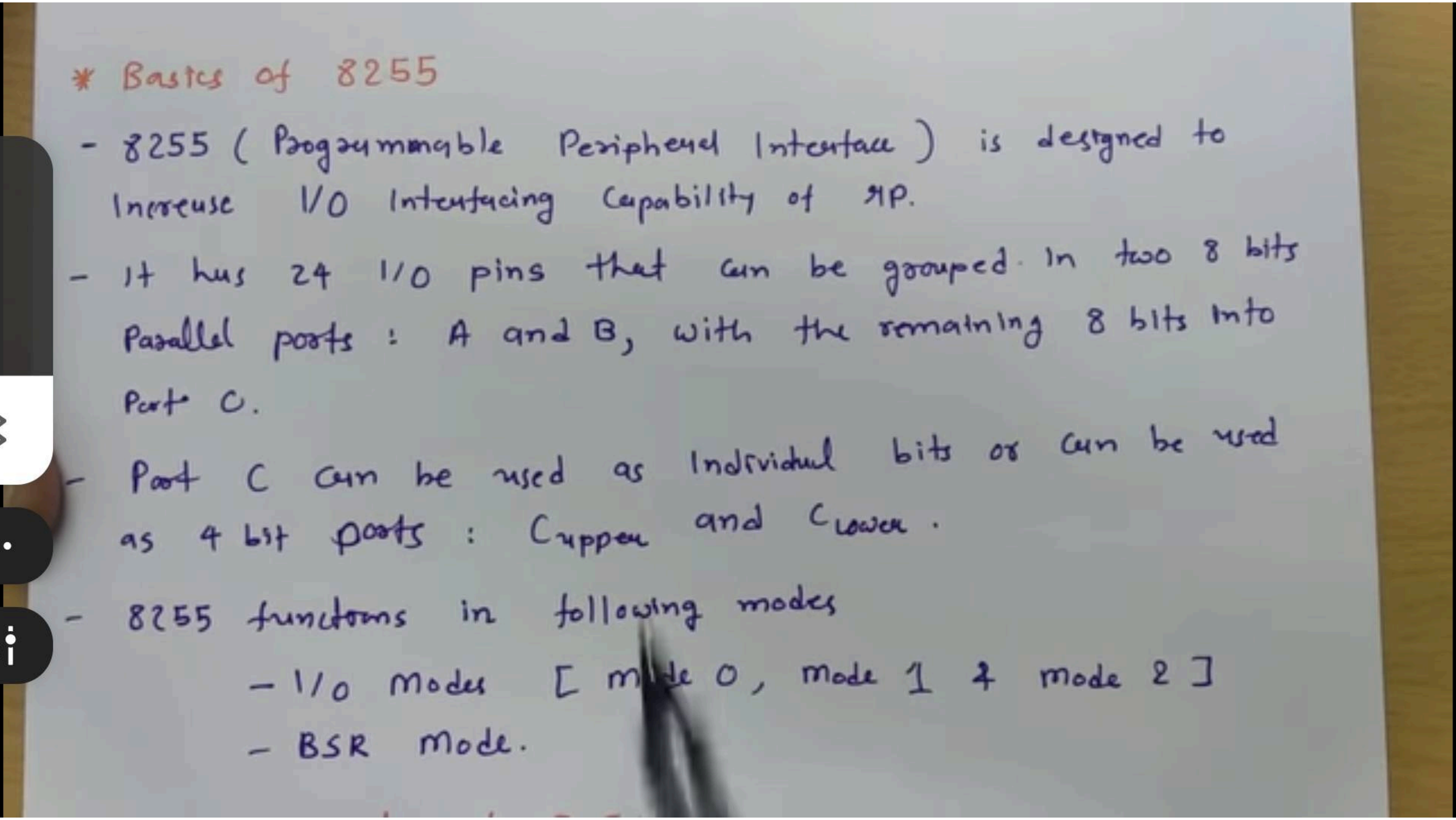
Interrupt Response in 8086

- *It decrements stack pointer by 2 and pushes the flag register on the stack.*
- *It disables the INTR interrupt input by clearing the interrupt flag (IF)in the flag register.*
- *It resets the trap flag (TF) in the flag register.*



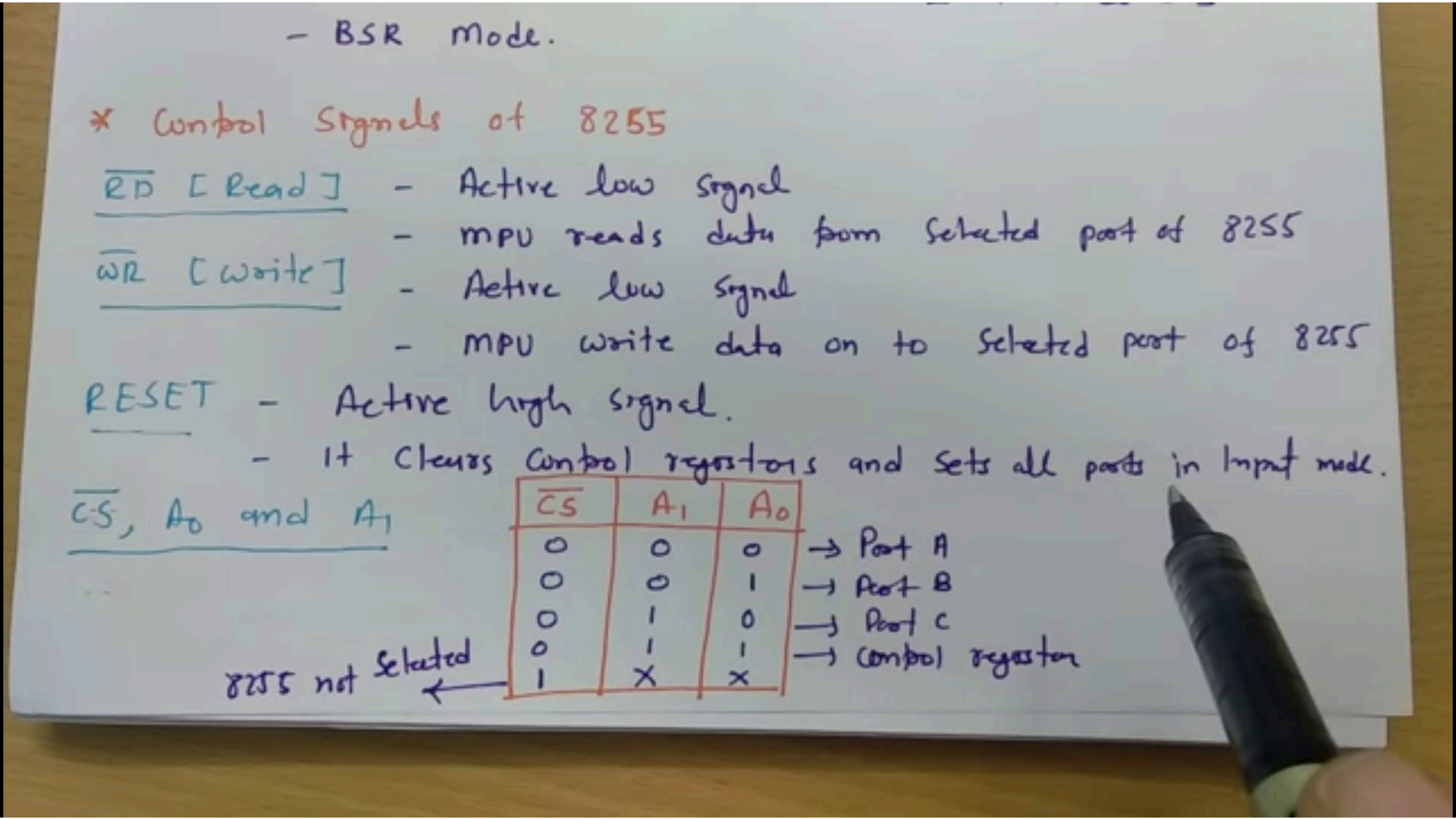
Interrupt Response in 8086

- It decrements stack pointer by 2 and pushes the current code segment register(CS)contents on the stack.
- It decrements stack pointer by 2 and pushes the current instruction pointer (IP) contents on the stack.
- It does an indirect far jump at the start of the procedure by loading the CS and IP values for the start of the interrupt service routine (ISR).



* Basics of 8255

- 8255 (Programmable Peripheral Interface) is designed to Increase I/O Interfacing Capability of μP.
- It has 24 I/O pins that can be grouped in two 8 bits Parallel ports : A and B, with the remaining 8 bits into Port C.
- Port C can be used as Individual bits or can be used as 4 bit ports : C_{upper} and C_{lower} .
- 8255 functions in following modes
  - I/O Modes [mode 0, mode 1 & mode 2]
  - BSR Mode.



- BSR mode.

* Control Signals of 8255

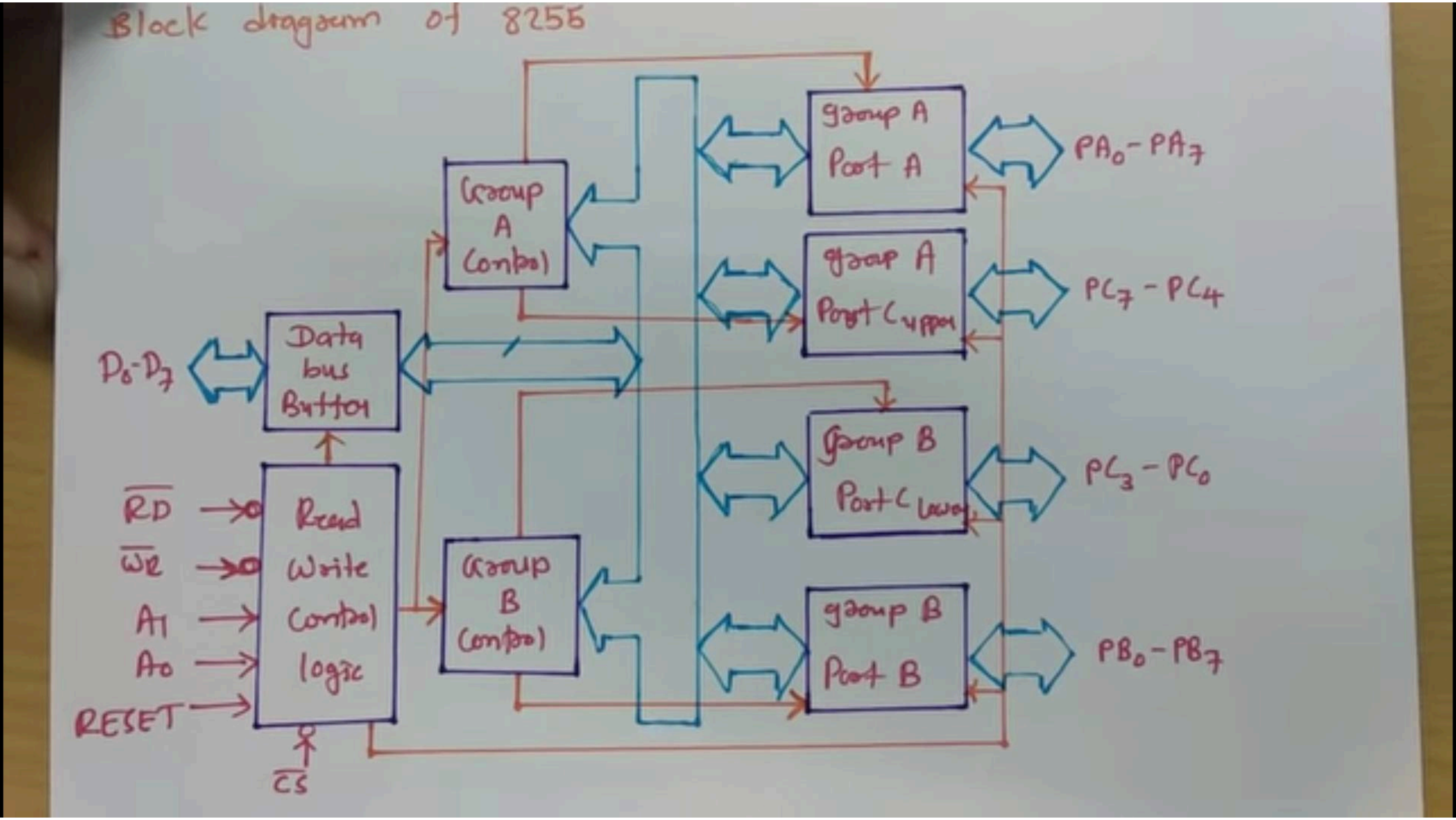
$\overline{RD}$ [Read] - Active low signal
- MPU reads data from Selected port of 8255

$\overline{WR}$ [write] - Active low signal
- MPU write data on to Selected port of 8255

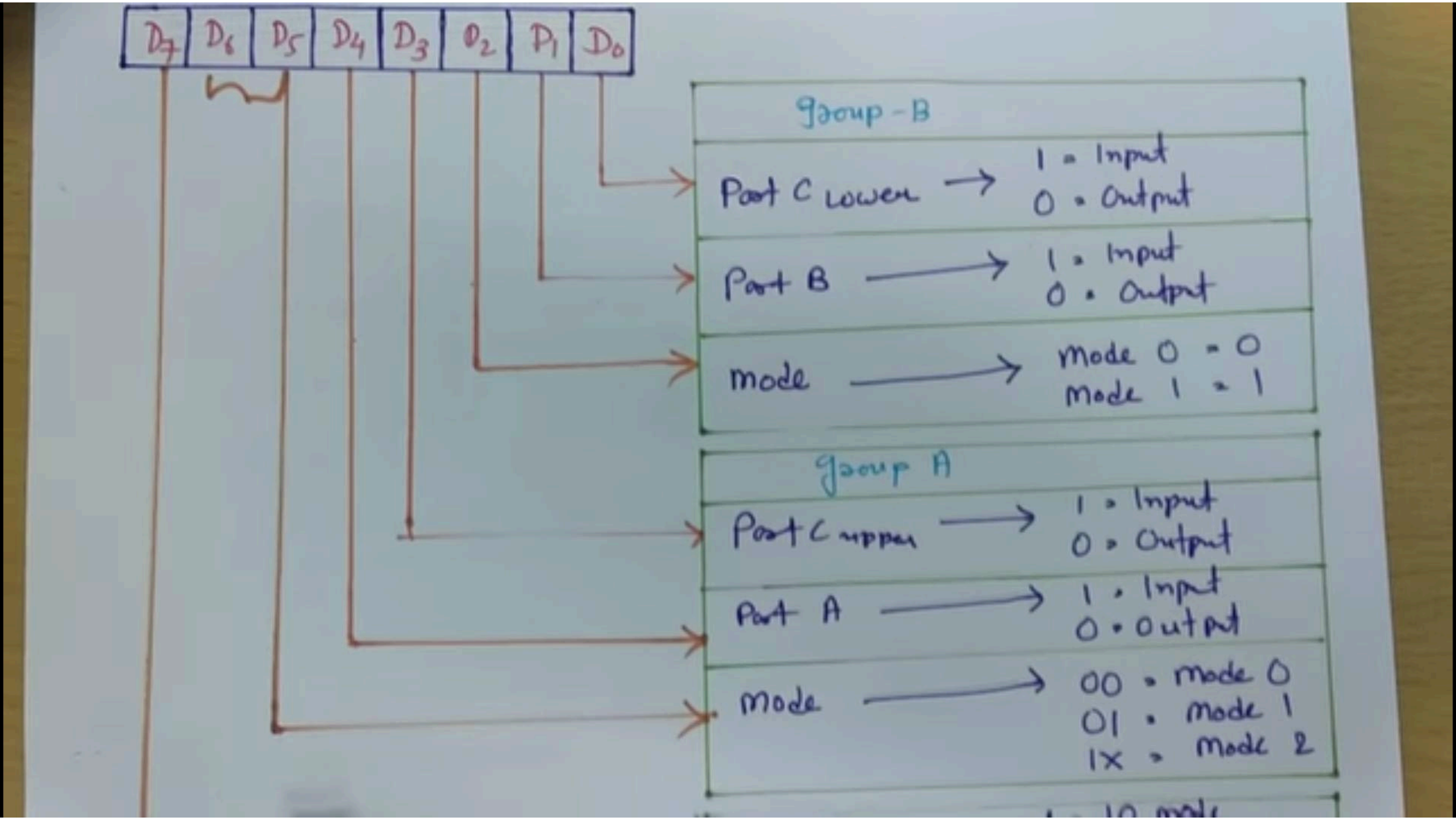
RESET - Active high signal.
- It Clears Control registors and sets all ports in Input mode.

$\overline{CS}$, A_0 and A_1

$\overline{CS}$	A_1	A_0	
0	0	0	→ Port A
0	0	1	→ Port B
0	1	0	→ Port C
0	1	1	→ Control register
1	X	X	8255 not Selected ←



Block dragram of 8255



group-B

Port C Lower → 1 = Input, 0 = Output

Port B → 1 = Input, 0 = Output

mode → mode 0 = 0, Mode 1 = 1

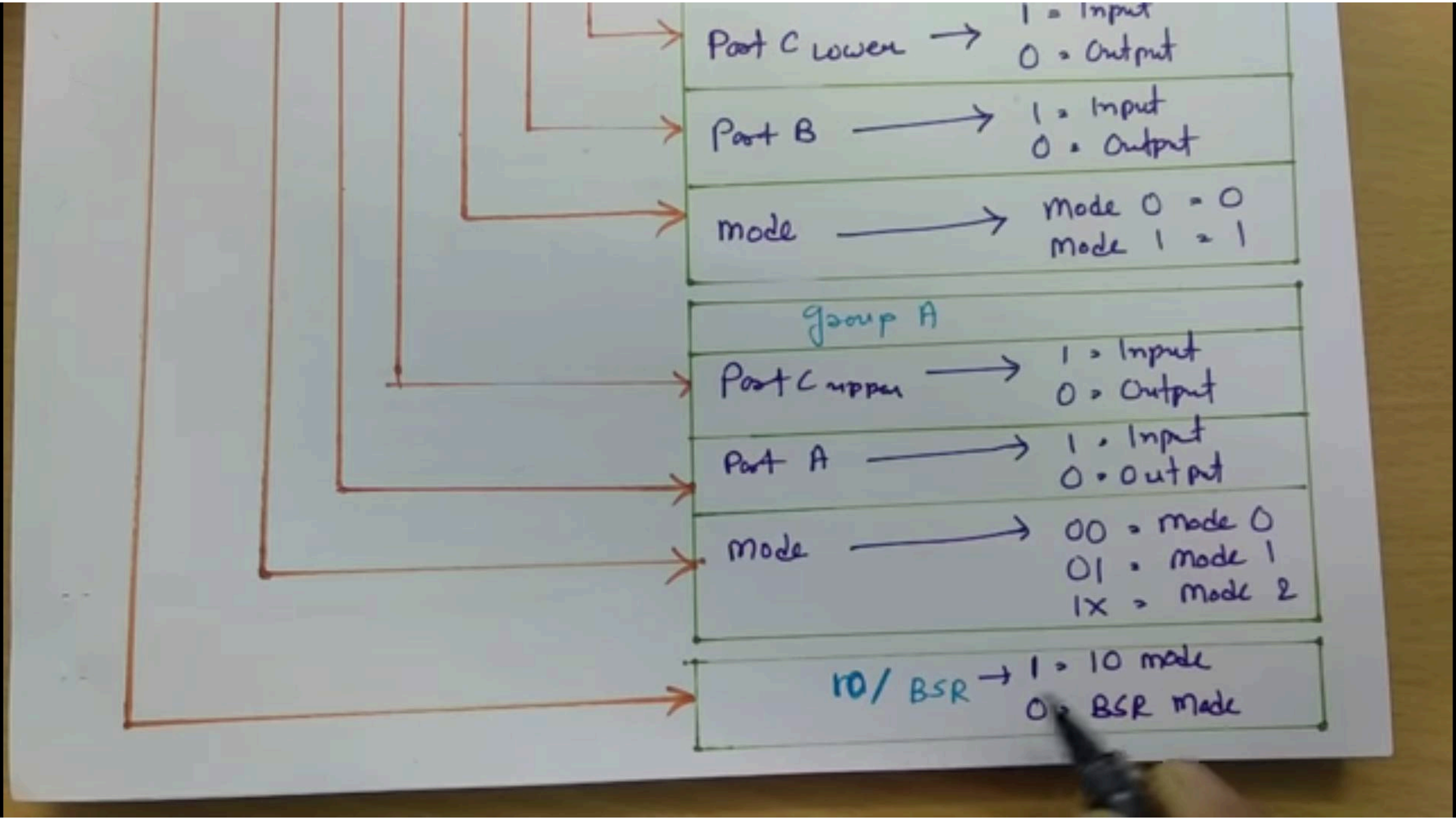
group A

Port C upper → 1 = Input, 0 = Output

Port A → 1 = Input, 0 = output

Mode → 00 = mode 0, 01 = mode 1, 1X = Mode 2

1 = IO mode



Port C Lower → 1 = Input
0 = Output

Port B → 1 = Input
0 = Output

mode → mode 0 = 0
mode 1 = 1

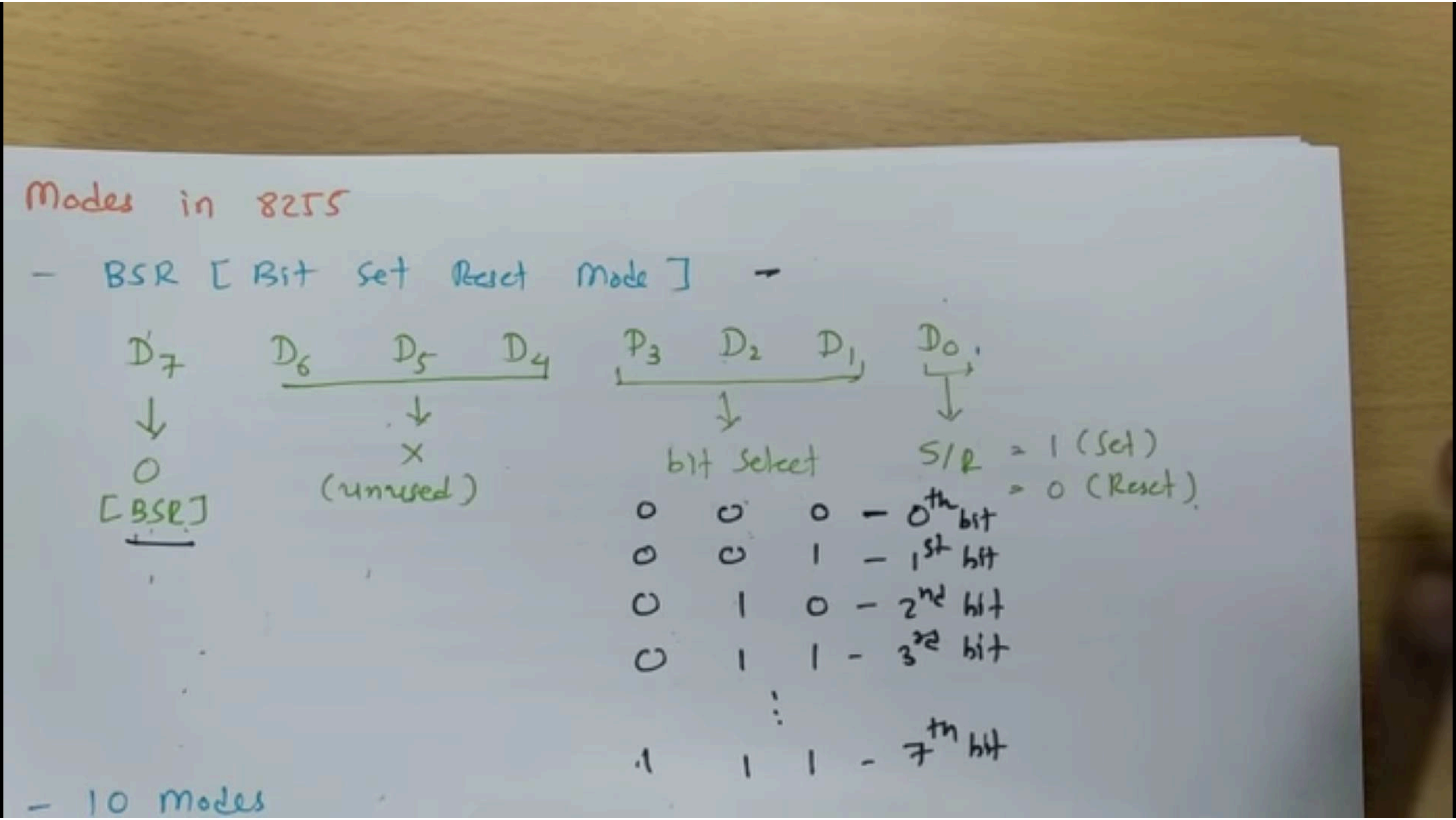
group A

Port C upper → 1 = Input
0 = Output

Port A → 1 = Input
0 = Output

mode → 00 = mode 0
01 = mode 1
1X = mode 2

IO/BSR → 1 = IO mode
0 = BSR Mode



Modes in 8255

- BSR [Bit Set Reset Mode] -

$D_7 \quad \underbrace{D_6 \quad D_5 \quad D_4} \quad \underbrace{D_3 \quad D_2 \quad D_1} \quad D_0$

D_7 ↓ 0 [BSR]

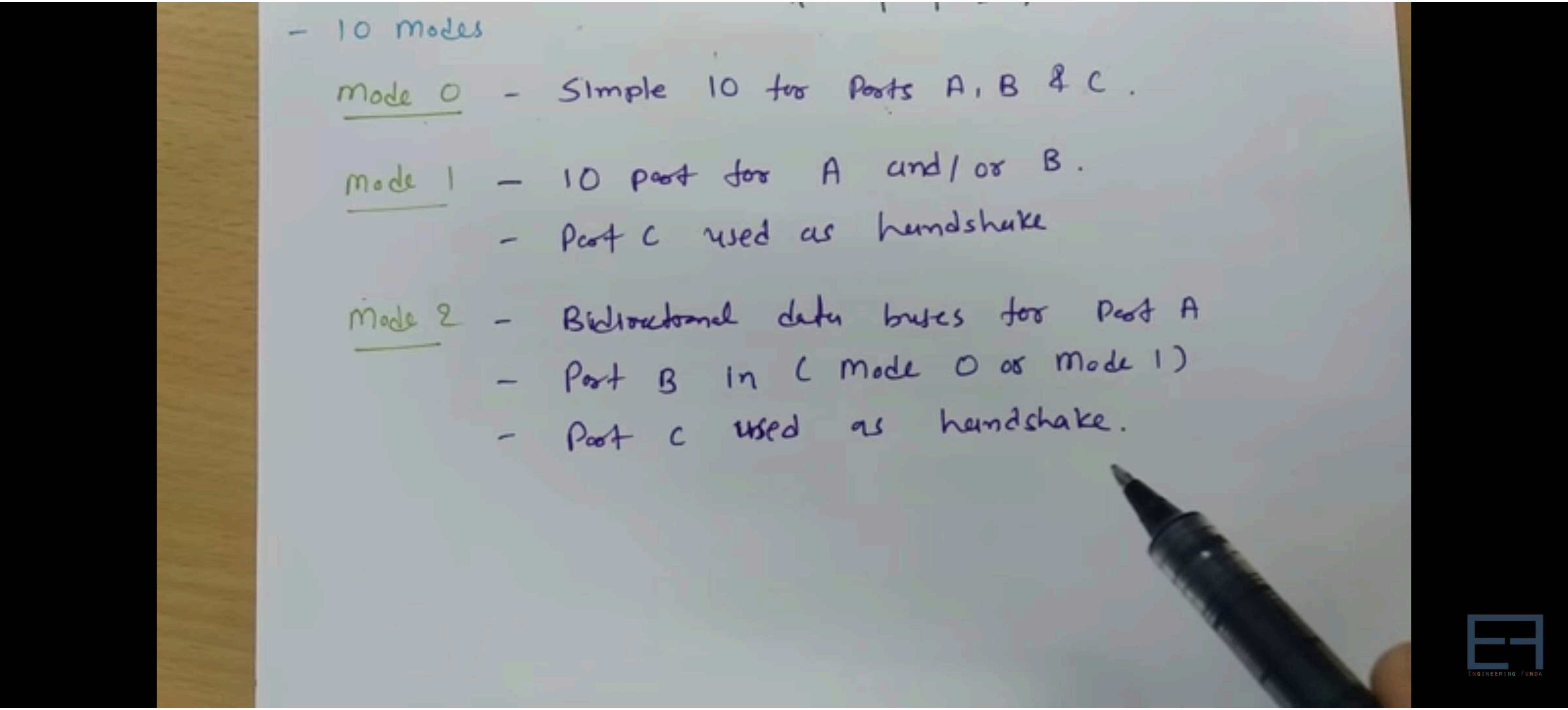
$D_6 \; D_5 \; D_4$ ↓ × (unused)

$D_3 \; D_2 \; D_1$ ↓ bit Select

D_0 ↓ S/R = 1 (Set)
= 0 (Reset)

0	0	0	- 0th bit
0	0	1	- 1st bit
0	1	0	- 2nd bit
0	1	1	- 3rd bit
	⋮		
1	1	1	- 7th bit

- 10 Modes



- 10 Modes

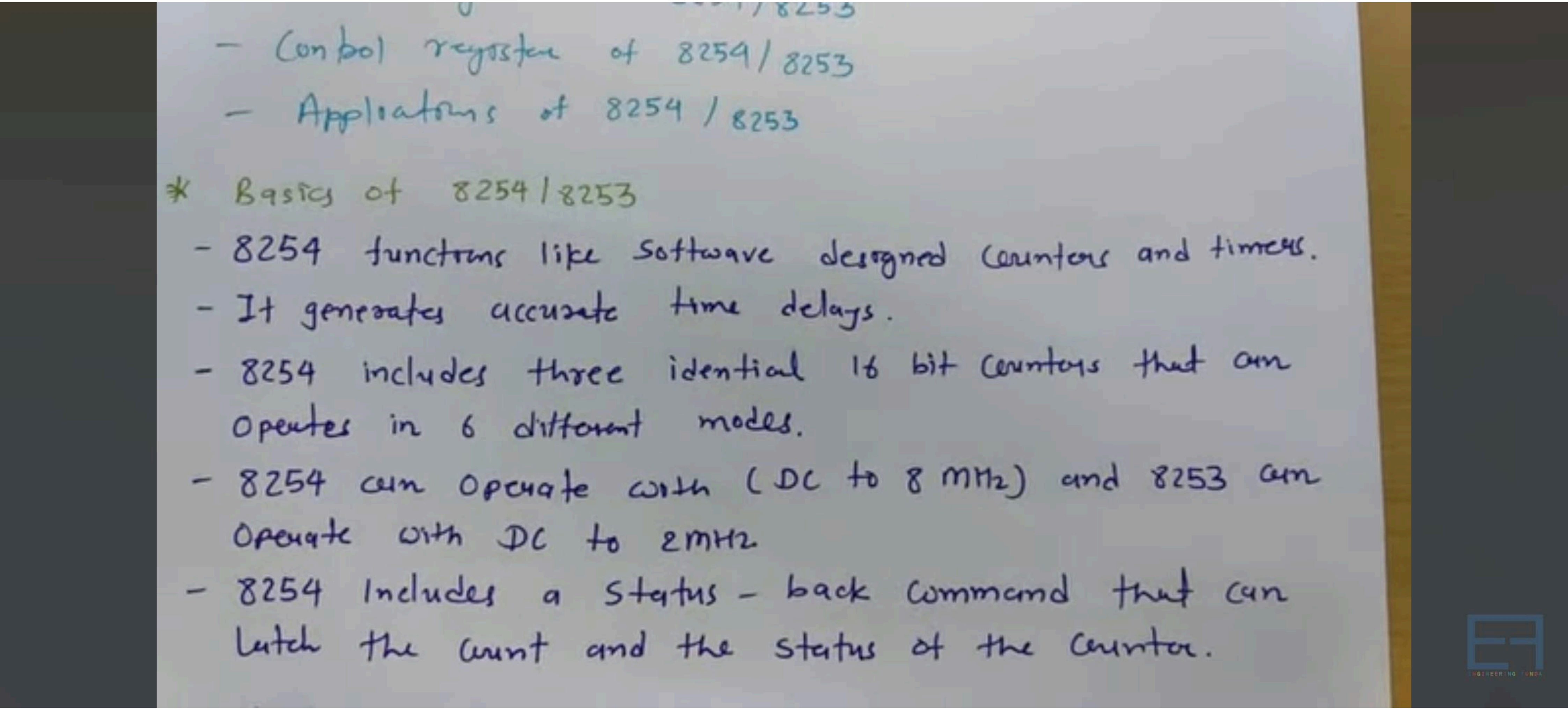
Mode 0 - Simple 10 for Ports A, B & C.

Mode 1 - 10 port for A and/or B.

- Port C used as handshake

Mode 2 - Bidirectional data buses for Port A

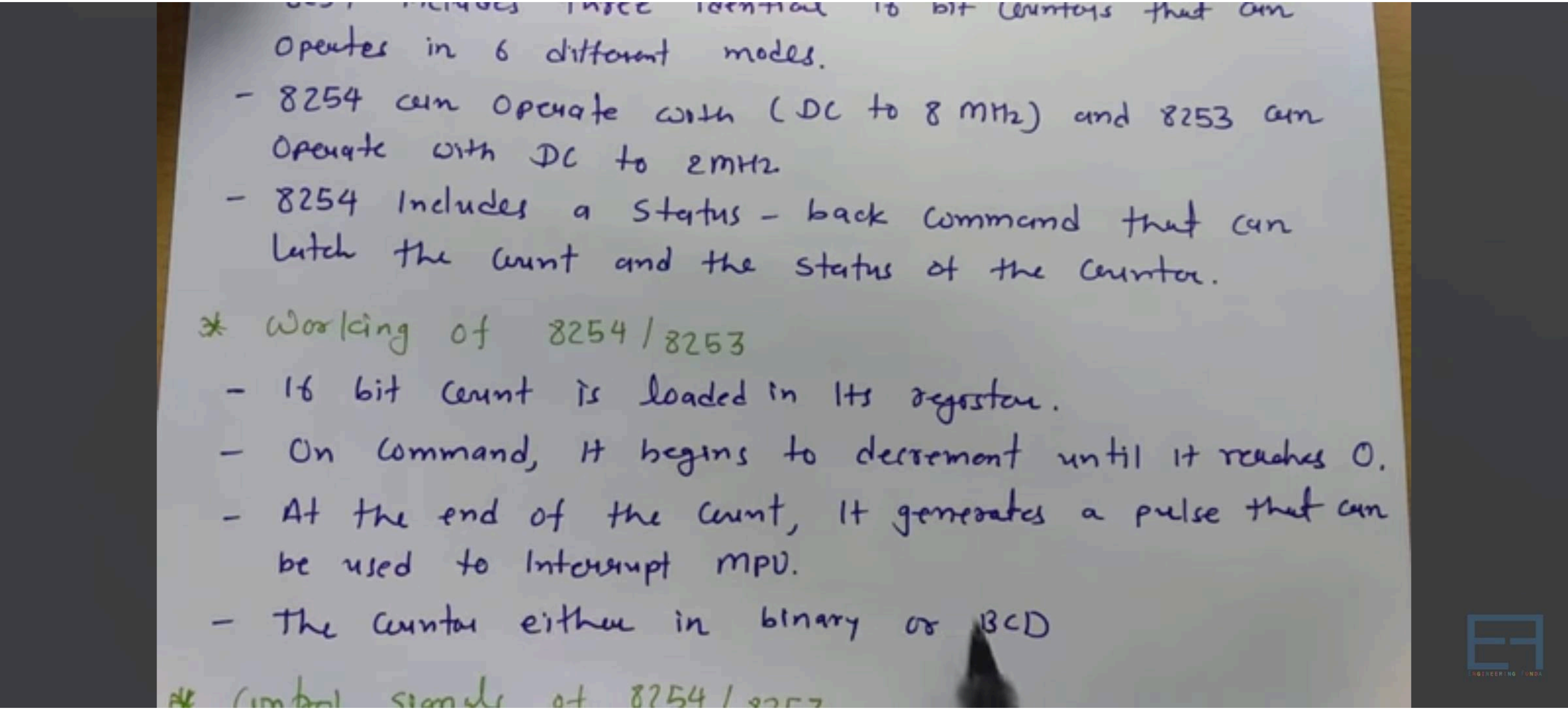
- Port B in (Mode 0 or Mode 1)
- Port C used as handshake.



- Control register of 8254/8253
- Applications of 8254 / 8253

* Basics of 8254/8253

- 8254 functions like Software designed counters and timers.
- It generates accurate time delays.
- 8254 includes three identical 16 bit counters that can operates in 6 different modes.
- 8254 can operate with (DC to 8 MHz) and 8253 can operate with DC to 2MHz
- 8254 Includes a Status - back command that can Latch the count and the status of the counter.



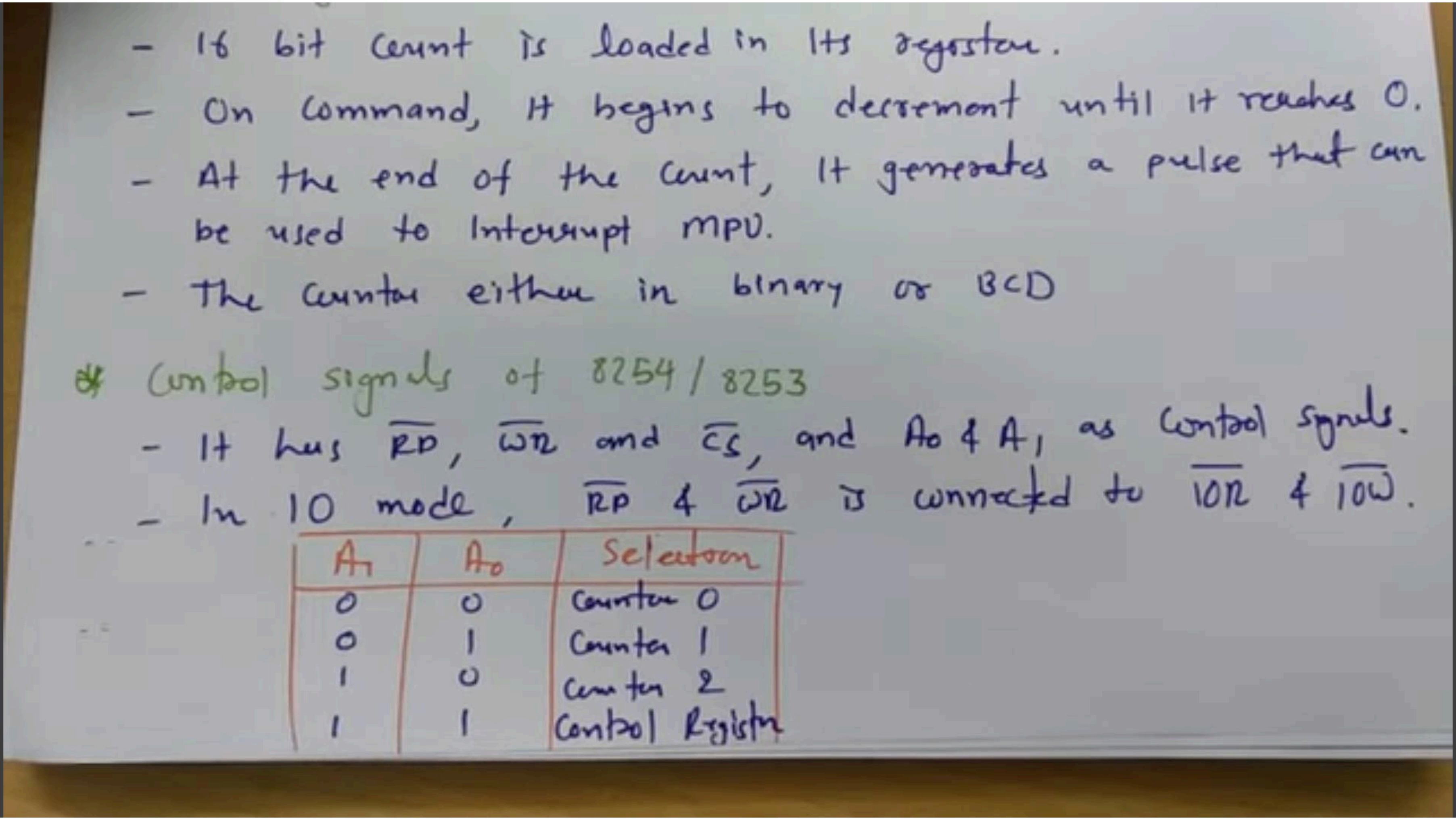
includes three identical 16 bit counters that can operates in 6 different modes.

- 8254 can operate with (DC to 8 MHz) and 8253 can operate with DC to 2MHz
- 8254 Includes a Status - back command that can latch the count and the status of the counter.

* Working of 8254/8253

- 16 bit count is loaded in its register.
- On command, it begins to decrement until it reaches 0.
- At the end of the count, it generates a pulse that can be used to Interrupt MPU.
- The counter either in binary or BCD

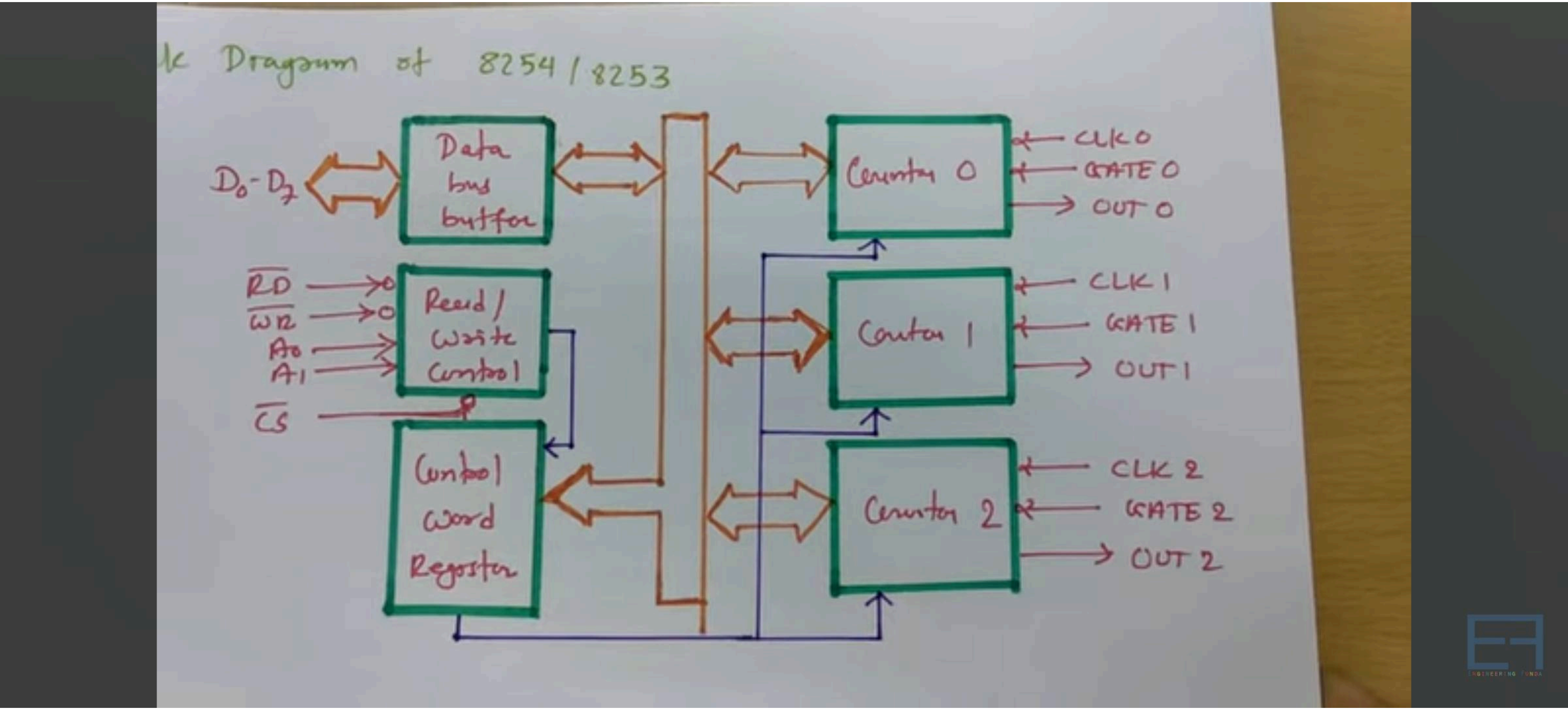
* Control signals of 8254/8253



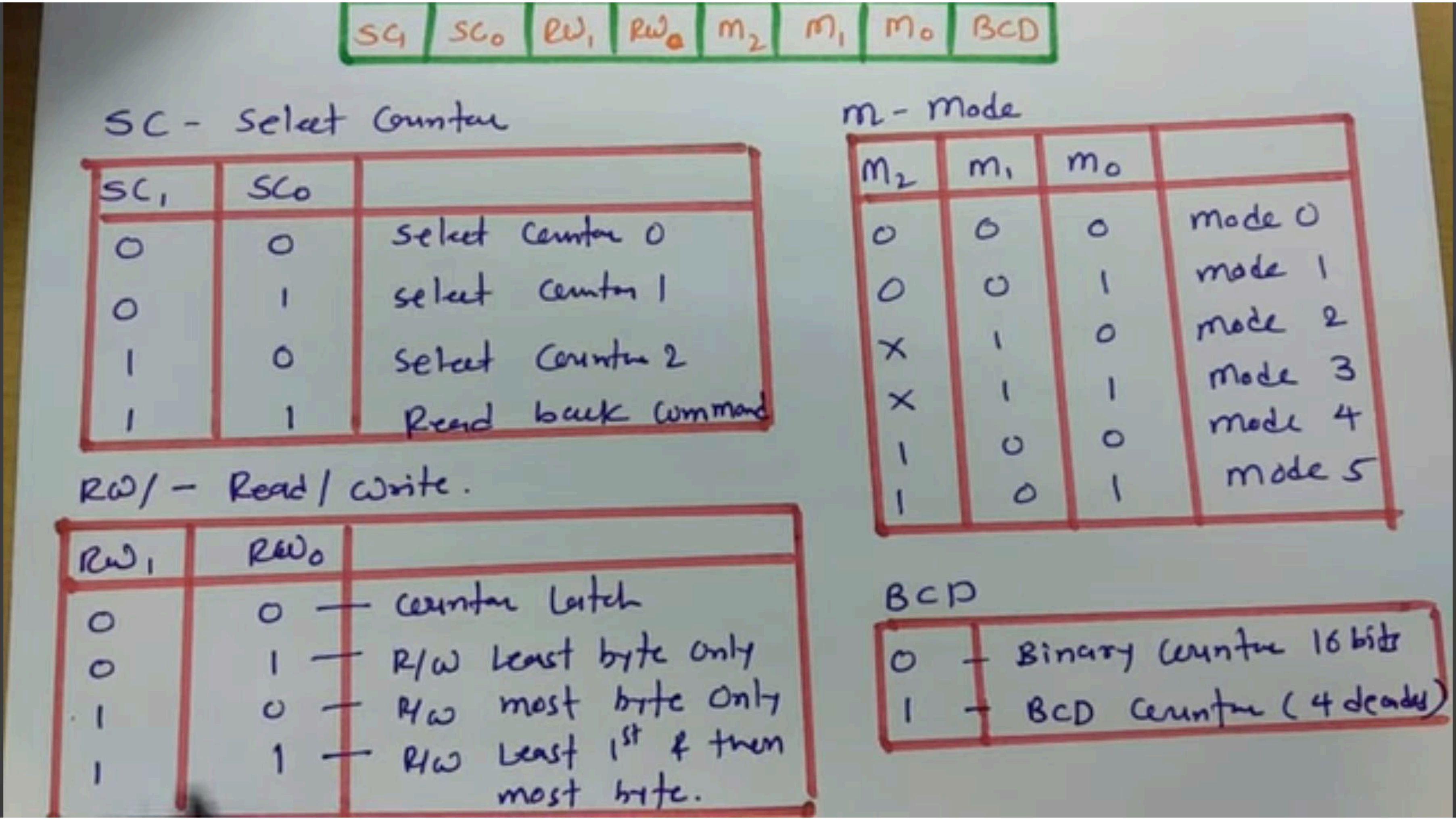
- 16 bit count is loaded in its register.
- On command, it begins to decrement until it reaches 0.
- At the end of the count, it generates a pulse that can be used to Interrupt MPU.
- The counter either in binary or BCD

* Control signals of 8254 / 8253
  - It has $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$, and A_0 & A_1 as control signals.
  - In IO mode, $\overline{RD}$ & $\overline{WR}$ is connected to $\overline{IOR}$ & $\overline{IOW}$.

A_1	A_0	Selection
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Register



k Dragram of 8254 / 8253



SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC - Select Counter

SC1	SC0	
0	0	select counter 0
0	1	select counter 1
1	0	select counter 2
1	1	Read back command

RW/ - Read / write.

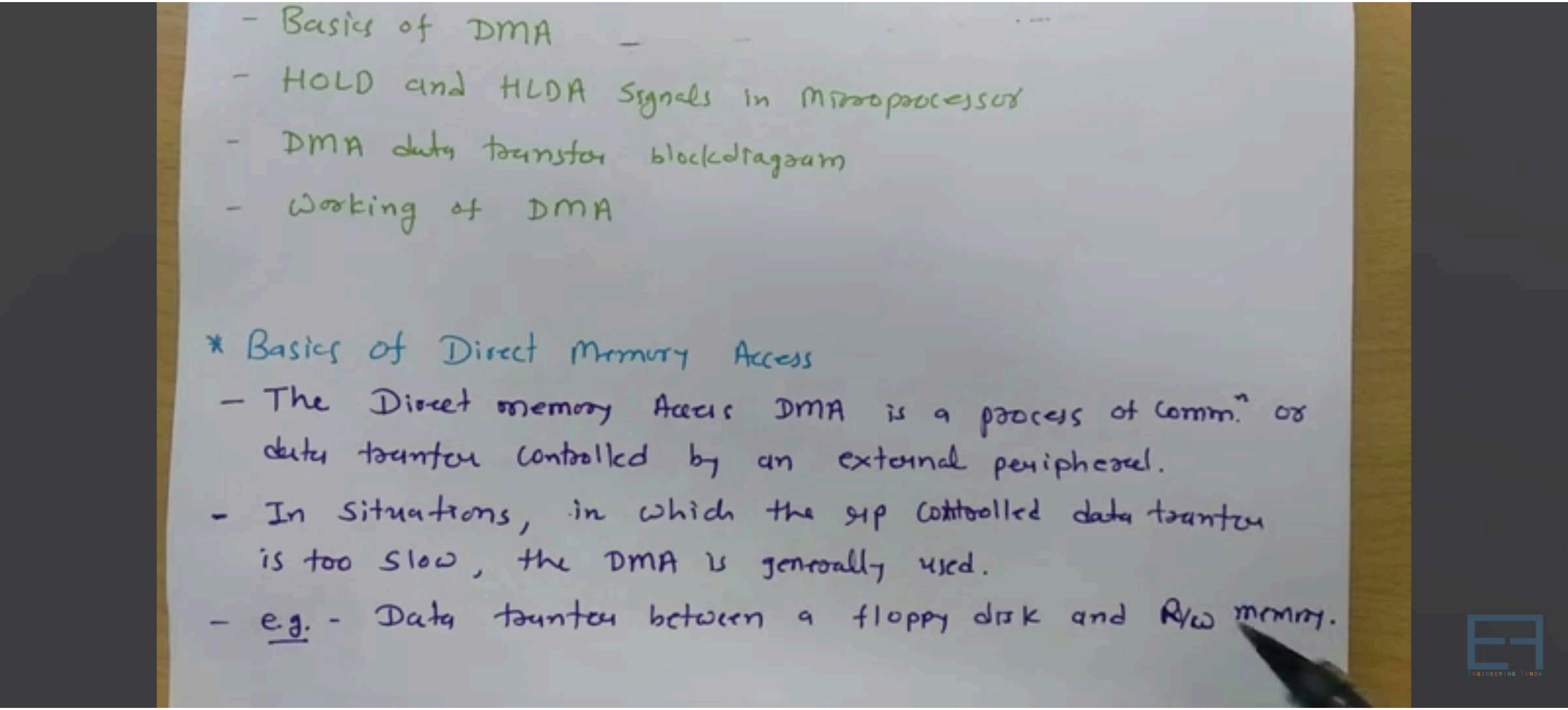
RW1	RW0	
0	0 —	counter Latch
0	1 —	R/W Least byte only
1	0 —	R/W most byte Only
1	1 —	R/W Least 1st & then most byte.

M - Mode

M2	M1	M0	
0	0	0	mode 0
0	0	1	mode 1
X	1	0	mode 2
X	1	1	mode 3
1	0	0	mode 4
1	0	1	mode 5

BCD

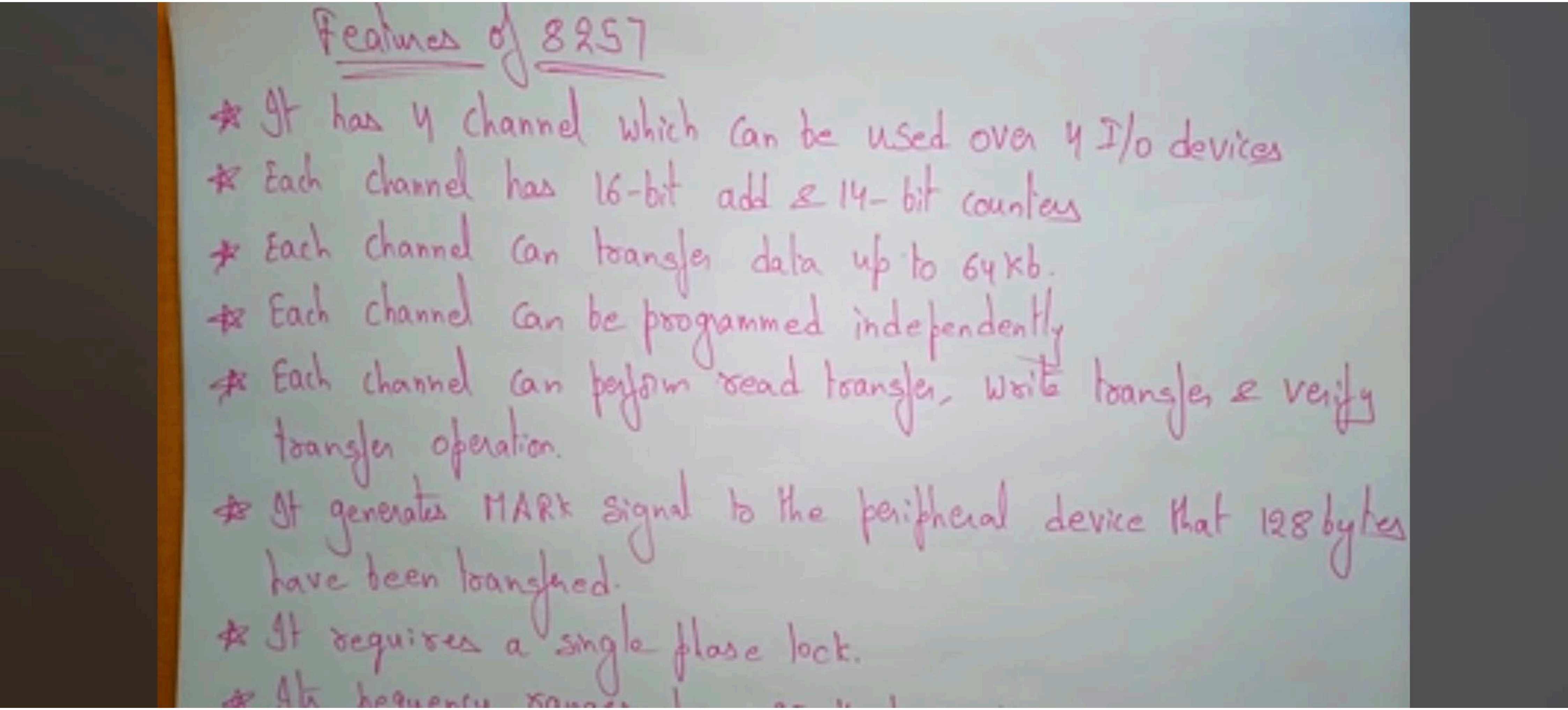
0	- Binary counter 16 bits
1	- BCD Counter (4 decades)



- Basics of DMA
- HOLD and HLDA Signals in Microprocessor
- DMA data transfer blockdiagram
- Working of DMA

* Basics of Direct Memory Access

- The Direct memory Access DMA is a process of comm^{n}. or data transfer controlled by an external peripheral.
- In situations, in which the μp controlled data transfer is too slow, the DMA is generally used.
- e.g. - Data transfer between a floppy disk and R/W memory.



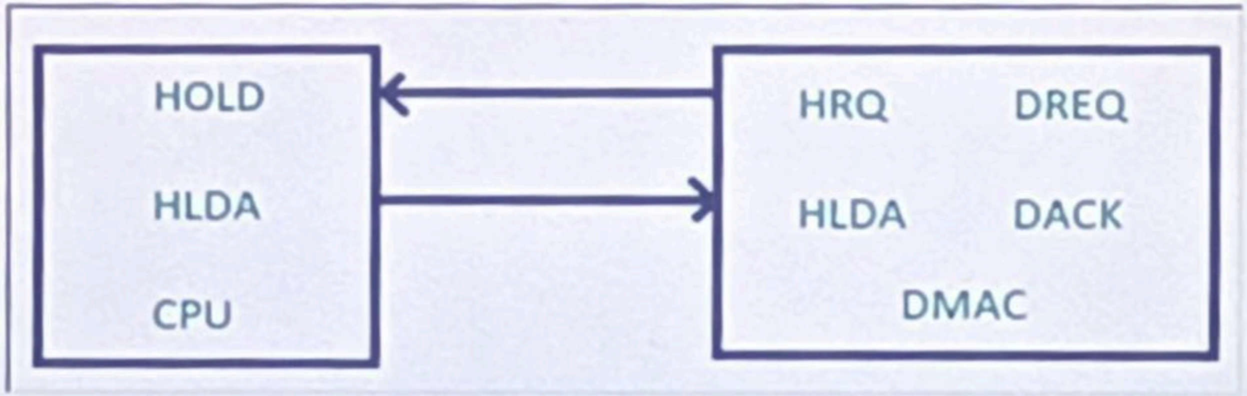
Features of 8257

* It has 4 channel which can be used over 4 I/O devices
* Each channel has 16-bit add & 14-bit counters
* Each channel can transfer data up to 64kb.
* Each channel can be programmed independently
* Each channel can perform read transfer, write transfer & verify transfer operation.
* It generates MARK signal to the peripheral device that 128 bytes have been transferred.
* It requires a single phase clock.
* Its frequency range

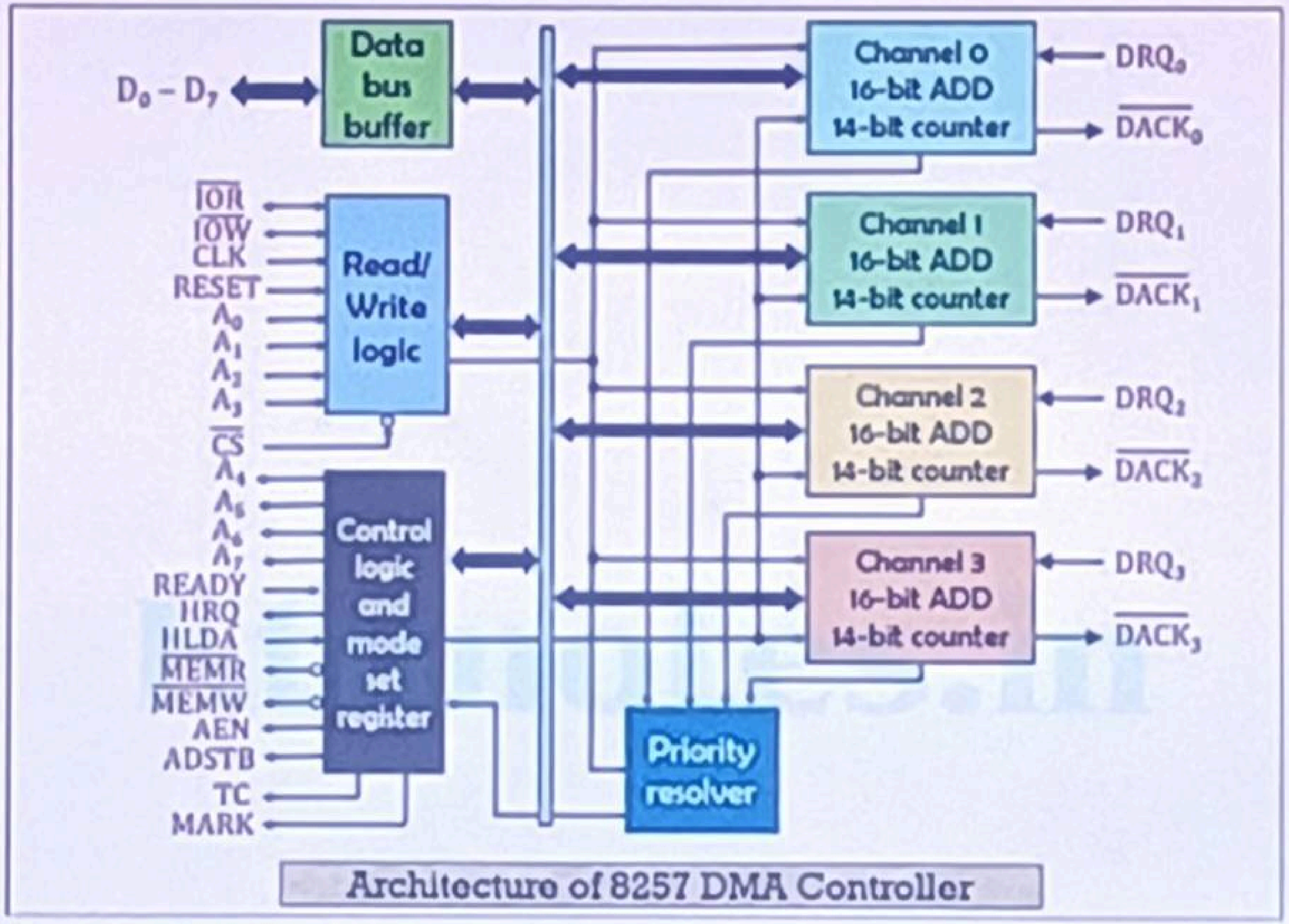
- Each channel can be programmed independently.
- Each channel can perform read transfer, write transfer and verify transfer operations.
- It generates MARK signal to the peripheral device that 128 bytes have been transferred.
- It requires a single-phase clock.
- Its frequency ranges from 250Hz to 3MHz.
- It operates in 2 modes, i.e., Master mode and Slave mode.

Modes of DMAC:

1. Single Mode – In this only one channel is used, means only a single DMAC is connected to the bus system.



8257 - Architecture

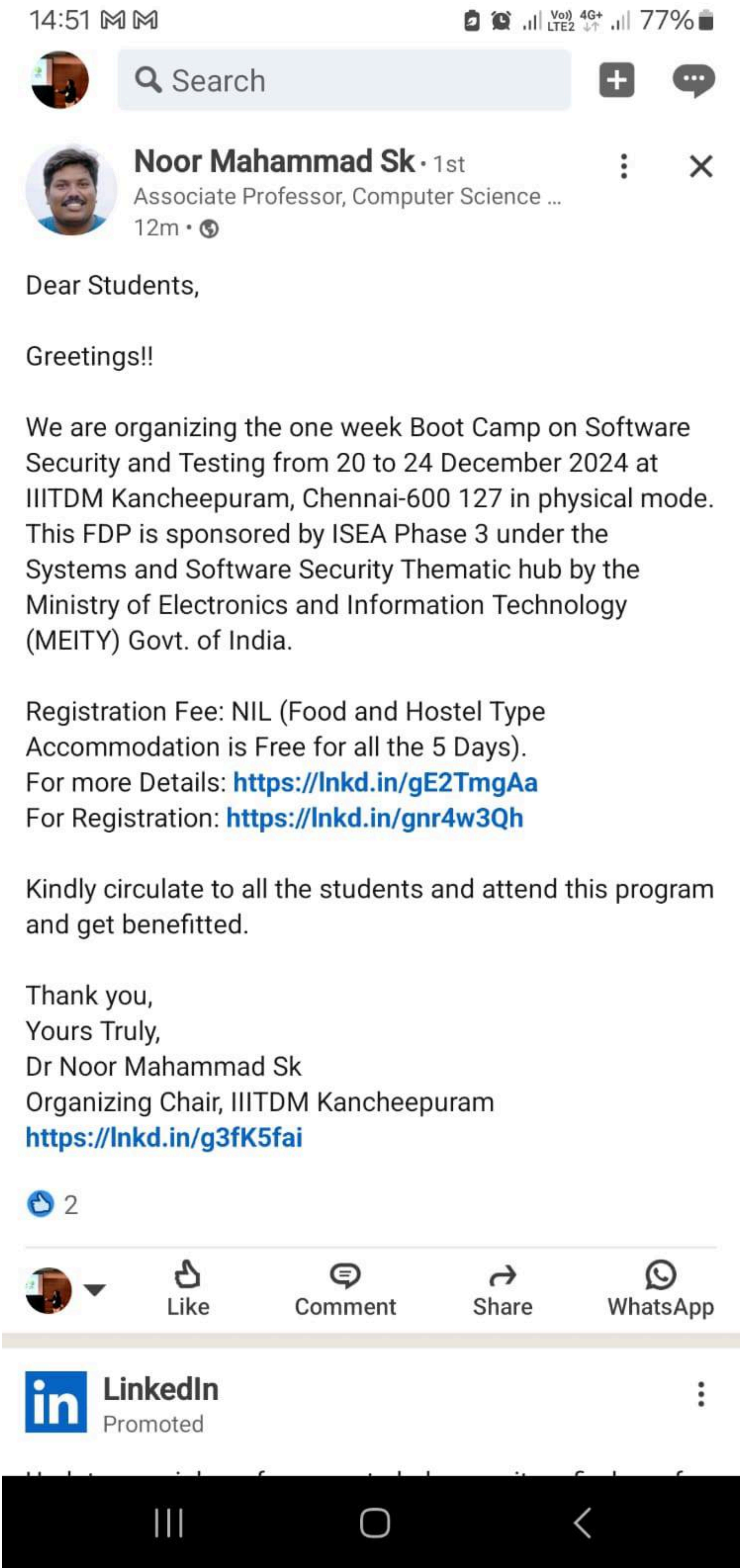


D0 – D7
Data bus buffer
IOR
IOW
CLK
RESET
A0
A1
A2
A3
Read/ Write logic
CS
A4
A5
A6
A7
READY
HRQ
HLDA
MEMR
MEMW
AEN
ADSTB
TC
MARK
Control logic and mode set register
Channel 0 16-bit ADD 14-bit counter
DRQ0
DACK0
Channel 1 16-bit ADD 14-bit counter
DRQ1
DACK1
Channel 2 16-bit ADD 14-bit counter
DRQ2
DACK2
Channel 3 16-bit ADD 14-bit counter
DRQ3
DACK3
Priority resolver

Architecture of 8257 DMA Controller

It consists of five functional blocks:

a) Data bus buffer
b) Control logic
c) Read/write logic
d) Priority Resolver
e) DMA channels



Noor Mahammad Sk • 1st
Associate Professor, Computer Science ...
12m •

Dear Students,

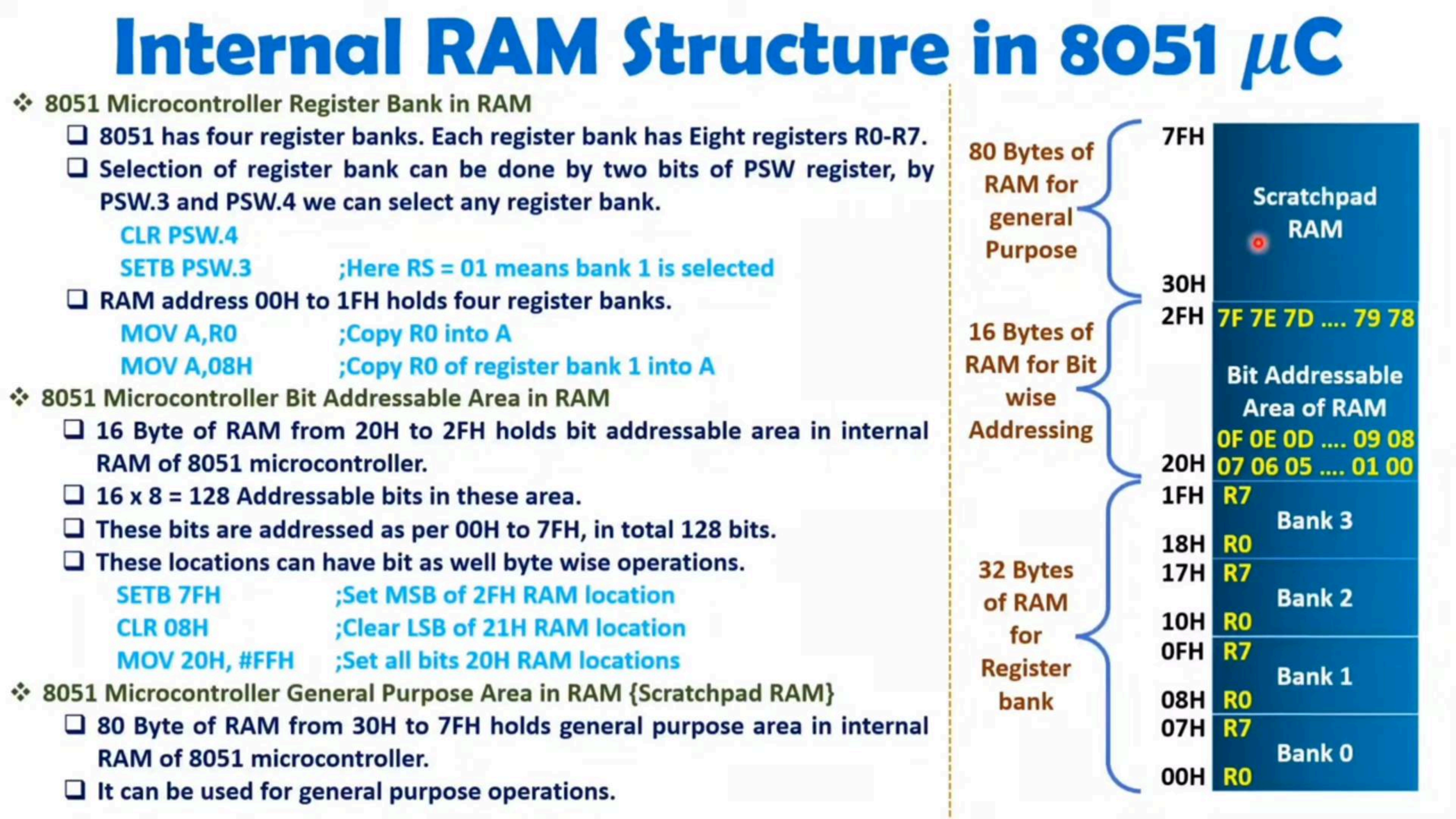
Greetings!!

We are organizing the one week Boot Camp on Software Security and Testing from 20 to 24 December 2024 at IIITDM Kancheepuram, Chennai-600 127 in physical mode. This FDP is sponsored by ISEA Phase 3 under the Systems and Software Security Thematic hub by the Ministry of Electronics and Information Technology (MEITY) Govt. of India.

Registration Fee: NIL (Food and Hostel Type Accommodation is Free for all the 5 Days).
For more Details: https://lnkd.in/gE2TmgAa
For Registration: https://lnkd.in/gnr4w3Qh

Kindly circulate to all the students and attend this program and get benefitted.

Thank you,
Yours Truly,
Dr Noor Mahammad Sk
Organizing Chair, IIITDM Kancheepuram
https://lnkd.in/g3fK5fai



Internal RAM Structure in 8051 μC

❖ 8051 Microcontroller Register Bank in RAM

- ❑ 8051 has four register banks. Each register bank has Eight registers R0-R7.
- ❑ Selection of register bank can be done by two bits of PSW register, by PSW.3 and PSW.4 we can select any register bank.

```
CLR PSW.4
SETB PSW.3          ;Here RS = 01 means bank 1 is selected
```

- ❑ RAM address 00H to 1FH holds four register banks.

```
MOV A,R0            ;Copy R0 into A
MOV A,08H           ;Copy R0 of register bank 1 into A
```

❖ 8051 Microcontroller Bit Addressable Area in RAM

- ❑ 16 Byte of RAM from 20H to 2FH holds bit addressable area in internal RAM of 8051 microcontroller.
- ❑ 16 x 8 = 128 Addressable bits in these area.
- ❑ These bits are addressed as per 00H to 7FH, in total 128 bits.
- ❑ These locations can have bit as well byte wise operations.

```
SETB 7FH            ;Set MSB of 2FH RAM location
CLR 08H             ;Clear LSB of 21H RAM location
MOV 20H, #FFH       ;Set all bits 20H RAM locations
```

❖ 8051 Microcontroller General Purpose Area in RAM {Scratchpad RAM}

- ❑ 80 Byte of RAM from 30H to 7FH holds general purpose area in internal RAM of 8051 microcontroller.
- ❑ It can be used for general purpose operations.

80 Bytes of RAM for general Purpose

7FH

Scratchpad RAM

30H

16 Bytes of RAM for Bit wise Addressing

2FH 7F 7E 7D 79 78

Bit Addressable Area of RAM

0F 0E 0D 09 08

20H 07 06 05 .... 01 00

32 Bytes of RAM for Register bank

1FH R7

Bank 3

18H R0

17H R7

Bank 2

10H R0

0FH R7

Bank 1

08H R0

07H R7

Bank 0

00H R0