Microprocessor Technologies (102045610)

MODULE 5

INTERFACING PERIPHERALS AND APPLICATIONS

Module 5

- Concepts of Interrupts and interrupt programming
- Vector interrupts and restart instructions
- Interfacing of programmable peripheral interface (8255)
- Interfacing of data converters

Interrupts in 8085

- An interrupt is a signal or condition that causes processor to stop its normal execution flow and makes it to jump to some other location for processing the interrupt/other task.
- 8085 has 4 mask-able interrupts and 1 non-mask-able interrupts. Maskable interrupts can be disabled by DI instruction.
- Among four Maskable interrupts one is non-vectored interrupt, that is processor cannot go to a fixed location as in case of vectored interrupt, the external device which caused interrupts needs to specify the vector address.

8085 interrupt response process:

- Interrupts should be enabled by using EI instruction, then only processor responds to all mask able interrupts.
- When microprocessor is executing a program, it checks for INTR line during execution of each instruction.
- If INTR is high then processor completes executing the current instruction, disables the interrupts and sends a $\overline{\text{INTA}}$ signal.
- $\overline{\text{INTA}}$ is used by the external hardware to specify the restart instruction to processor (since INTR is a non-vectored interrupt).
- When microprocessor receives the RST instruction, it saves PC content on stack and PC is loaded with the vector address specified by interrupt.
- Microprocessor executes the instructions at vector address.
- The interrupts should be enabled if required in the ISR(interrupt service routine)
- At the end of interrupt service routine, RET instruction loads the PC from the stack. So processor comes back to the instruction where it was interrupted previously.

3. Hardware Interrupt

Interrupt	Types	Instructions	Hardware	Trigger	Vector	Priority
TRAP (RST 4.5)	Nonmaskable	• Independent of EI and DI	No External Hardware	Level & Edge Sensitive	0024H	1 Highest
RST 7.5	Maskable	• Controlled by EI and DI • Unmasked by SIM	No External Hardware	Edge Sensitive	003CH	2
RST 6.5	Maskable	• Controlled by EI and DI • Unmasked by SIM	No External Hardware	Level Sensitive	0034H	3
RST 5.5	Maskable	• Controlled by EI and DI • Unmasked by SIM	No External Hardware	Level Sensitive	002CH	4
INTR	Maskable	• Controlled by EI and DI	No External Hardware	Level Sensitive	----	5 Lowest

Table 3.1: Summary of interrupt in 8085

- To get the vector location for RST interrupts, interrupt value is multiplied by 8 and the result is converted to hexadecimal notation. For example RST 5.5 instruction, multiply $5.5 \times 8=44$ in hex 2CH. So vector address is 002CH.
- When more than one interrupts occur at the same time, then processor responds to them according to the above mentioned priority

Difference between the vectored and non-vectored interrupts

- **VECTORED INTERRUPT**

- In vectored interrupts, the processor automatically branches to the specific address in response to an interrupt.
- In vectored interrupts, the manufacturer fixes the address of the ISR to which the program control is to be transferred.
- The TRAP, RST 7.5, RST 6.5 and RST 5.5 are vectored interrupts.
- TRAP is the only non-maskable interrupt in the 8085.

- **NON-VECTORED INTERRUPT**

- In non-vectored interrupts the interrupted device should give the address of the interrupt service routine (ISR).
- The INTR is a non-vectored interrupt.
- Hence when a device interrupts through INTR, it has to supply the address of ISR after receiving interrupt acknowledge signal.

Vectored Interrupt	Non-Vectored Interrupt
The source that interrupts the CPU provides the branch information. This information is called the interrupt vector.	In non-vectored interrupt, the branch address is assigned to the fixed address in the memory
It has a memory address.	It does not have a memory address.
It allows the CPU to be able to know what ISR(Interrupt Service Routine) to carry out in software.	When a non-vectored interrupt is received it jumps into the program counter to fixed address in hardware.

Difference between maskable and non maskable interrupts

Non-Maskable Interrupts	Maskable Interrupts
It cannot be masked off or made pending.	It can be masked off or made pending.
This interrupt disables all maskable interrupts.	This interrupt does not disables all nonmaskable interrupts.
It is used for emergency purposes like power failure, smoke detectore etc.	It is used to interface peripherals.
It has higher priority.	It has lower priority.
Vectored	Vectored or non vectored.
Response time is low.	

. Restart instructions:

- These instructions are like software interrupts to 8085. When these instructions are executed processor vectors (Jumps) to a specific predetermined hard-wired memory location called restart location.

Restart Interrupt	Call Location in Hex
RST 0	0000H
RST 1	0008H
RST 2	0010H
RST 3	0018H
RST 4	0020H
RST 5	0028H
RST 6	0030H
RST 7	0038H

Table 4.1: Restart Instruction in 8085

- To get the vector location 'n' value is multiplied by 8 and the result is converted to hexadecimal notation. For example RST 3 instruction, multiply 3 x 8=24. In hexadecimal 18H. So vector address is 0018H.

6. Interrupt instructions

- In 8085, instructions used to control interrupt are explained below.

1) EI(Enable Interrupt)

- This is a 1 byte instruction
- The instruction sets the interrupt enable flip flop and enables the interrupt process.

2) DI(Disable Interrupt)

- This is a 1 byte instruction.
- The instruction resets the interrupt enable flip flop and disables the interrupt process.
- It should be included in a program segment where an interrupt from an outside source cannot be tolerated.

3) SIM(Set Interrupt Mask):

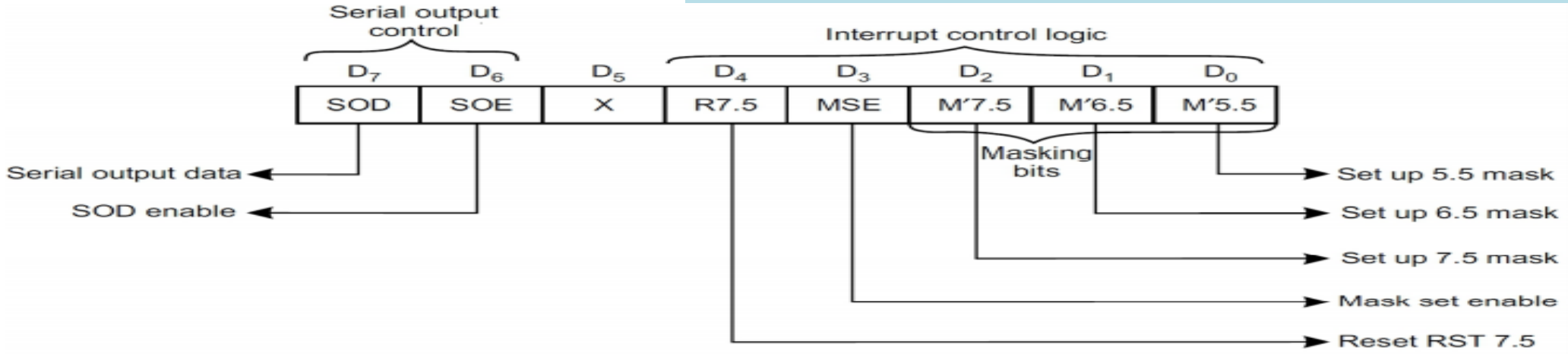


Figure 6.1: SIM instruction format

- D_7 and D_6 bits are utilized for serial outputting of data from accumulator.
- D_5 bit is don't care bit, while bits D_4–D_0 are used for interrupt control.
- D_4 bit can clear the D Flip Flop associated with RST 7.5.
- D_3 bit is mask set enable (MSE) bit, while bits D_2–D_0 are the masking bits corresponding to RST 7.5, RST 6.5 and RST 5.5 respectively.
- None of the flags are affected by SIM instruction.
- By employing SIM instruction, the three interrupts RST 7.5, RST 6.5 and RST 5.5 can be masked or unmasked.
- For masking any one of these three interrupts, MSE (i.e. bit D3) bit must be 1.
- For example let RST 7.5 is to be masked (disabled), while RST 6.5 and RST 5.5 are to be unmasked (enabled), then the content of the bits of the SIM instruction will be like

  0000 1100 = 0CH
- For this to be effective the following two instructions are written,

  MVI A, 0CH
  SIM
- Execution of SIM instruction allows copying of the contents of the accumulator into the interrupt masks.

4) RIM(Read Interrupt Mask):

- When RIM instruction is executed in software, the status of SID(to receive serial input data), pending interrupts and interrupt masks are loaded into the accumulator. Thus their status can be monitored.
- It may so happen that when one interrupt is being serviced, other interrupt(s) may occur.
- The status of these pending interrupts can be monitored by the RIM instruction.
- None of the flags are affected by RIM instruction.
- The pattern with bit significant is shown in below figure.

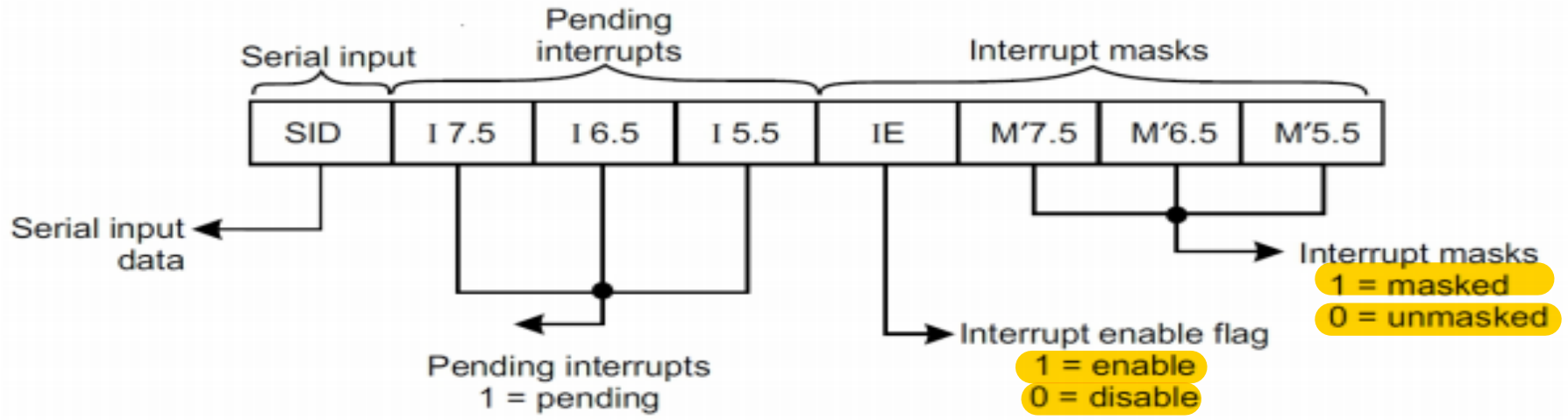


Figure 6.2: RIM instruction format

Interrupt Service Routine (ISR)

Definition

A small program or a routine that when executed, services the corresponding interrupting source is called an ISR.

- When a device interrupts, it actually wants the Microprocessor to give a service which is equivalent to asking the Microprocessor to call a **subroutine**.
- This subroutine is called **ISR** (Interrupt Service Routine)
- The '**EI**' instruction is a one byte instruction and is used to Enable the non-maskable interrupts.
- The '**DI**' instruction is a one byte instruction and is used to Disable the non-maskable interrupts.

Programmable Peripheral Interface 8255A

Introduction to 8255A

- The 8255A is a general purpose programmable, parallel I/O device.
- It is designed to transfer the data from simple I/O to interrupt I/O under certain conditions as required.
- It can be used with almost any microprocessor.
- It consists of three 8-bit bidirectional I/O ports (24 I/O lines) which can be configured as per the requirement.
- It is flexible, versatile and economical but somewhat complex.
- The intent is to provide complete I/O interface in one chip.
- This chip directly interfaces to data bus of the processor.

PIN DIAGRAM:

- 8255 is a 40 pin chip. It has three ports Port A, B and C. Each port has 8 lines. Port A – PA_0 to PA_7, Port B – PB_0 to PB_7 and Port C – PC_0 to PC_7.
- Each pin can be used as input pin or output pin. It can be configured from control word of 8255.

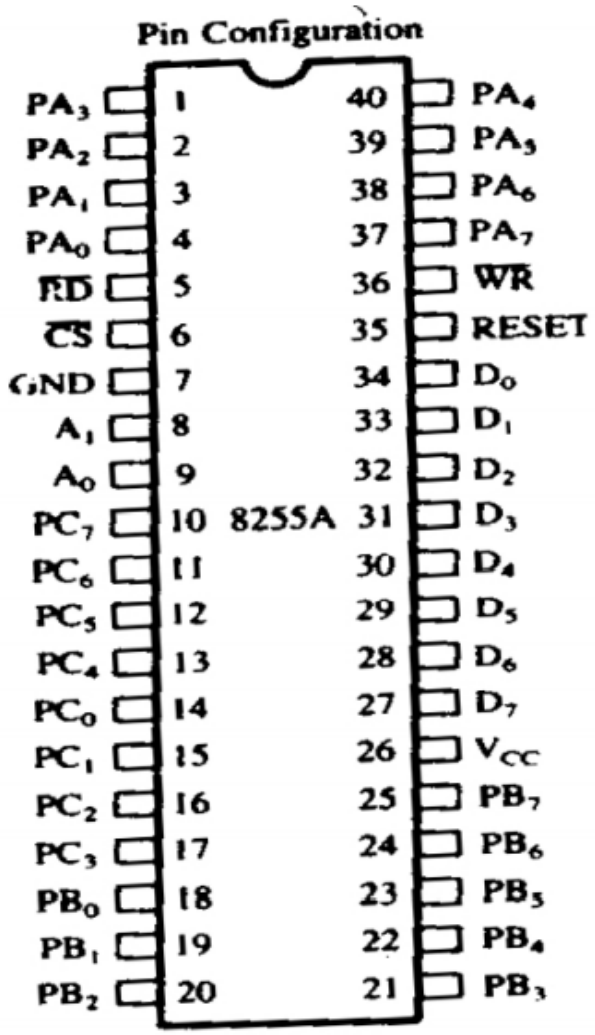


Figure 3.1: Pin Diagram of 8255

- Port C has three functions.
  - It can be used as Simple I/O ports.
  - It can also be used to provide handshake signals
  - It can also be used in Bit Set/Reset (BSR) mode to switch ON or OFF devices.
- Port C can be used as two 4-bit ports also.
- Same way it has D_0 to D_7, these lines are connected to D_0 to D_7 of 8085. These lines bring data to and from 8255.
- There are **SIX** control signals associated with 8255.

1. **$\overline{CS}$** – Chip select signal, it is active low signal, when it is low it enables the 8255.
2. **RESET** – It is used to reset all ports, control word and status word to initial values.
3. **$\overline{RD}$** – It is connected with 8085, whenever it goes low, 8255 will perform Read operation.
4. **$\overline{WR}$** – It is connected with 8085, whenever it goes low, 8255 will perform Write operation.
5. **A_0, A_1** – These are the address lines, based on status of A_0 and A_1, ports will be selected.

A_1	A_0	Port Selection
0	0	Port A
0	1	Port B
1	0	Port C

- There are two power signals.
  - V_{cc} connected to 5 V
  - GND connected to ground.

Ports of 8255A

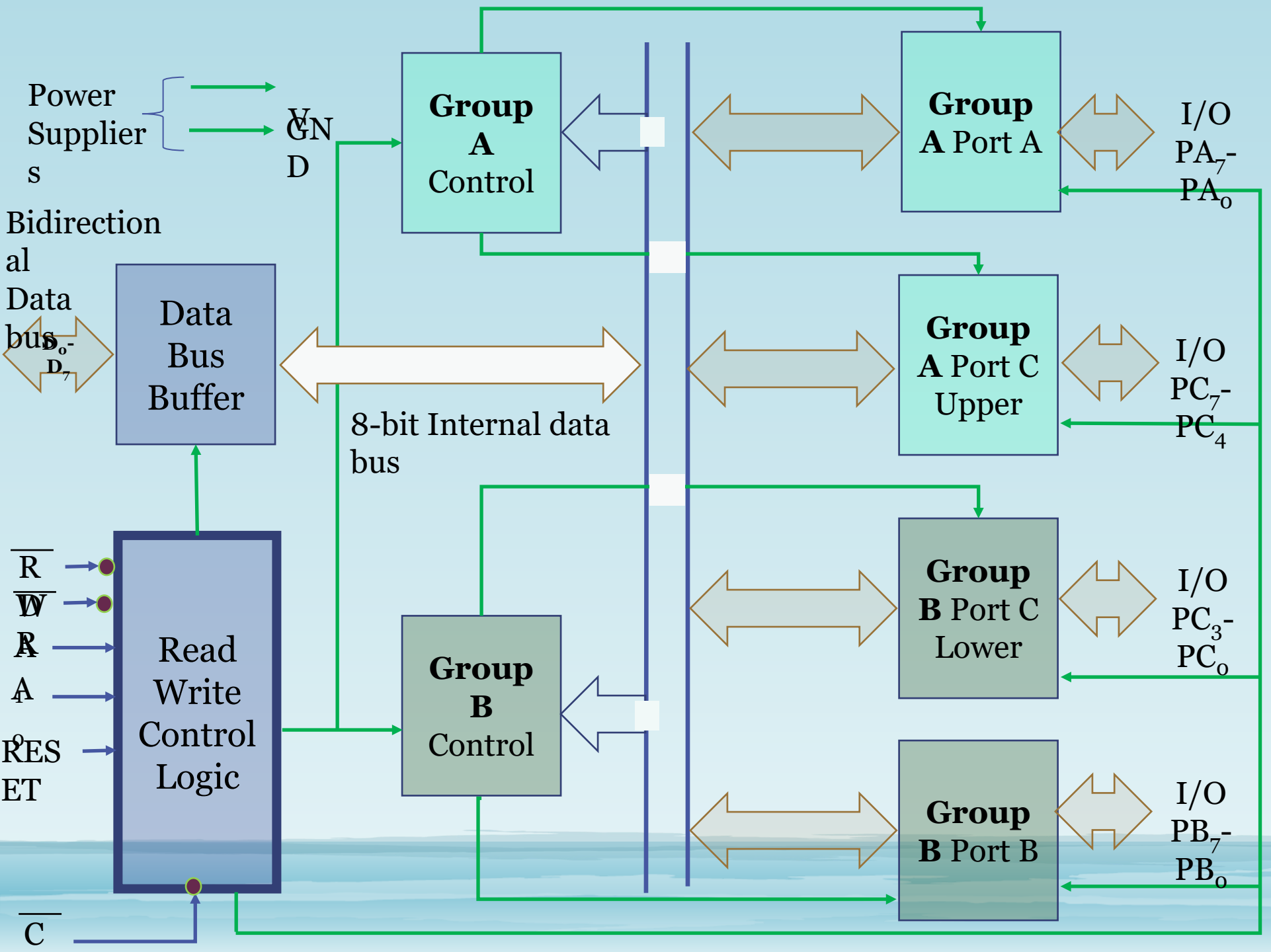
8255A has three ports, i.e., PORT A, PORT B, and PORT C.

- **Port A** contains 8-bit I/O latch/buffer.
- **Port B** is similar to PORT A.
- **Port C** splits into two parts, i.e. PORT C lower (PC_3-PC_0) and PORT C upper (PC_7-PC_4) by the control word.
- These three ports are further divided into two groups,
  - Group A includes PORT A and upper 4-bit of PORT C.
  - Group B includes PORT B and lower 4-bit of PORT C.

▪ These two groups can be programmed in three different modes, i.e. the first mode is named as mode 0, the second mode is named as Mode 1 and the third mode is named as Mode 2.

16

8255A ARCHITECTURE



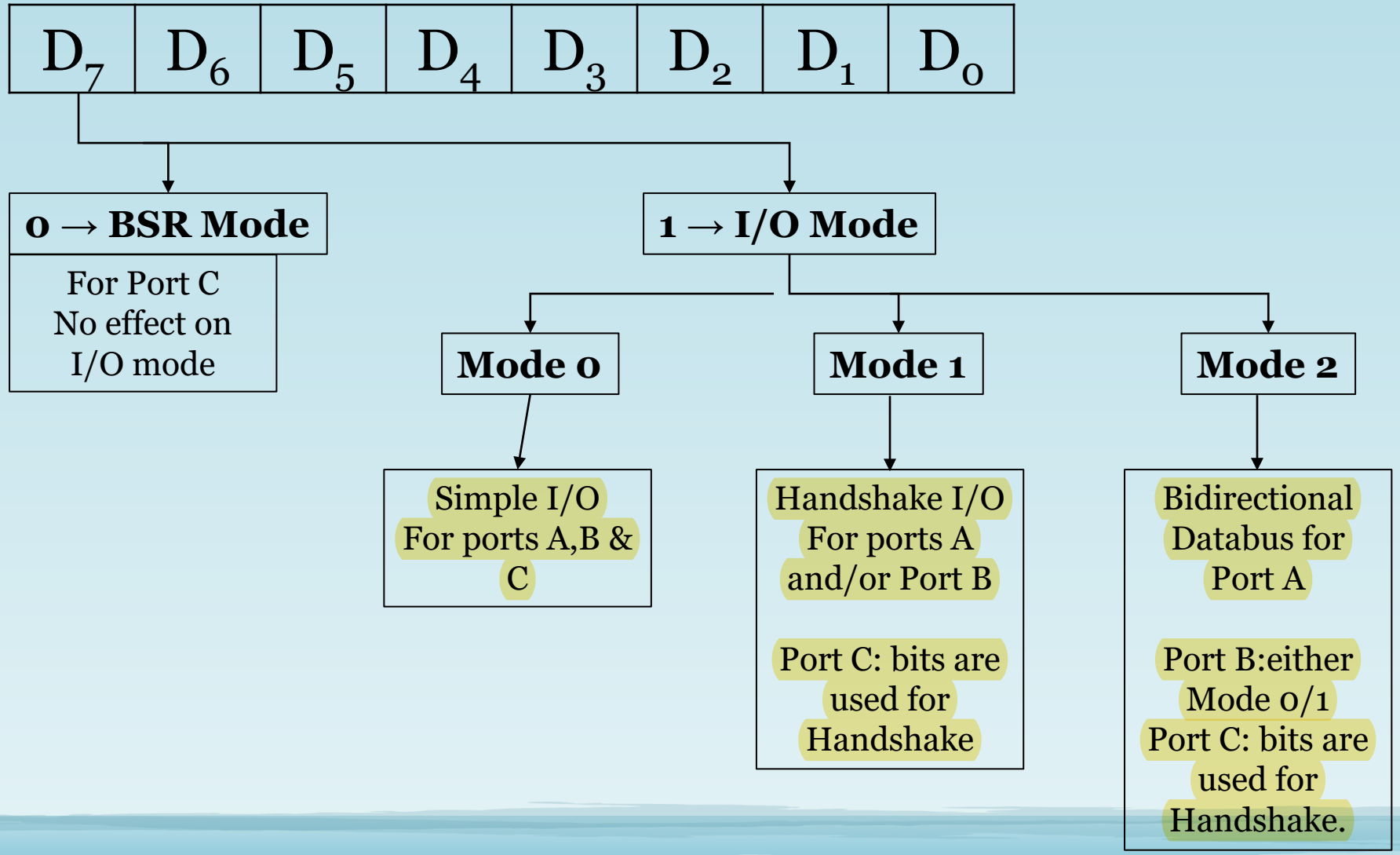
8255A Operating Modes

8255A has three different operating modes:

1. Mode 0
2. Mode 1
3. Mode 2

18

8255A I/O ports and modes



19

8255A Operating Modes

Mode 0

- Simple I/O for port A,B and C
- In this mode, Port A and B is used as two 8-bit ports and Port C as two 4-bit ports.
- Each port can be programmed in either input mode or output mode.
- Ports do not have handshake or interrupt capability.

20

8255A Operating Modes

Mode 1: Input or Output with Handshake

- Handshake signal are exchanged between MPU and peripheral prior to data transfer.
- In this mode, Port A and B are used as 8-bit I/O ports.
- **Mode 1** is a handshake Mode whereby ports A and/or B use bits from port C as handshake signals.
- In handshake mode, 2 types of I/O data transfer can be implemented: status check and interrupt.

21

8255A Operating Modes

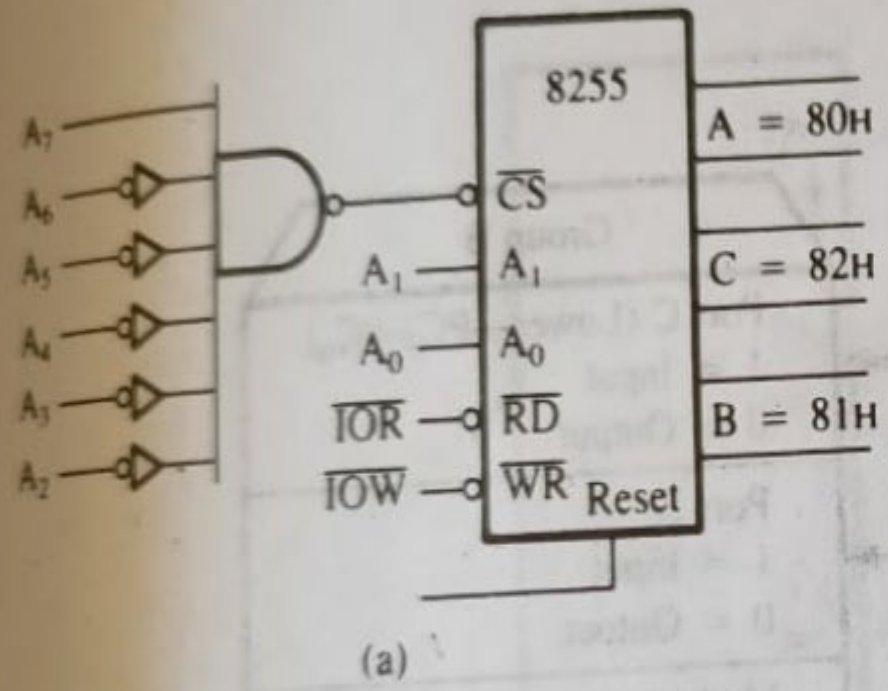
Mode 2

- In this mode, Port A can be configured as the bidirectional port and Port B either in Mode 0 or Mode 1.
- Port A uses 5 signals from Port C as handshake signals for data transfer.
- The remaining three signals from Port C can be used either as simple I/O or as handshake for port B.

22

8255A Architecture: Chip Select

CS	A1	A0	Selected
0	0	0	PORT A
0	0	1	PORT B
0	1	0	PORT C
0	1	1	Control Register
1	X	X	8255A is not selected



(a)

$\overline{CS}$								Hex Address	Port
A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0		
1	0	0	0	0	0	0	0	= 80H	A
						0	1	= 81H	B
						1	0	= 82H	C
						1	1	= 83H	Control Register

(b)

FIGURE 15.3
8255A Chip Select Logic (a) and I/O Port Addresses (b)

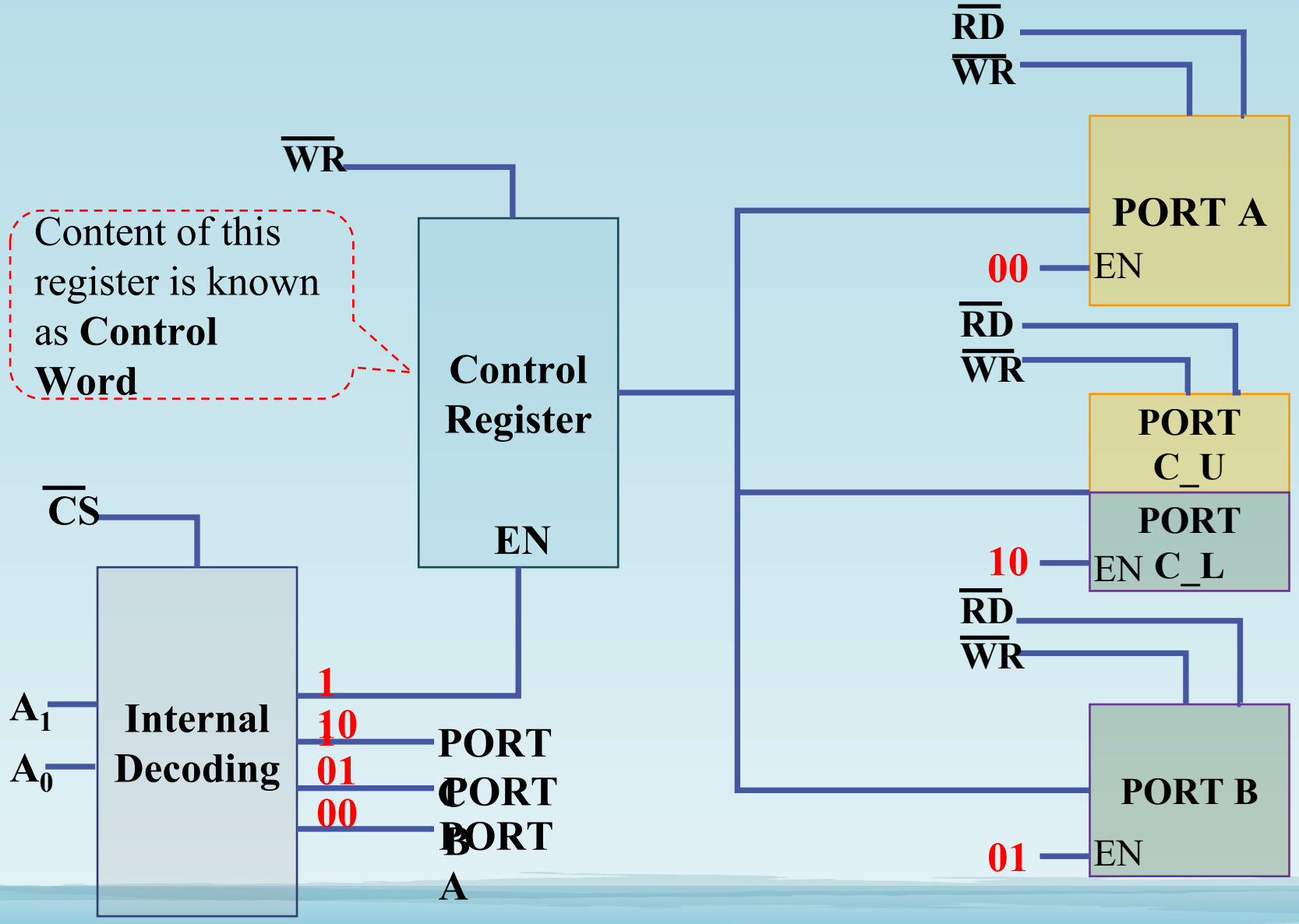
Features of 8255A

The prominent features of 8255A are as follows:

- It consists of three 8-bit IO ports i.e. PA, PB, and PC.
- Address/data bus must be externally demultiplexed.
- It has improved DC driving capability.

25

8255A: Control Register



26

8255A: Control Word

- Accessed to write a control word when A_0 and A_1 are at logic1.

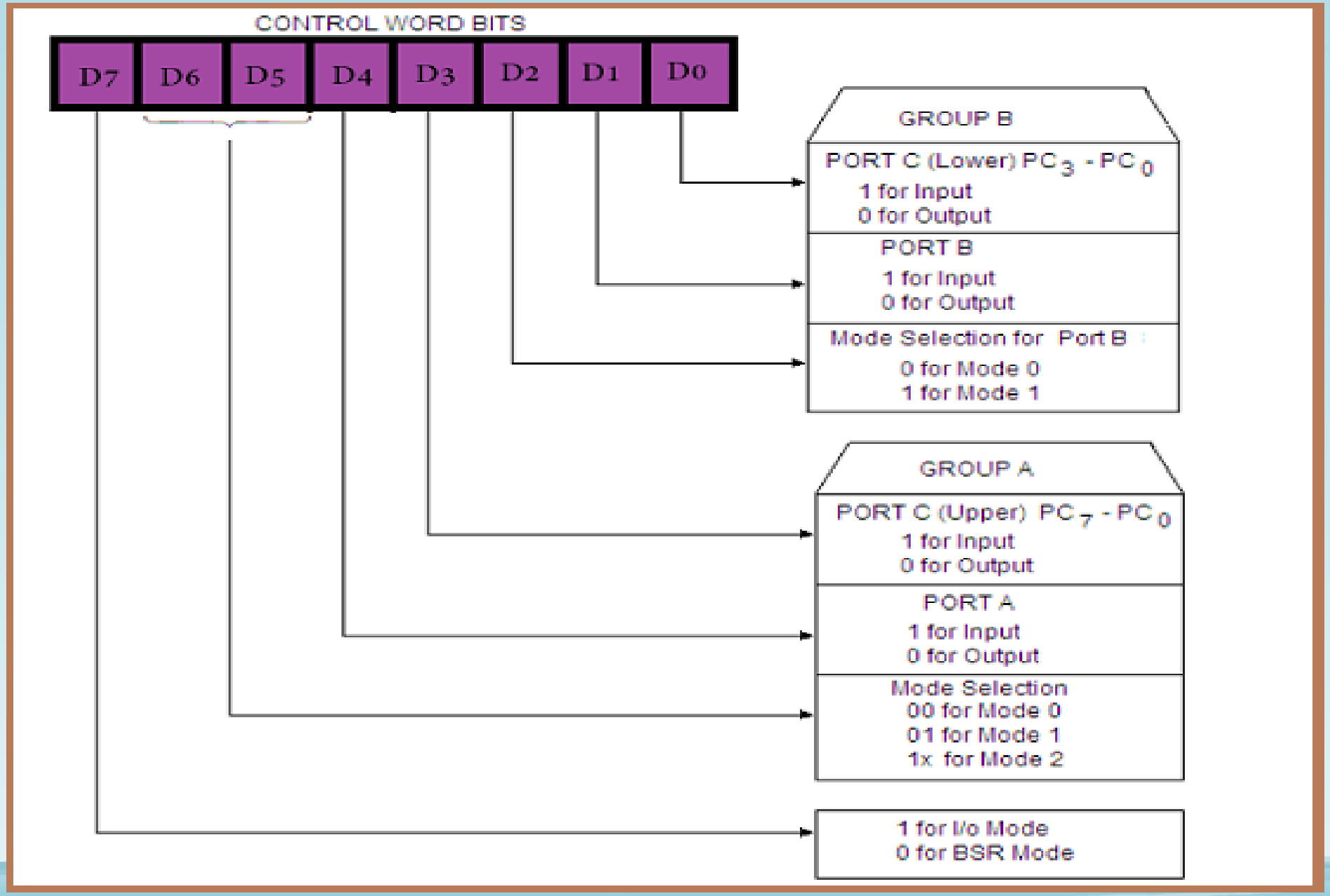
To communicate with peripherals through the 8255A, three steps are necessary:

1. Determine the addresses of ports A, B, and C and of the control register according to the Chip select logic and address lines A_0 and A_1.

2. Write a control word in the control register.

3. Write I/O instructions to communicate with peripherals through ports A, B, and C

27


CONTROL WORD BITS
D7
D6
D5
D4
D3
D2
D1
D0
GROUP B
PORT C (Lower) PC3 - PC0
1 for Input
0 for Output
PORT B
1 for Input
0 for Output
Mode Selection for Port B
0 for Mode 0
1 for Mode 1
GROUP A
PORT C (Upper) PC7 - PC0
1 for Input
0 for Output
PORT A
1 for Input
0 for Output
Mode Selection
00 for Mode 0
01 for Mode 1
1x for Mode 2
1 for I/o Mode
0 for BSR Mode

8255: Mode 0, Example 1

- Configure
  - Port A and port C_U as out port
  - Port B and port C_L as in port
- Interface to Read from I/P DIPs and Display at O/P LEDs
- Control word

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	1	1
I/O function	Port A in Mode 0		Port A as O/P	Port C_U As O/P	Port B in Mode 0	Port B As I/P	Port C_L As I/P

83H

8255A: Control Word

Write a program to read control word from Port B, display at Port A and read upper 4-bit from Port C.

1. **MVI A,D3H** ;Load accumulator with controlword
2. **OUT 83H** ;Write word in control register
3. **IN 81H** ;read switches at Port B
4. **OUT 80H** ;Display the reading at Port A
5. **IN 82H** ; Read at Port C
6. **ANI F0H** ; Mask lower 4-bit

30

1. Identify the port addresses in Figure 15.5.
2. Identify the Mode 0 control word to configure port A and port C_U as output ports and port B and port C_L as input ports.

3. Write a program to read the DIP switches and display the reading from port B at port A and from port C_L at port C_U.

1. Port Addresses This is a memory-mapped I/O; when the address line A_{15} is high, the Chip Select line is enabled. Assuming all don't care lines are at logic 0, the port addresses are as follows:

Port A	=	8000H ($A_1 = 0$, $A_0 = 0$)
Port B	=	8001H ($A_1 = 0$, $A_0 = 1$)
Port C	=	8002H ($A_1 = 1$, $A_0 = 0$)
Control Register	=	8003H ($A_1 = 1$, $A_0 = 1$)

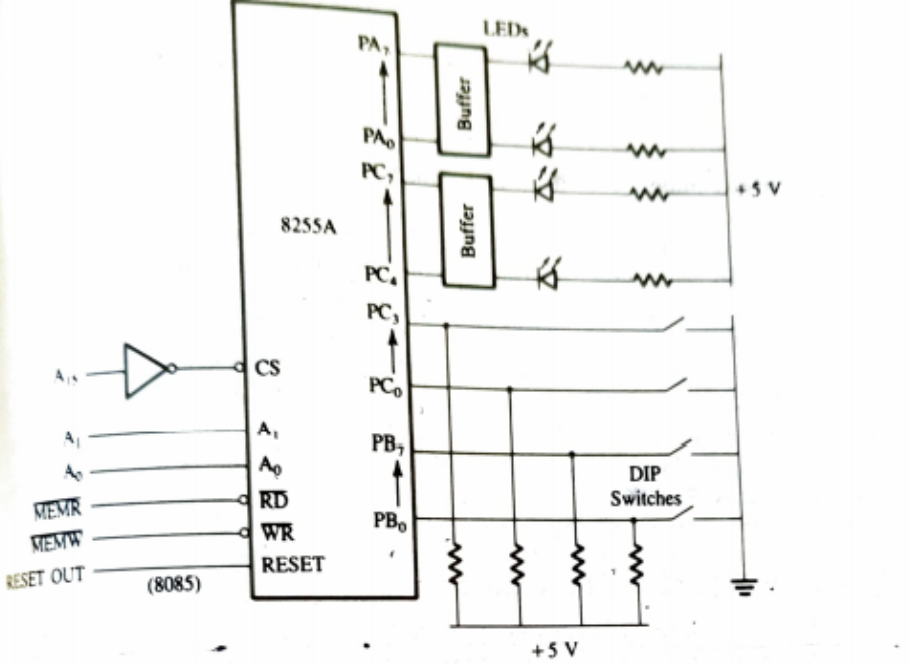
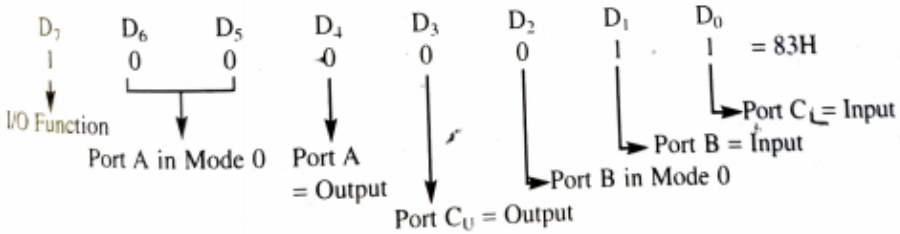


FIGURE 15.5
Interfacing 8255A I/O Ports in Mode 0

Simple IIO.

2. Control Word



3. Program

```
MVI A,83H    ;Load accumulator with the control word
STA 8003H    ;Write word in the control register to initialize the ports
LDA 8001H    ;Read switches at port B
STA 8000H    ;Display the reading at port A
LDA 8002H    ;Read switches at port C
ANI 0FH      ;Mask the upper four bits of port C; these bits are not input data
RLC
RLC          ;Rotate and place data in the upper half of the accumulator
```

```
RLC
RLC
STA 8002H      ;Display data at port CU
HLT
```

Program Description The circuit is designed for memory-mapped I/O; therefore, the instructions are written as if all the 8255A ports are memory locations.

The ports are initialized by placing the control word 83H in the control register. The instructions STA and LDA are equivalent to the instructions OUT and IN, respectively.

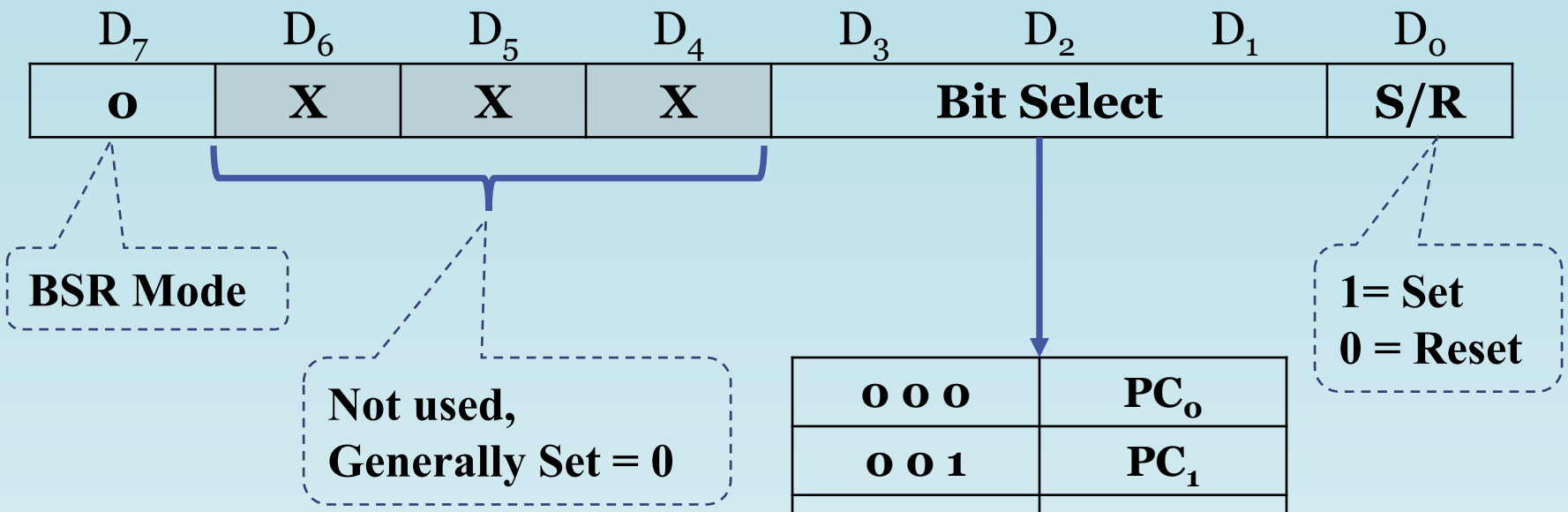
In this example, the low four bits of port C are configured as input and the high four bits are configured as output; even though port C has one address for both halves C_U and C_L (8002H). Read and Write operations are differentiated by the control signals $\overline{MEMR}$ and $\overline{MEMW}$. When the MPU reads port C (e.g., LDA 8002H), it receives eight bits in the accumulator. However, the high-order bits (D_7–D_4) must be ignored because the input data bits are in PC_3–PC_0. To display these bits at the upper half of port C, bits (PC_3–PC_0)

8255A: BSR(Bit Set/Reset) Mode

- In this mode any of the 8-bits of port C can be set or reset depending on D_0 of the control word.
- The bit to be set or reset is selected by bit select flags D_3, D_2 and D_1 of the CWR(Control Word Register).
- BSR Control Word affects one bit at a time.
- Does not affect the I/O mode.

34

8255A: BSR(Bit Set/Reset) Mode



0 0 0	PC_0
0 0 1	PC_1
0 1 0	PC_2
0 1 1	PC_3
1 0 0	PC_4
1 0 1	PC_5
1 1 0	PC_6
1 1 1	PC_7

35

Interfacing of data converters

All the real world quantities are analog in nature. We can represent these quantities electrically as analog signals. An analog signal is a time varying signal that has any number of values (variations) for a given time slot.

In contrast to this, a digital signal varies suddenly from one level to another level and will have only finite number of values (variations) for a given time slot.

This chapter discusses about the types of data converters and their specifications.

Types of Data Converters
The electronic circuits, which can be operated with analog signals are called as analog circuits. Similarly, the electronic circuits, which can be operated with digital signals are called as digital circuits. A data converter is an electronic circuit that converts data of one form to another.

There are two types of data converters −

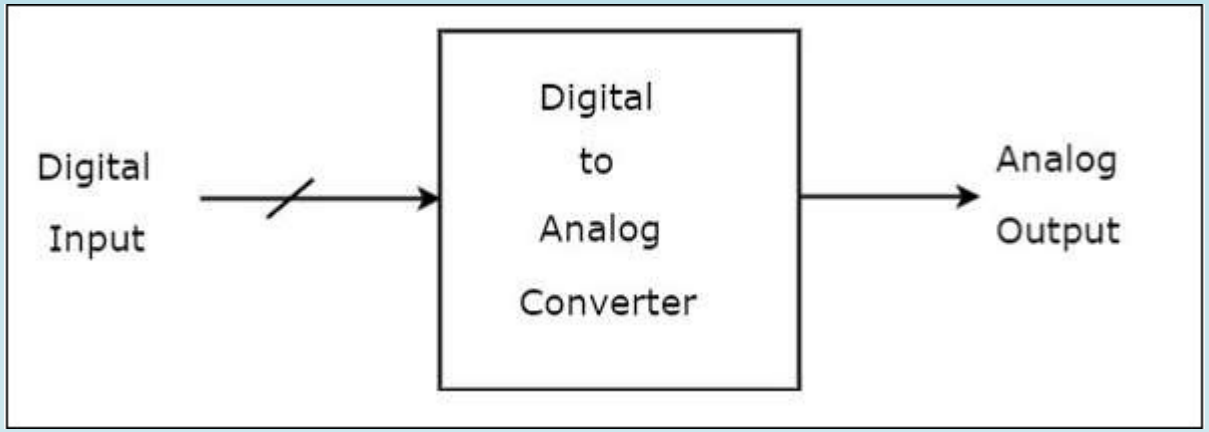
Analog to Digital Converter
Digital to Analog Converter

If we want to connect the output of an analog circuit as an input of a digital circuit, then we have to place an interfacing circuit between them. This interfacing circuit that converts the analog signal into digital signal is called as **Analog to Digital Converter**.

Similarly, if we want to connect the output of a digital circuit as an input of an analog circuit, then we have to place an interfacing circuit between them. This interfacing circuit that converts the digital signal into an analog signal is called as **Digital to Analog Converter**.

Digital to Analog Converter (DAC) converts a digital input signal into an analog output signal. The digital signal is represented with a binary code, which is a combination of bits 0 and 1. This chapter deals with Digital to Analog Converters in detail.

The block diagram of DAC is shown in the following figure −



A Digital to Analog Converter (DAC) consists of a number of binary inputs and a single output. In general, the **number of binary inputs** of a DAC will be a power of two.

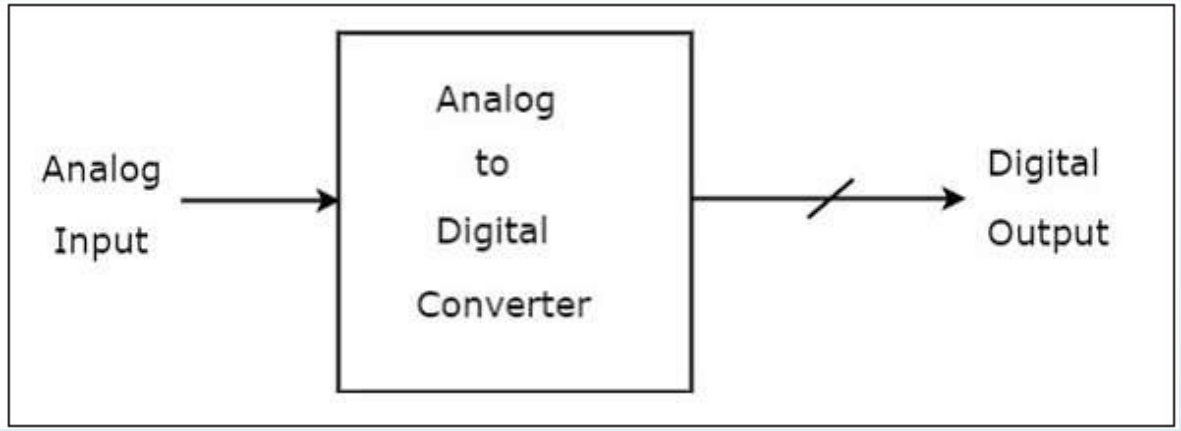
Types of DACs

There are **two types** of DACs

- Weighted Resistor DAC
- R-2R Ladder DAC

An Analog to Digital Converter **(ADC)** converts an analog signal into a digital signal. The digital signal is represented with a binary code, which is a combination of bits 0 and 1.

The **block diagram** of an ADC is shown in the following figure −



Observe that in the figure shown above, an Analog to Digital Converter **(ADC)** consists of a single analog input and many binary outputs. In general, the number of binary outputs of ADC will be a power of two.

There are **two types** of ADCs:

Direct type ADCs and Indirect type ADC.

If the ADC performs the analog to digital conversion directly by utilizing the internally generated equivalent digital (binary) code for comparing with the analog input, then it is called as **Direct type ADC**.

The following are the **examples** of Direct type ADCs −

- Counter type ADC
- Successive Approximation ADC
- Flash type ADC