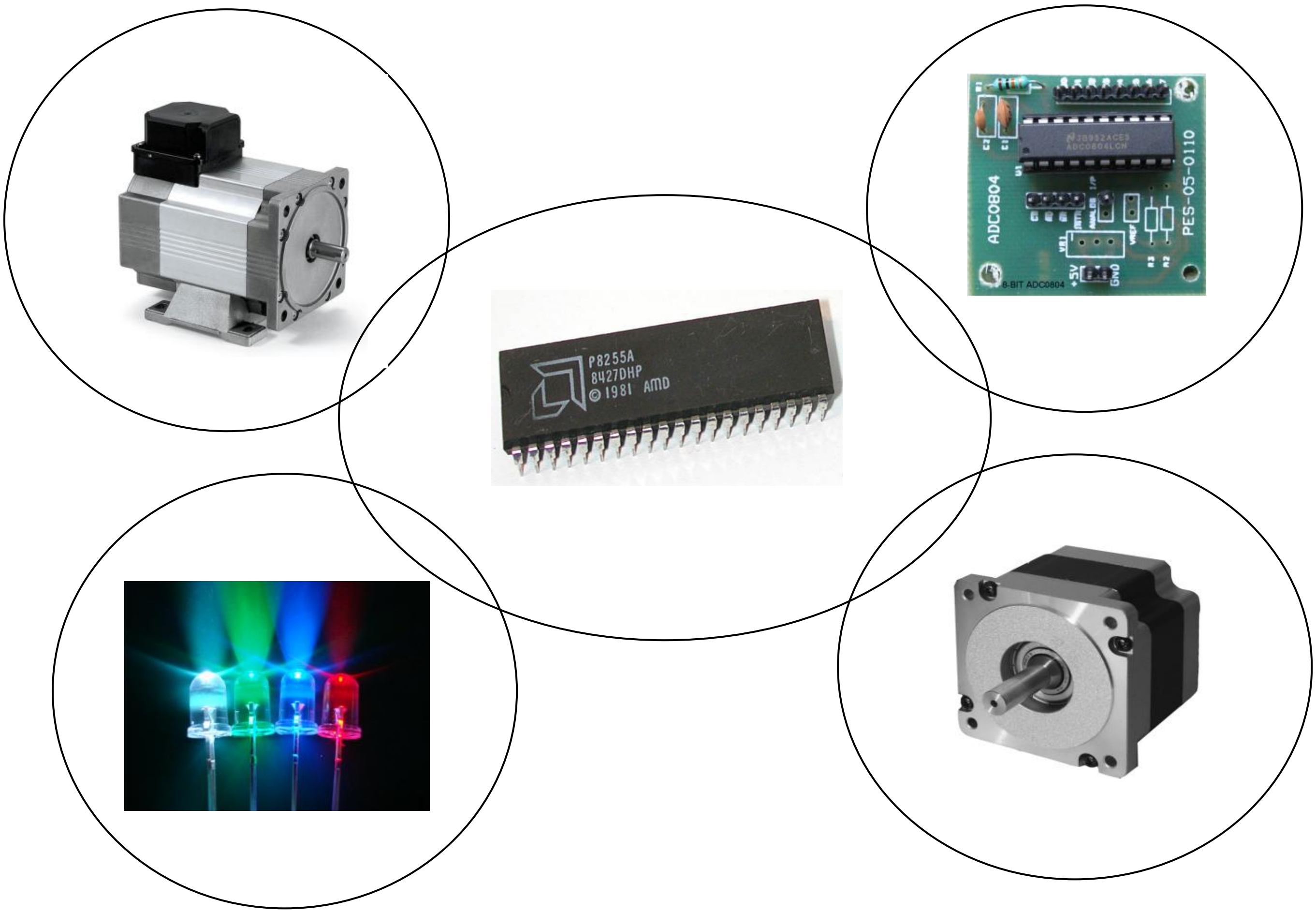
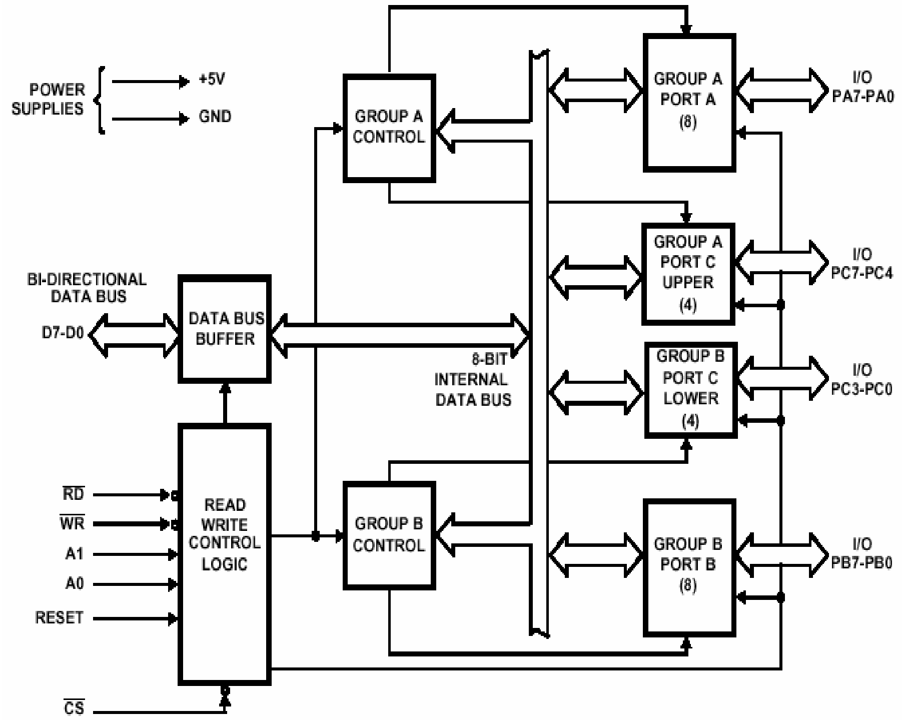


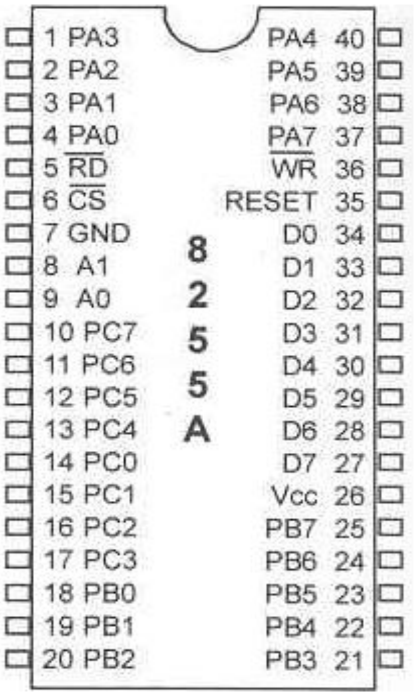
Programmable Peripheral Interface

8255A

Input and output devices, which are interfaced with 8085, are essential in any microprocessor-based-system. They can be interfaced using two schemes- I/O mapped I/O and memory mapped I/O. Both of these schemes has been designed by complex hardware circuit and also it is dedicated for only one type of device, such as Buffer based designed is accomplished only for I/P devices and Latch based designed is accomplished only for O/P devices; both of these design could not be used for reverse action. To reduce this complex hardware circuit and versatility Intel Corporation has introduce 8255 chip, which is commonly called Programmable Peripheral Interface [PPI]. The 8255 has 24 I/O pins divided into 3 groups of 8 pins each. The groups are denoted by port A, port B and port C respectively. Every one of the ports can be configured as either an input port or an output port.

Block diagram and Pin diagram of 8255:





Form the above block diagram it is noticed that any kind of I/O devices could be connected with three ports like Port A, Port B, Port C. due to this features buffer and latch devices are not needed for particular input or output application so, user has flexibility to use same ports as input or output application. Another advantage of this chip, that it can be operated in three different modes which are basically not included in simple I/O interfacing; these different types of operation extended the data transfer policies.

Functions of each block and pins:

Data Bus Buffer- The tri-state bidirectional 8 bit buffer is used to interface the 8255A to the microprocessor data bus (D0-D7). Data is transmitted or received by buffer upon the execution of input or output instructions by the CPU. Control word and status word also transferred through this data bus buffer.

Read write control logic- The function of this block is to manage all the internal and external transfers of both data and control or status word. This block also handles user information regarding operational mode selection, configuration of ports as an input or output; all of this information stored in 8-bit control word register (CWR). The details of each pin connected with this block are described below,

$\overline{CS}$ (Chip Select)- A “Low” on this input pin enables the communication between the 8085 and MPU.

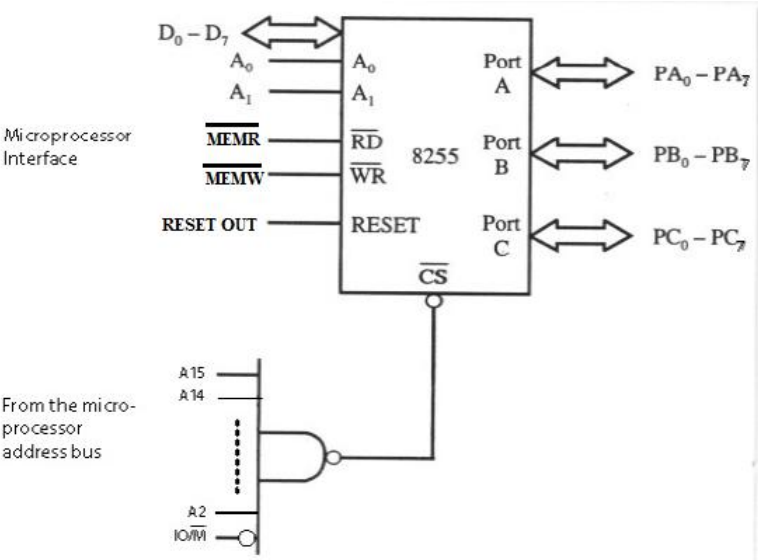
A_0 and A_1 - These are the address lines of 8255 which are directly connected to the MPU lower address lines (A_0, A_1). In conjunction with chip select, control the selection of one of the three ports or the control word register has been made. The bit combination of these signals are shown bellow-

$\overline{\boldsymbol{cs}}$	A_1	A_0	Selection
0	0	0	Port A
0	0	1	Port B
0	1	0	Port C
0	1	1	Control Word Register
1	X	X	8255 is not selected

It is obvious that above these signal is required for interfacing with MPU so, an example is illustrated here-

Q. **Design an interfacing circuit diagram between 8085 and 8255 in memory mapped I/O scheme where the address of port A, B and C are FFFC, FFFD, FFFE respectively also the address of CWR is FFFF. Use partial decoding technique.**

Ans. All ports and CWR address is 16bit and all address bits except A_0 and A_1 are in logic "1" so making a chip select signal using partial decoding technique we used the address bits of A_2 to A_{15} and $IO/\overline{M}$ signal by a NAND gate combination.



$\overline{RD}$ and $\overline{WR}$ - A "LOW" on this pin enable 8255 to send data or status information to MPU via data bus (i.e. read operation) or enable MPU to write data and control word register value to 8255 via data bus (i.e. write operation).

RESET -A "HIGH" on this pin clears the control word register and all ports (A,B,C) are set to input mode.

Group A and Group B controls-

The functional configuration of each port is programmed by the system software. The MPU "outputs" a control word to the 8255 to set some information such as mode, bit-set/reset, etc. that initialize the functional configuration of 8255. Each of control blocks (Group A and Group B) accept commands from read/write control logic, receives "control words" from the internal data bus and issue the proper commands to its associate ports.

Group A control- port A and Port C upper (PC_4 – PC_7)

Group B control – Port B and Port C lower (PC_0 – PC_3)

The control word register can be used for write operation and NO read operation is allowed in this register.

Ports A, B and C-

The 8255A contains three 8-bit ports (A, B and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features to enhance the power and flexibility of the 8255A. The major function of each ports has 8-bit Input or Output, buffer or latch and only Port C can be configured as double 4-bit latch, it is used for control signal output and status signal inputs in conjunction with ports A and B.

8255A Operational Description:

There are three basic I/O modes of operation that can be selected by the system software after properly sets the control word register format as per the requirements.

Mode 0 – Basic Input/ Output

Mode 1 – Strobe or Handshaking Input/ Output

Mode 2 – Bi-directional Bus

In Mode 0 all ports (A, B and C) can be used as 8-bit I/O ports and configured by the control word registers. When the RESET input goes "high" all ports will be set to input mode and after revoked of this signal all ports remain in same mode until any initialization established. In Mode 1 only Port A and B configured as I/O while the upper 4-bit of port C used as strobe signal for port A and lower 4-bit of port C used as strobe signal for port B. Mode 2 is available only for port A while port B can be used as simple I/O mode and bit's of port C used as strobe signal. Except of these three modes of operation 8255A offers single Bit Set/ Reset (BSR) features of port bits, which is limited to port C only. All of these operations are maintained by a 8-bit single register called Control Word Register (CWR). Before using this PPI chip user must be initialize all of these 8 bit ports as input or output with proper modes of operation according to the circuitry where it will be placed. This initialization can be done by CWR register. The format of CWR is shown bellow-

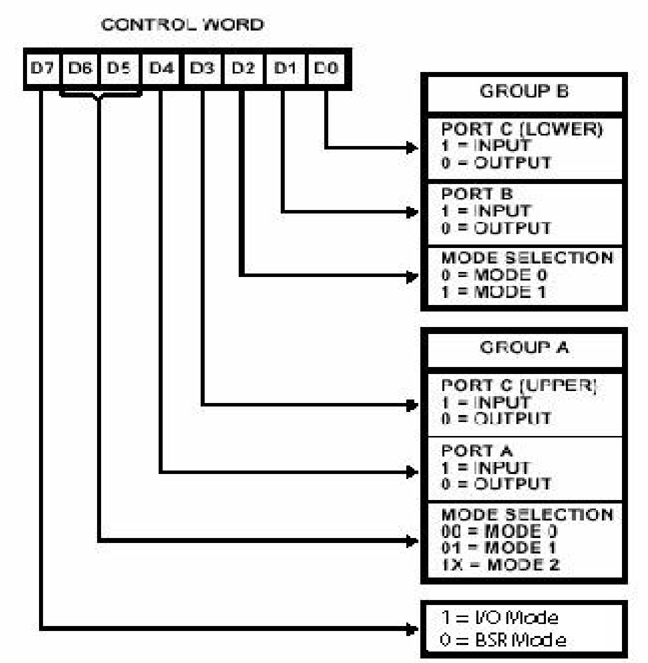


Fig.1 Control Word format in I/O Mode

Mode 0 – Basic Input/ Output

This mode of operation is used for the purpose of simple input or output device connection. To describe its operation we take a practical example; we have an input device like a set of 8 nos. of DIP switch and an output device like a set of 8 nos. of LED. Now we wish to arrange a circuit that can read the switch status (basically ON & OFF) and display it on LED's. Hence you have a question that how it can be done?

At first you have make a circuit using this PPI chip, 8085 microprocessor and your I/O devices. A schematic diagram of this operation are given bellow-

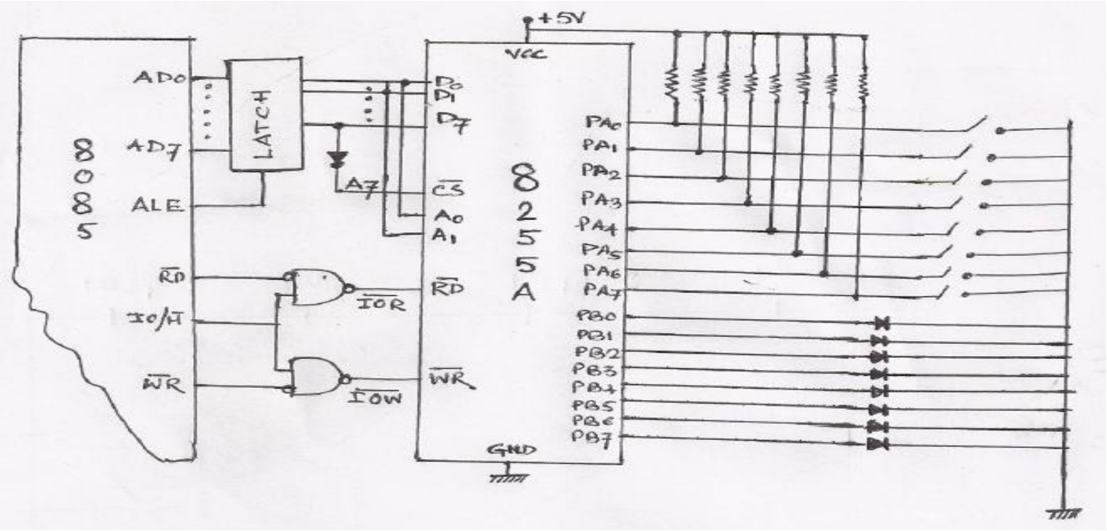
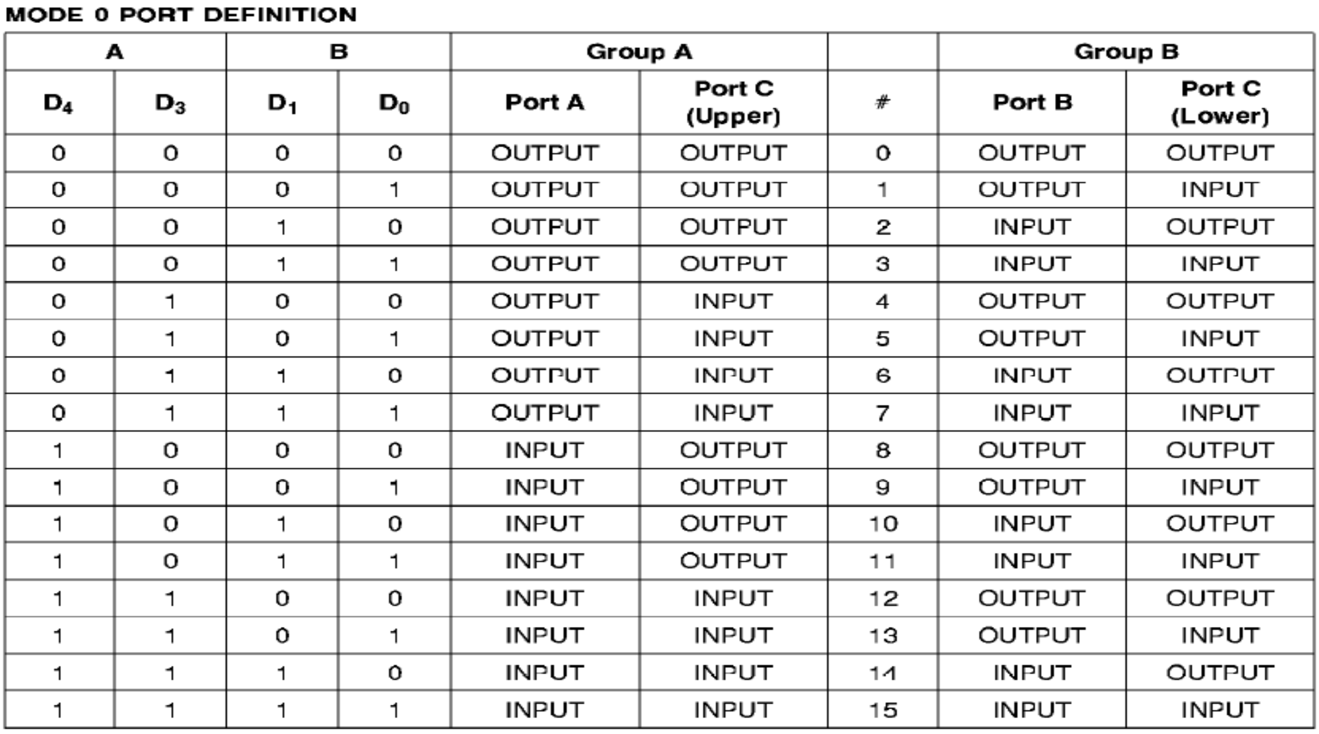


Figure 2. Schematic diagram of Mode-0 example.

Second step of this arrangement is write an ALP to read switch status form 8255 port A and reflect this binary status in LED's which are connected in port B. Before doing this job we have to configure Port A as input and Port B as output with the help of CWR and selecting Mode-0 operation. From fig.1 it is obvious that D_7 bit should be logic "1" because this operation related to I/O with port A, B & C. Bit D_6, D_5 (mode selection bits of Group A) and D_2 (mode selection bit of Group B) must be set at logic 0 to select Mode-0.

Input & output selection bits of different ports is maintain as per application i.e. in this example D_4 is set for port A as input & D_1 is reset for port B as output and the other bit related to port C (D_3 & D_0) make as either set or reset because it is not used here. A list of possible Input & output selection bits of different ports is given bellow-



MODE 0 PORT DEFINITION

A		B		Group A			Group B	
D_4	D_3	D_1	D_0	Port A	Port C (Upper)	#	Port B	Port C (Lower)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

After selection of proper bits in CWR it may show as –

I/O	M1	M0	Port A	Port C_{up}	M0	Port B	Port C_{lw}	
1	0	0	1	0	0	0	0	=90H

Before starting your Assembly Language Program (ALP) to perform the said task you have to know addresses of accessing ports and CWR from the Fig. 3. From this figure and previous knowledge it is obvious that combination of A0 & A1 determines the different address of each ports & CWR but rest of the address bits of 8bit address bus (A0 – A7; it is used for I/O mapped I/O interfacing only) is used for chip select signal of an interfaced chip by applying a suitable decoder or logic gates or direct connection to a single bit. In this schematic, only A7 bit is connected with chip select pin ($\boldsymbol{CS}$) by a NOT gate so, enable this chip A7 bit must be set at logic "1" to make output of NOT gate is low and remaining others bits except A0 & A1 are not used here so we can use them as don't care condition (i.e. set either logic 0 or 1). Hence addresses of different ports are as follows-

A7	A6	A5	A4	A3	A2	A1	A0	ADDRESS	PORT NAME
1	0	0	0	0	0	0	0	= 80H	PORT A
1	0	0	0	0	0	0	1	= 81H	PORT B
1	0	0	0	0	0	1	0	= 82H	PORT C
1	0	0	0	0	0	1	1	= 83H	CWR

Tips for writing a program:

Step1: write the control word register value [which is obtained by proper bit selection of its format] at the finding CWR address.

Step2: use IN or OUT instruction for read or write operation; which is performed by selected ports & address of its ports must be placed at the following instructions. In case of 16 bit address you have to use LDA or STA in replace of IN or OUT.

Note : To set or reset any bits of portC using BSR modes, do not use IN or OUT instruction followed by portC address. For this case always write BSR control word register value at the corresponding CWR address successively.

ALP for that example:

Memory Address	Instruction	Occupied space in memory (byte)	Purpose of this instruction
2000H	MVI A, 90H	2	Selected CWR bit pattern value (hex format) sent in CWR address (83H) by OUT instruction, due to this it is necessary to save this value in Accumulator first by MVI instruction.
2002H	OUT 83H	2	
2004H	IN 80H	2	Read portA bit pattern & save it accumulator by default
2006H	CMA	1	According to the circuit LED’S are connected in common cathode configuration so, turn ON the LED a logic “1” is required & vice versa for OFF. Hence, read bit pattern must be inverted before sending them in output because we want to display logic “1” by glowing the LED. This happens because a pressed switch gives logic “0” & vice versa for unpressed condition.
2007H	OUT 81H	2	Inverted bit pattern (as byte) is by default store in accumulator so it is easy to send this byte in portB by this instruction.
2009H	JMP 2004H	3	For repeatedly handle this situation a unconditional jump has been used by exempting CWR initialization because it is required one time only.
200CH	HLT	1	Due to use of unconditional jump; program terminating instruction is not required here but it is good practice to use this instruction at the end of program.

Mode 1 – Strobed or Handshaking Input/ Output

In Mode-0, 8255A used as a receiver & transmitter to exchange the data in between of microprocessor & input/output devices. But in this scheme exchange data may be lost by the both of devices due to unknown timing of data throwing in between them so, both of the devices should be fully devoted in this throughout process to successfully exchanging the data. This scenario may cause slow performance on device other task execution speed except data exchanging task, it is very effective issue for microprocessor rather than I/O devices task execution policies; where various task are executed serially. To resolve this problem & minimize the data lost rate 8255A made a strobe or handshaking data exchanging facilities into it.

This mode has some advantage over mode-0 respect to the task execution priorities of microprocessor as follows:

1. During exchanging the data with slow devices, µp can perform another task without scanning the port in a timely manner.
2. Data overwrites (during read or write operation) in a port can be resolved by a special status signal.

Port configuration in Mode-1

- Three ports divided into two groups (Group A & Group B); where portC split into two parts for the following groups.
- Each group contains one 8-bit data port & one 4-bit control port
- 8-bit data port can be used as input or output. Both of the cases data are latched.
- The 4-bit port is used for control or status of the 8-bit port.
- After selection of two groups port bits, remaining bits of portC may used as a simple input or output.

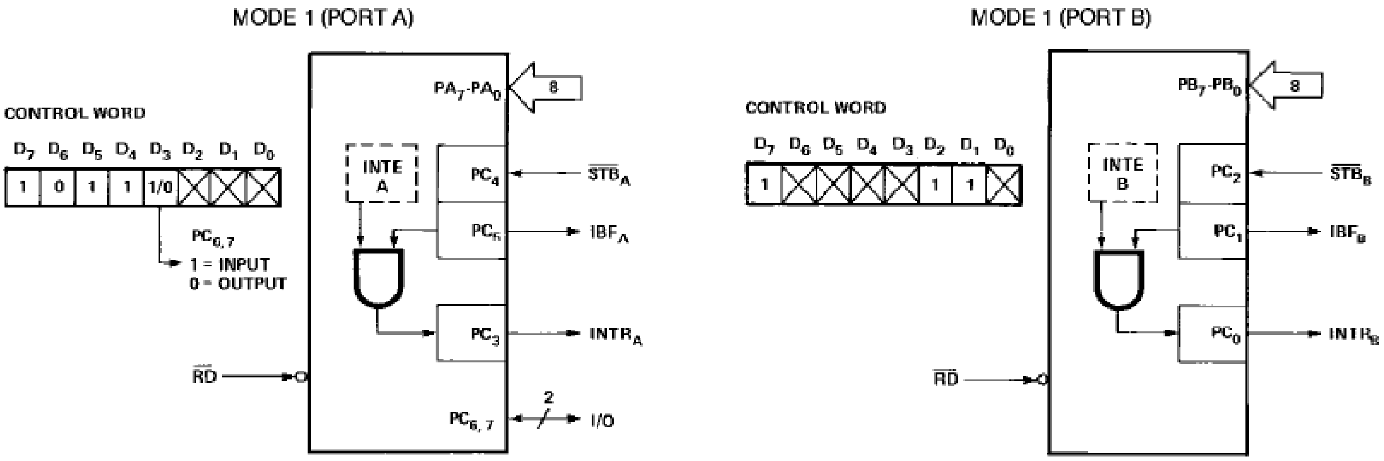


Figure 3. Mode -1 input port configuration

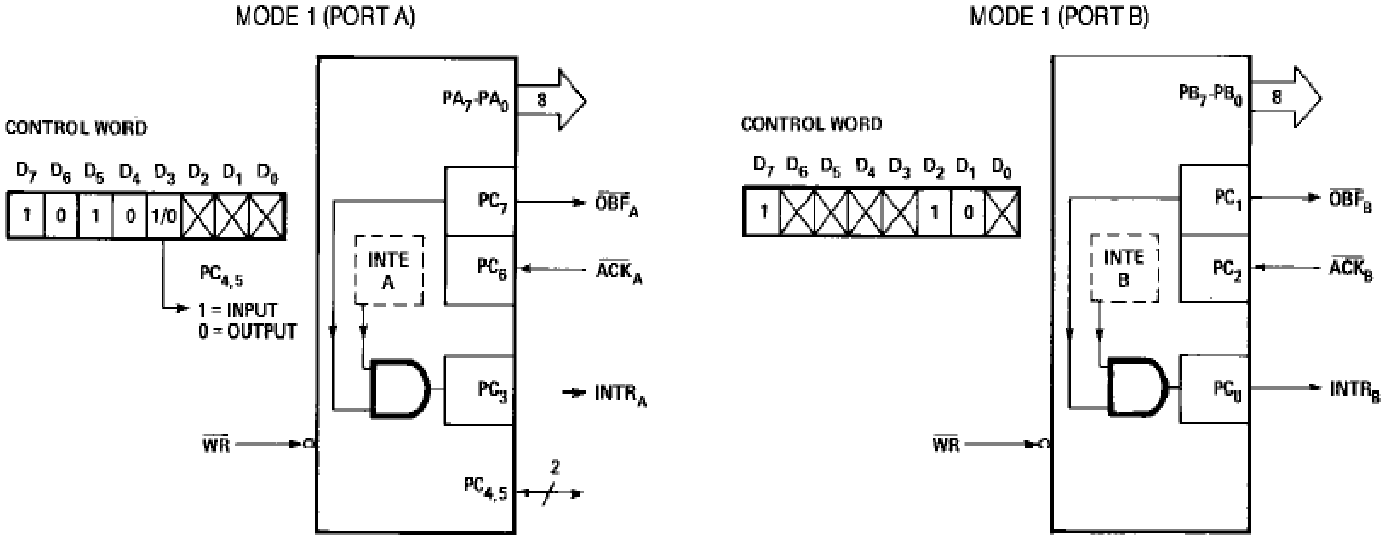


Figure 4. Mode -1 output port configuration

Input control signal definition:

$\overline{STB}$ (Strobe Input) : a "Low" on this input denotes that an input device loads data into the input latch (portA or portB) & after completing its task this signal remain in "High".

IBF (Input Buffer Full) : it is an output of a flip flop ("High") indicated to the input devices that input latch is now full & no data cannot be received yet until µp reads the data from latch. It will reset after read the data by CPU & thereafter input devices can send data again.

INTR (Interrupt Request) : it is an output signal ("High") used for interrupt the CPU to read the received data from input latch. This signal occurred when $\overline{STB}$**,** IBF & INTE are all in logic "High". After getting this signal CPU finishes its current execution & generates a $\overline{RD}$ signal to read the corresponding data & it is reset by falling edge of $\overline{RD}$.

INTE$_A$ & INTE$_B$ is controlled by bit set/reset of PC$_4$ & PC$_2$ respectively.

Output control signal definition:

$\overline{OBF}$ (Output Buffer Full) : a "Low' output of a flip flop indicates to the output devices that output latch (portA or portB) is now filled by data & no new data cannot be write yet by µp until output devices reads the data from latch. It will set after read the data by output devices, when it makes an acknowledgement signal.

$\overline{ACK}$ (Acknowledge Input) : a "Low" on this input pin denotes that an output device read the data from the output latch & after completing its read operation this signal remain in "High".

INTR (Interrupt Request) : it is a active "High" output signal used for interrupt the CPU to write the new data into the latch. This signal occurred when $\overline{\mathrm{OBF}}$**,** $\overline{\mathrm{ACK}}$ & INTE are all in logic "High". After getting this signal CPU finishes its current execution & generates a $\overline{WR}$ signal to write a new data & it is reset by falling edge of $\overline{WR}$.

INTE$_A$ & INTE$_B$ is controlled by bit set/reset of PC$_6$ & PC$_2$ respectively.

To illustrate this mode of operation let us assume an example of courier service role in a business, where you are the master of your business members & one day you wish to send a packet containing some raw material to another member by a third party courier service agency. So what happens in the process of transferring your material to the destination? Most likely you guess the following steps;

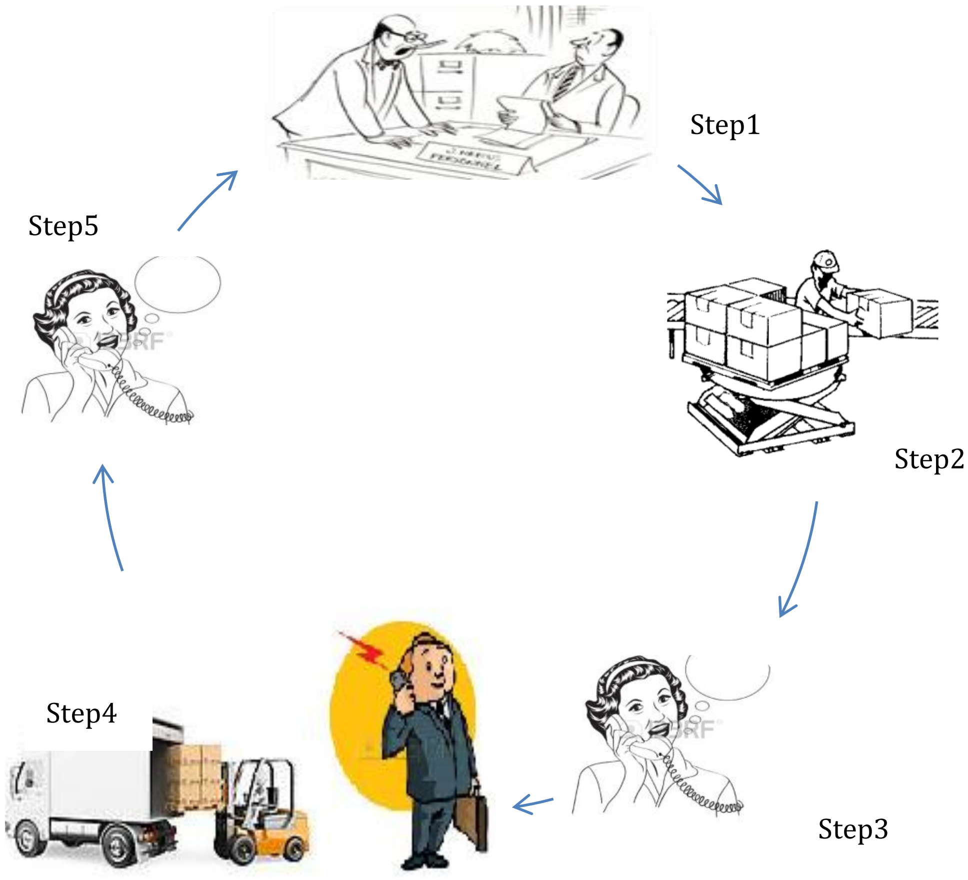


Figure 5. Representation of output control signal cycle

Step1: courier agency wants to know you that place of delivery; it seems like initialization of control word by µp into the 8255 to indicate the receiving & transmitting ports.

Step2: after mention the destination you place your material to the agency counter; likely to write data into the data bus. After that you may engage your another task until your agency informed you about the status of your delivery.

Step3: during this period your business partner waits for a call from courier agency; it is similarly the handshaking signal of $\overline{OBF}$**.**

Step4: when your partner wakeup by this call he/she take initiate to receive this material from specified delivery location & after getting this material he/she inform the agency that material has been pickup successfully. It is same as acknowledge input ($\overline{ACK}$) provided by output device to 8255.

Step5: after getting acknowledgement from your partner courier agency informed you that your parcel has been delivered successfully & you may send new material to your preferred one. It is similarly to the interrupt request signal used for wakeup the CPU.

Timing diagram of mode-1 control signal for input & output operation are given bellow-

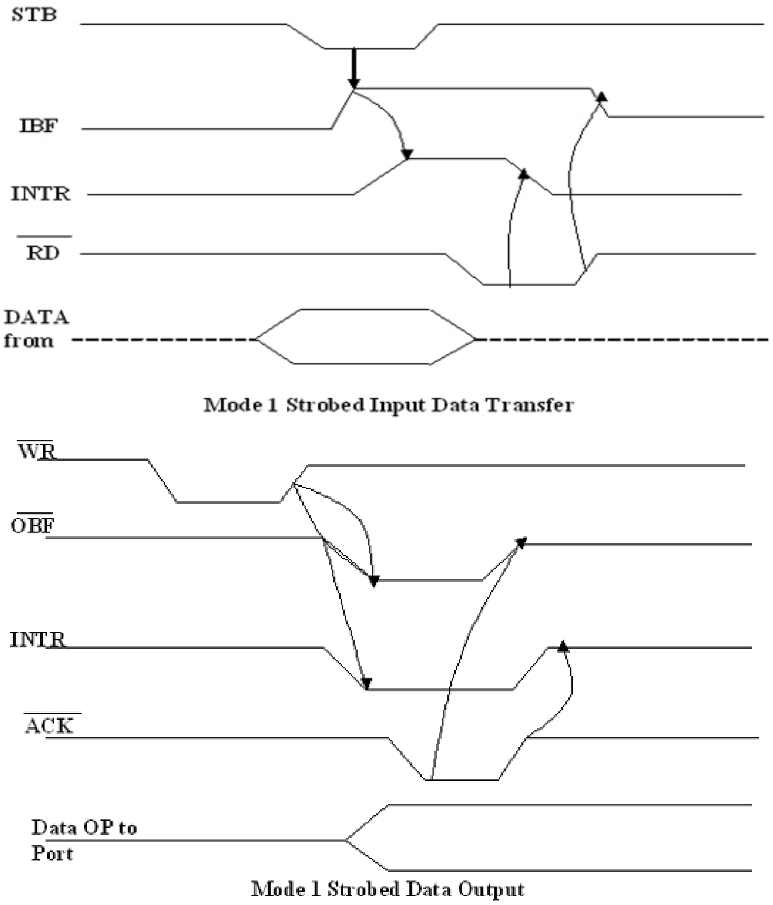
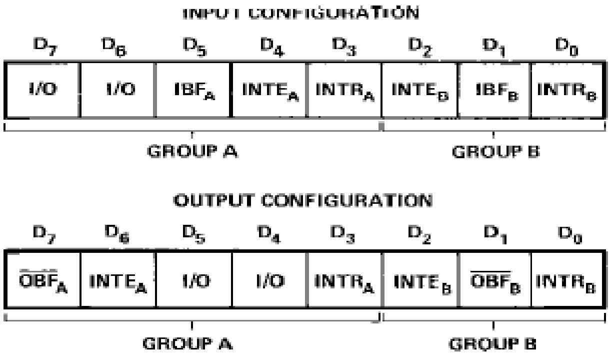


Figure 6. Timing diagram of Mode -1 control signal

From the above discussion it is obvious that CPU can read or write data from the input or output port either through interrupt control or through program controlled I/O. in interrupt control, CPU is usually busy in other system operation & it respond for read or write operation when interrupted. In program control, CPU is truly involved in read or writes operation & other system operation is still suspended. CPU performed this read or write operation by continuously monitoring the status word, which is basically represents port-C control bits used in mode-1 operation. The corresponding status word is accessed by reading the portC with both $\overline{RD}$ & $\overline{CS}$ LOW and combination of A1, A0 = 10. Fig.8 shows the format of status word for input & output configuration.



INPUT CONFIGURATION

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
I/O	I/O	IBF_A	$INTE_A$	$INTR_A$	$INTE_B$	IBF_B	$INTR_B$
GROUP A					GROUP B		

OUTPUT CONFIGURATION

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$\overline{OBF}_A$	$INTE_A$	I/O	I/O	$INTR_A$	$INTE_B$	$\overline{OBF}_B$	$INTR_B$
GROUP A					GROUP B		

Figure 7. Status word of Mode -1 input & output configuration

Mode 2 – Strobed Bidirectional Input/ Output Bus

PortA & PortB individually defined as an input or output port with their handshaking signal in mode-1 to support wide variety of I/O applications. But mode-2 configuration provides a single 8-bit bus for transmitting or receiving (is called bidirectional) data with handshaking signal to maintain the proper flow discipline in a similar manner of mode-1. Interrupt generation and enable/disable functions are also available in this mode.

Port configuration in Mode-2

- Only portA is used as a 8-bit bi-directional I/O bus.
- Handshaking signal for communication is provided by 5-bits of portC [PC3 to PC7].
- PC0 to PC2 is used as a simple input or output, is set by mode-0 operation.
- PortB can be programmed either in mode-0 or mode-1 configuration with PC0-PC2 used as a handshaking signal where it is applicable.

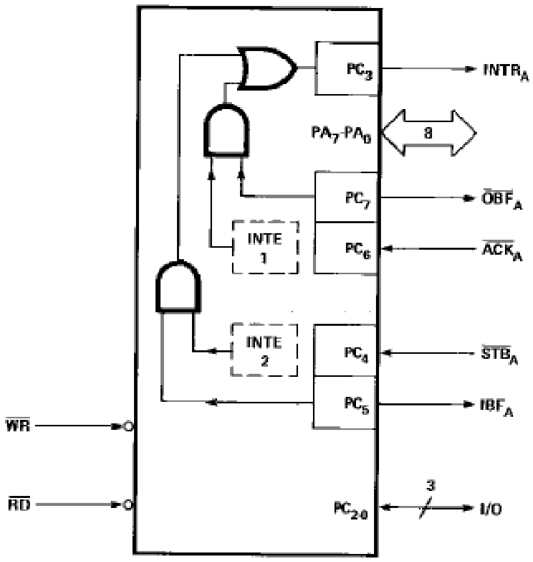


Figure 8. Port configuration of Mode-2 operation

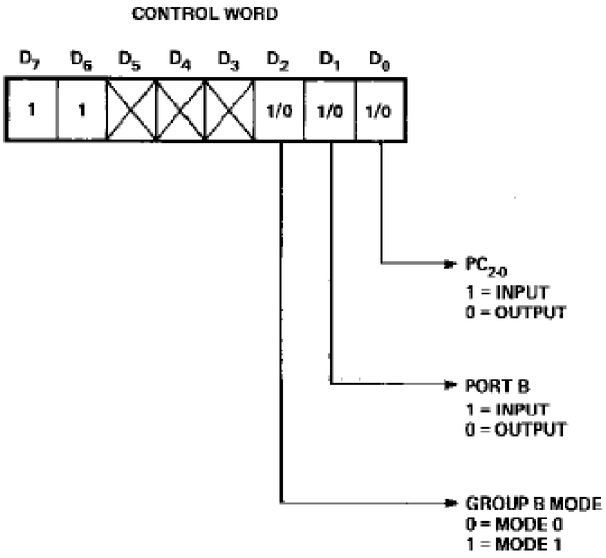


Figure 9. Mode-2 control word structure

For both input and output operations, the interrupt are generated on the same PC3 (INTR$_A$) line. The input or output control signal of this mode is described below:

Input control signal definition:

$\overline{STB}$ (Strobe Input) : This active LOW input signal is used to enable portA latch to loads data into it & after completing its task this signal remain in “High”.

IBF (Input Buffer Full) : it is an output of a flip flop (“High”) indicated to the peripheral devices that input latch is now full & no data cannot be received yet until µp reads the data from latch. It will reset after read the data by 8085 & thereafter input devices can send data again.

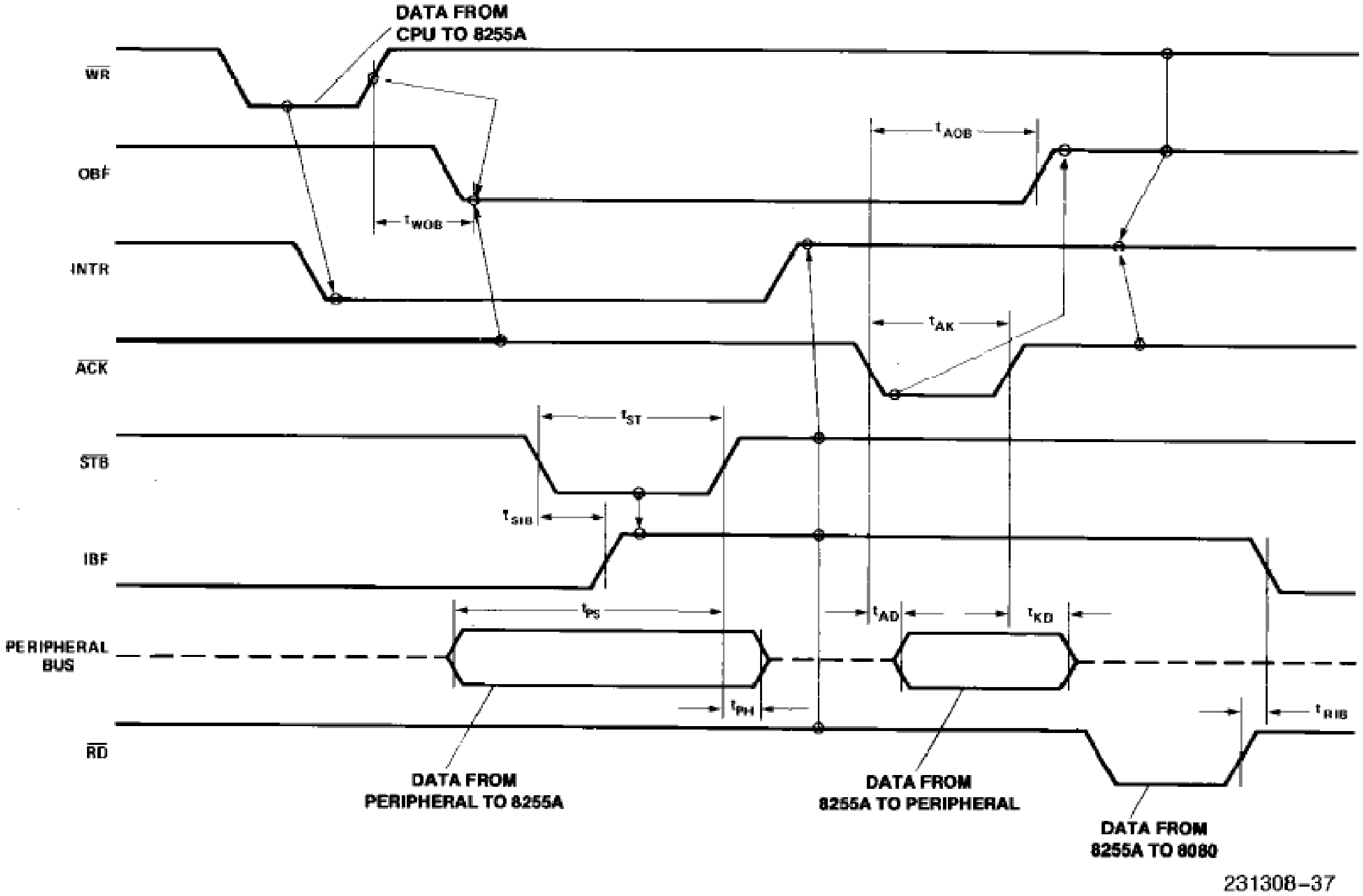
INTE$_1$: it is an interrupt enable flip-flop and controlled by set/reset of PC$_6$

Output control signal definition:

$\overline{OBF}$ **(Output Buffer Full) :** active "Low' output signal indicates to the peripheral devices that output latch (portA) is now filled by data & no new data cannot be write yet by µp until peripheral devices reads the data from latch. It will set after read the data by output devices, when it makes an acknowledgement signal.

$\overline{ACK}$ **(Acknowledge Input) :** a "Low" on this input pin denotes that peripheral device read the data from the output latch & after completing its read operation this signal remain in "High".

INTE$_2$: it is an interrupt enable flip-flop and controlled by set/reset of PC$_4$



NOTE:
Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
(INTR = IBF • $\overline{MASK}$ • STB • RD + $\overline{OBF}$ • $\overline{MASK}$ • ACK • WR)

Figure 10: Timing diagram of the bidirectional data transfer control signals

Interrupt control signal :

INTR (Interrupt Request) : it is a active "High" output signal used for interrupt the CPU to write the new data into the latch or read the received data from the latch. Write operation initiate when $\overline{\text{OBF}}$, $\overline{\text{ACK}}$ & INTE are all in logic "High" and read operation initiate when $\overline{\boldsymbol{STB}}$, IBF & INTE are all in logic "High". After getting this signal CPU finishes its current execution & generates a $\overline{WR}$ signal to write a new data or $\overline{RD}$ signal to read the received data. This signal goes low after the falling edge of $\overline{RD}$ or $\overline{WR}$ signal.

Like mode-1 operation, a programmer can verify the status of peripheral devices by reading the portC with both $\overline{RD}$ & $\overline{CS}$ LOW and combination of A1, A0 = 10. Fig.11 shows the format of status word for Mode-2 operation.

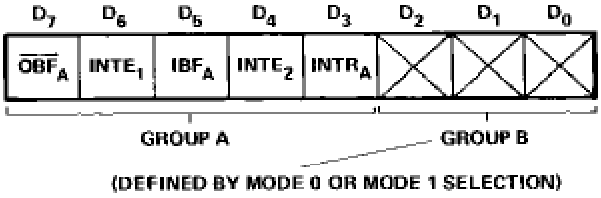
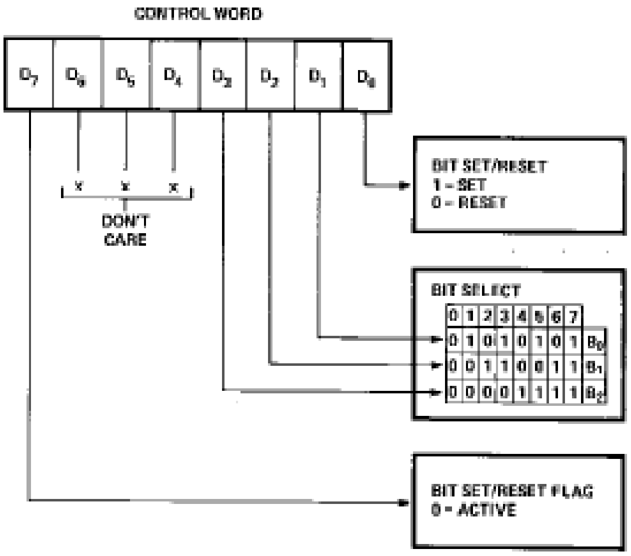


Figure 11. Status Word format in Mode-2 operation

8255A single Bit set/reset (BSR) features:

Rather than the I/O modes of operation 8255A has special bit controlling features. In this mode of operation, any of the 8-bits of portC can be set or reset using a single OUT instruction. This feature reduces software requirements in ON/OFF control applications.

This mode is enabled by resetting the D7 bit of control word register (CWR) & after choosing this bit the format of CWR is completely changed from I/O mode structure. The format of CWR in BSR mode is shown bellow –

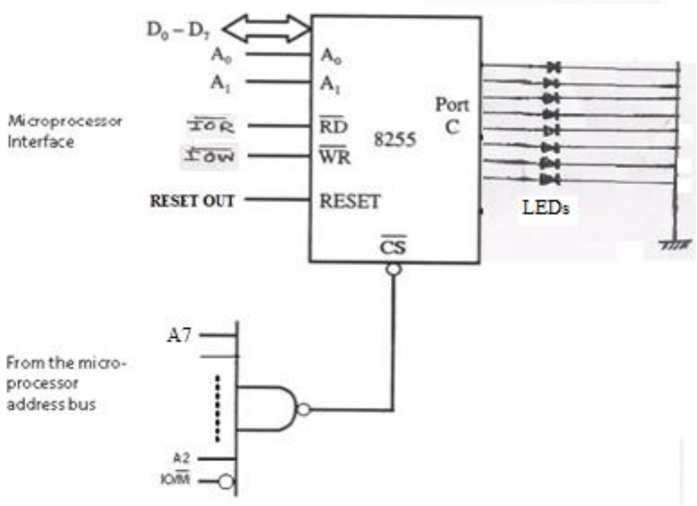


0	1	2	3	4	5	6	7	
0	1	0	1	0	1	0	1	B_0
0	0	1	1	0	0	1	1	B_1
0	0	0	0	1	1	1	1	B_2

Figure 12. Control Word format in BSR Mode

Any of the 8-bit of portC is selected by the combination of group of bits (D_3, D_2, D_1) as shown in fig.11 and selected portC bit is control by D_0 bit of CWR to ensure set or reset operation of the following bit. To illustrated this operation an example is provided bellow –

Let us assume that you have to blink eight LEDs one after another at an interval of 100ms using portC of 8255A. The schematic diagram of this objective is shown bellow.



A7	A6	A5	A4	A3	A2	A1	A0	ADDRESS	PORT NAME
1	1	1	1	1	1	0	0	= FCH	PORT A
1	1	1	1	1	1	0	1	= FDH	PORT B
1	1	1	1	1	1	1	0	= FEH	PORT C
1	1	1	1	1	1	1	1	= FFH	CWR

Address decoding of this schematic diagram

Figure 12. Schematic diagram of BSR Mode example & address decoding from interface

To turn ON & OFF the LEDs, binary “1” & “0” is needed because all LEDs are connected in common cathode configuration. An ALP is provided bellow to meet this objective utilizing two modes of operation.

Using BSR Modes			Using simple I/O Mode		
Memory Address	**Instruction**	**Purpose of this instruction**	**Mem. Add.**	**Instruction**	**Remarks**
2000_H	MVI C,08_H	Load counter value for 8bit rotation	2000_H	MVI C,08_H	Load counter value for 8bit rotation
2002_H	MVI A, 01H	Initialize Accumulator by a value for turn ON first D0 bit of portC according to the CWR format & send it in CWR address. [**not portC address**]	2002_H	MVI A,80_H	Initialize accumulator by a value for making portC as output in I/O mode operation & send it in CWR address.
2004_H	OUT FF_H		2004_H	OUT FF_H	
2006_H	CALL 3000_H	A 100ms delay program already written in 3000H location, we just call it & return in next line.	2006_H	MVI A,01_H	Initialize Accumulator by a value for turn ON first D0 bit of portC & send it in **portC address**.
			2008_H	OUT FE_H	
2009_H	INR A	For programming optimization, double increment is made hare to set next D1 or consecutive bits by following CWR format. <u>Note:</u> after any new CWR initialization, previous condition will be discard automatically.	$200A_H$	CALL 3000_H	A 100ms delay program already written in 3000H location, we just call it & return in next line.
			$200D_H$	RLC	For programming optimization, a left rotation is made hare to set next D1 or consecutive bits.
$200A_H$	INR A				
$200B_H$	DCR C	After elapse of 100ms new CWR value will be send & before doing this a counting value is decremented by one and check whether it is reached to zero or not; if not send new value otherwise follow next step	$200E_H$	DCR C	After elapse of 100ms new CWR value will be send & before doing this a counting value is decremented by one and check whether it is reached to zero or not; if not send new value otherwise follow next step
$200C_H$	JNZ 2004_H		$200F_H$	JNZ 2000	
$200F_H$	JMP 2000_H	This process is repeatable so, an unconditional jump is used here to repeat this program from beginning.	2012_H	JMP 2000_H	This process is repeatable so, an unconditional jump is used here to repeat this program from beginning.

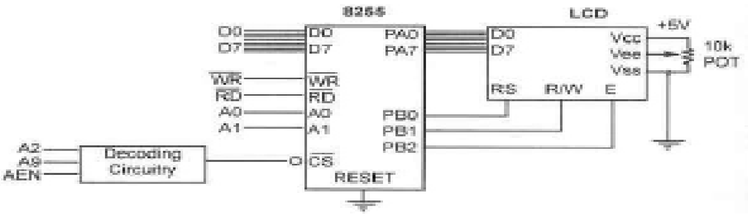
A comparative ALP programming is depicts the simplicity of BSR modes respect to the I/O modes of operation. Major confusing part of these two modes of operation is accessing the address of CWR and portC which is clearly mentioned in above two comparative statements by following “bold” text line.

Test Problem

i) Interface an 8 bit ADC 808 to port A. Derive control signals from port C. Write an ALP to read an analog signal & save data in reg.C.

ii) Interface an 8 bit DAC 08 to port A. Write an ALP to generate a ramp signal with a slope of 1V/s.

iii) Interface 16 ch x 1Line LCD to port A. Derive control signals from port C. Write an ALP to flash “**WELCOME TO MCET”.**

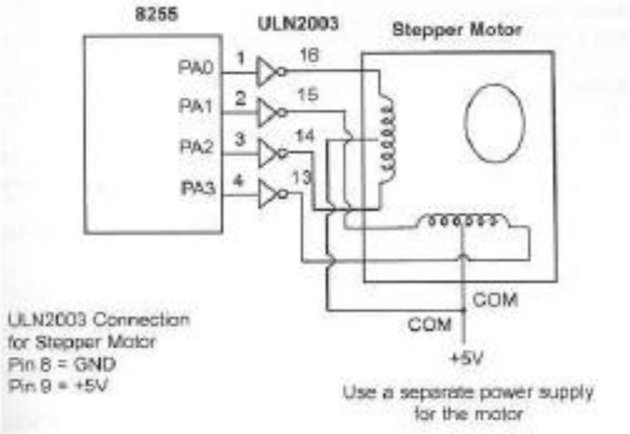


1. The LCD's data pins are connected to Port A of the 8255.
2. The LCD's RS pin is connected to PB0 of Port B of the 8255.
3. The LCD's R/W pin is connected to PB1 of Port B of the 8255.
4. The LCD's E pin is connected to PB2 of Port B of the 8255.
5. Both Ports A and B are configured as output ports.

iv) Interface a 4 * 4 keyboard with 8085 using 8255 an write an ALP for detecting a key closure and return the key code in AL. The debounce period for a key is 10ms. Use software debouncing technique. DEBOUNCE is an available 10ms delay routine.

• *Solution: Port A is used as output port for selecting a row of keys while Port B is used as an input port for sensing a closed key. Thus the keyboard lines are selected one by one through port A and the port B lines are polled continuously till a key closure is sensed. The routine DEBOUNCE is called for key debouncing. The key code is depending upon the selected row and a low sensed column.*

v) Drive an unipolar stepper motor which is interfaced with 8085 using 8255 PPI chip. Write an ALP for that interfacing.



ULN2003 Connection
for Stepper Motor
Pin 8 = GND
Pin 9 = +5V

Use a separate power supply
for the motor

CHAPTER REFERENCE:

[1]. Intel MCS-80/85^{TM} Family User’s Manual, Intel Corporation, 1979

[2]. Intel 8080/8085 Assembly Language Programming Manual, Intel Corporation, 1977-1981