mood-book

UNIT-3

I/O Interface

Introduction:

Any application of a microprocessor based system requires the transfer of data between external circuitry to the microprocessor and microprocessor to the external circuitry. User can give information to the microprocessor based system using keyboard and user can see the result or output information from the microprocessor based system with the help of display device. The transfer of data between keyboard and microprocessor, and microprocessor and display device is called input/output data transfer or I/O data transfer. This data transfer is done with the help of I/O ports.

Input port:

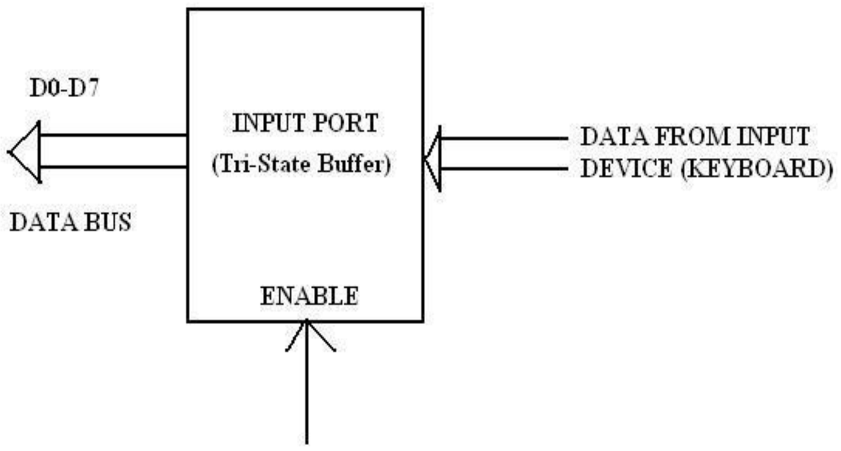


FIG.1 INPUT PORT

It is used to read data from the input device such as keyboard. The simplest form of input port is a buffer. The input device is connected to the microprocessor through buffer, as shown in the fig.1. This buffer is a tri-state buffer and its output is available only when enable signal is active. When microprocessor wants to read data from the input device (keyboard), the control signals from the microprocessor activates the buffer by asserting enable input of the buffer. Once the buffer is enabled, data from the input device is available on the data bus. Microprocessor reads this data by initiating read command.

Output port:

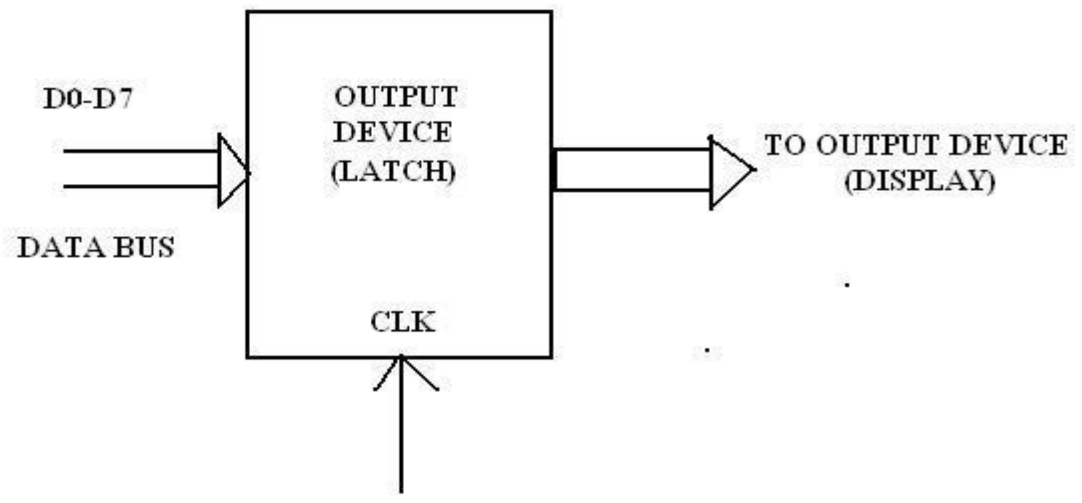


FIG.2 OUTPUT PORT

It is used to send data to the output device such as display from the microprocessor. The simplest form of output port is a latch. The output device is connected to the microprocessor through latch, as shown in the fig.2. When microprocessor wants to send data to the output device is puts the data on the data bus and activates the clock signal of the latch, latching the data from the data bus at the output of latch. It is then available at the output of latch for the output device.

Serial and Parallel Transmission:

In telecommunications, serial transmission is the sequential transmission of signal elements of a group representing a character or other entity of data. Digital serial transmissions are bits sent over a single wire, frequency or optical path sequentially. Because it requires less signal processing and less chance for error than parallel transmission, the transfer rate of each individual path may be faster. This can be used over longer distances as a check digit or parity bit can be sent along it easily.

In telecommunications, parallel transmission is the simultaneous transmission of the signal elements of a character or other entity of data. In digital communications, parallel transmission is the simultaneous transmission of related signal elements over two or more separate paths. Multiple electrical wires are used which can transmit multiple bits simultaneously, which allows for higher data transfer rates than can be achieved with serial transmission. This method is used internally within the computer, for example the internal buses, and sometimes externally for such things as printers, The major issue with this is "skewing" because the wires in parallel data transmission have slightly different properties (not intentionally) so some bits may arrive before others, which may corrupt the message. A parity bit can help to reduce this. However, electrical wire parallel data transmission is therefore less reliable for long distances because corrupt transmissions are far more likely.

Interrupt driven I/O:

In this technique, a CPU automatically executes one of a collection of special routines whenever certain condition exists within a program or a processor system. Example CPU gives response to devices such as keyboard, sensor and other components when they request for service. When the CPU is asked to communicate with devices, it services the devices. Example each time you type a character on a keyboard, a keyboard service routine is called. It transfers the character you typed from the keyboard I/O port into the processor and then to a data buffer in memory.

The interrupt driven I/O technique allows the CPU to execute its main program and only stop to service I/O device when it is told to do so by the I/O system as shown in fig.3. This method provides an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is completed, the processor would resume exactly where it left off.

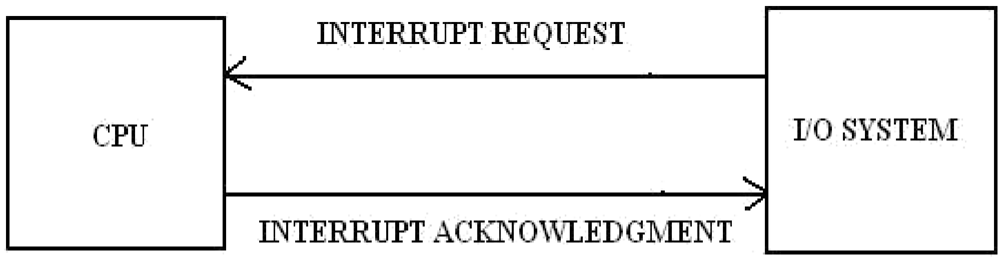


FIG.3 INTERRUPT DRIVEN I/O

An analogy to the interrupt concept is in the classroom, where the professor serves as CPU and the students as I/O ports. The classroom scenario for this interrupt analogy will be such that the professor is busy in writing on the blackboard and delivering his lecture.

The student raises his finger when he wants to ask a question (student requesting for service). The professor then completes his sentence and acknowledges student‟s request by saying “YES” (professor acknowledges the interrupt request). After acknowledgement from the professor, student asks the question and professor gives answer to the question (professor services the interrupt). After that professor continues its remaining lecture form where it was left.

PIO 8255:

The parallel input-output port chip 8255 is also called as programmable**peripheral input-output port.** The Intel‟s 8255 are designed for use with Intel‟s 8-bit, 16-bit and higher capability microprocessors. It has 24 input/output lineswhich may be individually programmed in two groups of twelve lines each, orthree groups of eight lines.

The two groups of I/O pins are named as Group A and Group B. Each of thesetwo groups contains a subgroup of eight I/O lines called as 8-bit port and anothersubgroup of four lines or a 4-bit port. Thus Group A contains an 8-bit port Aalong with a 4-bit port C upper.

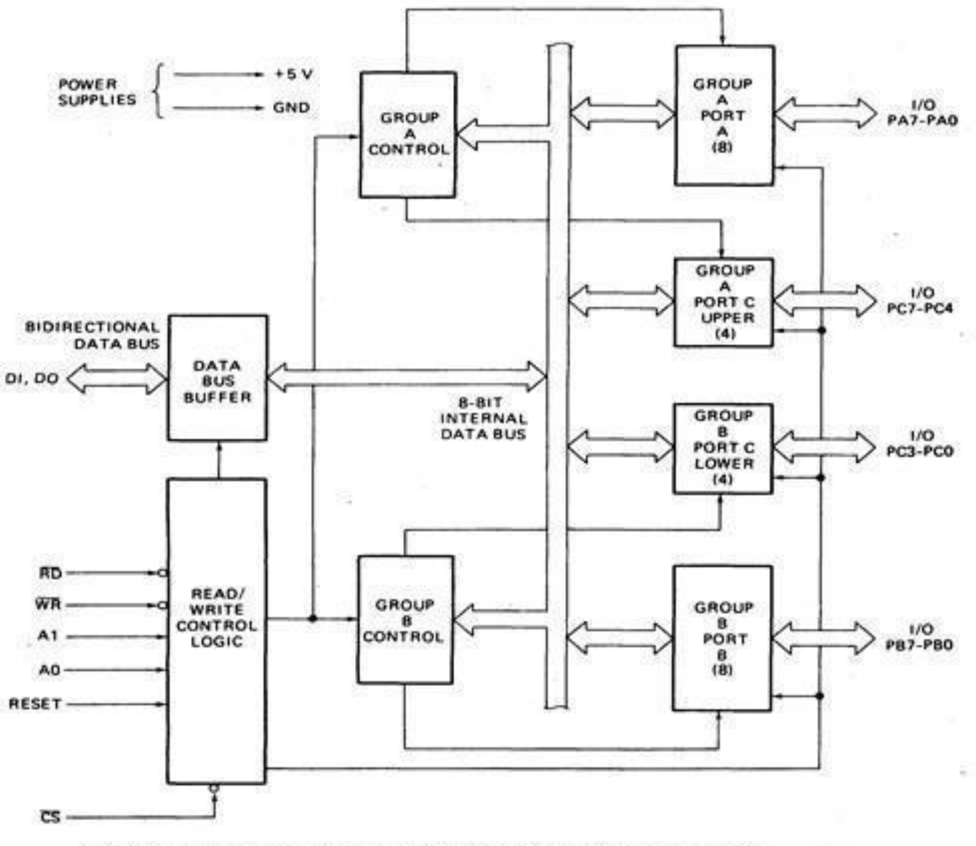
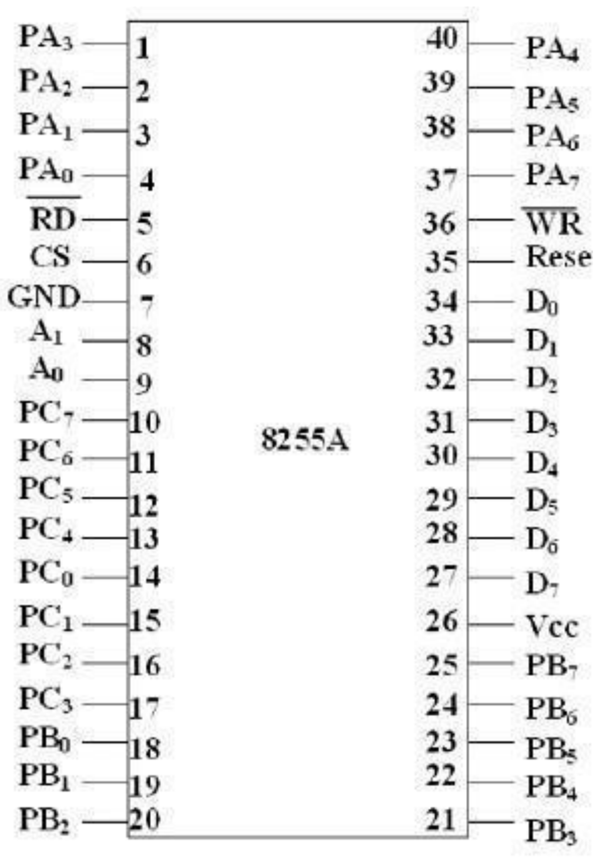


FIGURE Internal block diagram of 8255A programmable parallel port device. (*Intel Corporation*)

The port A lines are identified by symbols PA0-PA7 while the port C lines are identified as PC4-PC7 similarly. Group B contains an 8-bit port B, containing lines PB0-PB7 and a 4-bit port C with lower bits PC0-PC3. The port C upper and port C lower can be used in combination as an 8-bit port C. Both the port Cs is assigned the same address. Thus one may have either three 8-bit I/O ports or two 8-bit and two 4-bit I/O ports from 8255. All of these ports can function independently either as input or as output ports. This can be achieved by programming the bits of an internal register of 8255 called as control word register (CWR). The internal block diagram and the pin configuration of 8255 are shown in figs.

The 8-bit data bus buffer is controlled by the read/write control logic. The read/write control logic manages all of the internal and external transfer of both data and control words. RD, WR, A1, A0 and RESET are the inputs, provided by the microprocessor to READ/WRITE control logic of 8255. The 8-bit, 3-state bidirectional buffer is used to interface the 8255 internal data bus with the external system data bus. This buffer receives or transmits data upon the execution of input or output instructions by the microprocessor. The control words or status information is also transferred through the buffer.

Pin Diagram of 8255A



8255A Pin Configuration

The pin configuration of 8255 is shown in fig.

- The port A lines are identified by symbols PA0-PA7 while the port C lines are
- Identified as PC4-PC7. Similarly, Group B contains an 8-bit port B, containing lines PB0-PB7 and a 4-bit port C with lower bits PC0- PC3. The port C upper and port C lower can be used in combination as an 8-bit port C.
- Both the port C is assigned the same address. Thus one may have either three 8-bit I/O ports or two 8-bit and two 4-bit ports from 8255. All of these ports can function independently either as input or as output ports. This can be

achieved by programming the bits of an internal register of 8255 called as control word register (CWR).

The 8-bit data bus buffer is controlled by the read/write control logic. The read/write control logic manages all of the internal and external transfers of both data and control words.

RD,WR, A1, A0 and RESET are the inputs provided by the microprocessor to the READ/ WRITE control logic of 8255. The 8-bit, 3-state bidirectional buffer is used to interface the 8255 internal data bus with the external system data bus.

This buffer receives or transmits data upon the execution of input or output instructions by the microprocessor. The control words or status information is also transferred through the buffer.

The signal description of 8255 is briefly presented as follows:

PA7-PA0: These are eight port A lines that acts as either latched output or buffered input lines depending upon the control word loaded into the control word register.

PC7-PC4: Upper nibble of port C lines. They may act as either output latches or input buffers lines.

This port also can be used for generation of handshake lines in mode1 or mode2.

PC3-PC0: These are the lower port C lines; other details are the same as PC7-PC4 lines.

PB0-PB7: These are the eight port B lines which are used as latched output lines or buffered input lines in the same way as port A.

RD: This is the input line driven by the microprocessor and should be low to indicate read operation to 8255.

WR: This is an input line driven by the microprocessor. A low on this line indicates write operation.

CS: This is a chip select line. If this line goes low, it enables the 8255 to respond to RD and WR signals, otherwise RD and WR signal are neglected.

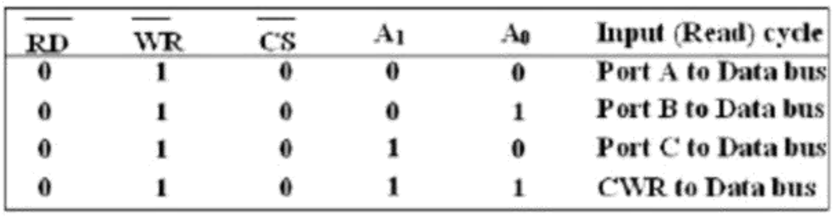
D0-D7: These are the data bus lines those carry data or control word to/from the microprocessor.

RESET: Logic high on this line clears the control word register of 8255. All ports are set as input ports by default after reset.

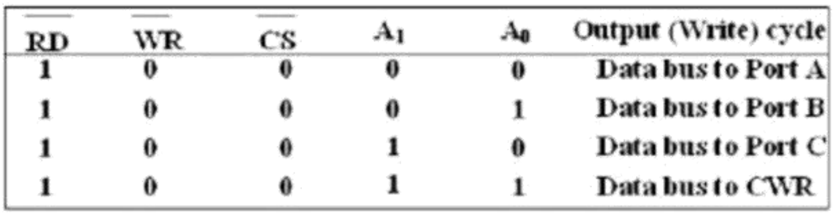
A1-A0: These are the address input lines and are driven by the microprocessor.

These lines A1-A0 with RD, WR and CS from the following operations for 8255. These address lines are used for addressing any one of the four registers, i.e. three ports and a control word register as given in table below.

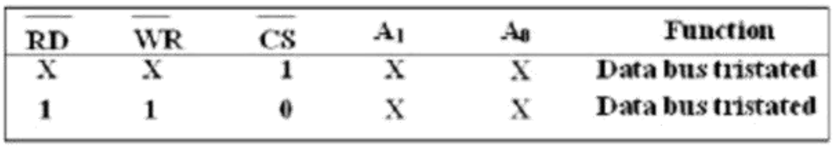
In case of 8086 systems, if the 8255 is to be interfaced with lower order data bus, the A0 and A1 pins of 8255 are connected with A1 and A2 respectively.



$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	A_1	A_0	Input (Read) cycle
0	1	0	0	0	Port A to Data bus
0	1	0	0	1	Port B to Data bus
0	1	0	1	0	Port C to Data bus
0	1	0	1	1	CWR to Data bus



$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	A_1	A_0	Output (Write) cycle
1	0	0	0	0	Data bus to Port A
1	0	0	0	1	Data bus to Port B
1	0	0	1	0	Data bus to Port C
1	0	0	1	1	Data bus to CWR



$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	A_1	A_0	Function
X	X	1	X	X	Data bus tristated
1	1	0	X	X	Data bus tristated

Control Word Register

Modes of Operation of 8255

These are two basic modes of operation of 8255. I/O mode and Bit Set-Reset mode (BSR).

In I/O mode, the 8255 ports work as programmable I/O ports, while in BSR mode only port C (PC0-PC7) can be used to set or reset its individual port bits.

Under the I/O mode of operation, further there are three modes of operation of 8255, so as to support different types of applications, mode 0, mode 1 and mode 2.

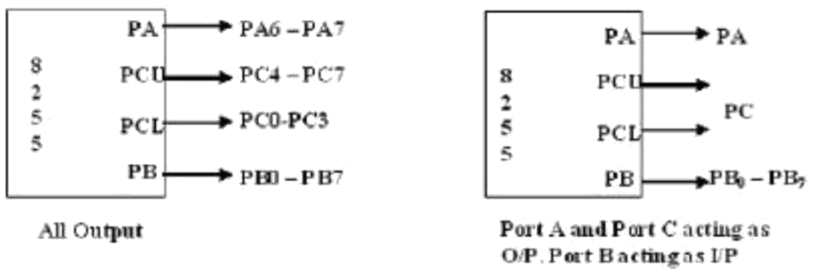
BSR Mode: In this mode any of the 8-bits of port C can be set or reset depending on D0 of the control word. The bit to be set or reset is selected by bit select flags D3, D2 and D1 of the CWR as given in table.

I/O Modes:

a) Mode 0 (Basic I/O mode): This mode is also called as basic input/output Mode. This mode provides simple input and output capabilities using each of the threeports. Data can be simply read from and written to the input and output portsrespectively, after appropriate initialization.

D_3	D_2	D_1	Selected bits of port C
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

BSR Mode : CWR Format



Mode 0

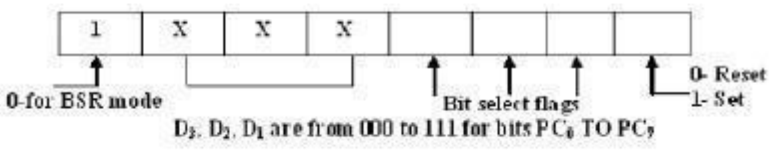
The salient features of this mode are as listed below:

1. Two 8-bit ports (port A and port B) and two 4-bit ports (port C upper and lower) are available. The two 4-bit ports can be combined used as a third 8-bit port.
2. Any port can be used as an input or output port.
3. Output ports are latched. Input ports are not latched.
4. A maximum of four ports are available so that overall 16 I/O configurations arepossible.

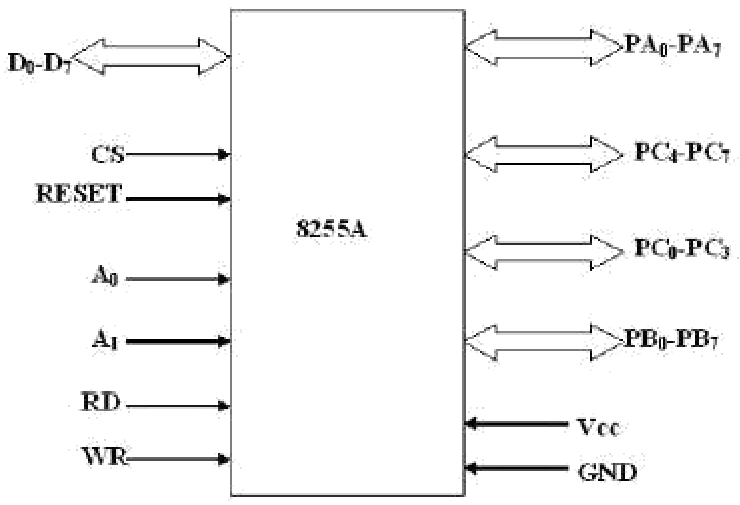
All these modes can be selected by programming a register internal to 8255known as CWR.

The control word register has two formats. The first format is valid for I/O modesof operation, i.e. modes 0, mode 1 and mode 2 while the second format is validfor bit set/reset (BSR) mode of operation.

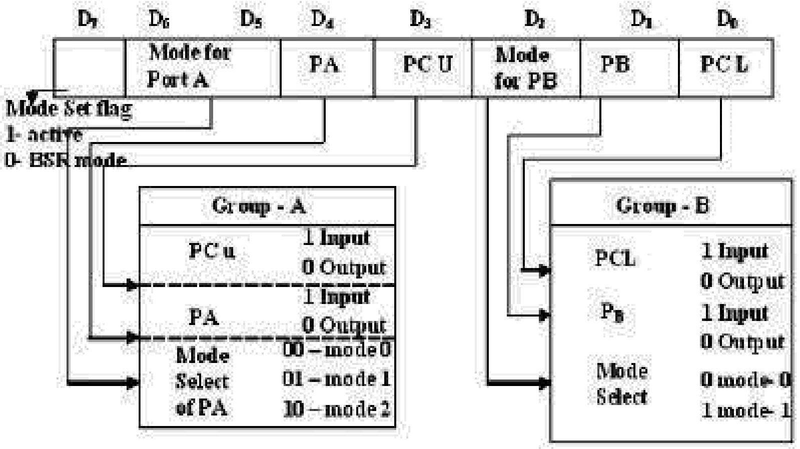
These formats are shown in followingfig.


1
X
X
X
0-for BSR mode
Bit select flags
0- Reset
1- Set
D_3, D_2, D_1 are from 000 to 111 for bits PC_0 TO PC_7

I/O Mode Control Word Register Format and
BSR Mode Control Word Register Format



Signals of 8255



Control Word Format of 8255

b) Mode 1: (Strobed input/output mode) In this mode the handshaking control the input and output action of the specified port. Port C lines PC0-PC2, provide strobe orhandshake lines for port B. This group which includes port B and PC0-PC2 is called asgroup B for Strobed data input/output. Port C lines PC3-PC5 provides strobe lines for portA.This group including port A and PC3-PC5 from group A. Thus port C is utilized forgenerating handshake signals.

The salient features of mode 1 are listed as follows:

1. Two groups – group A and group B are available for strobed data transfer.
2. Each group contains one 8-bit data I/O port and one 4-bit control/data port.
3. The 8-bit data port can be either used as input and output port. The inputs andoutputs both are latched.
4. Out of 8-bit port C, PC0-PC2 are used to generate control signals for port B andPC3-PC5 are used to generate control signals for port A. the lines PC6, PC7 may be used as independent data lines.

The control signals for both the groups in input and output modes areexplained as follows:

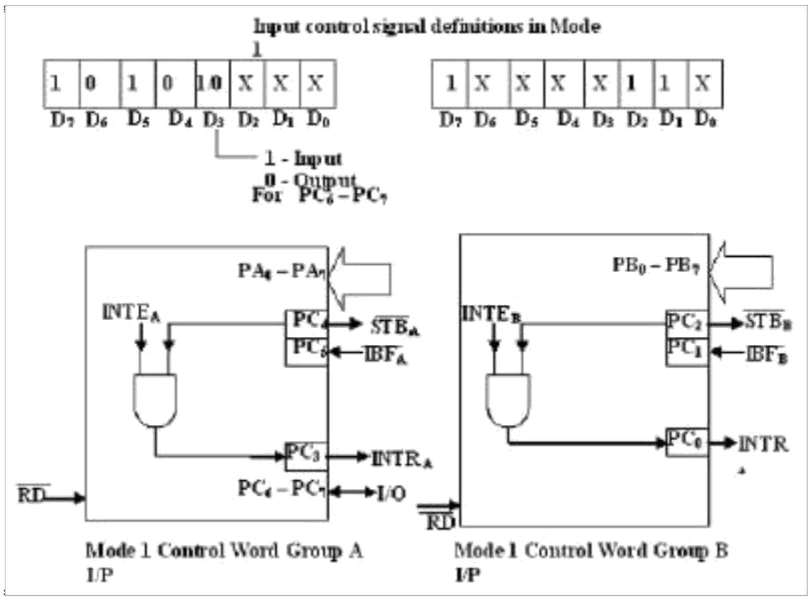
Input control signal definitions (mode 1):

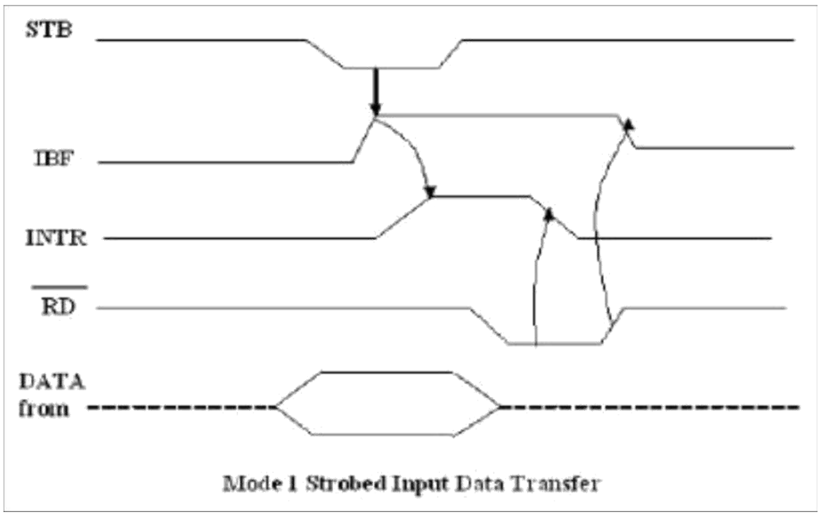
- **STB** (Strobeinput) – If this lines falls to logic low level, the data available at 8-bit input port is loaded into input latches.
- **IBF** (Input buffer full) – If this signal rises to logic 1, it indicates that data hasbeen loaded into latches, i.e. it works as an acknowledgement. IBF is set by a lowon STB and is reset by the rising edge of RD input.
- **INTR** (Interruptrequest) – This active high output signal can be used tointerrupt the CPU whenever an input device requests the service. INTR is set by ahigh STBpin and a high at IBF pin. INTE is an internal flag that can be controlledby the bit set/reset mode of either PC4 (INTEA) or PC2 (INTEB) as shown in fig.
- INTR is reset by a falling edge of RD input. Thus an external input device can berequest the service of the processor by putting the data on the bus and sending thestrobe signal.

Output control signal definitions (mode 1):

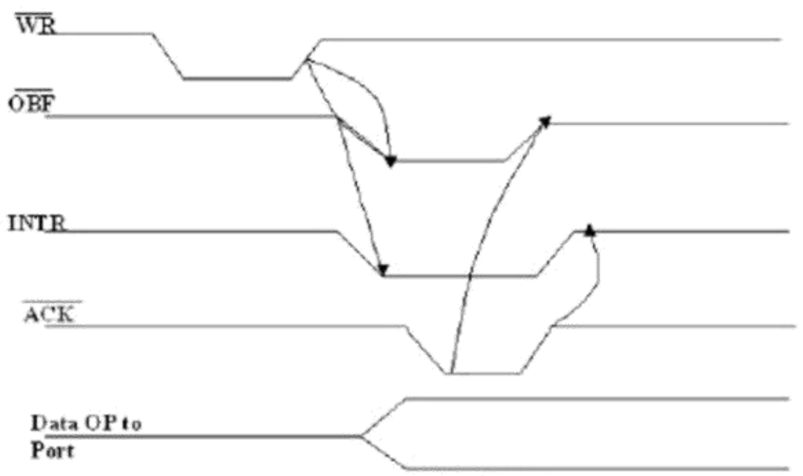
- **OBF** (Output buffer full) – This status signal, whenever falls to low, indicatesthat CPU has written data to the specified output port. The OBF flip-flop will beset by a rising edge of WR signal and reset by a low going edge at the ACKinput.
- ACK (Acknowledgeinput) – ACK signal acts as an acknowledgement to begiven by an output device. ACK signal, whenever low, informs the CPU that thedata transferred by the CPU to the output device through the port is received bythe output device.
- **INTR** (Interruptrequest) – Thus an output signal that can be used to interruptthe CPU when an output device acknowledges the data received from the CPU.INTR is set when ACK, OBF and INTE are 1. It is reset by a

fallingedge on WRinput. The INTEA and INTEB flags are controlled by the bit set-reset mode ofPC6 and PC2 respectively.



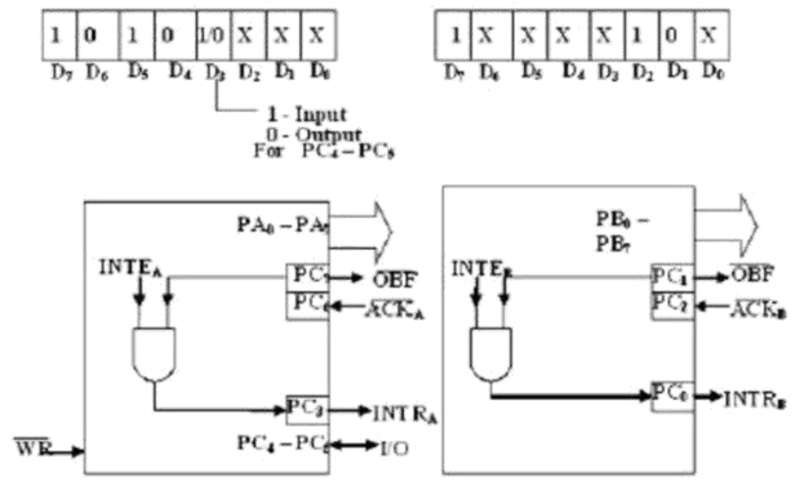


Mode 1 Strobed Input Data Transfer



Mode 1 Strobed Data Output

Output control signal definitions Mode 1



Mode 1 Control Word Group A   Mode 1 Control Word Group B

c) Mode 2 (Strobed bidirectional I/O): This mode of operation of 8255 is alsocalled as strobed bidirectional I/O. This mode of operation provides 8255 with additional features for communicating with a peripheral device on an 8-bit databus. Handshaking signals are provided to maintain proper data flow andsynchronization between the data transmitter and receiver. The interruptgeneration and other functions are similar to mode 1.

In this mode, 8255 is a bidirectional 8-bit port with handshake signals. The Rdand WR signals decide whether the 8255 is going to operate as an input port oroutput port.

The Salient features of Mode 2 of 8255 are listed as follows:

1. The single 8-bit port in group A is available.
2. The 8-bit port is bidirectional and additionally a 5-bit control port is available.
3. Three I/O lines are available at port C.(PC2 – PC0)
4. Inputs and outputs are both latched.

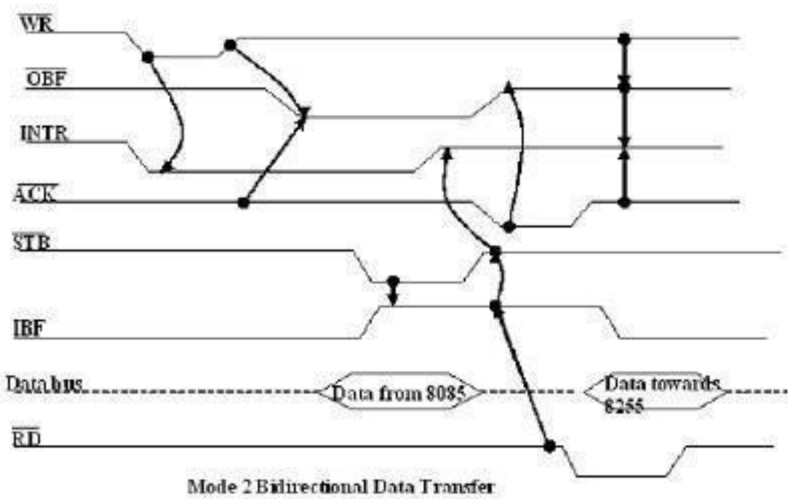
5. The 5-bit control port C (PC3-PC7) is used for generating / accepting handshakesignals for the 8-bit data transfer on port A.

Control signal definitions in mode 2:

- **INTR** – (Interrupt request) As in mode 1, this control signal is active high and isused to interrupt the microprocessor to ask for transfer of the next data byteto/from it. This signal is used for input (read) as well as output (write) operations.
- **Control Signals for Output operations**:
- OBF (Output buffer full) – This signal, when falls to low level, indicates that theCPU has written data to port A.
- ACK (Acknowledge) This control input, when falls to logic low level, Acknowledges that the previous data byte is received by the destination and nextbyte may be sent by the processor. This signal enables the internal tristate buffersto send the next data byte on port A.
- **INTE1** (A flag associated with OBF) This can be controlled by bit set/resetmode with PC6.

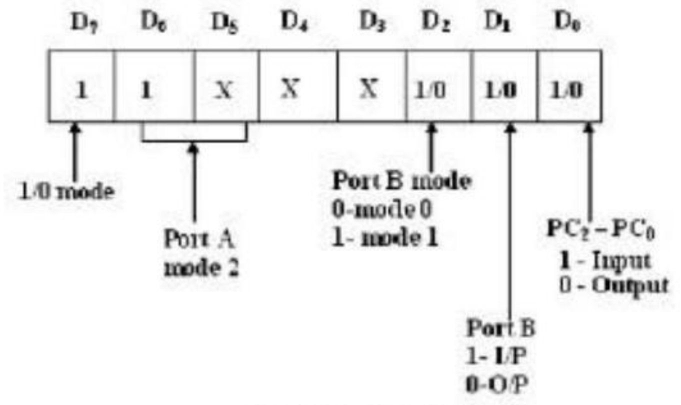
Control signals for input operations:

- STB (Strobe input)a low on this line is used to strobe in the data into the inputLatches of 8255.
- **IBF** (Input buffer full) when the data is loaded into input buffer, this signal risesto logic „1‟. This can be used as an acknowledge that the data has been receivedby the receiver.
- The waveforms in fig show the operation in Mode 2 for output as well as inputport.
- Note: WR must occur before ACK and STB must be activated before RD.

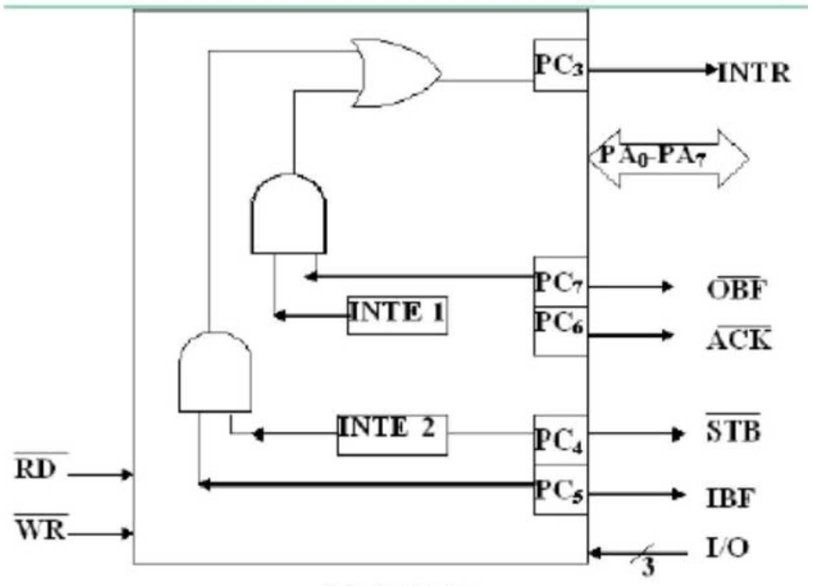


Mode 2 Bidirectional Data Transfer

- The following fig shows a schematic diagram containing an 8-bit bidirectionalport, 5-bit control port and the relation of INTR with the control pins. Port B caneither be set to Mode 0 or 1 with port A(Group A) is in Mode 2.
- Mode 2 is not available for port B. The following fig shows the control word.
- The INTR goes high only if IBF, INTE2, STB and RD go high or OBF,
- INTE1, ACK and WR go high. The port C can be read to know the status of theperipheral device, in terms of the control signals, using the normal I/Oinstructions.



Mode 2 control word



Mode 2 pins

Interfacing Analog to Digital Data Converters:

- In most of the cases, the PIO 8255 is used for interfacing the analog to digital converters with microprocessor.
- We have already studied 8255 interfacing with 8086 as an I/O port, in previous section. This section we will only emphasize the interfacing techniques of analog to digital converters with 8255.
- The analog to digital converters is treated as an input device by the microprocessor that sends an initializing signal to the ADC to start the analogy to digital data conversation process. The start of conversation signal is a pulse of a specific duration.
- The process of analog to digital conversion is a slow
- Process and the microprocessor have to wait for the digitaldata till the conversion is over. After the conversion isover, the ADC sends end of conversion EOC signal toinform themicroprocessor that the conversion is over andthe result is ready at the output buffer of the ADC. Thesetasks of issuing an SOC pulse to ADC, reading EOC signalfrom the ADC and reading the digital output of the ADCare carried out by the CPU using 8255 I/O ports.
- The time taken by the ADC from the active edge of SOCpulse till the active edge of EOC signal is called as theconversion delay of the ADC.
- It may range anywhere from a few microseconds in caseof fast ADC to even a few hundred milliseconds in case ofslow ADCs.
- The available ADC in the market use different conversiontechniques for conversion of analog signal to digitals.Successive approximation techniques and dual slopeintegration techniques are the most popular techniquesused in the integrated ADC chip.
- General algorithm for ADC interfacing contains thefollowing steps:
- Ensure the stability of analog input, applied to the ADC.
- Issue start of conversion pulse to ADC
- Read end of conversion signal to mark the end ofconversion processes.
- Read digital data output of the ADC as equivalent digitaloutput.
- Analog input voltage must be constant at the input of theADC right from the start of conversion till the end of theconversion to get correct results. This may be ensured by asample and hold circuit which samples the analog signaland holds it constant for specific time duration. Themicroprocessor may issue a hold signal to the sample andhold circuit.
- If the applied input changes before the completeconversion process is over, the digital equivalent of theanalog input calculated by the ADC may not be correct.

ADC 0808/0809:

- The analog to digital converter chips 0808 and 0809 are 8-bit CMOS, successive approximation converters. This technique is one of the fast techniques for analog to digital conversion. The conversion delay is 100µs at a clock frequency of 640 KHz, which is quite low as compared to other converters. These converters do not need any external zero or full scale adjustments as they are already taken care of by internal circuits.
- These converters internally have a 3:8 analog multiplexer so that at a time eight different analog conversion by using address lines - ADD A, ADD B, ADD C, as shown. Using these address inputs, multichannel data acquisition system can be designed using a single ADC. The CPU may drive these lines using output port lines in case of multichannel applications. In case of single input applications, these may be hardwired to select the proper input.
- There are unipolar analog to digital converters, i.e. they are able to convert only positive analog input voltage to their digital equivalent. These chips do not contain any internal sample and hold circuit.
- If one needs a sample and hold circuit for the conversion of fast signal into equivalent digital quantities, it has to be externally connected at each of the analog inputs.

Fig (1) and Fig (2) show the block diagrams and pin diagrams for ADC 0808/0809.

Table.1

Analog I/P selected	**Address lines**		
	C	**B**	**A**
I/P 0	0	0	0
I/P 1	0	0	1
I/P 2	0	1	0
I/P 3	0	1	1
I/P 4	1	0	0
I/P 5	1	0	1
I/P 6	1	1	0
I/P 7	1	1	1

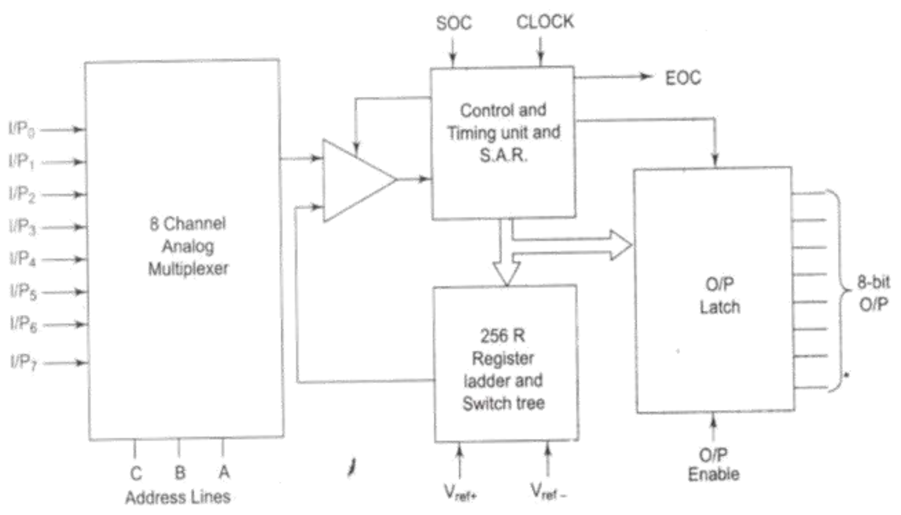
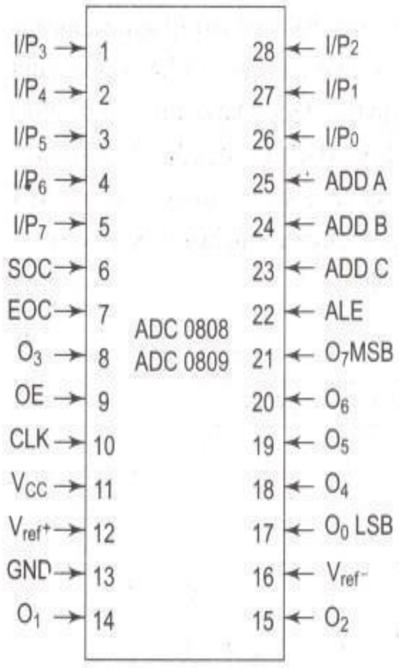


Fig.1 Block Diagram of ADC 0808/0809



I/P_0 - I/P_7	Analog inputs
ADD A, B, C	Address lines for selecting analog inputs
O_7 - O_0	Digital 8-bit output with O_7 MSB and O_0 LSB
SOC	Start of conversion signal pin
EOC	End of conversion signal pin
OE	Output latch enable pin, if high enable output
CLK	Clock input for ADC
V_{CC}, GND	Supply pins +5V and GND
V_{ref+} and V_{ref-}	Reference voltage positive (+5 Volts maximum) and Reference voltage negative (0V minimum)

Fig.2 Pin Diagram of ADC 0808/0809

Some Electrical Specifications Of The ADC 0808/0809 Are Given In Table.2.

Table.2

Minimum SOC pulse width	100 ns
Minimum ALE pulse width	100 ns
Clock frequency	10 to 1280 kHz
Conversion time	100 ms at 640 kHz
Resolution	8-bit
Error	+/–1 LSB
V_{ref+}	Not more than +5V
V_{ref-}	Not less than GND
+ V_{cc} suυply	+ 5 V DC
Logical 1 i/p voltage	minimum V_{cc} –1.5 V
Logical 0 i/p voltage	maximum 1.5 V
Logical 1 o/p voltage	minimum V_{cc}–0.4 V
Logical 0 o/p voltage	maximum 0.45 V

The Timing Diagram Of Different Signals Of Adc0808 Is Shown In Fig.3

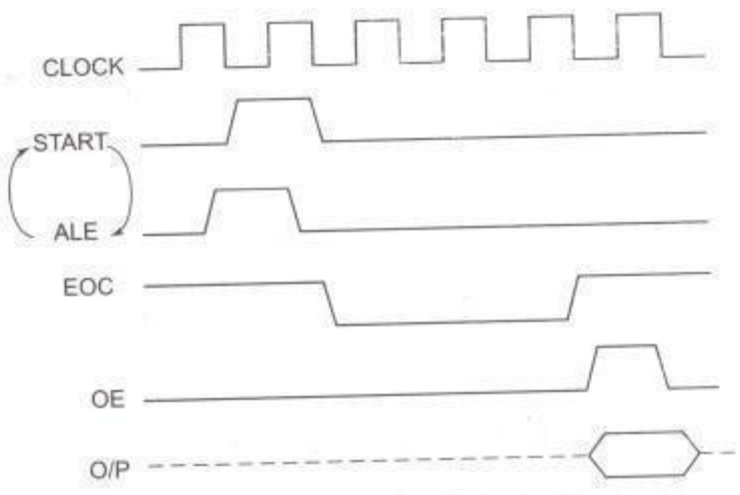
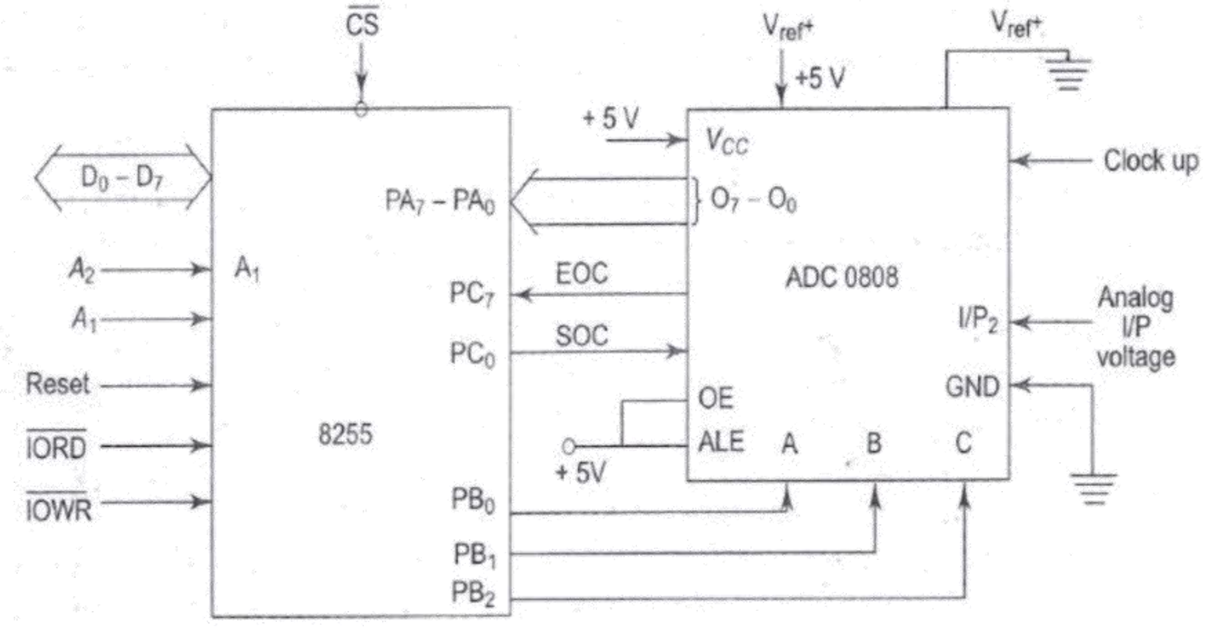


Fig.3 Timing Diagram Of ADC 0808.



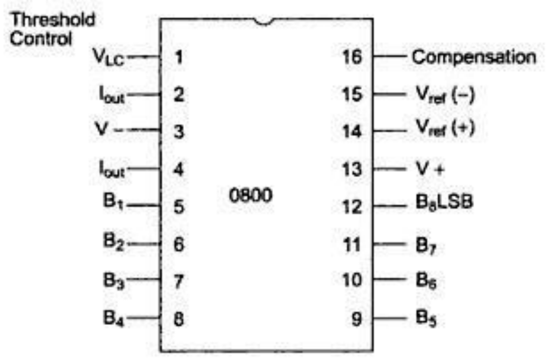
Interfacing ADC0808 with 8086

Interfacing Digital To Analog Converters:

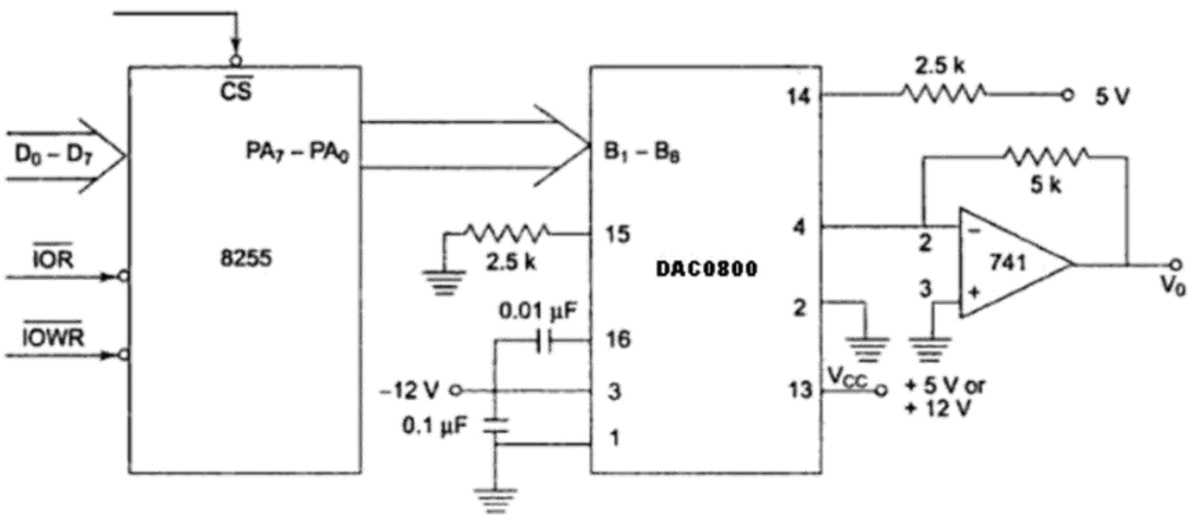
The digital to analog converters convert binary numbers into their analog equivalent voltages. The DAC find applications in areas like digitally controlled gains, motor speed controls, programmable gain amplifiers, etc.

DAC0800 8-bit Digital to Analog Converter

- The DAC 0800 is a monolithic 8-bit DAC manufactured by National Semiconductor.
- It has settling time around 100ms and can operate on a range of power supply voltages i.e. from 4.5V to +18V.
- Usually the supply V+ is 5V or +12V.
- The V-pin can be kept at a minimum of -12V.



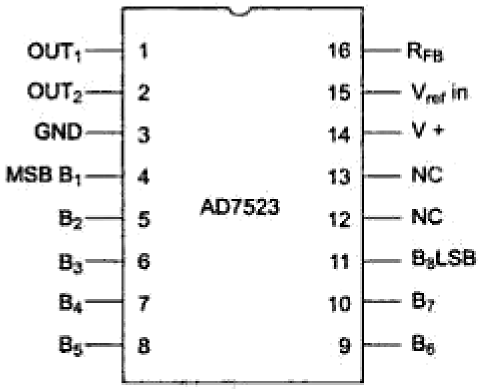
Pin Diagram of DAC 0800



Interfacing DAC0800 with 8086

Ad 7523 8-Bit Multiplying DAC:

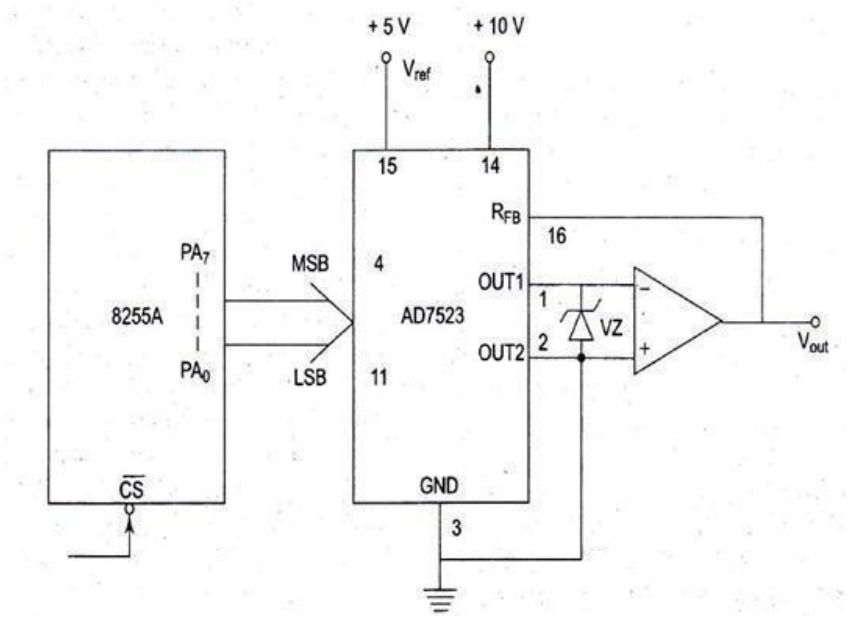
- Intersil‟s AD 7523 is a 16 pin DIP, multiplying digital to analog converter, containing R-2R ladder(R=10KΩ) for digital to analog conversion along with single pole double through NMOS switches to connect the digital inputs to the ladder.



Pin Diagram of AD7523

- The supply range extends from +5V to +15V , while Vref may be anywhere between -10V to +10V. The maximum analog output voltage will be +10V, when all the digital inputs are at logic high state. Usually a Zener is connected between OUT1 and OUT2 to save the DAC from negative transients.
- An operational amplifier is used as a current to voltage converter at the output of AD 7523 to convert the current output of AD7523 to a proportional output voltage.

- It also offers additional drive capability to the DAC output. An external feedback resistor acts to control the gain. One may not connect any external feedback resistor, if no gain control is required.



Interfacing AD7523 with 8086

Stepper Motor Interfacing:

- A stepper motor is a device used to obtain an accurate position control of rotating shafts. It employs rotation of its shaft in terms of steps, rather than continuous rotation as in case of AC or DC motors. To rotate the shaft of the stepper motor, a sequence of pulses is needed to be applied to the windings of the stepper motor, in a proper sequence.
- The number of pulses required for one complete rotation of the shaft of the stepper motor is equal to its number of internal teeth on its rotor. The stator teeth and the rotor teeth lock with each other to fix a position of the shaft.
- With a pulse applied to the winding input, the rotor rotates by one teeth position or an angle x. The angle x may be calculated as:

$$X=360^0/\text{no. of rotor teeth}$$

- After the rotation of the shaft through angel x, the rotor locks itself with the next tooth in the sequence on the internal surface of stator.
- The internal schematic of a typical stepper motor with four windings is shown in fig.1.
- The stepper motors have been designed to work with digital circuits. Binary level pulses of 0-5V are required at its winding inputs to obtain the rotation of shafts. The sequence of the pulses can be decided, depending upon the required motion of the shaft.

- Fig.2 shows a typical winding arrangement of the stepper motor.
- Fig.3 shows conceptual positioning of the rotor teeth on the surface of rotor, for a six teeth rotor.

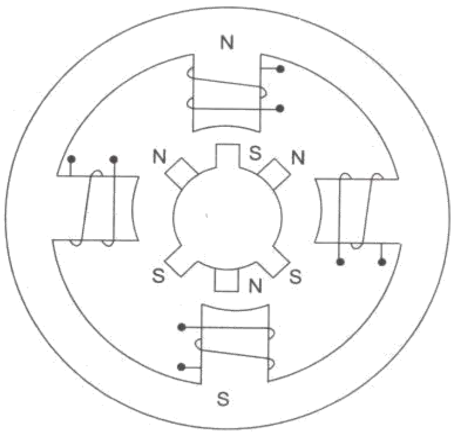


Fig.1 Internal schematic of a four winding stepper motor

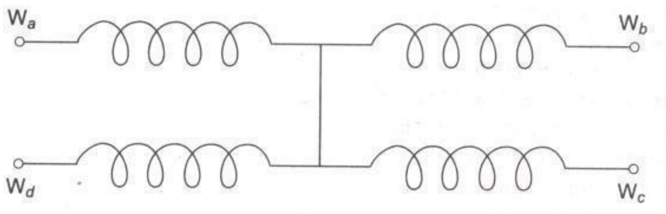


Fig.2 Winding arrangement of a stepper motor.

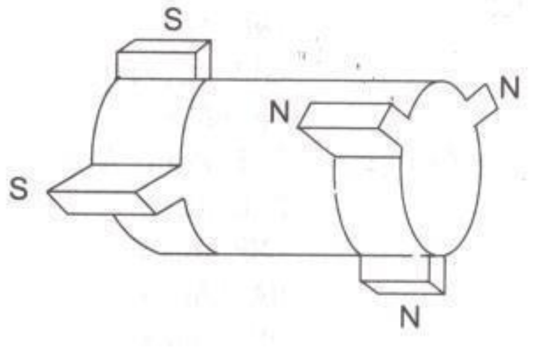


Fig.3 Stepper motor rotor

- The circuit for interfacing a winding Wn with an I/O port is given in fig.4. Each of the windings of a stepper motor needs this circuit for its interfacing with the output port. A typical stepper motor may have parameters like torque 3 Kg-cm, operating voltage 12V, current rating 0.2 A and a step angle 1.8^0 i.e. 200 steps/revolution (number of rotor teeth).

- A simple schematic for rotating the shaft of a stepper motor is called a wave scheme. In this scheme, the windings Wa, Wb, Wc and Wd are applied with the required voltages pulses, in a cyclic fashion. By reversing the sequence of excitation, the direction of rotation of the stepper motor shaft may be reversed.
- Table.1 shows the excitation sequences for clockwise and anticlockwise rotations. Another popular scheme for rotation of a stepper motor shaft applies pulses to two successive windings at a time but these are shifted only by one position at a time. This scheme for rotation of stepper motor shaft is shown in table2.

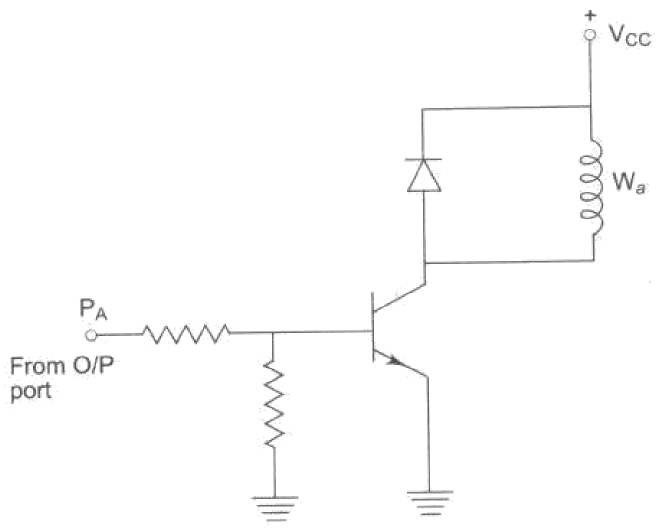


Fig.4 interfacing stepper motor winding.

Table.1 Excitation sequence of a stepper motor using wave switching scheme.

Motion	step	A	B	C	D
Clock wise	1	1	0	0	0
	2	0	1	0	0
	3	0	0	1	0
	4	0	0	0	1
	5	1	0	0	0
Anticlock wise	1	1	0	0	0
	2	0	0	0	1
	3	0	0	1	0
	4	0	1	0	0
	5	1	0	0	0

Table.2 An alternative scheme for rotating stepper motor shaft

Motion	step	A	B	C	D
Clock wise	1	0	0	1	1
	2	0	1	1	0
	3	1	1	0	0
	4	1	0	0	1
	5	0	0	1	1
Anticlock wise	1	0	0	1	1
	2	1	0	0	1
	3	1	1	0	0
	4	0	1	1	0
	5	0	0	0	0

Keyboard Interfacing

- In most keyboards, the key switches are connected in a matrix of Rows and Columns.
- Getting meaningful data from a keyboard requires three major tasks:

1. D e t e c t a k e y p r e s s
2. D e b o u n c e t h e k e y p r e s s .
3. Encode the keypress (produce a standard code for the pressed key).

Logic „0‟ is read by the microprocessor when the key is pressed.

Key Debounce:

Whenever a mechanical push-bottom is pressed or released once,the mechanical components of the key do not change the positionsmoothly; rather it generates a transient response. These may be interpreted as the multiple pressures and responded accordingly.

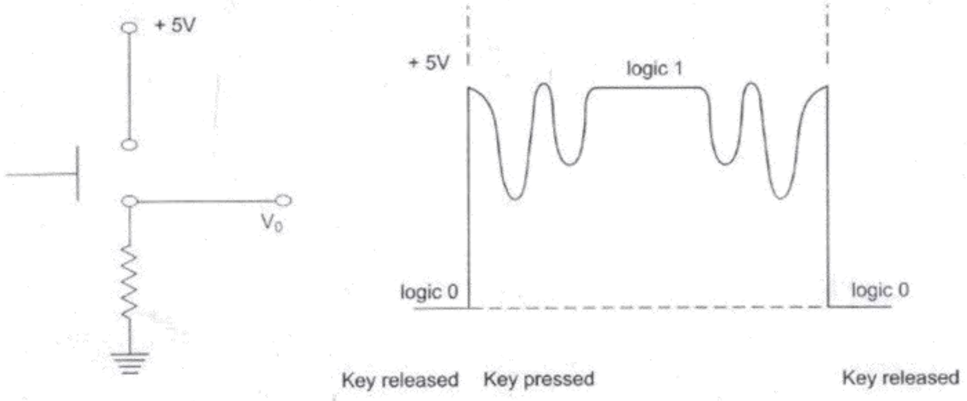


Fig. 5.23 *A Mechanical Key and Its Response*

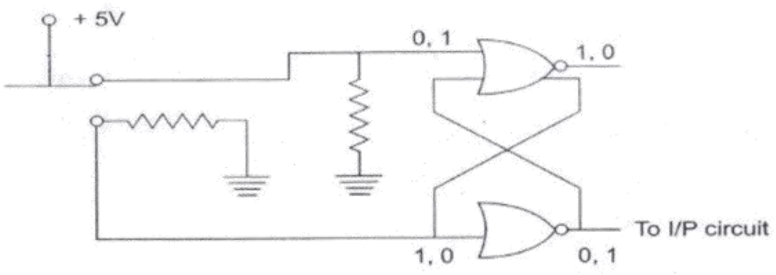
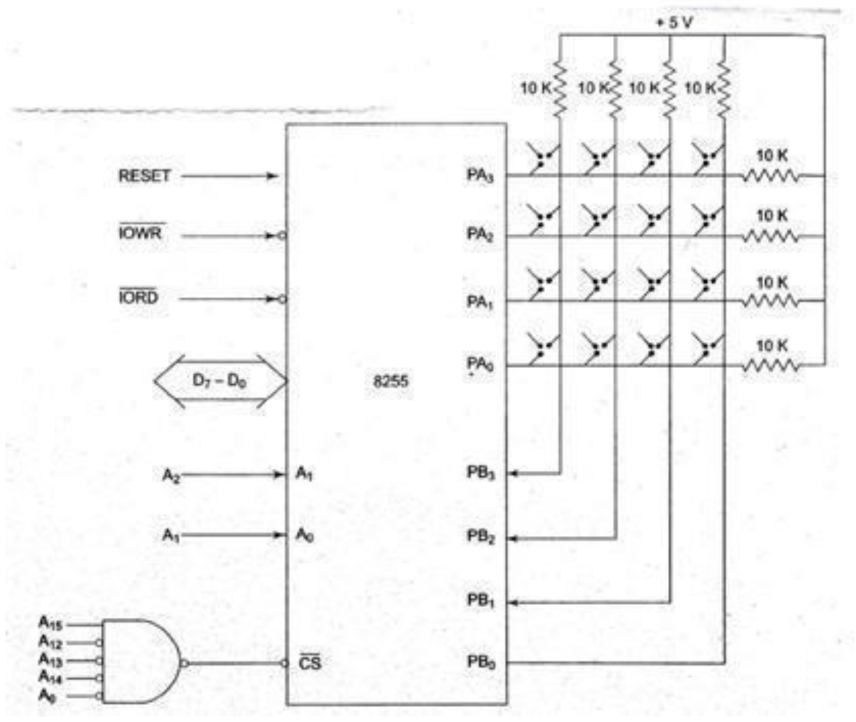


Fig. 5.24 *Hardware Debouncing Circuit*

- The rows of the matrix are connected to four output Port lines, &columns are connected to four input Port lines.
- When no keys are pressed, the column lines are held high by the pull-up resistors connected to +5v.
- Pressing a key connects a row & a column.
- To detect if any key is pressed is to output 0‟s to all rows & then check columns to see it a pressed key has connected a low (zero) to a column.
- Once the columns are found to be all high, the program enters another loop, which waits until a low appears on one of the columns i.e indicating a key press.
- A simple 20/10 msec delay is executed to debounce task.
- After the debounce time, another check is made to see if the key is still pressed. If the columns are now all high, then no key is pressed & the initial detection was caused by a noise pulse.
- To avoid this problem, two schemes are suggested:
    1. Use of Bistablemultivibrator at the output of the key to debounce it.
    2. The microprocessor has to wait for the transient period (at least for 10 ms), so that the transient response settles down and reaches a steady state.

- If any of the columns are low now, then the assumption is made that it was a valid key press.

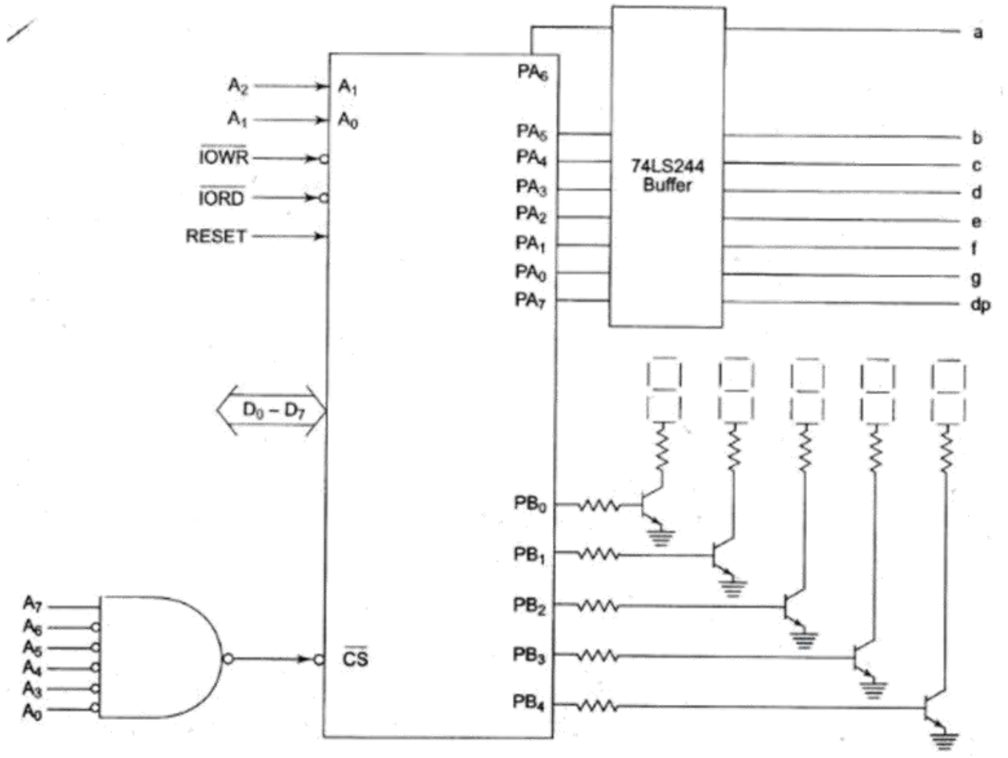
- The final task is to determine the row & column of the pressed key &convert this information to Hex-code for the pressed key.
- The 4-bit code from I/P port & the 4-bit code from O/P port (row &column) are converted to Hex-code.



Interfacing 4x4 keyboard

Display Interface

Number to be displayed	PA7d p	PA6 a	PA5 b	PA4 c	PA3 d	PA2 e	PA1 f	PA0 g	Code
1	1	1	0	0	1	1	1	1	CF
2	1	0	0	1	0	0	1	0	92
3	1	0	0	0	0	1	1	0	86
4	1	1	0	0	1	1	0	0	CC
5	1	0	1	0	0	1	0	0	A4



Interfacing multiplexed 7-segment display

Interfacing with Advanced devices

4.1 MEMORY AND I/O INTERFACING

(Ref: Interfacing through Microprocessors by K. Subba Rao, Hi-tech publishers, P. 163-166)

4.1.1 I/O Interface

Any application of a microprocessor system requires the transfer of data between microprocessor and external environment and also with in the microprocessor. This is known as Input/Output. There are three different ways that the data transfer can take place. They are

(1) Program controlled I/O
(2) Interrupt Program Controlled I/O
(3) Hardware controlled I/O

In program controlled I/O data transfer scheme the transfer of data is completely under the control of the microprocessor program. In this case an I/O operation takes place only when an I/O transfer instruction is executed.

In an interrupt program controlled I/O an external device indicates directly to the microprocessor its readiness to transfer data by a signal at an interrupt input of the microprocessor. When microprocessor receives this signal the control is transferred to ISS (Interrupt service subroutine) which performs the data transfer.

Hardware controlled I/O is also known as direct memory access DMA. In this case the data transfer takes place directly between an I/O device and memory but not through microprocessors. Microprocessor only initializes the process of data transfer by indicating the starting address and the number of words to be transferred.

The instruction .set of any microprocessor contains instructions that transfer information to an I/O device and to read information from an I/O device. In 8086 we have IN, OUT instructions for this purpose. OUT instruction transfers information to an I/O device where as IN instruction is used to read information from an I/O device. Both the instructions perform the data transfer using accumulator AL or AX. The I/O address is stored in register DX.

The port number is specified along with IN or OUT instruction. The external I/O interface decodes to find the address of the I/O device. The 8 bit fixed port number appears on address bus A_0 - A_7 with A_8 - A_{15} all zeros. The address connections above A_{15} are undefined for an I/O instruction. The 16 bit variable port number appears on address connections A_0 - A_{15}. The above notation indicates that first 256 I/O port addresses 00 to FF are accessed by both the fixed and variable I/O instructions. The I/O addresses from 0000 to FFFF are accessed by the variable I/O address.

I/O devices can be interfaced to the microprocessors using two methods. They are I/O mapped I/O and memory mapped I/O. The I/O mapped I/O is also known as isolated I/O or direct I/O. In I/O mapped I/O the IN and OUT instructions transfer data between the accumulator or memory and I/O device. In memory mapped I/O the instruction that refers memory can perform the data transfer.

I/O mapped I/O is the most commonly used I/O transfer technique. In this method I/O locations are placed separately from memory. The addresses for isolated I/O devices are separate from memory. Using this method user can use the entire memory. This method allows data transfer only by using instructions IN, OUT. The pins M/ $\overline{IO}$ and W/R are used to indicate I/O read or an I/O write operations. The signals on these lines indicate that the address on the address bus is for I/O devices.

Memory mapped I/O does not use the IN, OUT instruction it uses only the instruction that transfers data between microprocessor and memory. A memory mapped I/O device is treated as memory location. The disadvantage in this system is the overall memory is reduced. The advantage of this system is that any memory transfer instruction can be used for data transfer and control signals like I/O read and I/O write are not necessary which simplify the hardware.

4.1.2 Memory interfacing

Memory is an integral part of a microcomputer system. There are two main types of memory.

(i) **Read only memory (ROM):** As the name indicates this memory is available only for reading purpose. The various types available under this category are PROM, EPROM, EEPROM which contain system software and permanent system data.

(ii) **Random Access memory (RAM):** This is also known as Read Write Memory. It is a volatile memory. RAM contains temporary data and software programs generally for different applications.

While executing particular task it is necessary to access memory to get instruction codes and data stored in memory. Microprocessor initiates the necessary signals when read or write operation is to be performed. Memory device also requires some signals to perform read and write operations using various registers. To do the above job it is necessary to have a device and a circuit, which performs this task is known as interfacing device and as this is involved with memory it-is known as memory interfacing device. The basic concepts of memory interfacing involve three different tasks. The microprocessor should be able to read from or write into the specified register. To do this it must be able to select the required chip, identify the required register and it must enable the appropriate buffers.

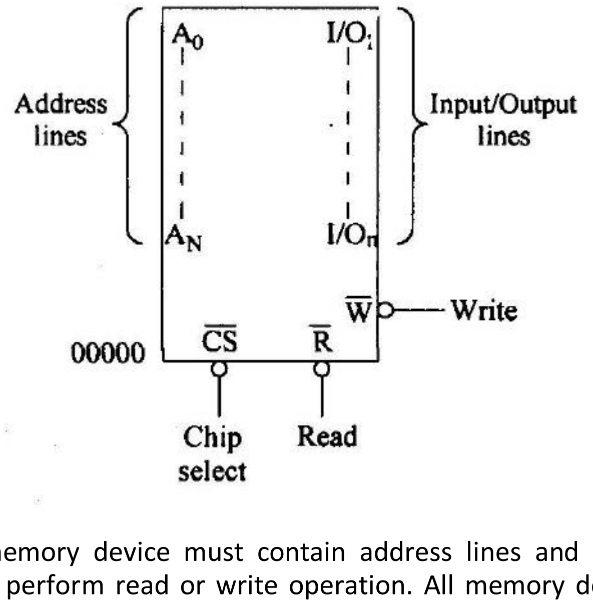


Fig. 3.13 Simple memory device

Any memory device must contain address lines and Input, output lines, selection input, control input to perform read or write operation. All memory devices have address inputs that select memory location with in the memory device. These lines are labeled as A_O ...... A_N. The number of address lines indicates the total memory capacity of the memory device. A 1K memory requires 10 address lines A_0-A_9. Similarly a 1MB requires 20 lines A_0-A_{19} (in the case of 8086). The memory devices may have separate I/O lines or a common set of bidirectional I/O lines. Using these lines data can be transferred in either direction. Whenever output buffer is activated the operation is read whenever input buffers are activated the operation is write. These lines are labelled as I/O,......... I/O_n or D_0 ....D_n. The size of a memory location is dependent upon the number of data bits. If the number of data lines are eight D_0 - D_7 then 8 bits or 1 byte of data can be stored in each location. Similarly if numbers of data bits are 16 (D_0 - D_{15}) then the memory size is 2 bytes. For example 2K x 8 indicates there are 2048 memory locations and each memory location can store 8 bits of data.

Memory devices may contain one or more inputs which are used to select the memory device or to enable the memory device. This pin is denoted by $\overline{CS}$ (Chip select) or $\overline{CE}$ (Chip enable). When this pin is at logic '0' then only the memory device performs a read or a write operation. If this pin is at logic '1' the memory chip is disabled. If there are more than one $\overline{CS}$ input then all these pins must be activated to perform read or write operation.

All memory devices will have one or more control inputs. When ROM is used we find $\overline{OE}$ output enable pin which allows data to flow out of the output data pins. To perform this task both $\overline{CS}$ and $\overline{OE}$ must be active. A RAM contains one or two control inputs. They are R / W or $\overline{RD}$ and $\overline{WR}$. If there is only one input R/ W then it performs read operation when R/ $\overline{W}$ pin is at logic 1. If it is at logic 0 it performs write operation. Note that this is possible only when $\overline{CS}$ is also active.

4.4 Memory Interface using RAMS, EPROMS and EEPROMS

(Ref: Advanced Microprocessors and Peripherals by A.K. Ray & K.M. Bhurchandi, McGraw-Hill, 2nd Edition.P.158-164)

Semiconductor Memory Interfacing:

Semiconductor memories are of two types, viz. RAM (Random Access Memory) and ROM (Read Only Memory).

Static RAM Interfacing:

The semiconductor RAMs are of broadly two types-static RAM and dynamic RAM. The semiconductor memories are organised as two dimensional arrays of memory locations. For example, 4K x 8 or 4K byte memory contains 4096 locations, where each location contains 8-bit data and only one of the 4096 locations can be selected at a time. Obviously, for addressing 4K bytes of memory, twelve address lines are required. In general, to address a memory location out of N memory locations , we will require at least *n* bits of address, i.e. *n* address lines where $n = \text{Log}_2$ N. Thus if the microprocessor has *n* address lines, then it is able to address at the most N locations of memory, where 2^n = N. However, if out of *N* locations only *P* memory locations are to be interfaced, then the least significant *p* address lines out of the available *n* lines can be directly connected from the microprocessor to the memory chip while the remaining *(n-p)* higher order address lines may be used for address decoding (as inputs to the chip selection logic). The memory address depends upon the hardware circuit used for decoding the chip select ($\overline{CS}$). The output of the decoding circuit is connected with the $\overline{CS}$ pin of the memory chip. The general procedure of static memory interfacing with 8086 is briefly described as follows:

1. Arrange the available memory chips so as to obtain 16-bit data bus width. The upper 8-bit bank is called 'odd address memory bank' and the lower 8-bit bank is called 'even address memory bank'.

2. Connect available memory address lines of memory chips with those of the microprocessor and also connect the memory $\overline{RD}$ and $\overline{WR}$ inputs to the corresponding processor control signals. Connect the 16-bit data bus of the memory bank with that of the microprocessor 8086.

3. The remaining address lines of the microprocessor, $\overline{BHE}$ and A_0 are used for decoding the required chip select signals for the odd and even memory banks. $\overline{CS}$ of memory is derived from the O/P of the decoding circuit.

As a good and efficient interfacing practice, the address map of the system should be continuous as far as possible, i.e. there should be no windows in the map. A memory location should have a single address corresponding to it, i.e. absolute decoding should be preferred, and minimum hardware should be used for decoding. In a number of cases, linear decoding may be used to minimise the required hardware.Let us now consider a few example problems on memory interfacing with 8086.

Problem 5.1

Interface two 4K × 8 EPROMS and two 4K × 8 RAM chips with 8086. Select suitable maps.

Solution We know that, after reset, the IP and CS are initialised to form address FFFF0H. Hence, this address must lie in the EPROM. The address of RAM may be selected any where in the 1MB address space of 8086, but we will select the RAM address such that the address map of the system is continuous, as shown in Table 5.1.

Table 5.1 *Memory Map for Problem 5.1*

Address	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_{09}	A_{08}	A_{07}	A_{06}	A_{05}	A_{04}	A_{03}	A_{02}	A_{01}	A_{00}
FFFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
							EPROM						8K × 8							
FE000H	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
FDFFFH	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
							RAM					8K × 8								
FC000H	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Total 8K bytes of EPROM need 13 address lines $A_0 - A_{12}$ (since $2^{13} = 8K$). Address lines $A_{13} - A_{19}$ are used for decoding to generate the chip select. The $\overline{BHE}$ signal goes low when a transfer is at odd address or higher byte of data is to be accessed. Let us assume that the latched address, $\overline{BHE}$ and demultiplexed data lines are readily available for interfacing. Figure 5.1 shows the interfacing diagram for the memory system.

The memory system in this example contains in total four 4K × 8 memory chips.

The two 4K × 8 chips of RAM and ROM are arranged in parallel to obtain 16-bit data bus width. If A_0 is 0, i.e. the address is even and is in RAM, then the lower RAM chip is selected indicating 8-bit transfer at an even address. If A_0 is 1, i.e. the address is odd and is in RAM, the $\overline{BHE}$ goes low, the upper RAM chip is selected, further indicating that the 8-bit transfer is at an odd address. If the selected addresses are in ROM, the respective ROM chips are selected. If at a time A_0 and $\overline{BHE}$ both are 0, both the RAM or ROM chips are selected, i.e. the data transfer is of 16 bits. The selection of chips here takes place as shown in Table 5.2.

Table 5.2 *Memory Chip Selection for Problem 5.1*

Decoder I/P →	*A_2*	*A_1*	*A_0*	*Selection/*
Address/$\overline{BHE}$ →	*A_{13}*	*A_0*	*$\overline{BHE}$*	*Comment*
Word transfer on $D_0 - D_{15}$	0	0	0	Even and odd addresses in RAM
Byte transfer on $D_7 - D_0$	0	0	1	Only even address in RAM
Byte transfer on $D_8 - D_{15}$	0	1	0	Only odd address in RAM
Word transfer on $D_0 - D_{15}$	1	0	0	Even and odd addresses in ROM
Byte transfer on $D_0 - D_7$	1	0	1	Only even address in ROM
Byte transfer on $D_8 - D_{15}$	1	1	0	Only odd address in ROM

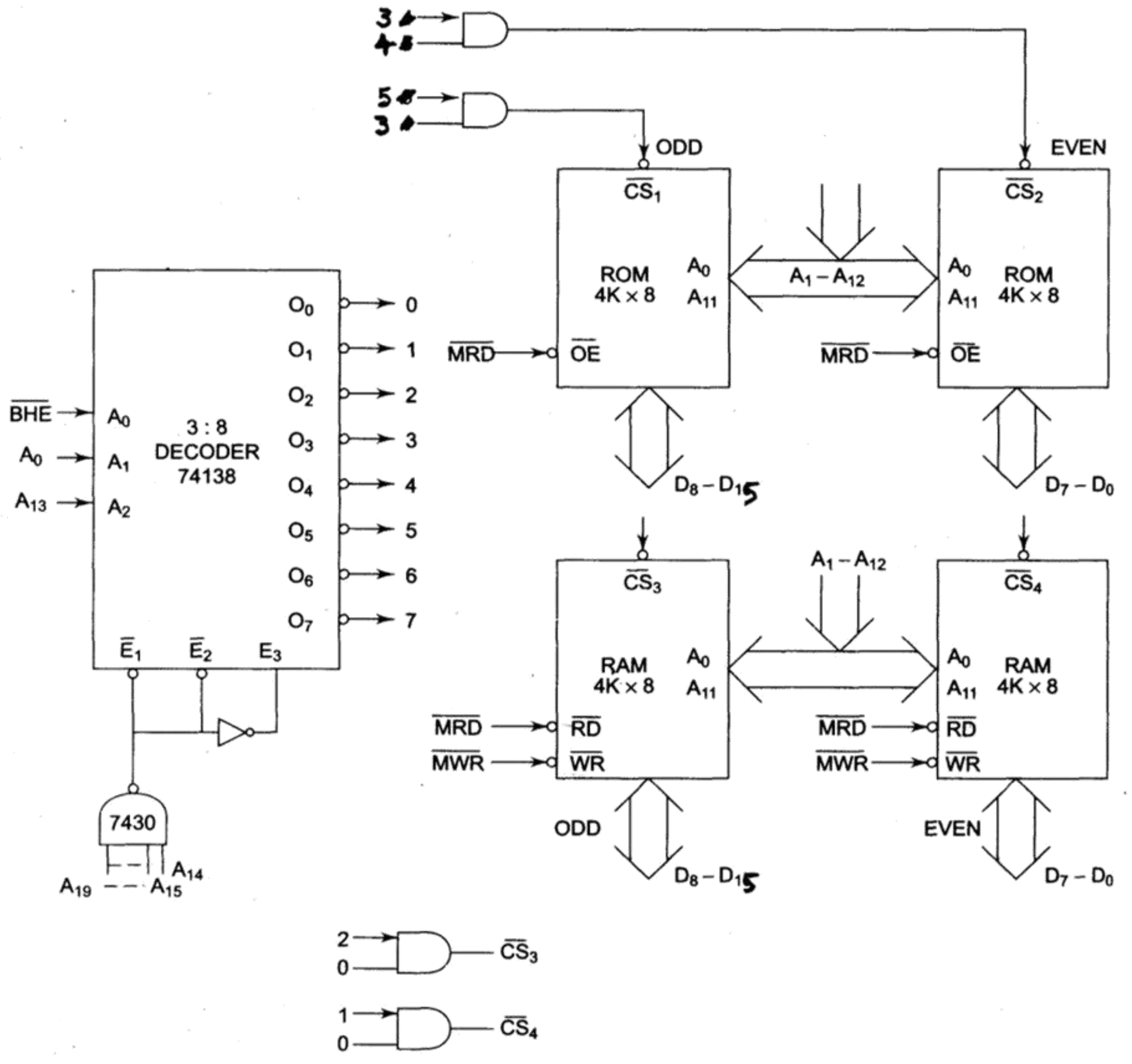


Fig. 5.1 *Interfacing Problem 5.1*

Problem 5.2

Design an interface between 8086 CPU and two chips of 16K × 8 EPROM and two chips of 32K × 8 RAM. Select the starting address of EPROM suitably. The RAM address must start at 00000H.

Solution: The last address in the map of 8086 is FFFFFH. After resetting, the processor starts from FFFF0H. Hence this address must lie in the address range of EPROM. Figure 5.2 shows the interfacing diagram, and Table 5.3 shows complete map of the system.

Table 5.3 *Address Map for Problem 5.2*

Addresses	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_{09}	A_{08}	A_{07}	A_{06}	A_{05}	A_{04}	A_{03}	A_{02}	A_{01}	A_{00}
FFFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
								32KB		EPROM										
F8000H	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0FFFFH	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
							64KB RAM													
00000H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

It is better not to use a decoder to implement the above map because it is not continuous, i.e. there is ome unused address space between the last RAM address (0FFFFH) and the first EPROM address F8000H). Hence the logic is implemented using logic gates, as shown in Fig. 5.2.

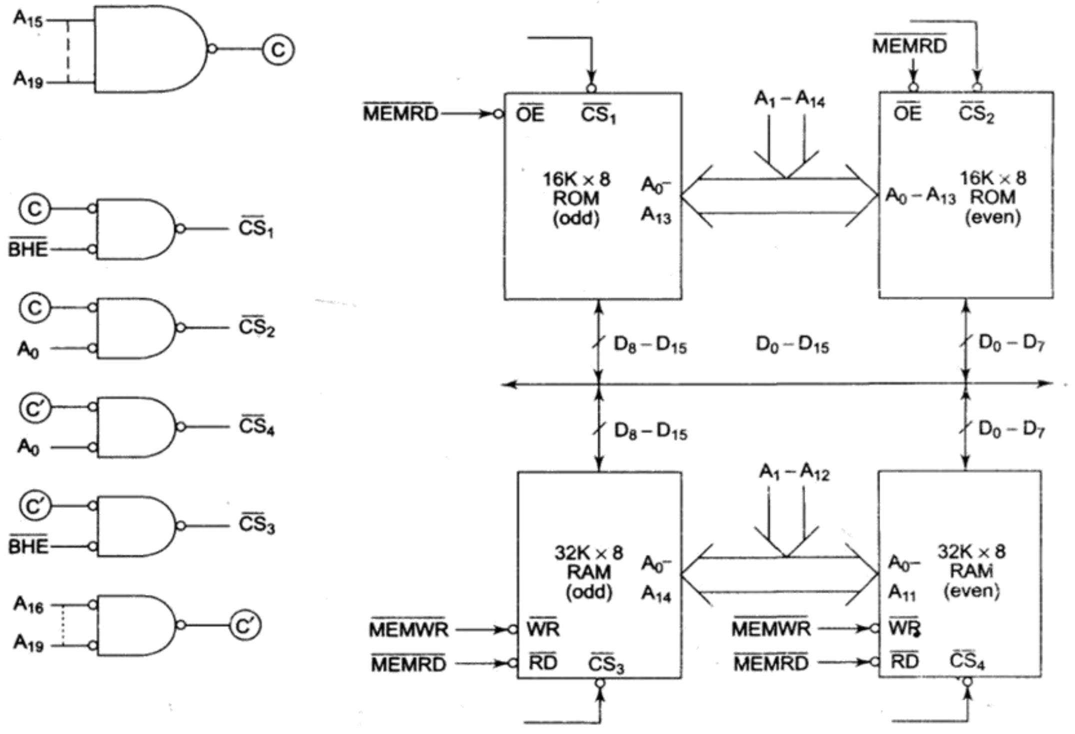


Fig. 5.2 *Interfacing Problem 5.2*

Problem 5.3

It is required to interface two chips of 32K × 8 ROM and four chips of 32K × 8 RAM with 8086, according to the following map.

```
ROM 1 and 2 F0000H - FFFFFH, RAM 1 and 2 D0000H - DFFFFH
RAM 3 and 4 E0000H - EFFFFH
```

Show the implementation of this memory system.

Solution Let us write the memory map of the system as shown in Table 5.6.

The implementation of the above map is shown in Fig. 5.3 using the same technique as in Problem 5.1 and Problem 5.2. All the address, data and control signals are assumed to be readily available.

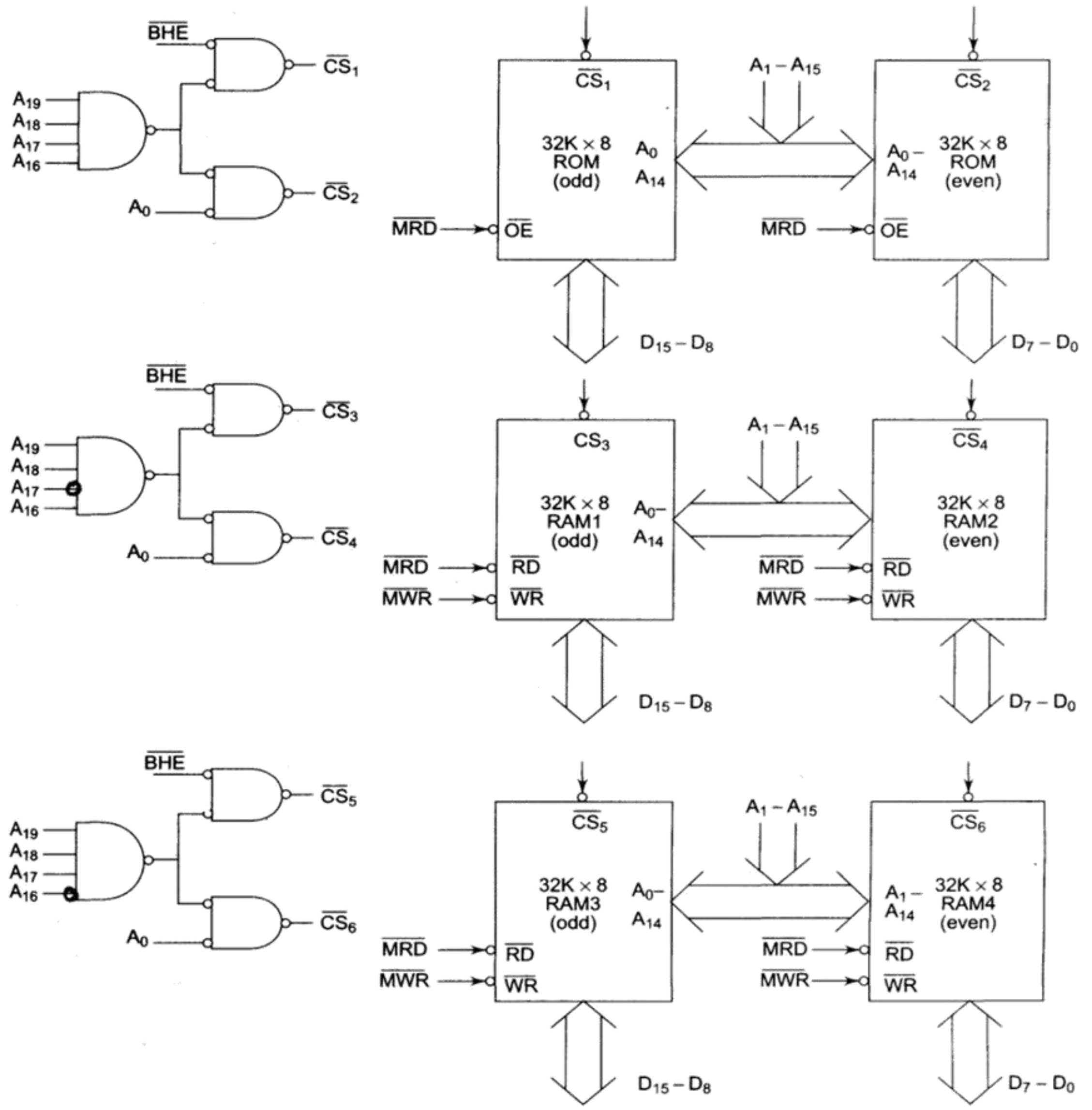


Fig. 5.3 *Interfacing Problem 5.3*

SERIAL COMMUNICATION STANDARDS

(Ref: Interfacing through Microprocessors by K. Subba Rao, Hi-tech publishers, P. 250-260)

Most of devices are parallel in nature. These devices transfer data simultaneously on data lines. But parallel data transfer process is very complicated and expensive. Hence in some situations the serial I/O mode is used where one bit is transferred over a single line at a time. In this type of transmission parallel word is converted into a stream of serial bits which is known as parallel to serial conversion. The rate of transmission in serial mode is BAUD, i.e., bits per second. The serial data transmission involves starting, end of transmission, error verification bits along with the data. Any serial I/O involves the following concepts.

(a) Interfacing requirements (b) Alphanumeric codes (c) Transmission format (d) Error checks in data communication (e) Data communication over lines (f) Standards in serial I/O

The microprocessor has to identify the port address to perform read or write operation. Serial I/O uses only one data line, chip select, read, write control signals.

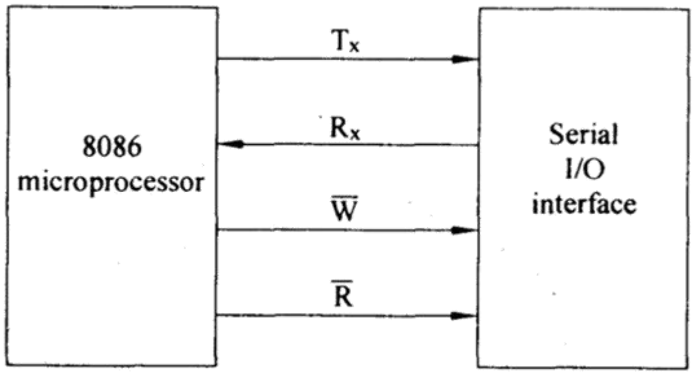
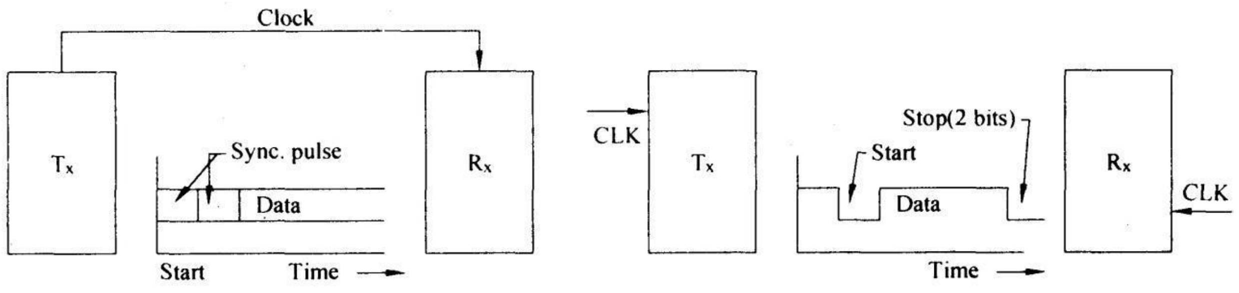


Fig. 5.1 Block diagram of serial I/O interfacing

Data transfer takes place using ASCII code (American standard code for Information Interchange) which is 7 bit code with 128 combinations. The data can be transmitted by taking various parameters into consideration such as synchronization or asynchronization, direction of data flow speed, errors, medium of data transmission etc. In synchronous transmission both transmitter and receiver operate, in synchronous to each other.

Synchronization used for high speed operations. In asynchronous data transmission data is transmitted between Start and Stop bits with logic 1 as mark logic 0 as space. In asynchronous we get around 11 bits for data transmission one start, 8 bits of data, 2 stop bits. A synchronous data transmission is used for less than 20 Kbits /second transmission.



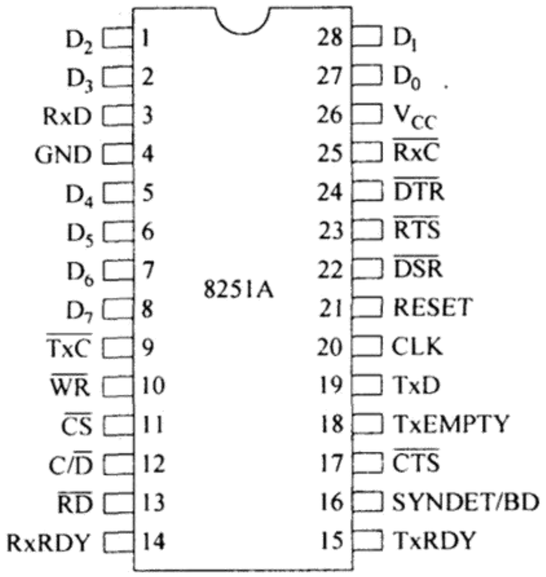
(a) Synchronous data transmission (b) Asynchronous data transmission

Fig. 5.2 Data communication

DIFFERENCE BETWEEN SYNCHRONOUS AND ASYNCHRONOUS TRANSMISSION

S.No.	Synchronous	Asynchronous
1.	Same clock pulse is applied to both Tx & Rx simultaneously	Different clock pulses are applied to Tx & Rx sepeartely
2.	Only hardware is required to implement this.	Both hardware and software are requied for this.
3.	Group or a set of characters can be transmitted at a time.	Only one character is transmitted at a time.
4.	Synchronous pulses are required	Synchronous pulses are not required but uses start and stop bits.
5.	Uses for high speed Tx.	Used for low speed Tx.

5.2 UNIVERSAL SYNCHRONOUS/ASYNCHRONOUS RECEIVER/TRANSMITTER (USART)



Pin diagram of 8521

The 8251A is Universal Synchronous/Asynchronous Receiver/Transmitter (USART) designed for the data communication with Intel's family of microprocessor such as 8085, 8086 and 8088. Like other I/O devices in a microcomputer system, its functional configuration is programmed by the system's software for maximum flexibility. The USART accepts data characters from the CPU in the parallel format and converts them into continuous serial data stream for transmission. Simultaneously, it can receive serial data streams arid convert them into parallel data characters for the CPU. The CPU can read the complete status of USART at any time, these includes data transmission errors, control signals etc.

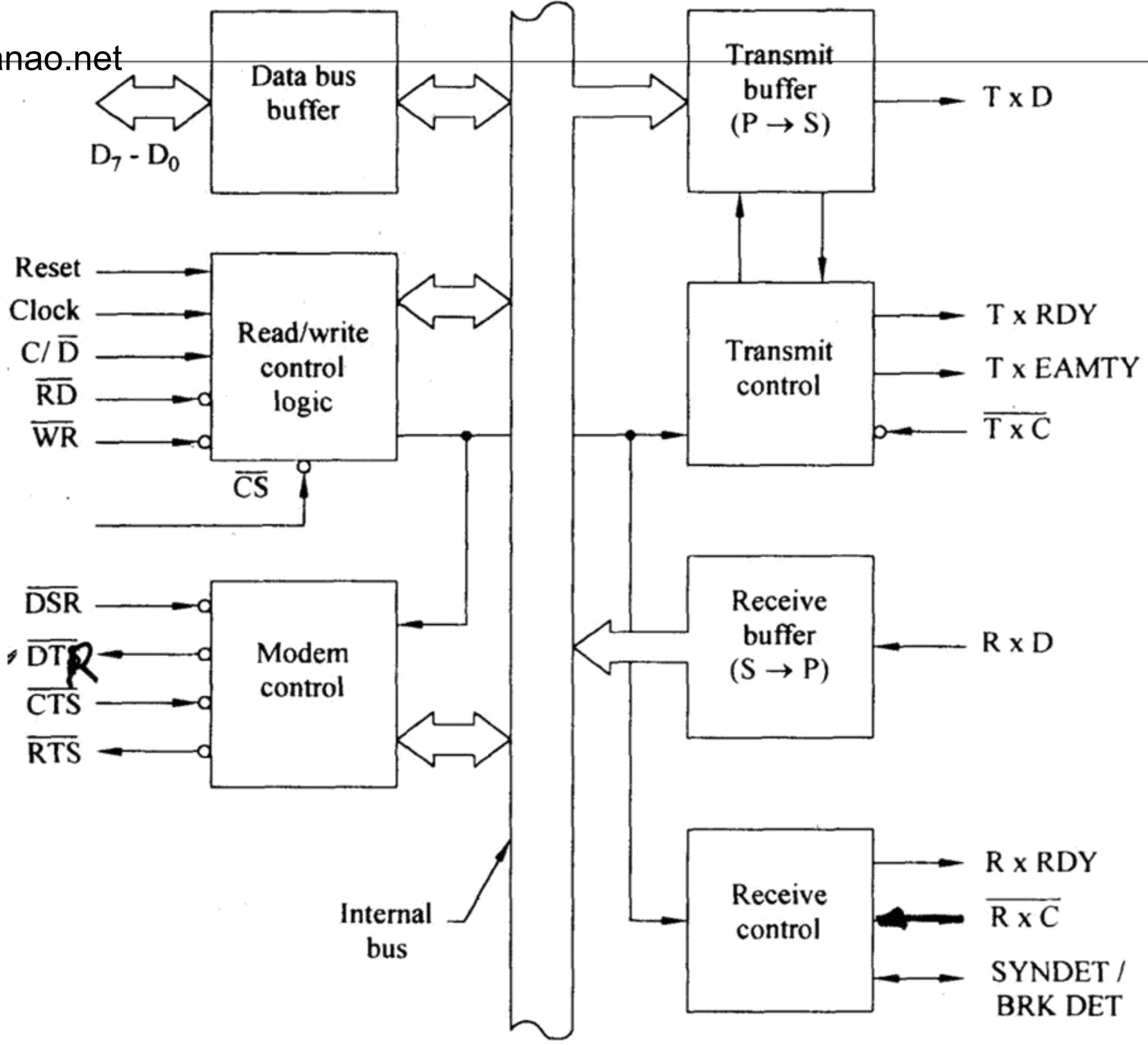


Fig. 5.7 Block diagram of 8251

Fig. 5.7 shows the block diagram of 8251 A. The block diagram shows all the elements of a programmable chip; it includes the interfacing signals, the control register and the status register. The functions of various blocks are described below:

(A) Data bus buffer: This 3-state, bidirectional buffer is used to interface the 8251A to the system data bus. Data is transmitted or received by the buffer upon execution of input and output instruction of the CPU Command words and status information are also transferred through the data bus buffer. The command, status and data in and data out are separate 8-bit registers to provide double buffering.

The functional block accepts inputs form the control bus and generates control signals for overall device operation. It contains the control word register and command word register that store the various control formats for the device functional definition.

(B) Read/Write logic and Registers:

This section includes R/W control logic, six input signals, control logic, and three buffer registers; data register, control register and status register. The input signals to control logic are as follows:

RESET: A high on this input forces the 8251A into an idle mode. The device will remain at idle until a new set of control words is written into the 8251A to program its functional definition.

A command reset operation also puts the device into the idle state.

CLK (Clock): The CLK input is used to generate internal device timing and is normally connected to the phase 2 (TTL) output of the Clock Generator. No external inputs or outputs are referenced to CLK but the frequency of CLK must be greater than 30 times the Receiver or Transmitter data bit rates.

$\overline{\text{WR}}$ (Write): A "low" on this input informs the 8251A that the CPU is writing data or control words to the 8251A.

$\overline{\text{RD}}$ (Read): A "low" on this input informs the 8251A that the CPU is reading data or status information from the 8251A.

C/$\overline{\text{D}}$	RD	WR	CS	
0	0	1	0	8251 data-data bus
0	1	0	0	Data bus-8251A data
1	0	1	0	Status-data bus
1	1	0	0	Data bus-control
×	1	1	0	Data bus-3 state
×	×	×	1	Data bus 3 state

C/$\overline{\text{D}}$ (Control/Data): This input, in conjuction with the $\overline{\text{WR}}$ and $\overline{\text{RD}}$ inputs informs the 8251A that the word on the Data bus is either a data character, control word or status information.

1 = CONTROL/STATUS; 0 = DATA

$\overline{\text{CS}}$ (Chip Select) : A "low" on this input selects the 8251A. No reading or writing will occur unless the device is selected. When CS is high, the Data Bus is in the float state and RD and WR have no effect on the chip.

(C) Modem Control:

The 8251A has a set of control inputs and outputs that can be used to simplify the interface to almost any modem. The modem control signals are general purpose in nature and can be used for functions other than modem control, if necessary.

$\overline{\text{DSR}}$ (Data Set Ready) : The $\overline{\text{DSR}}$ input signal is a general-purpose, 1-bit inverting input port. Its condition can be tested by the CPU using a Status Read operation. The DSR input is normally used to test modem condition such as Data Set Ready.

$\overline{DTR}$ (Data Terminal Ready): The $\overline{DTR}$ output signal is a general-purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command instruction word. The $\overline{DTR}$ output signal is normally used for modem control such as Data Terminal Ready.

$\overline{RTS}$ (Request to Send): The $\overline{RTS}$ output signal is a general-purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command instruction word. The $\overline{RTS}$ output signal is normally used for modem control such as Request to send.

$\overline{CTS}$ (Clear to Send): A "low" on this input enables the 8251A to transmit serial data if the Tx Enable bit in the Command byte is set to a "one", if either a T× Enable off or $\overline{CTS}$ off condition occurs while the T× is in operation, the T× will transmit all the data in the USART, written prior to T× Disable command before shutting down.

(D) Transmitter Buffer:

The transmitter Buffer accepts parallel data from the Data Bus Buffer, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin on the falling edge of $\overline{T x C}$. The transmitter will begin transmission upon being enabled, if $\overline{CTS}$ = 0. The $\overline{T x C}$ line will be held in the marking state immediately upon a master Reset or when Tx Enable or $\overline{CTS}$ is off or the transmitter is empty.

(E) Transmitter Control:

The Transmitter Control manages all activities associated with the transmission of serial data. It accepts and issues signals both externally and internally to accomplish this function.

T x RDY (Transmitter Ready): This output signals the CPU that the transmitter is ready to accept a data character. The T x RDY output pin can be used as an interrupt to the system, since it is masked by T x Enable; or, for Polled operation, the CPU can check T x RDY using a Status Read operation. T x RDY is automatically reset by the leading edge of $\overline{WR}$ when a data character is loaded from the CPU.

T x E (Transmitter Empty): When the 8251A has no character to send, the T x empty will go high. It resets upon receiving a character from CPU if the transmitter is enabled, Tx empty remains high when the transmitter is disabled. T x Empty can be used to indicate the end of transmission node, so that the CPU knows when to turn around in the half-duplex operation mode.

In the synchronous mode, a high on this output indicates that a character has not been loaded and the SYNC character or characters are about to be are being transmitted as filters. T x Empty does not go low when the Sync characters are being shifted out.

T x C (Transmitter Clock): The Transmitter Clock control the rate at which the character is to be transmitted. In the Synchronous transmission mode, the Baud Rate (1x) is equal to the T x C frequency. In Asynchronous transmission mode, the baud rate is a fraction of the actual T x C frequency. A portion of the mode instruction selects this factor it can be 1,1/16 or 1/64 the T x C.

For example

If Baud rate equals 220 Baud

$\overline{TXC}$ equals 220 Hz in the 1x mode.

$\overline{TXC}$ equals 3.52 KHz in the 16x mode.

$\overline{TXC}$ equals 14.08 KHz in the 64x mode.

The falling edge of $\overline{TXC}$ shifts the serial data out of the 8251A.

(F) Receiver Buffer:

The Receiver accepts serial data, converts this serial input to parallel format checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU. Serial data is input to R × D pin, and is clocked in on this rising edge of $\overline{\text{R x C}}$.

(G) Receiver Control:

This functional block shown in Fig. manages all receiver - related activities which consists of the following features.

The R × D initialization circuits prevents the 8251A from mistaking and unused input line for an active low data line in the break condition. Before starting to receive serial characters on the R × D line, a valid '1' must first be detected after a chip master reset. Once this has been determined, a search for a valid low bit (start bit) is enabled. This feature is only active in the asynchronous mode, and is only done once for each master reset.

The false start bit detection circuit prevents false starts due to a transient noise spike by first detecting the falling edge and then strobing the nominal center of the start (R × D = low).

Parity error detection sets the corresponding status bit.

The framing error status bit is set if the stop bit is absent at the end of the data byte (asynchronous mode).

R × RDY (Receiver Ready) : This output indicates the the 8251A contains a character that is ready to be input to the CPU. R × RDY can be connected to the interrupt structure of the CPU or, for polled operation, the CPU can check the condition of R × RDY using a status read operation.

R × Enable, when off, holds R × RDY in the reset condition. For asynchronous mode, to set R × RDY, the receiver must be enabled to sense a start bit and a complete character must be assembled and transferred to the data output register.

Failure to read the received character from the R × Data output register prior to the assembly of the next R × data character will set overrun condition error and the previous character will be written over and lost. If the R × data is being read by the CPU when the internal transfer is occurring, overrun error will be set and the old character will be lost.

$\overline{\text{R x C}}$ (Receiver Clock) : The receiver clock controls the rate at which the character is to be received. In synchronous mode, the baud rate (1×) is equal to the actual frequency of $\overline{\text{R x C}}$. In asynchronous mode, the baud rate is a fraction of the actual $\overline{\text{R x C}}$ frequency. A portion of the mode instruction selects this factor: 1, 1/16 of 1/64 the $\overline{\text{R x C}}$.

For Example:

Baud rate equals 200 baud, if

$\overline{\text{R x C}}$ equals 200 Hz in the 1 × mode.

$\overline{\text{R x C}}$ equals 3200 Hz in the 16 × mode.

$\overline{\text{R x C}}$ equals 128 KHz in the 64 × mode.

SYNDET (SYNC Detect/BRKDET Break Detect): This is used in Synchronous Mode for SYNDET and may be used as either input or output, programmable through the Control Word. It is reset to output mode low upon RESET. When used as an output (internal Sync mode), the SYNDET pin will go "high" to indicate that the 8251A has located the SYNC character in the Receive mode. If the 8251A is programmed to use double Sync characters (bi-sync), then SYNDET will go "high" in the middle of the last bit of the second Sync character SYNDET is automatically reset upon a status Read operation.

When used as an input (external SYNC detect mode), a positive going signal will cause the 8251A to start assembling data characters on the rising edge of the next $\overline{\text{R x C}}$. Once in SYNC, the "high" input signal can be removed. When External SYNC Detect is programmed, internal SYNC Detect is disabled.

BREAK (Async Mode Only): This output will go high whenever the receiver remains low through two consecutive stop bit sequences (including the start bits, data bits, and parity bits); Break Detect may also be read as a Status bit. It is reset only upon a master chip Reset of R × Data returning to a "one" state.

INTERFACING STANDARDS

(Ref: Interfacing through Microprocessors by K. Subba Rao, Hi-tech publishers, P. 266)

Serial I/O is used to interface various devices or for connecting various equipment to the system. Common understanding is necessary among various manufacturers such that a standard notation is followed for interfacing these components. These standards may be provided by IEEE or by any standard professional organisation. The serial I/O standards must specify clearly voltage levels, speed of data transfer, length of cables etc. In serial I/O data can be transmitted as either current or voltage 20 mA or 60 mA current loops are used if data is transmitted using current. Current flow takes place when the system is at logic 1. The current flow is stopped when the system is at logic 0. In the current loop method the signals are relatively noise-free and they are best suited for long distance transmission.

RS-232 is developed long before which is used for communication between terminals and modems. Using RS-232C data can be transmitted as voltage. The data terminals equipment and data communication equipment are used to communicate using RS-232C cable. RS-232C is not compatible with TTL logic and cannot be used for long distance transmission.

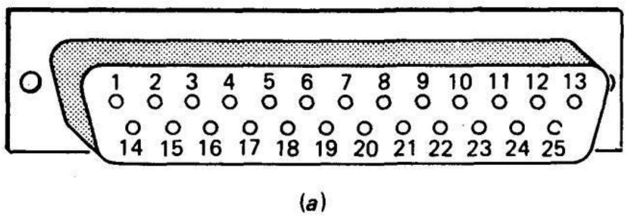
RS-232C Serial Data Standard

(Ref: Microprocessors and interfacing by Douglas V. Hall, 2nd edition, TMH, P.494-495)

OVERVIEW

Modems were developed so that terminals could use phone lines to communicate with distant computers. As we stated earlier, modems and other devices used to send serial data are often referred to as *data communication equipment* or DCE. The terminals or computers that are sending or receiving the data are referred to as *data terminal equipment* or DTE. In response to the need for signal and handshake standards between DTE and DCE, the Electronic Industries Association (EIA) developed EIA standard RS-232C. This standard describes the function of 25 signal and handshake pins for serial-data transfer. It also describes the voltage levels, impedance levels, rise and fall times, maximum bit rate, and maximum capacitance for these signal lines.

RS-232C specifies 25 signal pins, and it specifies that the DTE connector should be a male and the DCE connector should be a female. A specific connector is not given, but the most commonly used connectors are the DB-25P male shown in Figure 14-7a. For systems where many of the 25 pins are not needed, a 9-pin DIN connector such as the DE-9P male connector shown in Figure 14-7b is used.



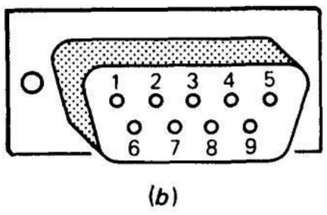


FIGURE 14-7 Connectors often used for RS-232C connections. (*a*) DB-25P 25-pin male. (*b*) DE-9P 9-pin male DIN connector.

The voltage levels for all RS-232C signals are as follows. A logic high, or mark, is a voltage between -3V and -15 V under load (-25 V no load). A logic low or space is a voltage between +3 V and +15 V under load (+ 25 V no load). Voltages such as ±12 V are commonly used.

RS-232C to TTL INTERFACING

Obviously a USART such as the 8251A is not directly compatible with RS-232C signal levels. The standard way to interface between RS-232C and TTL levels is with MCI488 quad TTL-to-RS-232C drivers and MCI489 quad RS-232C-to-TTL receivers shown in Figure 14-8.

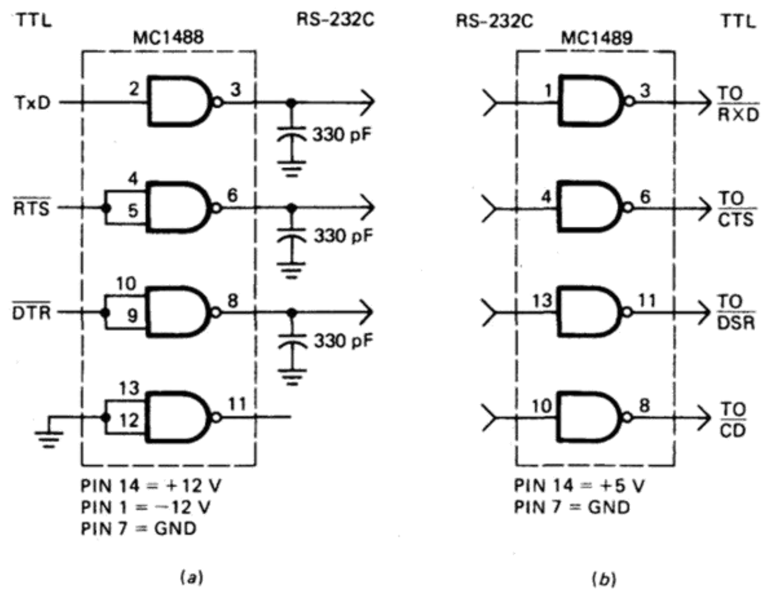


FIGURE 14-8 TTL to RS-232C to TTL signal conversion. (*a*) MC1488 used to convert TTL to RS-232C. (*b*) MC1489 used to convert RS-232C to TTL.

The MCI488s require + and - supplies, but the MCI489s require only+ 5 V. Note the capacitor to ground on the outputs of the MCI488 drivers is to reduce cross talk between adjacent wires, the rise and fall times for RS-232C signals are limited to 30 V/µs.

RS-232C SIGNAL DEFINITIONS

PIN NUMBERS FOR 9 PINS	PIN NUMBERS FOR 25 PINS	COMMON NAME	RS-232C NAME	DESCRIPTION	SIGNAL DIRECTION ON DCE
	1		AA	PROTECTIVE GROUND	–
3	2	TXD	BA	TRANSMITTED DATA	IN
2	3	RXD	BB	RECEIVED DATA	OUT
7	4	$\overline{RTS}$	CA	REQUEST TO SEND	IN
8	5	$\overline{CTS}$	CB	CLEAR TO SEND	OUT
6	6	$\overline{DSR}$	CC	DATA SET READY	OUT
5	7	GND	AB	SIGNAL GROUND (COMMON RETURN)	–
1	8	$\overline{CD}$	CF	RECEIVED LINE SIGNAL DETECTOR	OUT
	9		–	(RESERVED FOR DATA SET TESTING)	–
	10		–	(RESERVED FOR DATA SET TESTING)	–
	11			UNASSIGNED	–
	12		SCF	SECONDARY RECEIVED LINE SIGNAL DETECTOR	OUT
	13		SCB	SECONDARY CLEAR TO SEND	OUT
	14		SBA	SECONDARY TRANSMITTED DATA	IN
	15		DB	TRANSMISSION SIGNAL ELEMENT TIMING (DCE SOURCE)	OUT
	16		SBB	SECONDARY RECEIVED DATA	OUT
	17		DD	RECEIVER SIGNAL ELEMENT TIMING (DCE SOURCE)	OUT
	18			UNASSIGNED	–
	19		SCA	SECONDARY REQUEST TO SEND	IN
4	20	$\overline{DTR}$	CD	DATA TERMINAL READY	IN
	21		CG	SIGNAL QUALITY DETECTOR	OUT
9	22		CE	RING INDICATOR	OUT
	23		CH/CI	DATA SIGNAL RATE SELECTOR (DTE/DCE SOURCE)	IN/OUT
	24		DA	TRANSMIT SIGNAL ELEMENT TIMING (DTE SOURCE)	IN
	25			UNASSIGNED	–

FIGURE 14-9 RS-232C pin names and signal directions.

Figure 14-9 shows the signal names, signal direction, and a brief description for each of the 25 pins denned for RS-232C. For most applications only a few of these pins are used.

Note that the signal direction is specified with respect to the DGE, this convention is part of the standard. Note that there is both a chassis ground (pin 1) and a signal ground (pin 7). To prevent large ac-induced ground currents in the signal ground, these two should be connected together only at the power supply in the terminal or the computer.

The TxD, RxD, and handshake signals shown with common names in Figure 14-9 are the ones most often used for simple systems. These signals control what is called the *primary or forward* communications channel of the modem. Some modems allow communication over a secondary or *backward* channel, which operates in the reverse direction from the forward channel and at a much lower baud rate. Pins 12, 13, 14, 16, and 19 are the data and handshake lines for this backward channel. Pins 15, 17, 21, and 24 are used for synchronous data communication.