PROGRAMMABLE PERIPHERAL INTERFACE 8255

The 8255A is generally seen as 8-bit bidirectional data buffer, which is specially designed to transfer the data with the execution of input output instructions requested by the CPU. It has the ability to use with almost any microprocessor. It consists of three 8-bit bidirectional I/O ports (24I/O lines) which can be configured with their different functional characteristics, each possessing unique features to upgrade the flexibility of 8255.

Ports of 8255A

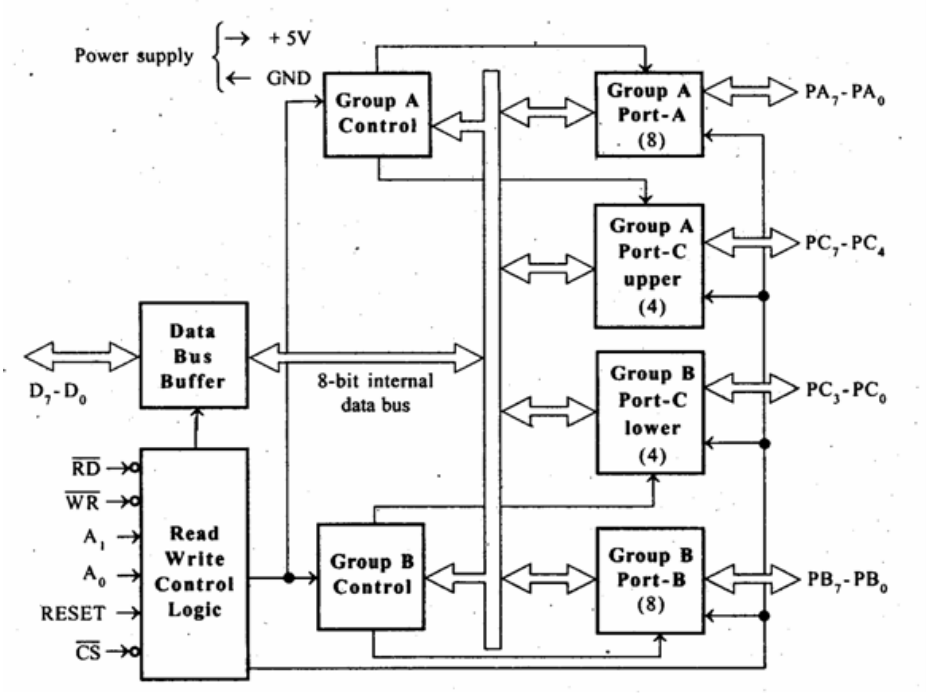
8255A consists of three ports, i.e., PORT A, PORT B, and PORT C.

Port A contains one 8-bit output latch/buffer and one 8-bit input buffer possessing both pull-up and pull-down devices present in Port A.

Port B is similar to PORT A.

Port C can be split into two parts, i.e. PORT C lower (PC0-PC3) and PORT C upper (PC7-PC4) with the help of control word.

These three ports are further classified into two groups, i.e. Group A includes PORT A and upper PORT C. Group B includes PORT B and lower PORT C. These two groups can be programmed in three different modes, i.e. the first mode is named as mode 0, the second mode is named as Mode 1 and the third mode is named as Mode 2.



Data Bus Buffer : It is a tri-state 8-bit bidirectional data buffer, which helps in interfacing the microprocessor to the system data bus. Data is transmitted or received by the buffer with the input output instructions given by the CPU. Control words and status information is also transferred using this bus.

Read/Write Control Logic : This block helps in controlling the transfer of data internally/externally. It accepts the input from the CPU address and control buses. This gives command to both the control groups.

$\overline{CS}$: It stands for Chip Select. It is active-low signal that enables the communication between the 8255A and the CPU. It is connected to the decoded address, and A_0 & A_1 are connected to the microprocessor address lines.
Their result depends on the following conditions –

$\overline{CS}$	A_1	A_0	Result
0	0	0	PORT A
0	0	1	PORT B
0	1	0	PORT C
0	1	1	Control Register
1	X	X	No Selection

It stands for write. This control signal obviously enables the write operation. When this signal becomes low, the microprocessor writes into a selected I/O port or control register.
RESET : This is an active high signal. The control register is cleared by this signal and sets all ports in the input mode.
$\overline{RD}$: It stands for Read option. This control signal helps to perform the Read operation. When the signal becomes low, the microprocessor reads the data from the selected I/O port of the 8255.
A0 and A1 : These input signals work with RD, WR, and one of the control signal. Following is the table showing their various signals with their result.

A_1	A_0	Result
0	0	PORT A
0	1	PORT B
1	0	PORT C

Modes of Operation
There are two basic modes of operation of 8255
1.I/O mode
2.BSR mode

BSR Mode :

In Bit Set-Reset mode, only port-C (PC_0-PC_7) can be used to set (or) reset its individual port bits. It can be set (or) reset the bits by using the control word register.

0	X	X	X	B3	B2	B1	B0

B0 – 0 →Bit Reset,
1 →Bit Set

B3	B2	B1	Port C
0	0	0	PC0
0	0	1	PC1
0	1	0	PC2
0	1	1	PC3
1	0	0	PC4
1	0	1	PC5
1	1	0	PC6
1	1	1	PC7

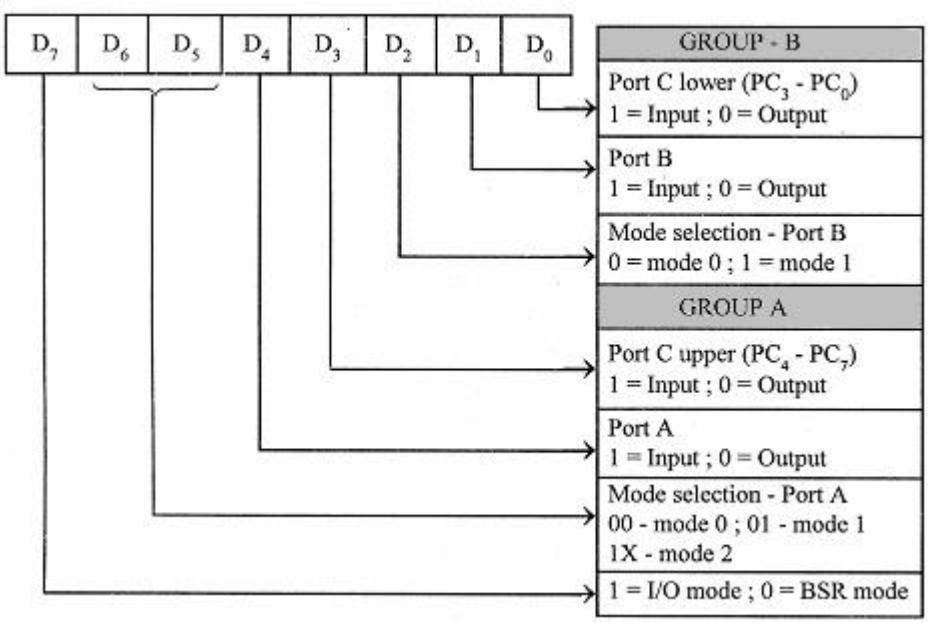
I/O Mode:

In I/O mode, the 8255 ports work as programmable I/O ports. Under I/O mode of operation, there are three modes of operation in 8255 to support different types of application.

1.Mode-0 --> Simple I/O

2.Mode-1 --> Handshake I/O port

3.Mode-2 --> Bidirectional I/O port



Control word Format

Mode-0: Simple I/O

- All the three ports PA, PB & PC can be programmed either as input (or) Output port.
- Output ports are latched, input ports are not latched.
- Port C is used for Handshake signals

Mode-1: Handshake I/O (or) Storbed I/O

This mode is also called as strobed input/Output mode. In this mode, the handshaking signals control the input (or) output action of the specified port.

- Only port A & B can be programmed either as input and output port.
- Handshake signals are exchanged between the processor & peripherals prior to data transfer.
- Port C is used for Handshake Signals.

Mode-1 as input Port:

Control signals for input mode are

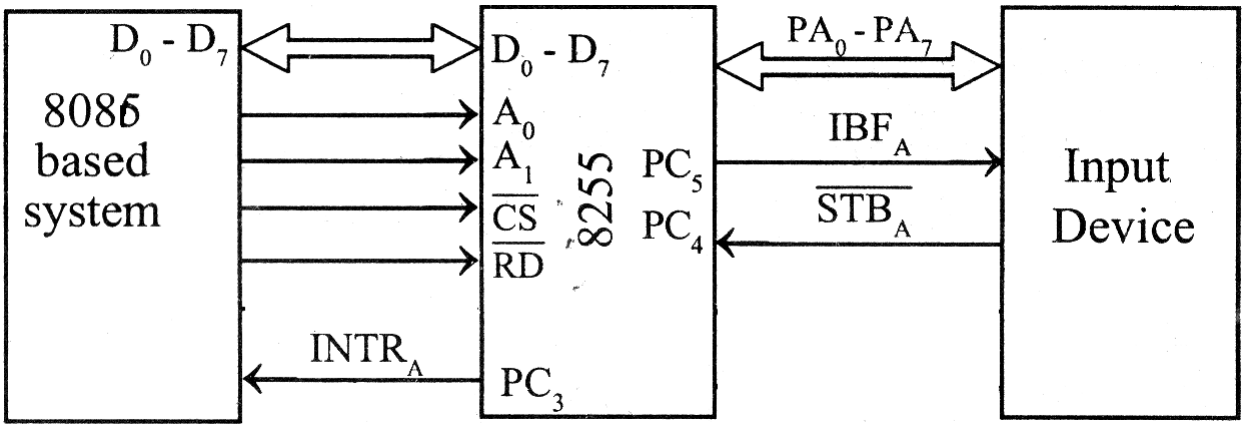
1. $\overline{STB}$: If this falls to low, the date available at 8 bit input port is loaded into input latches.

2. $\overline{IBF}$: If this logic 1, it indicates that data has been loaded into the latches, works as an acknowledgement .

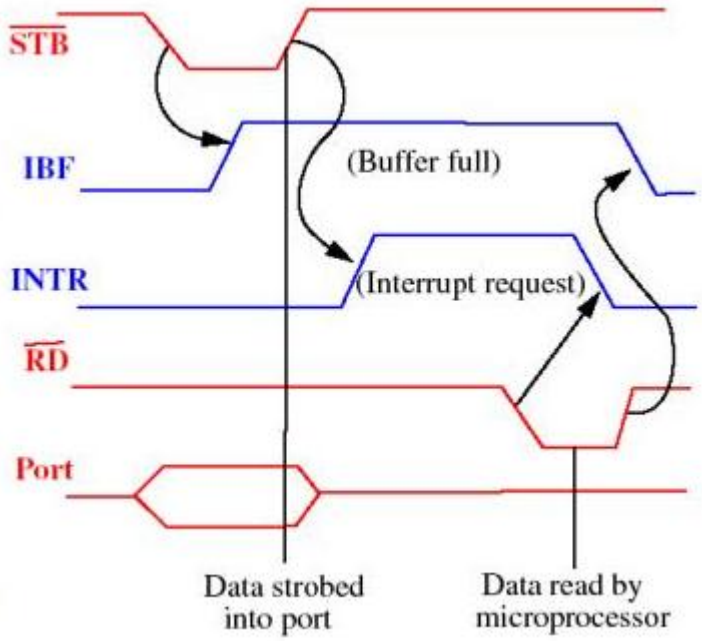
3.INTR : This active high output signal can be used to interrupt the CPU whenever an input device request the service. INTR is set by high at $\overline{STB}$ pin and low at $\overline{IBF}$ pin.

- When $\overline{IBF}_A$ is low, data is placed in PA_0-PA_7 by asserting $\overline{STB}_A$ low.

- When $\overline{STB}_A$ is low, IBF is high, data is latched to the port and $INTR_A$ is set high.
- Processor is thus interrupted, RD is lowed to read the data , $INTR_A$ is resetted, $\overline{IBF}_A$ is low and input device can send the next data.



Mode 1 as Input Port



Timing Diagram of Mode 1 as Input Port

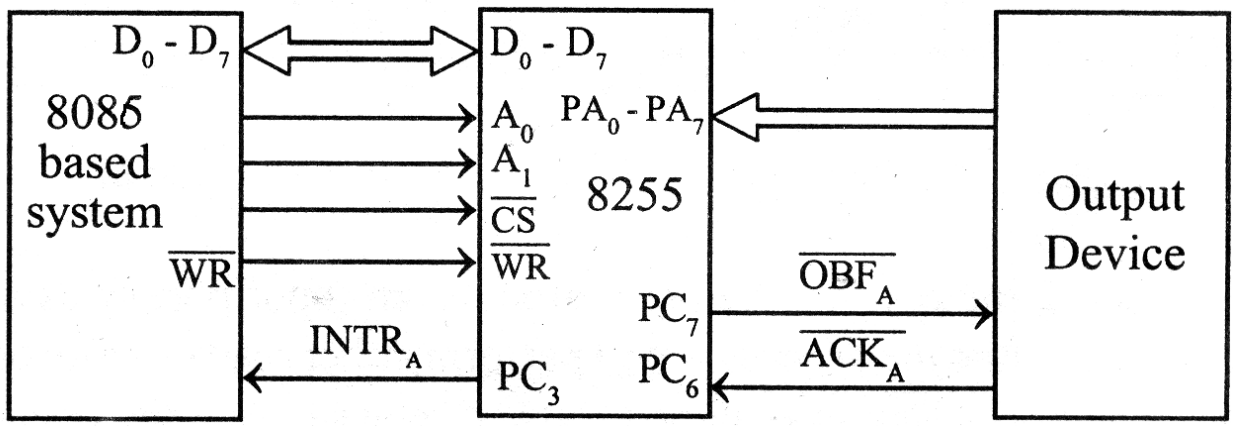
Mode-1 as output port

Control signals for output mode are

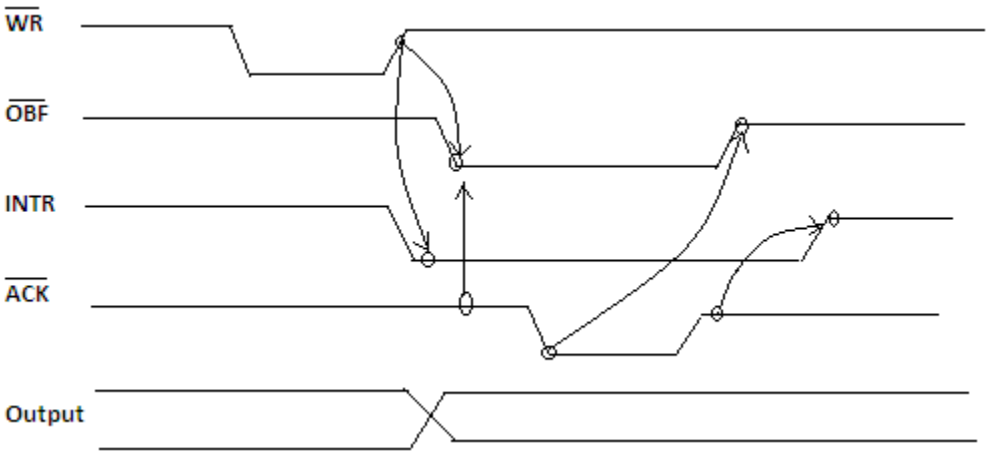
1. $\overline{OBF}$ (Output Buffer Full) – This status signal falls to logic low to indicate that CPU has written data to the specified output port.

2. $\overline{ACK}$ (Acknowledgement Input) – This signal acts as an acknowledgement to be given by an output device . $\overline{ACK}$ signal, whenever low, informs the CPU that the data transferred by the CPU to the output device through the port is received by the output device.

3.INTR (Interrupt Request) – This is the output signal used to interrupt the CPU when an output device acknowledges the data received from the CPU. INTR is set when $\overline{ACK}$, $\overline{OBF}$ and INTR are '1'.



Mode 1 as Output Port



Timing Diagram of Mode 1 as Output Port

Mode 2 : Strobed Bidirectional Port

This mode provides 8255 with an additional feature for communicating with a peripheral device on an 8-bit data bus. Handshaking signals are provided to maintain proper data flow and synchronization between the data transmitter and receiver. The $\overline{RD}$ and $\overline{WR}$ signals decide whether the 8255 is going to operate as an input port (or) output port.

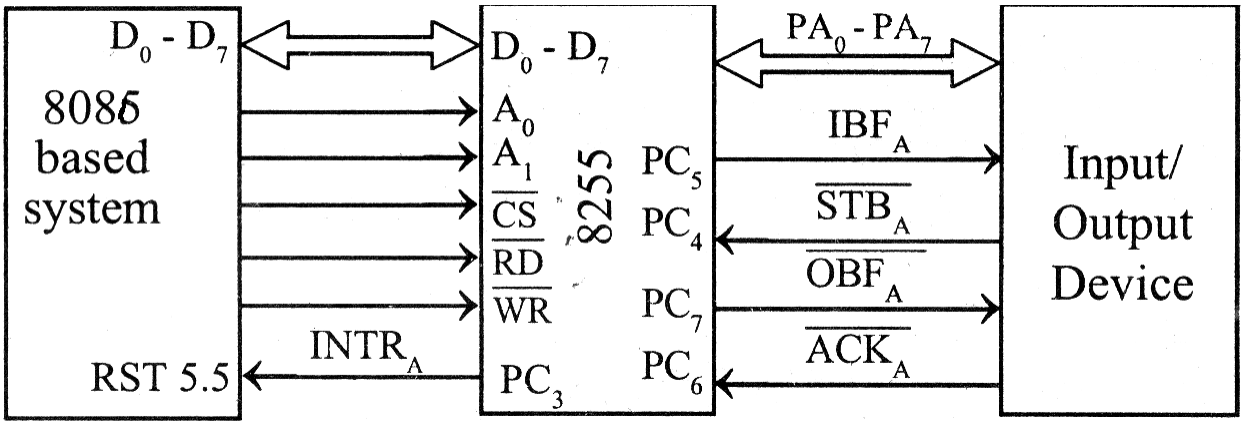
Features:

- Only Port A can be programmed in this mode.
- Inputs and outputs are both latched.
- The 5-bit control port (PC3-PC7) is used for generating/accepting handshake signals for the 8-bit data transfer on port A.
- Used in application such as data transfer between teo computers (or) floppy disk controller interface.

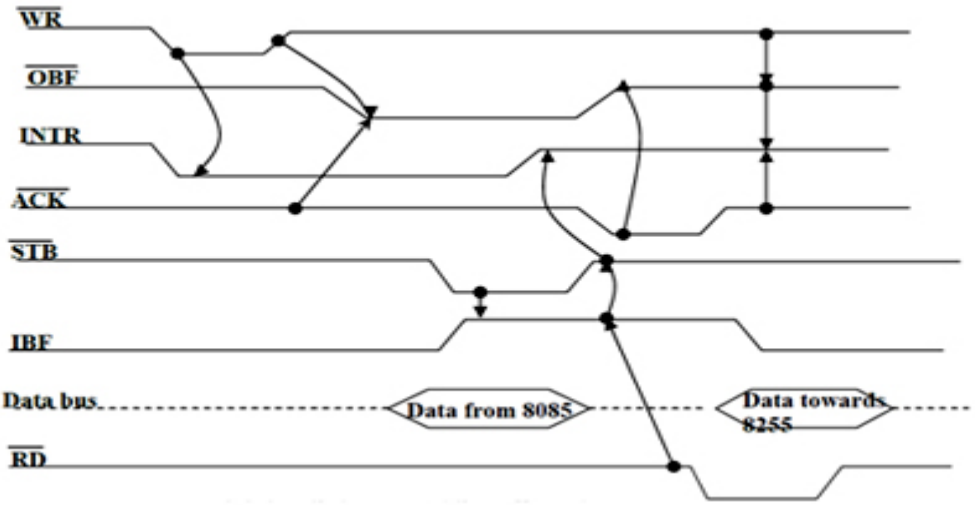
Control Signals

- **INTR** (Interrupt Request) : As in mode 1, it is used to interrupt the microprocessor to ask for transfer of the next data byte to/from it. This signal is used for input (read) as well as output (write) operation.

- . $\overline{OBF}$ (Output Buffer Full): This signal indicates that the CPU has written dat to port A.
- . $\overline{ACK}$ (Acknowledgement) : This control input, acknowledges when the previous data byte is received by the destination and the next byte may be sent by the processor.
- . $\overline{STB}$: If this falls to low, the date available at 8 bit input port is loaded into input latches.
- $\overline{IBF}$: If this logic 1, it indicates that data has been loaded into the latches, works as an acknowledgement .



Mode 2



Timing Diagram of Mode 2

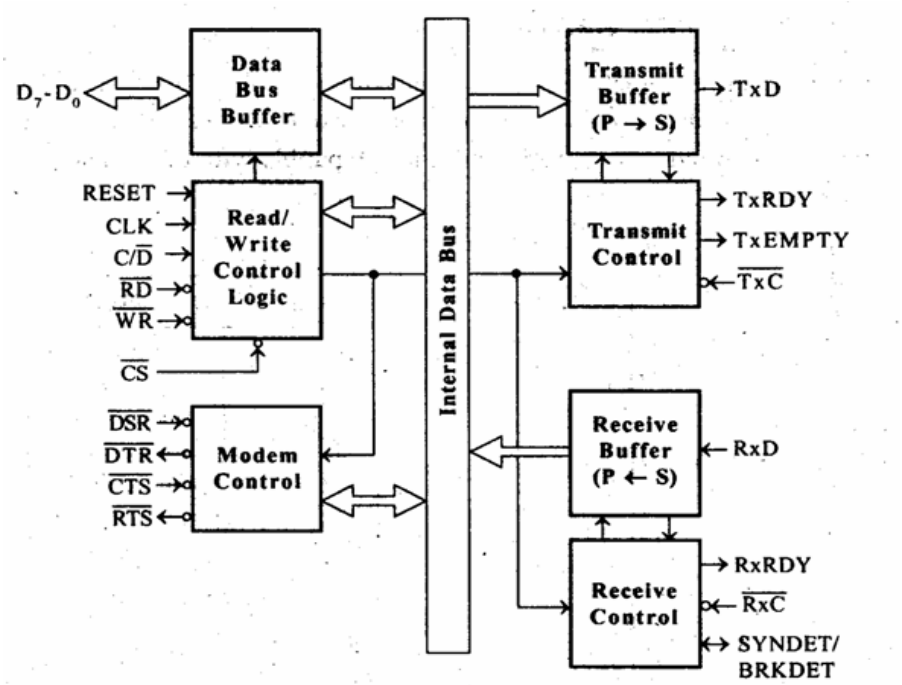
8251A (USART)

- The 8251 is a programmable chip designed for synchronous and asynchronous serial data communication
- The INTEL 8251 is the industry standard Universal Synchronous/Asynchronous Receiver/Transmitter (USART) designed for data communications
- The 8251 is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique
- The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission
- Simultaneously, it can receive serial data streams and convert them into parallel data character for the CPU
- The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU

Block Diagram :

The block diagram includes five section

- Read/write Control Logic
- Transmitter
- Receiver
- Data Bus Buffer and
- Modem Control

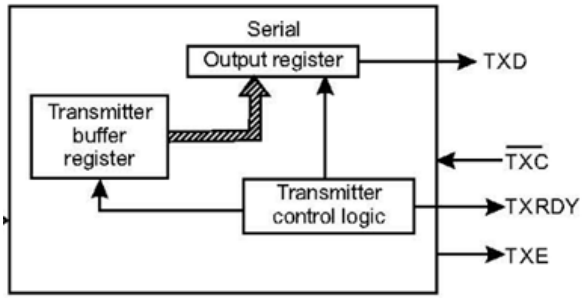


Read/Write control logic:

- The Read/Write Control logic interfaces the 8251A with CPU, determines the functions of the 8251A according to the control word written into its control register.
- It monitors the data flow.
- This section has three registers and they are control register, status register and data buffer.
- The active low signals RD, WR, CS and $C/\overline{D}$ are used for read/write operations with these three registers.

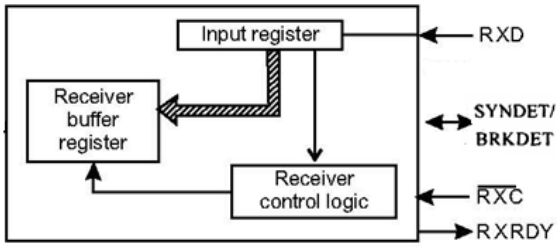
- When $C/\overline{D}$ is high, the control register is selected for writing control word or reading status word.
- When $C/\overline{D}$ is low, the data buffer is selected for read/write operation.
- When the reset is high, it forces 8251A into the idle mode.
- The clock input is necessary for 8251A for communication with CPU and this clock does not control either the serial transmission or the reception rate.

Transmitter section:



- The transmitter section accepts parallel data from CPU and converts them into serial data.
- The transmitter section is double buffered, i.e., it has a buffer register to hold an 8-bit parallel data and another register called output register to convert the parallel data into serial bits.
- When output register is empty, the data is transferred from buffer to output register. Now the processor can again load another data in buffer register.
- If buffer register is empty, then TxRDY is goes to high.
- If output register is empty then TxEMPTY goes to high.
- The clock signal, TxC (low) controls the rate at which the bits are transmitted by the USART.
- The clock frequency can be 1,16 or 64 times the baud rate.

Receiver Section:



- The receiver section accepts serial data and convert them into parallel data
- The receiver section is double buffered, i.e., it has an input register to receive serial data and convert to parallel, and a buffer register to hold the parallel data.
- When the RxD line goes low, the control logic assumes it as a START bit, waits for half a bit time and samples the line again.
- If the line is still low, then the input register accepts the following bits, forms a character and loads it into the buffer register.
- The CPU reads the parallel data from the buffer register.
- When the input register loads a parallel data to buffer register, the RxRDY line goes high.
- The clock signal RxC (low) controls the rate at which bits are received by the USART.
- During asynchronous mode, the signal SYNDET/BRKDET will indicate the break in the data transmission.
- During synchronous mode, the signal SYNDET/BRKDET will indicate the reception of synchronous character.

Data Bus Buffer

This is 8 bit bidirectional register can be addressed as an input and an output port when the $C/\overline{D}$ pin is low

Control/Data pin ($C/\overline{D}$) : When this signal is high, the control register or the status register is addressed; when it is low, the data buffer is addressed. The control register and the status register are differentiated by and signals respectively.

MODEM Control:

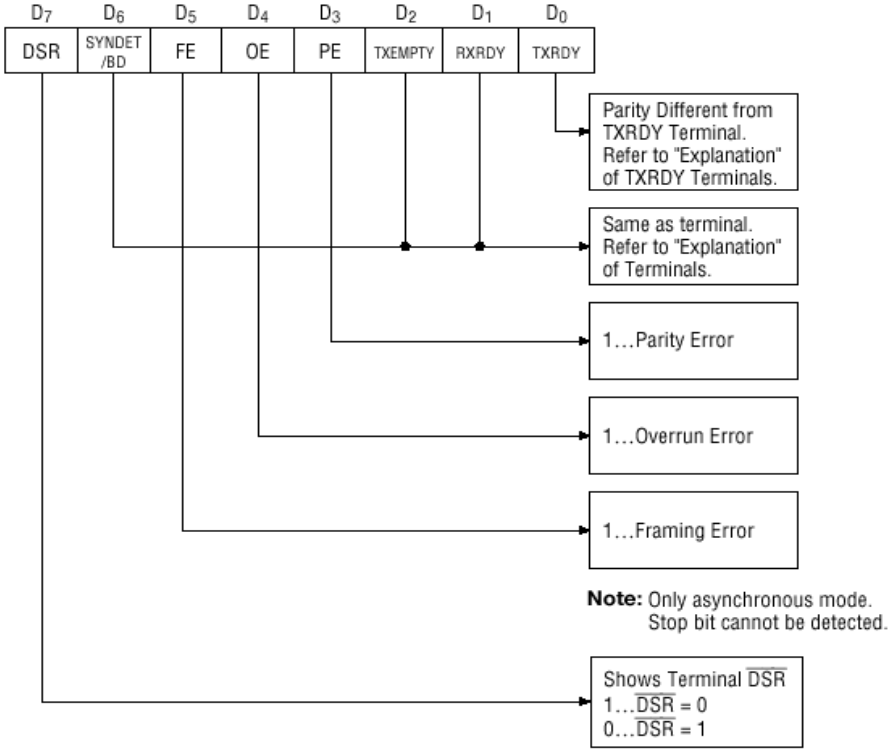
- DSR - Data Set Ready : Checks if the Data Set is ready when communicating with a modem.
- DTR - Data Terminal Ready : Indicates that the device is ready to accept data when the 8251 is communicating with a modem.
- CTS - Clear to Send : If its low, the 8251A is enabled to transmit the serial data provided the enable bit in the command byte is set to '1'.
- RTS - Request to Send Data : Low signal indicates the modem that the receiver is ready to receive a data byte from the modem.

Command register

EH	IR	RTS	ER	SBRK	RxE	DTR	TxE

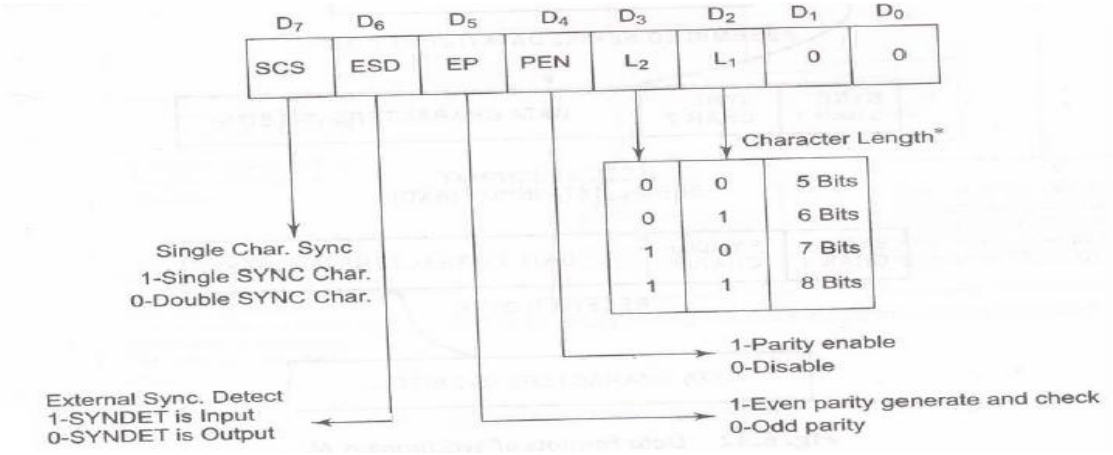
TxE: transmit enable
DTR: data terminal ready
RxE: receiver enable
SBPRK: send break character
ER: error reset
RTS: request to send
IR: internal reset
EH: enter hunt mode

Status Register

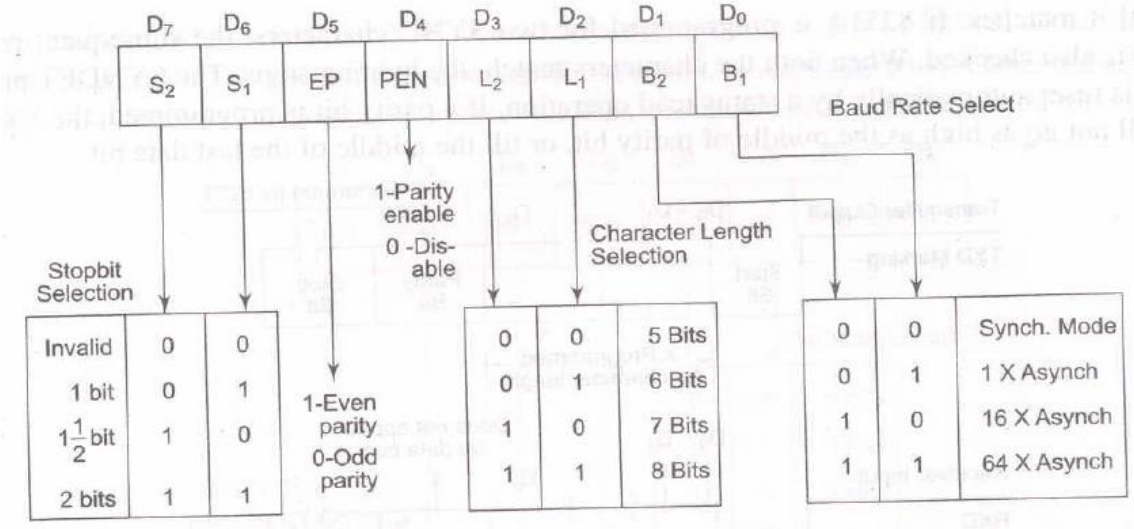


Mode Register :

Synchronous



Asynchronous

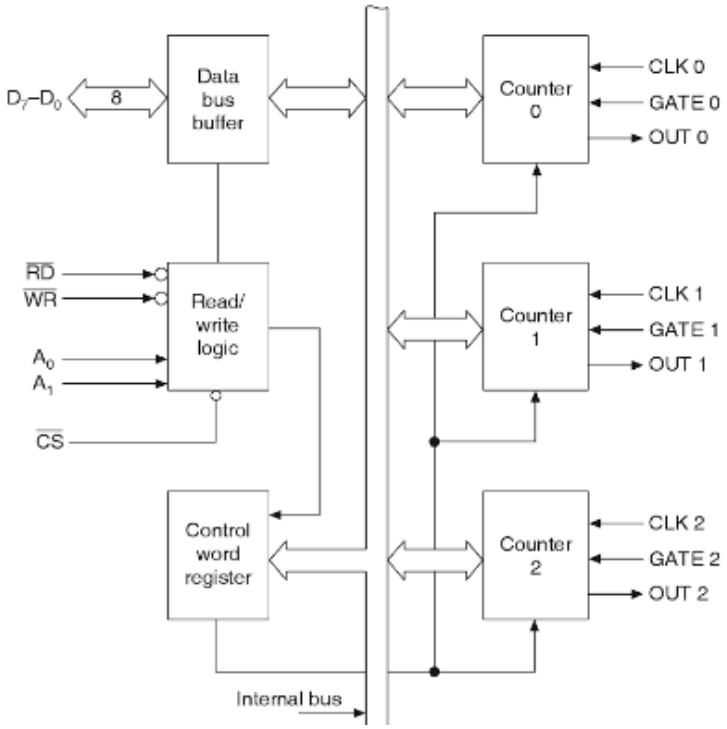


PROGRAMMMABLE INTERVAL TIMER – 8254

8254 is used as a timing and delay generation peripheral. The timer device may perform counting tasks. So, the microprocessor becomes free from the task related to the counting process and can execute the program in memory. Hence the software overhead on the microprocessor is minimized.

ARCHITECTURE:

The 8-bit bidirectional data buffers interfaces 8254 to the microprocessor system bus. Data is transmitted or received by the buffer upon the execution of IN or OUT instruction.



Internal Block Diagram of 8254

READ/WRITE LOGIC The Read/Write Logic accepts inputs from the system bus and generates control signals for the other functional blocks of the 8254. A1 and A0 select one of the three counters or the Control Word Register to be read from/written into. A ‘‘low’’ on the RD input tells the 8254 that the CPU is reading one of the counters. A ‘‘low’’ on the WR input tells the 8254 that the CPU is writing either a Control Word or an initial count. Both RD and WR are qualified by CS; RD and WR are ignored unless the 8254 has been selected by holding CS low.

A_1	A_0	Result
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Word Register

CONTROL WORD REGISTER The Control Word Register is selected by the Read/Write Logic when A1,A0= 11. If the CPU then does a write operation to the 8254, the data is stored in the Control Word Register and is interpreted as a Control Word used to define the operation of the Counters. The Control Word Register can only be written to; status information is available with the Read-Back Command.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC—Select Counter

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command

M—Mode

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

RW—Read/Write

RW1	RW0	
0	0	Counter Latch Command
0	1	Read/Write least significant byte only
1	0	Read/Write most significant byte only
1	1	Read/Write least significant byte first, then most significant byte

BCD

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

Status Word:

The status word of a counter can be read to check the programmed status of the counter and to verify whether the count value has reached terminal count i.e., zero or not.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Output	Null Count	RW1	RW0	M2	M1	M0	BCD

D_7 1 = OUT Pin is 1
0 = OUT Pin is 0

D_6 1 = Null Count
0 = Count available for reading

D_5–D_0 Counter programmed mode

COUNTER 0, COUNTER 1, COUNTER 2 : These three functional blocks are identical in operation. The Counters are fully independent. Each Counter may operate in a different Mode. The actual counter is labelled CE (for ‘‘Counting Element’’). It is a 16-bit presettable synchronous down counter. Similarly, there are two 8-bit registers called CRM and CRL (for ‘‘Count Register’’). Both are normally referred to as one unit and called just CR. When a new count is written to the Counter, the count is stored in the CR and later transferred to the CE. The Control Logic allows one register at a time to be loaded from the internal bus. Both bytes are transferred to the CE simultaneously. CRM and CRL are cleared when the Counter is programmed. In this way, if the Counter has been programmed for one byte counts (either most significant byte only or least significant byte only) the other byte will be zero. Note that the CE cannot be written into; whenever a count is written, it is written into the CR. The Control Logic is also shown in the diagram. CLK n, GATE n, and OUT n are all connected to the outside world through the Control Logic.

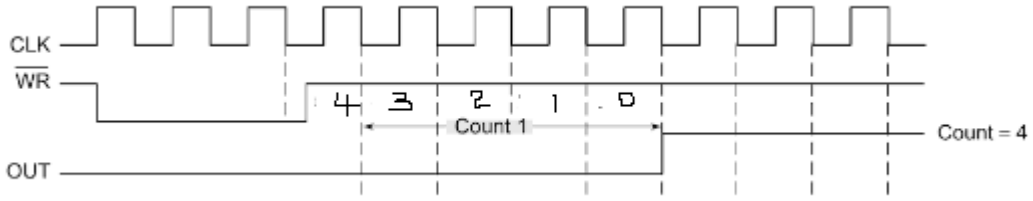
OPERATING MODES OF 8254:

Each of the counters of 8254 can be operated in one of the following six modes of operations

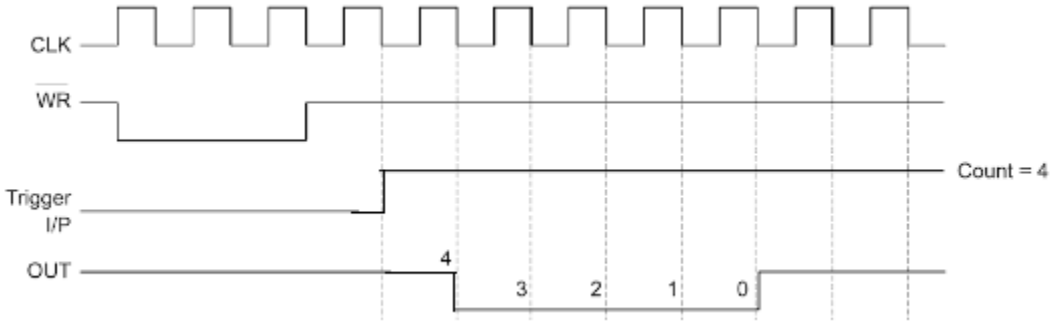
1. Mode 0 (Interrupt on Terminal Count)

2. Mode 1 (Programmable One-Shot)
3. Mode 2 (Rate Generator / Divide by N Counter)
4. Mode 3 (Square Wave Generator)
5. Mode 4 (Software Triggered Strobe)
6. Mode 5 (Hardware Triggered Strobe)

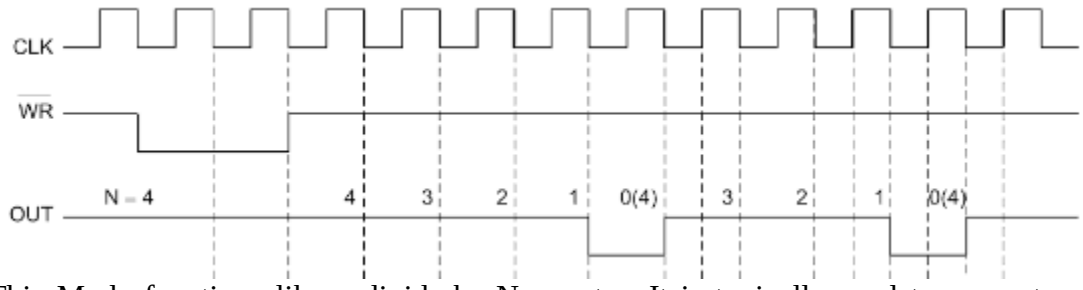
Mode 0 (Interrupt on Terminal Count): Mode 0 is typically used for event counting. After the Control Word is written, OUT is initially low, and will remain low until the Counter reaches zero. OUT then goes high and remains high until a new count or a new Mode 0 Control Word is written into the Counter. GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT. It is used to interrupt the processor.



Mode 1 (Programmable One-Shot): OUT will be initially high. OUT will go low on the CLK pulse following a trigger to begin the one-shot pulse, and will remain low until the Counter reaches zero. OUT will then go high and remain high until the CLK pulse after the next trigger. After writing the Control Word and initial count, the Counter is armed. A trigger results in loading the Counter and setting OUT pin low on the next CLK pulse, thus starting the one-shot pulse. An initial count of N will result in a one-shot pulse N CLK cycles in duration. The one-shot is retriggerable, hence OUT will remain low for N CLK pulses after any trigger. The one-shot pulse can be repeated without rewriting the same count into the counter. GATE has no effect on OUT.



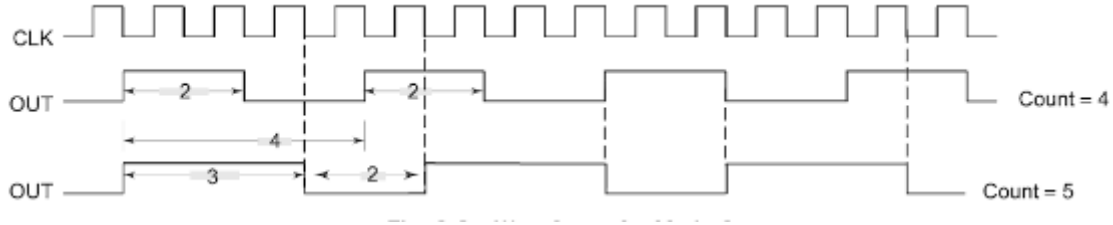
Mode 2 (Rate Generator / Divide by N Counter):



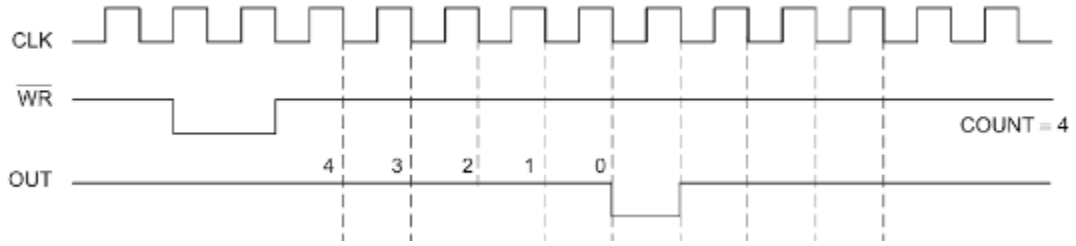
This Mode functions like a divide-by-N counter. It is typically used to generate a Real Time Clock interrupt. OUT will initially be high. When the initial count has decremented to 1, OUT goes low for one CLK pulse. OUT then goes high again, the Counter reloads the initial count and the process is repeated. Mode 2 is periodic; the same sequence is repeated indefinitely. For an

initial count of N, the sequence repeats every N CLK cycles. GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low during an output pulse, OUT is set high immediately. A trigger reloads the Counter with the initial count on the next CLK pulse; OUT goes low N CLK pulses after the trigger. Thus the GATE input can be used to synchronize the Counter.

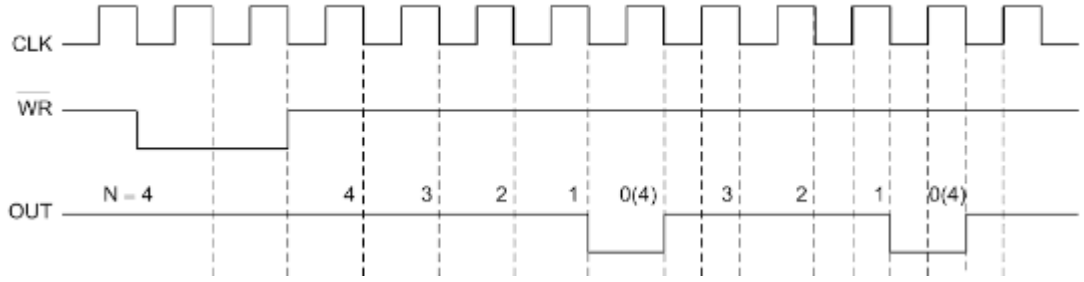
Mode 3 (Square Wave Generator): It is typically used for Baud rate generation. Mode 3 is similar to Mode 2 except for the duty cycle of OUT. OUT will initially be high. When half the initial count has expired, OUT goes low for the remainder of the count. Mode 3 is periodic; the sequence above is repeated indefinitely. An initial count of N results in a square wave with a period of N CLK cycles. GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low while OUT is low, OUT is set high immediately; no CLK pulse is required. A trigger reloads the Counter with the initial count on the next CLK pulse. Thus the GATE input can be used to synchronize the Counter.



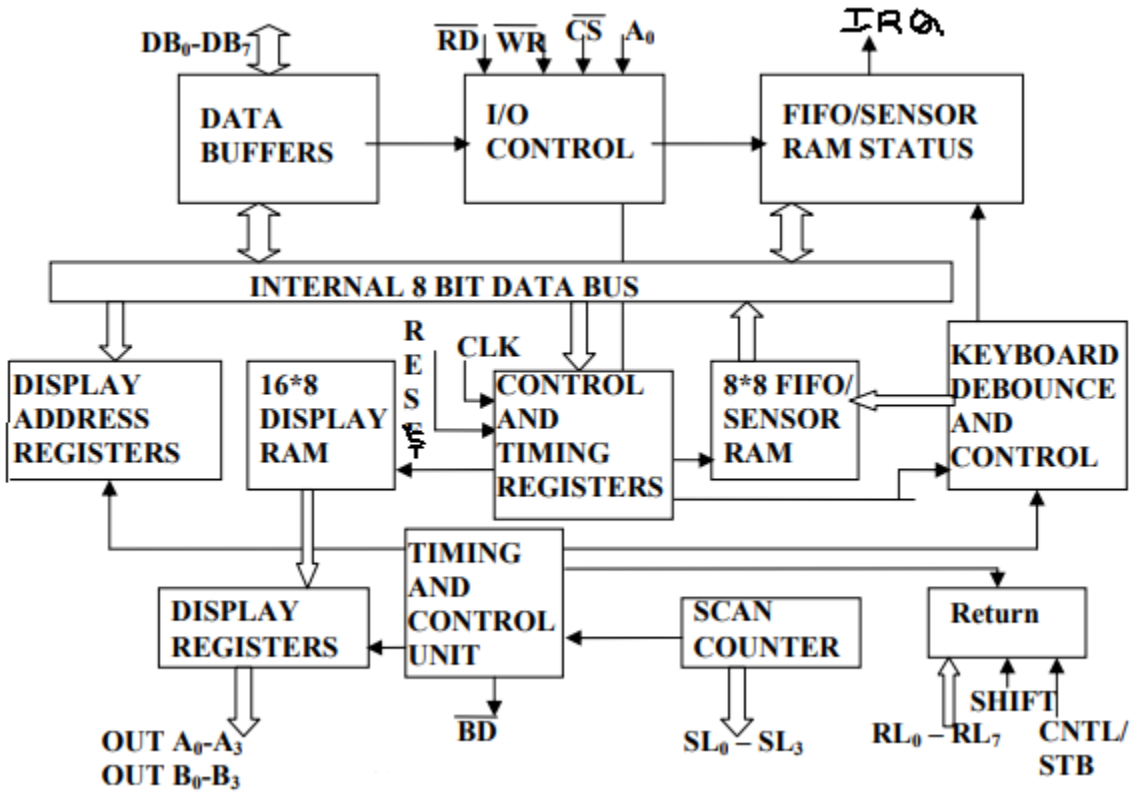
Mode 4 (Software Triggered Strobe): The output goes high, when this mode is set. When a count is loaded, counting down starts. On reaching the terminal count, the output goes low for one clock cycle and then it again goes high. This low pulse can be used as a strobe, while interfacing the microprocessor with other peripherals.



Mode 5 (Hardware Triggered Strobe): OUT will initially be high. Counting is triggered by a rising edge of GATE. When the initial count has expired, OUT will go low for one CLK pulse and then go high again. After writing the Control Word and initial count, the counter will not be loaded until the CLK pulse after a trigger. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N a 1 CLK pulses after a trigger. A trigger results in the Counter being loaded with the initial count on the next CLK pulse. The counting sequence is retriggerable. OUT will not strobe low for N a 1 CLK pulses after any trigger. GATE has no effect on OUT.



PROGRAMMMABLE KEYBOARD / DISPLAY INTERFACE – 8279



Internal block diagram of 8279

The device runs internally at 100 kHz. Since the CLK input is at 3 MHz, the clock has to be divided inside to 100 kHz. For doing so, proper Program Clock Word must be sent to the device and then the device is ready for Data I/O. The block diagram shows all the internal blocks for both keyboard and display control.

I/O Control and Data Buffer : This unit controls the flow of data through the microprocessor. It is enabled only when D is low. Its data buffer interfaces the external bus of the system with the internal bus of the microprocessor. The pins A0, RD, and WR are used for command, status or data read/write operations.

Control and Timing Register and Timing Control : This unit contains registers to store the keyboard, display modes, and other operations as programmed by the CPU. The timing and control unit handles the timings for the operation of the circuit.

Scan Counter : It has two modes , Encoded mode and Decoded mode. In the **encoded mode**, the counter provides the binary count that is to be externally decoded to provide the scan lines for the keyboard and display.

In the **decoded scan mode**, the counter internally decodes the least significant 2 bits and provides a decoded 1 out of 4 scan on SL_0-SL_3.

Return Buffers, Keyboard Debounce, and Control : This unit first scans the key closure row-wise, if found then the keyboard debounce unit debounces the key entry. In case, the same key is

detected, then the code of that key is directly transferred to the sensor RAM along with SHIFT & CONTROL key status.

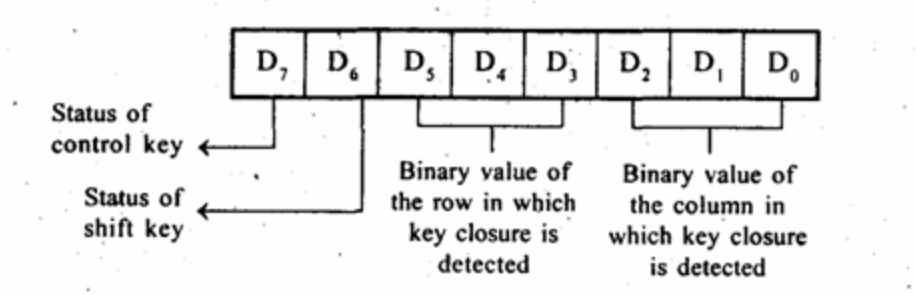
FIFO/Sensor RAM and Status Logic: This unit acts as 8-byte first-in-first-out (FIFO) RAM where the key code of every pressed key is entered into the RAM as per their sequence. The status logic generates an interrupt request after each FIFO read operation till the FIFO gets empty.

In the scanned sensor matrix mode, this unit acts as sensor RAM where its each row is loaded with the status of their corresponding row of sensors into the matrix. When the sensor changes its state, the IRQ line changes to high and interrupts the CPU.

Display Address Registers and Display RAM : This unit consists of display address registers which holds the addresses of the word currently read/written by the CPU to/from the display RAM.

Display is achieved using 1) the internal 16×8 Display RAM, 2) the Timing control unit and 3) the Scan Counter. CPU fills the display RAM with data. The data must be in seven segment format. The scan counter counts from "0000" to "1111" and places its output on the line SL3-SL0. Based on the scan count generated, the Decoder shown in Figure 1 selectively turns one of the sixteen characters in the display unit. The timing control for the display is done by the signal BD. The characters are displayed from digit 1 to 16 and then returns back to 1. The time 8279 takes between two updates to the same digit position is called the scan time.

The 16 character keyboard is organized as a 4×4 matrix keyboard. We use only the last two bits of the scan counter, namely SL1 and SL0. Keyscan is achieved by driving one row to '0' at a time and detecting for one of the columns being '0'. If there is a key press, the corresponding column goes low and the column information is passed on to 8279 through the lines RL3-RL0. 8279 knows the row code that is being generated and we can scan lines RL3-RL0, effectively detecting the exact key that is pressed. After detecting a keypress, 8279 waits for a debounce time and scans the key again. If keypress is still present, it produces a 8-bit keycode based on the column and the row in which the key is pressed. The keycode is then placed in the internal 8-byte FIFO RAM. If a valid keypress is found, it asserts an Interrupt Request to the CPU through the line IRQ. It is up to the CPU to read the data. Meanwhile, 8279 increments FIFO count in the internal status register.



The two operating modes of keyboard section are 2-key lockout and N-key rollover. In the 2-key lockout mode, if two keys are pressed simultaneously, only the first key is recognized. In the N-key rollover mode, simultaneous keys are recognized and their codes are stored in FIFO.

Modes of Operation

There are two modes of operation on 8279 − **Input Mode** and **Output Mode**.

Input Mode

This mode deals with the input given by the keyboard and this mode is further classified into 3 modes.

- **Scanned Keyboard Mode** − In this mode, the key matrix can be interfaced using either encoded or decoded scans. In the encoded scan, an 8×8 keyboard or in the decoded scan, a 4×8 keyboard can be interfaced. The code of key pressed with SHIFT and CONTROL status is stored into the FIFO RAM.
- **Scanned Sensor Matrix** − In this mode, a sensor array can be interfaced with the processor using either encoder or decoder scans. In the encoder scan, 8×8 sensor matrix or with decoder scan 4×8 sensor matrix can be interfaced.
- **Strobed Input** − In this mode, when the control line is set to 0, the data on the return lines is stored in the FIFO byte by byte.

Output Mode

This mode deals with display-related operations. This mode is further classified into two output modes.

- **Display Scan** − This mode allows 8/16 character multiplexed displays to be organized as dual 4-bit/single 8-bit display units.
- **Display Entry** − This mode allows the data to be entered for display either from the right side/left side.

Command Words of 8279:

Keyboard Display mode set : The format of the command word to select different modes of operation of 8279 is

0	0	0	D	D	K	K	K

D	D	Display Modes
0	0	8 8-bit character Left Entry
0	1	16 8-bit character Left Entry
1	0	8 8-bit character Right Entry
1	1	16 8-bit character Right Entry

K	K	K	Keyboard modes
0	0	0	Encoded scan keyboard – 2 key Lockout
0	0	1	Decoded Scan keyboard – 2 key Lockout
0	1	0	Encoded Scan keyboard – N-key Rollover
0	1	1	Decoded Scan keyboard – N-key Rollover
1	0	0	Encoded Scan Sensor Matrix
1	0	1	Decoded Scan Sensor Matrix
1	1	0	Strobed input, Encoded Display Scan
1	1	1	Strobed Input, Decoded Display Scan

Read FIFO/Sensor RAM : The format of this command is given as shown below

0	1	0	AI	X	A	A	A

X - don't care

AI - Auto increment flag

AAA - Address pointer to 8 bit FIFO RAM
This word is written to set up 8279 for reading FIFO/Sensor RAM. In scanned keyboard mode, AI and AAA bits are of no use. The 8279 will automatically drive data bus for each subsequent read, in the same sequence, in which the data was entered.

Read Display RAM

This command enables a programmer to read the display RAM data

0	1	1	AI	A	A	A	A

The CPU writes this command word to 8279 to prepare it for display RAM read operation. AI is auto incremented flag and AAAA, the 4-bit address, points to the 16-byte display RAM that is to be read. If AI = 1, the address will be automatically, incremented after each read or write to the display RAM.

Write Display RAM
The format of this command is given as shown below

1	0	0	AI	A	A	A	A

AI - Auto increment flag
AAAA - 4-bit address for 16-bit display RAM to be written
Other details of this command are similar to the ‘Read Display RAM Command.

PROGRAMMABLE INTERRUPT CONTROLLER - INTEL 8259

FEATURES OF 8259:

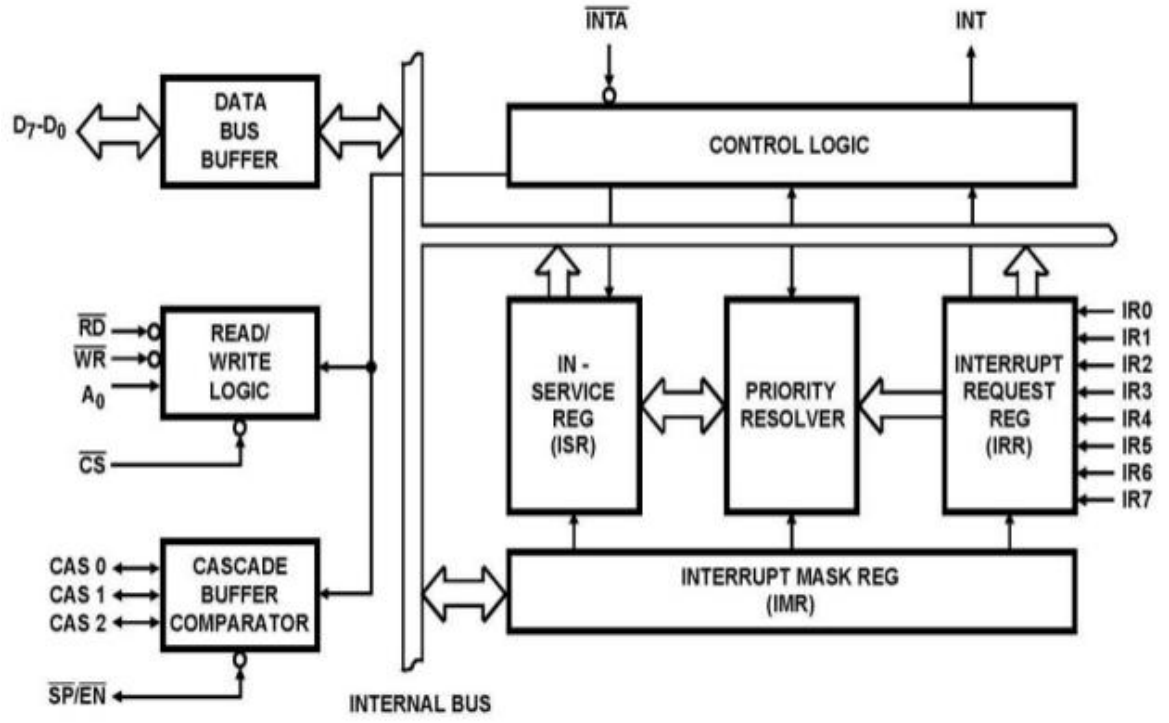
1. It is programmed to work with either 8085 or 8086 processor.
2. It manage 8-interrupts according to the instructions written into its control registers.
3. In 8086 processor, it supplies the type number of the interrupt and the type number is programmable. The priorities of the interrupts are programmable.
4. The interrupts can be masked or unmasked individually.
5. The 8259s can be cascaded to accept a maximum of 64 interrupts.

FUNCTIONAL BLOCK DIAGRAM OF 8259:

The 8259A (PIC) has eight interrupt request inputs – IR7 - IR0. The 8259A uses its INT output to interrupt the 8086 via INTR pin. The 8259A receives interrupt acknowledge pulses from the μp at its INTA input. Vector address, used by the 8086 to transfer control to the service subroutine of the interrupting device, is provided by the 8259A on the data bus. The 8259A is a programmable device that must be initialized by command words sent by the microprocessor. After initialization the 8259A mode of operation can be changed by operation command words from the microprocessor.

It has eight functional blocks. They are,

1. Control logic
2. Read Write logic
3. Data bus buffer
4. Interrupt Request Register (IRR)
5. In-Service Register (ISR)
6. Interrupt Mask Register (IMR)
7. Priority Resolver (PR)
8. Cascade buffer.



The descriptions of various blocks are given below:

Data bus buffer: This 3- state, bidirectional 8-bit buffer is used to interface the 8259A to the system data bus. Control words and status information from the microprocessor to PIC and from PIC to microprocessor respectively, are transferred through the data bus buffer.

Control Logic : INT (Interrupt output) pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU, thus it is connected to the CPU's interrupt pin (INTR). In case of master-slave configuration, the interrupt pin of slave 8259A is connected to interrupt request input of master 8259A. INTA (Interrupt Acknowledge)pin is used to enable 8259A interrupt vector data on the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.

Read/Write & Control Logic: The function of this block is to accept output commands sent from the CPU. It contains the initialization command word (ICW) registers and operation command word (OCW) registers which store the various control formats for device operation. This function block also allows the status of 8259A to be transferred to the data bus.

Interrupt Request Register (IRR): Interrupt request register (IRR) stores all the interrupt inputs that are requesting service. It is an 8-bit register – one bit for each interrupt request. Basically, it keeps track of which interrupt inputs are asking for service. If an interrupt input is unmasked, and has an interrupt signal on it, then the corresponding bit in the IRR will be set. The content of this register can be read to know the status of pending interrupts.

Interrupt Mask Register (IMR): The IMR is used to disable (Mask) or enable (Unmask) individual interrupt request inputs. This is also an 8-bit register. Each bit in this register corresponds to the interrupt input with the same number. The IMR operates on the IRR. Masking f higher priority input will not affect the interrupt request lines of lower priority. To unmask any interrupt the corresponding bit is set '0'.

In-service Register (ISR): The in-service register keeps track of which interrupt inputs are currently being serviced. For each input that is currently being serviced the corresponding bit of in-service register (ISR) will be set. In 8259A, during the service of an interrupt request, if another higher priority interrupt becomes active, it will be acknowledged and the control will be transferred from lower priority interrupt service subroutine (ISS) to higher priority ISS. Thus, more than one bit of ISR will be set indicating the number of interrupts being serviced. Each of these 3-registers can be read as status register.

Priority Resolver: This logic block determines the priorities of the interrupts set in the IRR. It takes the information from IRR, IMR and ISR to determine whether the new interrupt request is having highest priority or not. If the new interrupt request is having the highest priority, it is selected and processed. The corresponding bit of ISR will be set during interrupt acknowledge machine cycle.

Cascade Buffer/Comparator: This function block stores and compares the IDs of all 8259A's in the system. The associated 3-I/O lines (CAS2-CAS0) are outputs when 8259A is used as a master and are inputs when 8259A is used as a slave. As a master, the 8259A sends the ID of the interrupting slave device onto the CAS2-0 lines. The slave 8259As compare this ID with their

own programmed ID. Thus selected 8259A will send its pre-programmed subroutine address on to the data bus during the next one or two successive INTA pulses. SP / EN (Salve Program/Enable Buffer) is a dual function pin. When the chip is programmed in buffered mode, the pin can be used as an output and when not in the buffered mode it is used as an input. In non-buffered mode it is used as an input pin to determine whether the 8259A is to be used as a master (SP / EN = 1) or as a slave (SP / EN = 0). In buffered mode, normally data bus buffers are used. These buffers need to be enabled or disabled during transfer of vector information depending upon whether 8259A is connected before the buffer or after the buffer. To disable/enable the data bus transceivers (buffers) when data are being transferred from the 8259A to the CPU, this pin is made low or high.

The Interrupt sequence in an 8086-8259A system is described as follows:

1. One or more IR lines are raised high that set corresponding IRR bits.
2. 8259A resolves priority and sends an INT signal to CPU.
3. The CPU acknowledge with INTA pulse.
4. Upon receiving an INTA signal from the CPU, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 8259A does not drive data during this period.
5. The 8086 will initiate a second INTA pulse. During this period 8259A releases an 8-bit pointer on to a data bus from where it is read by the CPU.
6. This completes the interrupt cycle. The ISR bit is reset at the end of the second INTA pulse if automatic end of interrupt (AEOI) mode is programmed. Otherwise ISR bit remains set until an appropriate EOI command is issued at the end of interrupt subroutine.

The command words of 8259A are classified in two groups
1. Initialization command words (ICW) and
2. Operation command words (OCW).

Initialization Command Words (ICW): Before it starts functioning, the 8259A must be initialized by writing two to four command words into the respective command word registers. These are called as initialized command words. There are 4 ICW in 8259

Operation Command Words: Once 8259A is initialized using the previously discussed command words for initialisation, it is ready for its normal function, i.e. for accepting the interrupts but 8259A has its own way of handling the received interrupts called as modes of operation. These modes of operations can be selected by programming, i.e. writing three internal registers called as operation command words. There are 3 OCW in 8259

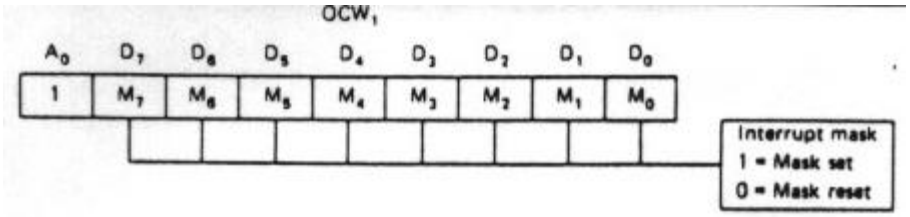
ICW_1

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	**A_7**	**A_6**	**A_5**	**1**	**LITM**	**ADI**	**SNGL**	**IC_4**

D_0 1=ICW_4 Needed
0=No ICW_4 Needed
D_1 1=Single
0=Cascaded
D_2 Call Address Interval
1=Interval of 4 bytes
0=Interval of 8 bytes

A_7-A_5 of Interrupt vector address
MCs 80/85 mode only

D_3 1=Level Triggered
0=Edge Triggered



Operating Modes of 8259

The different modes of operation of 8259A can be programmed by setting or resting the appropriate bits of the ICW or OCW as discussed previously. The different modes of operation of 8259A are explained in the following.

- **Fully Nested Mode :** This is the default mode of operation of 8259A. IR0 has the highest priority and IR 7 has the lowest one. When interrupt request are noticed, the highest priority request amongst them is determined and the vector is placed on the data bus. The corresponding bit of ISR is set and remains set till the microprocessor issues an EOI command just before returning from the service routine or the AEOI bit is set. If the ISR (in service) bit is set, all the same or lower priority interrupts are inhibited but higher levels will generate an interrupt, that will be acknowledge only if the microprocessor interrupt enable flag IF is set. The priorities can afterwards be changed by programming the rotating priority modes.

- **End of Interrupt (EOI) :** The ISR bit can be reset either with AEOI bit of ICW1 or by EOI command, issued before returning from the interrupt service routine. There are two types of EOI commands specific and non-specific. When 8259A is operated in the modes that preserve fully nested structure, it can determine which ISR bit is to be reset on EOI. When non-specific EOI command is issued to 8259A it will be automatically reset the highest ISR bit out of those already set. When a mode that may disturb the fully nested structure is used, the 8259A is no longer able to determine the last level acknowledged. In this case a specific EOI command is issued to reset a particular ISR bit. An ISR bit that is masked by the corresponding IMR bit, will not be cleared by non-specific EOI of 8259A, if it is in special mask mode.

- **Automatic Rotation :** This is used in the applications where all the interrupting devices are of equal priority. In this mode, an interrupt request IR level receives priority after it is served while the next device to be served gets the highest priority in sequence. Once all the device are served like this, the first device again receives highest priority.

- **Automatic EOI Mode :** Till AEOI=1 in ICW 4, the 8259A operates in AEOI mode. In this mode, the 8259A performs a non-specific EOI operation at the trailing edge of the last INTA pulse automatically. This mode should be used only when a nested multilevel interrupt structure is not required with a single 8259A.

- **Specific Rotation :** In this mode a bottom priority level can be selected, using L2, L1 and L0 in OCW 2 and R=1, SL=1, EOI=0. The selected bottom priority fixes other priorities. If IR 5 is selected as a bottom priority, then IR 5 will have least priority and IR4 will have a next higher priority. Thus IR 6 will have the highest priority. These priorities can be changed during an EOI command by programming the rotate on specific EOI command in OCW 2.

PROGRAMMABLE DMA CONTROLLER - INTEL 8257

DMA stands for Direct Memory Access. It is designed by Intel to transfer data at the fastest rate. It allows the device to transfer the data directly to/from memory without any interference of the CPU.
Using a DMA controller, the device requests the CPU to hold its data, address and control bus, so the device is free to transfer data directly to/from the memory. The DMA data transfer is initiated only after receiving HLDA signal from the CPU.

How DMA Operations are Performed?

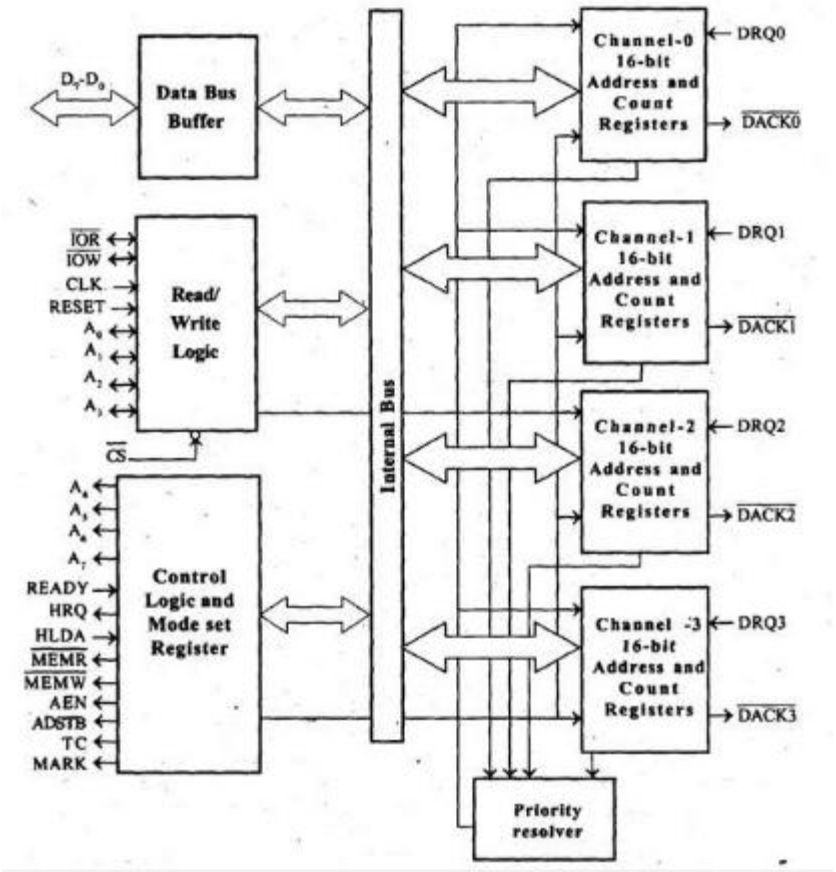
Following is the sequence of operations performed by a DMA −

- Initially, when any device has to send data between the device and the memory, the device has to send DMA request (DRQ) to DMA controller.
- The DMA controller sends Hold request (HRQ) to the CPU and waits for the CPU to assert the HLDA.
- Then the microprocessor tri-states all the data bus, address bus, and control bus. The CPU leaves the control over bus and acknowledges the HOLD request through HLDA signal.
- Now the CPU is in HOLD state and the DMA controller has to manage the operations over buses between the CPU, memory, and I/O devices.

The 8257 is programmable. Direct Memory Access (DMA) device which, when coupled with single Intel® 8212 I/O port device, provides complete four-channel DMA controller for use in Intel® microcomputer systems. After being initialized by software, the 8257 can transfer block of data, containing up to 16.384 bytes, between memory and peripheral device directly, without further intervention required of the CPU. Upon receiving DMA transfer request from an enabled peripheral, the 8257:
1. Acquires control of the system bus.
2. Acknowledges that requesting peripheral which is connected to the highest priority channel.
3. Outputs the least significant eight bits of the memory address onto system address lines A0-A7. outputs the most significant eight bits of the memory address to the 8212 I/O port via.the data bus (the 8212 places these address bits on lines A8-A15), and
4. Generates the appropriate memory and I/O read/ write control signals that cause the peripheral to receive or deposit a data byte directly from or to the addressed location in memory.

The 8257 will retain control of the system bus and repeat the transfer sequence, as long as peripheral maintains its DMA request. Thus, the 8257 can transfer block of data to/from high speed peripheral (e.g.. sector of data on floppy disk) in single "burst". When the specified number of data bytes have been transferred, the 8257 activates its Terminal Count (TC) output, informing the CPU that the operation is complete.

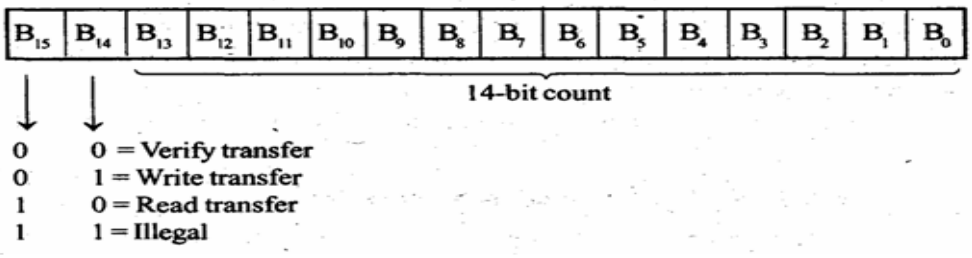
The 8257 offers three different modes of operation: (1) DMA read, which causes data to be transferred from memory to peripheral: (2) DMA write, which causes data to be transferred from peripheral to memory: and (3) DMA verify, which does not actually involve the transfer of data. When an 8257 channel is in the DMA verify mode, it will respond the same as described for transfer operations, except that no memory or I/O read/write control signals will be generated, thus preventing the transfer of data



8257 DMA Controller

Block Diagram Description

1. DMA Channels: The 8257 provides four separate DMA channels (labeled CH-0 to CH-3). Each channel includes two sixteen-bit registers: (1) DMA address register, and (2) terminal count register. Both registers must be initialized before channel is enabled. The DMA address register is loaded with the address of the first memory location to be accessed. The value loaded into the low-order 14-bits of the terminal count register specifies the number of DMA cycles minus one before the Terminal Count (TC) output is activated. For instance, terminal count of would cause the TC output to be active in the first DMA cycle for that channel. In general, if the number of desired DMA cycles, load the value N-1 into the low-order 14-bits of the terminal count register. The most significant two bits of the terminal count register specify the type of DMA operation for that channel. These two bits are not modified during DMA cycle, but can be changed between DMA blocks. Each channel accepts DMA Request (DRQn) input and provides DMA Acknowledge ($\overline{DACK\,n}$) output.



(DRQ 0-DRQ 3) DMA Request: These are individual asynchronous channel request inputs used by the peripherals to obtain DMA cycle. If not in the rotating priority mode then DRQ 0 has the highest priority and DRQ has the lowest. request can be generated by raising the request line and holding it high until DMA acknowledge. For multiple DMA cycles (Burst Mode) the request line is held high until the DMA acknowledge of the last cycle arrives.

($\overline{DACK0}$ - $\overline{DACK 3}$) DMA Acknowledge: An active low level on the acknowledge output informs the peripheral connected to that channel that it has been selected for DMA cycle. The DACK output acts as "chip select'* for the peripheral device requesting service. This line goes active (low) and inactive (high) once for each byte transferred even if burst of data is being transferred.

2. **Data Bus Buffer** :This three-state, bi-directional, eight bit buffer interfaces the 8257 to the system data bus.

(D_0-D_7)Data Bus Lines: These are bi-directional three-state lines. When the 8257 is being programmed by the CPU. eightbits of data for DMA address register, terminal count register or the Mode Set register are received on the data bus. When the CPU reads DMA address register, terminal count register or the Status register, the data is sent to the CPU over the data bus. During DMA cycles (when the 8257 is the bus master), the 8257 will output the most significant eight-bits of the memory address (from one of the DMA address registers) to the 8212 latch via the data bus. These address bits will be transferred at the beginning of the DMA cycle: the bus will then be released to handle the memory data transfer during the balance of the DMA cycle.

3**. Read/Write Logic** :When the CPU is programming or reading one of the 8257*s registers (i.e., when the 8257 is "slave" device on the system bus), the Read/Write Logic accepts the I/O Read (USE) or I/O Write signal, decodes the least significant four address bits, (A0-A3), and either writes the contents of the data bus into the addressed register (if $\overline{I/OW}$ is true) or places the contents of the addressed register onto the data bus (if $\overline{I/OR}$ is true). During DMA cycles (i.e., when the 8257 is the bus "master"), the Read/Write Logic generates the I/O read and memory write (DMA write cycle) or I/O Write and memory read (DMA read cycle) signals which control the data link with the peripheral that has been granted the DMA cycle. Note that during DMA transfers Non-DMA I/O devices should be de-selected (disabled) using "AEN" signal to inhibit I/O device decoding of the memory address as an erroneous device address.

$\overline{I/OR}$ I/O Read: An active-low, bi-directional three-state line. In the "slave" mode, it is an input which allows the 8-bit status register or the upper/lower byte of 18-bit DMA address register or terminal count register to be read. In the "master" mode, $\overline{I/OR}$ is control output which is used to access data from peripheral during the DMA write cycle.

$\overline{I/OW}$ **I/O Write**: An active-low, bi-directional three-state line. In the "slave" mode, it is an input which allows the contents of the data bus to be loaded into the 8-bit mode set register or the upper/lower byte of 18-bit DMA address register or terminal count register. In the "master" mode. $\overline{I/OW}$ is control output which allows data to be output to peripheral during DMA read cycle.

(CLK) Clock Input: Generally from an Intel® 8224 Clock Generator device. (*2 TTL) or Intel® 8085A CLK output.

(RESET) Reset: An asynchronous input (generally from an 8224 or 8085 device) which disables all DMA channels by clearing the mode register and 3-states all control lines.

Address Lines: These least significant four address lines are bi-directional. In the "slave" mode they are inputs which select one of the registers to be read or programmed. In the "master" mode, they are outputs which constitute the least significant four bits of the 16-bit memory address generated by the 8257.

($\overline{CS}$) **Chip Select:** An active-low input which enables the I/O Read or I/O Write input when the 8257 is being read or programmed in the "slave" mode. In the "master" mode. $\overline{CS}$ is automatically disabled to prevent the chip from selecting itself while performing the DMA function.

4. **Control Logic:** This block controls the sequence of operations during all DMA cycles by generating the appropriate control signals and the 16-bit address that specifies the memory location to be accessed.

Address Lines (A_0-A_7): These four address lines are three-state outputs which constitute bits through of the 16-bit memory address generated by the 8257 during all OMA cycles.

(READY) Ready: This asynchronous input is used to elongate the memory read and write cycles in the 8257 with wait states if the selected memory requires longer cycles. READY must conform to specified setup and hold times.

(HRQ) Hold Request: This output requests control of the system bus. In systems with only one 8257, HRQ will normally be applied to the HOLD input on the CPU. HRQ must conform to specified setup and hold times.

(HLDA) Hold Acknowledge: This input from the CPU indicates that the 8257 has acquired control of the system bus.

($\overline{MEMR}$) **Memory Read:** This active-low three-state output is used to read data from the addressed memory location during DMA Read cycles.

($\overline{MEMW}$) **Memory Write:** This active-low three-state output is used to write data into the addressed memory location during DMA Write cycles.

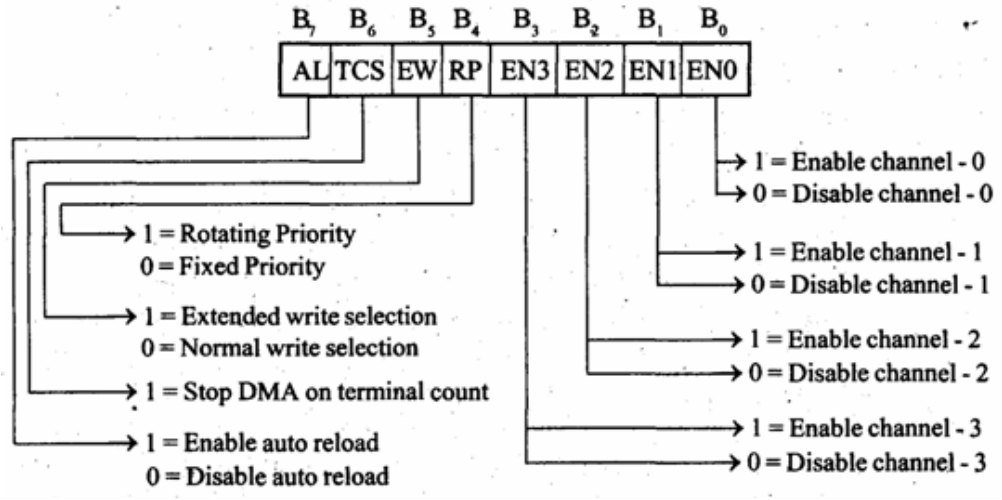
(AOSTB) Address Strobe: This output strobes the most significant byte of the memory address into the 8212 device from the data bus.

(AEN) Address Enable: This output is used to disable (float) the System Data Bus and the System Control Bus. It may also be used to disable (float) the System Address Bus by use of an enable on the Address Bus drivers in systems to inhibit non-DMA devices from responding during DMA cycles.

(TC) Terminal Count: This output notifies the currently selected peripheral that the present DMA cycle should be the last cycle for this data block. If the TC STOP bit in the Mode Set register is set. the selected channel will be automatically disabled at the end of that DMA cycle. TC is activated when the 14-bit value in the selected channel's terminal count register equals zero. Recall that the loworder 14-bits of the terminal count register should be loaded with the values (n-1). where the desired number of the DMA cycles.

(MARK) Modulo 128 Mark: This output notifies the selected peripheral that the current DMA cycle is the 128th cycle since the previous MARK output. MARK always occurs at 128 (and all multiples of 128) cycles from the end of the data block.

Mode set Register

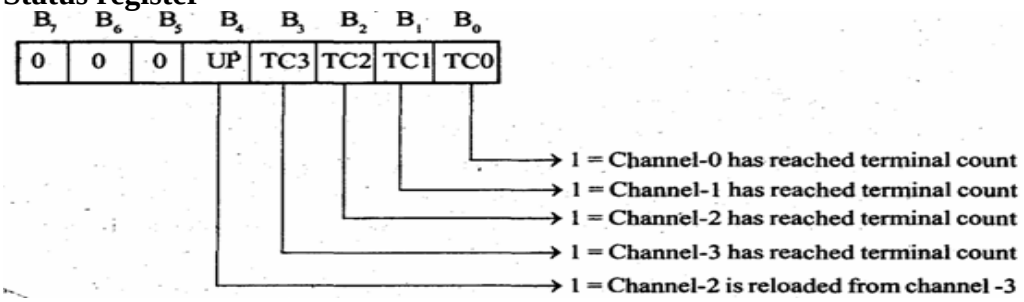


- The bits B0, B1, B2, and B3 of mode set register are used to enable/disable channel -0, 1, 2 and 3 respectively. A one in these bit position will enable a particular channel and a zero will disable it.
- If the bit B4 is set to one, then the channels will have rotating priority and if it zero then the channels wilt have fixed priority.

1. In rotating priority after servicing a channel its priority is made as lowest.
2. In fixed priority the channel-0 has highest priority and channel-2 has lowest priority.

- If the bit B5 is set to one, then the timing of low write signals (MEMW and IOW) will be extended.
- If the bit B6 is set to one then the DMA operation is stopped at the terminal count.
- The bit B7 is used to select the auto load feature for DMA channel-2.
- When bit B7 is set to one, then the content of channel-3 count and address registers are loaded in channel-2 count and address registers respectively whenever the channel-2 reaches terminal count. When this mode is activated the number of channels available for DMA reduces from four to three.

Status register



- The bit B0, B1, B2, and B3 of status register indicates the terminal count status of channel-0, 1,2 and 3 respectively. A one in these bit positions indicates that the particular channel has reached terminal count. These status bits are cleared after a read operation by microprocessor.
- The bit B4 of status register is called update flag and a one in this bit position indicates that the channel-2 register has been reloaded from channel-3 registers in the auto load mode of operation.