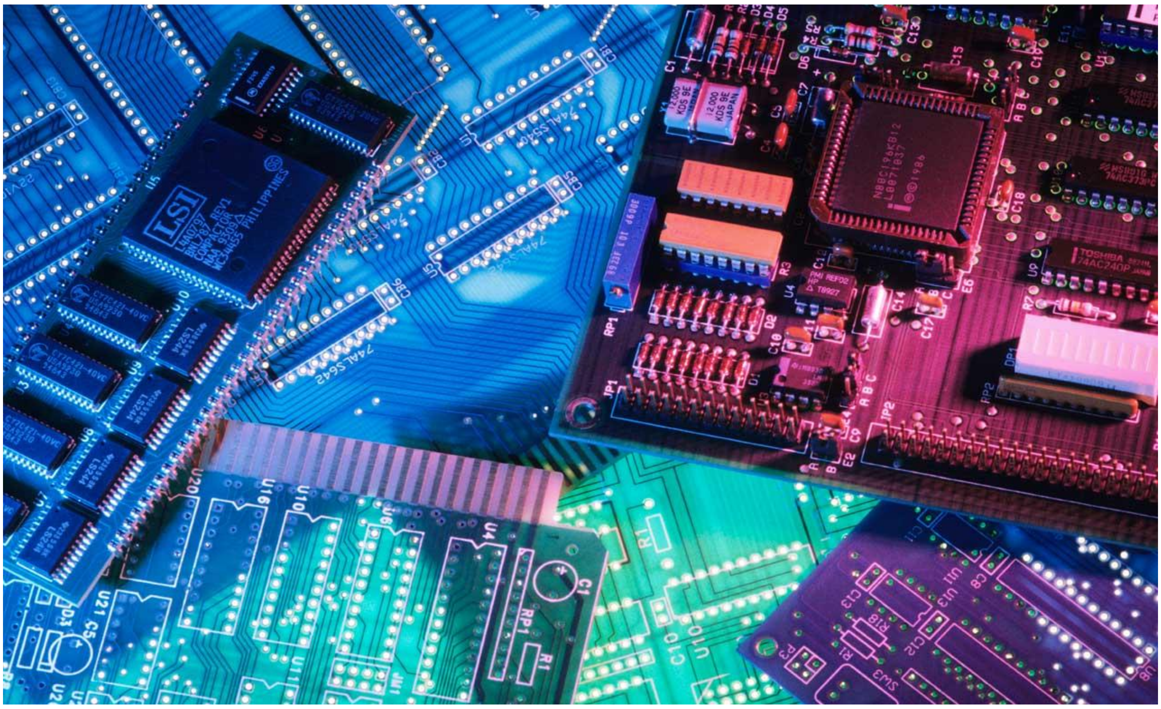
MICROPROCESSORS AND INTERFACING

III B.TECH I SEMESTER



DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

LENDI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institute, Approved by AICTE & Permanently Affiliated to JNTU-GV, Vizianagaram)

(Accredited By NAAC with A Grade and Accredited by NBA Tier-1)

Jonnada (Village), Denkada (Mandal), Vizianagaram District – 535 005

Phone No. 08922-241111, 241112

E-Mail: lendi_2008@yahoo.com website: www.lendi.org

UNIT – V
Interfacing peripherals

8257 DMA controller- DMA transfers and operations,8279 programmable Keyboard/Display-Keyboard modes and display modes,8251 USART interfacing-Synchronous and asynchronous mode.
Case study: Air pollution monitoring.

INDEX

UNIT – V
INTERFACING PERIPHERALS

DMA CONTROLLER 8257

The Direct Memory Access or DMA mode of transfer is fastest amongst all the modes of data transfer. In this mode, the device may transfer data directly to/from memory without any interference from the CPU. The device requests the CPU (through a DMA controller) to hold its data, address and control bus, so that the device may transfer data directly from the CPU.

The Intel’s 8257 is a 4-channel direct memory access (DMA) controller. The 8257 on behalf of the devices, requests the CPU for bus access using local bus request input i.e. HOLD in minimum mode. In maximum mode of the microprocessor RQ’/GT’ pin is used as bus request input. On receiving the HLDA signal (in minimum mode) or RQ’/GT’ signal (in maximum mode) from the CPU, the requesting device gets the access of the bus, and it completes the required number of DMA cycles for the data transfer and then hands over the control of the bus back to the CPU.

INTERNAL ARCHITECTURE OF 8257

The internal architecture of 8257 is shown in the below figure. The chip supports four DMA channels, i.e. four peripheral devices can independently request for DMA data transfer through these channels at a time. The DMA controller has 8-bit internal data buffer, a read/write unit, a priority resolving unit along with a set of registers.

Data Bus Buffer:

The 8 bit, tristate, bidirectional buffer interfaces the internal bus of 8257 with the external system bus under the control of various control signals.

Read/Write Logic:

In a **slave mode**, the read/write logic accepts the I/O Read or Write Signals decodes the A0-A3 lines and either writes the contents of data bus to addressed internal register or reads the contents of the selected register depending upon whether IOW’ or IOR’ signal is activated.

In the **master mode**, the read/write logic generates the IOW’ and IOR’ signals to control the data flow to or from the selected peripheral.

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. It generates the signals like AEN, ADSTB, MEMR’, MEMW’, TC and MARK along with the address lines A4-A7, in master mode.

Priority Resolver:

It resolves the priority of the four DMA channels depending upon whether normal priority or rotating priority is programmed.

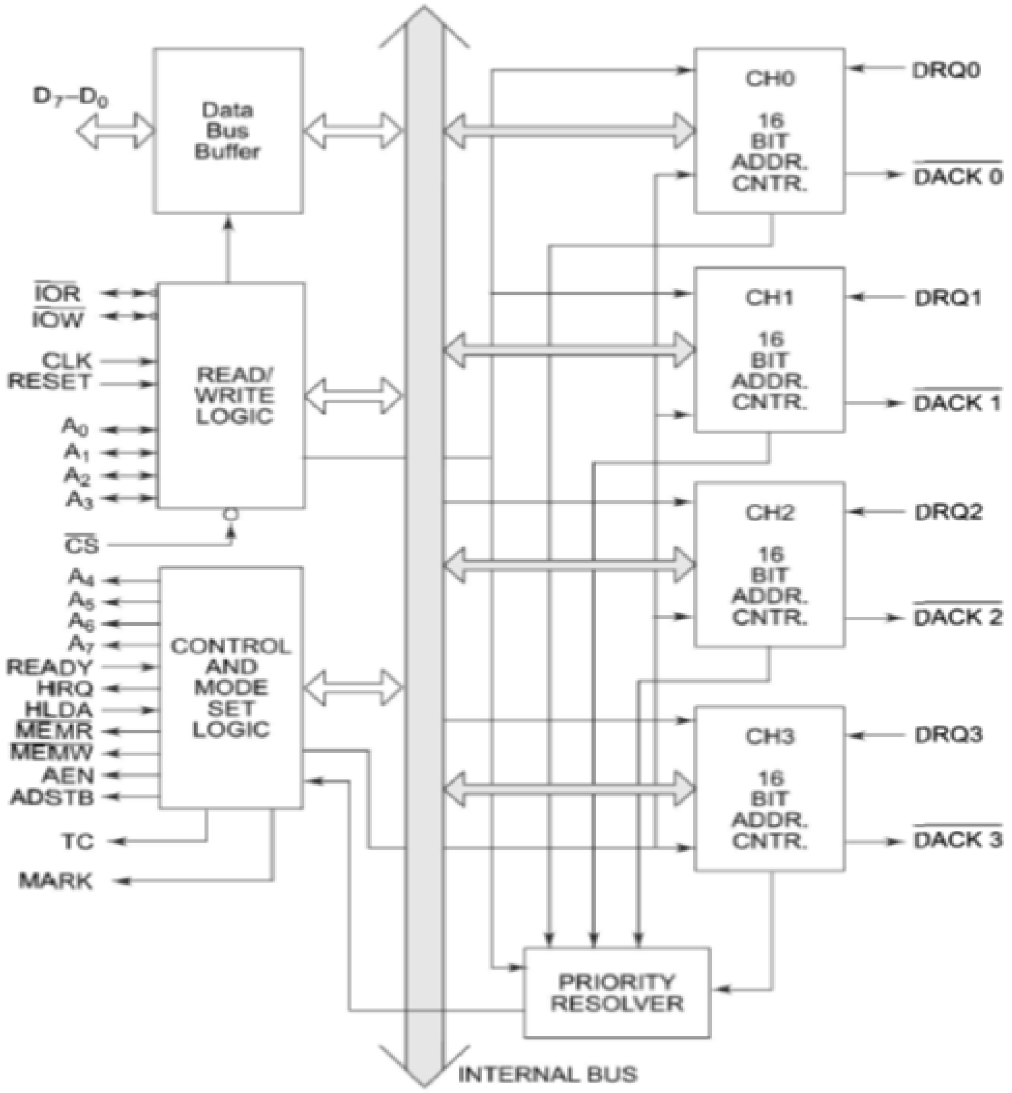


Figure: Internal architecture of 8257

REGISTER ORGANIZATION OF 8257:

The 8257 provides four separate DMA channels (labeled CH-0 to CH-3). Each channel includes two sixteen-bit registers: (1) a DMA address register, and (2) a terminal count register. Both registers must be initialized before a channel is enabled. Also, there are two common registers for all the channels, namely, mode set register and status register. Thus there are total of 10 registers. The CPU selects one of these ten registers using address lines A0-A3.

DMA address registers:

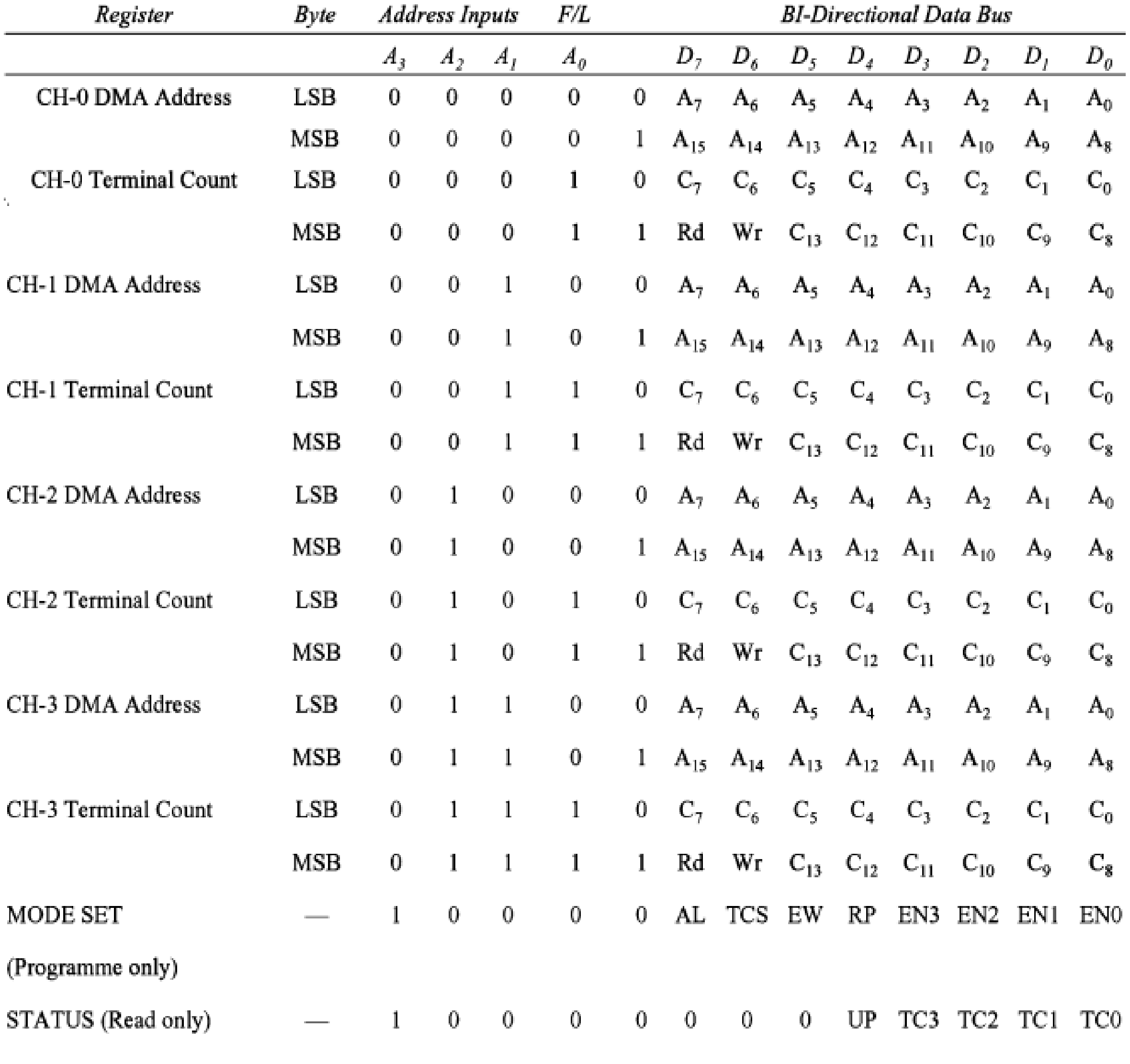
Each DMA channel has one DMA address register. The function of this register is to store the address of the starting memory location, which will be accessed by the DMA channel. Thus the starting address of the memory block which will be accessed by the device is first loaded in DMA address register of the channel.

Terminal Count Register:

Each of the four DMA channels of 8257 has one terminal count register (TC). The 16-bit register is used for ascertaining that the data transfer through a DMA channel stops after the required number of DMA cycles. Thus this register should be appropriately written before the actual DMA operation starts. The low-order 14-bits of the terminal count register are initialized with the binary equivalent of the number of required DMA cycles minus one. After each DMA cycle, the terminal count register content will be decremented by one and finally it becomes zero after the required number of DMA cycles are over. In general, if N = the number of desired DMA cycles, load the value N-1 into the low-order 14-bits of the terminal count register.

The bits 14 and 15 of this register indicate the type of DMA operation for that channel. If the device wants to write data into the memory, the DMA operation is called DMA write operation. Bit 14 of the register in this case will be set to one and bit 15 will be set to be zero.

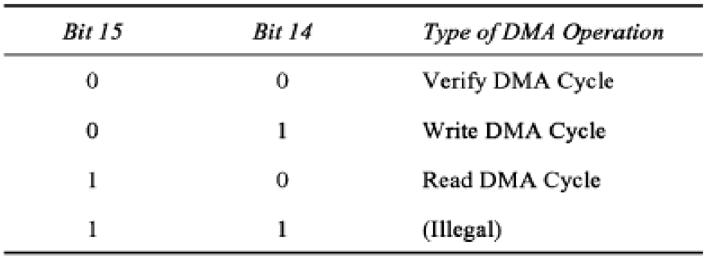
Table: 8257 DMA register selection



Register	*Byte*	*Address Inputs*			*F/L*		*BI-Directional Data Bus*							
		A_3	A_2	A_1	A_0		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CH-0 DMA Address	LSB	0	0	0	0	0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
	MSB	0	0	0	0	1	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
CH-0 Terminal Count	LSB	0	0	0	1	0	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0
	MSB	0	0	0	1	1	Rd	Wr	C_{13}	C_{12}	C_{11}	C_{10}	C_9	C_8
CH-1 DMA Address	LSB	0	0	1	0	0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
	MSB	0	0	1	0	1	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
CH-1 Terminal Count	LSB	0	0	1	1	0	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0
	MSB	0	0	1	1	1	Rd	Wr	C_{13}	C_{12}	C_{11}	C_{10}	C_9	C_8
CH-2 DMA Address	LSB	0	1	0	0	0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
	MSB	0	1	0	0	1	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
CH-2 Terminal Count	LSB	0	1	0	1	0	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0
	MSB	0	1	0	1	1	Rd	Wr	C_{13}	C_{12}	C_{11}	C_{10}	C_9	C_8
CH-3 DMA Address	LSB	0	1	1	0	0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
	MSB	0	1	1	0	1	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
CH-3 Terminal Count	LSB	0	1	1	1	0	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0
	MSB	0	1	1	1	1	Rd	Wr	C_{13}	C_{12}	C_{11}	C_{10}	C_9	C_8
MODE SET	—	1	0	0	0	0	AL	TCS	EW	RP	EN3	EN2	EN1	EN0
(Programme only)														
STATUS (Read only)	—	1	0	0	0	0	0	0	0	UP	TC3	TC2	TC1	TC0

The LSB and MSB of each register for a specific channel has the same address, but they are differentiated by an internal First/Last (F/L) flip flop. If F/L =0, LSB is to be read or written. F/L =1, MSB is to be read or written. The F/L FF needs to be cleared by resetting 8257. The least significant three address bits A0-A2 indicate the specific register for a specific channel. The A3 address line is used to differentiate between all the channel registers and the mode set register, status registers. The higher order bits (A4-A15) may be used to derive the chip select signal CS of 8257.

Table: DMA operation selection using bit 15 and bit 14 of Terminal Count Register



Bit 15	*Bit 14*	*Type of DMA Operation*
0	0	Verify DMA Cycle
0	1	Write DMA Cycle
1	0	Read DMA Cycle
1	1	(Illegal)

Mode Set Register:

This is used for programming the 8257 as per the requirements of the system. The function of it is to enable the DMA channels individually and also to set the various modes of operation. A DMA channel should not be enabled till the DMA Address Register and Terminal Count Register contain valid information, otherwise, an unwanted DMA request may initiate a DMA cycle, probably destroying the valid memory data.

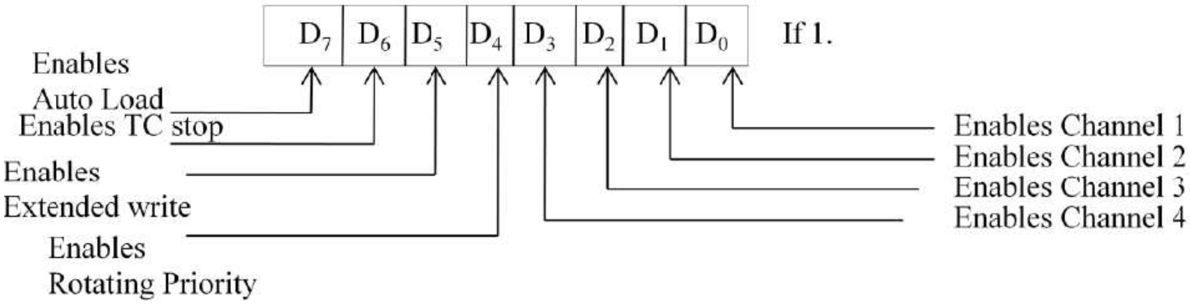


Figure: Bit definitions of the Mode Set Register of 8257

The bits D0-D3 enable one of the four DMA channels. For example, if **D0** is 1, **channel 0 is enabled**. If bit **D4** is set, **rotating priority is enabled**. If the **TC STOP bit** is set, the selected channel is disabled after the TC condition is reached, and it further prevents any DMA cycle in the channel. To enable channel again, this bit must be reprogrammed. If TC STOP bit is programmed to 0, the channel is not disabled, even after the terminal count reaches to zero and further requests are allowed on the same channel.

The **Auto Load bit**, if set, **enables channel 2 for the repeat block chaining operations**, without immediate software intervention between the two successive blocks. The channel 2 registers are used as usual, while the **channel 3 registers are used to store the block reinitialization** parameter, i.e. the DMA starting address and the terminal count. After the first block is transferred using DMA, the channel 2 registers are reloaded with the corresponding channel 3 registers for the next block transfer, if **update flag is set.**

The **extended write bit**, if set to ‘1’, extends the duration of MEMW’ and IO’ signals by activating them earlier. This is useful in interfacing the peripherals with different access times. If the peripheral is not accessed with in the stipulated time, it is expected to give the ‘NOT READY’ indication to 8257, to request it, to add one or more wait states in the DMA cycle. **The mode set register can only be written into.**

Status Register:

The **lower order 4 bits** of this register contains the terminal count status for the four individual channels. If any of these four bits are set, it indicates that the **specific cannel as reached the terminal count condition.** These bits remain set till either the status is read by CPU or 8257 is reset.

The **update flag** is not affected by the read operation. This flag can only be cleared by resetting 8257 or by resetting auto load bit of the mode set register. If **update flag is set**, the contents of the **channel 3 registers are reloaded to the corresponding registers of channel 2**, whenever the channel 2 reached a terminal count condition, after transferring one block and the next block is to be transferred using auto load feature of 8257. The update flag is set every time, the channel 2 registers are loaded with the contents of channel 3 registers. It is cleared by the completion of the first DMA cycle of the new block. **The status register can only be read.**

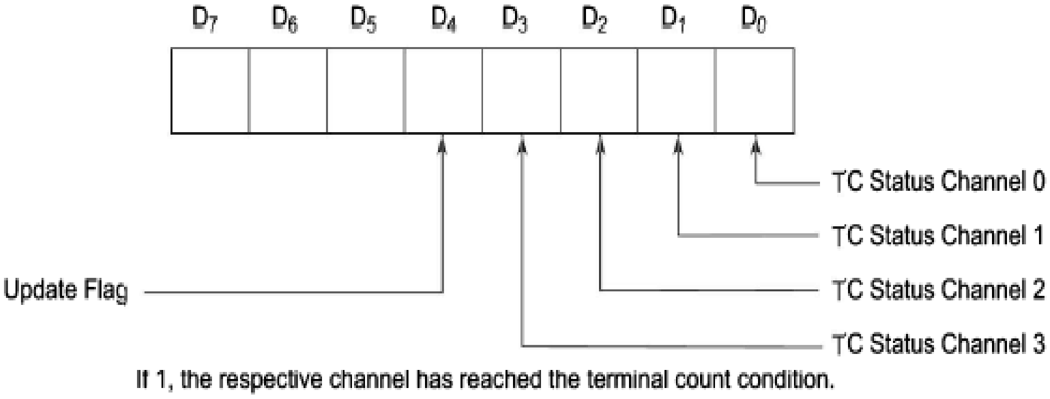


Figure: Bit definitions of status register of 8257

SIGNAL DESCRIPTION OF 8257

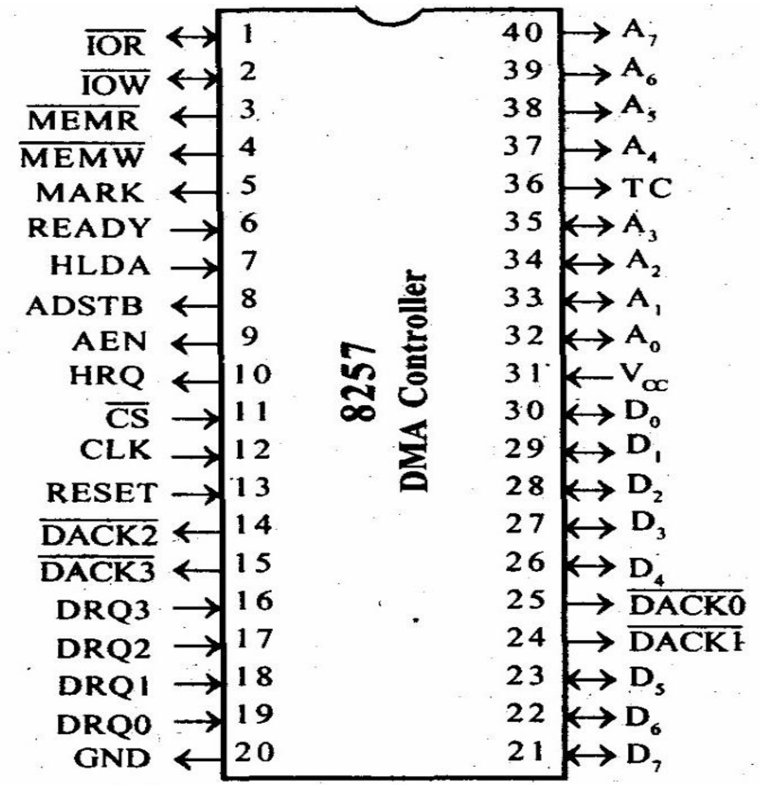


Figure: Pin diagram of 8257

The functional description of each of the signal is given below:

(DRQ_0 - DRQ_3) DMA Request:

These are four individual channel DMA request inputs, used by the peripheral devices to obtain a DMA cycle. In fixed priority mode, DRQ0 has the highest priority and DRQ3 has the lowest.

($\overline{DACK_0}$ - $\overline{DACK_3}$) DMA Acknowledge:

An active low level on the acknowledge output informs the peripheral connected to that channel that it has been selected for a DMA cycle. The DACK output acts as a “chip select” for the peripheral device requesting service. This line goes active (low) and inactive (high) once for each byte transferred even if a burst of data is being transferred.

(D_0 - D_7) Data Bus Lines:

These are bi-directional, data lines used to interface the system bus with the internal data bus of 8257. These lines carry command words to 8257 and status word from 8257, in **slave mode**, i.e. under the control of CPU. The data over these lines may be transferred in both the directions. When the 8257 is the **bus master** (master mode, i.e. not under CPU control), the 8257 uses D_0-D_7 lines to send higher byte of the generated address to the latch. This address is further latched using ADSTB signal. The address is transferred over D_0-D_7 during the first clock cycle of the DMA cycle. During the rest of the period, data is available on the data bus.

($\overline{IOR}$) I/O Read:

An active-low, bi-directional tristate line. In the **slave mode**, it is an input which allows the 8-bit status register or the upper/lower byte of a 16-bit DMA address register or terminal count register to be read. In the **master mode**, IOR is a control output which **is used to access data from a peripheral during the DMA memory write cycle.**

($\overline{IOW}$) 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 a 16-bit DMA address register or terminal count register. In the master mode, IOW is a control output which allows data to be output to a peripheral during a DMA memory read cycle (write to peripheral).

(CLK) Clock Input:

This is a clock frequency input required to derive basic system timings for the internal operation of 8257.

(**RESET) Reset:**

This active high asynchronous input which disables all DMA channels by clearing the mode register and tristates all control lines.

(A0-A3) 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 chip select input line that enables the read or write operations from/to 8257, in the slave mode. In the master mode, CS is automatically disabled to prevent the chip from getting selected (by CPU) while performing the DMA operation.

(A_4-A_7) Address Lines:

This is higher nibble of the lower byte address generated by 8257 during master mode of DMA operation.

(READY) Ready:

This active high asynchronous input is used to stretch the memory read and write cycles in the 8257 with wait states if the selected memory requires longer cycles. READY is used while interfacing slower peripherals.

(HRQ) Hold Request:

This output requests control of the system bus. In systems with only one 8257 (**non-cascaded**), HRQ will normally be applied to the HOLD input on the CPU. In **cascaded mode**, this pin of a slave is connected with a DRQ input line of the master 8257, while the master is connected with HOLD input of the CPU.

(HLDA) Hold Acknowledge:

The CPU drives this input to the DMA controller high, while granting the bus to the device. This pin connected to the HLDA output of the CPU. This input, if high, indicates to the DMA controller that the bus has been granted to the requesting device by the CPU.

($\overline{MEMR}$) Memory Read:

This active-low three-state output is used to read data from the addressed memory locations during DMA Read cycles.

($\overline{MEMW}$) Memory Write:

This active-low three-state output is used to write data to the addressed memory location during DMA Write operation.

(ADSTB) Address Strobe:

This output from 8257 strobes the higher byte of the memory address generated by DMA controller into the 8212 device (latches) from the data bus.

(AEN) Address Enable:

This output is used to disable (float) the system data bus and the system control bus driven by the CPU. It may also be used to disable the system address and data bus by using the enable input of the bus drivers to inhibit the non-DMA devices from responding during DMA operations. This also may be used to transfer the higher byte of the generated address over the data bus. When the **8257 I/O mapped,** this AEN output should be used to disable the selection of other I/O devices, when the DMA controller address is on the address bus.

(TC) Terminal Count:

This output notifies the currently selected peripheral that the present DMA cycle should be the last cycle for the previously programmed 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. The TC pin is activated when the 14-bit content of the terminal count register of the selected channel equals zero. The lower order 14-bits of the terminal count register should be loaded with the values (n-1), where n = 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. The mark will be activated after each 128 cycles or integral multiples of it from the beginning of the data block (the first DMA cycle), if the total number of the required DMA cycles (n) is completely divisible by 128.

Vcc:

This is a +5V supply voltage pin required for the operation of the circuit.

(GND) Ground:

This is a return line for the supply. (Ground pin of the IC).

DMA TRANSFERS AND OPERATIONS

The 8257 is able to accomplish three types of operations, i.e., verify DMA operation, write operation and read operation. The complete operational sequence of 8257 is described using a state diagram given below for a single channel.

Single Byte Transfers:

A single byte transfer using 8257 may be requested by an I/O device using any one of 8257 DRQ inputs. In response, the 8257 sends HRQ signal to the CPU at its HLD input and waits for the acknowledgement at the HLDA input. If the HLDA signal is received by the DMA controller, it indicates that the bus is available for the transfer. The DACK' line of the used channel is pulled down by the DMA controller to indicate the I/O device that its request for DMA transfer has been honoured by the CPU. The 8257 then generates the read and write commands and byte transfer occurs between the selected I/O device and memory. The DACK' line is pulled transfer is over, to indicate the DMA controller that the transfer, requested by the device, is over. The HRQ line is lowered by the DMA controller to indicate the CPU that it may regain the control of the bus. The DRQ must be high until DACK is issued to be recognized and must go LOW before S4 state of the DMA operation state diagram to prevent another unwanted transfer.

Consecutive Transfers:

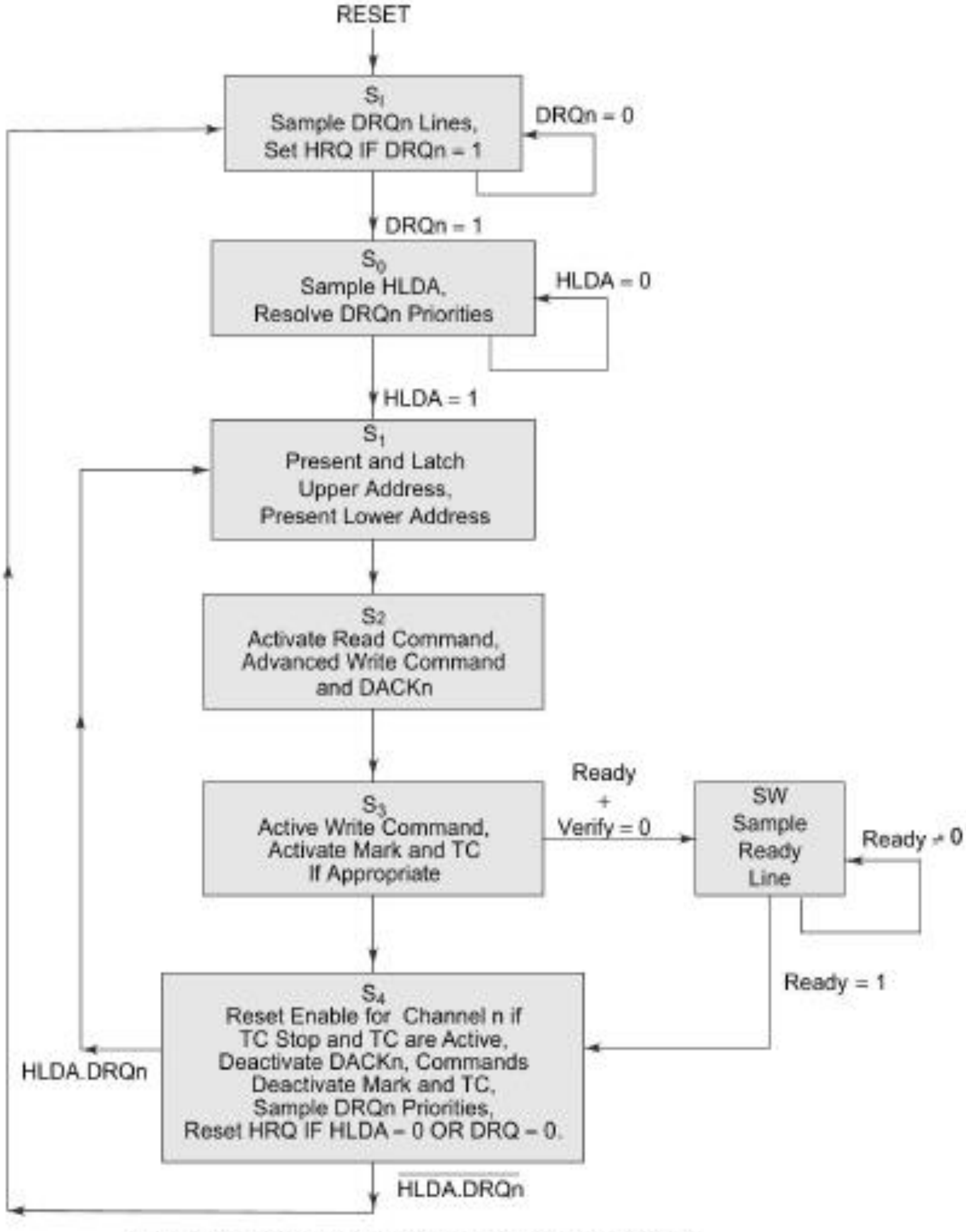
If **more than one channel requests service simultaneously**, the transfer will occur as **burst transfer**. No overhead is required by switching from one channel to another. In each S4, the DRQ lines are sampled and the highest priority request is recognized during the next transfer. A burst mode transfer in a lower priority channel will be overridden by a higher priority request. Once the high priority transfer has completed, the lower priority transfer requests may be served, provided their DRQ lines are still active. The HRQ line is maintained active till all the DRQ lines go low.

Control Override:

The burst or continuous DMA transfer mode described above can be interrupted by an external device by lowering the HLDA line. After each DMA transfer the 8257 checks the HLDA line to insure that it is still active. If it is not active, the 8257 completes the current transfer, releases the HRQ line (LOW) and returns to the idle state. If DRQ lines are still active the 8257 will raise the HRQ line in the third cycle and proceed normally. The 8257 uses four clock cycles to complete a transfer.

Not Ready:

The 8257 has a Ready input to interface it with low speed devices. The Ready line is checked in State 3 of the state diagram. If Ready is LOW the 8257 enters a wait state. Ready is checked during every wait state. When Ready returns HIGH the 8257 proceeds to State 4 to complete the transfer. The 8257 can be interfaced as a memory mapped device or an I/O mapped device. If it is connected as a memory mapped device, proper care must be taken while programming Rd/A15 and Wr/A14 bits in the terminal count register.



DRQn refers to anv DRQ line of an enabled DMA channel

Figure: DMA operation state diagram

PRIORITIES OF THE DMA REQUESTS:

The 8257 can be programmed to select any of the two priority schemes using command register.

1) Fixed Priority scheme
2) Rotating priority scheme

Fixed Priority scheme:

In this scheme, each device connected to a channel is assigned a fixed priority. Here, the DREQ3 has the lowest priority followed by DREQ2 and DREQ1 with next higher priorities, while **DREQ0** has the **highest** priority.

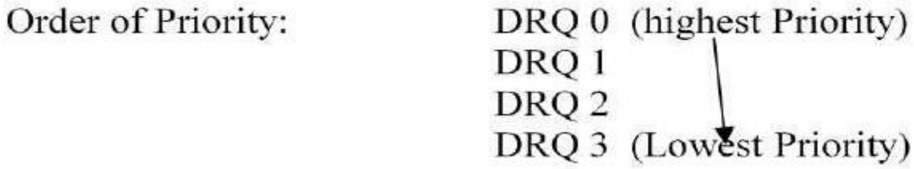


Figure: Priority allotment in fixed priority mode

Rotating Priority scheme:

In the rotating priority mode, the priority of the channels has a circular sequence. At any point of time, suppose DRQ0 has the highest priority and DRQ3 has the lowest, then after the device at channel 0 gets the service, its priority goes down and channel 0 becomes the lowest priority channel. Channel 1 now becomes the highest priority channel, and remain the highest priority channel till it gets the service. Once channel 1 is served, it becomes the lowest priority channel and the channel 2 now becomes the highest priority channel. If you select the rotating priority, in a single chip DMA system and device requesting the service is guaranteed to be recognized after no more than three higher priority requests, thus avoiding dominance of any one channel. The priority allotment in the rotating priority mode is as shown below:

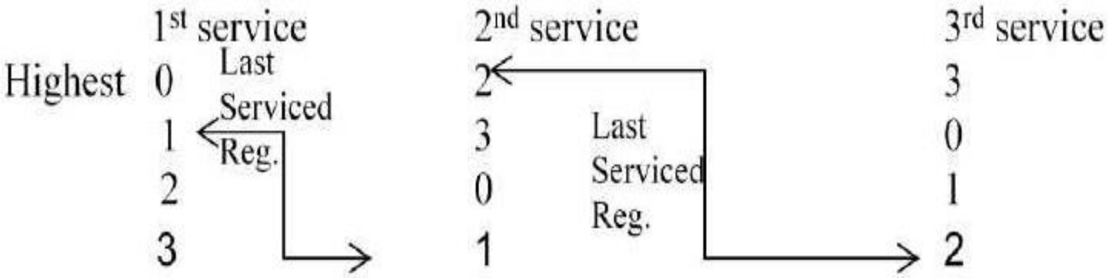


Figure: Priority allotment in rotating priority mode

INTERFACING 8257 DMA WITH 8086

Once a DMA controller is initialized by a CPU properly, it is ready to take control of the system bus on a DMA request, either from a peripheral or itself (in case of memory-to-memory transfers). The DMA controller sends a HOLD request to the CPU and waits for the CPU to assert the HLDA signal. The CPU relinquishes the control of the bus before asserting the HLDA signal. Once the HLDA signal goes high, the DMA controller activates the DACK' to the requesting peripheral and gains the control of the system bus. The DMA controller is the sole master of the bus, till the DMA operation is over. The CPU remains in the HOLD status, till the DMA controller is the master of the bus. A conceptual implementation of the system is shown below:

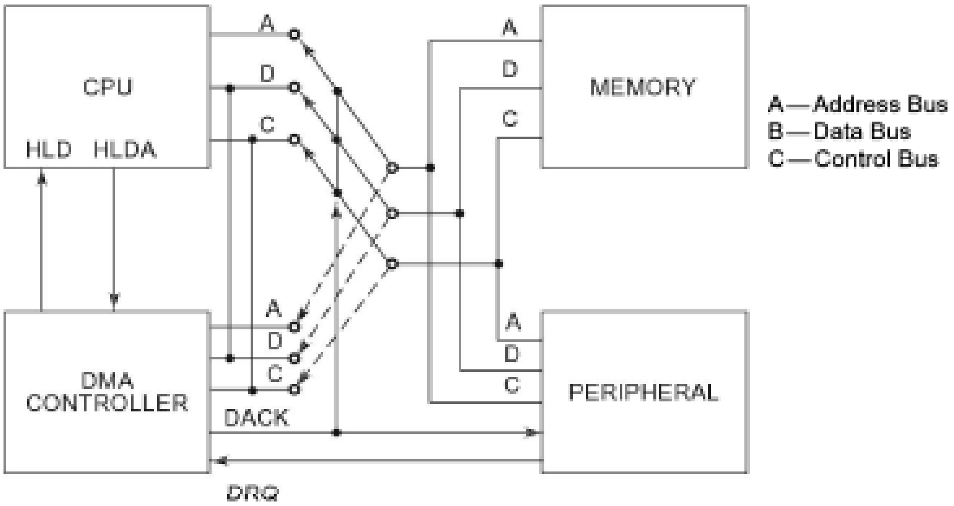


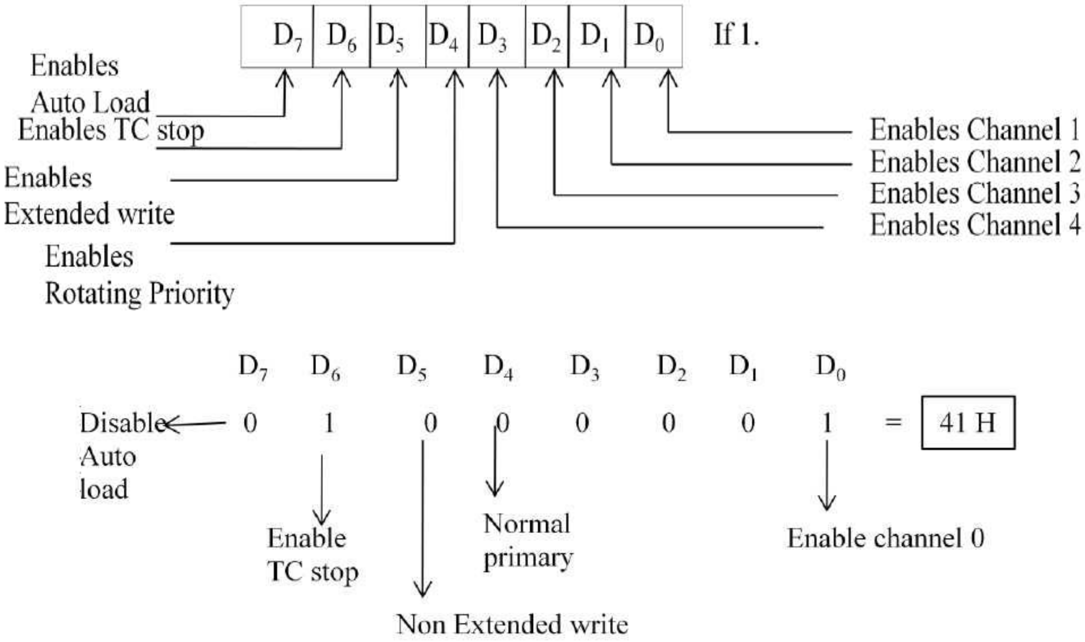
Figure: Interfacing DMA controller with a system

Problem: Interface DMA controller 8257 with 8086 so that the channel 0 DMA addresses register has an I/O address 80H and the mode set register has an address 88H. Initialize the 8257 with normal priority, TC stop and non-extended write. Auto load is not required. The transfer has to take place using channel 0. Write an ALP to move 2KB of data from peripheral device to memory address 2000H: 5000H, with the above initialization.

Solution:

In the problem, it is given as to initialize the 8257 with normal priority, TC stop and non-extended write. Auto load is not required. So the mode set register should be loaded with the value 41H.

Mode set register

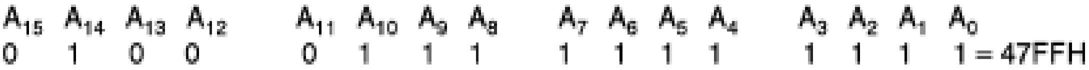


DMA address register:

This should contain the starting address of memory block i.e. 5000H

Terminal count register:

In the problem it is given as, move 2KB of data from peripheral device to memory address 2000H: 5000H, by this we can say it is **DMA write** operation. The bits A14 and A15 are used for specification of type of. Operation**.** Then A15=0 A14 =1. The remaining 14-bits will contain the binary equivalent of the required of DMA cycles i.e. no. of bytes to be transferred minus one.



A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1 = 47FFH

Assembly language program

```
ASSUME       CS: CODE, DS: DATA

DATA          SEGMENT

        MSR EQU 4141H                  ; Mode set register content
        DMAL EQU 0000H                 ; DMA address register lower byte
        DMAH EQU 5050H                 ; DMA address register higher byte
        TCL EQU FFFFH                  ; Terminal count register lower byte
        TCH EQU 4747H                  ; Terminal count register higher byte

DATA ENDS

CODE        SEGMENT

START:      MOV AX, 2000H;
            MOV DS, AX;               Initialize data segment
            MOV AX, MSR;
            OUT 88H, AX;              Initialize MODE SET REGISTER
            MOV AX, DMAL;
            OUT 80H, AX;
            MOV AX, DMAH;
            OUT 80H, AX;              Initialize DMA ADDRESS REGISTER
            MOV AX, TCL;
            OUT 81H, AX;
             MOV AX, TCH;
             OUT 81H, AX;             Initialize TERMINAL COUNT REGISTER
             MOV AH, 4CH;
             INT 21H

CODE         ENDS

             END START
```

8279 PROGRAMMABLE KEYBOARD/DISPLAY or THE KEYBOARD AND DISPLAY CONTROLLER

The disadvantage of interfacing keyboards and displays using 8255 with 8086 is that processor has to refresh the display and check the status of keyboard periodically using polling technique. Thus a considerable amount of CPU time is wasted, reducing the system operating speed and hence the throughput.

Intel's 8279 is a general purpose keyboard display controller that simultaneously drives the display of a system and interfaces a keyboard with the CPU, leaving it free for its routine task. The Keyboard Display interface scans the keyboard to identify if any key has been pressed and sends the code of the pressed key to the CPU. It also transmits the data received from the CPU, to the display device. Both of these functions are performed by the controller in repetitive fashion without involving the CPU.

The Keyboard is interfaced either in the **interrupt or the polled mode**. In the **interrupt mode**, the processor is requested service only if any key is pressed, otherwise the CPU can proceed with its main task. In the **polled mode**, the CPU periodically reads an internal flag of 8279 to check for a key pressure.

The Keyboard section can interface an **array of a maximum of 64 keys with the CPU**. The Keyboard entries (key codes) are debounced and stored in an 8-byte FIFO RAM, which is further accessed by the CPU to read the key codes. If **more than eight characters are entered** in the FIFO (i.e. more than eight keys are pressed), before any FIFO read operation, the **overrun status is set**. If a FIFO contains a valid key entry, the CPU is interrupted (in interrupt mode) or the CPU checks the status (in polling) to read the entry. Once the CPU reads a key entry, the FIFO is updated, i.e. the key entry is pushed out of the FIFO to generate space for new entries.

The 8279 normally provides a maximum of sixteen 7-seg display interface with CPU. It contains a **16-byte display RAM** that can be used either as an integrated block of 16×8-bits or two 16×4-bit block of RAM. The data entry to RAM block is controlled by CPU using the command words of the 8279.

ARCHITECTURE OF 8279

The Keyboard display controller chip 8279 provides

1. A set of four scan lines and eight return lines for interfacing keyboards.
2. A set of eight output lines for interfacing display.

I/O Control and Data Buffer:

The I/O control section controls the flow of data to/from the 8279. The data buffer interface the external bus of the system with internal bus of 8279. The I/O section is enabled only if D' is low. The pin A_0, RD' and WR' select the command, status or data read/write operations carried out by the CPU with 8279.

Control and Timing Register and Timing Control:

These registers store the keyboard and display modes and other operating conditions programmed by CPU. The **registers are written with A_0=1 and WR' =0**. The timing and control unit controls the basic timings for the operation of the circuit. Scan counter divide down the operating frequency of 8279 to derive scan keyboard and scan display frequencies.

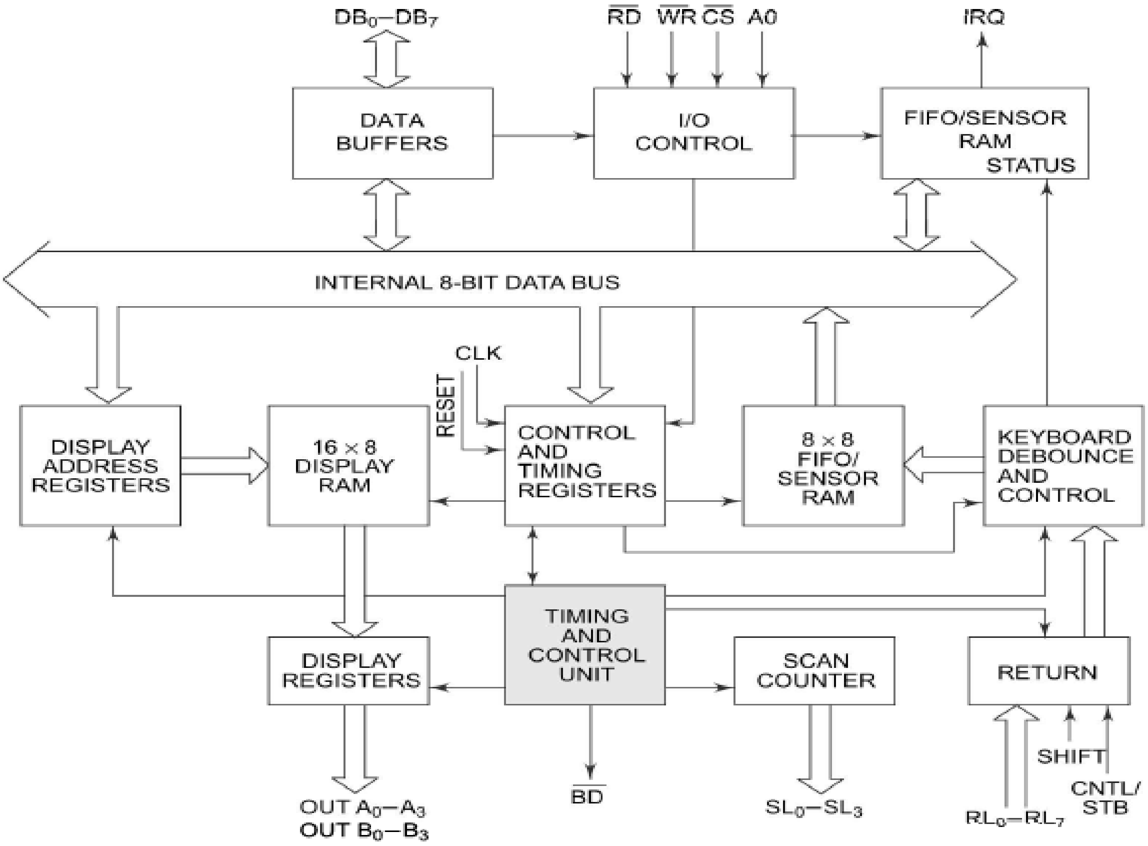


Figure: Internal architecture of 8279

Scan Counter:

The Scan Counter has **two modes** to **scan the key matrix** and **refresh the display**. In the **encoded mode**, the counter provides a binary count that is to be externally decoded to provide the scan lines for keyboard and display (four externally decoded scan lines may drive up to 16 displays). 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 (four internally decoded scan lines may drive up to 4 displays). The keyboard and display both are in the same mode at a time.

Return Buffers and Keyboard Debounce and Control:

This section scans for a **key closure row-wise**. If it is detected, the keyboard debounce unit debounces the key entry (i.e. wait for 10 ms). After the debounce period, if the key continues to be detected. The code of the key is directly transferred to the sensor RAM along with SHIFT and CONTROL key status.

FIFO/Sensor RAM and Status Logic:

In **Keyboard or strobed input mode**, this block acts as 8-byte first-in-first-out (FIFO) RAM. Each key code of the pressed key is entered in the order of the entry, and in the meantime, read by the CPU, till the RAM becomes empty. The status logic generates an interrupt request after each FIFO read operation till the FIFO is empty. In **scanned sensor matrix mode**, this unit acts as sensor RAM. Each row of the sensor RAM is loaded with the status of the corresponding row of sensors in the matrix. If a sensor changes its state, the IRQ line goes high to interrupt the CPU.

Display Address Registers and Display RAM:

The Display address registers hold the addresses of the word currently being written or read by the CPU to or from the display RAM. The contents of the registers are automatically updated by 8279 to accept the next data entry by CPU. The 16-byte display RAM contains the 16-byte of data to be displayed on the sixteen 7-seg displays in the encoded scan mode.

SIGNAL DESCRIPTION OF 8279

The pin diagram of 8279 is given below:

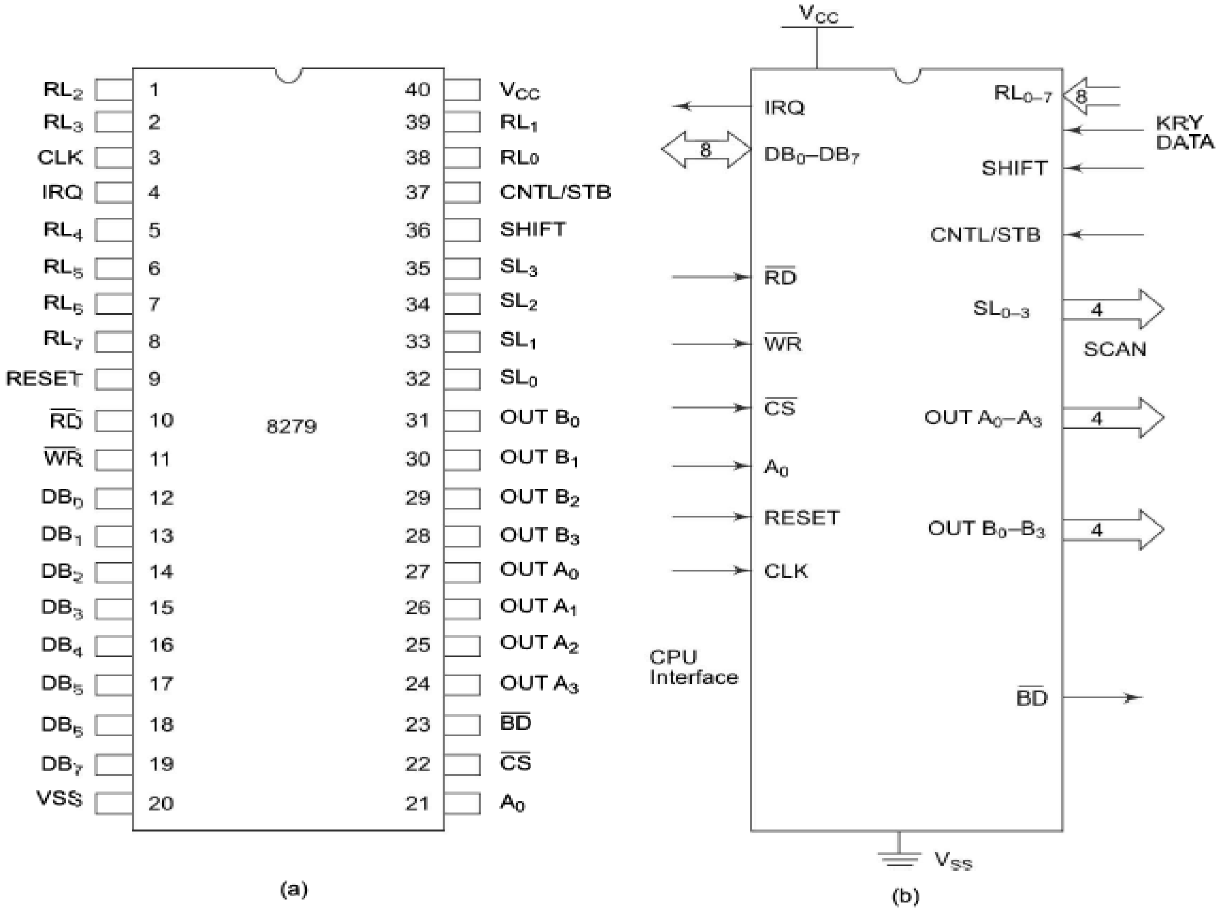


Figure: Pin configuration and logic diagram of 8279

The signal description of each of the pins of 8279 is given below in brief:

DB_0 - DB_7: These are bidirectional data bus lines. The data and command words to and from the CPU are transferred on these lines.

CLK: This is a clock input used to generate internal timings required by 8279.

RESET: This pin is used to reset 8279. A high on this line resets 8279. After resetting 8279, it’s in sixteen 8-bit display, left entry encoded scan, 2-key lock out mode. The clock prescaler is set to 31.

CS’ chip select: A low on this line enables 8279 for normal read or write operations. Otherwise this pin should be high.

A_0: A **high** on the A_0 line indicates the **transfer of a command or status information**. A **low** on this line indicates the **transfer of data**. This is used to select one of the internal registers of 8279.

RD’, WR’: (Input/Output) READ/WRITE input pins enable the data buffer to receive or send data over the data bus.

IRQ: This interrupt output line goes high when there is data in the FIFO sensor RAM. The interrupt line goes low with each FIFO RAM read operation. However, if the FIFO RAM further contains any key-code entry to be read by the CPU, this pin again goes high to generate an interrupt to the CPU.

Vss, Vcc: These are the ground and power supply lines for the circuit.

SL_0-SL_3 – Scan Lines: These lines are used to scan the keyboard matrix and display digits. These lines can be programmed as encoded or decoded, using the mode control register.

RL_0-RL_7 – Return Lines: These are the input lines which are connected to one terminal of keys, while the other terminal of the keys are connected to the decoded scan lines. These are normally high, but pulled low when a key is pressed.

SHIFT: The status of the shift input line is stored along with each key code in FIFO in the scanned keyboard mode. Till it is pulled low with a key closure it is pulled up internally to keep it high.

CNTL/STB - CONTROL/STROBED I/P Mode: In the keyboard mode, this line is used as a control input and stored in FIFO on a key closure. The line is a strobe line that enters the data into FIFO RAM, in the strobed input mode. It has an internal pull up. The line is pulled down with a Key closure.

BD – Blank Display: This output pin is used to blank the display during digit switching or by a blanking command.

$OUTA_0$ – $OUTA_3$ and $OUTB_0$ – $OUTB_3$: These are the output ports for two 16×4 (or one 16 × 8) internal display refresh registers. The data from these lines is synchronized with the scan lines to scan the display and keyboard. The two 4-bit ports may also be used as one 8-bit port.

MODES OF OPERATION OF 8279

The Modes of operation of 8279 are

i. Keyboard (Input) modes

ii. Display (Output) modes

KEYBOARD (INPUT) MODES:

8279 provides three input modes, they are:

1. Scanned Keyboard Mode:

This mode allows a key matrix to be interfaced using either encoded or decoded scans. In the encoded scan, an 8 × 8 keyboard or in 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.

2. Scanned Sensor Matrix:

In this mode, a sensor array can be interfaced with 8279 using either encoder or decoder scans. With encoder scan 8 × 8 sensor matrix or with decoder scan 4 × 8 sensor matrix can be interfaced. The sensor codes are stored in the CPU addressable sensor RAM.

3. Strobed Input:

In this mode, if the control line goes low, the data on return lines, is stored in the FIFO byte by byte.

DISPLAY (OUTPUT) MODES:

8279 provides two output modes for selecting the display options.

1. Display Scan:

In this mode, 8279 provides 8 or 16 character multiplexed displays those can be organized as dual 4-bit or single 8-bit display units.

2. Display Entry:

The 8279 allows options for data entry on the displays. The display data is entered for display either from the right side or from the left side.

Details of Modes of Operation

Keyboard Modes

1. Scanned Keyboard Mode with 2 Key Lockout

In this mode of operation, when a key is pressed, a **debounce logic** comes into operation. During the next two scans, other keys are checked for closure and if no other key is pressed the first pressed key is identified. The key code of the identified key is entered into the FIFO with SHIFT and CNTL status, provided the FIFO is not full, i.e. it has at least one byte free. If the FIFO does not have any free byte, naturally the key data will not be entered and the error flag is set. If FIFO has at least one byte free, the above code is entered into it and the 8279 generates an interrupt (on IRQ line) to the CPU to inform about the previous key closures.

If another key is found closed during the subsequent two scans, no entry to FIFO is made. If all other keys are released before the first key, the key code is entered into FIFO. If the first pressed key is released before the others, the first will be ignored. A key code is entered to FIFO only once for each valid depression, independent of other keys pressed along with it, or released before it. If two keys are pressed within a debounce cycle (simultaneously), no key is recognized till one of them remains closed, and the other is released. The last key that remains depressed is considered as single valid key depression.

2. Scanned Keyboard with N-key Rollover

In this mode, each **key depression is treated independently**. When a key is pressed, the debounce circuit waits for 2 keyboard scans and then checks whether the key is still depressed. If it is still depressed, the code is entered in FIFO RAM. Any number of keys can be pressed simultaneously and recognized in the order, the keyboard scan record them. All the codes of such keys are entered into FIFO. In this mode, the first pressed key need not be released before the second is pressed. All the keys are sensed in the order of their depression, rather in the order of the keyboard scans senses them, and independent of the order of their release.

3. Scanned Keyboard Special Error Mode

This mode is **valid only under the N-Key rollover mode**. This mode is **programmed using end interrupt/error mode set command**. If during a single debounce period (two keyboard scans) two keys are found pressed, this is considered a simultaneous depression and an error flag is set. This flag, if set, prevents further writing in FIFO but allows generation of further interrupts to the CPU for FIFO read. The error flag can be read by reading the FIFO status word. The error flag is set by sending normal clear command with CF=1.

4. Sensor Matrix Mode

In the Sensor Matrix mode, the debounce logic is inhibited. The 8-byte FIFO RAM now acts as 8×8 bit memory matrix. The status of the sensor switch matrix is fed directly to sensor RAM matrix. Thus the sensor RAM bits contains the row-wise and column-wise status of the sensors in the sensor matrix. The IRQ lines goes high, if any change in sensor value is detected at the end of a sensor matrix scan or the sensor RAM has a previous entry to read by the CPU. The IRQ line is reset by the first data read operation, if AI=0, otherwise, by issuing the end interrupt command. AI is a bit in read sensor RAM word.

Display Modes

There are various options of data display. For example, the command number of characters can be 8 or 16, with each character organized as single 8 bit or dual 4 bit codes. Similarly there are two display formats. The first one is known as **left entry mode or type writer mode**. Since in a type writer the first character typed appears at the left-most position, while the subsequent characters appears successively to the right of the first one. The other display format is known as **right entry mode, or calculator mode,** since in a calculator the first character entered appears at the right-most position and this character is shifted one position left when the next character is entered. Thus all the previously entered characters are shifted left by one position when a new character is entered.

1. Left Entry Mode

In the Left entry mode, the data is entered from the left side of the display unit. Address 0 of the display RAM contains the leftmost display character and address 15 of the RAM contains the rightmost display character. It is just like writing in our note books, i.e. from left to write. If the 8279 is in autoincrement mode, the display RAM address is automatically updated with successive reads or writes. The first entry is displayed on the left most display and the sixteenth entry on the right most display. The seventeenth entry is again displayed at the end of the left most displayed.

2. Right Entry Mode

In the right entry mode, the first entry to be displayed is entered on the rightmost display. The next entry is also placed in the right most display but after the previous display is shifted left by one display position. The left most character is shifted out of that display at the seventeenth entry and is lost, i.e. it pushed out of the display RAM.

8251 - UNIVERSAL SYNCHRONOUS AND ASYNCHRONOUS RECEIVER AND TRANSMITTER (USART) or PROGRAMMABLE COMMUNICATION INTERFACE

The 8251A is a universal synchronous asynchronous receiver and transmitter compatible with Intel's processors. This may be programmed to operate in any of the serial communication modes built into it. This chip converts the parallel data into a serial stream of bits suitable for serial transmission. It is also able to receive a serial stream of bits and converts it into parallel data bytes to be read by a microprocessor.

Methods of data communication:

The data transmission between two points involves unidirectional or bidirectional transmission of meaningful digital data through a medium. There are basically three modes of data transmission:

(i) Simplex
(ii) Duplex
(iii) Half Duplex

In **simplex mode**, the data is transmitted only in **one direction over a single communication channel.** For example, a computer (CPU) may transmit data for a CRT display unit in this mode. In **duplex mode**, data may be **transferred between two transreceivers in both directions** simultaneously.

In **half duplex mode**, data transmission may take place in **either direction, but at a time data may be transmitted only in one direction**. For example, a computer may communicate with a terminal in this mode. When the terminal sends data (i.e. terminal is sender), the message is received by the computer (i.e. computer is receiver). However, it is not possible to transmit data from the computer to the terminal and from terminal to the computer simultaneously.

ARCHITECTURE OF 8251 USART

The functional block diagram of 8251A consists five sections. They are: Data bus buffer, Read/Write control logic, Modem control, Transmitter and Receiver.

The **data buffer** interfaces the internal bus of the circuit with the system bus. The **read write control logic** controls the operation of the peripheral depending upon the operations initiated by the CPU. This unit selects one of the two internal addresses those are control address or data address at the behest of the C/D' signal. The **modem control unit** handles the modem handshake signals to coordinate the communication between the modem and USART.

The **transmit control unit** transmits the data byte received by the data buffer from the CPU for further serial communication. This decides the transmission rate which is controlled by the TXC' input frequency. This unit also derives two transmitter status signals namely TXRDY and TXEMPTY. These may be used by the CPU for handshaking. The transmit buffer is a parallel to serial converter that receives a parallel byte for conversion into a serial signal and further transmission onto the communication channel. The **receive control unit** decides the receiver frequency as controlled by the RXC' input frequency. This unit generates a receiver ready (RXRDY) signal that may be used by the CPU for hand shaking. This unit also detects a break in the data string while 8251 is in asynchronous mode. In synchronous mode, the 8251 detects SYNC characters using SYNDET/BD pin.

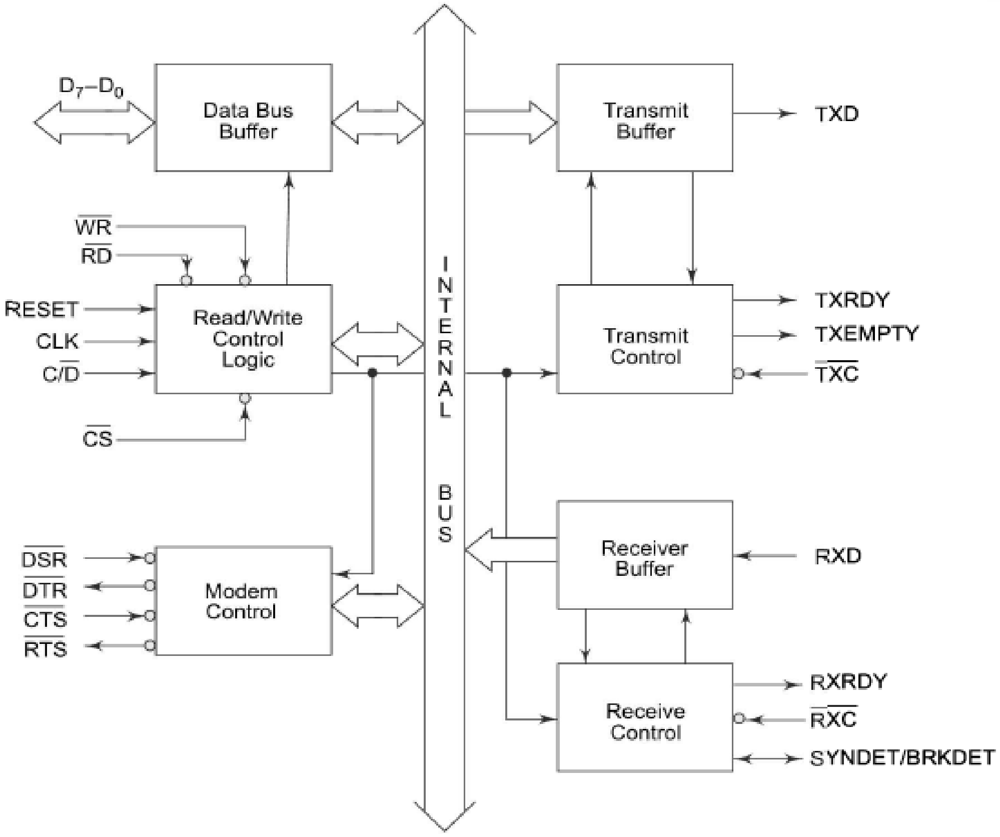


Figure: 8251 internal architecture

SIGNAL DESCRIPTION OF 8251

The pin configuration of 8251 is shown below:

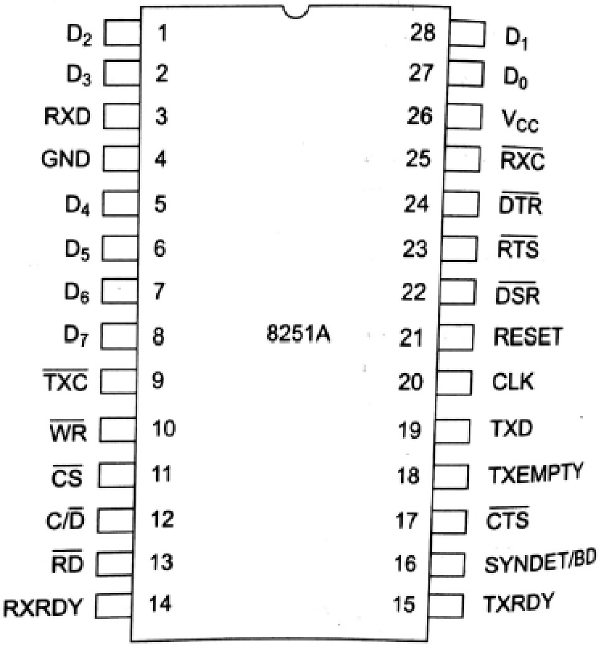


Figure: Pin configuration of 8251

D_0 to D_7 (I/O terminal)

This is an 8-bit bidirectional data bus used to read or write status, command word or data from or to the 8251.

C/D’ (Input terminal)

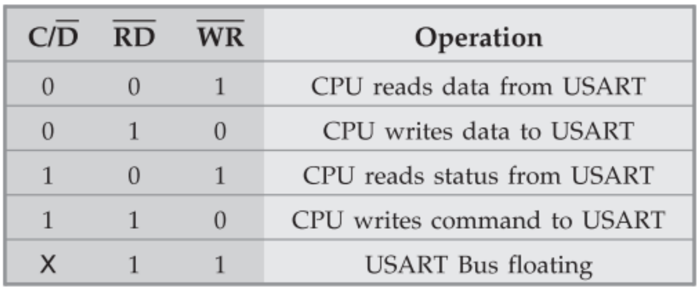
This is an input terminal together with RD’ and WR’ inputs, informs the 8251 that the word on the data bus is either a data or command words/status words. If C/D’ = low, data will be accessed. If C/D’ = high, command word or status word will be accessed.

RD’ (Input terminal)

This active low input to 8251 is used to inform it that the CPU is reading either data or status information from its internal registers.

WR’ (Input terminal)

This active low input to 8251 is used to inform it that the CPU is writing either data or control word to 8251.



$C/\overline{D}$	$\overline{RD}$	$\overline{WR}$	Operation
0	0	1	CPU reads data from USART
0	1	0	CPU writes data to USART
1	0	1	CPU reads status from USART
1	1	0	CPU writes command to USART
X	1	1	USART Bus floating

CS’ (Input terminal)

This is an active low chip select input of 8251. If it is high, no read or write operation can be carried out on 8251. The data bus is tristated if the pin is high.

CLK (Input terminal)

CLK signal is used to generate internal device timings and is normally connected to clock generator output. This input frequency should be at least 30 times greater than the receiver or transmitter data bit transfer rate.

RESET (Input terminal)

A high on this input forces the 8251 into idle state. The device will remain idle till this input signal goes low and new set of control word is written into it. The minimum required reset pulse width is **six clock states**, for the proper reset operation.

TXC’: Transmitter Clock Input (Input terminal)

This transmitter clock input controls the rate at which the character is to be transmitted. In **synchronous transmission mode**, the **baud rate** will be the **same as the frequency of TXC’**. In **asynchronous mode**, it is possible to select the baud rate factor by mode instruction. It can be **1, 1/16 or 1/64** of the **TXC’**. The serial data is shifted out on the successive negative edge of TXC’.

TXD: Transmitted Data Output (output terminal)

This is an output pin carries serial stream of transmitted data bits along with other information like start bit, stop bits and parity bit etc.

RXC’: Receiver Clock Input (Input terminal)

This is a clock input signal which determines the transfer speed of received data. In synchronous mode, the baud rate is the same as the frequency of RXC’. In asynchronous mode, it is possible to select the baud rate factor by mode instruction. It can be 1, 1/16, 1/64 the RXC’. The received data is read into the 8251 on the rising edge of RXC’.

RXD: Receive Data Input (input terminal)

This input terminal which receives composite stream of the data to be received by 8251.

RXRDY: Receive Ready Output (Output terminal)

This output indicates that the 8251 contains a character to be read by the CPU. The RXRDY signal may be used either to interrupt the CPU or may be polled by the CPU. To set the RXRDY signal in **asynchronous mode**, the receiver must be enabled to sense a start bit and a complete character must be assembled and then transferred to the data output register.

In **synchronous mode**, to set the RXRDY signal, the receiver must be enabled and a character must finish assembly and then be transferred to the data output register. If the data is not successfully read from the receiver data output register before assembly of the next data byte, the **overrun condition error flag is set** and the previous byte is overwritten by the next byte of the incoming data and hence it is lost.

TXRDY: Transmitter Ready (output terminal)

This output signal indicates to the CPU that the internal circuit of the transmitter is ready to accept a new character for transmission from the CPU. The TXRDY signal is set by the leading edge of the write signal if a data character is loaded into it from the CPU. In the polled operation, the TXRDY status bit will indicate the empty or full status of the transmitter data input register.

DSR’: Data Set Ready (Input terminal)

This input may be used as a general purpose one bit inverting input port. Its status can be checked by the CPU using status read operation. This is normally used to check if the data set is ready when communicating with a modem.

DTR’: Data Terminal Ready (Output terminal)

This output may be used as a general purpose one bit inverting output port. This can be programmed low using the command word. This is used to indicate that the device is ready to accept data when the 8251 is communicating with a modem.

RTS’: Request to Send Data (Output terminal)

This output also may be used as a general purpose one bit inverting output port that can be programmed low to indicate the modem that the receiver is ready to receive a data byte from the modem. This signal is used to communicate with a modem.

CTS’: Clear to Send (Input terminal)

If the clear to send the input line is low, the 8251 is enabled to transmit the serial data, provided the enable bit in the command byte is set to ‘1’. If a Tx disable or CTS’ disable command occurs, while the 8251 is transmitting data, the transmitter transmits all the data written to the USART prior to disabling the CTS’ or Tx. If the CTS’ disable or Tx disable command occurs just before the last character appears in the serial bit string, the character will be retransmitted again whenever the CTS' is enabled or the Tx enable occurs.

TXE: Transmitter Empty (Output terminal)

While transmitting, if the 8251 has **no characters to transmit**, the **TXE output goes high** and it automatically goes low when a character is received from the CPU, for further transmission. In the synchronous mode, a high on this output line indicates that a character has not been loaded and the SYNC character or characters are about to be or being transmitted automatically as **fillers.** The TXE signal can be used to **indicate the end of a transmission mode.**

SYNDET/BD: Synch Detect / Break Detect (Input or output terminal)

This pin is used in **synchronous mode** for detection of synchronous characters (SYNDET) and may be used as either input or output. This can be programmed using the control word. After resetting, it is in the output mode. When used as an output, the SYNDET pin will go high to indicate that 8251 has located a SYNC character in the receive mode. The SYNDET signal is automatically reset upon a following status read operation. When this is used as an input, a positive going signal will cause the 8251 to start assembling data character on the rising edge of the next RXC.

In the **asynchronous mode**, this pin acts as a break detector output. This goes high whenever the RXD pin remains low through two consecutive stop bit sequences. A stop bit sequence contains a stop bit, a start bit, data bits and parity bits. This is reset only with master chip reset or the RXD returning high. If the RXD returns to '1' during the last bit of the next character after the break, the break detect is latched up. The 8251 can be cleared only with chip reset.

OPERATING MODES OF 8251- SYNCHRONOUS AND ASYNCHRONOUS MODES

The 8251A is Universal Synchronous, Asynchronous, Receiver, and Transmitter can be programmed to operate in its various modes using its mode control words. A set of control words is written into the internal registers of 8251A to make it operate in the desired mode.

Once the 8251 is programmed as required, the TXRDY output is raised high to signal the CPU that the 8251 is ready to receive a data byte from it that is to be further converted to serial format and transmitted. This automatically goes low when CPU writes a data byte into 8251. In receiver mode, the 8251 receives a serial data byte from a modem or an I/O device. After receiving the entire data byte, the RXRDY signal is raised high to inform the CPU that the 8251 has a character ready for it. The RXRDY signal is automatically reset after the CPU reads the received byte from the 8251. The 8251 cannot initiate transmission until the TX enable bit in the command word is set and a CTS' signal is received from modem or receiving I/O device.

The control words of 8251 are divided into two functional types:

1. Mode instruction control word
2. Command instruction control word

ASYNCHRONOUS MODE:

Mode Instruction Control Word: This defines the general operational characteristics of 8251A. After internal (reset command) or external (reset input pin) reset, this must be written to configure the 8251A as per the required operation. Once this has been written into 8251A, SYNC characters or command instructions may be programmed further as per requirements. To change the mode of operation from synchronous to asynchronous or vice versa, the 8251A has to be reset using master chip reset.

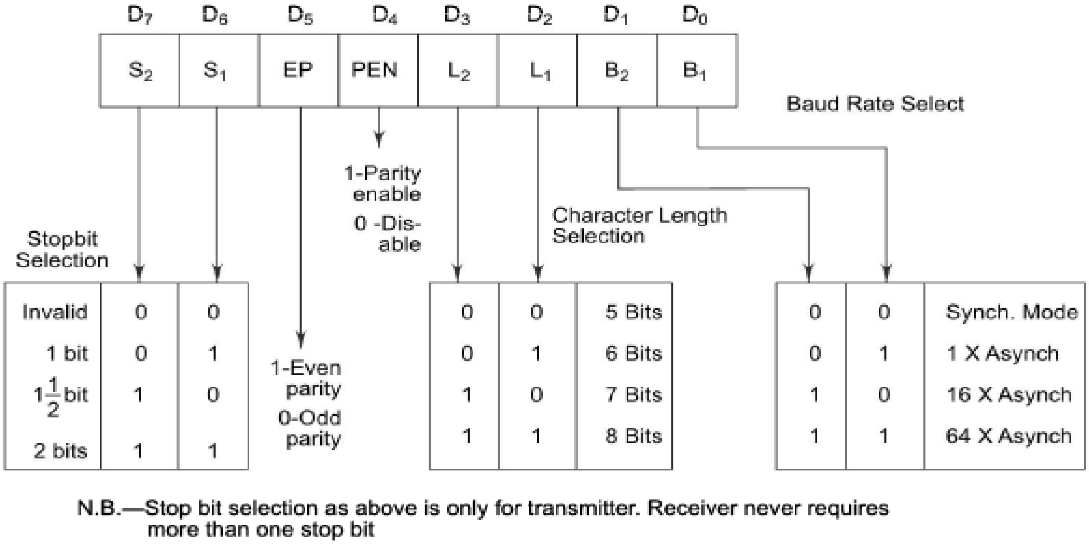


Figure: Mode instruction format- Asynchronous mode

Asynchronous Mode (Transmission)

When a data character is sent to 8251A by the CPU, it adds start bits prior to the serial data bits, followed by optional parity bit and stop bits using the asynchronous mode instruction control word format. This sequence is then transmitted using TXD output pin on the falling edge of TXC’. When no data characters are sent by the CPU to 8251A the TXD output remains ‘high’, if a ‘break’ has not been detected.

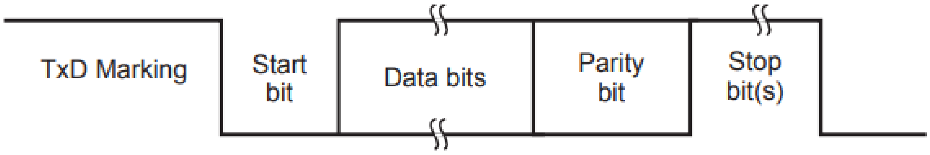


Figure: Transmitter output in asynchronous mode

Asynchronous Mode (Receive)

A falling edge on RXD input line marks a start bit. At baud rates of 16× and 64×, this start bit is again checked at the center of the start bit pulse and if detected low, it is a valid start bit which starts counting. The bit counter locates the data bits, parity bit and stop bit. If a **parity error** occurs, the parity error flag is set. If a low level is detected as the stop bit, the **framing error** flag is set. The receiver requires only one stop bit to mark end of data bit string, regardless of the stop bit programmed at the transmitting end. This 8-bit character is then loaded into parallel I/O buffer of 8251A. RXRDY pin is then raised high to indicate to the CPU that a character is ready for it. If the previous character has not been read by the CPU, the new character replaces it, and the **overrun flag** is set indicating that the previous character is lost. These error flags can be cleared using an error reset instruction. If the character length is 5 to 7 bits then the remaining bits are set to 0.

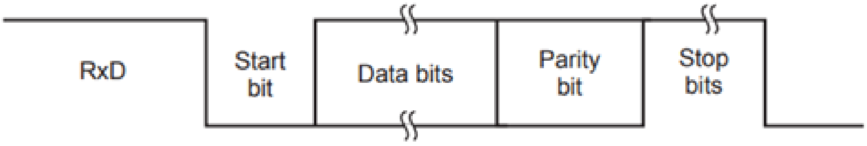


Figure: Receiver input in asynchronous mode

The below figure shows asynchronous mode transmission and receiver data formats.

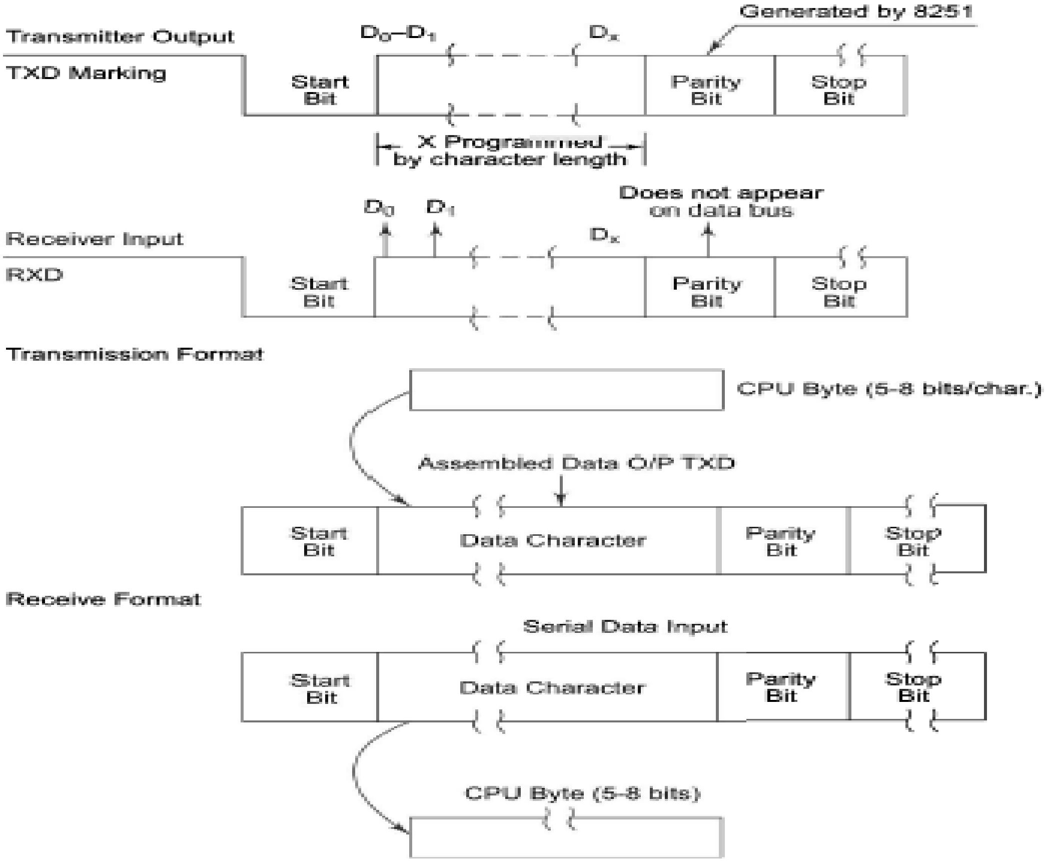
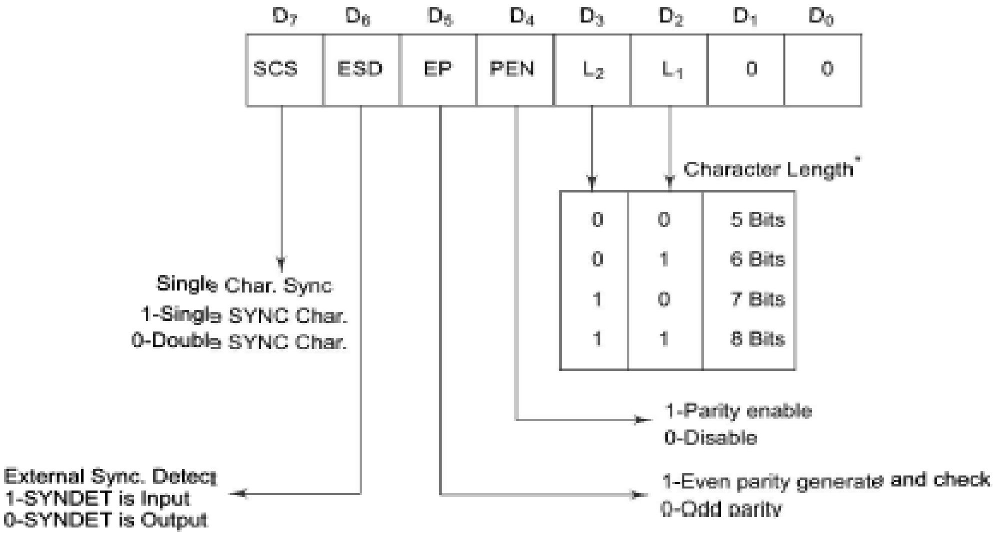


Figure: Asynchronous mode transmission and receiver data formats

SYNCHRONOUS MODE

Mode Instruction Control Word: The synchronous mode instruction format with its bit definitions are given below:



* If the character size less than 8-bits, the remaining bits are set to 'O'.

Figure: Mode instruction format- Synchronous mode

The below figure shows the synchronous mode transmit and receive data formats.

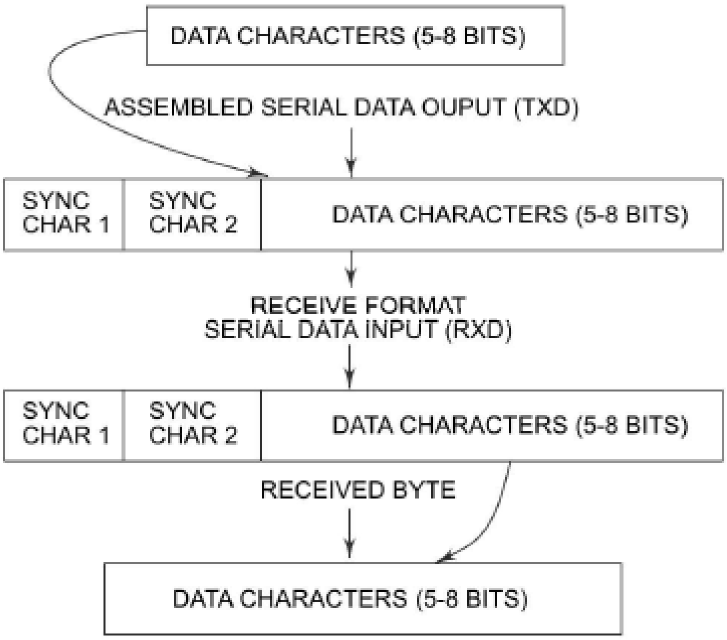

DATA CHARACTERS (5-8 BITS)
ASSEMBLED SERIAL DATA OUPUT (TXD)
SYNC CHAR 1
SYNC CHAR 2
DATA CHARACTERS (5-8 BITS)
RECEIVE FORMAT
SERIAL DATA INPUT (RXD)
SYNC CHAR 1
SYNC CHAR 2
DATA CHARACTERS (5-8 BITS)
RECEIVED BYTE
DATA CHARACTERS (5-8 BITS)

Figure: Data formats of synchronous mode

Synchronous Mode (Transmission)

The TXD output is high until the CPU sends a character to 8251A which usually is a SYNC character. When CTS' line goes low, the first character is serially transmitted out. All the characters are shifted out on the falling edge of TXC'. Data is shifted out at the same rate as TXC', over the TXD output line. If the CPU buffer becomes empty, the SYNC character or characters are inserted in the data stream over TXD output. The TXEMPTY pin is raised high to indicate that the 8251A is empty (i.e. it does not have any byte to transmit) and is transmitting SYNC characters. The TXEMPTY pin is reset, automatically when a data character is written to 8251A by the CPU. The below figure shows the relation between TXEMPTY and SYNC character insertion.

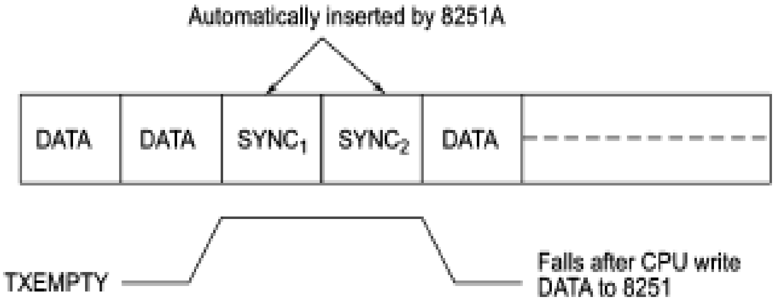

Automatically inserted by 8251A
DATA
DATA
SYNC$_1$
SYNC$_2$
DATA
TXEMPTY
Falls after CPU write DATA to 8251

Figure: TXEMPTY signal and SYNC characters

Synchronous Mode (Receiver)

Internal SYNC:

In this mode, character synchronization can be achieved internally or externally. If this mode is programmed, then ENTER HUNT command should be included in the first command instruction word written into the 8251A. The data on RXD pin is sampled on the rising edge of the RXC'.

The **content of the receiver buffer is compared with the first SYNC character** at every edge until it matches. If the 8251A is programmed for two SYNC characters, the subsequent received character is also checked. When **both the characters match**, the **hunting stops**. The SYNDET pin is set high and is reset automatically by a status read operation.

External SYNC:

In the external SYNC mode, synchronization is achieved by applying a high level on the SYNDET input pin, which forces 8251A out of the HUNT mode. The high level can be removed after one RXC' cycle. An ENTER HUNT command has no meaning in asynchronous mode. The parity and overrun error both are checked in the same way as in asynchronous mode.

Command Instruction Control Word: The command instruction controls the actual operations of the selected format like enable transmit/receive, error reset and modem control. Once the mode instruction has been written into 8251A and the SYNC characters are inserted internally by the 8251A, all further control words are written with C/D' =1 will load a command instruction. A reset operation returns back to mode instruction format. The command instruction format is given in the below figure with its bit definitions.

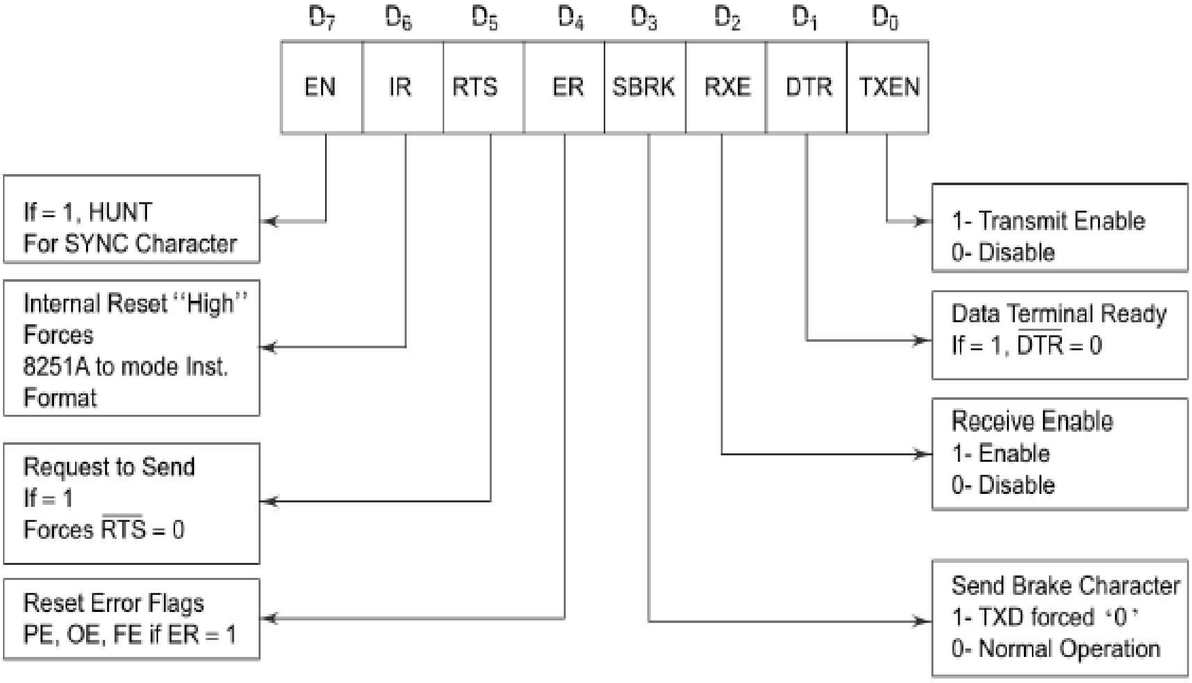

D_7
D_6
D_5
D_4
D_3
D_2
D_1
D_0
EN
IR
RTS
ER
SBRK
RXE
DTR
TXEN
If = 1, HUNT For SYNC Character
Internal Reset "High" Forces 8251A to mode Inst. Format
Request to Send If = 1 Forces $\overline{RTS}$ = 0
Reset Error Flags PE, OE, FE if ER = 1
1- Transmit Enable 0- Disable
Data Terminal Ready If = 1, $\overline{DTR}$ = 0
Receive Enable 1- Enable 0- Disable
Send Brake Character 1- TXD forced '0' 0- Normal Operation

Figure: Command Instruction Format

Status Word Register: This status read definition is used by the CPU to read the status of the active 8251A to confirm if any error condition or other conditions like the requirement of processor service has been detected, during the operation.

A read command is issued by processor with C/D'=1 to accomplish this function. Some of the bits in this register have the same significance as those of the pins of 8251A. These are used to interface the 8251A in a polled configuration, besides the interrupt controlled mode. The pin TXRDY is an exception. The status read format is given in the below figure with its bit definitions.

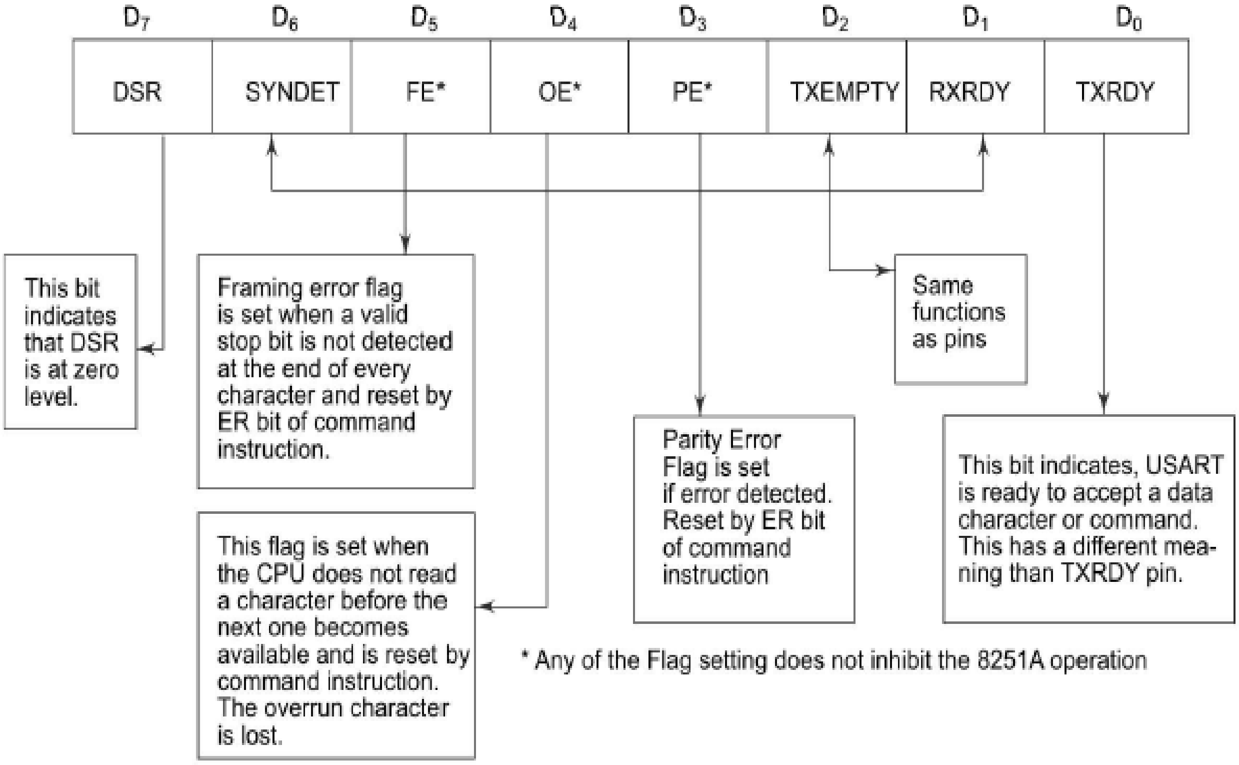

D7
D6
D5
D4
D3
D2
D1
D0
DSR
SYNDET
FE*
OE*
PE*
TXEMPTY
RXRDY
TXRDY
This bit indicates that DSR is at zero level.
Framing error flag is set when a valid stop bit is not detected at the end of every character and reset by ER bit of command instruction.
This flag is set when the CPU does not read a character before the next one becomes available and is reset by command instruction. The overrun character is lost.
Parity Error Flag is set if error detected. Reset by ER bit of command instruction
Same functions as pins
This bit indicates, USART is ready to accept a data character or command. This has a different meaning than TXRDY pin.
* Any of the Flag setting does not inhibit the 8251A operation

Figure: Status Read Instruction Format

Error Definitions

Parity Error: At the time of transmission of data an even or odd parity bit is inserted in the data stream. At the receiver end, if parity of the character does not match with the pre-defined parity, PARITY ERROR occurs.

Overrun Error: In the receiver section, received character is stored in the receiver buffer. The CPU is supposed to read this character before reception of the next character. But if CPU fails in reading the character loaded in the receiver buffer, the next received character replaces the previous one and the OVERRUN error occurs.

Framing Error: If valid stop bit is not detected at the end each character, FRAMING ERROR occurs.

All these errors, when occur, set the corresponding bits in the status register. These error bits are reset by setting ER bit in the command instruction.

INTERFACING AND PROGRAMMING 8251A WITH 8086:

Problem:

Design the hardware interface circuit for interfacing 8251 with 8086. Set the 8251A in asynchronous mode as a transmitter and receiver with even parity enabled, 2 stop bits, 8-bit character length, frequency 160 KHz and baud rate 10K.

(a) Write an ALP to transmit 100 bytes of data string starting at location 2000:5000H

(b) Write an ALP receive 100 bytes of data string and store it at 3000:4000H

Solution:

Asynchronous mode control word

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
S_2	S_1	EP	PEN	L_2	L_1	B_2	B_1	
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	1	1	1	1	1	1	0	= 0FE H
2 stop bits		Even parity enabled		8-bit format		CLK scaled by 16		

(a) Write an ALP to transmit 100 bytes of data string starting at location 2000:5000H

```
ASSUME CS : CODE
CODE SEGMENT
START:  MOV AX, 2000H        ; DS points to byte string segment
        MOV DS, AX
        MOV SI, 5000H        ; SI points to byte string
        MOV CL, 64H          ; length of the string in CL(hex)
        MOV AL, 0FEH         ; Mode control word out to
        OUT 0FEH, AL         ; D0–D7.
        MOV AX, 11H          ; Load command word
        OUT 0FEH, AL         ; to transmit enable and error reset
  WAIT:  IN AL, 0FEH         ; Read status,
        AND AL, 01H          ; check transmitter enable
        JZ WAIT              ; bit, if zero wait for the transmitter to be ready
        MOV AL, [SI]         ; If ready, first byte of string data
        OUT 0FCH, AL         ; is transmitted.
        INC SI               ; Point to next byte.
        DEC CL               ; Decrement counter.
        JNZ WAIT             ; If CL is not zero, go for next byte.
        MOV AH, 4CH          ; If CX is zero, return to DOS
        INT 21H        ; highlighted instructions can be replaced with INT 03H
CODE ENDS
END START
```

(b) Write an ALP receive 100 bytes of data string and store it at 3000:4000H

```
ASSUME CS : CODE
CODE SEGMENT
START:  MOV AX, 3000H      ; Data segment set to 3000H
        MOV DS, AX
        MOV SI, 4000H      ; Pointer to destination offset
        MOV CL, 64H        ; Byte count in CL
        MOV AL, 7EH        ; Only one stop bit for receiver is set
```

```
            OUT 0FEH, AL
            MOV AL, 14H         ; Load command word to enable the receiver and disable
transmitter
            OUT 0FEH, AL
NXTBT: IN AL, 0FEH              ; Read status
            AND 38H             ; Check FE, OE and PE
            JZ READY            ; If zero, jump to READY
            MOV AL, 14H         ; If not zero, clear them
            OUT 0FEH, AL
READY:  IN AL, 0FEH             ; Check RXRDY. If the receiver is not ready, wait
            AND 02H
            JZ READY
            IN AL, 0FCH         ; If it is ready, receive the character
            MOV [SI], AL        ; Store the received character
            INC SI              ; Increment pointer to next byte
            DEC CL              ; Decrement counter
            JNZ NXTBT           ; If CL is 0, return to DOS
 CODE ENDS
END START
```

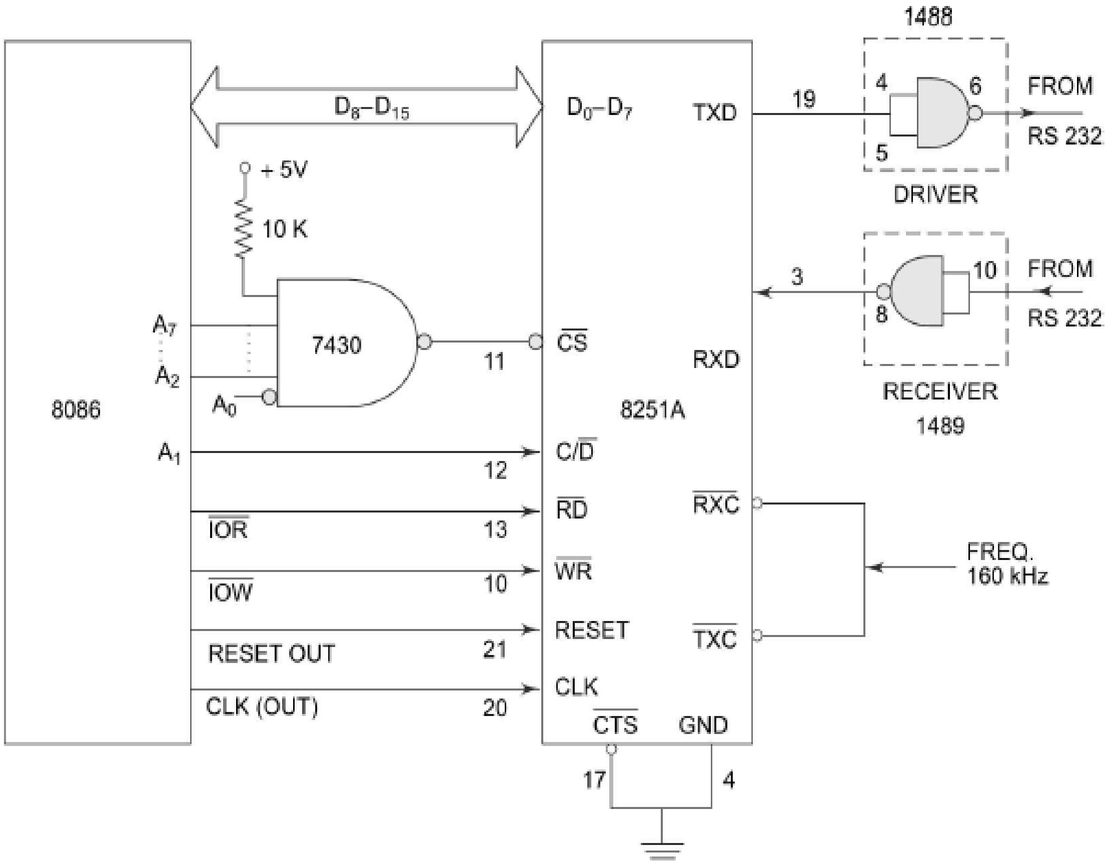


Figure: Interfacing 8251A with 8086