DATA COMMUNICATION
Data Communications

- Data communications refers to the ability of one computer to exchange data with another computer or a peripheral.
- Standard data communication interfaces and standards are needed.
- Centronics’s parallel printer interface.
- RS-232 defines a serial communications standard.
- 8251 USART (Universal Synchronous/Asynchronous Receiver/Transmitter) is the key component for converting parallel data to serial form and vice versa.
- Two types of serial data communications are widely used:
  - Asynchronous communications
  - Synchronous communications
Parallel/Serial Transmissions

Parallel Transmission

Serial Transmission
Communication Modes

- When data is transmitted between two pieces of equipment, 3 modes of communication are used:
  - **Simplex**
    - Data is transmitted in one direction only.
  - **Half Duplex**
    - This is used when two devices want information alternatively, but one after another.
  - **Full Duplex**
    - This is used when data is to be exchanged between two devices in both directions simultaneously.
Communication Modes

Simplex

Transmitter → Receiver

Half Duplex

Transmitter ↔ Receiver

Full Duplex

Transmitter → Receiver → Transmitter
Transmission Modes

- For receiving device to interpret bit pattern correctly, it must able to determine the following:
  - Bit Synchronization
    Start of each bit cell period
  - Character Synchronization
    Start and end of each character or byte
  - Frame Synchronization
    Start and end of each complete message block (frame)

- Types Synchronization
  - Asynchronous Transmission
  - Synchronous Transmission
Asynchronous Transmission

- In Asynchronous Transmission receiver clock runs (RxC) in unsynchronized with respect to the incoming signal (RxD)
- Additional start and stop bits are added in character (byte) data
- State of signal on transmission line between characters is idle
Asynchronous communications

- In asynchronous communications, the data, such as ASCII characters, are packed between a start bit and a stop bit, a process called **framing**.

  ASCII character "A", binary \texttt{0100 0001}, framed between the start bit and 2 stop bits.

  - The **start** bit is always one bit and always a 0. (\textit{low})
  - The **stop** bit can be one or two bits, and is 1 (\textit{high}).
Asynchronous Transmission

**Example:**

Construct the transmitted frame using *asynchronous transmission mode* which contains the following data: **GO**. Assume that the number of stop bits is 2 and parity bit is used.
Asynchronous Transmission

- **Baud** *(signaling rate)* is used to define number of line signal transition per second.

- **Bit rate** is the number of bits transmitted per second.

- **Special case**: *(Baud = bit rate)* when a signal has only two levels: 0 or one.

- **Example**: A signaling rate of 300 baud with 4 bits per signaling element would yield a bit rate of 1200 bps.
Asynchronous Transmission

Principle of Operation and Timing:

- TxC
- TxD
  - Start bit
  - 7/8 bit per character
  - Stop bit(s)

- RxD
  - Actual edge within one clock cycle

- RxC
  - Time
  - Idle
Asynchronous Transmission

Bit Synchronization in Asynchronous Transmission:

- The local receiver clock is $N$ times the transmitted bit rate ($N=16$ is common).
- The first $1 \rightarrow 0$ transition is associated with the start bit.
- Each bit is sampled at the center to avoid delay distortion problem.
- After the first transition is detected, the signal is sampled after $N/2$ clock cycles and then subsequently after $N$ clock cycles for each bit in the character.
Asynchronous Transmission

Bit Synchronization in Asynchronous Transmission:

\[ N = 1 \]

- **Start bit**
- **1st data bit**
- **2nd data bit**

**Rx D**

**Rx C**

*(×1)*

**Sampling pulse**

**Bit rate counter**

preset to 1

**Actual bit cell centers**

**Time**
Asynchronous Transmission

Bit Synchronization in Asynchronous Transmission:

- Start bit
- 1st data bit
- 2nd data bit

- RxD
- RxC (×4)
- Sampling pulse
- 2 RxC periods
- Bit rate counter preset to 2
- Bit rate counter preset to 4
- 4 RxC periods
- Actual bit cell centers

N = 4
Asynchronous Transmission

Principle of operation and Timing:

- Transmitter:
  - PISO = Parallel-in, serial-out
  - TXD = Transmit Data out
  - Transmit clock (TXC)

- Receiver:
  - SIPO = Serial-in, parallel-out
  - RXD = Receive Data in
  - Receive clock (RXC = N x TXC)
  - Counter

Parallel-in:
- msb
- lsb

Serial-out:
- TXD
- RXD

Diagram details and definitions are provided in the image.
Asynchronous Transmission

Character Synchronization in Asynchronous Transmission:
- After the start bit is detected, the receiver achieves character synchronization simply by counting the programmed number of bits.
Synchronous Transmission

- The complete block or frame of data is transmitted as a contiguous stream with no delay between each 8-bit element.
Bit Synchronization using Synchronous Transmission:

- With synchronous transmission, the receiver clock (RxC) operates in synchronism with the received data signal (RxD).

- **Clock Encoding and Extraction:** The clock information is embedded into the transmitted signal and subsequently extracted by the receiver.
Synchronous Transmission

Bit-Oriented Synchronous Transmission:

- Line idle
- Opening flag
- Frame contents
- Closing flag

Diagram showing transmitter and receiver with zero bit insertion and deletion, enable/disable, PISO, TxC, TxD, RxD, RxC, and SIPO.
RS232

- For compatibility in data communication equipment, an interfacing standard called RS232 was set by the Electronics Industries Association (EIA) in 1960.
- Today’s most widely used serial I/O interface standard

Pins and their labels for the RS232 cable, which is commonly referred to as the DB-9 connector.

Table 17-1: IBM PC DB-9 Signals

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data carrier detect (DCD)</td>
</tr>
<tr>
<td>2</td>
<td>Received data (RxD)</td>
</tr>
<tr>
<td>3</td>
<td>Transmitted data (TxD)</td>
</tr>
<tr>
<td>4</td>
<td>Data terminal ready (DTR)</td>
</tr>
<tr>
<td>5</td>
<td>Signal ground (GND)</td>
</tr>
<tr>
<td>6</td>
<td>Data set ready (DSR)</td>
</tr>
<tr>
<td>7</td>
<td>Request to send (RTS)</td>
</tr>
<tr>
<td>8</td>
<td>Clear to send (CTS)</td>
</tr>
<tr>
<td>9</td>
<td>Ring indicator (RI)</td>
</tr>
</tbody>
</table>

Fig DB9 9-Pin Connector
Digital Data Transmission using MODEM
Digital Data Transmission using MODEM

DTE = DATA TERMINAL EQUIPMENT
DCE = DATA COMMUNICATION EQUIPMENT
Sequence of Modem Control Signals

- After the PC power is turned on and the PC runs any self-checks, it asserts the **data-terminal-ready (DTR/)** signal to tell the modem it is ready.

- When the modem is powered up and ready to transmit and receive data, the modem will assert the **data-set-ready (DSR/)**.

- The calling PC sends the telephone number of the modem associated with the called computer. The modem then dials up the called computer.

- When the called modem receives ring tones, it will set the **ring indicator (RI)** line to **on** and the called computer responds by setting the **request-to-send (RTS/)** line **on**.

- In response, the called modem sends a carrier signal - the data tone for a binary 1- to the calling modem to indicate that the call has been accepted by the called computer.
Sequence of Modem Control Signals

- After a short delay to allow the calling modem to prepare to receive data, the called modem sets the **clear-to-send (CTS/)** line **on** to inform the called computer that it can start sending data.

- On detecting the carrier signal, the calling modem sets the **carrier-detect (CD/)** line **on**.

- The called computer starts by sending a short message over the set-up connection.

- When this message has been sent, it prepares itself to receive the response from the calling terminal by setting the **RTS/** line **off**, and on detecting this, the called modem stops sending the carrier signal and sets the **CTS/** line **off**.

- At the calling side, the removal of carrier signal is detected by the calling modem and, in response, it sets the **CD/** line **off**.

- The calling PC sets **RTS/** line **on** in order to send the response message and, on receipt of the **CTS/** signal from the modem, starts to send the message.
Digital Data Transmission using MODEM
8251 PROGRAMMABLE COMMUNICATION INTERFACE
Introduction

- 8251 is a USART (Universal Synchronous Asynchronous Receiver Transmitter) for serial data communication.

- Programmable peripheral designed for synchronous /asynchronous serial data communication, packaged in a 28-pin DIP.

- Receives parallel data from the CPU & transmits serial data after conversion.

- Also receives serial data from the outside & transmits parallel data to the CPU after conversion.
Block diagram of the 8251 USART
Pin diagram

8251A

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>D2</td>
</tr>
<tr>
<td>2</td>
<td>D3</td>
</tr>
<tr>
<td>3</td>
<td>RXD</td>
</tr>
<tr>
<td>4</td>
<td>GND</td>
</tr>
<tr>
<td>5</td>
<td>D4</td>
</tr>
<tr>
<td>6</td>
<td>D5</td>
</tr>
<tr>
<td>7</td>
<td>D6</td>
</tr>
<tr>
<td>8</td>
<td>D7</td>
</tr>
<tr>
<td>9</td>
<td>TXC</td>
</tr>
<tr>
<td>10</td>
<td>WR</td>
</tr>
<tr>
<td>11</td>
<td>CS</td>
</tr>
<tr>
<td>12</td>
<td>C/D</td>
</tr>
<tr>
<td>13</td>
<td>RD</td>
</tr>
<tr>
<td>14</td>
<td>RXRDY</td>
</tr>
<tr>
<td>15</td>
<td>TXRDY</td>
</tr>
<tr>
<td>16</td>
<td>SYNDET/BD</td>
</tr>
<tr>
<td>17</td>
<td>CTS</td>
</tr>
<tr>
<td>18</td>
<td>TXEMPTY</td>
</tr>
<tr>
<td>19</td>
<td>TXD</td>
</tr>
<tr>
<td>20</td>
<td>CLK</td>
</tr>
<tr>
<td>21</td>
<td>RESET</td>
</tr>
<tr>
<td>22</td>
<td>DSR</td>
</tr>
<tr>
<td>23</td>
<td>RTS</td>
</tr>
<tr>
<td>24</td>
<td>DTR</td>
</tr>
<tr>
<td>25</td>
<td>RXC</td>
</tr>
<tr>
<td>26</td>
<td>VCC</td>
</tr>
<tr>
<td>27</td>
<td>D0</td>
</tr>
<tr>
<td>28</td>
<td>D1</td>
</tr>
</tbody>
</table>
Signals of 8251

- **CS** – Chip Select: When signal goes low, the 8251A is selected by the MPU for communication.

- **C/D** – Control/Data: When signal is high, the control or status register is addressed; when it is low, data buffer is addressed. (Control register & status register are differentiated by WR and RD signals)

- **WR** : When signal is low, the MPU either writes in the control register or sends output to the data buffer.

- **RD** : When signal goes low, the MPU either reads a status from the status register or accepts data from data buffer.

- **RESET** : A high on this signal reset 8252A & forces it into the idle mode.

- **CLK** : Clock input, usually connected to the system clock for communication with the microprocessor.
Signals of 8251

- The 8251A is **doubled-buffered**. This means that one character can be loaded into a **data-out buffer register** while another character is being shifted out of the actual **transmit shift register**.

- The **TxRDY** output of the 8251A will go high when:
  - The **data-out buffer register** is Empty for another character from the CPU.
  - The **CTS/** input has been asserted low.
  - The **transmit-enable (TxEN)** bit of the 8251A's command word is set.

- The **TxEMPTY** output of the 8251A will go high when both the **data-out buffer register** and the **transmit shift register** are empty.

- The **RxRDY** output of the 8251A will go high when:
  - The **data-in buffer register** is full and is ready to be read by the CPU.
  - The **receive-enable (RxE)** bit of the 8251A's command word is set.

- If the CPU does not read a character from the **data-in buffer register** before another character is shifted in, the first character will be overwritten and lost.
Signals of 8251

- The sync-detect/break-detect (SYNDET/BD) pin has two uses:

  1. When the device is operating in **asynchronous mode**, the pin will go high if the serial data input line (RxD) stays low for more than 2 character times (i.e., the RxD remains low through two consecutive stop bit sequences including the start bits, data bits, and parity bits). This signal then indicates an intentional break in data transmission. It is reset only upon chip **RESET** or **RxD** returning to a "one" state.

  2. When the device is operating in **synchronous mode**, the SYNDET pin can be programmed as either input or output. When used as an output, then SYNDET pin will go high to indicate that the 8251A has located the **SYN** character in the receiver mode. If the 8251 A is programmed to use double **SYN** characters, then the **SYNDET** will go high in the middle of the last bit of the second **SYN** character. **SYNDET** is automatically reset upon a status read operation. If the search for **SYN** characters is conducted by an external device, then **SYNDET** can be used to input a signal, indicating that a match has been found by the external device.
Sections of 8251

- Data Bus buffer
- Read/Write Control Logic
- Modem Control
- Transmitter
- Receiver

1. Data Bus Buffer

- D0-D7 : 8-bit data bus used to read or write status, command word or data from or to the 8251A
2. Read/Write Control logic

- Includes a control logic, six input signals & three buffer registers: Data register, control register & status register.

- Control logic: Interfaces the chip with MPU, determines the functions of the chip according to the control word in the control register & monitors the data flow.
3. 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.
4. Transmitter section

- Accepts parallel data from MPU & converts them into serial data.

- Has two registers:
  - Buffer register: To hold eight bits
  - Output register: To convert eight bits into a stream of serial bits.
- The MPU writes a byte in the buffer register.
- Whenever the output register is empty; the contents of buffer register are transferred to output register.
- Transmitter section consists of three output & one input signals
  - TxD - Transmitted Data Output : Output signal to transmit the data to peripherals
  - TxC - Transmitter Clock Input : Input signal, controls the rate of transmission.
  - TxRDY - Transmitter Ready : Output signal, indicates the buffer register is empty and the USART is ready to accept the next data byte.
  - TxE - Transmitter Empty : Output signal to indicate the output register is empty and the USART is ready to accept the next data byte.
5. Receiver Section

- Accepts serial data on the RxD pin and converts them to parallel data.
- Has two registers:
  - Receiver input register
  - Buffer register
- When RxD goes low, the control logic assumes it is a start bit, waits for half bit time, and samples the line again. If the line is still low, the input register accepts the following data, and loads it into buffer register at the rate determined by the receiver clock.

- **RxRDY** - Receiver Ready Output: Output signal, goes high when the USART has a character in the buffer register & is ready to transfer it to the MPU.

- **RxD** - Receive Data Input: Bits are received serially on this line & converted into a parallel byte in the receiver input register.

- **RxC** - Receiver Clock Input: Clock signal that controls the rate at which bits are received by the USART.
Control Register

- 16-bit register for a control word consist of two independent bytes namely mode word & command word.
- Mode word : Specifies the general characteristics of operation such as baud, parity, number of bits etc.
- Command word : Enables the data transmission and reception.
- Register can be accessed as an output port when the Control/Data pin is high.
**Status register**

- Checks the ready status of the peripheral.

- Status word in the status register provides the information concerning register status and transmission errors.

**Data register**

- Used as an input and output port when the C/D is low

<table>
<thead>
<tr>
<th>CS</th>
<th>C/D</th>
<th>WR</th>
<th>RD</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>MPU reads data from data buffer</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>MPU writes data from data buffer</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>MPU writes a word to control register</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>MPU reads a word from status register</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Chip is not selected for any operation</td>
</tr>
</tbody>
</table>
Interfacing 8251 to 8088

<table>
<thead>
<tr>
<th>C/D</th>
<th>RD/</th>
<th>WR/</th>
<th>CS/</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>8251A DATA → DATA BUS</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>DATA BUS → 8251A DATA</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>STATUS → DATA BUS</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>DATA BUS → Control</td>
</tr>
<tr>
<td>X</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>DATA BUS → 3-STATE</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>DATA BUS → 3-STATE</td>
</tr>
</tbody>
</table>
8251 Communication Interface
Initializing 8251

- C/D = 1
  - Mode Instruction
  - Sync Character 1
  - Sync Character 2
  - Command Instruction
  - Data
- C/D = 0
  - Data
  - Command Instruction

Flowchart:
- Reset Operation
  - Put Output in Mode Register
  - Asynchronous
    - 2 sync char
      - Put Output in Sync Char 1
      - Put Output in Sync Char 2
    - A_0=1
      - Put Output in data-out buffer register
- Reset
  - Put Output in Control Register
Initializing 8251

- To implement serial communication the MPU must inform the 8251 about the mode, baud, stop bits, parity etc. A set of control words must be loaded.
  - Mode Words
    - Specifies general characteristics of the operation.
  - Command Words
    - Enables the data transmission and/or reception
  - Status Word provides the information concerning register status and transmission errors.

- Any control word written into the control register after a mode word is interpreted as a command word; that means a command word can be changed anytime, however 8251 should be reset prior to writing a Mode word.

- 8251 can be reset internally by using the Internal Reset Bit D6.
8251 Mode Word

Asynchronous

Synchonous
8251 Command Word

- **EN**: Transmit Enable
  - 1 = enable
  - 0 = disable

- **IR**: Receive Enable
  - 1 = enable
  - 0 = disable

- **RTS**: Data Terminal Ready
  - “high” will force DTR output to zero

- **ER**: Receive Enable
  - 1 = enable
  - 0 = disable

- **SBRK**: Send Break Character
  - 1 = forces TxD “low”
  - 0 = normal operation

- **AXE**: Error Reset
  - 1 = reset error flags
  - PE, OE, FE

- **DTR**: Request to Send
  - “high” will force RTS output to zero

- **TxE**: Internal Reset
  - “high” returns 8251A to Mode Instruction Format

- **ENTER HUNT MODE**: 1 = enable search for Sync Characters
8251 Command Word

- Initializing the **TxEN** bit to 1 will enable the transmitter section of the 8251A and the **TxRDY** output.

- Initializing **DTR**/bit to 1 will cause the DTR/ output of the 8251A to be asserted low. This signal is used to tell a modem that a PC or terminal is operational.

- Initializing **RxE** bit to 1 will enable the RxRDY output of the 8251A.

- Initializing **SBRK** bit to 1 will cause the 8251A to output characters of 0's including start bits, data bits, and parity bits (**break character**). A break character is used to indicate the end of block of transmitted data.

- Initializing **ER** bit to 1 will cause the 8251A to reset the **parity**, **overrun**, and **framing** error flags in the 8251A status register.

- Initializing **RTS** bit to 1 will cause the 8251A to assert its **request-to-send (RTS/**) output low. This signal is sent to a modem to ask whether a modem and the receiving system are ready for a data character to be sent.

- Initializing **IR** bit to 1 will cause 8251A to be internally reset. After the software- reset command, a new mode word must be sent.

- Initializing **EH** bit to 1 will cause 8251A to enter hunt mode (search for **SYN** characters, and is used only in synchronous mode.)
## 8251 Status Word

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSR</td>
<td>SYNDET</td>
<td>FE</td>
<td>OE</td>
<td>FE</td>
<td>TxE</td>
<td>RxRDY</td>
<td>TxRDY</td>
</tr>
</tbody>
</table>

### Data Set READY
- **DSR** is general purpose.
- Normally used to test modem conditions such as Data Set Ready.

### SYNC DETECT
- When set for internal sync detect indicates that character sync has been achieved and 8251 is ready for data.

### Framing Error
- *Asynchronous Only*
- FE flag is set when a valid stop bit is not detected at end of each character. It is reset by ER bit of Command instruction. FE does not inhibit operation of 8251.

### OVERRUN ERROR
- The OE flag is set when the CPU does not read a character before the next one becomes available. It is reset by the ER bit of the Command instruction. OE does not inhibit operation of 8251; however the previously overrun character is lost.

### PARITY ERROR
- PE flag is set when a parity error is detected. It is reset by the ER bit of the Command instruction. OE does not inhibit operation of 8251.

### TRANSMITTER READY
- Indicates USART is ready to accept a data character or command.

### RECEIVE READY
- Indicates USART has received a character on its serial input and is ready to transfer it to the CPU.

### TRANSMITTER EMPTY
- Indicates that parallel to serial converter in transmitter is empty.
Simple Serial I/O Procedures

- **Read**

  1. **start**
  2. Check RxRDY
  3. Is it logic 1?
     - Yes: Read data register*
     - No: return to Check RxRDY

* This clears RxRDY

- **Write**

  1. **start**
  2. Check TxRDY
  3. Is it logic 1?
     - Yes: Write data register*
     - No: return to Check TxRDY

* This clears TxRDY
Fig. 2 Bit Configuration of Mode Instruction (Asynchronous)
Fig. 3 Bit Configuration of Mode Instruction (Synchronous)
Fig. 4 Bit Configuration of Command

Note: Search mode for synchronous characters in synchronous mode.
Fig. 5 Bit Configuration of Status Word

Parity Different from TXRDY Terminal. Refer to "Explanation" of TXRDY Terminals.

Same as terminal. Refer to "Explanation" of Terminals.

1…Parity Error

1…Overrun Error

1…Framing Error

Note: Only asynchronous mode. Stop bit cannot be detected.

Shows Terminal DSR
1…DSR = 0
0…DSR = 1
8251 A Serial Communication Interface

- The 8251A internally interprets the C/D, RD and WR signals as follows:

<table>
<thead>
<tr>
<th>C/D (=Ao)</th>
<th>RD</th>
<th>WR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- Whether the mode, control or sync character register is selected depends on the accessing sequence.
Example 1

- A program sequence which initializes the mode register and gives a command to enable the transmitter and begin an asynchronous transmission of 7-bit characters followed by an even-parity bit and 2 stop bits is:

  \[
  \begin{align*}
  \text{MOV AL,} & \quad 11111010B \\
  \text{OUT 51H,} & \quad \text{AL} \\
  \text{MOV AL,} & \quad 00110011B \\
  \text{OUT 51H,} & \quad \text{AL}
  \end{align*}
  \]


Example 2

• This sequence assumes that the mode and control registers are at address 51H and the clock frequencies are to be 16 times the corresponding baud rates.

The sequence:

```
MOV AL,00111000B
OUT 51H,AL
MOV AL,16H
OUT 51H,AL
OUT 51H,AL
MOV AL,10010100B
OUT 51H,AL
```

would cause the same 8251A to be put in synchronous mode and to begin searching for two successive ASCII sync characters.
Format of the status register

- **DSR**
  - When 1 it indicates the DSR pin is active.
  - Indicates data-out buffer register empty; unlike the **RxRDY** pin, this bit is not affected by CTS pin and **TxEN** control bit.

- **SYNDET**
  - Same as **SYNDET** pin.

- **FE**

- **OE**

- **PE**

- **TxE**
  - Same as **RxRDY** pin.

- **RxDY**
  - Same as **TxRDY** pin.

- **TxRDY**

When 1 it indicates the occurrence of a:
- Framing error
- Overrun error
- Parity error
Example 3

- A typical program sequence which uses programmed I/O to input 80 characters from the 8251A, whose data buffer register's address is 0050, and put them in the memory buffer beginning at LINE.
MOV  AL,00110101B ;ENABLE TRANSMITTER AND RECEIVER
OUT  51H,AL     ;AND CLEAR ERROR BITS
MOV  DI,0       ;INITIALIZE INDEX
MOV  CX,80      ;PUT COUNT IN CX

BEGIN:  IN  AL,51H
TEST  AL,02H
JZ   BEGIN
IN  AL,50H     ;INPUT CHARACTER AND
MOV  LINE[DI],AL ;PUT IN LINE BUFFER
INC  DI
IN  AL,51H     ;CHECK ERROR
TEST  AL,00111000B ;BITS AND
JNZ  ERROR ;IF NO ERROR IS FOUND
LOOP  BEGIN ;CONTINUE INPUTTING
JMP  SHORT EXIT

ERROR:  CALL  NEAR PTR ER_ROUT ;ELSE CALL ERR_ROUT
EXIT:    .
Example 4

- 1000 000 0: data register address: xx80h
- 1000 000 1: control or status register address: xx81h
- Mode word:
  - 2 stop bits. no parity, 8 bit characters. Baud rate factor of 16 (1200 Kbps)
  - 1110 1110 = EEh
- Command Word:
- 0001 0101 = 15h; enable TxRDY and RxRDY and reset all flags first
Initialize the Mode Word and Command Word

Receive Ready?

If Ready get data

Send data if the T buffer register is available

INIT8251: MOV AL, 0EEh
OUT 81h, AL
MOV AL, 15h
OUT 81h, AL

CHKRX: IN AL, 81h
ROR AL, 1
ROR AL, 1
JNC CHKRX

IN AL, 80h
NOT AL
MOV BL, AL

CHKTX: IN AL, 81h
ROR AL, 1
JNC CHKTX
OUT 80h, AL
JMP CHKRX
Thank You