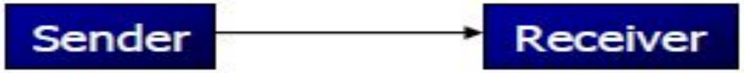
8251 USART & RS232 Interface

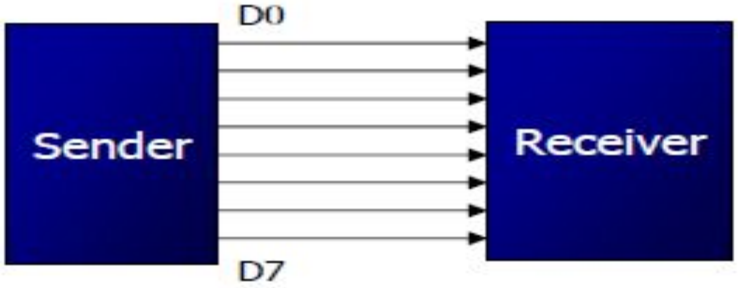
Serial Communication

- Computers transfer data in two ways:
  - Parallel
    - Often 8 or more lines (wire conductors) are used to transfer data to a device that is only a few feet away
  - Serial
    - To transfer to a device located many meters away, the serial method is used
    - The data is sent one bit at a time

Serial Transfer



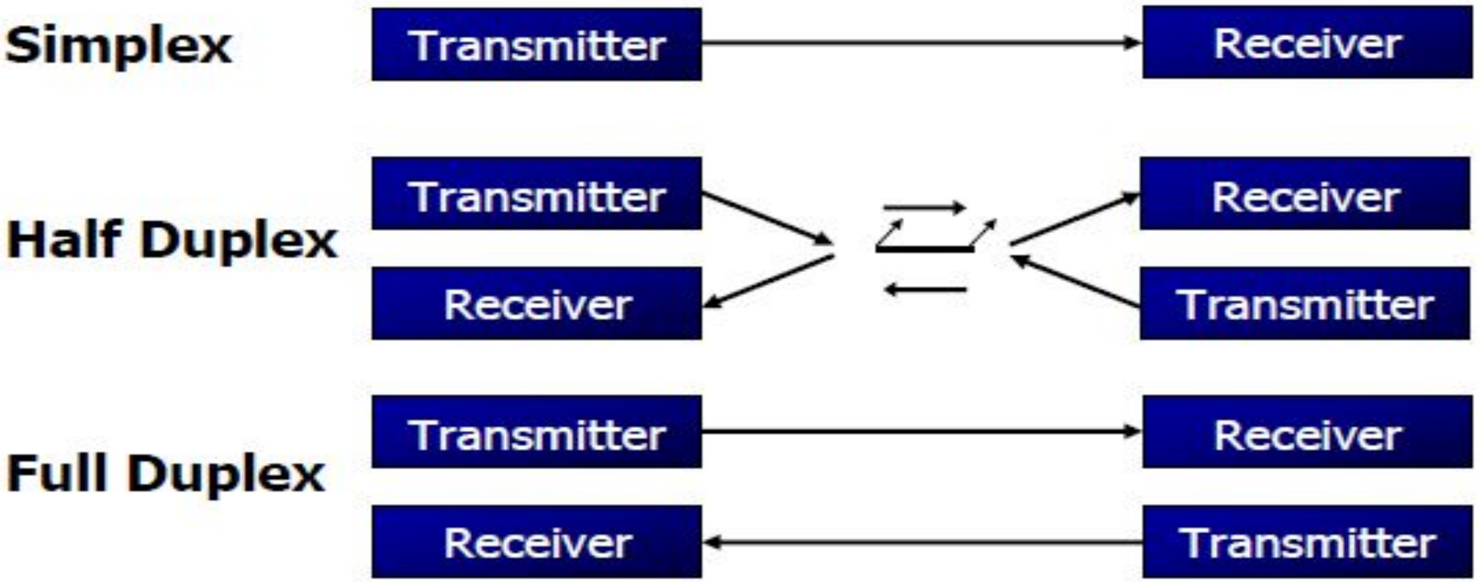
Parallel Transfer



- At the transmitting end, the byte of data must be converted to serial bits using parallel-in-serial-out shift register
- At the receiving end, there is a serial-in-parallel-out shift register to receive the serial data and pack them into byte
- When the distance is short, the digital signal can be transferred as it is on a simple wire and requires no modulation
- If data is to be transferred on the telephone line, it must be converted from 0s and 1s to audio tones
    - This conversion is performed by a device called a *modem*, "Modulator/demodulator"

- Serial data communication uses two methods
  - *Synchronous* method transfers a block of data at a time
  - *Asynchronous* method transfers a single byte at a time
- It is possible to write software to use either of these methods, but the programs can be tedious and long
  - There are special IC chips made by many manufacturers for serial communications
    - UART (universal asynchronous Receiver-transmitter)
    - USART (universal synchronous-asynchronous Receiver-transmitter)

- If data can be transmitted and received, it is a *duplex* transmission
  - If data transmitted one way a time, it is referred to as *half duplex*
  - If data can go both ways at a time, it is *full duplex*
- This is contrast to *simplex* transmission

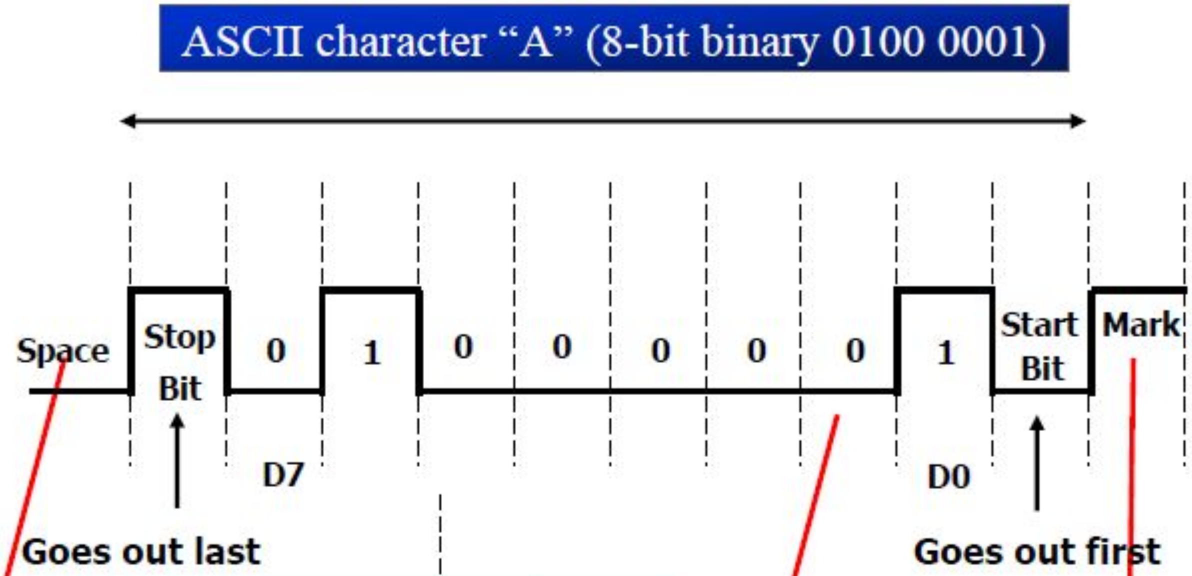


- A *protocol* is a set of rules agreed by both the sender and receiver on
  - How the data is packed
  - How many bits constitute a character
  - When the data begins and ends
- Asynchronous serial data communication is widely used for character-oriented transmissions
  - Each character is placed in between start and stop bits, this is called *framing*
  - Block-oriented data transfers use the synchronous method
- The start bit is always one bit, but the stop bit can be one or two bits

BASICS OF SERIAL COMMUNICA-TION

Start and Stop Bits (cont')

- The start bit is always a 0 (low) and the stop bit(s) is 1 (high)



The 0 (low) is referred to as *space*

The transmission begins with a start bit followed by D0, the LSB, then the rest of the bits until MSB (D7), and finally, the one stop bit indicating the end of the character

When there is no transfer, the signal is 1 (high), which is referred to as *mark*

BASICS OF SERIAL COMMUNICATION

Data Transfer Rate

- The rate of data transfer in serial data communication is stated in *bps* (bits per second)
- Another widely used terminology for bps is *baud rate*
  - It is modem terminology and is defined as the number of signal changes per second
  - In modems, there are occasions when a single change of signal transfers several bits of data
- As far as the conductor wire is concerned, the baud rate and bps are the same, and we use the terms interchangeably

BASICS OF SERIAL COMMUNICATION

RS232 Standards

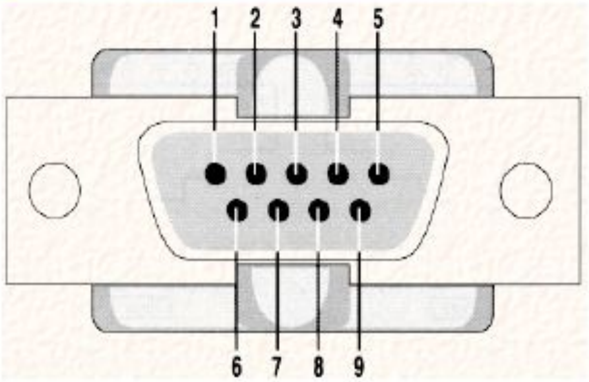
- An interfacing standard RS232 was set by the Electronics Industries Association (EIA) in 1960
- The standard was set long before the advent of the TTL logic family, its input and output voltage levels are not TTL compatible
  - In RS232, a 1 is represented by -3 ~ -25 V, while a 0 bit is +3 ~ +25 V, making -3 to +3 undefined

BASICS OF SERIAL COMMUNICA-TION

RS232 Standards (cont')

- Since not all pins are used in PC cables, IBM introduced the DB-9 version of the serial I/O standard

RS232 Connector DB-9



RS232 DB-9 Pins

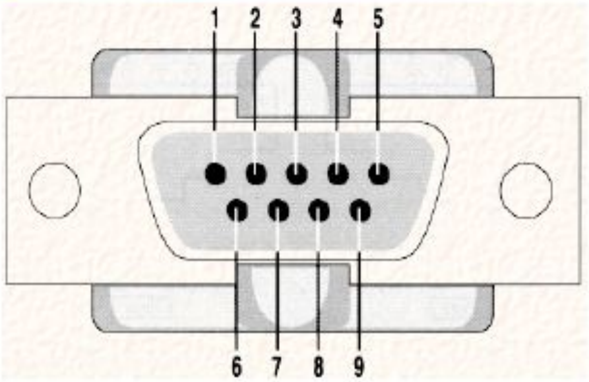
Pin	Description
1	Data carrier detect (-DCD)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready (-DSR)
7	Request to send (-RTS)
8	Clear to send (-CTS)
9	Ring indicator (RI)

BASICS OF SERIAL COMMUNICATION

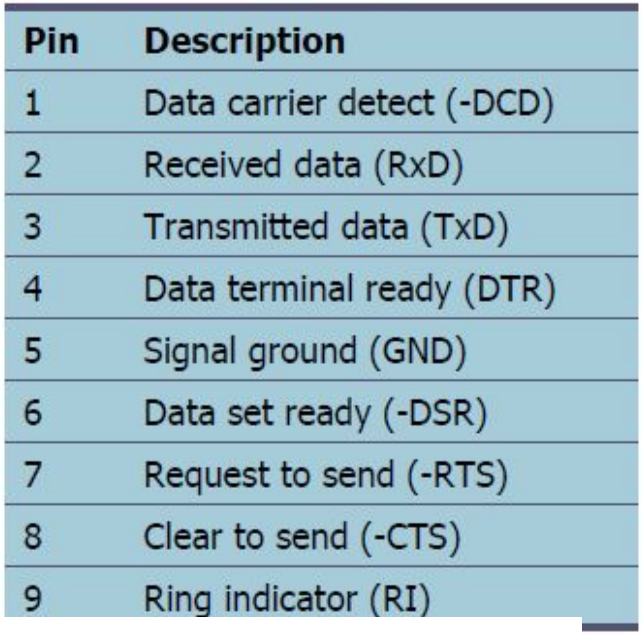
RS232 Standards (cont')

- Since not all pins are used in PC cables, IBM introduced the DB-9 version of the serial I/O standard

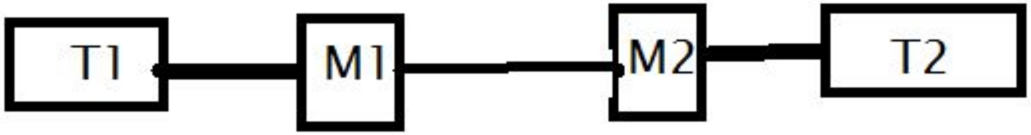
RS232 Connector DB-9



RS232 DB-9 Pins



Pin	Description
1	Data carrier detect (-DCD)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready (-DSR)
7	Request to send (-RTS)
8	Clear to send (-CTS)
9	Ring indicator (RI)

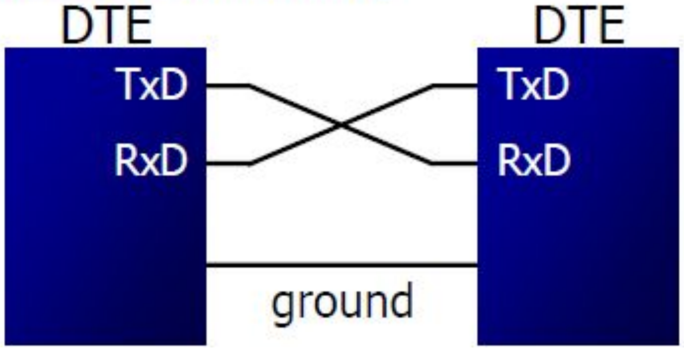


BASICS OF SERIAL COMMUNICATION

Data Communication Classification

- Current terminology classifies data communication equipment as
  - DTE (data terminal equipment) refers to terminal and computers that send and receive data
  - DCE (data communication equipment) refers to communication equipment, such as modems
- The simplest connection between a PC and microcontroller requires a minimum of three pins, TxD, RxD, and ground

Null modem connection



BASICS OF SERIAL COMMUNICATION

RS232 Pins

- DTR (data terminal ready)
  - When terminal is turned on, it sends out signal DTR to indicate that it is ready for communication
- DSR (data set ready)
  - When DCE is turned on and has gone through the self-test, it assert DSR to indicate that it is ready to communicate
- RTS (request to send)
  - When the DTE device has byte to transmit, it assert RTS to signal the modem that it has a byte of data to transmit
- CTS (clear to send)
  - When the modem has room for storing the data it is to receive, it sends out signal CTS to DTE to indicate that it can receive the data now

BASICS OF SERIAL COMMUNICATION

RS232 Pins (cont')

- DCD (data carrier detect)
  - The modem asserts signal DCD to inform the DTE that a valid carrier has been detected and that contact between it and the other modem is established
- RI (ring indicator)
  - An output from the modem and an input to a PC indicates that the telephone is ringing
  - It goes on and off in synchronous with the ringing sound

CONNECTION TO RS232

- A line driver such as the MAX232 chip is required to convert RS232 voltage levels to TTL levels, and vice versa
- two pins that are used specifically for transferring and receiving data serially
  - These two pins are called TxD and RxD
  - These pins are TTL compatible; therefore, they require a line driver to make them RS232 compatible

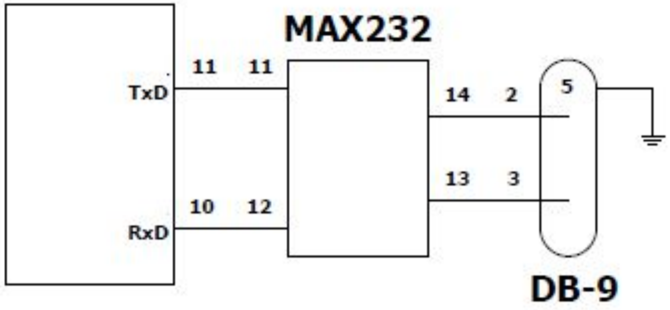
8251

- The 8251A is a programmable serial communication interface chip designed for synchronous and asynchronous serial data communication.
- It supports the serial transmission of data.
- It is packed in a 28 pin DIP.

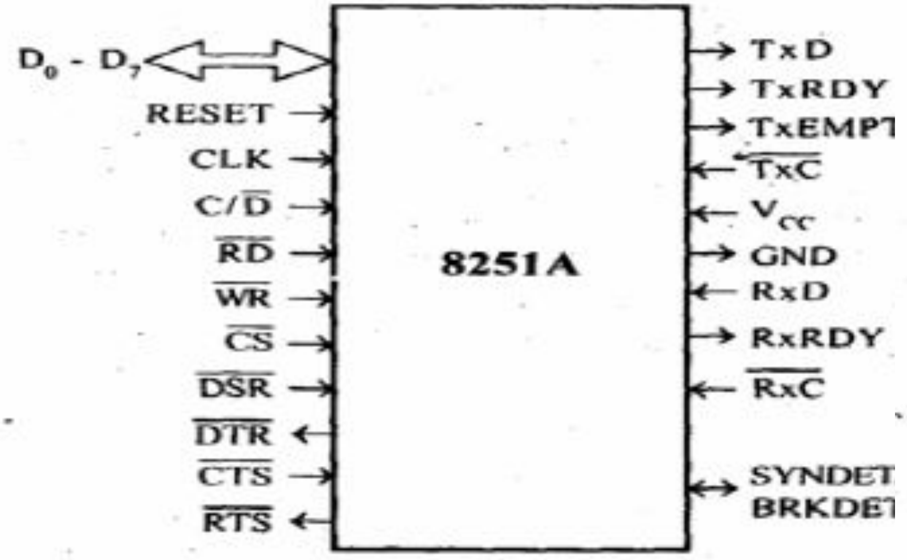
CONNECTION TO RS232

MAX232

- We need a line driver (voltage converter) to convert the R232's signals to TTL voltage levels that will be acceptable to TxD and RxD pins

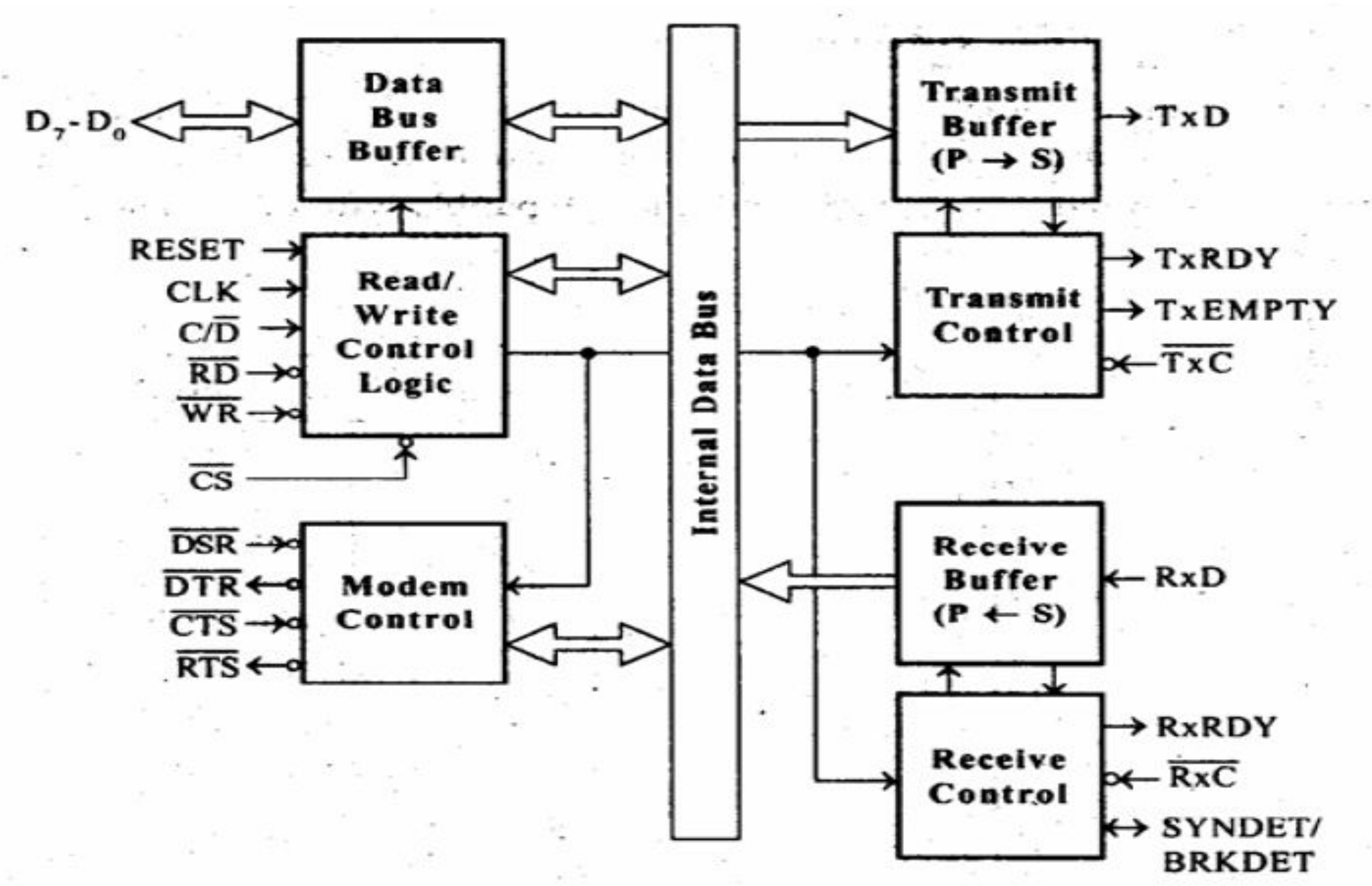


Pin details



Pin	Description
D_0-D_7	Parallel data
C/$\overline{D}$	Control register or Data buffer select
$\overline{RD}$	Read control
$\overline{WR}$	Write control
$\overline{CS}$	Chip Select
CLK	Clock pulse (TTL)
RESET	Reset
$\overline{TxC}$	Transmitter Clock
TxD	Transmitter Data
$\overline{RxC}$	Receiver Clock
RxD	Receiver Data
RxRDY	Receiver Ready
TxRDY	Transmitter Ready
$\overline{DSR}$	Data Set Ready
$\overline{DTR}$	Data Terminal Ready
SYNDET/ BRKDET	Synchronous Detect / Break Detect
$\overline{RTS}$	Request To Send Data
$\overline{CTS}$	Clear To Send Data
TxEMPTY	Transmitter Empty
V_{cc}	Supply (+5V)
GND	Ground (0 V)

Architecture



Arch - details

- The functional block diagram of 825 1A consists five sections.

They are:

- Read/Write control logic
- Transmitter
- Receiver
- Data bus buffer
- Modem control.

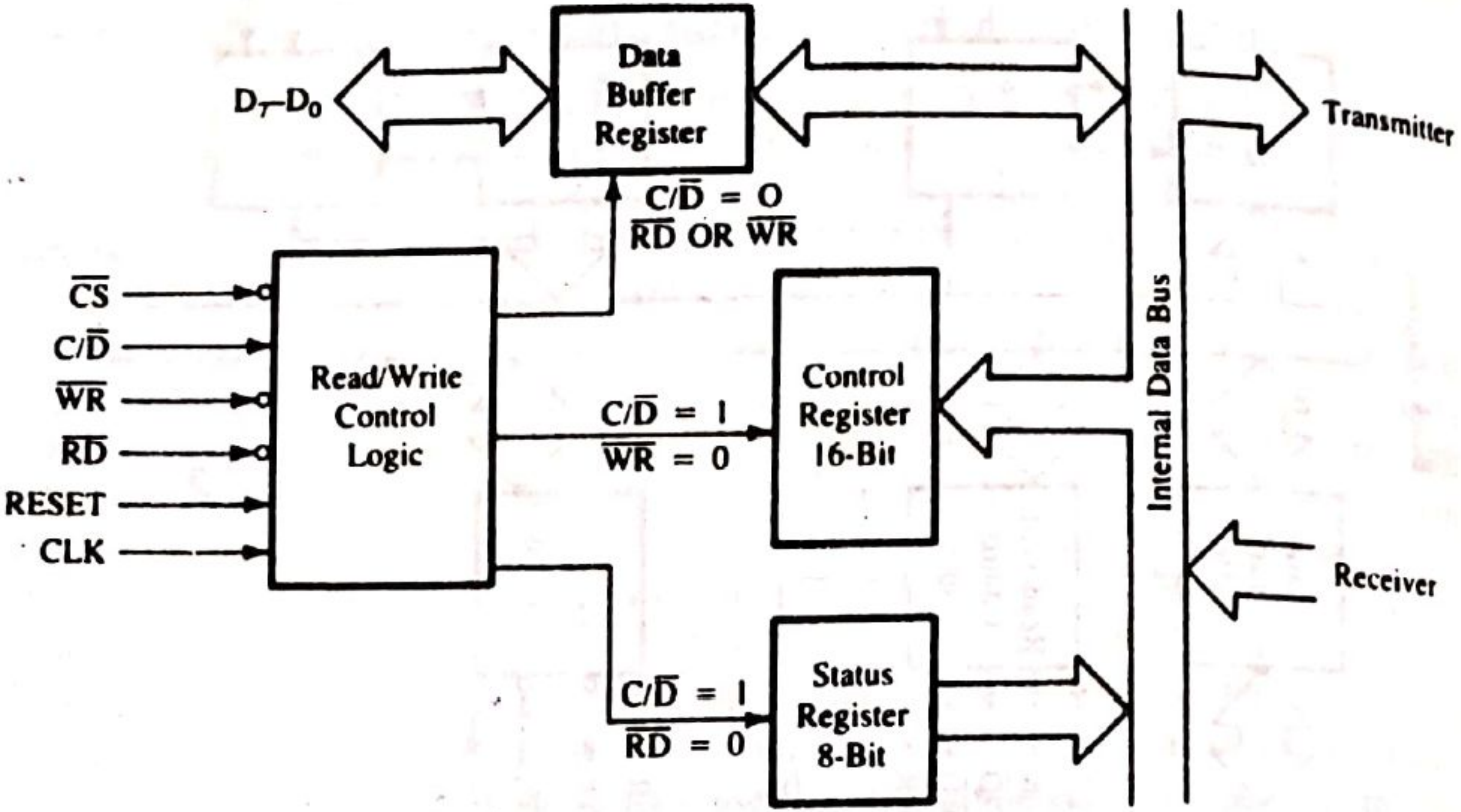
Read/Write control logic

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

Read/Write control logic

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

Control Logic and Registers



Summary of Control Signals for the 8251A

CS'	C/D'	RD'	WR'	Function
0	1	1	0	MPU writes instructions in the control registers
0	1	0	1	MPU reads status from the status registers
0	0	1	0	MPU outputs data to the Data Buffer
0	0	0	1	MPU accepts data from the Data Buffer
1	X	X	X	USART is not selected

The 8251A: Expanded Block Diagram of Control Logic and Registers

Control Register This 16-bit register for a control word consists of two independent bytes: the first byte is called the **mode instruction** (word) and the second byte is called the **command instruction** (word). This register can be accessed as an output port when the C/$\overline{D}$ pin is high.

Status Register This input register checks the ready status of a peripheral. This register is addressed as an input port when the C/$\overline{D}$ pin is high; it has the same port address as the control register.

Data Buffer This bidirectional register can be addressed as an input port and an output port when the C/$\overline{D}$ pin is low. Table 16.4 summarizes all the interfacing and control signals.

TRANSMITTER SECTION

The transmitter accepts parallel data from the MPU and converts them into serial data. It has two registers: a buffer register to hold eight bits and an output register to convert eight bits into a stream of serial bits (Figure 16.14). The MPU writes a byte in the buffer register; whenever the output register is empty, the contents of the buffer register are transferred to the output register. This section transmits data on the TxD pin with the appropriate framing bits (Start and Stop). Three output signals and one input signal are associated with the transmitter section.

□ **TxD—Transmit Data:** Serial bits are transmitted on this line.

□ **TxC—Transmitter Clock:** This input signal controls the rate at which bits are transmitted by the USART. The clock frequency can be 1, 16, or 64 times the baud.

□ **TxRDY—Transmitter Ready:** This is an output signal. When it is high, it indicates that the buffer register is empty and the USART is ready to accept a byte. It can be used either to interrupt the MPU or to indicate the status. This signal is reset when a data byte is loaded into the buffer.

□ **TxE—Transmitter Empty:** This is an output signal. Logic 1 on this line indicates that the output register is empty. This signal is reset when a byte is transferred from the buffer to the output registers.

RECEIVER SECTION

The receiver accepts serial data on the RxD line from a peripheral and converts them into parallel data. The section has two registers: the receiver input register and the buffer register (Figure 16.14).

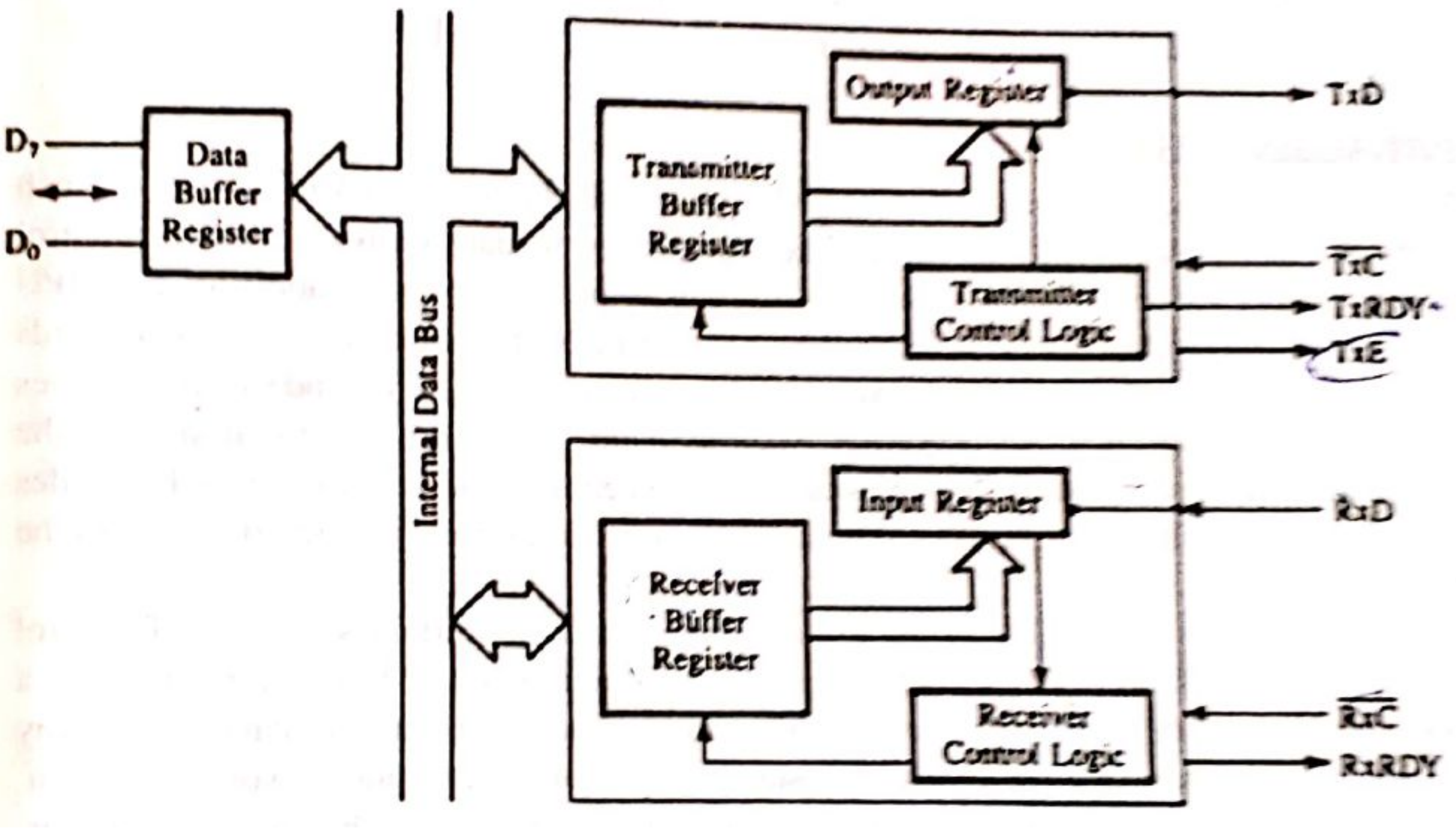
When the RxD line goes low, the control logic assumes it is a Start bit, waits for half a bit time, and samples the line again. If the line is still low, the input register accepts the following bits, forms a character, and loads it into the buffer register. Subsequently, the parallel byte is transferred to the MPU when requested. In the asynchronous mode, two input signals and one output signal are necessary, as described below.

□ **RxD—Receive Data:** Bits are received serially on this line and converted into a parallel byte in the receiver input register.

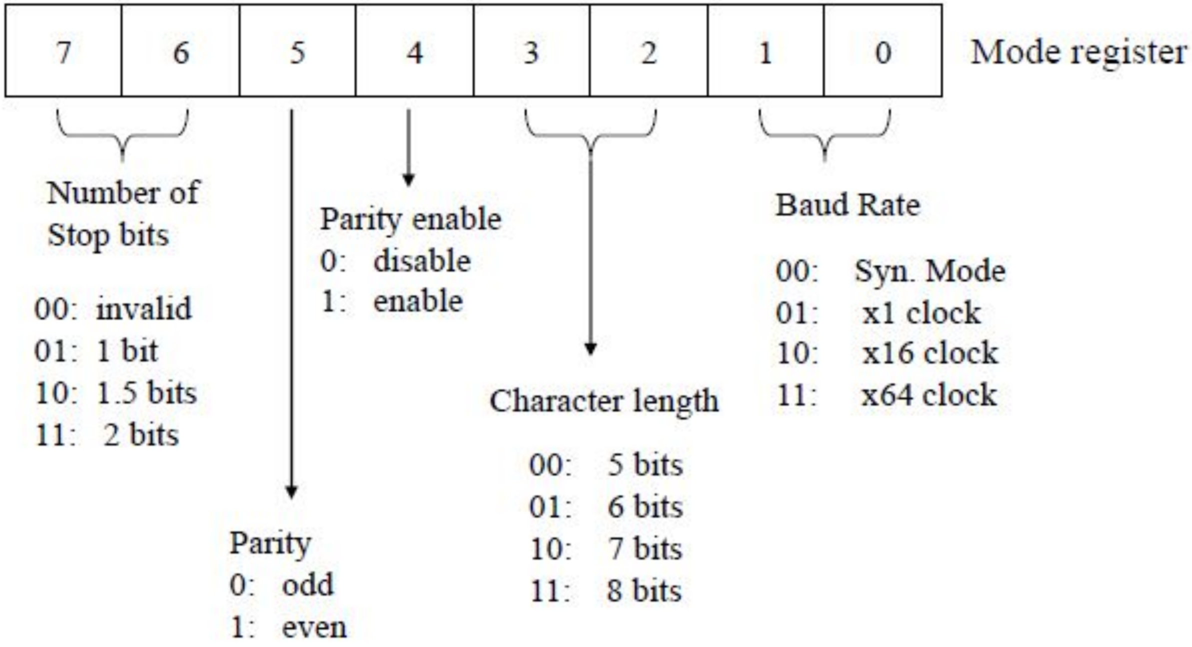
□ **RxC—Receiver Clock:** This is a clock signal that controls the rate at which bits are received by the USART. In the asynchronous mode, the clock can be set to 1, 16, or 64 times the baud.

□ **RxRDY—Receiver Ready:** This is an output signal. It goes high when the USART has a character in the buffer register and is ready to transfer it to the MPU. This line can be used either to indicate the status or to interrupt the MPU.

Transmitter Section and Receive Section



8251 mode register



7	6	5	4	3	2	1	0

Mode register

Number of
Stop bits

00: invalid
01: 1 bit
10: 1.5 bits
11: 2 bits

Parity enable
0: disable
1: enable

Parity
0: odd
1: even

Character length

00: 5 bits
01: 6 bits
10: 7 bits
11: 8 bits

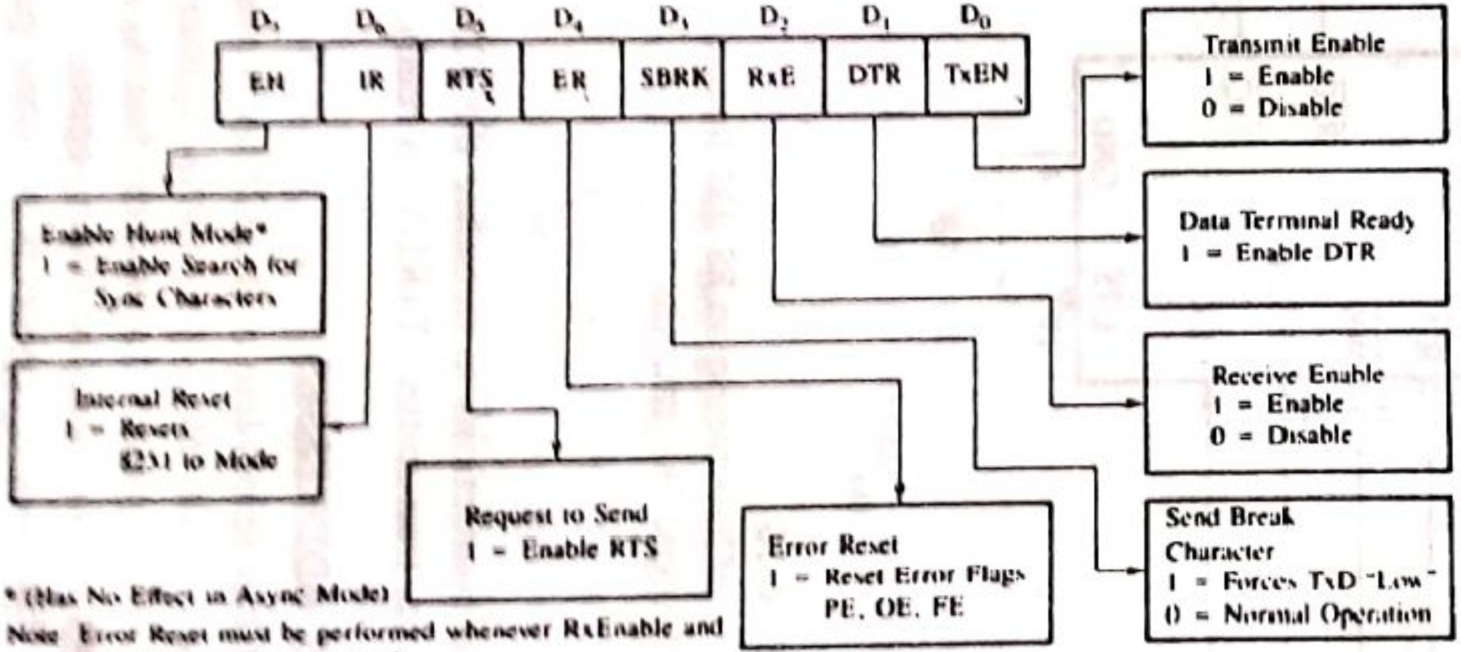
Baud Rate

00: Syn. Mode
01: x1 clock
10: x16 clock
11: x64 clock

8251 command register

EH	IR	RTS	ER	SBRK	RxE	DTR	TxE

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



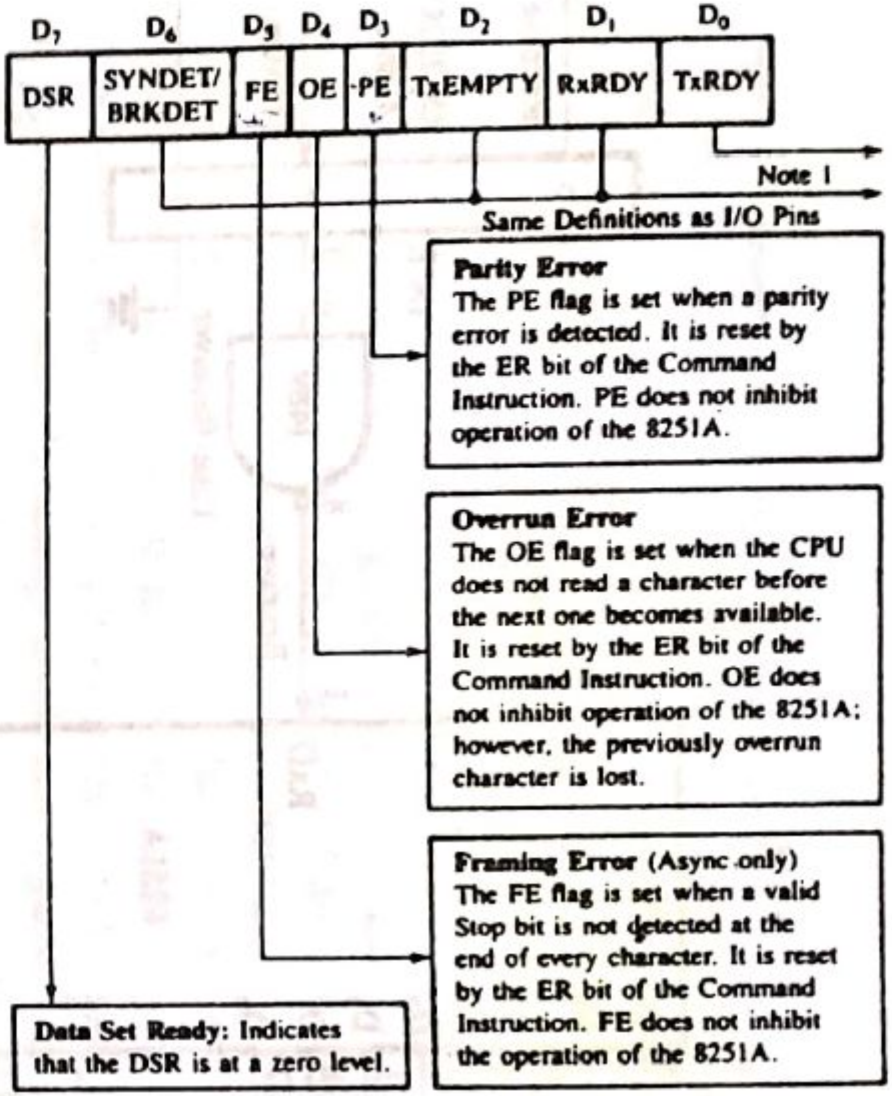
* (Has No Effect in Async Mode)

Note: Error Reset must be performed whenever RxEnable and Enter Hunt are programmed.

8251 status register

DSR	SYNDET	FE	OE	PE	TxEMPTY	RxRDY	TxRDY

TxRDY:	transmit ready
RxRDY:	receiver ready
TxEMPTY:	transmitter empty
PE:	parity error
OE:	overrun error
FE:	framing error
SYNDET:	sync. character detected
DSR:	data set ready



Parity Error
The PE flag is set when a parity error is detected. It is reset by the ER bit of the Command Instruction. PE does not inhibit operation of the 8251A.

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 the 8251A; however, the previously overrun character is lost.

Framing Error (Async only)
The FE flag is set when a valid Stop bit is not detected at the end of every character. It is reset by the ER bit of the Command Instruction. FE does not inhibit the operation of the 8251A.

Data Set Ready: Indicates that the DSR is at a zero level.