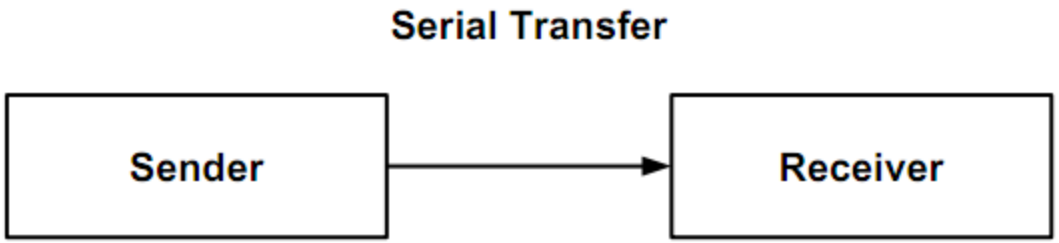
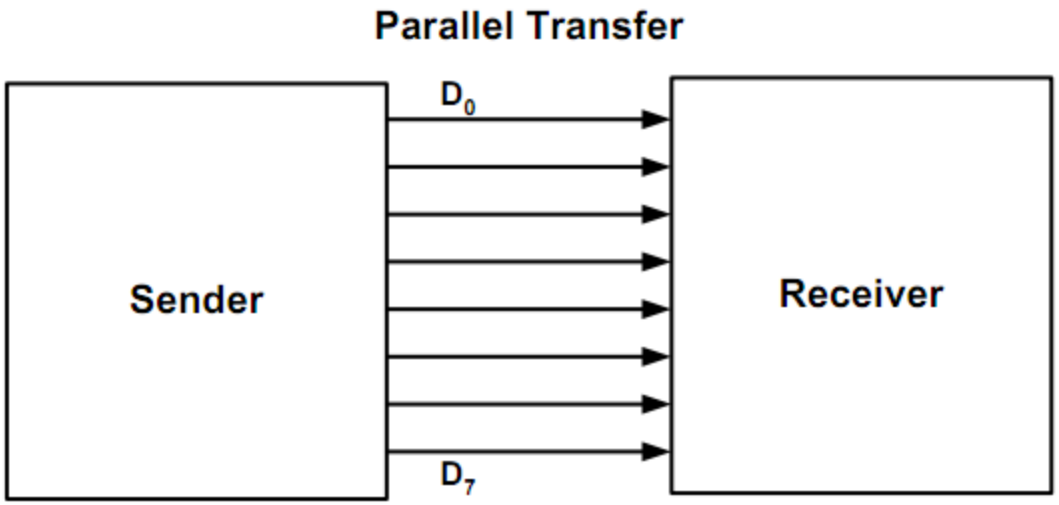
Programmable Communication Interface – PCI

Serial Vs Parallel Data Transfer



Serial communication uses a single line data.

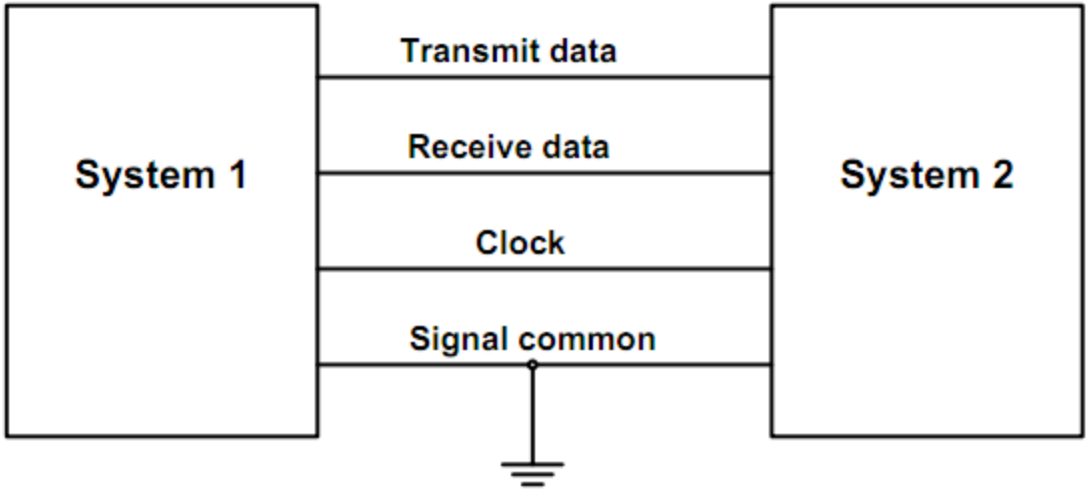


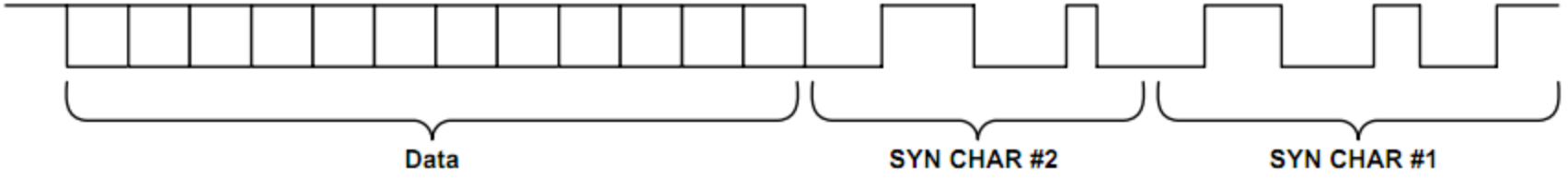
Parallel communication uses n-bit data line.

Synchronous Vs Asynchronous

- **Asynchronous** transfer does not require clock signal.
- However, it transfers **extra bits**(start bits and stop bits) during data communication.
- **Synchronous** does not transfer extra bits. However, it requires clock signal.

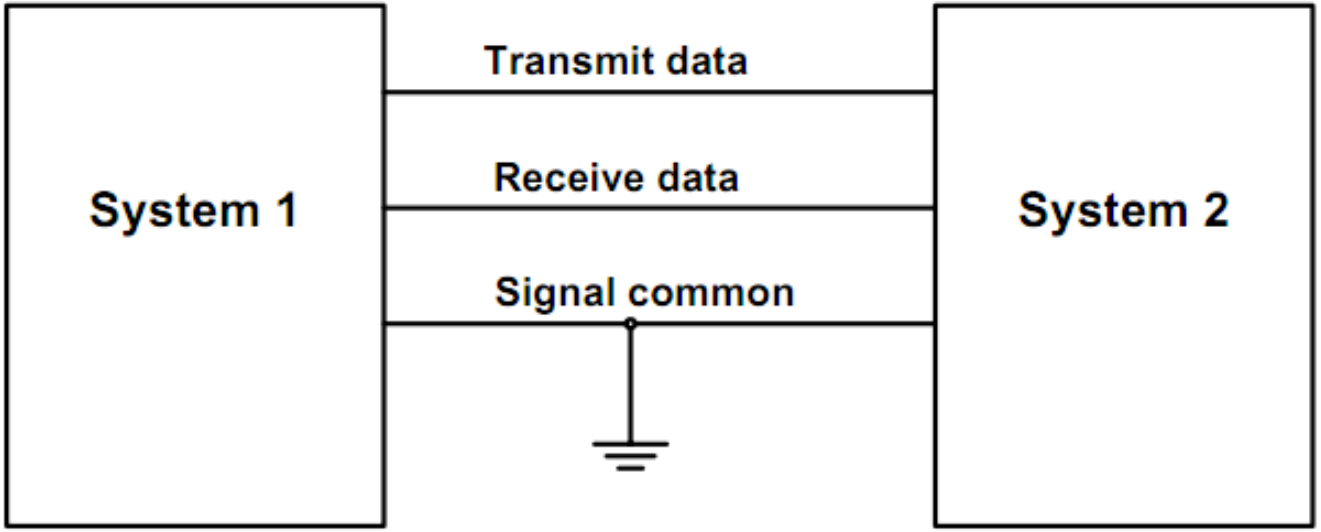
Synchronous Data Communication

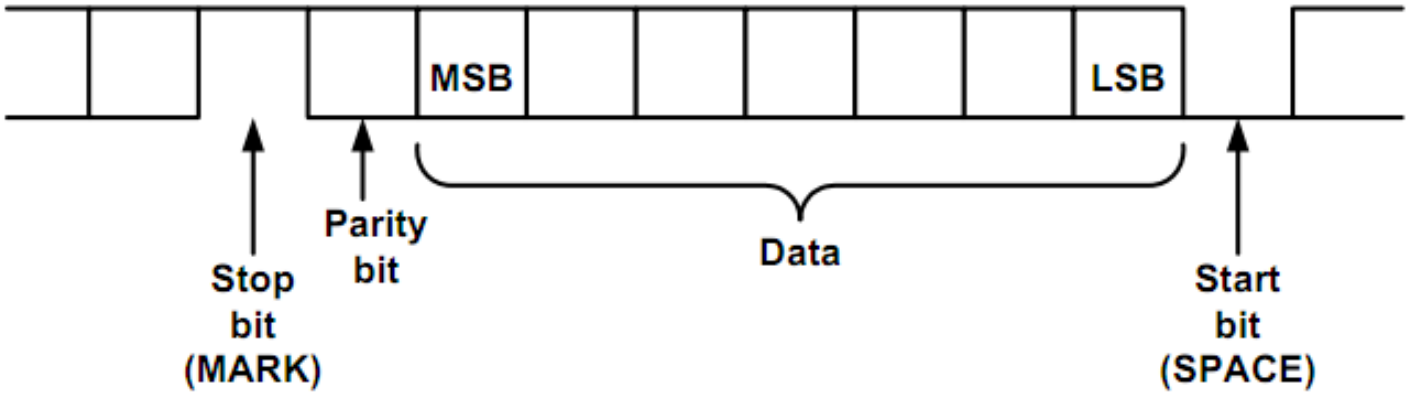




Synchronous data-transmission format

Asynchronous Data Communication



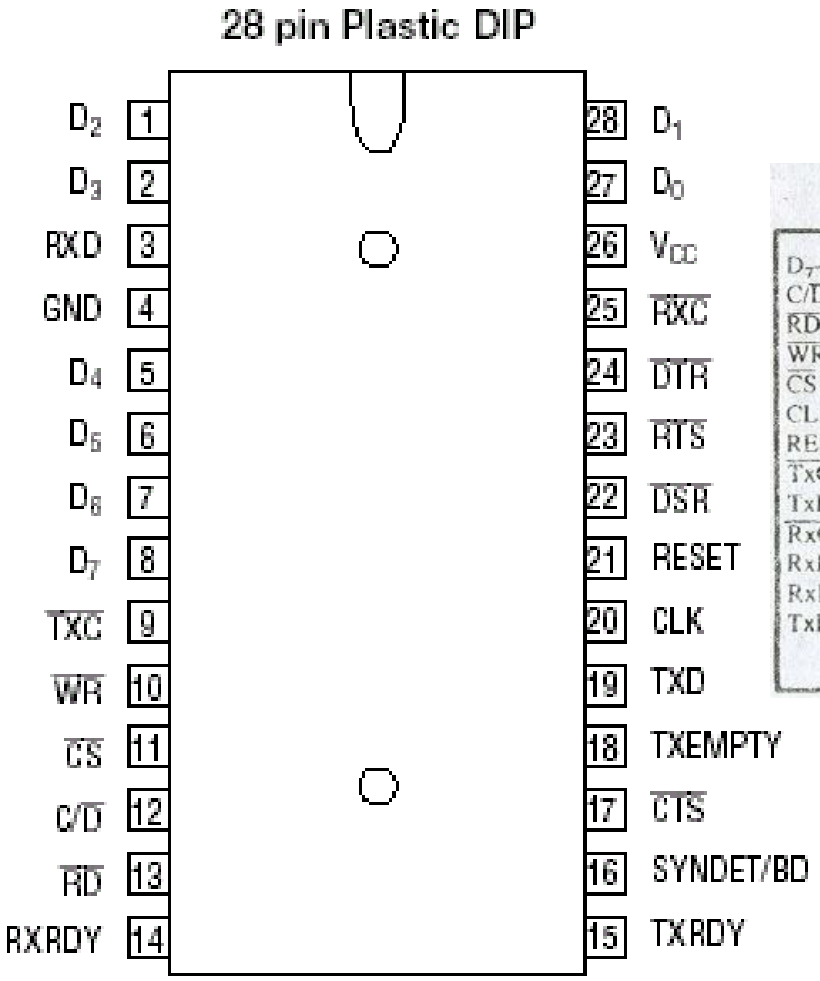


Asynchronous data-transmission format

8251 USART

- The 8251 USART (**Universal Synchronous Asynchronous Receiver Transmitter**) is capable of implementing either an asynchronous or synchronous serial data communication.
- As a peripheral device of a microcomputer system, the **8251 receives parallel data from the CPU and transmits serial data after conversion.**
- This device also **receives serial data from the outside and transmits parallel data to the CPU after conversion.**

8251 Pin Diagram

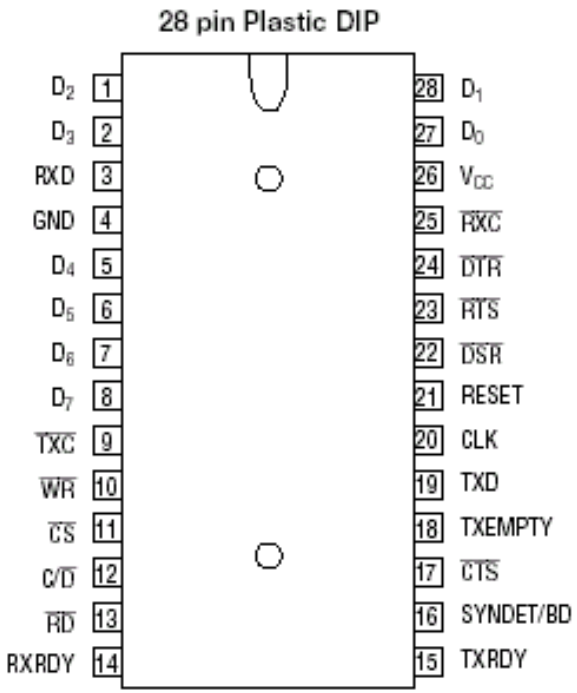


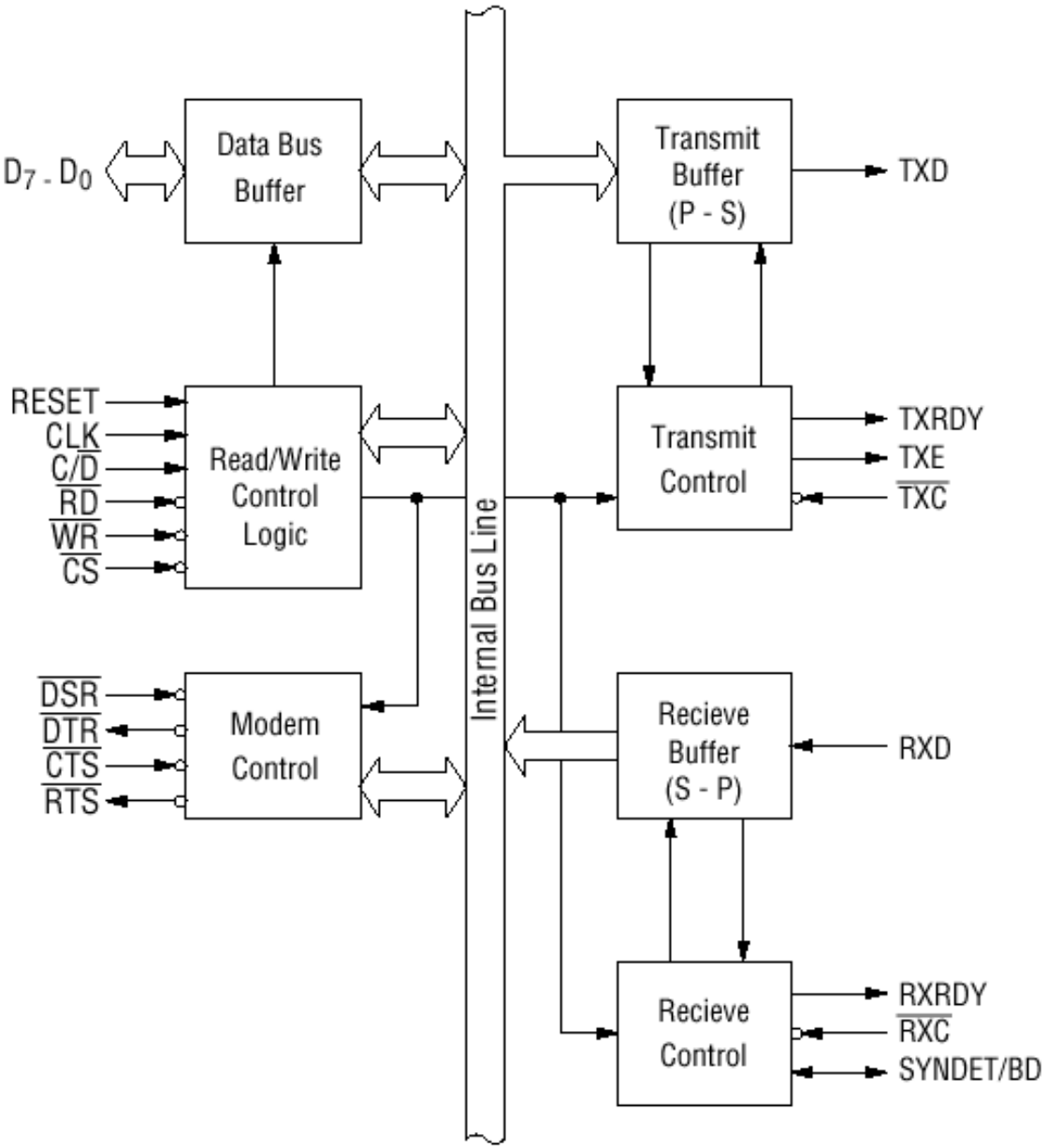
Pin Names

D_7–D_0	Data Bus (8 bits)
C/D	Control or Data is to be Written or Read
$\overline{RD}$	Read Data Command
$\overline{WR}$	Write Data or Control Command
$\overline{CS}$	Chip Enable
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/BD	Sync Detect/ Break Detect
$\overline{RTS}$	Request to Send Data
$\overline{CTS}$	Clear to Send Data
TxE	Transmitter Empty
V_{CC}	+5 Volt Supply
GND	Ground

8251 Block Diagram





Pin Description

- **D0 - D7 -** This is **bidirectional data bus** which receive control words and transmits data from the CPU and sends status words and received data to CPU.

- **RESET -** A "High" on this input forces the 8251 into **"reset status".** The min. reset width is six clock inputs during the operating status of CLK.

- **CLK -** CLK signal is used to generate **internal device timing**. CLK signal is independent of RXC or TXC.

- **WR -** This is the "active low" input terminal which receives a signal for **writing transmit data and control words** from the CPU into the 8251.

Pin Description

- **RD -** This is the "active low" input terminal which receives a signal for **reading receive data and status words** from the 8251.

- **C/D -** This is an input terminal which receives a signal for **selecting data or command words and status words** when the 8251 is accessed by the CPU.

- **CS -** This is the "active low" input terminal which **selects the 8251** at low level when the CPU accesses.

Pin Description

- **TXD -** This is an output terminal for **transmitting data** from which serial-converted data is sent out.

- **TXRDY -** This is an output terminal which indicates that the 8251 is **ready to accept a transmitted data character**.

- **TXEMPTY -** This is an output terminal which indicates that the 8251 **has transmitted all the characters** and had no data character.

- **TXC -** This is a clock input signal (Active Low) which determines the **transfer speed of transmitted data**.
  - *In "synchronous 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 the TXC.*

Pin Description

- **RXD -** This is a terminal which **receives serial data**.

- **RXRDY - ** This is a terminal which indicates that the 8251 contains a character that is **ready to READ**.
  - If the CPU reads a data character, RXRDY will be reset by the leading edge of RD signal. Unless the CPU reads a data character before the next one is received completely, the preceding data will be lost. In such a case, an overrun error flag status word will be set.

- **RXC -** 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.

Pin Description

- **SYNDET/BD -** This is a terminal whose function changes according to mode.
  - **In “internal synchronous mode“,** this terminal is at high level, if sync characters are received and synchronized.
  - If a status word is read, the terminal will be reset.
  - **In “external synchronous mode”,** this is an input terminal. A "High" on this input forces the 8251 to start receiving data characters.
  - **In “asynchronous mode”,** this is an output terminal which generates "high level“ output upon the detection of a "break" character if receiver data contains a "low-level" space between the stop bits of two continuous characters.
  - The terminal will be reset, if RXD is at high level. After Reset is active, the terminal will be output at low level.

Pin Description

- **DSR -** This is an input port for MODEM interface. The input status of the terminal can be recognized by the CPU reading status words.

- **DTR -** This is an output port for MODEM interface. It is possible to set the status of DTR by a command.

- **CTS -** This is an input terminal for MODEM interface which is used for controlling a transmit circuit.
  - The terminal controls data transmission if the device is set in "TX Enable" status by a command. Data is transmittable if the terminal is at low level.

- **RTS -** This is an output port for MODEM interface. It is possible to set the status RTS by a command.

8251 functional configuration

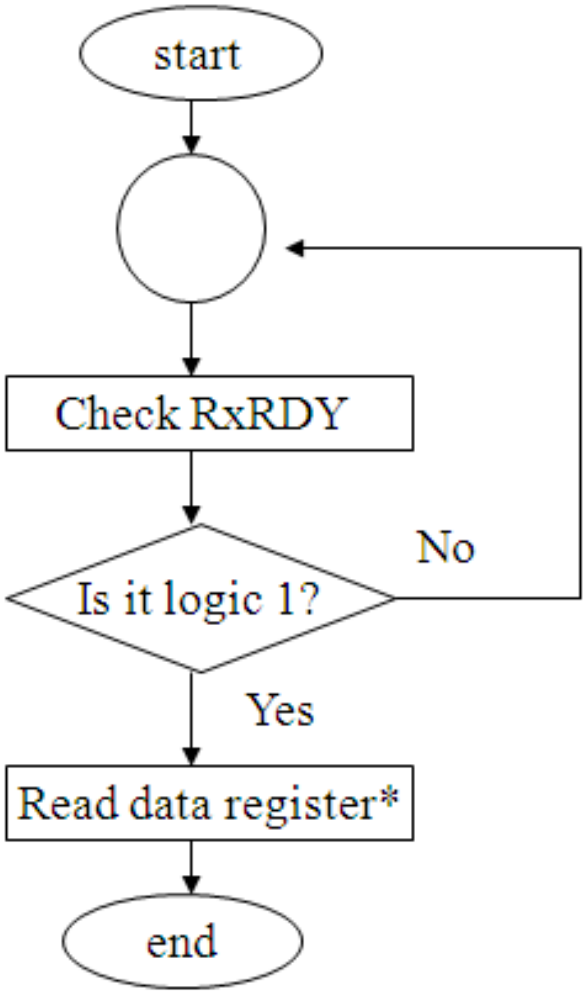
- The 8251 functional configuration is programmed by software.
- Operation between the 8251 and a CPU is executed by program control.
- Table 1 shows the operation between a CPU and the device.

$\overline{CS}$	C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	
1	×	×	×	Data Bus 3-State
0	×	1	1	Data Bus 3-State
0	1	0	1	Status → CPU
0	1	1	0	Control Word ← CPU
0	0	0	1	Data → CPU
0	0	1	0	Data ← CPU

Table 1 Operation between a CPU and 8251

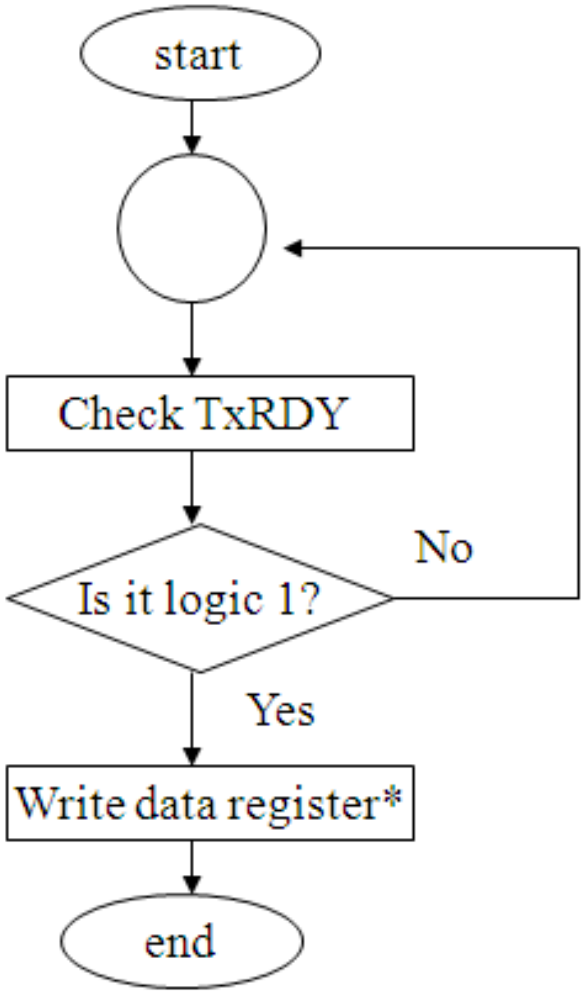
Simple Serial I/O Procedures

❑ Read



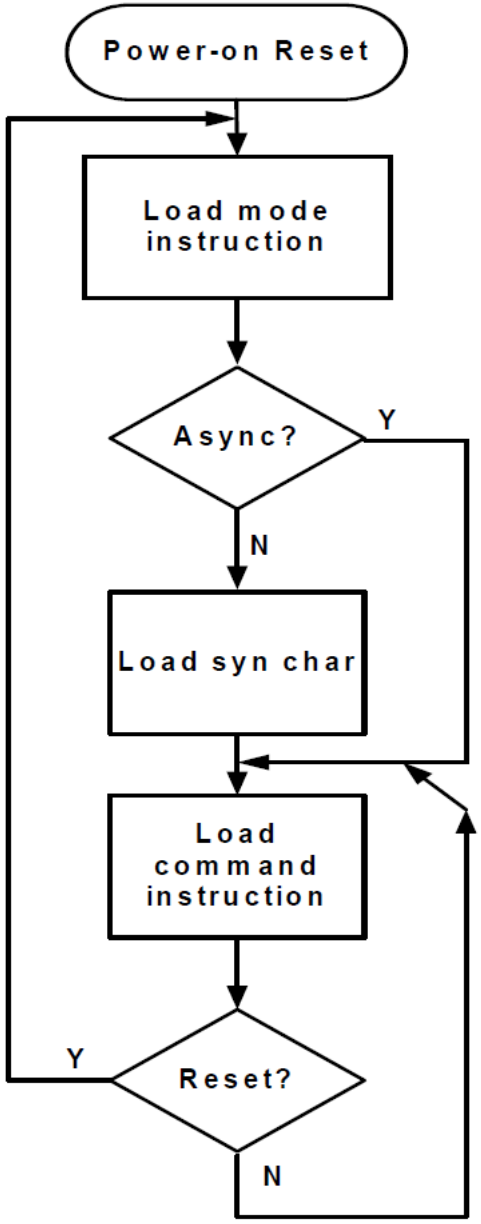
* This clears RxRDY

❑ Write



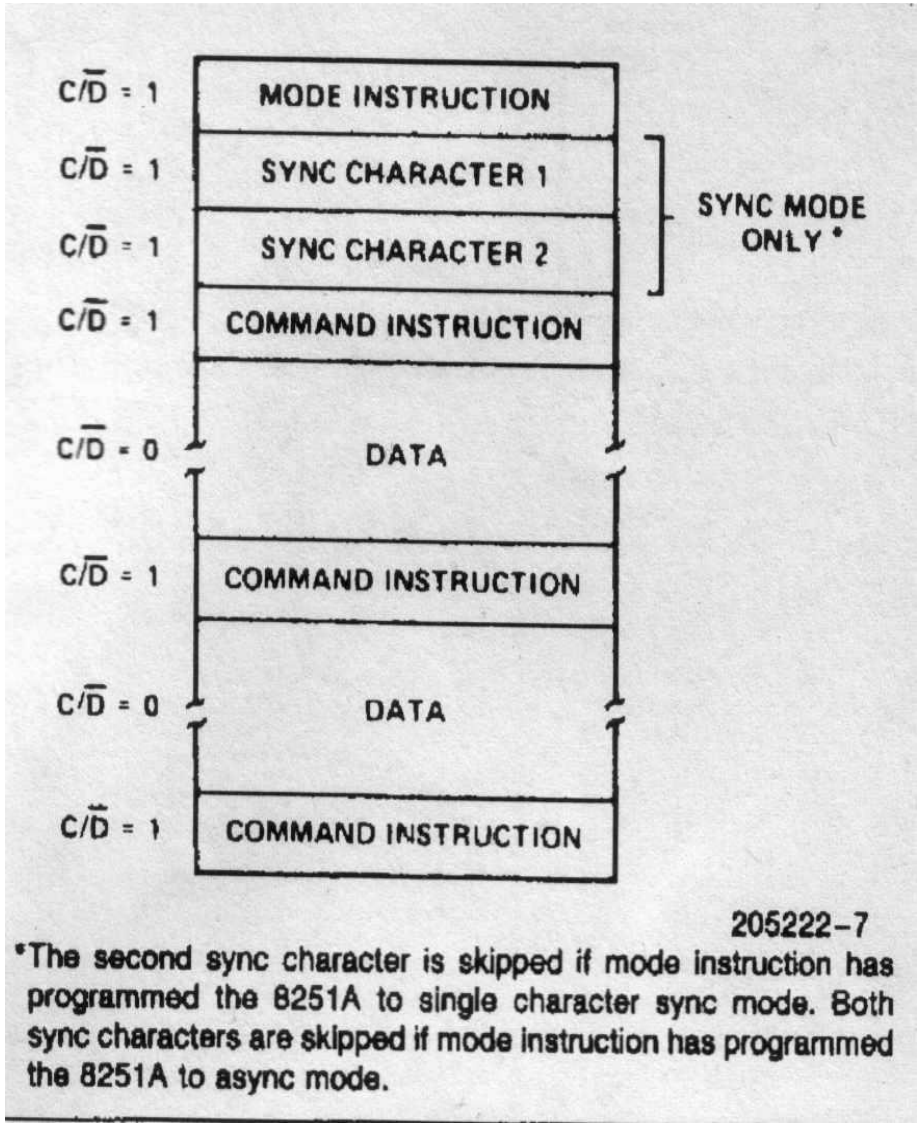
* This clears TxRDY

8251 Initialization



- Before the 8251 can be used to receiver or transmit characters, its **mode control and command registers** must be initialized.
- The 8251 has only one address for a few control registers.
- The only readable register is a status register. The other registers must be written in sequence.

8251 Initialization



*The second sync character is skipped if mode instruction has programmed the 8251A to single character sync mode. Both sync characters are skipped if mode instruction has programmed the 8251A to async mode.

8251 Control Words

- There are two types of control word.
    1. **Mode instruction (setting of function)**
    2. **Command (setting of operation)**

- Apart from the control words, a Status Word is present in 8251 to see the internal status.

1. Mode instruction word

- Mode instruction is used for **setting the function** of the 8251.
- Mode instruction will be in "wait for write" at either internal reset or external reset.
- That is, the writing of a control word **after resetting** will be recognized as a "mode instruction.“
- Items set by mode instruction are as follows:
  - ***Synchronous/asynchronous mode***
  - ***Stop bit length (asynchronous mode)***
  - ***Character length***
  - ***Parity bit***
  - ***Baud rate factor (asynchronous mode)***
  - ***Internal/external synchronization (synchronous mode)***
  - ***Number of synchronous characters (Synchronous mode)***

Mode Instruction - Asynchronous

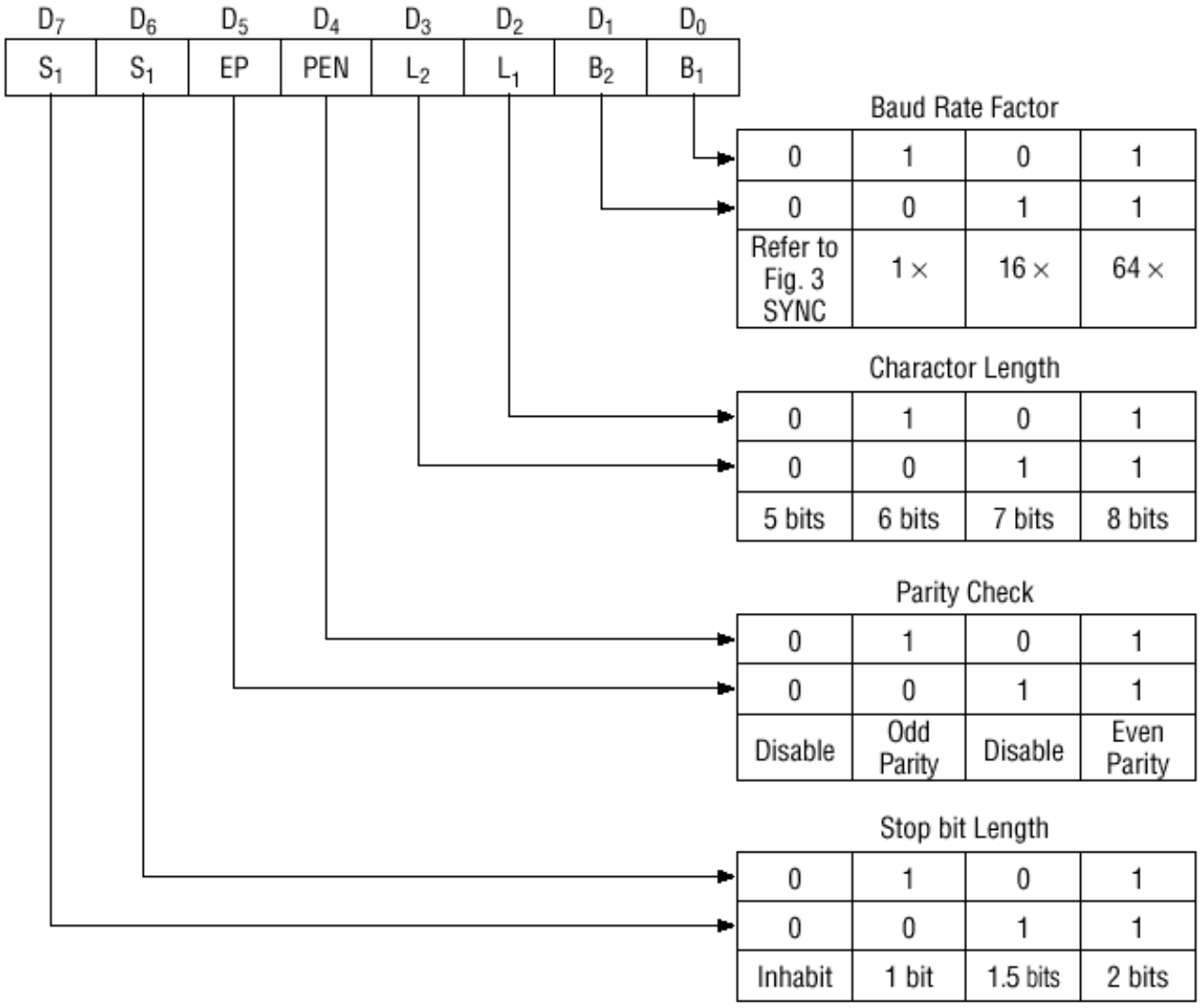


Fig. 2 Bit Configuration of Mode Instruction (Asynchronous)

Mode Instruction - Synchronous

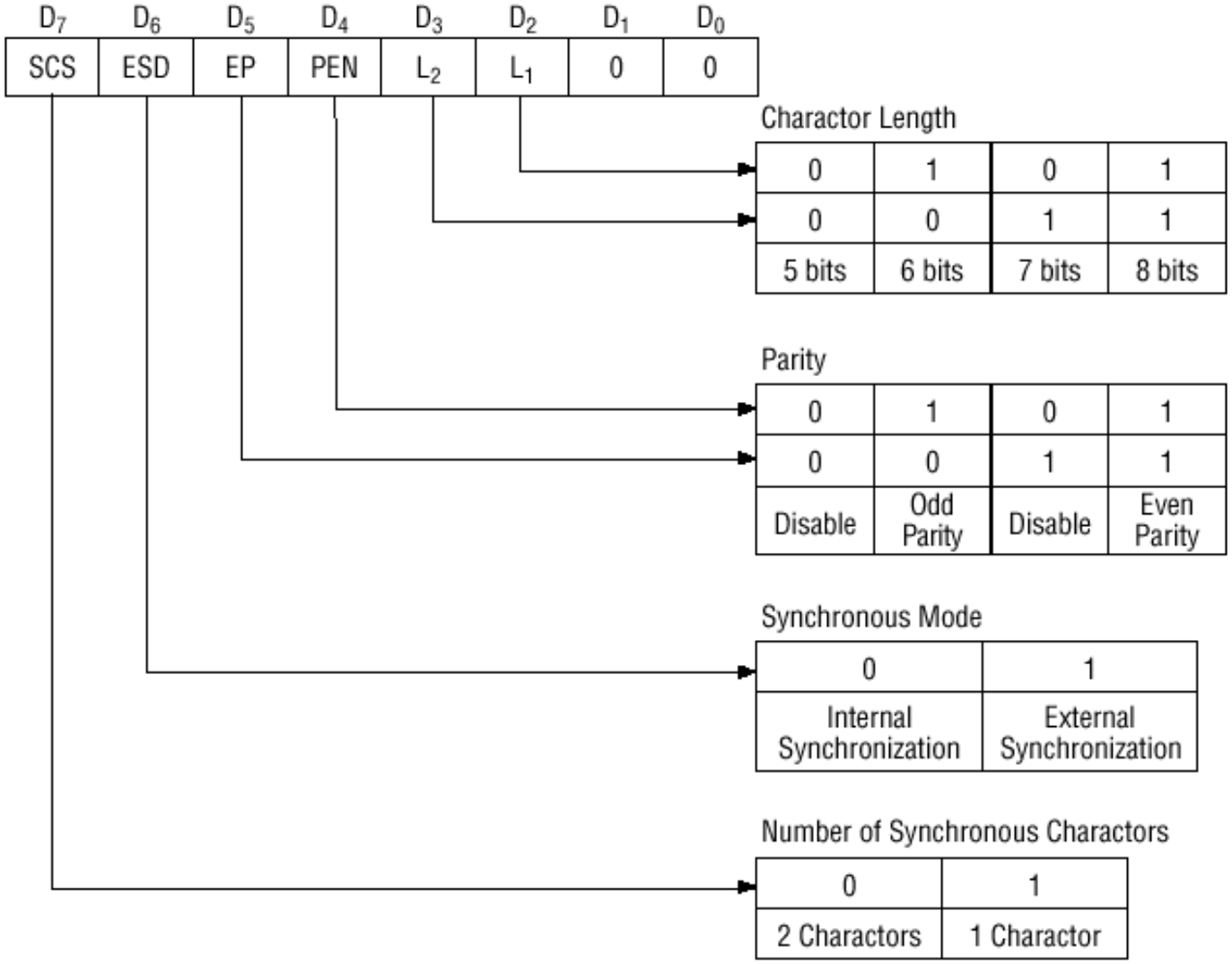


Fig. 3 Bit Configuration of Mode Instruction (Synchronous)

2. Command Word

- Command is used for setting the **operation of the 8251**.
- It is possible to write a command **whenever necessary** after writing a mode instruction and sync characters.
- Items to be set by command are as follows:
  - *Transmit Enable/Disable*
  - *Receive Enable/Disable*
  - *DTR, RTS Output of data.*
  - *Resetting of error flag.*
  - *Sending to break characters*
  - *Internal resetting*
  - *Hunt mode (synchronous mode)*

Bit Configurable Command Word Format

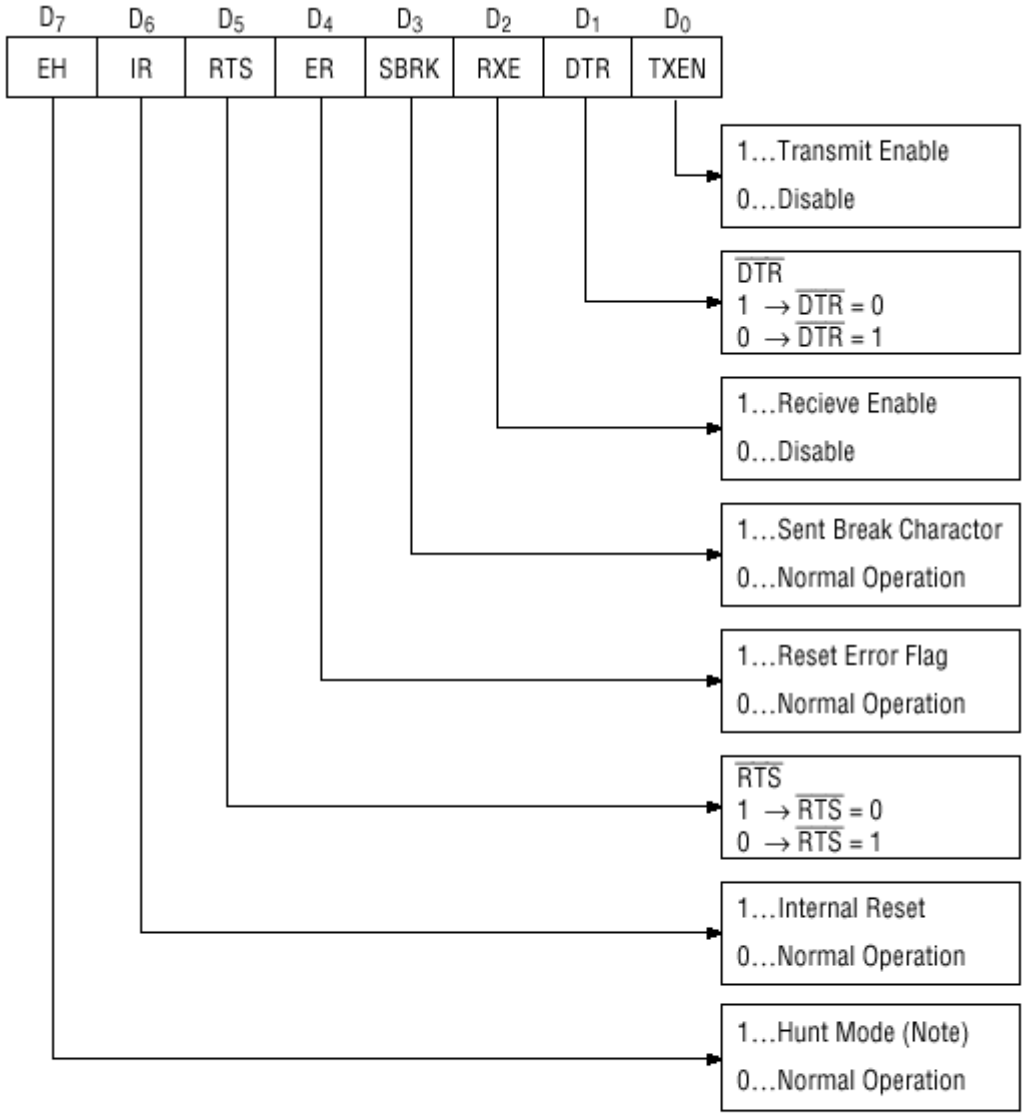


Fig. 4 Bit Configuration of Command

Status Word Format

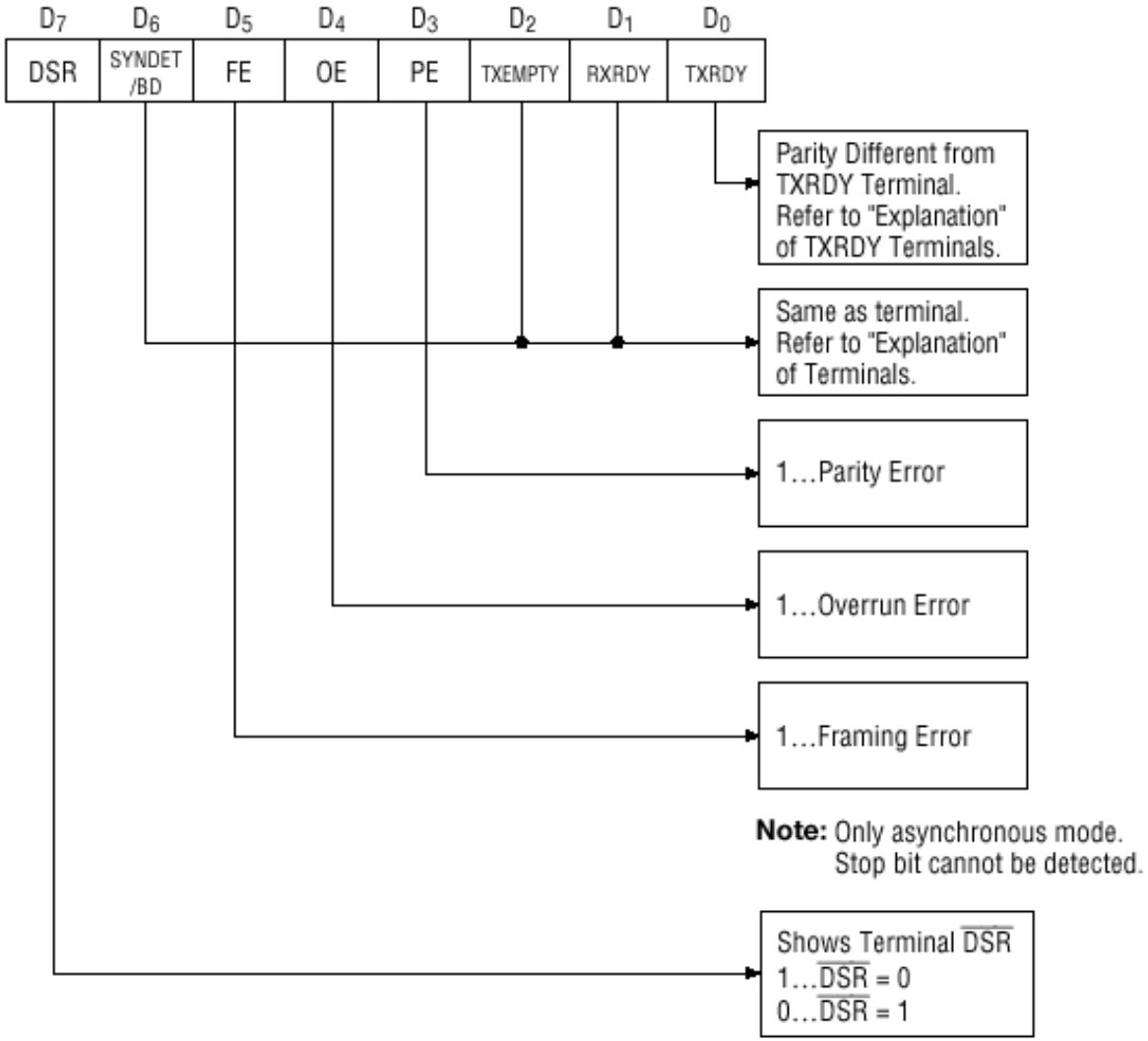
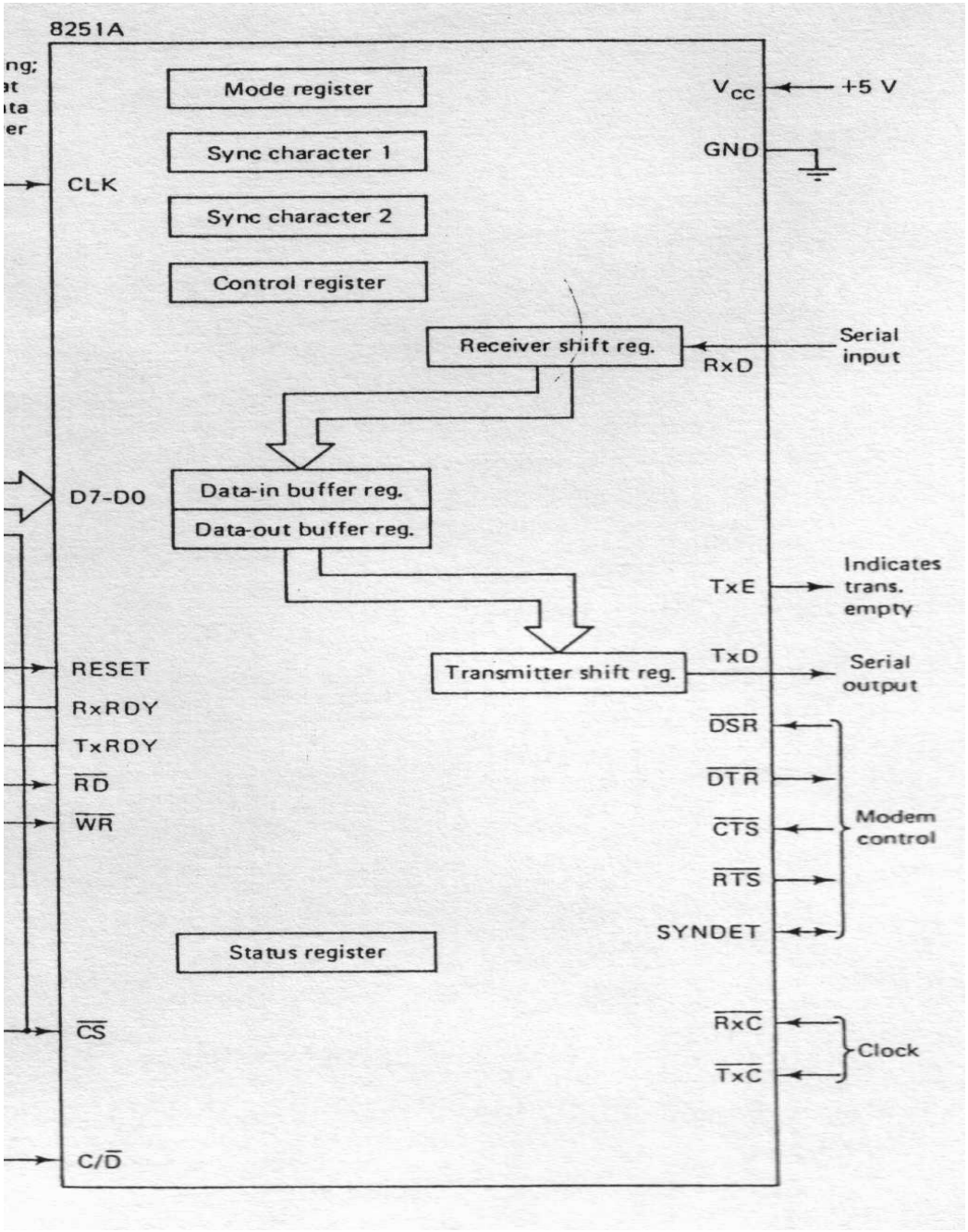


Fig. 5 Bit Configuration of Status Word

8251 Internal Diagram



8251 Initialization program

```
; 8086 instructions to initialize the 8251A on an
; SDK-86 board

   MOV  DX, 0FFF2H ; point at command register address
   MOV  AL, 00H    ; send 0's to guarantee device is
   OUT  DX, AL     ; in the command instruction format
   MOV  CX, 2      ; before the RESET command is
D0:LOOP D0         ; issued and delay after sending
   OUT  DX, AL     ; each command instruction
   MOV  CX, 2
D1:LOOP D1
   OUT  DX, AL
   MOV  CX, 2
D2:LOOP D2
   MOV  AL, 40H    ; Sent internal reset command to
   OUT  DX, AL     ; return device to idle state
   MOV  CX, 2      ; Load delay constant
D3:LOOP D3         ; and delay
   MOV  AL,11001110B; Load mode control word & send it
   OUT  DX, AL
```

```
; 1 1 0 0 1 1 1 0    Mode Word
; \ \ \ \ \ \ \_\___baud rate factor of 16x
;  \ \ \ \ \_\_____character length of 8 bits
;   \ \ \_\ _______parity disabled
;    \_\___________2 stop bits

   MOV  CX, 2      ; and delay
D4:LOOP D4
   MOV  AL,00110111B ; Load command word and send it
   OUT  DX, AL
; 0 0 1 1 0 1 1 1    Command word
; \ \ \ \ \ \ \ \____Transmit enable
; \ \ \ \ \ \ \_____Data terminal ready, DTR will
;  \ \ \ \ \ \       output 0
;   \ \ \ \ \ \_____Receive enable
;    \ \ \ \ \______Normal operation
;     \ \ \ \_______Reset all error flags
;      \ \ \________RST output 0, request to send
;       \ \_________Do not return to mode
;        \          instruction form
;         \_________Disable hunt mode
```

Programming Examples

```
; Instructions for transmitting data using an
; SDK-86 8251A using polling method

  MOV DX, 0FFF2H        ; Point at control register
TEST1:                  ; address
  IN  AL, DX            ; Read status
  AND AL, 10000001B     ; and check status of
;           \_____\___data set ready & transmit ready
  CMP AL, 10000001B     ; Is it ready?
  JNE TEST1             ; Continue to poll if not ready
  MOV DX, 0FFF0H        ; otherwise point at data address
  MOV AL, DATA_TO_SEND  ; Load data to send
  OUT DX, AL            ; and send it
```

(a)

Programming Examples

```
; Instructions for receiving data with an
; SDK-86 8251A using polling method

  MOV DX, 0FFF2H      ; Point at control register
TEST2:                ; address
  IN  AL, DX          ; Read status
  AND AL, 00000010B   ; and check status of RxRdy
  JZ  TEST2           ; Continue to poll if not ready
  MOV DX, 0FFF0H      ; otherwise point at data
  IN  AL, DX          ; address and get data
```