



USART 8251A

CEN433

King Saud University

Dr. Mohammed Amer Arafah

Standard Interchange Codes

1. American Standards Committee for Information Interchange (ASCII):

Bit Positions				7	0	0	0	0	1	1	1	1
				6	0	0	1	1	0	0	1	1
				5	0	1	0	1	0	1	0	1
4	3	2	1									
0	0	0	0	NUL	DLE	SP	0	@	P	\	p	
0	0	0	1	SOH	DC1	!	1	A	Q	a	q	
0	0	1	0	STX	DC2	“	2	B	R	b	r	
0	0	1	1	ETX	DC3	#	3	C	S	c	s	
0	1	0	0	EOT	DC4	\$	4	D	T	d	t	
0	1	0	1	ENQ	NAK	%	5	E	U	e	u	
0	1	1	0	ACK	SYN	&	6	F	V	f	v	
0	1	1	1	BEL	ETB	'	7	G	W	g	w	
1	0	0	0	BS	CAN	(8	H	X	h	x	
1	0	0	1	HT	EM)	9	I	Y	i	y	
1	0	1	0	LF	SUB	*	:	J	Z	j	z	
1	0	1	1	VT	ESC	+	;	K	[k	{	
1	1	0	0	FF	FS	,	<	L	\	l		
1	1	0	1	CR	GS	-	=	M]	m	}	
1	1	1	0	SO	RS	.	>	N	^	n	~	
1	1	1	1	SI	US	/	?	O	_	o	DEL	

Standard Interchange Codes

1. American Standards Committee for Information Interchange (ASCII)

Bit Positions				7	0	0	0	0	1	1	1	1
				6	0	0	1	1	0	0	1	1
				5	0	1	0	1	0	1	0	1
4	3	2	1									
0	0	0	0	NUL	DLE	SP	0	@	P	\	p	
0	0	0	1	SOH	DC1	!	1	A	Q	a	q	
0	0	1	0	STX	DC2	”	2	B	R	b	r	
0	0	1	1	ETX	DC3	#	3	C	S	c	s	
0	1	0	0	EOT	DC4	\$	4	D	T	d	t	
0	1	0	1	ENQ	NAK	%	5	E	U	e	u	
0	1	1	0	ACK	SYN	&	6	F	V	f	v	
0	1	1	1	BEL	ETB	'	7	G	W	g	w	
1	0	0	0	BS	CAN	(8	H	X	h	x	
1	0	0	1	HT	EM)	9	I	Y	i	y	
1	0	1	0	LF	SUB	*	:	J	Z	j	z	
1	0	1	1	VT	ESC	+	;	K	[k	{	
1	1	0	0	FF	FS	,	<	L	\	l		
1	1	0	1	CR	GS	-	=	M]	m	}	
1	1	1	0	SO	RS	.	>	N	^	n	~	
1	1	1	1	SI	US	/	?	O	_	o	DEL	

Standard Interchange Codes

2. Extended Binary Coded Decimal Interchange Code (EBCDIC):

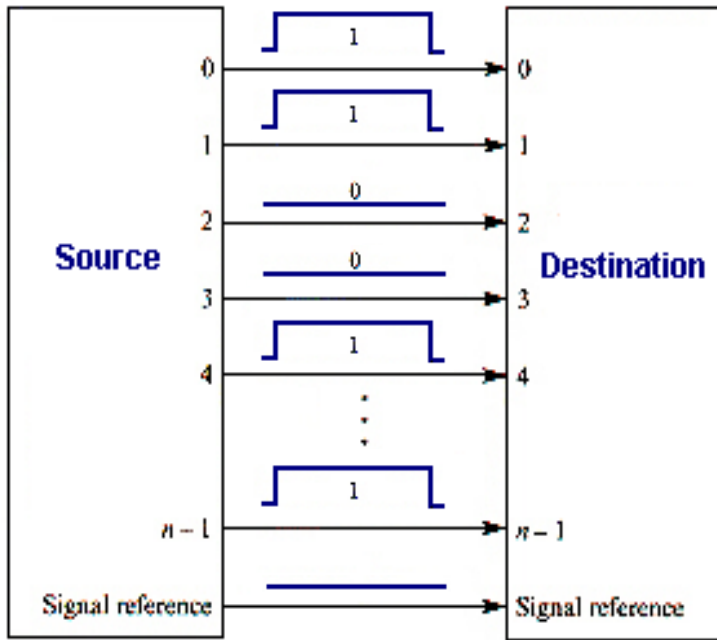
Bit Positions				8	7	6	5														
				0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1		
				0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
4	3	2	1																		
0	0	0	0	NUL	DLE	DS		SP	&	-								0			
0	0	0	1	SOH	DC1	SOS			/		a	j			A	J		1			
0	0	1	0	STX	DC2	FS	SYN				b	k	s		B	K	S	2			
0	0	1	1	ETX	DC3						c	l	t		C	L	T	3			
0	1	0	0	PF	RES	BYP	PN				d	m	u		D	M	U	4			
0	1	0	1	HT	NL	LF	RS				e	n	v		E	N	V	5			
0	1	1	0	LC	BS	EOB	UC				f	o	w		F	O	W	6			
0	1	1	1	DEL	IL	PRE	EOT				g	p	x		G	P	X	7			
1	0	0	0		CAN						h	q	y		H	Q	Y	8			
1	0	0	1		EM						i	r	z		I	N	Z	9			
1	0	1	0	SMM	CC	SM		¢	!	:											
1	0	1	1	VT				.	\$	'	#										
1	1	0	0	FF	IFS		DC4	<	*	%	@										
1	1	0	1	CR	IGS	ENQ	NAK	()	-	,										
1	1	1	0	SO	IRS	ACK		+	;	>	=										
1	1	1	1	SI	IUS	BEL	SUB		¬	?	”							□			

Standard Interchange Codes

2. Extended Binary Coded Decimal Interchange Code (EBCDIC):

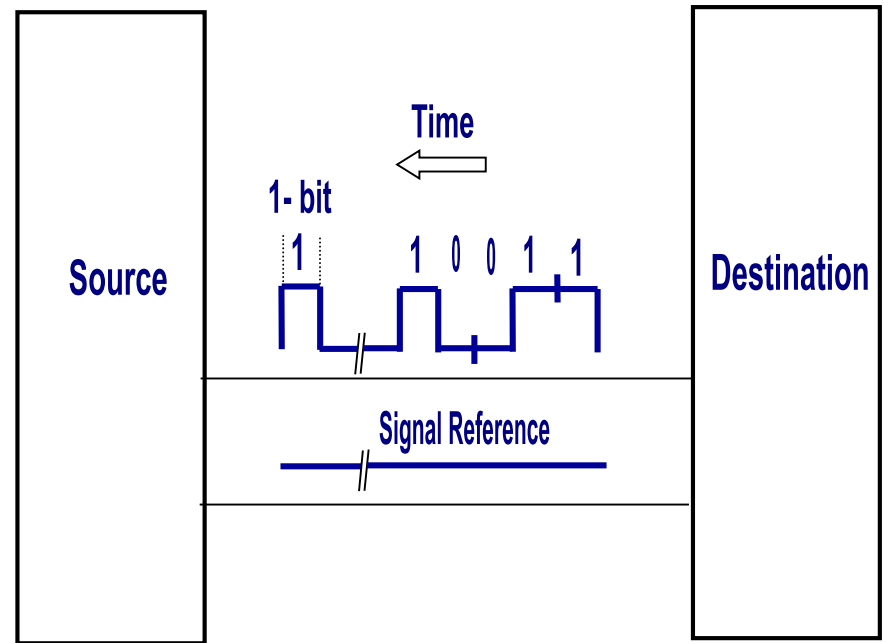
Bit Positions				8	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
				7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1
				6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1
				5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1
4	3	2	1																		
0	0	0	0	NUL	DLE	DS		SP	&	-								0			
0	0	0	1	SOH	DC1	SOS			/		a	j			A	J		1			
0	0	1	0	STX	DC2	FS	SYN				b	k	s		B	K	S	2			
0	0	1	1	ETX	DC3						c	l	t		C	L	T	3			
0	1	0	0	PF	RES	BYP	PN				d	m	u		D	M	U	4			
0	1	0	1	HT	NL	LF	RS				e	n	v		E	N	V	5			
0	1	1	0	LC	BS	EOB	UC				f	o	w		F	O	W	6			
0	1	1	1	DEL	IL	PRE	EOT				g	p	x		G	P	X	7			
1	0	0	0		CAN						h	q	y		H	Q	Y	8			
1	0	0	1		EM						i	r	z		I	N	Z	9			
1	0	1	0	SMM	CC	SM		¢	!	:											
1	0	1	1	VT				.	\$	'	#										
1	1	0	0	FF	IFS		DC4	<	*	%	@										
1	1	0	1	CR	IGS	ENQ	NAK	()	-	,										
1	1	1	0	SO	IRS	ACK		+	;	>	=										
1	1	1	1	SI	IUS	BEL	SUB		¬	?	”							□			

Parallel/Serial Transmissions



$n = 8, 16, 32$

Parallel Transmission



Serial Transmission

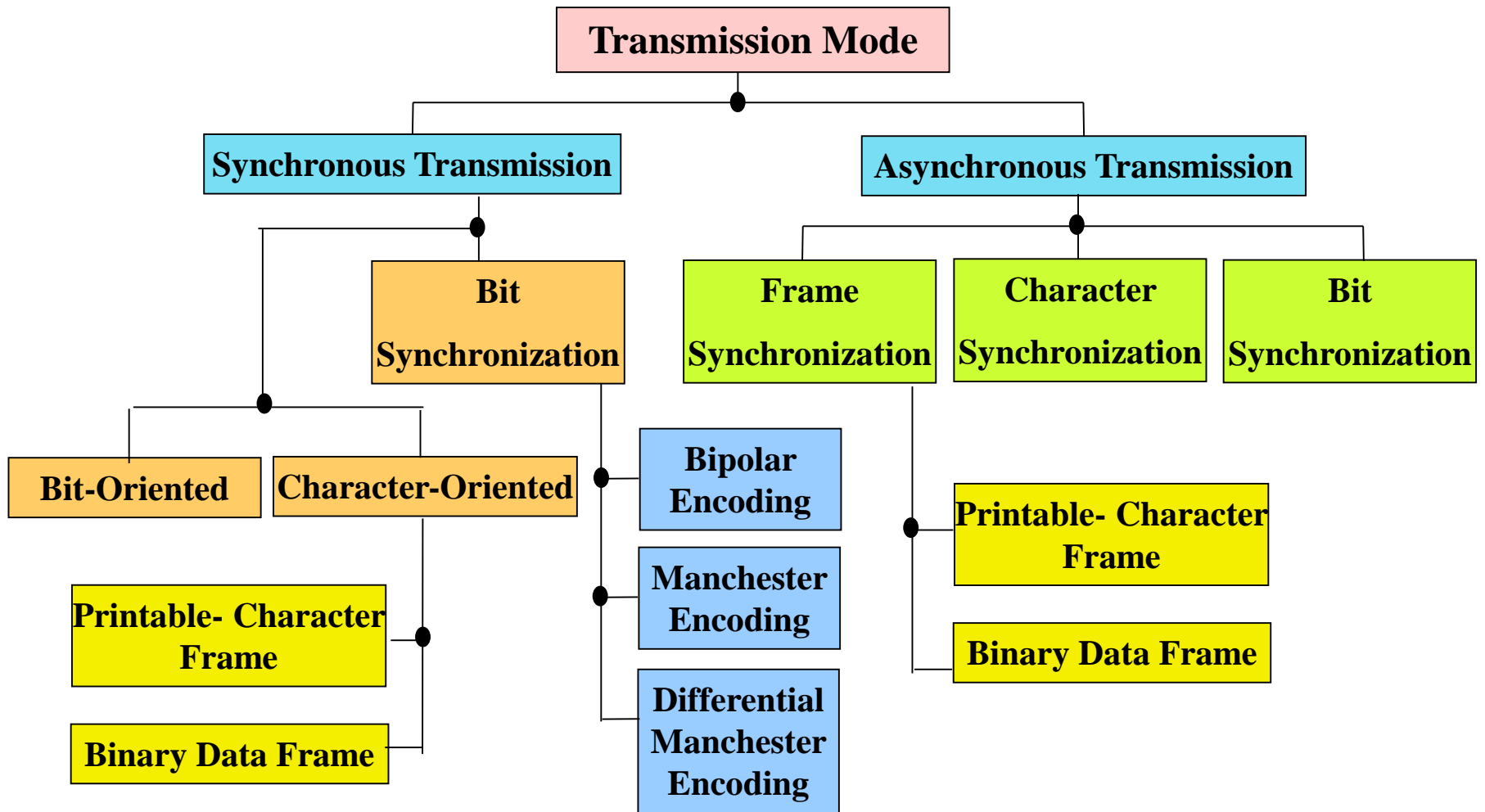
Communication Modes

- When data is transmitted between two pieces of equipment, three communication modes of operation can be used.
 - **Simplex:** Data is to be transmitted in one direction only.
 - **Half-Duplex:** This is used when two interconnected devices which to interchange information alternately.
 - **Duplex or full-Duplex:** This is used when data is to be exchanged between two connected devices in both directions simultaneously.

Transmission Modes

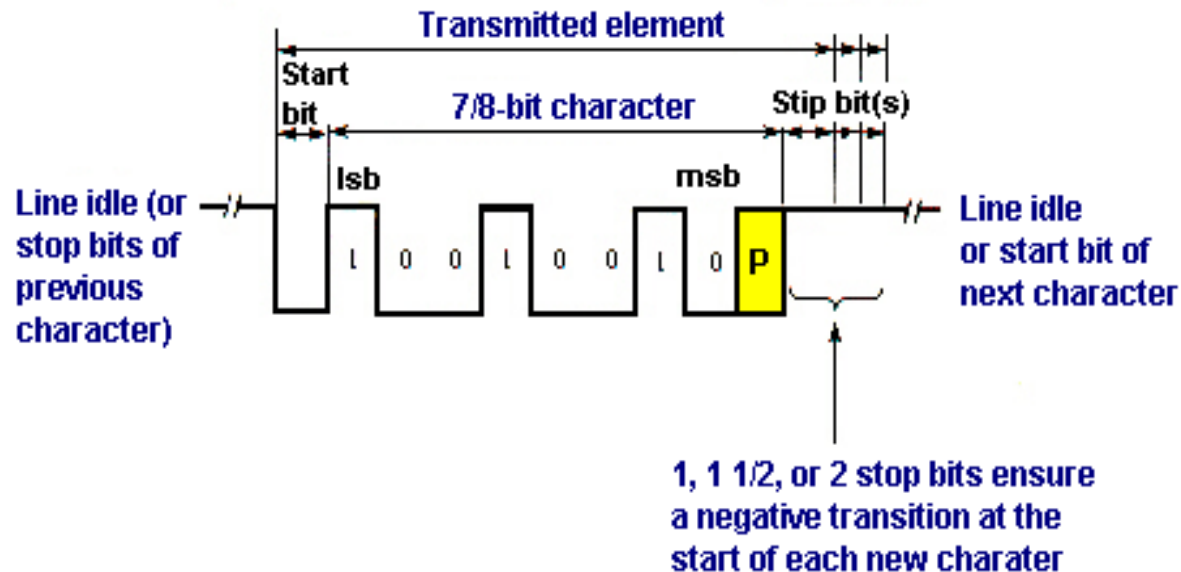
- For the receiving device to interpret the bit pattern correctly, it must be able to determine the following:
 - **Bit Synchronization**: The start of each bit cell period (in order to sample the incoming signal in the middle of the bit cell).
 - **Character Synchronization**: The start and end of each character or byte.
 - **Frame Synchronization**: The start and end of each complete message block (*frame*).
- We can accomplish synchronization in one of two ways:
 - 1. Asynchronous Transmission.**
 - 2. Synchronous Transmission.**

Transmission Modes



Asynchronous Transmission

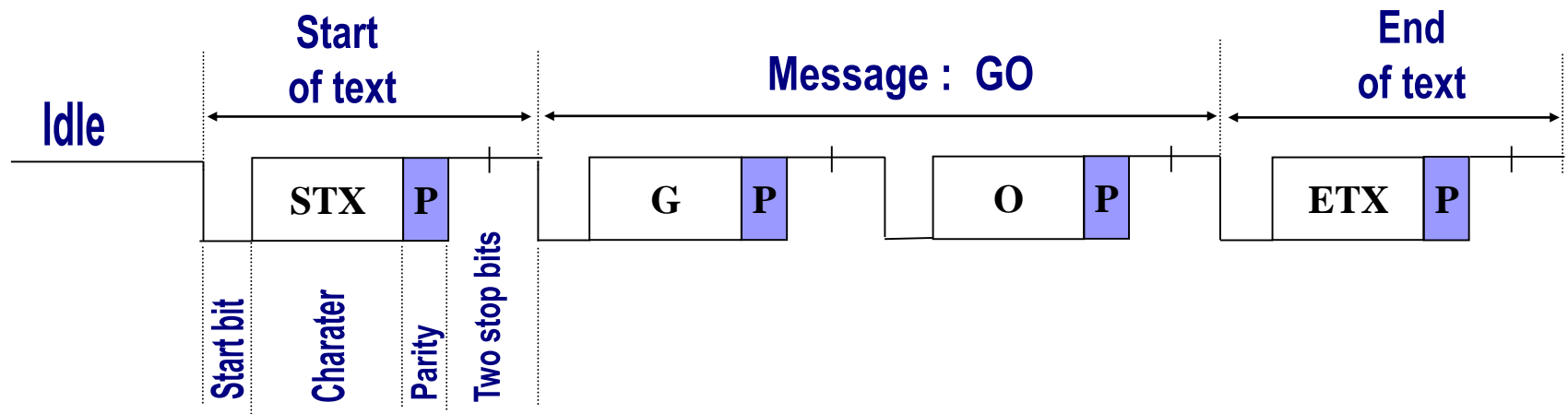
- In *asynchronous transmission*, the receiver clock ($R \times C$) runs *unsynchronized* with respect to the incoming signal ($R \times D$).
- Each character (byte) is encapsulated between an additional **start bit** and one or more **stop bits**.
- The state of the signal on the transmission line between characters is **idle** state.



Asynchronous Transmission

Example:

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

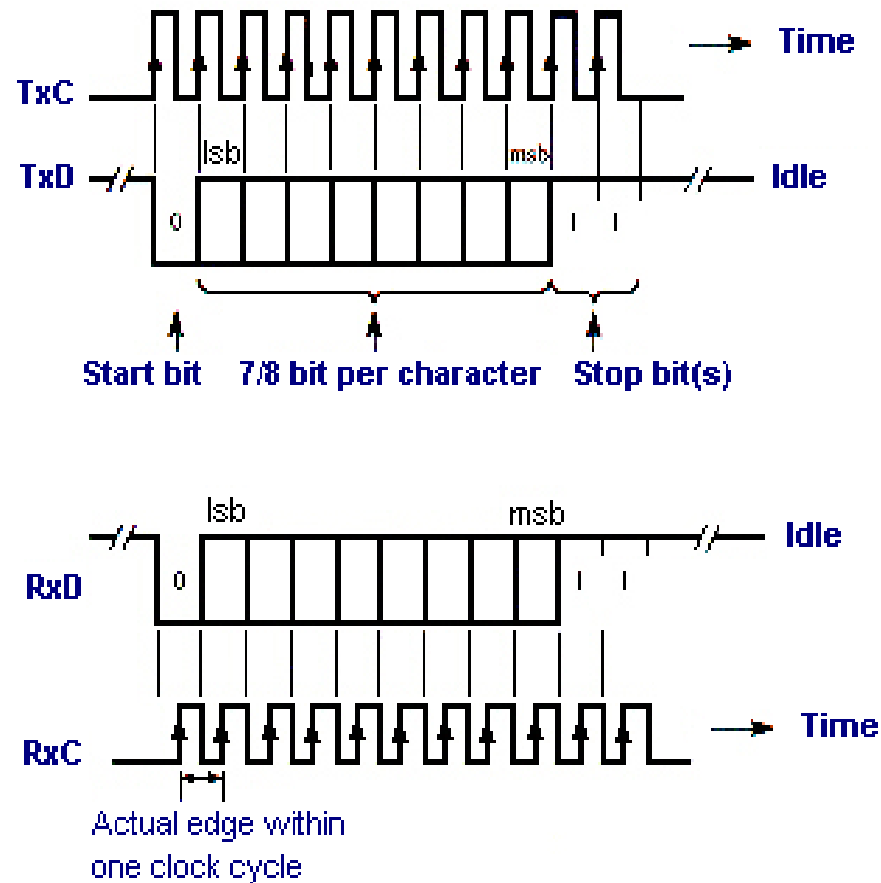


Asynchronous Transmission

- **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:



Asynchronous Transmission

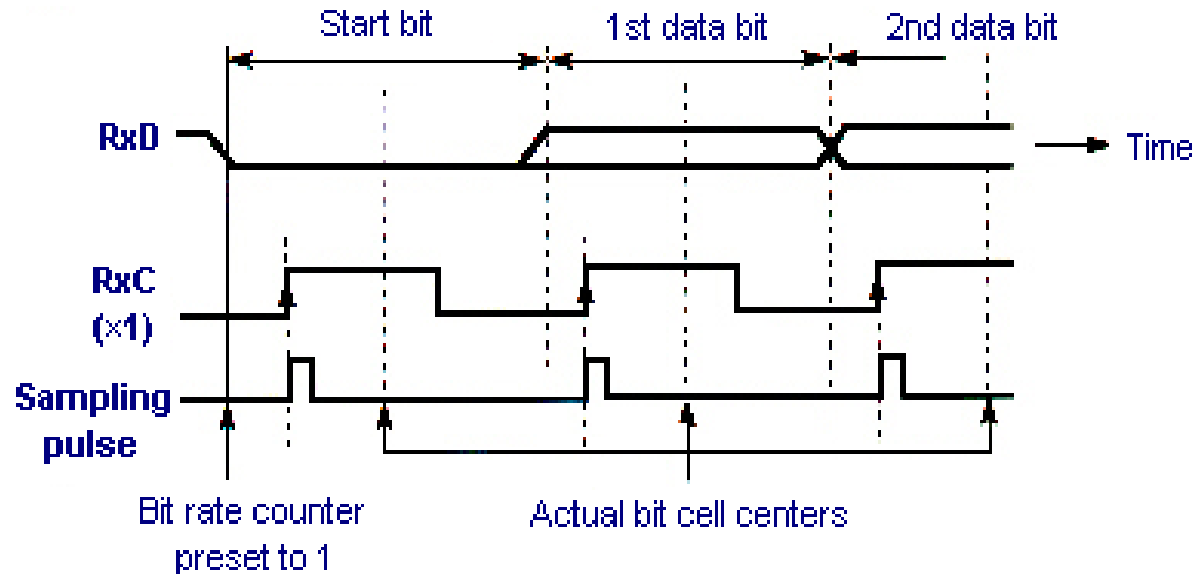
Bit Synchronization in Asynchronous Transmission:

- 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:

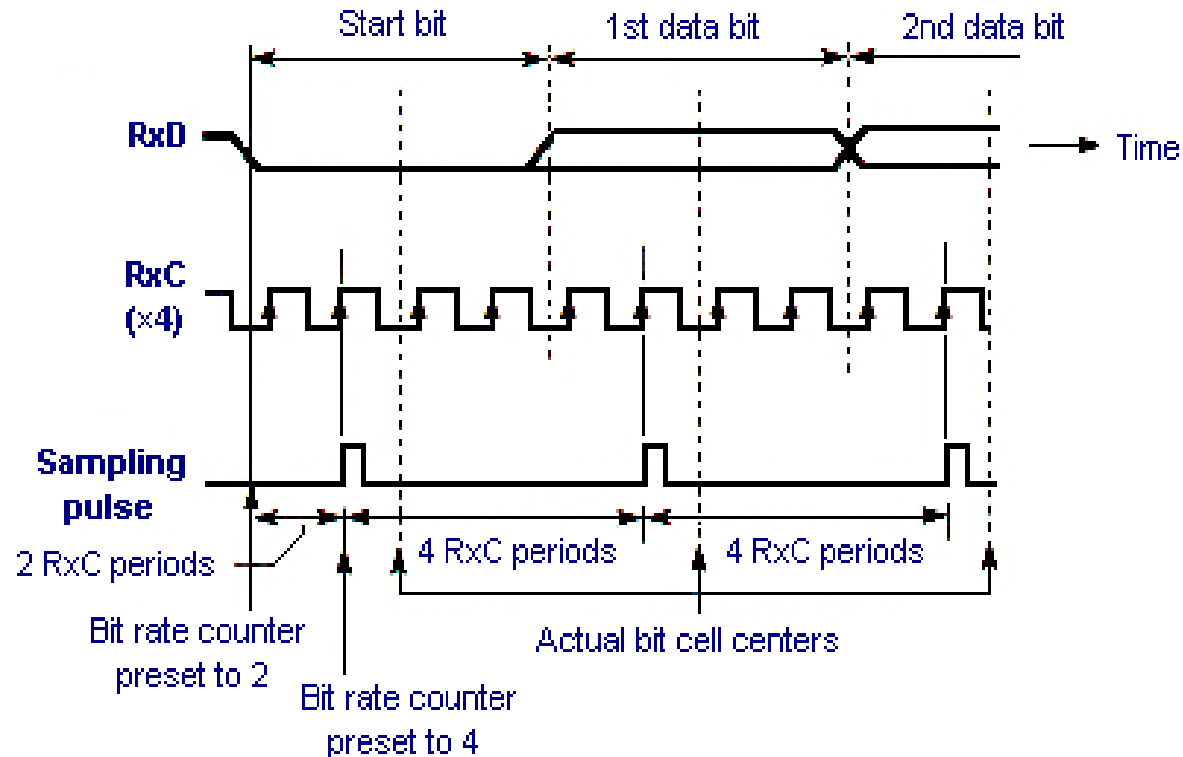
$N = 1$



Asynchronous Transmission

Bit Synchronization in Asynchronous Transmission:

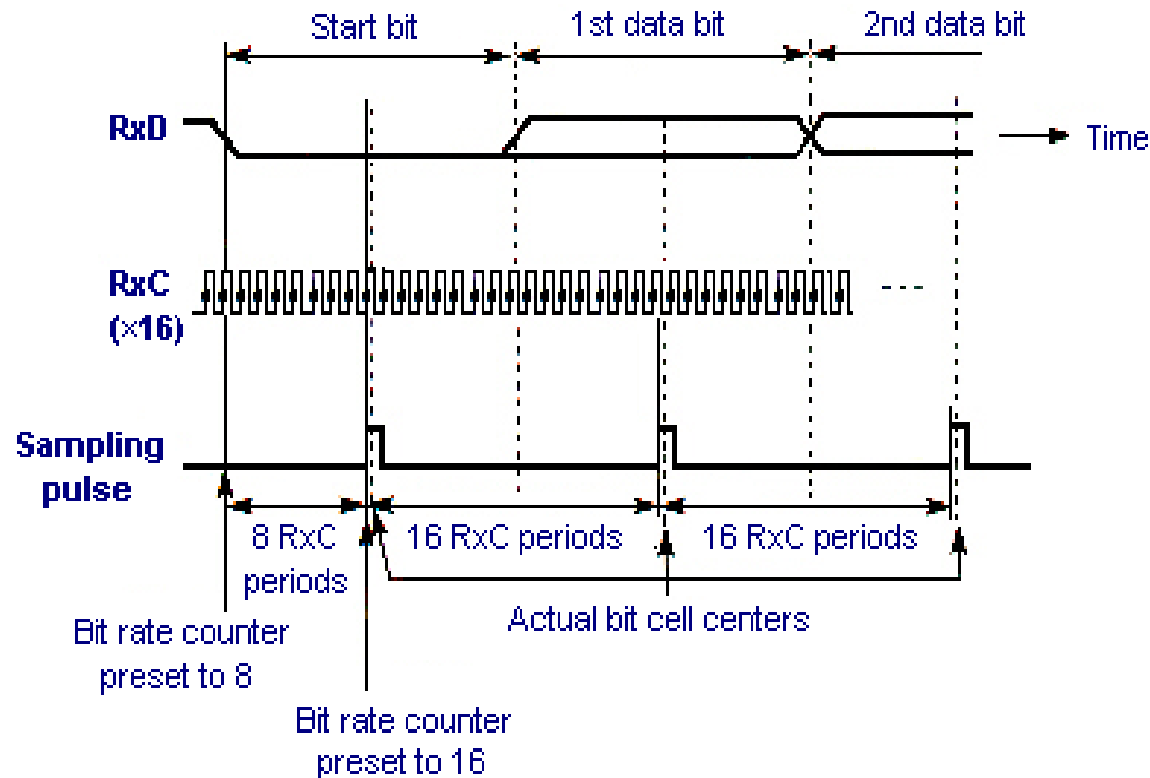
$N = 4$



Asynchronous Transmission

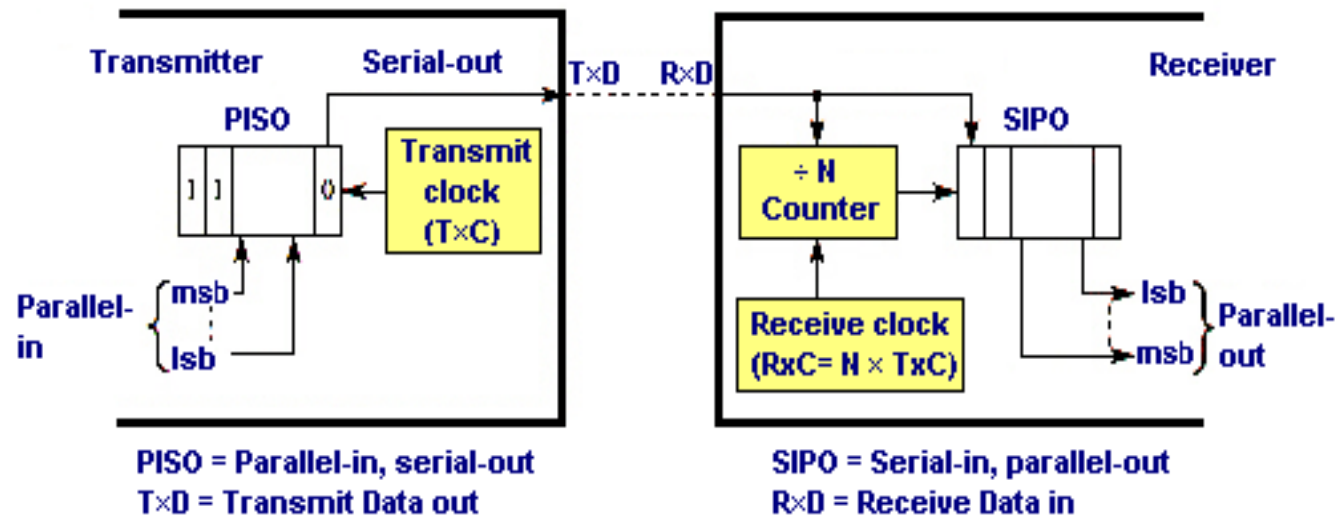
Bit Synchronization in Asynchronous Transmission:

$N = 16$



Asynchronous Transmission

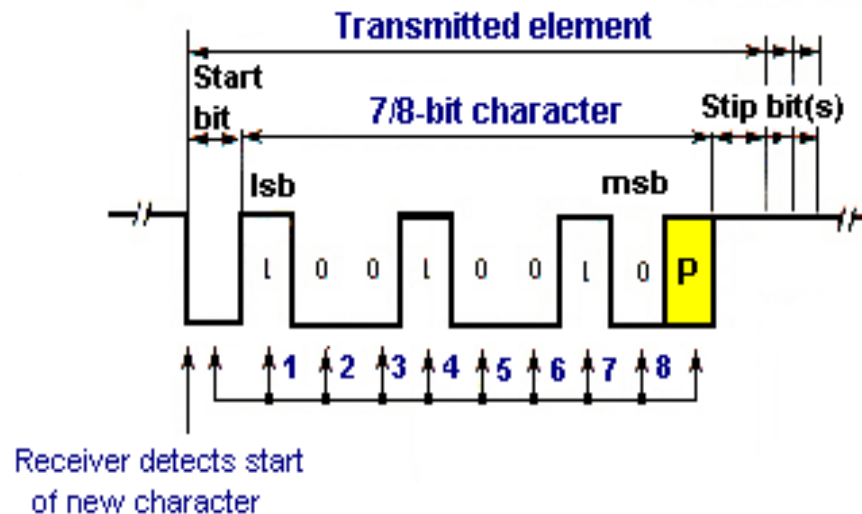
Principle of operation and Timing:



Asynchronous Transmission

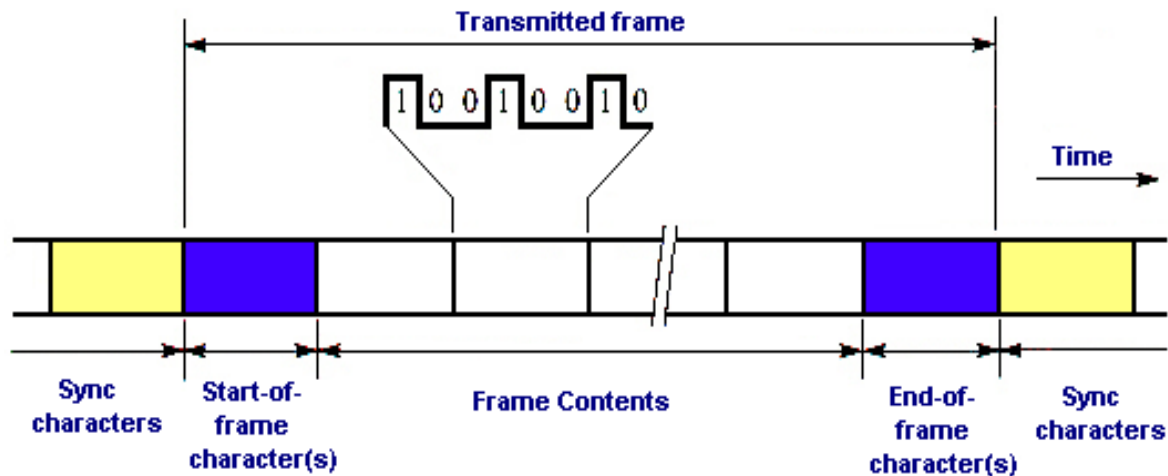
Character Synchronization in Asynchronous Transmission:

- After the start bit is detected, the receiver achieves character synchronization simply by counting the programmed number of bits.



Synchronous Transmission

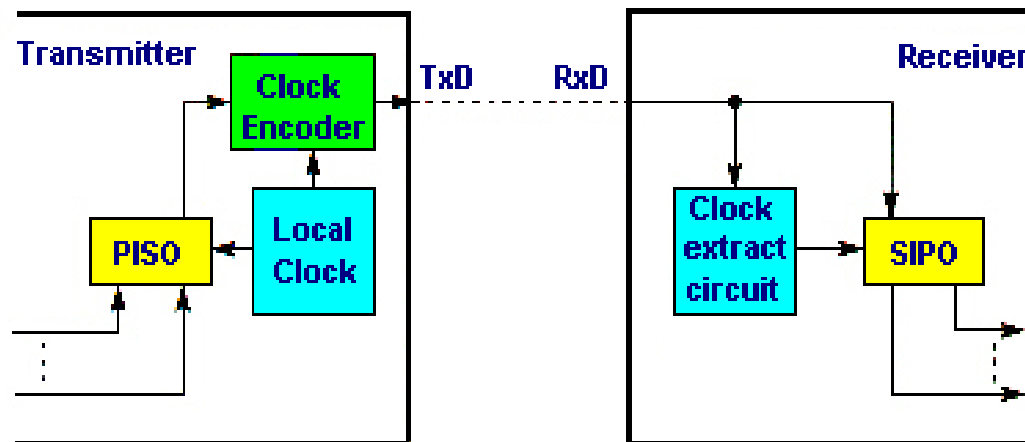
- The complete block or frame of data is transmitted as a **contiguous stream** with no delay between each 8-bit element.



Synchronous Transmission

Bit Synchronization using Synchronous Transmission:

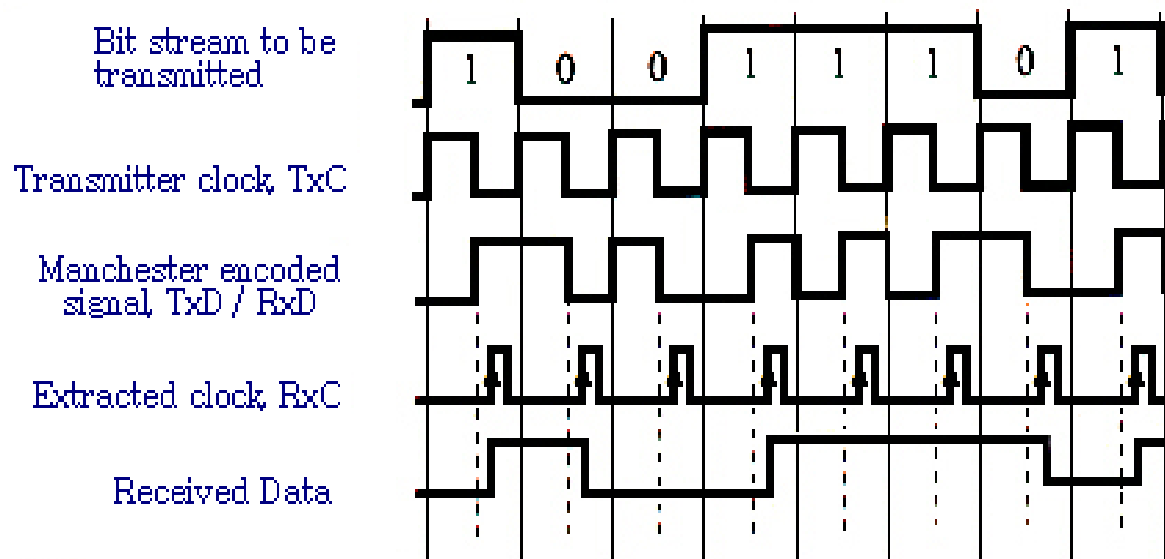
- With synchronous transmission, the receiver clock ($R \times C$) operates in synchronism with the received data signal ($R \times D$).
- **Clock Encoding and Extraction:** The clock information is embedded into the transmitted signal and subsequently extracted by the receiver.



Synchronous Transmission

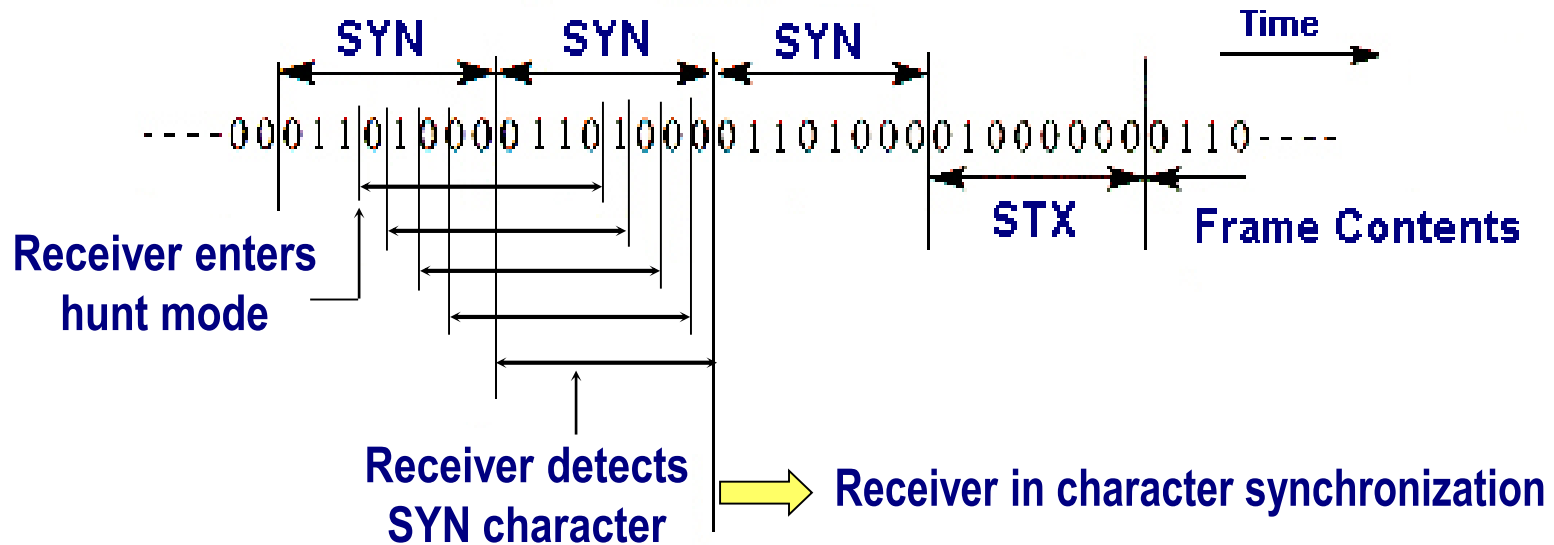
Bit Synchronization using Synchronous Transmission:

Clock Encoding and Extraction:



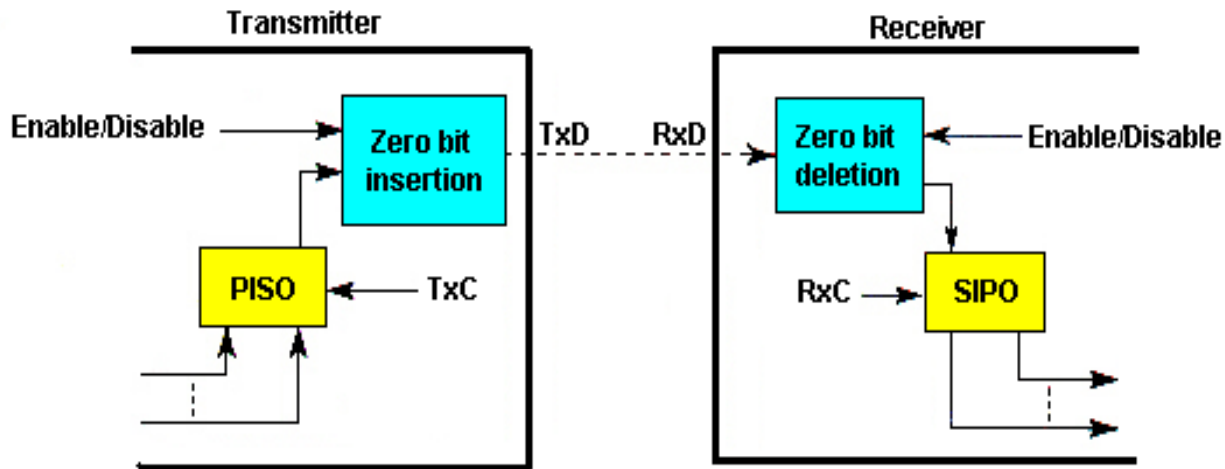
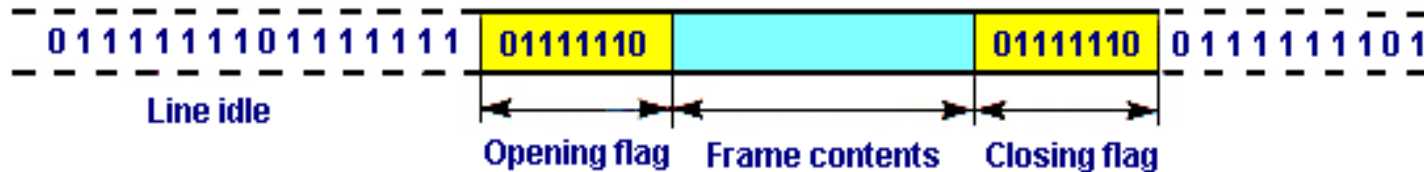
Synchronous Transmission

1. Character-Oriented Synchronous Transmission:



Synchronous Transmission

2. Bit-Oriented Synchronous Transmission:



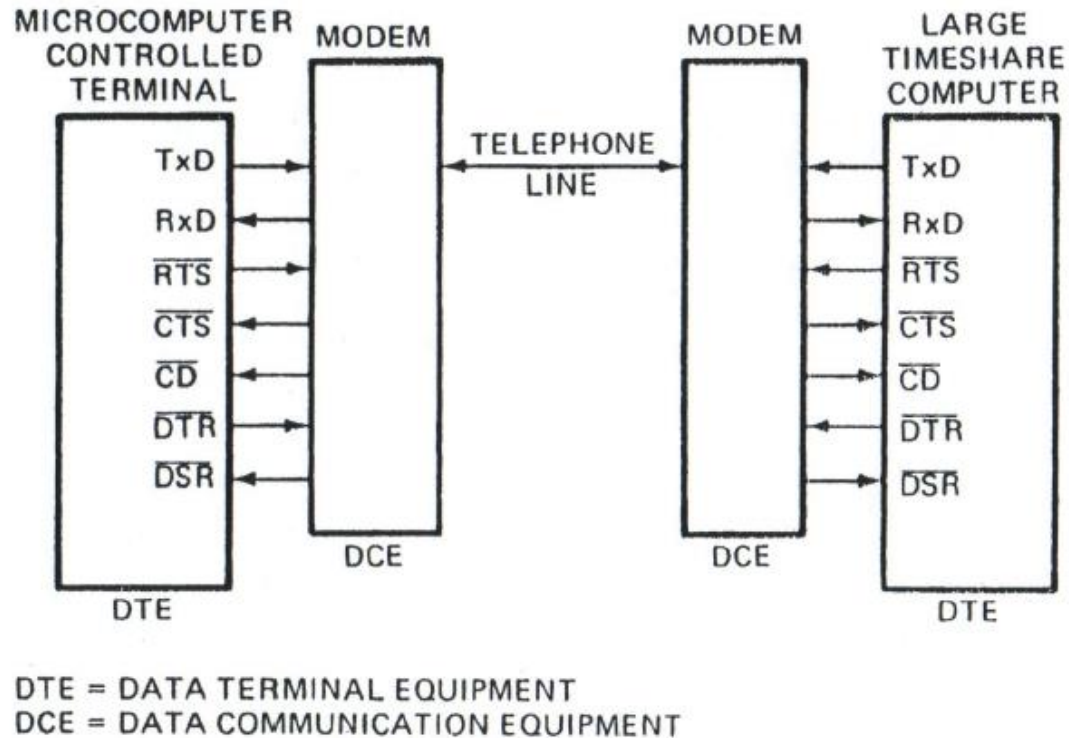
Sequence of Modem Control Signals

- 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 sets 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.

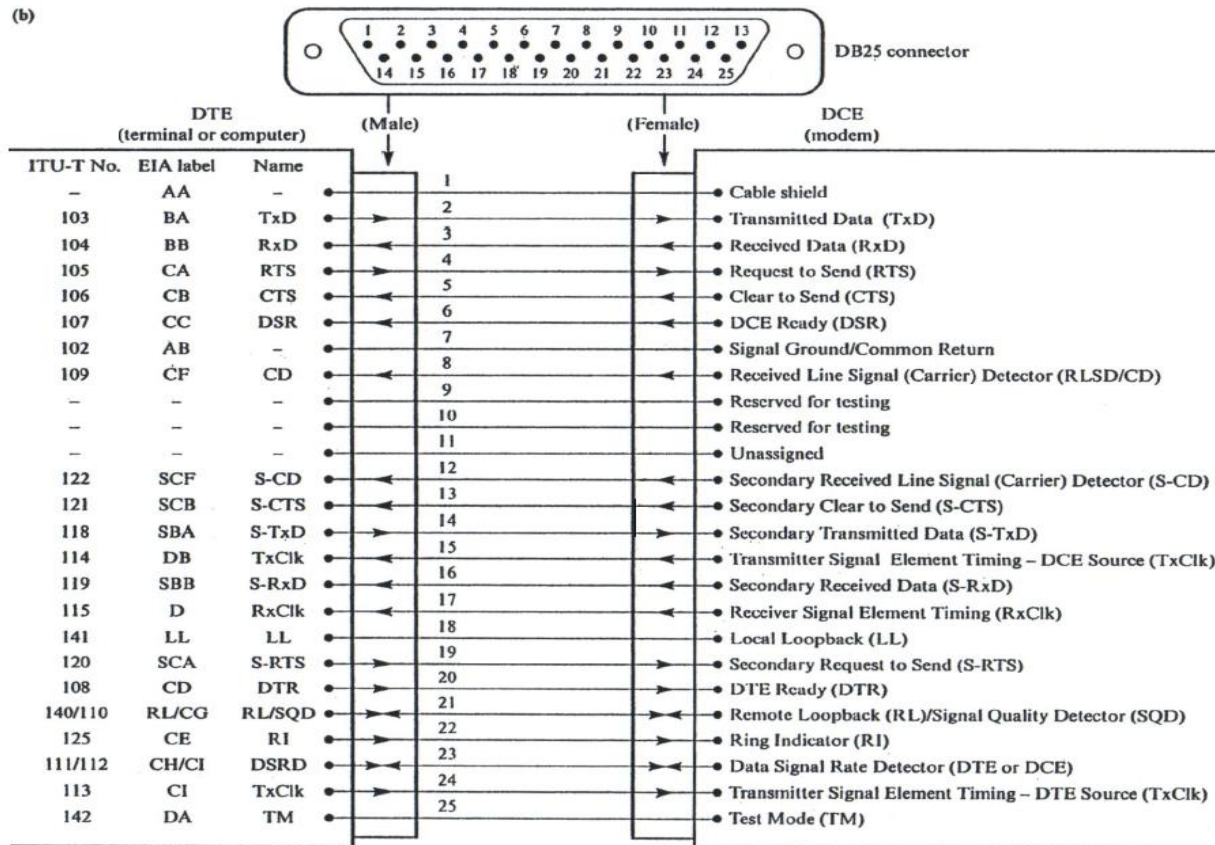
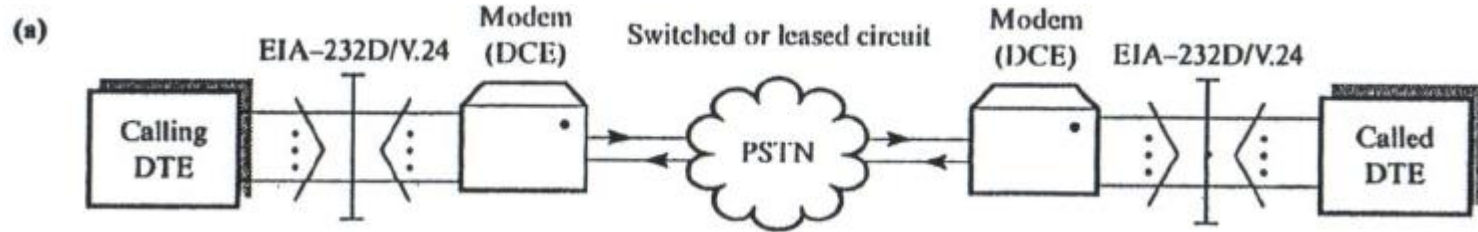
Sequence of Modem Control Signals

- 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 set 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.

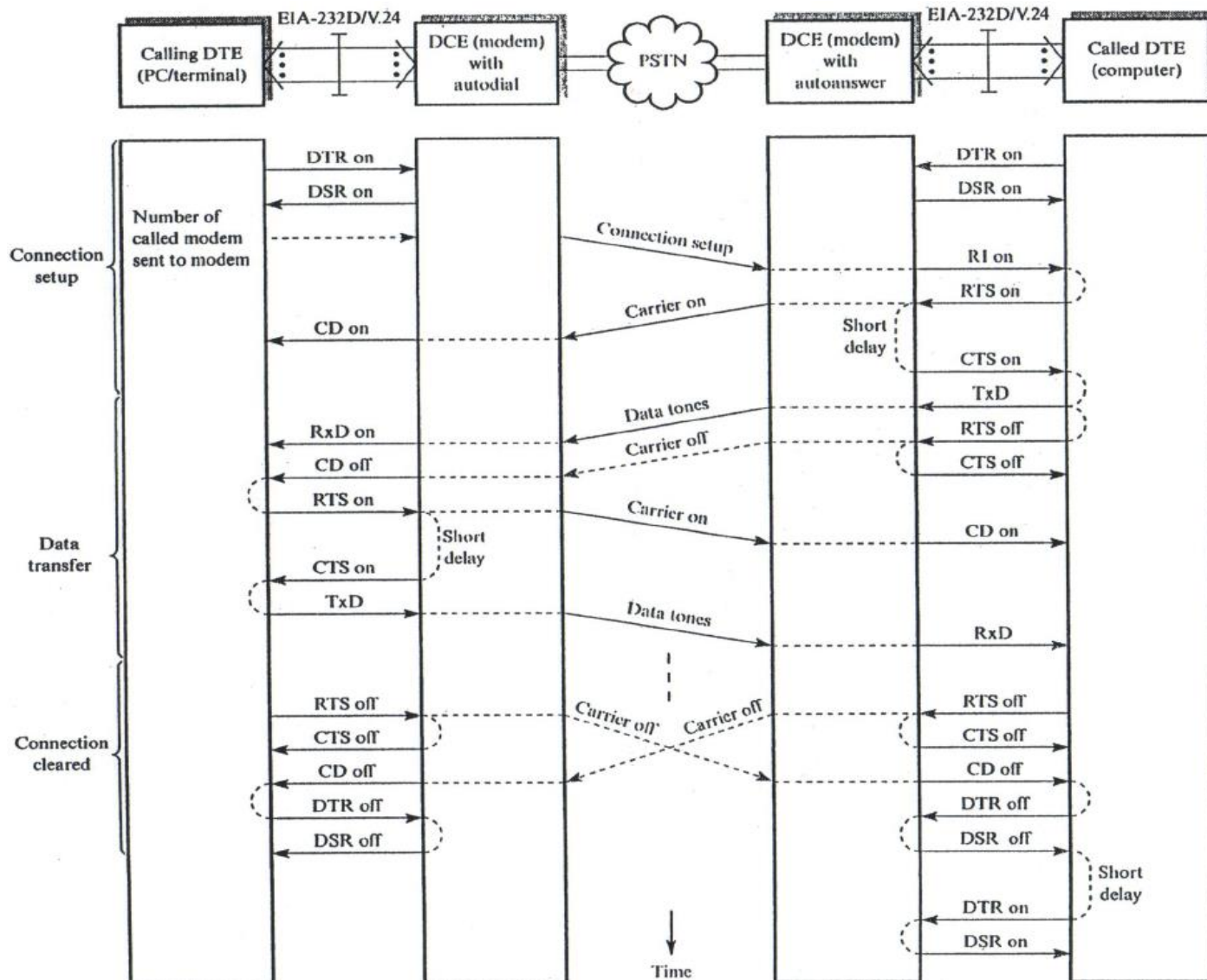
Digital Data Transmission using Modem



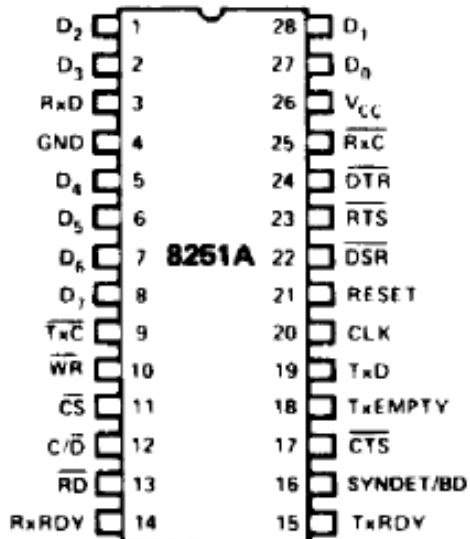
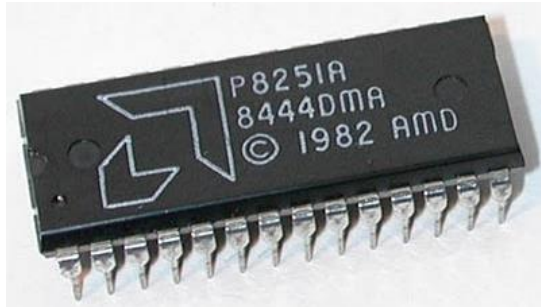
Digital Data Transmission using Modem



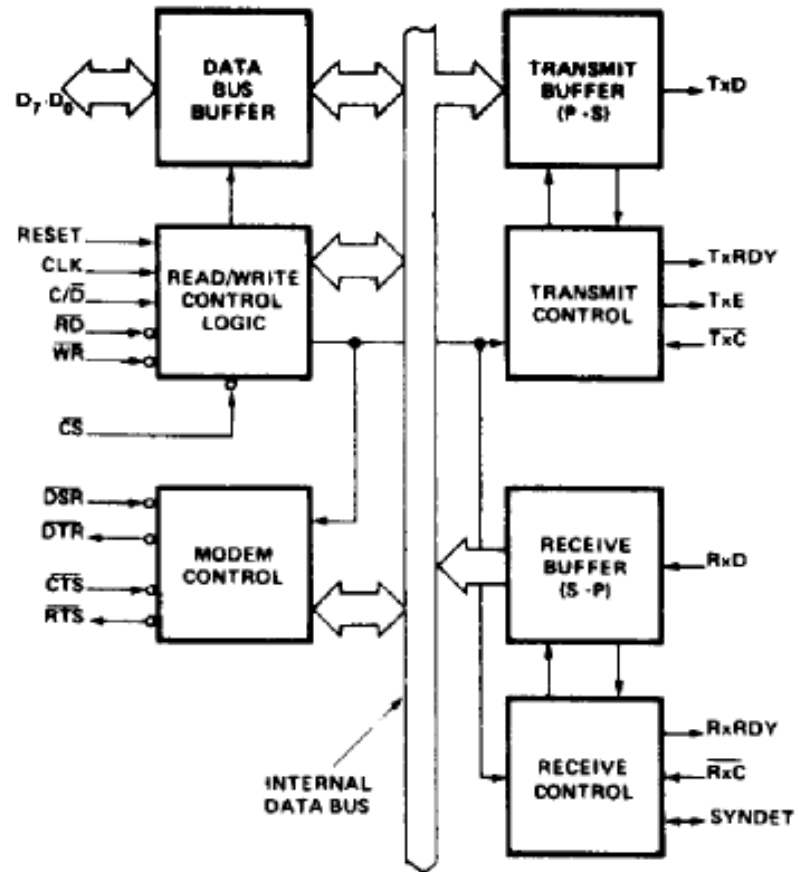
Digital Data Transmission using Modem



USART 8251A

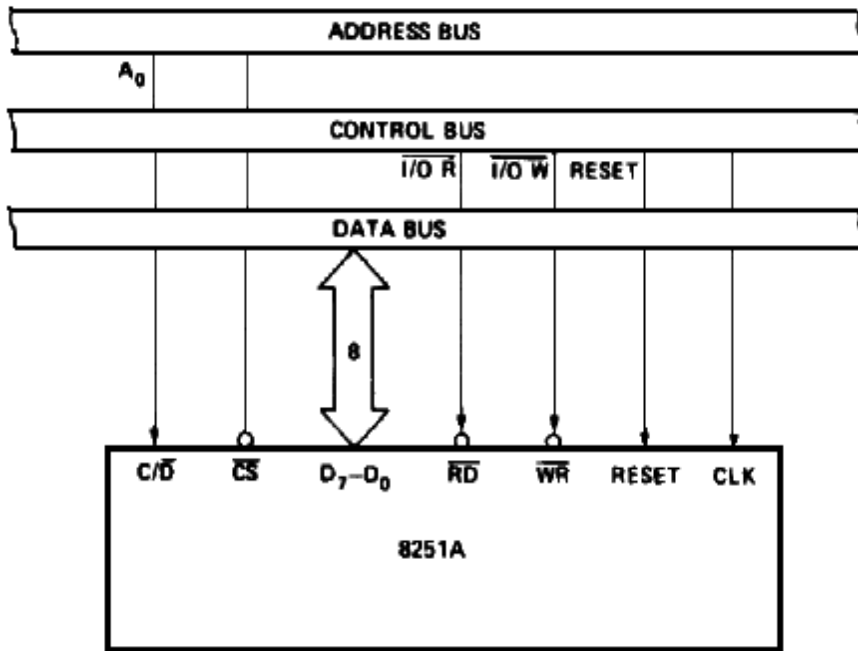


Pin Configuration



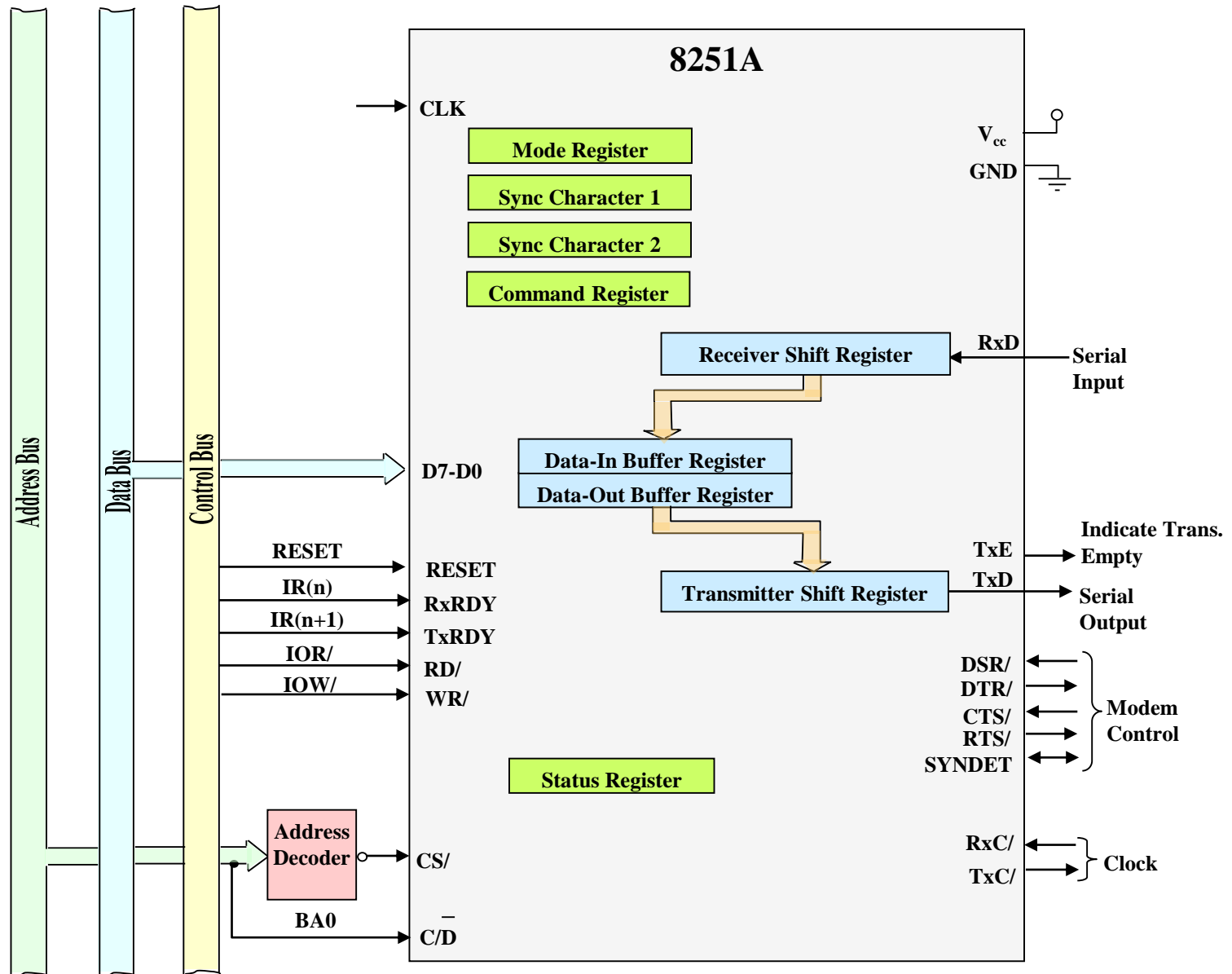
Block Diagram

Interfacing 8251A to 8088



$\overline{C/D}$	RD/	WR/	CS/	Function
0	0	1	0	8251A DATA → DATA BUS
0	1	0	0	DATA BUS → 8251A DATA
1	0	1	0	STATUS → DATA BUS
1	1	0	0	DATA BUS → Control
X	1	1	0	DATA BUS → 3-STATE
X	X	X	1	DATA BUS → 3-STATE

8251A Communication Interface



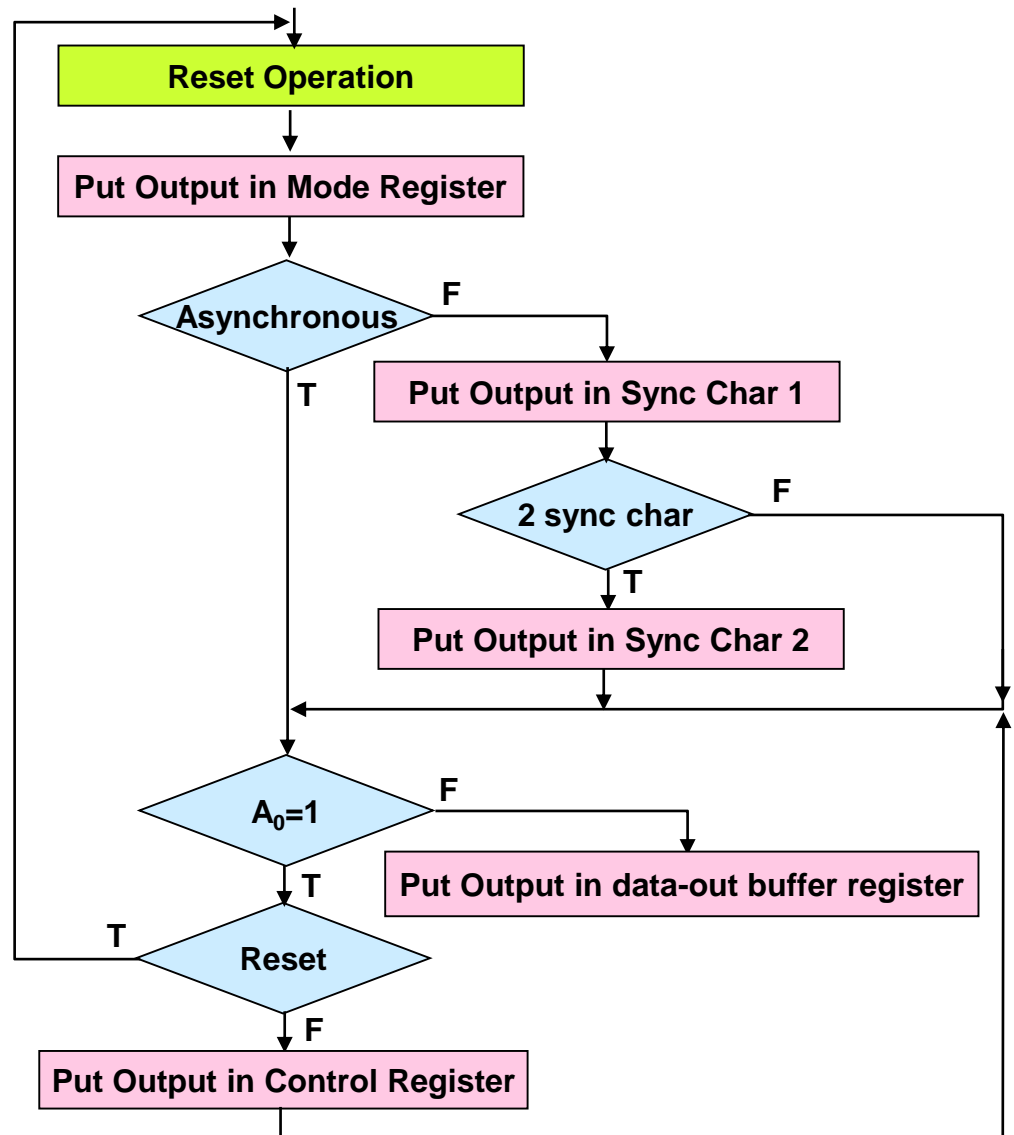
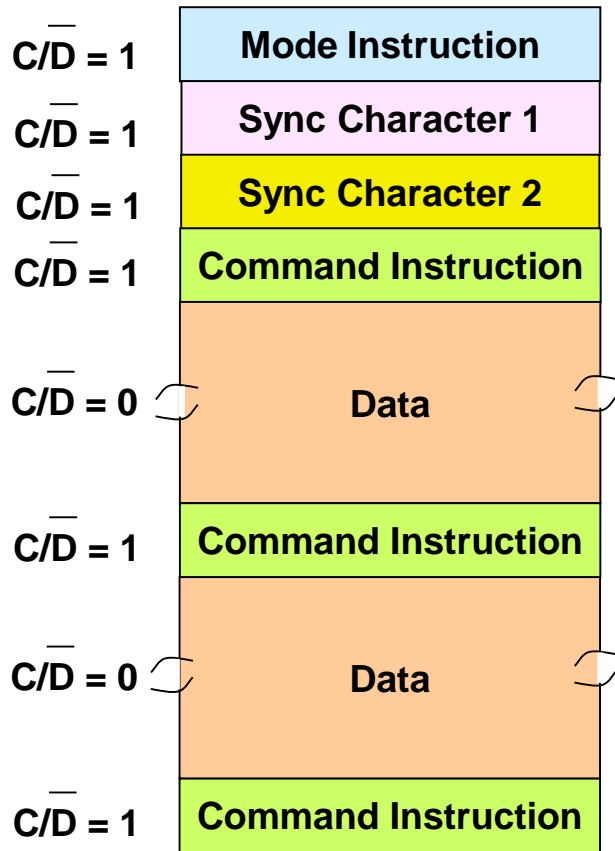
Signals of 8251A

- 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.

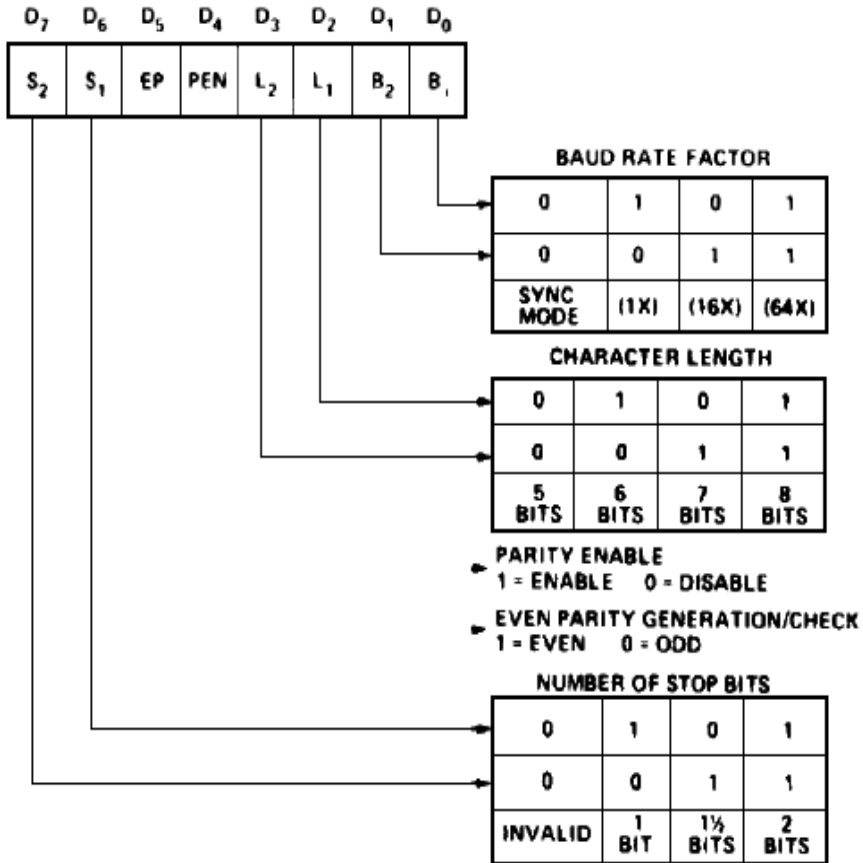
Signals of 8251A

- 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.

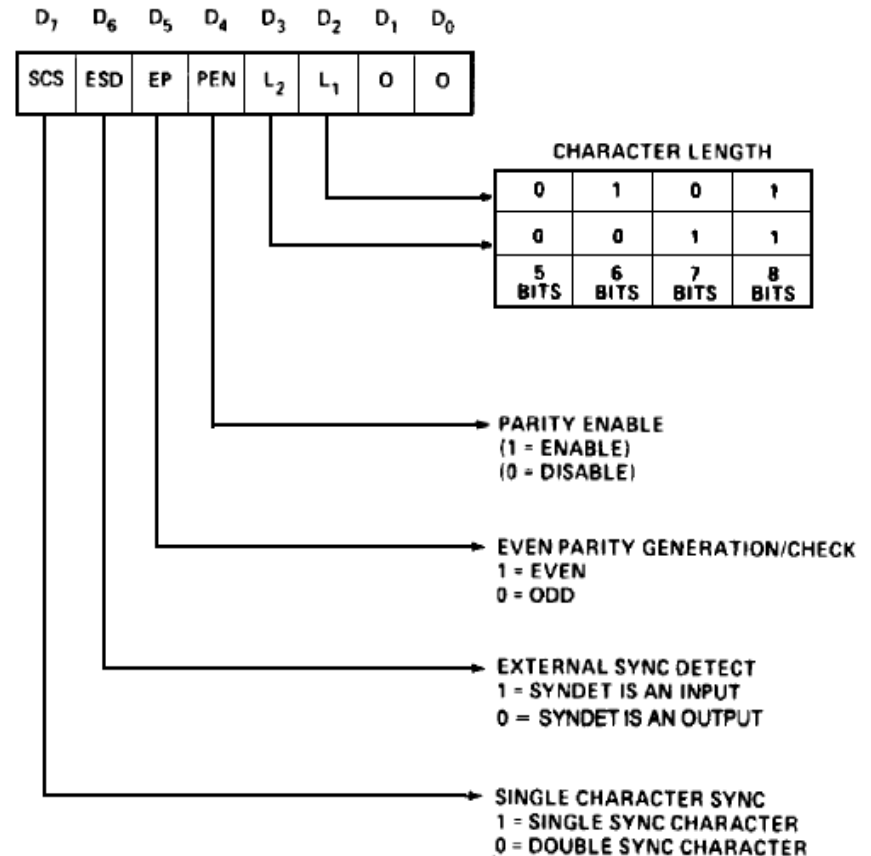
Initializing 8251A



8251A Mode Word

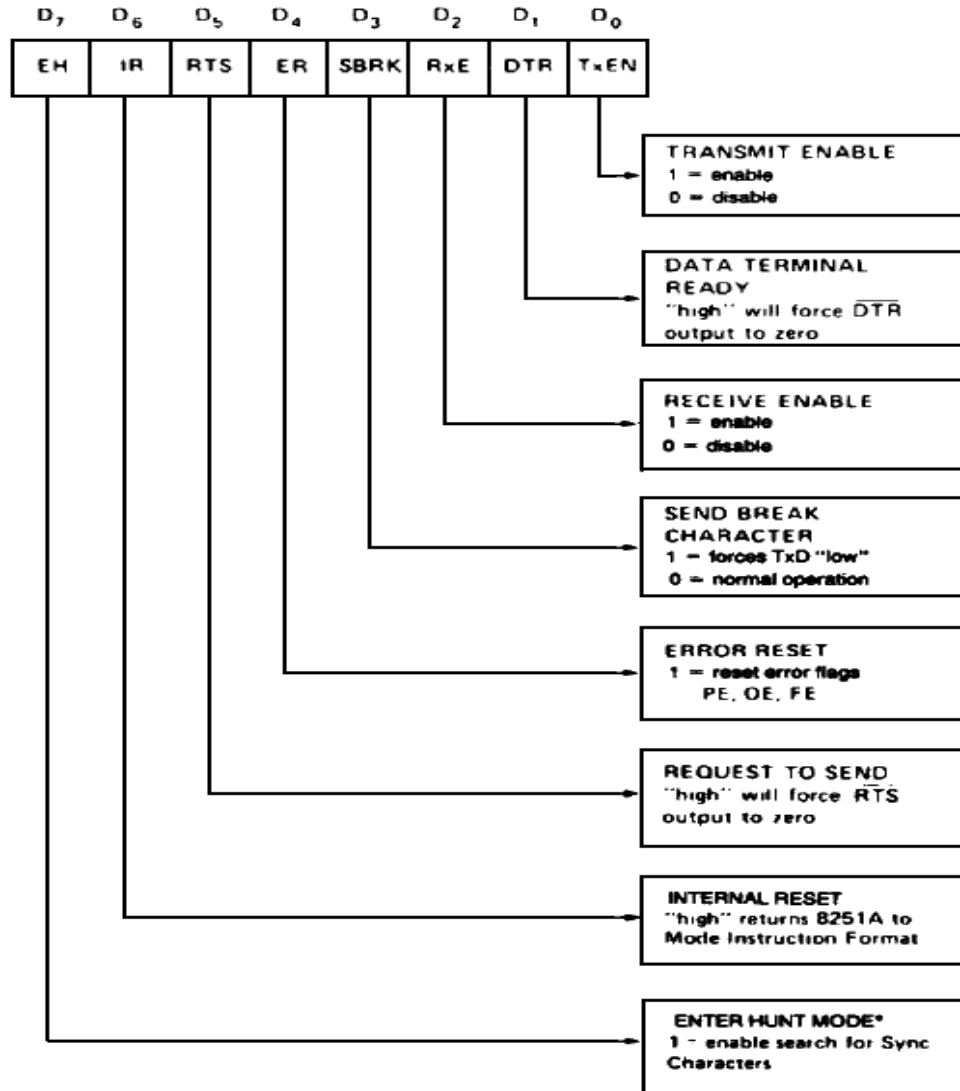


Asynchronous



Synchronous

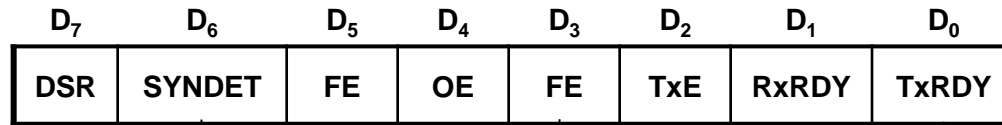
8251A Command Word



8251A 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).

8251A Status Word



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

Indicate 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.

8251A Internal Reset on Power-Up

- When power is first applied, the 8251A may come up in the mode, SYN character or command format.
- It is safest to execute the worst-case initialization sequence (**SYNC** mode with two **SYN** characters). Loading three 00H consecutively into the device with C/#D = 1.
- An **internal reset command** (40H) may then be issued to return the device to **mode word**.
- The **mode word** must then be issued, and followed by the **command word**.

Example1: 8251A Worst-Case Initialization

```

S0      EQU 4010H
S1      EQU 4011H
;
; . . .
MOV     DX, S1
MOV     AL, 00H
OUT     DX, AL
MOV     CX, 2
D0: LOOP D0
OUT     DX, AL
MOV     CX, 2
D1: LOOP D1
OUT     DX, AL
MOV     CX, 2
D2: LOOP D2
OUT     DX, AL
MOV     DX, S1
MOV     AL, 40H
OUT     DX, AL
MOV     CX, 2
D3: LOOP D3
MOV     AL, 11001110B
OUT     DX, AL
MOV     CX, 2
D4: LOOP D4
;
;
MOV     AL, 00110111B
OUT     DX, AL
; Send three zeros to guarantee device is in
; the command instruction.
; Send internal reset command.
; 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.
; 1 1 0 0 1 1 1 0 ; Command Word.
; \ \ \ \ \ \ \ \ \ \ ; Transmit Enable.
; \ \ \ \ \ \ \ \ \ \ ; DTR will Output 0.
; \ \ \ \ \ \ \ \ \ \ ; Receive Enable.
; \ \ \ \ \ \ \ \ \ \ ; Normal Operation.
; \ \ \ \ \ \ \ \ \ \ ; Reset All Error Flags.
; \ \ \ \ \ \ \ \ \ \ ; RST Output 0.
; \ \ \ \ \ \ \ \ \ \ ; Do not Return to Mode Word.
; \ \ \ \ \ \ \ \ \ \ ; Diable Hunt Mode.

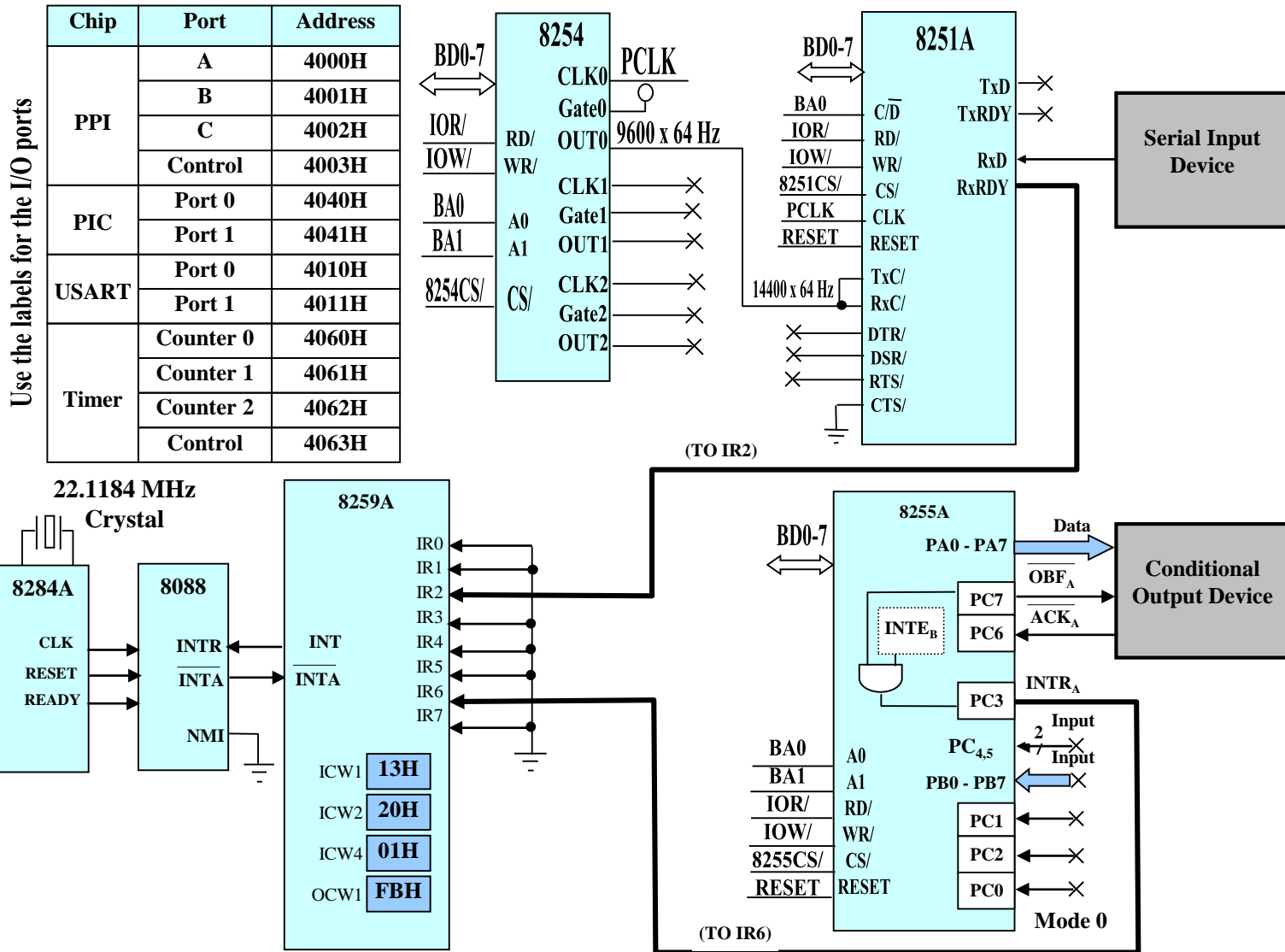
```

Example 1: Transmitting & Receiving Using Programmed I/O

```
; Instructions for Transmitting Data Using Programmed I/O
    MOV     DX, 4011H
TEST1:    IN      AL, DX           ; Read Status
    TEST   AL, 0000001B        ; Test Transmit Ready
    JZ     TEST1
    MOV    DX, 4010H
    MOV    AL, VAR1            ; Load Data to Send it
    OUT   DX, AL
```

```
; Instructions for Receiving Data Using Programmed I/O
    MOV     DX, 4011H
TEST2:    IN      AL, DX           ; Read Status
    TEST   AL, 0000010B        ; Test Transmit Ready
    JZ     TEST2
    MOV    DX, 4010H
    IN     AL, DX             ; Get Data
    MOV    VAR1, AL
```

Example 2: Interrupt Driven I/O



Example 2: Interrupt Driven I/O (Cont'd)

LOC	OBJ	LINE	SOURCE
		1	NAME PPI4
----		2	DATA SEGMENT AT 40H
0000		3	ORG 0H
0000	??	4	VAR1 DB ?
----		5	DATA ENDS
		6	;
----		7	STACK SEGMENT AT 50H
0000	(50	8	DW 50 DUP(?)
	????		
)		
0064		9	STK_TOP LABEL WORD
----		10	STACK ENDS
		11	;
----		12	IVT SEGMENT AT 0H
0148		13	ORG 52H*4
0148		14	INT52 LABEL DWORD
0148	000300FE	15	DD FIRST
0158		16	ORG 56H*4
0158		17	INT56 LABEL DWORD
0158	000500FE	18	DD SECOND
----		19	IVT ENDS
		20	;
----		21	EPROM SEGMENT AT 0FE00H
		22	ASSUME CS:EPROM, SS:STACK
0000		23	ORG 0H
0000		24	BEGIN LABEL FAR
		25	; IVT Initialization
		26	ASSUME DS: IVT
0000	B80000	27	MOV AX, IVT
0003	8ED8	28	MOV DS, AX
0005	C70648010003	29	MOV WORD PTR INT52, OFFSET INT52_SR_P
000B	C7064A0100FE	30	MOV WORD PTR INT52+2, SEG INT52_SR_P
0011	C70658010005	31	MOV WORD PTR INT56, OFFSET INT56_SR_P
0017	C7065A0100FE	32	MOV WORD PTR INT56+2, SEG INT56_SR_P

Example 2: Interrupt Driven I/O (Cont'd)

```

                                33      ;
                                34      ; DS Initialization
                                35      ASSUME    DS:DATA
001D B84000                      36      MOV      AX, DATA

                                37      ; 8254 Initialization
0020 BA6380                      38      MOV      DX, 8063H
0023 B016                        39      MOV      AL, 00010110B ; Counter0, LSB, Mode 3
0025 EE                          40      OUT      DX, AL ; Timer Control Word
0026 BA6380                      41      MOV      DX, 8063H
0029 B006                        42      MOV      AL, 6 ; (22.1184 MHz/6)/(9600*64)= 6
002B EE                          43      OUT      DX, AL ; Timer Control Word

                                44      ; 8259A Initialization
002C BA4080                      45      MOV      DX, 8040H
002F B013                        46      MOV      AL, 13H
0031 EE                          47      OUT      DX, AL ; ICW1
0032 42                          48      INC      DX
0033 B050                        49      MOV      AL, 50H
0035 EE                          50      OUT      DX, AL ; ICW2
0036 B001                        51      MOV      AL, 1H
0038 EE                          52      OUT      DX, AL ; ICW4
0039 B0FB                        53      MOV      AL, 0FBH
003B EE                          54      OUT      DX, AL ; OCW1
```

Example 2: Interrupt Driven I/O (Cont'd)

```

55      ; 8251A Initialization
003C BA1180 56      MOV     DX, 8011H
003F B000   57      MOV     AL, 00H
0041 EE     58      OUT     DX, AL      ; Send three zeros
0042 B90200 59      MOV     CX, 2
0045 E2FE   60      D0:     LOOP    D0
0047 EE     61      OUT     DX, AL
0048 B90200 62      MOV     CX, 2
004B E2FE   63      D1:     LOOP    D1
004D EE     64      OUT     DX, AL
004E B90200 65      MOV     CX, 2
0051 E2FE   66      D2:     LOOP    D2
0053 EE     67      OUT     DX, AL
0054 B040   68      MOV     AL, 40H
0056 EE     69      OUT     DX, AL      ; Send internal reset
0057 B90200 70      MOV     CX, 2
005A E2FE   71      D3:     LOOP    D3
005C B0DF   72      MOV     AL, 11011111B ; 8251 Mode Word
005E EE     73      OUT     DX, AL
005F B90200 74      MOV     CX, 2
0062 E2FE   75      D4:     LOOP    D4
0064 B037   76      MOV     AL, 00110111B ; 8251 Command Word
0066 EE     77      OUT     DX, AL
78      ; 8255A Initialization
0067 BA0380 79      MOV     DX, 8003H
006A B0B9   80      MOV     AL, 0B9H
006C EE     81      OUT     DX, AL
006D B009   82      MOV     AL, 09H
006F EE     83      OUT     DX, AL
84      ;
0070 FB     85      STI
0071 EBF0   86      AGAIN:  JMP     AGAIN
----      87      EPROM   ENDS
88      ;-----
```

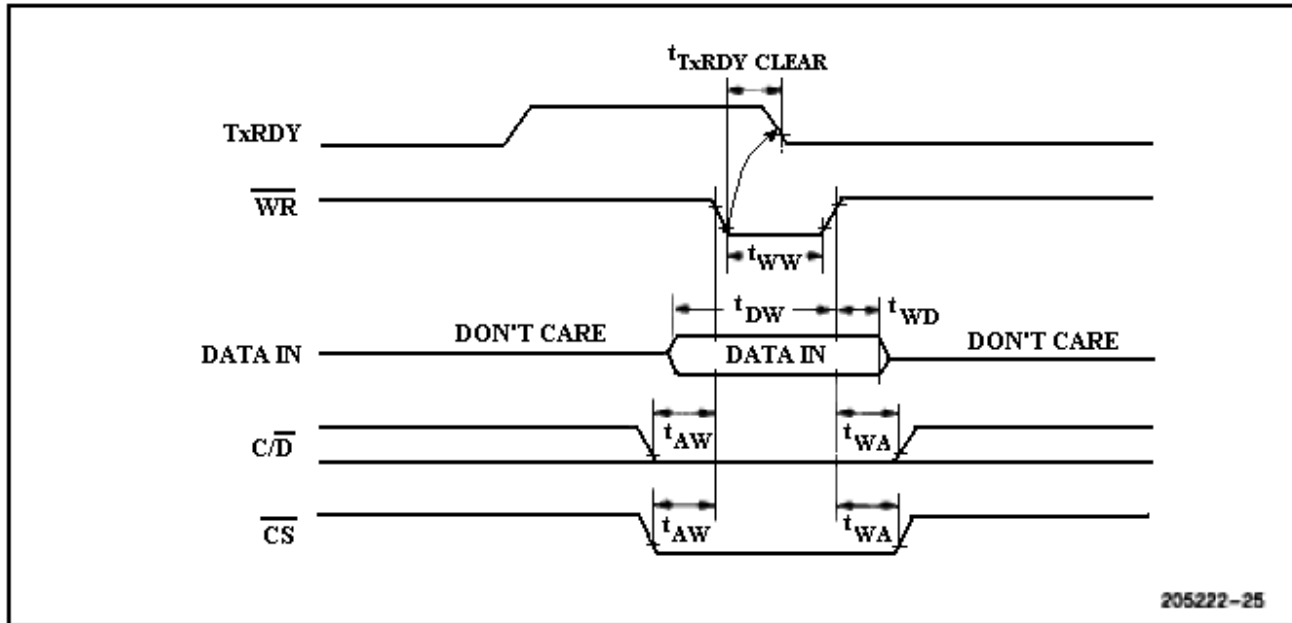
Example 2: Interrupt Driven I/O (Cont'd)

```
-----
                                89      ;Interrupt Service Routine (TYPE 52)
                                90      INT52_SR      SEGMENT AT 0FE00H
                                91
                                92      ASSUME CS:INT52_SR, SS:STACK
0300                                93
0300                                94      INT52_SR_P    PROC FAR
0300                                95      FIRST      LABEL FAR
0300 52                                96      PUSH      DX
0301 50                                97      PUSH      AX
0302 BA1080                            98      MOV       DX, 8010H
0305 EC                                99      IN       AL, DX      ; Read Serial Port
0306 A20000                            100     MOV      VAR1, AL
0309 BA4180                            101     MOV      DX, 8041H
030C B0BF                              102     MOV      AL, 10111111B
030E EE                              103     OUT      DX, AL      ; Unmask IR6
030F 4A                              104     DEC      DX
0310 B020                              105     MOV      AL, 20H
0312 EE                              106     OUT      DX, AL      ; EOI
0313 58                              107     POP      AX
0314 5A                              108     POP      DX
0315 CF                              109     IRET
-----                                110     INT52_SR_P    ENDP
                                111     INT52_SR      ENDS
                                ;-----
```

Example 2: Interrupt Driven I/O (Cont'd)

```
-----
112      ;Interrupt Service Routine (TYPE 56)
113      INT56_SR      SEGMENT AT 0FE00H
114
115
116      ASSUME CS:INT56_SR, SS:STACK
0500
117      INT56_SR_P    PROC FAR
118      SECOND        LABEL FAR
0500 52      PUSH      DX
0501 50      PUSH      AX
0502 BA0080    MOV       DX, 8000H
0505 A00000    MOV       AL, VAR1
0508 EE       OUT       DX, AL      ; Write to Port A
0509 BA4180    MOV       DX, 8041H
050C B0FB     MOV       AL, 11111011B
050E EE       OUT       DX, AL      ; Unmask IR2
050F 4A       DEC       DX
0510 B020     MOV       AL, 20H
0512 EE       OUT       DX, AL      ; EOI
0513 58       POP       AX
0514 5A       POP       DX
0515 CF       IRET
132      INT56_SR_P    ENDP
133      INT56_SR      ENDS
134      ;-----
135
136      CODE           SEGMENT AT 0FFFFH
137
138      ASSUME CS:CODE, SS:STACK
0000
139      START:        CLI
0001 B85000    MOV       AX, STACK
0004 8ED0     MOV       SS, AX
0006 BC6400    MOV       SP, OFFSET STK_TOP
0009 EA000000FE JMP       BEGIN
-----
144      CODE           ENDS
145      END           START
ASSEMBLY COMPLETE, NO ERRORS FOUND
```

WRITE DATA CYCLE (CPU → USART)



READ DATA CYCLE (CPU ← USART)

