

Microprocesadores 1121060

Tema 8. PUERTOS DE E/S Y SU INTERFAZ.

Tema 8. PUERTOS DE E/S Y SU INTERFAZ.

1. Conceptos de transmisión serie y paralelo
 2. Decodificación de dirección de puertos de E/S
 3. Interfaz de puertos de E/S con el Microprocesador
 4. Programación de Puertos de E/S
 5. Interrupciones por hardware
 6. Temporizadores
-

Decodificación de la memoria

Conceptos básicos en la interfaz de memoria.

- La función principal de la interfaz de memoria es la de permitir al procesador leer o escribir en un registro dado de la memoria. Para realizar esta operación, el microprocesador deberá:
 1. Ser capaz de activar el circuito integrado
 2. Identificar el registro que se va a acceder
 3. Habilitar el buffer apropiado
-

Decodificación de dispositivos de E/S

¿Cómo se realiza la lectura de puertos con el 8086?.

- De acuerdo al diagrama de tiempos del ciclo de bus. Para leer los dispositivos de E/S el 8086 realiza los siguientes pasos:
 - 1. Coloca una dirección de 16 bits en el bus de direcciones
 - 2. Coloca la señal M/\overline{IO}' en 0 para indicar que se trata de una dirección de dispositivos de E/S.
 - 3. Coloca la señal \overline{RD}' en 0 para indicar que se requiere una operación de lectura.
-

Decodificación de puertos de E/S

- Conceptos básicos en la interfaz de puertos de E/S.
 - La función principal de la interfaz de E/S es la de permitir al procesador leer o escribir en un registro dado del dispositivo de E/S. Para realizar esta operación, el microprocesador deberá.
 - 1.Ser capaz de activar el circuito integrado
 - 2.Identificar el registro
 - 3.Habilitar el buffer apropiado
-

Decodificación de puertos de E/S

Ser capaz de activar el circuito integrado

- Para conectar un dispositivo de E/S con el microprocesador, se requiere decodificar la dirección enviada al dispositivo de E/S. Esta decodificación provoca que el dispositivo de E/S funcione en una sección única del mapa de E/S, por ejemplo de la dirección F000H a la F002H. Y en cualquier otra dirección fuera de este rango no se activará.
-

Decodificación de la Memoria

Es importante recordar que el 8086 utiliza las señales BHE' (habilitación del bus alto) y Ao (bit de dirección Ao) para seleccionar uno o ambos bancos de puertos de acuerdo al tamaño de la instrucción que se está ejecutando para la transferencia de datos fuera del procesador y considerando la siguiente tabla.

<u>BHE'</u>	<u>Ao</u>	<u>Función</u>
0	0	Habilitación de ambos bancos 16 bits (D15-D0)
0	1	Habilitación banco alto 8 bits dirección non (D15-D8)
1	0	Habilitación banco bajo 8 bits dirección par (D7-D0)
1	1	No se habilita ningún banco

RESUMEN DE INTERFAZ DE PUERTOS DE E/S

1. Conectar las líneas del bus de dirección requeridas a las líneas de direcciones del circuito integrado de puertos.
 2. Decodificar las líneas de direcciones fijas junto con la señal M/IO' , a través de una compuerta NAND o con decodificador.
 3. La señal BHE' se utiliza para habilitar el banco alto de puertos de E/S, en la entrada CS' junto con la decodificación del punto anterior.
 4. La señal A_0 se utiliza para habilitar el banco bajo de puertos de E/S, en la entrada CS' junto con la decodificación del punto dos.
-

Mapeo de dispositivos de E/S

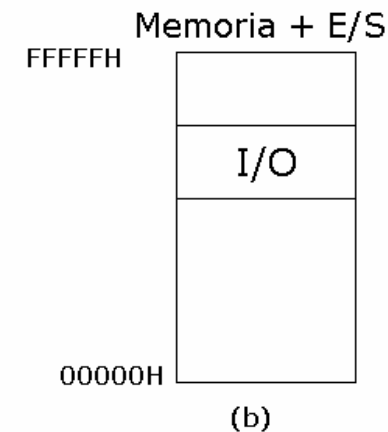
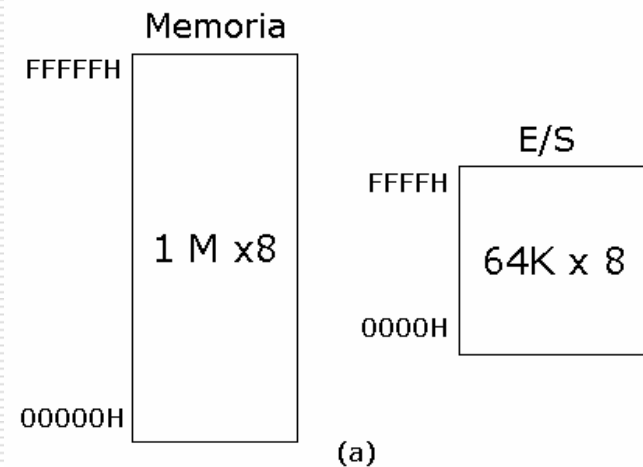
- Mapa de Memoria y de E/S del 8086

a) Mapeo Aislado

Decodificar 16 líneas de direcciones, $M/\overline{IO}'=0$ usar instrucciones IN y OUT

b) Mapeo de dispositivos de E/S en Memoria

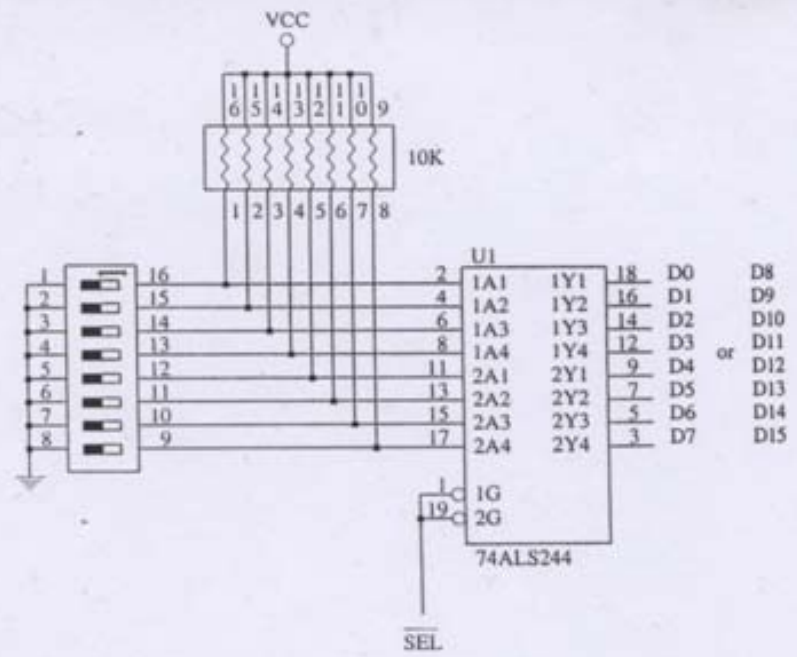
Decodificar 20 líneas de direcciones, $M/\overline{IO}'=1$ usar instrucciones MOV para leer y escribir en los puertos.



Mapeo de dispositivos de E/S

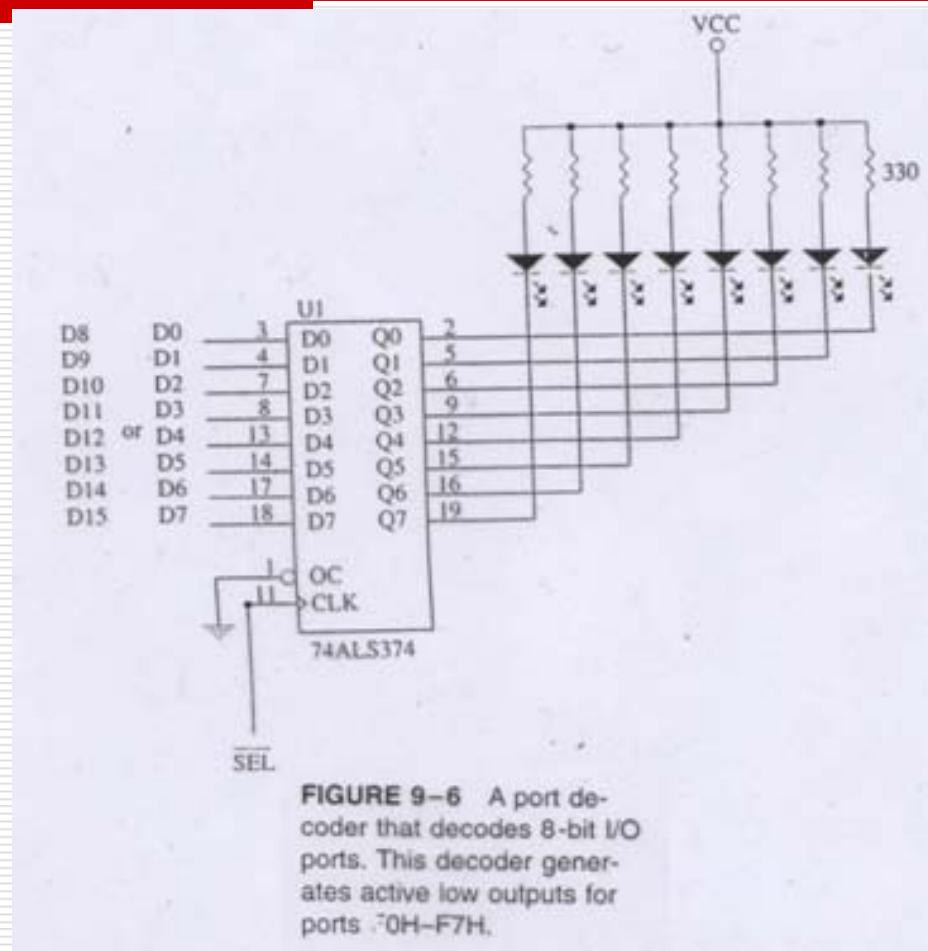
□ Puerto de Entrada

FIGURE 9-3 The basic input interface illustrating the connection of eight switches. Note that the 74ALS244 is a three-state buffer that controls the application of the switch data to the data bus.



Mapeo de dispositivos de E/S

- Puerto de salida



COMUNICACIÓN Serie/Paralelo

□ Paralelo

- Transmite los datos a través de n líneas de datos
 - n depende del tamaño de datos que se manejen: 8 bits, 16 bits, 32 bits
 - Más rápido.
 - En cortas distancias resulta más efectivo
 - Los datos a transmitir no necesitan pre-tratamiento
 - A largas distancias resulta más costoso por la mayor disposición a generar errores
-

COMUNICACIÓN Serie/Paralelo

□ Serie

- Transmiten los datos a través de 1 única línea de datos con independencia del formato. La transmisión de bits es uno detrás de otro sobre la misma línea.
 - Mucho menos costoso
 - número reducido de líneas
 - Menor disposición a errores
 - Los datos necesitan ser serializados/deserializados
 - Se requiere un protocolo de transmisión
-

Sistemas Digitales con Microprocesadores

112135

Tema 8 Tema 8. Comunicación Paralela y Serial.

Interfaz Paralela

1.1 Puerto Paralelo 8255A

Interfaz Serie.

2.1 Conceptos de transmisión serial

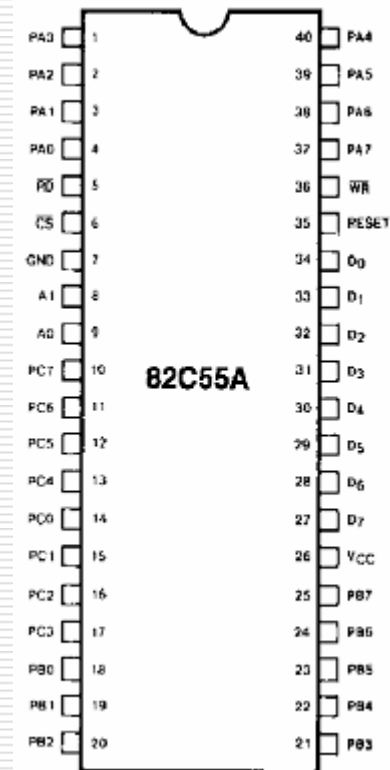
2.2 UART 8250

2.3 USART 8251A

82C55A

- Interfaz E/S de propósito general que conecta periféricos con el bus del microprocesador
- Realiza tareas de "buffering" y "latching"
- Programable por software
- Utilizado como interfaz de teclado, puerto paralelo.

Pin Names	
D ₇ -D ₀	Data Bus (Bi-Directional)
RESET	Reset Input
\overline{CS}	Chip Select
\overline{RD}	Read Input
\overline{WR}	Write Input
A ₀ , A ₁	Port Address
PA ₇ -PA ₀	Port A (BIT)
PB ₇ -PB ₀	Port B (BIT)
PC ₇ -PC ₀	Port C (BIT)
V _{CC}	+ 5 Volts
GND	0 Volts



82C55A

Pin Names	
D ₇ -D ₀	Data Bus (Bi-Directional)
RESET	Reset Input
\overline{CS}	Chip Select
\overline{RD}	Read Input
\overline{WR}	Write Input
A ₀ , A ₁	Port Address
PA ₇ -PA ₀	Port A (BIT)
PB ₇ -PB ₀	Port B (BIT)
PC ₇ -PC ₀	Port C (BIT)
V _{CC}	+ 5 Volts
GND	0 Volts

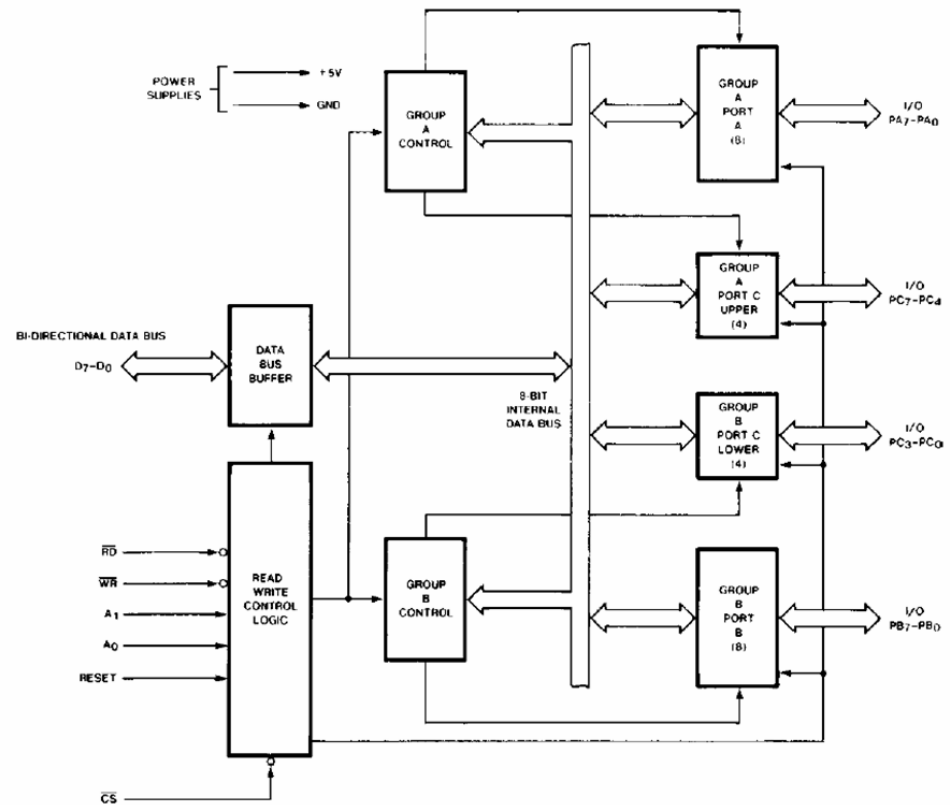


Figure 1. 82C55A Block Diagram

82C55A

- 24 pines E/S que se pueden programar de forma Independiente.
- 3 Modos de operación distintos

Modo 0. Las 8 líneas del puerto A, B y C se programan como entrada o salida. El puerto C se puede trabajar como 2 puertos de 4 bits de entrada o de salida.

Modo 1. Las 8 líneas del puerto A y/o B se programan como entrada o salida + 3 bits del puerto C para protocolo y control de interrupción

Modo 2 Bidireccional. El puerto A trabaja con 8 líneas bidireccionales + 5 bits del puerto C para protocolo e interrupción.

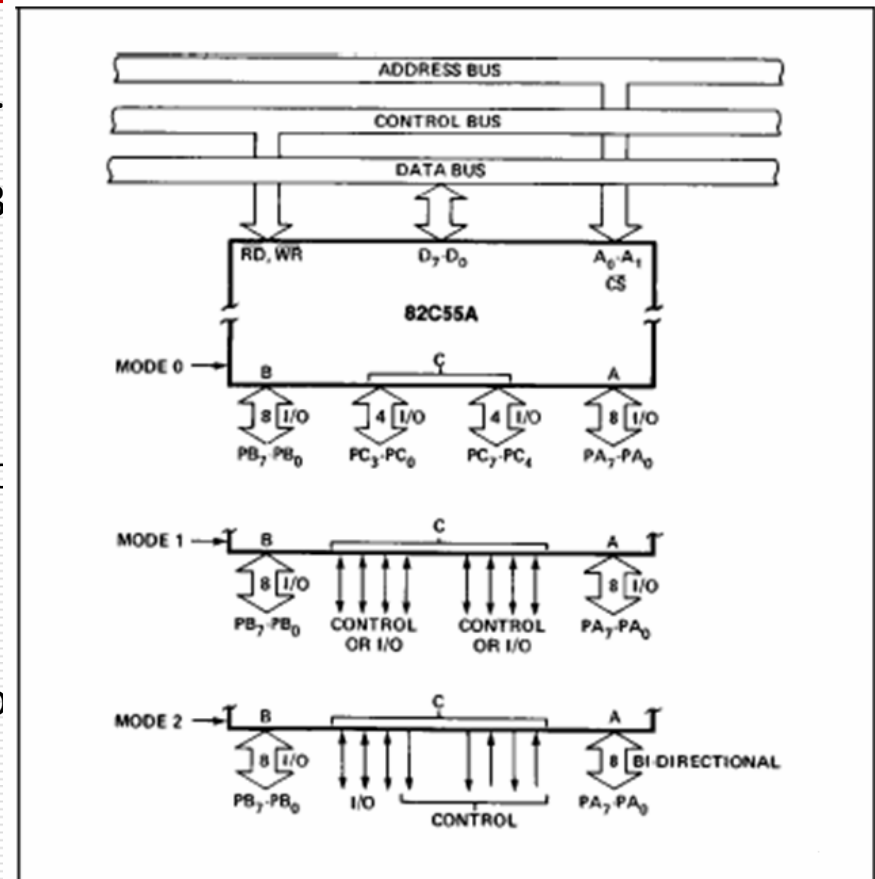


Figure 5. Basic Mode Definitions and Bus Interface

82C55A

□ Palabra de Programación

□ MOV AL, _____

□ OUT _____, AL

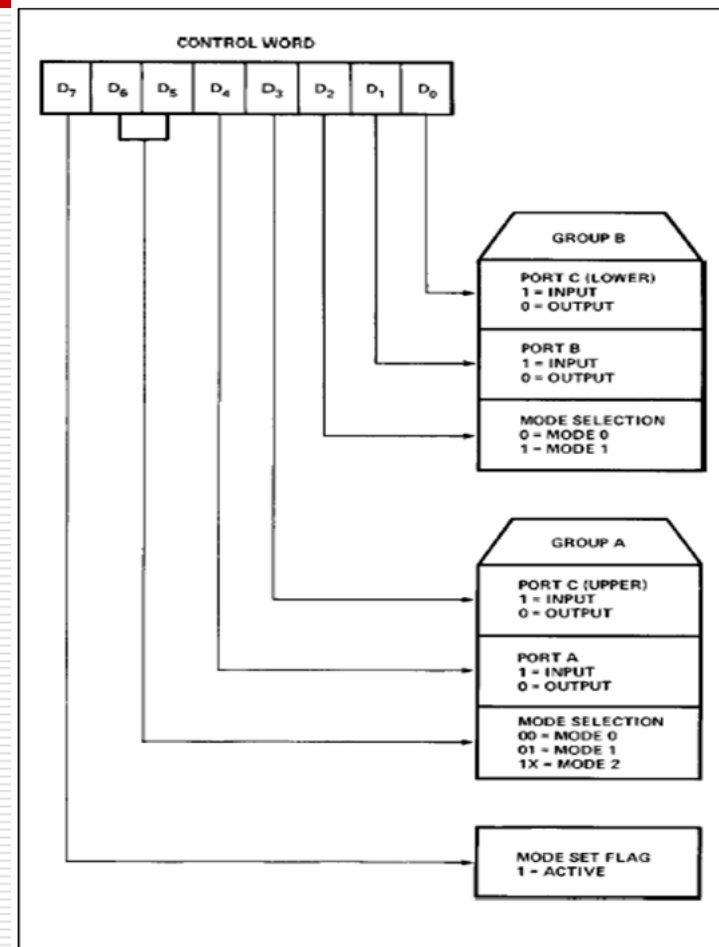
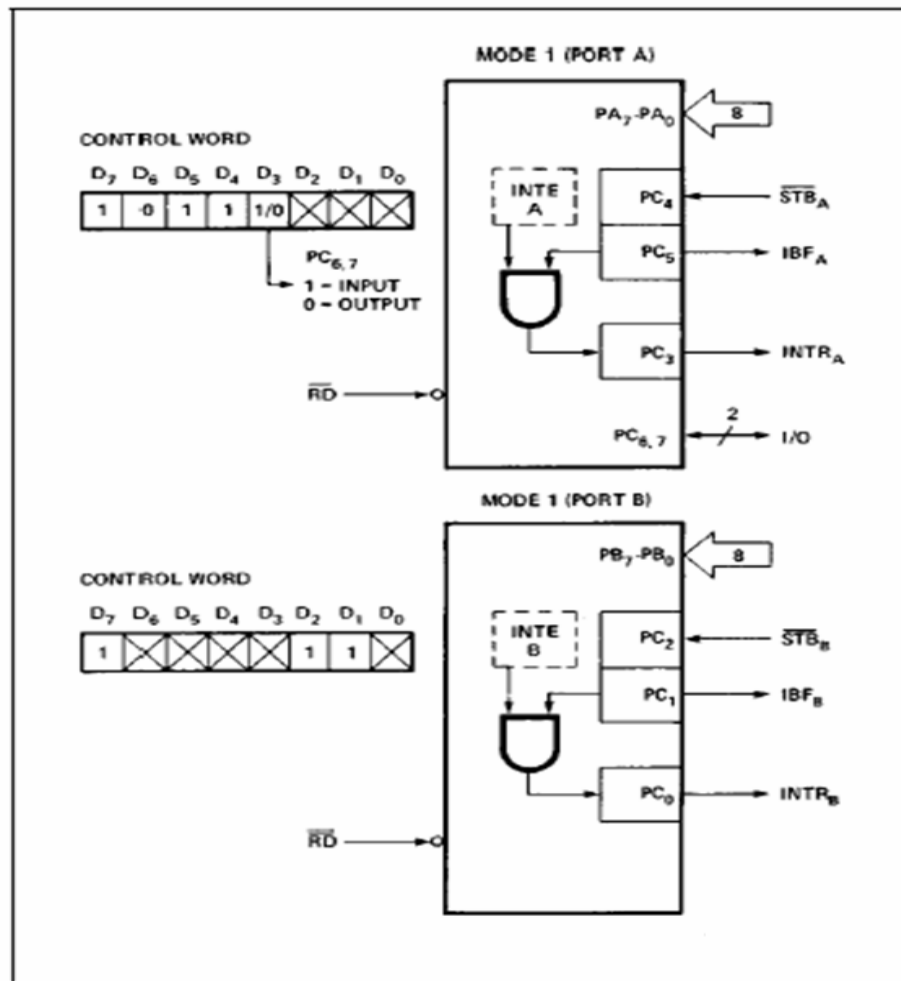


Figure 6. Mode Definition Format

82C55A

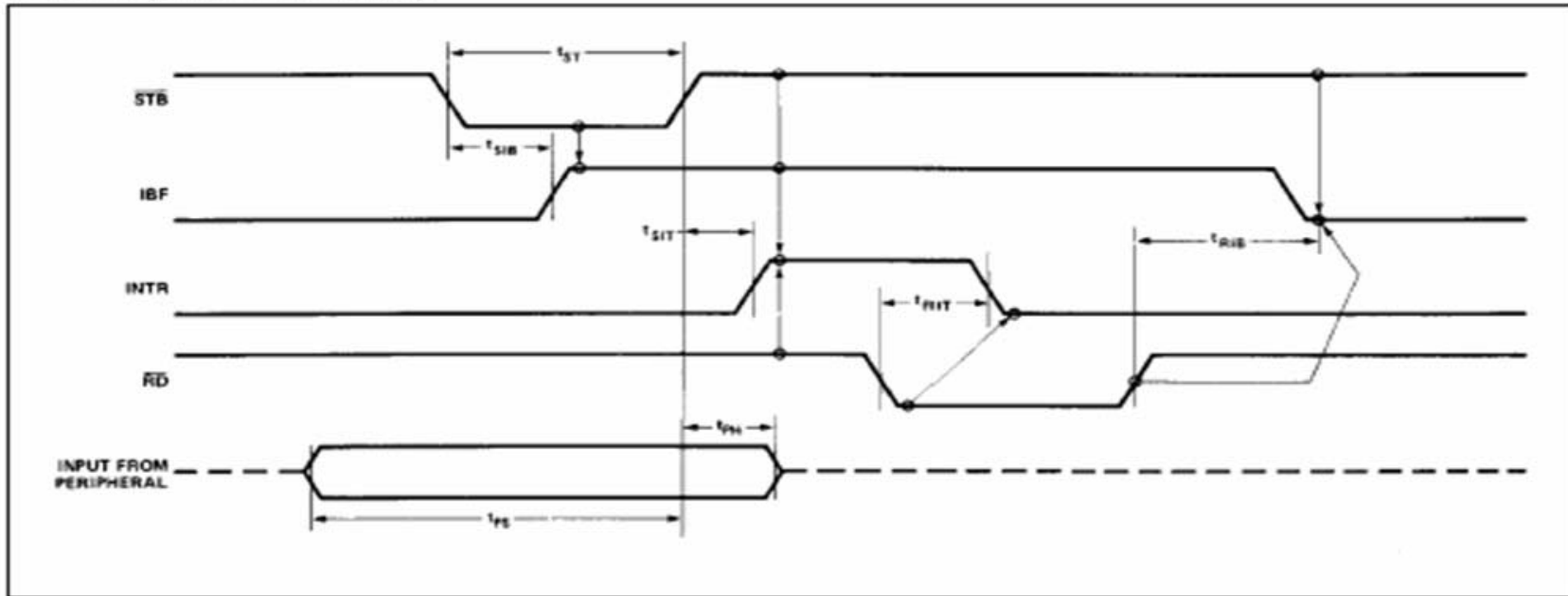


INTE A
Controlled by bit set/reset of PC₄.
INTE B
Controlled by bit set/reset of PC₂.

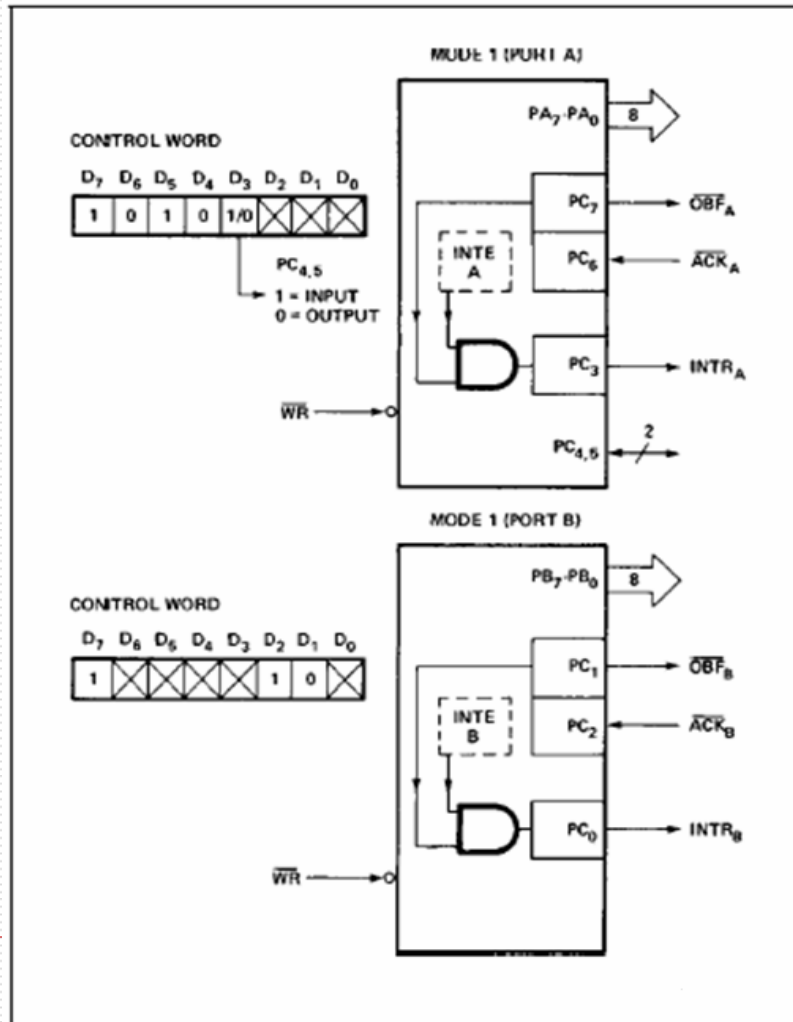
Figure 8. MODE 1 Input

82C55A

MODE 1 (STROBED INPUT)



82C55A



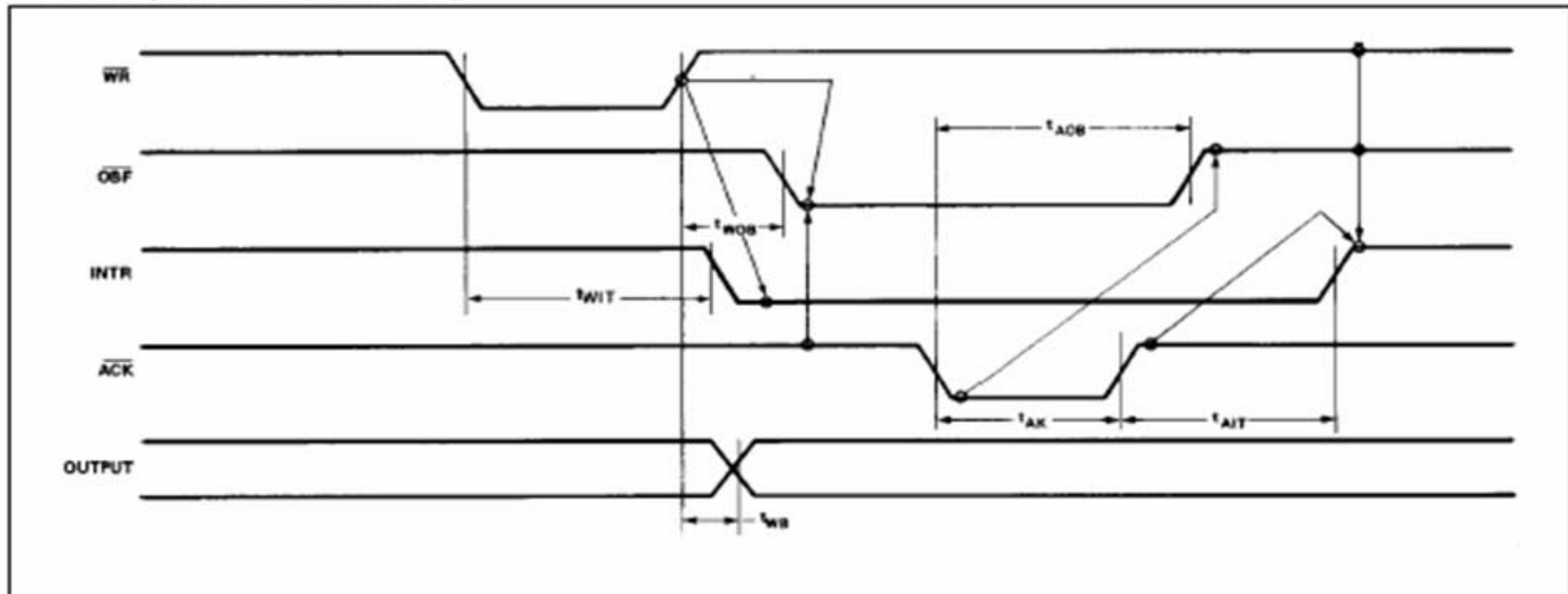
INTE A
Controlled by bit set/reset of PC₆.

INTE B
Controlled by bit set/reset of PC₂.

Figure 10. MODE 1 Output

82C55A

MODE 1 (STROBED OUTPUT)



82C55A

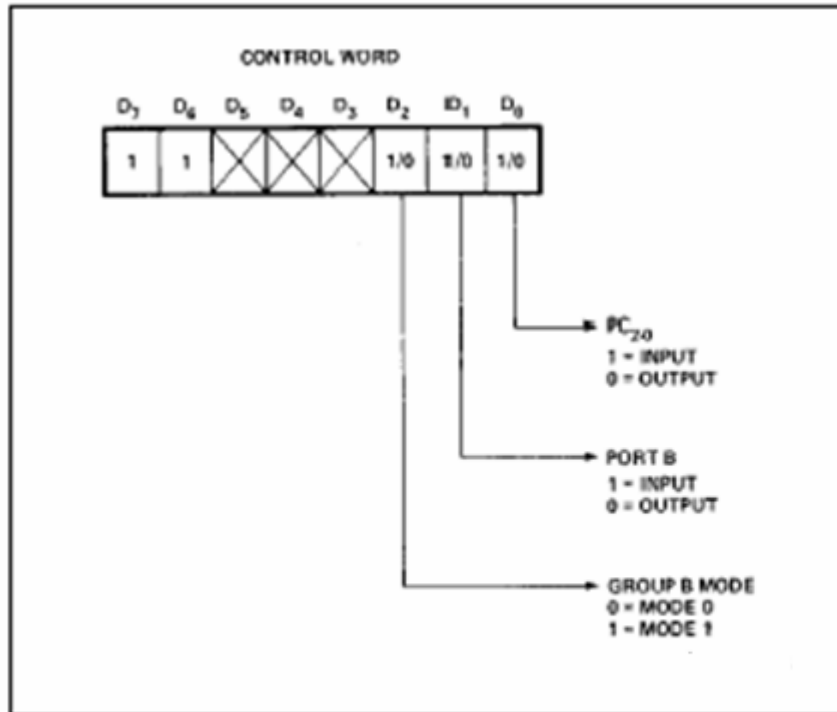


Figure 13. MODE Control Word

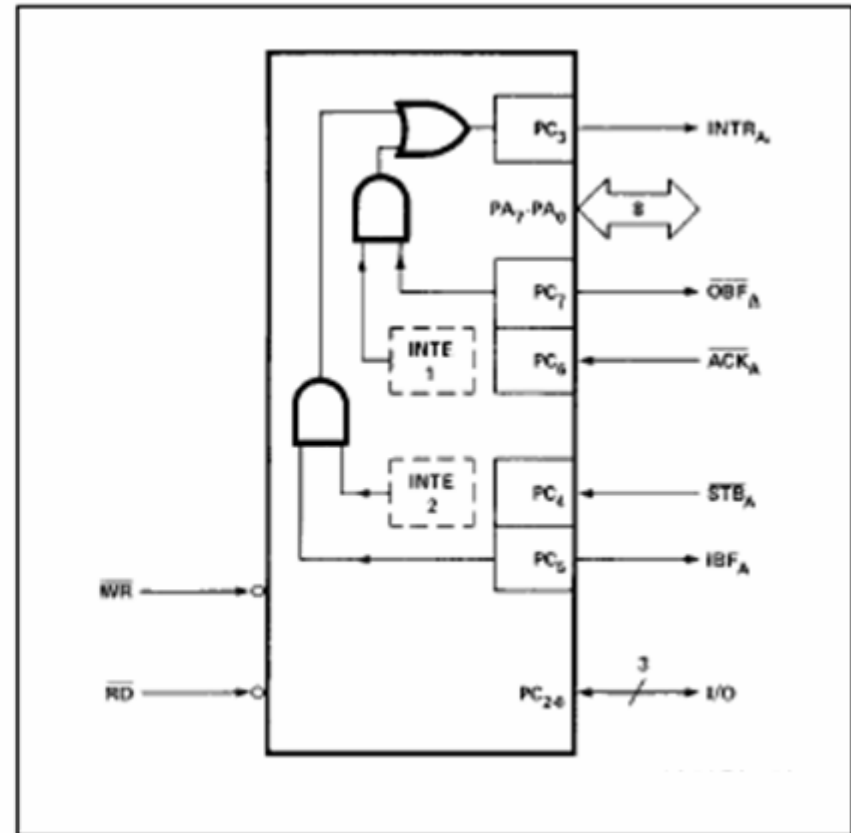


Figure 14. MODE 2

82C55A

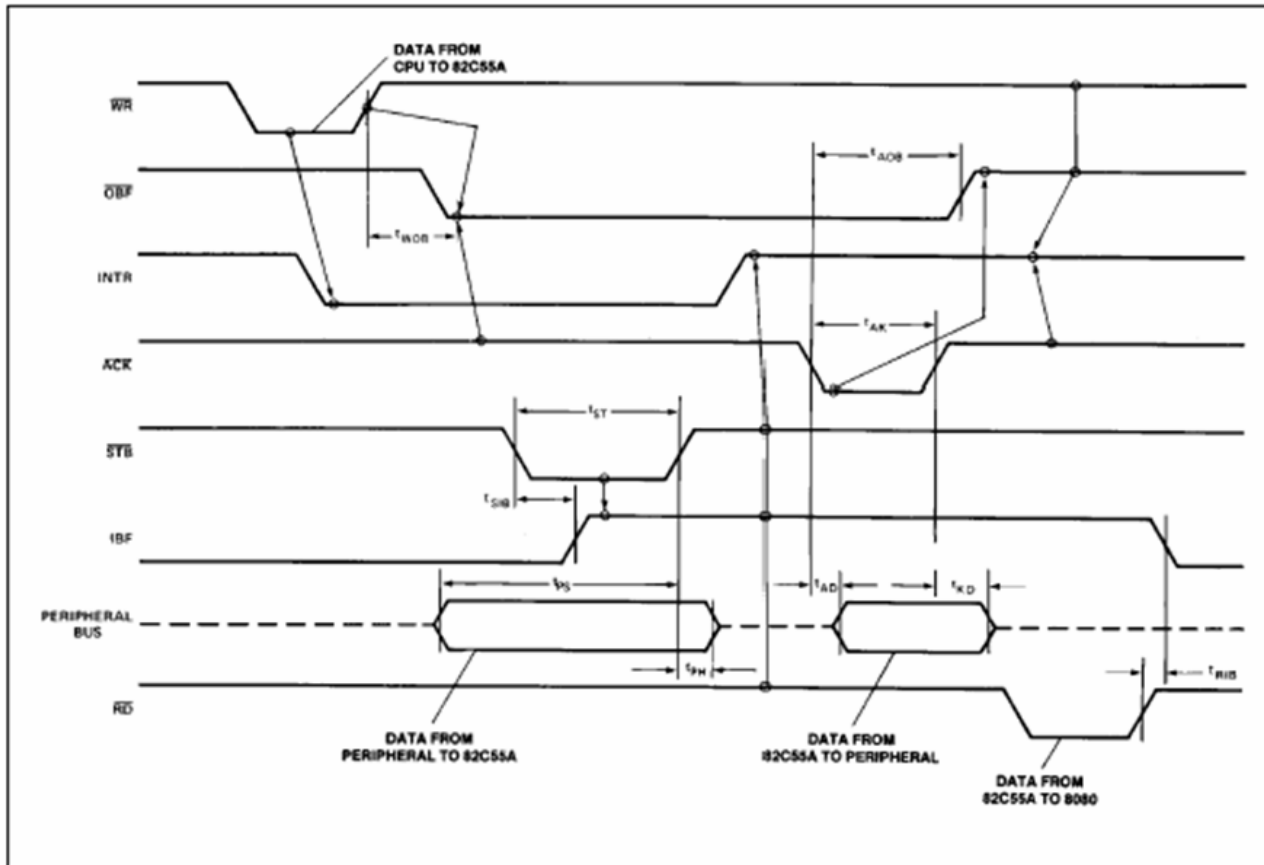


Figure 15. MODE 2 (Bidirectional)

Interrupt Enable Flag	Position	Alternate Port C Pin Signal (Mode)
INTE B	PC2	\overline{ACK}_B (Output Mode 1) or \overline{STB}_B (Input Mode 1)
INTE A2	PC4	\overline{STB}_A (Input Mode 1 or Mode 2)
INTE A1	PC6	\overline{ACK}_A (Output Mode 1 or Mode 2)

Figure 18. Interrupt Enable Flags in Modes 1 and 2

82C55A

Mode Definition Summary

	MODE 0	
	IN	OUT
PA ₀	IN	OUT
PA ₁	IN	OUT
PA ₂	IN	OUT
PA ₃	IN	OUT
PA ₄	IN	OUT
PA ₅	IN	OUT
PA ₆	IN	OUT
PA ₇	IN	OUT
PB ₀	IN	OUT
PB ₁	IN	OUT
PB ₂	IN	OUT
PB ₃	IN	OUT
PB ₄	IN	OUT
PB ₅	IN	OUT
PB ₆	IN	OUT
PB ₇	IN	OUT
PC ₀	IN	OUT
PC ₁	IN	OUT
PC ₂	IN	OUT
PC ₃	IN	OUT
PC ₄	IN	OUT
PC ₅	IN	OUT
PC ₆	IN	OUT
PC ₇	IN	OUT

MODE 1	
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
INTR _B	INTR _B
IBF _B	OBFB
STB _B	ACK _B
INTR _A	INTR _A
STB _A	I/O
IBF _A	I/O
I/O	ACK _A
I/O	OBFA

MODE 2	
GROUP A ONLY	
↔	
↔	
↔	
↔	
↔	
↔	
↔	
↔	
—	
—	
—	
—	
—	
—	
—	
I/O	
I/O	
I/O	
^{INT1} RA	
IBFA	
ACK _A	
OBFA	

MODE 0
OR MODE 1
ONLY

82C55A

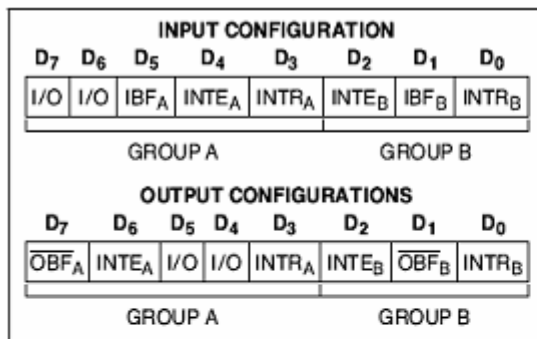


Figure 17a. MODE 1 Status Word Format

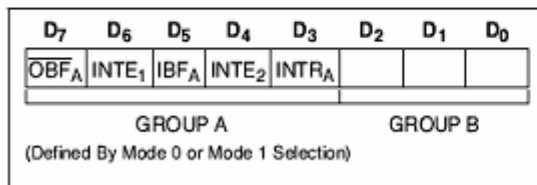


Figure MODE 2 Status Word Format

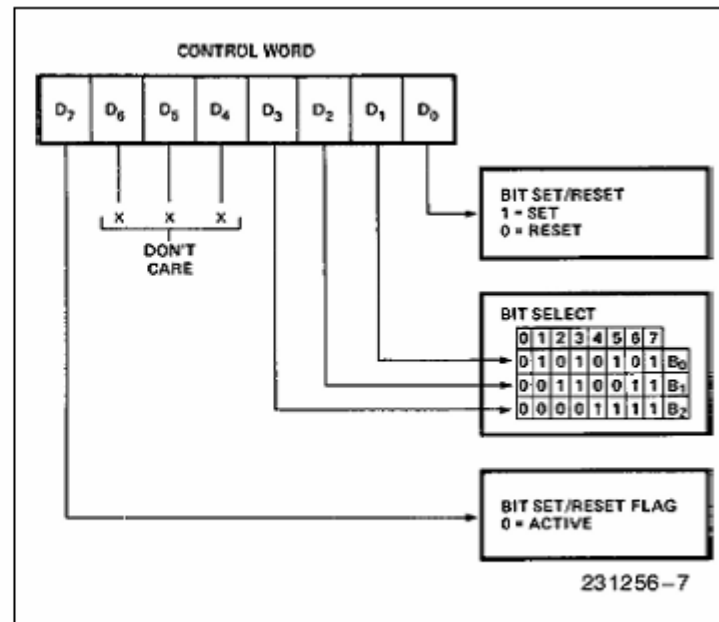


Figure Bit Set/Reset Format

COMUNICACIÓN Serie/Paralelo

□ Paralelo

- Transmite los datos a través de n líneas de datos
- n depende del tamaño de datos que se manejen: 8 bits, 16 bits, 32 bits
- Más rápido.
- En cortas distancias resulta más efectivo
- Los datos a transmitir no necesitan pre-tratamiento
- A largas distancias resulta más costoso por la mayor disposición a generar errores

□ Serie

- Transmiten los datos a través de 1 única línea de datos con independencia del formato. La transmisión de bits es uno detrás de otro sobre la misma línea.
 - Mucho menos costoso
 - número reducido de líneas
 - Menor disposición a errores
 - Los datos necesitan ser serializados/deserializados
 - Se requiere un protocolo de transmisión
-

CONCEPTOS BASICOS DE TRANSMISIÓN SERIAL

Los conceptos básicos concernientes al modo de E/S serial pueden clasificarse en las siguientes categorías.

- Requerimientos de interface.
 - Formatos de Entrada/Salida serial.
 - Chequeo de error en comunicación de datos.
 - Estándares en E/S serial
-

TRANSMISIÓN SÍNCRONA VS. TRANSMISIÓN ASÍNCRONA

- ❑ Las comunicaciones seriales ocurren en uno de dos formatos: **síncrono y asíncrono**.

Formato síncrono

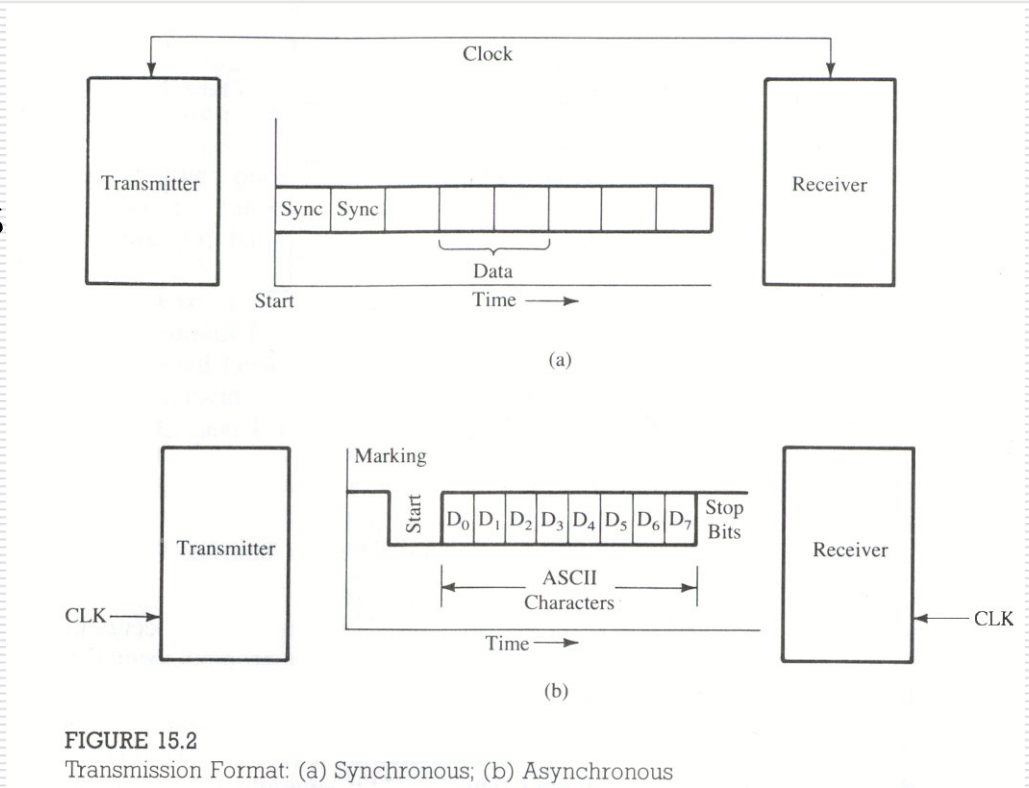
- ❑ En el formato síncrono, un receptor y un transmisor se sincroniza con el mismo reloj, y un bloque de caracteres se transmite después de caracteres de sincronía (SYNC).
- ❑ Este formato es generalmente usado para altas velocidades de transmisión (más de 20 K bits/seg).

Formato asíncrono

- ❑ El formato asíncrono es orientado a carácter. La transmisión asíncrona puede ocurrir en cualquier momento, es impredecible en relación al tiempo. Por lo tanto cada carácter debe contener información de cuando inicia la transmisión y cuando termina. Esta información se incluye en cada carácter agregando un bit de inicio y uno o dos bits de paro. Cuando no se transfiere información, el receptor permanece en estado alto. La transmisión empieza con un bit de inicio (en estado de cero) seguido por un carácter (de 5,6,7,8 bits) y 1 ó 2 bits de paro en estado alto. Esto se conoce como trama. En transmisión serial, el estado lógico "1" se conoce también como "Marca".
 - ❑ El formato asíncrono es generalmente usado en transmisiones a baja velocidad (menos de 20 Kbits/seg).
-

Formatos Serie

- ❑ **FORMATO Serie Síncrono**
- ❑ La comunicación síncrona exige al menos dos líneas: reloj, datos
- ❑ Mejora la tasa de transferencia porque reduce la relación de bits de protocolo/bits de datos.
- ❑ Los datos se transmiten por bloques
- ❑ Lo que distingue a los protocolos es los bytes de control que utilizan



Interfaces asíncronas

- ❑ Es de crucial importancia mantener la sincronización entre ambos comunicantes
- ❑ Puede ser necesario la introducción de métodos de codificación que aseguren la sincronización

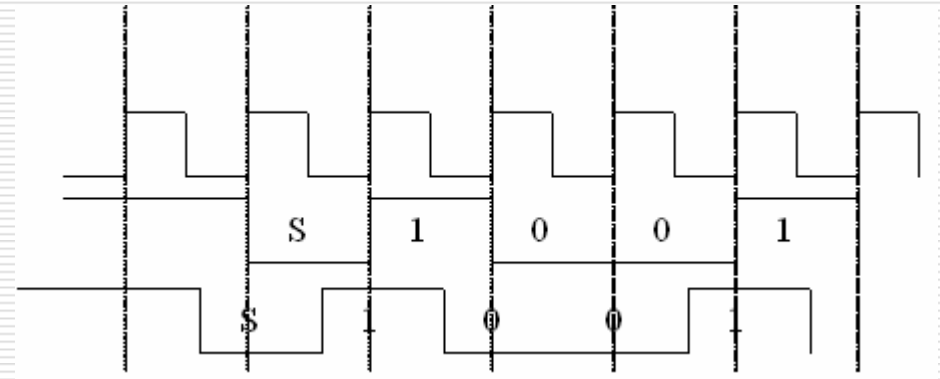


Tabla resumen de Formato síncrono vs. asíncrono

Resumen de Formatos Síncrono y Asíncrono de Comunicación Serial

Formato	Síncrono	Asíncrono
Formato de datos	Grupo de caracteres	Un caracter a la vez
Velocidad	Alta (20 K bits/segundo) o mayor)	20 K bits/segundo o menor
Información de la Trama	Caracteres de Sincronía se envían con cada grupo	Bits de inicio y paro presentes con cada caracter
Implementación	Hardware	Hardware y Software
Dirección de datos	Simplex, Half y Full duplex	Simplex, Half y Full duplex

Modos de Transmisión de Datos

Según el sentido y simultaneidad de la transmisión podemos encontrarnos con tres tipos diferentes:

- ❑ **Simplex:**
Este modo de transmisión permite que la información discorra en un solo sentido y de forma permanente, con esta fórmula es difícil la corrección de errores causados por deficiencias de línea. Como ejemplos de la vida diaria tenemos, [la televisión](#) y [la radio](#).
 - ❑ **Half Duplex:**
En este modo, la transmisión fluye en un único sentido a la vez, pero no de una manera permanente, pues el sentido puede cambiar. Como ejemplo tenemos los radios de banda civil.
 - ❑ **Full Duplex.**
Es el [método](#) de [comunicación](#) más aconsejable, puesto que en todo momento la comunicación puede ser en los dos sentidos posibles simultáneamente. El ejemplo típico sería el [teléfono](#).
-

VELOCIDAD DE TRANSMISION

- En transmisión serial el receptor debe activar la recepción de bits a la misma velocidad que el transmisor, de otra manera el receptor no sería capaz de detectar la diferencia entre dos 0's o 1's consecutivos.
- La velocidad a la cual los bits son transmitidos (bits/seg) se conoce como "baud", sin embargo, técnicamente esta unidad se define como el número de cambios de señal /seg.

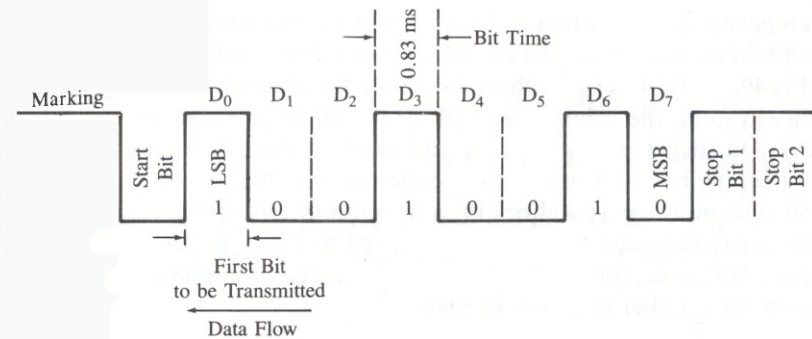


FIGURE 15.3
Serial Bit Format for ASCII Character "I" at 1,200 Baud

$$1 \text{ s} / 1200 \text{ bits} = 0.83 \text{ ms por c/bit}$$

Esquema de codificación:

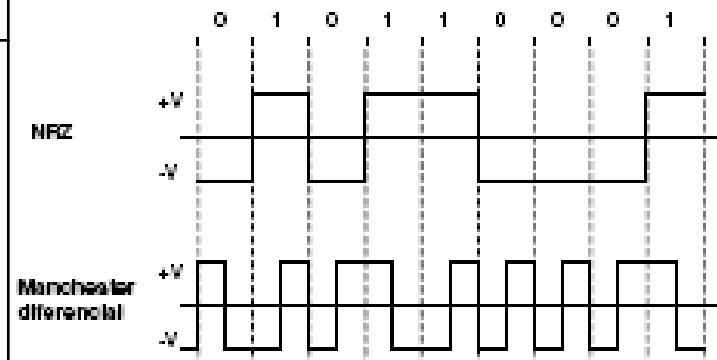
Es la correspondencia entre los elementos de datos (bits) y los elementos de señal (símbolos). El esquema de codificación puede mejorar las presentaciones de la transmisión.

Un criterio es la sincronización que permite determinar el principio y el fin de cada bit. Lo cual se puede lograr al transmitir una señal de reloj por separado, pero resulta costoso.

La sincronización puede ir incluida en el propio esquema de codificación. Ejemplo codificación Manchester.

Esquema de codificación:

Codificación Manchester	Codificación NRZ (no retorno a cero)
<ul style="list-style-type: none">• Transición en medio de cada periodo• La transición sirve para la sincronía.• La transición de un nivel bajo a uno alto representa un uno• La transición de un nivel alto a uno bajo representa un 0	<ul style="list-style-type: none">• Dos voltajes distintos para 0 y 1• Voltaje constante para el intervalo• Unipolar: 0 volts para 0 lógico y V volts para 1 lógico.• Bipolar: -V volts para 0, +V volts para 1



Estándares en Transmisión serial

□ ¿Qué es RS-232?

RS-232 (Estándar ANSI/EIA-232) es el conector serial de las PCs IBM y compatibles. Se ha utilizado para una gran variedad de propósitos, como conectar un ratón, impresora o módem, así como instrumentación de medición. Gracias a las mejoras que se han ido desarrollando en las líneas de transmisión y en los cables, existen aplicaciones en las que se aumenta el desempeño de RS-232 en lo que respecta a la distancia y velocidad del estándar. RS-232 está limitado a comunicaciones de punto a punto entre los dispositivos y el puerto serial de la computadora. El hardware de RS-232 se puede utilizar para comunicaciones seriales en distancias de hasta 50 pies.

TABLE 15.2
Comparison of Serial I/O Standards

Specifications	RS-232C	RS-422A	RS-423A
Speed	20 kbaud	10 Mbaud at 40 ft 100 kbaud at 4000 ft	100 kbaud at 30 ft 1 kbaud at 4000 ft
Distance	50 ft	4000 ft	4000 ft
Logic 0	> +3 to +25 V	B > A*	+4 to +6 V
Logic 1	< -3 to -25 V	B < A	-4 to -6 V
Receiver Input	±15 V	±7 V	±12 V

*B and A are differential input to the op amp.

□ Qué es RS-422?

RS-422 (Estándar EIA RS-422-A) es el conector serial utilizado en las computadoras Apple de Macintosh. RS-422 usa señales eléctricas diferenciales, en comparación con señales referenciadas a tierra como en RS-232. La transmisión diferencial, que utiliza dos líneas para transmitir y recibir, tiene la ventaja que es más inmune al ruido y puede lograr mayores distancias que RS-232. La inmunidad al ruido y la distancia son dos puntos clave para ambientes y aplicaciones industriales.

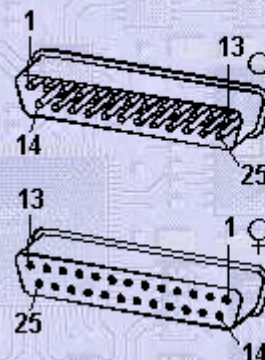
Estándar RS-232

Numero	de Pin	Señal	Descripción	E/S
En DB-25	En DB-9			
1	1	-	Masa chasis	-
2	3	TxD	Transmit Data	S
3	2	RxD	Receive Data	E
4	7	RTS	Request To Send	S
5	8	CTS	Clear To Send	E
6	6	DSR	Data Set Ready	E
7	5	SG	Signal Ground	-
8	1	CD/DCD	(Data) Carrier Detect	E
15	-	TxC(*)	Transmit Clock	S
17	-	RxC(*)	Receive Clock	E
20	4	DTR	Data Terminal Ready	S
22	9	RI	Ring Indicator	E
24	-	RTxC(*)	Transmit/Receive Clock	S

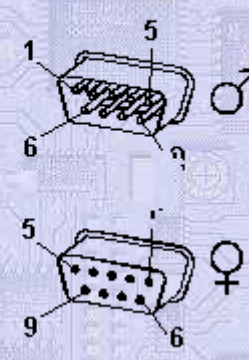
Pin Función

TXD (Transmitir Datos)
 RXD (Recibir Datos)
 DTR (Terminal de Datos Listo)
 DSR (Equipo de Datos Listo)
 RTS (Solicitud de Envío)
 CTS (Libre para Envío)
 DCD (Detección de Portadora)

Conector DB 25



Conector DB 9

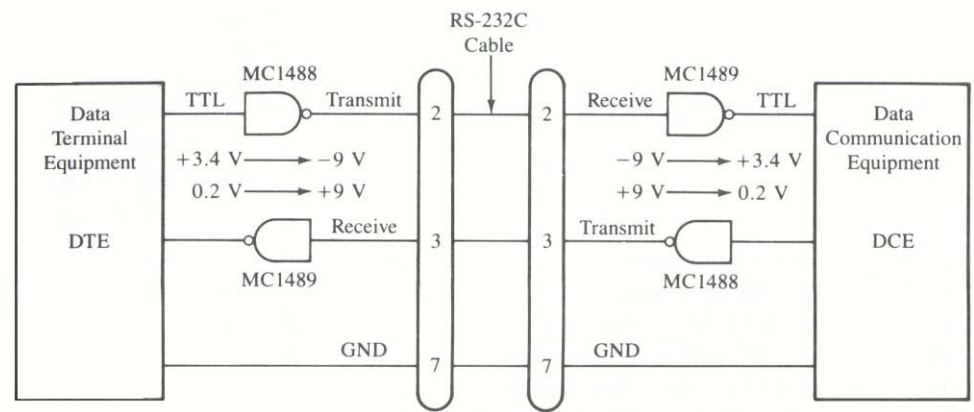


Estándar RS-232

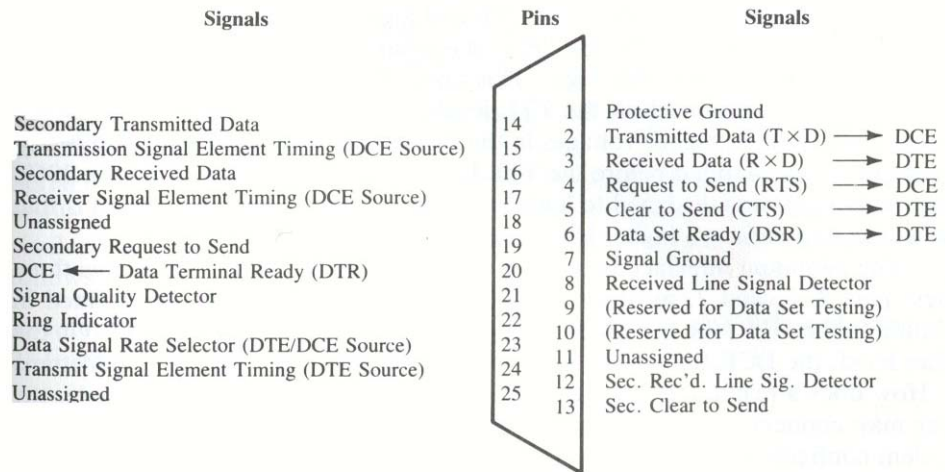
- El método de comunicación usado por RS-232 requiere de una conexión muy simple, utilizando sólo tres líneas: Tx, Rx, y GND.
 - Formas más populares de *handshaking* o intercambio de pulsos de sincronización con RS-232: handshaking for software y handshaking por hardware.
 - **Handshaking por software:** Esta forma de sincronización utiliza bytes de datos como caracteres de control. Las líneas necesarias para la comunicación siguen siendo Tx, Rx, y GND, ya que los caracteres de control se envían a través de las líneas de transmisión como si fueran datos.
 - **Handshaking por hardware:** Utiliza líneas de hardware. De manera similar a las líneas Tx y Rx, las líneas RTS/CTS y DTR/DSR trabajan de manera conjunta siendo un par la entrada y el otro par la salida. El primer par de líneas es RTS (*Request to Send*) y CTS (*Clear to Send*). Cuando el receptor está listo para recibir datos, cambia la línea RTS a estado alto; este valor será leído por el transmisor en la línea CTS, indicando que está libre para enviar datos. El siguiente par de líneas es DTR (*Data Terminal Ready*) y DSR (*Data Set Ready*). Estas líneas se utilizan principalmente para comunicación por módem, permiten al puerto serial y módem indicarse mutuamente su estado. Por ejemplo, cuando el módem se encuentra preparado para que la PC envíe datos, cambia la línea DTR a estado alto indicando que se ha realizado una conexión por la línea de teléfono. Este valor se lee a través de la línea DSR y la PC comienza a enviar datos. Como regla general, las líneas DTR/DSR se utilizan para indicar que el sistema está listo para la comunicación, mientras que las líneas RTS/CTS se utilizan para paquetes individuales de datos.
-

RS-232

Acoplamiento de voltajes



(a)



(b)

FIGURE 15.5

(a) Minimum Configuration of RS-232C Signals and Voltage Levels; (b) RS-232C Signal Definitions and Pin Assignments

SOURCE: Courtesy of Electronic Industries Association.

TRANSMISION SERIAL

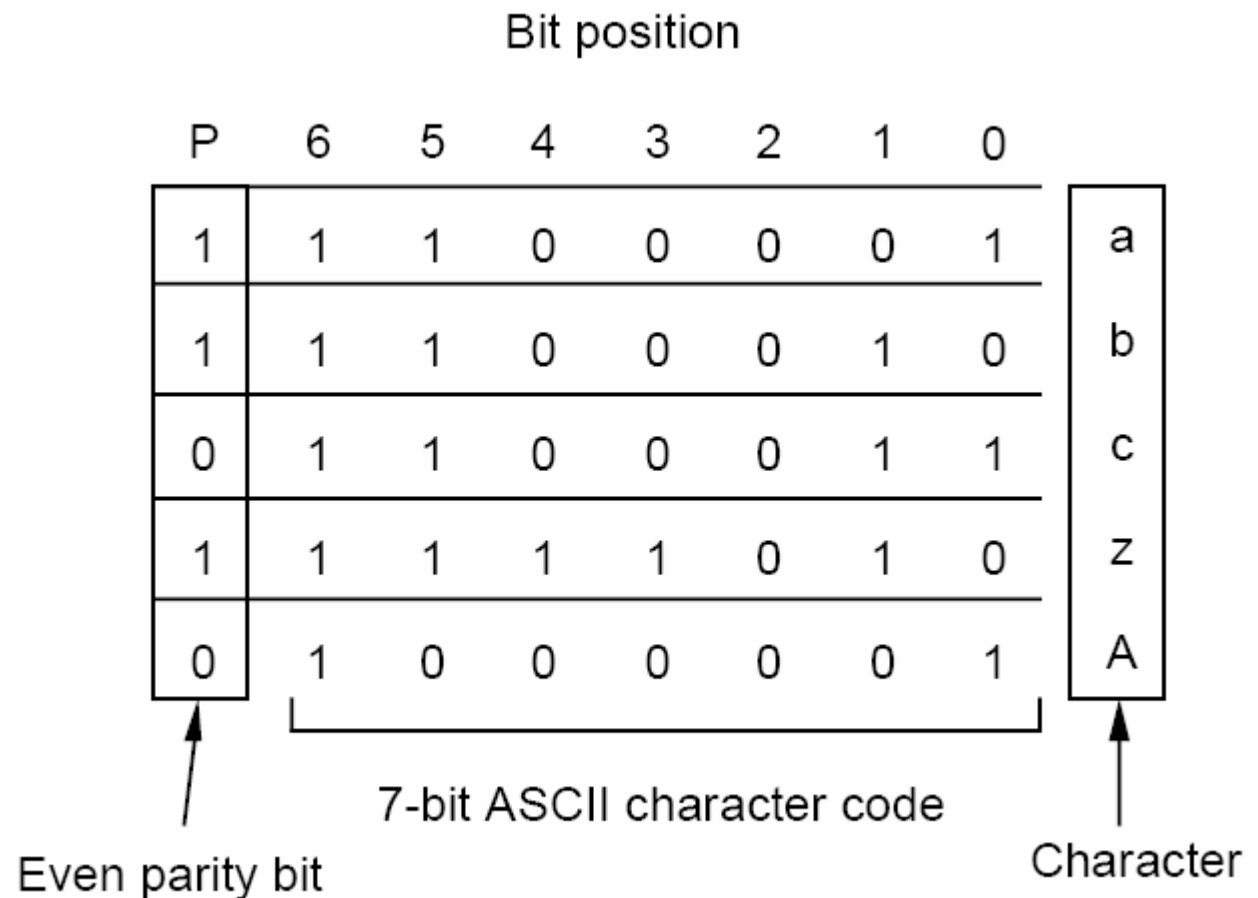
Chequeo de Errores

- Paridad.
 - Checksum
 - Chequeo de Redundancia Cíclica (CRC)
-

Chequeo de errores: PARIDAD

□ Paridad.

Se añade un bit a los bits de datos, llamado bit de paridad que se elige de forma tal que el número total de 1 sea siempre par o impar de acuerdo a la paridad seleccionada.



Chequeo de errores: Checksum

- ❑ La **paridad horizontal y vertical** es utilizada en algunos [códigos de bloque](#) para una combinación de chequeo de redundancia vertical (VRC) combinada con chequeo de redundancia longitudinal u horizontal (LRC) para [detectar errores](#).
- ❑ En este chequeo de errores, un byte llamado **checksum** se agrega al final del bloque de datos que será transmitido, este byte se compone de los bits de paridad columna por columna del bloque de datos.
- ❑ Este byte es calculado y transmitido por la máquina transmisora y recalculado y comparado por la máquina receptora. En caso de detectar algún error el receptor deberá solicitar una retransmisión

<i>P</i>	<i>Code</i>	<i>Character</i>
0	1 0 0 0 0 0 1	A
0	1 0 0 0 0 1 0	B
1	1 0 0 0 0 1 1	C
0	1 0 0 0 1 0 0	D
1	1 0 0 0 1 0 1	E
1	1 0 0 0 1 1 0	F
0	1 0 0 0 1 1 1	G
0	1 0 0 1 0 0 0	H
1	0 0 0 1 0 0 0	Checksum

Chequeo de errores: Checksum

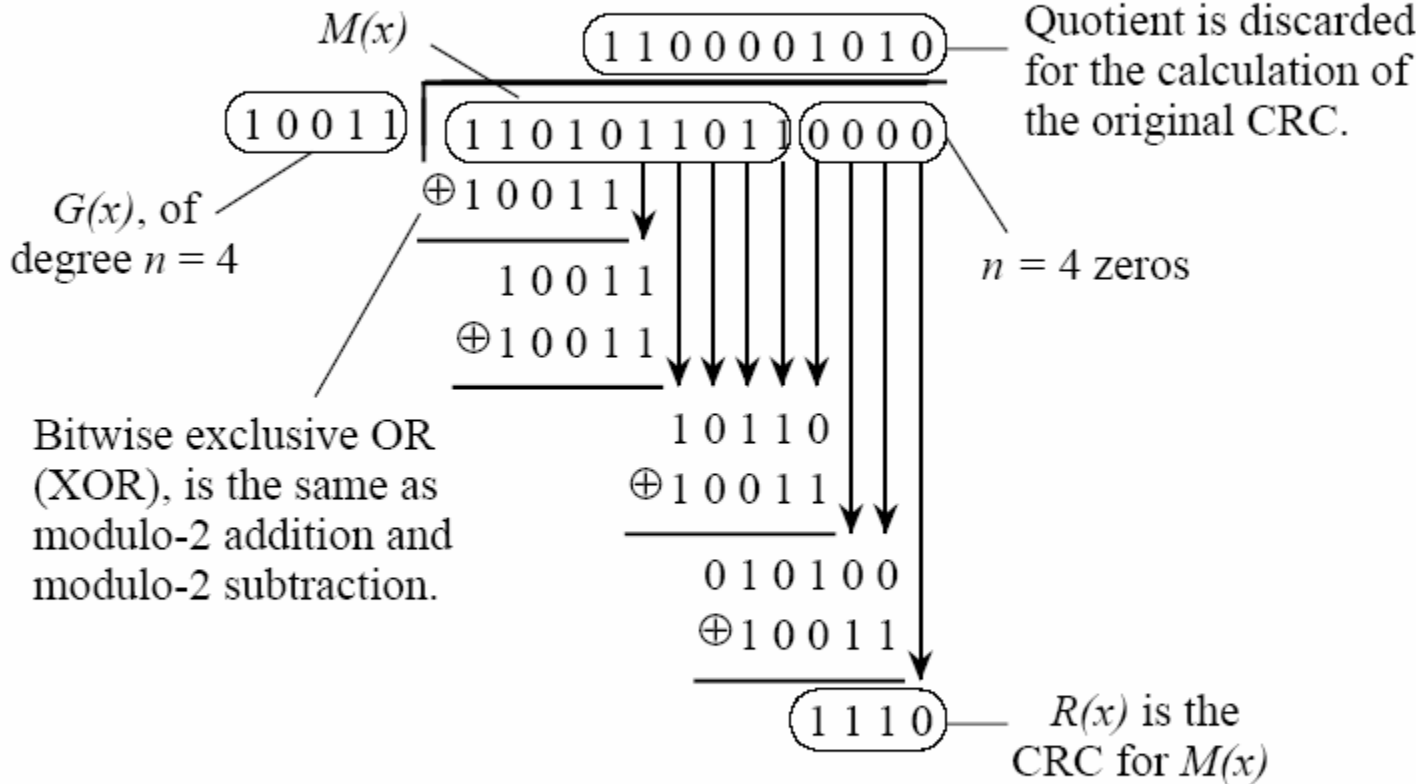
Proceso

El proceso para calcular la [paridad](#) de bloque es el siguiente:

- ❑ Los caracteres a transmitir se agrupan en bloques de n filas y m columnas
 - ❑ Se calcula el [bit](#) de [paridad](#) de cada fila y se añade al principio (o al final, según convenio) de la fila
 - ❑ Se calcula el [bit](#) de paridad de cada columna y se añade al principio (o al final, según convenio) de la columna
 - ❑ El bloque final a transmitir tendrá por tanto una fila y una columna más que el original.
 - ❑ La nueva columna estará formada por los bits de paridad horizontal de todas las filas
 - ❑ La nueva fila estará formada por los bits de paridad vertical de todas las columnas.
 - ❑ Adicionalmente se emplea un bit de paridad cruzada que se calcula a partir de los bits de paridad de filas y columnas.
 - ❑ Los chequeos de paridad horizontal y vertical se usan para detectar y corregir los posibles errores que se puedan producir durante la transmisión de datos.
-

Chequeo de errores:

Cyclic Redundancy Checking



Transmitted frame $T(x) = \underbrace{1101011011}_{M(x)} \underbrace{1110}_{R(x)}$

Chequeo de errores: Chequeo de redundancia cíclica

La técnica se basa en relaciones matemáticas de polinomios, se utiliza en comunicación de datos síncrona. Un conjunto de datos puede representarse como un polinomio el cual es dividido por un polinomio constante conocido como polinomio generador. El residuo de la división (CRC) se envía junto con el conjunto de datos como chequeo de error. El receptor realiza la división del polinomio (datos mas el CRC) entre el polinomio generador y verifica el residuo para detectar errores en la transmisión conforme al siguiente procedimiento.

1. Un conjunto de bits de datos se puede representar como:

$$M(x) = b_n X^0 + b_{n-1} X^1 + \dots + b_0 X^n, \quad \text{donde}$$

b_0 = bit menos significativo

b_n = bit más significativo

por ejemplo, el polinomio del número hexadecimal 8AH (10001010) es

$$M(x) = 1X^0 + 0X^1 + 0X^2 + 0X^3 + 1X^4 + 0X^5 + 1X^6 + 0X^7$$

$$M(x) = 1X^0 + 1X^4 + 1X^6 = X^6 + X^4 + 1$$

2. Asumiendo que la longitud del código CRC es de cuatro bits, en las transmisiones de las redes actuales el polinomio es de grado 16. Para obtener la división, primero, el polinomio se multiplica por el grado del polinomio generador (en el ejemplo x^4 , ver paso 3) y, posteriormente se divide entre el polinomio generador $G(x)$. La formula es:

$$\frac{M(x) * X^4}{G(x)} = Q(x) + R(x)$$

Donde $Q(x)$ es el cociente obtenido por la aritmética de módulo 2 y $R(x)$ es el residuo.

Chequeo de errores: Chequeo de redundancia cíclica

2. Asumiendo que la longitud del código CRC es de grado 4, en las transmisiones de las redes actuales el polinomio es de grado 16. Para obtener la división, primero, el polinomio se multiplica por el grado del polinomio generador (en el ejemplo x^4 , ver paso 3) y, posteriormente se divide entre el polinomio generador $G(x)$. La formula es:

$$\frac{M(x) * X^4}{G(x)} = Q(x) + R(x)$$

Donde $Q(x)$ es el cociente obtenido por la aritmética de módulo 2 y $R(x)$ es el residuo.

3. Considerando como polinomio generador: $G(x) = X^4 + 1$, tenemos:

$$\frac{M(x) * X^4}{G(x)} = \frac{X^{10} + X^8 + X^4}{X^4 + 1} = (X^6 + X^4 + X^2) + X^2$$

cociente $Q(x)$ residuo $R(x)$

4. Este residuo se suma al polinomio modificado

$$M(x) * X^4 + R(x) = X^{10} + X^8 + X^4 + X^2 \text{ y es transmitido como:}$$

	X^{10}		X^8				X^4		X^2		
0	1	0	1	0	0	0	1	0	1	0	0
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11

Dato = 8AH

El flujo de bits de datos transmitido incluye el byte original 8AH en orden invertido, más los bits del CRC obtenido que se colocan al final.

Chequeo de errores: Chequeo de redundancia cíclica

5. El receptor divide el polinomio transmitido entre $G(x)$, y si el residuo es 0, esto indica que no existió error (divide $X^{10} + X^8 + X^4 + X^2$ entre $X^4 + 1$ y checa el resultado).

(Nota: el polinomio generador de la CCITT es

$$G(x) = x^{16} + x^{12} + x^5 + 1)$$

$$\begin{array}{r}
 X^4 + 1 \overline{) \begin{array}{l} X^6 + X^4 + X^2 \\ X^{10} + X^8 + X^4 \\ \hline X^{10} + X^6 \\ \hline X^8 + X^6 + X^4 \\ X^8 + X^4 \\ \hline X^6 \\ X^6 + X^2 \\ \hline X^2 \end{array} \\
 \hline
 \text{Residuo}
 \end{array}$$

El puerto serie: UART 8250 (Universal Asynchronous Receiver/Transmitter)

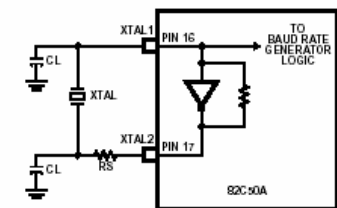
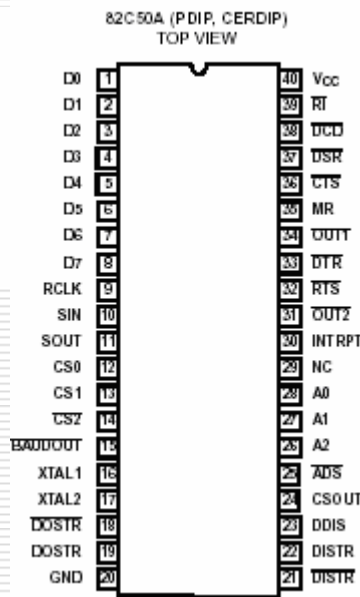
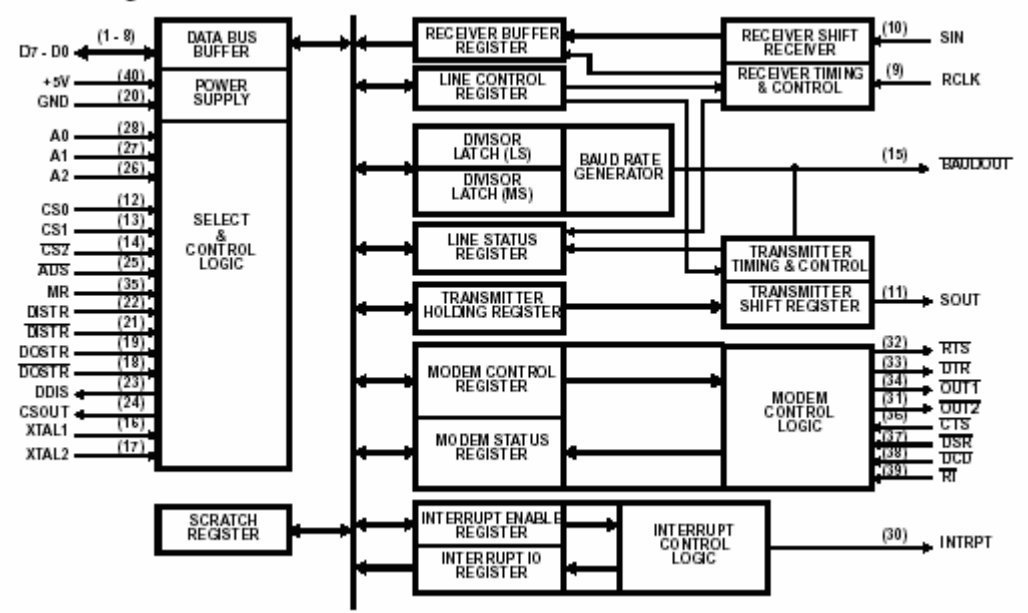


FIGURE 2. TYPICAL CRYSTAL OSCILLATOR CIRCUIT



Block Diagram



- El 8250 dispone de 11 registros pero sólo 3 líneas de dirección para seleccionarlos. Lo que permita distinguir unos de otros será, aparte de las líneas de direcciones, el sentido del acceso (en lectura o escritura) y el valor de un bit de uno de los registros: el bit DLAB del registro LCR, que es el bit 7 de dicho registro. Realmente, DLAB se emplea sólo puntualmente para poder acceder y programar los registros que almacenan el divisor de velocidad; el resto del tiempo, DLAB estará a 0 para acceder a otros registros.

El puerto serie: UART 8250 (Universal Asynchronous Receiver/Transmitter)

Terminales:

RCLK Receiver Clock Input. Debe ser igual a $(\text{Receiver Baud Rate}) * 16$

BAUDOUT' Baud Output. Salida del generador de rango de baud programable. Frecuencia = $\text{Baud Rate} * 16$

XIN External crystal Input. Se usa para el oscilador generador de Baud Rate.

XOUT External Crystal Output.

DOSTR y DOSTR' (WR) Write Line

DISTR, DISTR' (RD) Read Line

DDIS Driver Disable. Se pone en "0" cuando el CPU lee del UART

ADS' Address Strobe. Se usa si las señales no son estables durante los ciclos de lectura o escritura.

INTRPT Interrupt Output

MR Master Reset

DCD Data Carrier Detect

CSOUT Chip Select Out

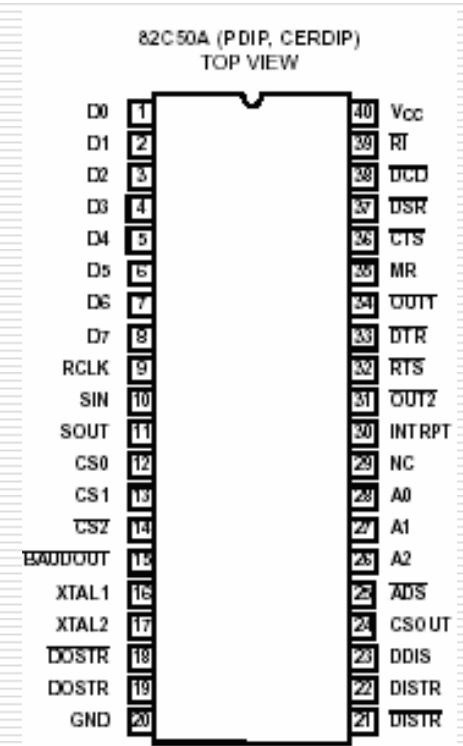
CS0-CS1,CS2' Chip Select

D0..D7: Data Bits 0..7: Bus triestado bidireccional de 8 líneas

SOUT: Serial Data Output: Salida de datos en serie.

SIN Serial Data Input: Entrada de datos en serie.

A2,A1,A0 Líneas de dirección para seleccionar Registro interno



SIGNIFICADO DE LAS LÍNEAS DEL 8250

- ❑ DISTR, DISTR': Data In Strobe. (RD) Read Line.
 - ❑ DOSTR: Data Out Strobe. (WR) Write Line.
 - ❑ D0..D7: Data Bits 0..7: Bus triestado bidireccional de 8 líneas para transmitir datos, información de control y de estado entre la CPU y el 8250. XTALx: Crystal/Clock: Conexiones para el cristal del cuarzo del BRG.
 - ❑ XTAL1, XTAL2 entrada de reloj externa.
 - ❑ SOUT: Serial Data Output: Salida de datos en serie del 8250. Una marca es un '1' y un espacio es un '0'. SOUT está en marca cuando el transmisor está inhibido, MR está a 1, el registro de transmisión está vacío o en el modo lazo (LOOP) del 8250. No es afectado por -CTS.
 - ❑ CTS': Clear To Send: Línea de entrada. El estado lógico de esta señal puede consultarse en el bit CTS del Modem Status Register (MSR) -como el bit CTS es el bit 4 del MSR se referencia MSR(4)-. Un cambio en el estado de -CTS desde la última lectura del MSR provoca que se active DCTS (bit MSR(0)). Cuando -CTS está activo (a 0) el modem indica que el dato en SOUT puede ser transmitido. -CTS no afecta al modo lazo (LOOP) del 8250.
 - ❑ DSR': Data Set Ready: Línea de entrada. El estado lógico de esta señal puede consultarse en MSR(5).
-

SIGNIFICADO DE LAS LÍNEAS DEL 8250

- ❑ DDSR (bit MSR(1)) indica si -DSR ha cambiado desde la última lectura del MSR. Cuando -DSR está activo el modem indica que está listo para intercambiar datos con el 8250; ello depende del estado del DCE (Data Communications Equipment) local y no implica que haya comunicación con la estación remota.
 - ❑ DTR':Data Terminal Ready. Línea de salida que puede activarse (poner a 0) escribiendo un 1 en MCR(0), y desactivarse escribiendo un 0 en dicho bit o ante la activación del pin MR. Con -DTR activo se indica al DCE que el 8250 puede recibir datos. En algunas circunstancias, esta señal se usa como LED de 'power on'. Si está inactivo, el DCE desconecta el modem del circuito de telecomunicaciones.
 - ❑ RTS':Request To Send. Línea de salida que habilita el modem. Se activa (poner a 0) escribiendo un 1 en MCR(1). Esta señal se pone en alto en respuesta a MR. -RTS indica al DCE que el 8250 tiene un dato listo para transmitir. En la modalidad half-duplex, esta señal se utiliza para controlar la dirección de la línea.
 - ❑ BAUDOUT':Esta línea de salida contiene una señal de reloj 16 veces mayor que la frecuencia usada para transmitir. Equivale a la frecuencia de entrada en el oscilador dividida por el BRG. La estación receptora podría emplear esta señal conectándola a RCLK (para compartir el mismo reloj).
 - ❑ OUTx': Estas dos salidas de propósito general se pueden activar (poner a 0) escribiendo un 1 en MCR(2) y MCR(3). Son desactivadas por la señal MR. En el modo lazo (LOOP o bucle), están también inactivas.
 - ❑ RI':Ring Indicator. Esta línea de entrada indica si el modem ha detectado que llaman por la línea y puede consultarse en MSR(6). El bit TERI (MSR(2)) indica si esta línea ha cambiado desde la última lectura del MSR. Si las interrupciones están habilitadas (IER(3) activo) esta patilla provoca una interrupción al activarse. -RI permanece activo durante el mismo intervalo de tiempo que la zona activa del ciclo de llamada e inactivo en los intervalos de la zona inactiva (o cuando el DCE no detecta la llamada). El circuito no se corta por culpa de -DTR.
-

El puerto serie: UART 8250

Registros

DLAB	A2	A1	A0	MODO	NOMBRE	Dir.PC	SIGNIFICADO
0	0	0	0	R	RBR	3F8H	Receiver Buffer Register (Registro buffer de recepción)
0	0	0	0	W	THR	3F8H	Transmitter Holding Register (Registro de retención de transmisión)
0	0	0	1	R/W	IER	3F9H	Interrupt Enable Register (Registro de habilitación de interrupciones)
X	0	1	0	R	IIR	3FAH	Interrupt Identification Register (Registro de identificación de interrupciones)
X	0	1	1	R/W	LCR	3FBH	Line Control Register (Registro de control de línea) BIT 7 es DLAB
X	1	0	0	R/W	MCR	3FCH	Modem Control Register (Registro de control del modem)
X	1	0	1	R/W	LSR	3FDH	Line Status Register (Registro de estado de la línea)
X	1	1	0	R/W	MSR	3FEH	Modem Status Register (Registro de estado del modem)
X	1	1	1	R/W	SCR	3FFH	Scratch Register (Registro residual)
1	0	0	0	R/W	DLL	3F8H	Divisor Latch LSB (Divisor de velocidad, parte baja)
1	0	0	1	R/W	DLM	3F9H	Divisor latch MSB (Divisor de velocidad, parte alta)

Para programar al 8250 primero se debe enviar las palabras para seleccionar la velocidad de transmisión, lo cual se realiza en los registros DLL y DLM, sin embargo como estos registros se encuentran multiplexados a través del bit DLAB (bit 7 del registro LCR) primero debemos poner a "1" dicho bit.

El puerto serie: UART 8250

Registros

REGISTER MNEMONIC	REGISTER BIT NUMBER							
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
3F8H-- RBR (Read Only)	Data Bit 7 (MSB)	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0 (LSB)*
3F8H-- THR (Write Only)	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
3F8H-- DLL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
3F9H-- DLM	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
3F9H-- IER	0	0	0	0	(EDSSI) Enable Modem Status Interrupt	(ELSI) Enable Receiver Line Status Interrupt	(ETBEI) Enable Transmitter Holding Register Empty Interrupt	(ERBFI) Enable Received Data Available Interrupt
3FAH-- IIR (Read Only)	0	0	0	0	0	Interrupt ID Bit (1)	Interrupt ID Bit (0)	"0" 1F Interrupt Pending
3FBH-- LCR	(DLAB) Divisor Latch Access Bit	Set Break	Stick Parity	(EPS) Even Parity Select	(PEN) Parity Enable	(STB) Number of Stop Bits	(WLSB1) Word Length Select Bit 1	(WLSB0) Word Length Select Bit 0
3FCH-- MCR	0	0	0	Loop	Out 2	Out 1	(RTS) Request to Send	(DTR) Data Terminal Ready
3FDH-- LSR	0	(TEMT) Transmitter Empty	(THRE) Transmitter Holding Register Empty	(BI) Break Interrupt	(FE) Framing Error	(PE) Parity Error	(CE) Overrun Error	(DR) Data Ready
3FEH-- MSR	(DCD) Data Carrier Detect	(RI) Ring Indicator	(DSR) Data Set Ready	(CTS) Clear to Send	(DDCD) Delta Data Carrier Detect	(TERI) Trailing Edge Ring Indicator	(DDSR) Delta Data Set Ready	(DCTS) Delta Clear to Send
3FFH-- SCR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

*LSB, Data Bit 0 is the first bit transmitted or received.

El puerto serie: UART 8250

Programación

```
;DLAB =1
```

```
MOV DX,3FBH
```

```
MOV AL, 80H
```

```
OUT DX,AL
```

```
;Velocidad de Transmisión.
```

```
;Para programar la velocidad a 9600 baudios
```

```
MOV DX,3F8H
```

```
MOV AL,12
```

```
OUT DX,AL
```

```
MOV DX,3F9H
```

```
MOV AL,00
```

```
OUT DX,AL
```

```
;Programación de la trama
```

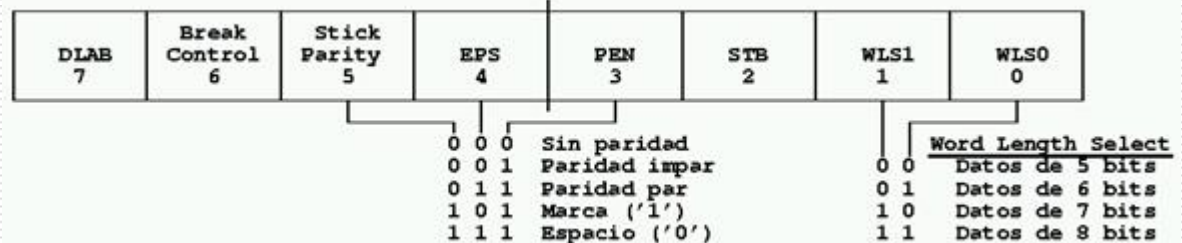
```
MOV DX, 3FBH
```

```
MOV AL,00001111B
```

```
OUT DX,AL
```

TABLE 4. BAUD RATES USING 1.8432MHZ CRYSTAL

DESIRED BAUD RATE	DIVISOR USED TO GENERATE 16 x CLOCK	PERCENT ERROR DIFFERENCE BETWEEN DESIRED AND ACTUAL
50	2304	-
75	1536	-
110	1047	0.026
134.5	857	0.058
150	768	-
300	384	-
600	192	-
1200	96	-
1800	64	-
2000	58	0.69
2400	48	-
3600	32	-
4800	24	-
7200	16	-
9600	12	-
19200	6	-
38400	3	-
56000	2	2.86



El puerto serie: UART 8250

Programación (POLEO)

```
;-----DLAB =1
MOV DX,3FBH
MOV AL, 80H
OUT DX,AL
;-----Velocidad de Transmisión.
;Para programar la velocidad a
;9600 baudios
MOV DX,3F8H
MOV AL,12
OUT DX,AL
MOV DX,3F9H
MOV AL,00
OUT DX,AL
;-----Programación de la trama
MOV DX, 3FBH
MOV AL,00001111B
OUT DX,AL
```

```
;Trabajo del puerto por Poleo
;-----Poleo para transmisión
XX:    MOV DX,3FDH
        IN AL,DX
        AND AL, 20H
        JZ XX
;-----Transmitir dato
        MOV AL, DATO
        MOV DX, 3F8H
        OUT DX,AL
;-----;Poleo para recepción
YY:    MOV DX,3FDH
        IN AL,DX
        AND AL, 01H
        JZ YY
;-----Recibir dato
        MOV DX,3F8H
        IN AL ,DX
```

El puerto serie: UART 8250

Programación

Detalle Programación de la trama

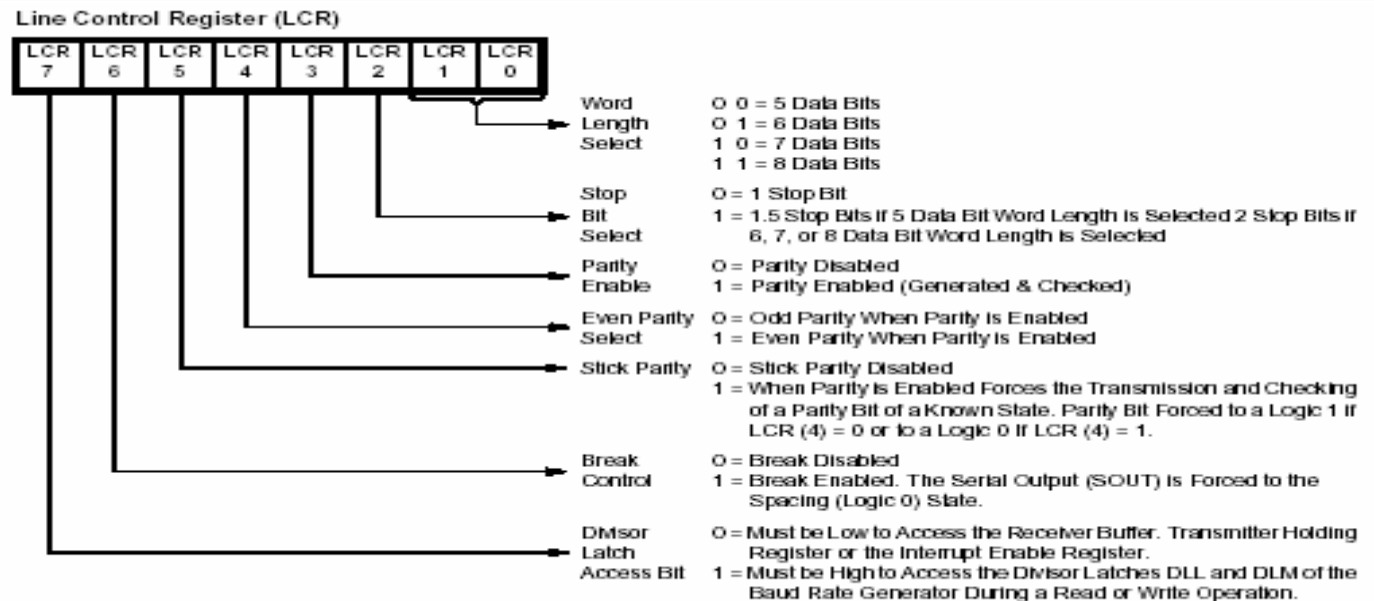
La trama se programa a través del registro LCR (**Line Control Register**).

Por ejemplo si queremos programar con 8 bits de datos, 2 bits de paro y paridad non, la programación quedará de la siguiente forma. El bit DLAB deberá apagarse.

```
MOV DX, 3FBH
```

```
MOV AL,00001111B
```

```
OUT DX,AL
```



El puerto serie: UART 8250

Programación

Detalle POLEO

Para trabajar por poleo es necesario leer el registro de status **LSR (Line Status Register)** y consultar el bit correspondiente (de transmisión o recepción).

;Poleo para transmisión

```
XX:    MOV DX,3FDH
        IN AL,DX
        AND AL, 20H
        JZ XX
```

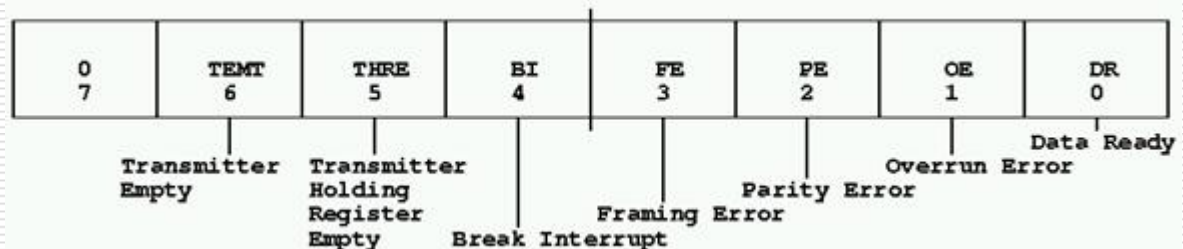
;------Recibir dato del teclado

```
MOV AL, DATO
MOV DX, 3F8H
OUT DX,AL
```

;------Desplegar caracter en ventana de transmisión

;Poleo para recepción

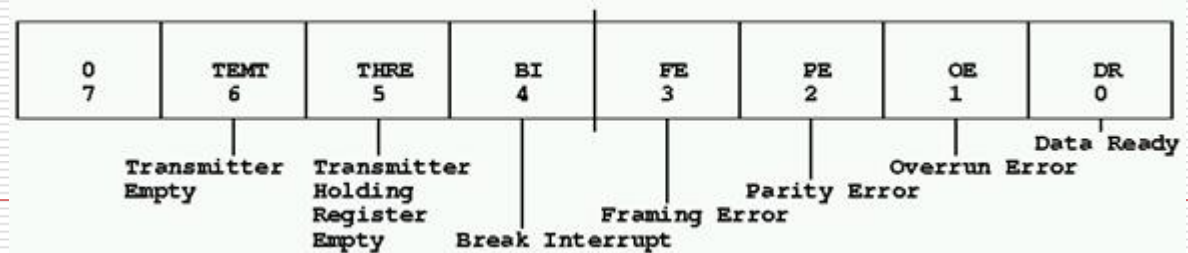
```
YY:    MOV DX,3FDH
        IN AL,DX
        AND AL, 01H
        JZ YY
        MOV DX,3F8H
        IN AL ,DX
```



El puerto serie: UART 8250

Programación

- El bit DR está activo cuando hay un carácter listo en el RBR (buffer de recepción) y es puesto a 0 cuando se lee el RBR. Los bits 1 al 4 de este registro (OE, PE, FE y BI) son puestos a 0 al consultarlos -cuando se lee el LSR- y al activarse pueden generar una interrupción de prioridad 1 si ésta interrupción está habilitada. OE se activa para indicar que el dato en el RBR no ha sido leído por la CPU y acaba de llegar otro que lo ha sobrescrito. PE indica si hay un error de paridad. FE indica si el carácter recibido no tiene los bit de stop correctos. BI se activa cuando la entrada de datos es mantenida en espacio (a 0) durante un tiempo superior al de transmisión de un carácter (bit de inicio + bits de datos + bit de paridad + bit de parada).
- THRE indica que el 8250 puede aceptar un nuevo carácter para la transmisión: este bit se activa cuando el THR queda libre y se desactiva escribiendo un nuevo carácter en el THR. Se puede producir, si está habilitada; la interrupción THRE (prioridad 3). El 8250 emplea un registro interno para ir desplazando los bit y mandarles en serie (el Transmitter Shift Register), dicho registro se carga desde el THR. Cuando ambos registros (THR y el Transmitter Shift) están vacíos, TEMT se activa; volverá a desactivarse cuando se deje otro dato en el THR hasta que el último bit salga por SOUT



El puerto serie: UART 8250

Programación

```

; activar interrupción RX y TX
MOV DX,3F9H
MOV AL,03H
OUT DX,AL

```

```

; poner bit OUT2=1
MOV DX,3FCH
MOV AL, 08H
OUT DX,AL

```

```

; Quitar máscara de interrupción
IN AL, 21H
AND AL,0EFH
OUT 21H,AL

```

TABLE 2. INTERRUPT IDENTIFICATION REGISTER

INTERRUPT IDENTIFICATION				INTERRUPT SET AND RESET FUNCTIONS		
BIT 2	BIT 1	BIT 0	PRIORITY LEVEL	INTERRUPT FLAG	INTERRUPT SOURCE	INTERRUPT RESET CONTROL
X	X	1		None	None	
1	1	0	First	Receiver Line Status	OE, PE, FE, or BI	LSR Read
1	0	0	Second	Received Data Available	Receiver Data Available	RBR Read
0	1	0	Third	THRE	THRE	IIR Read if THRE is the Interrupt Source or THR Write
0	0	0	Fourth	Modem Status	CTS, DSR, RI, DCD	MSR Read

NOTE: X = Not Defined, May Be 0 or 1

```

; Subrutina de Interrupción

```

```

SUBINT PROC

```

```

    MOV DX,3FAH

```

```

    IN AL,DX

```

```

    CMP AL,04

```

```

    JE RECIBIR

```

```

    MOV AL,DATO

```

```

    MOV DX,3F8H

```

```

    OUT DX,AL

```

```

    JMP SALIR

```

```

RECIBIR: MOV DX,3F8H

```

```

    IN AL,DX

```

```

SALIR: IRET

```

```

SUBINT ENDP

```

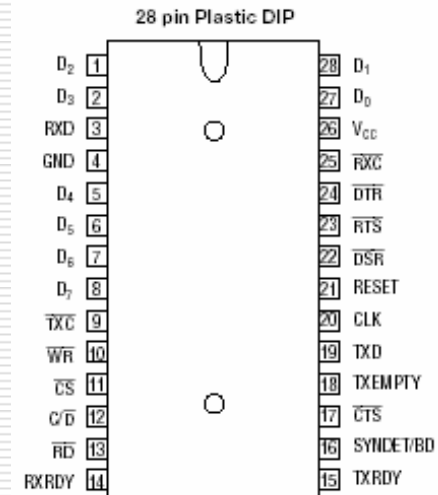
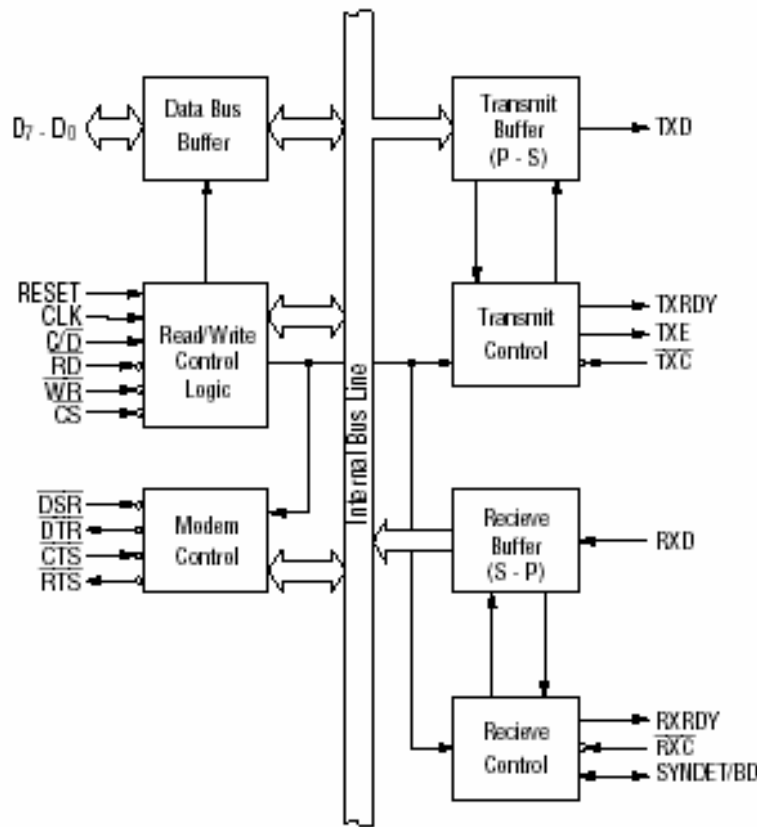
El puerto serie: UART 8250

Programación (Interrupciones)

```
;Cambio de vector (OCH)  
;-----DLAB = 1  
MOV DX,3FBH  
MOV AL, 80H  
OUT DX,AL  
;-----Velocidad de Transmisión.  
;Para programar la velocidad a  
;9600 baudios  
MOV DX,3F8H  
MOV AL,12  
OUT DX,AL  
MOV DX,3F9H  
MOV AL,00  
OUT DX,AL  
;-----Programación de la trama  
MOV DX, 3FBH  
MOV AL,00001111B  
OUT DX,AL  
;poner bit OUT2=1  
MOV DX,3FCH  
MOV AL, 08H  
OUT DX,AL
```

```
; Quitar máscara de interrupción  
IN AL, 21H  
AND AL,0EFH  
OUT 21H,AL  
; activar interrupción TX y RX  
MOV DX,3F9H  
MOV AL,03H  
OUT DX,AL  
:  
:  
; Subrutina de Interrupción  
SUBINT PROC  
    MOV DX,3FAH  
    IN AL,DX  
    CMP AL,04  
    JE RECIBIR  
TRANSMITIR: MOV AL,DATO  
    MOV DX,3F8H  
    OUT DX,AL  
    JMP SALIR  
RECIBIR: MOV DX,3F8H  
    IN AL,DX  
SALIR: IRET  
SUBINT ENDP
```

El puerto serie: USART 8251A



USART significa "Universal Synchronous/Asynchronous Receiver-Transmitter."

Es un dispositivo que sirve para transmitir o recibir datos secuenciales de manera síncrona o asíncrona.

El puerto serie: USART 8251A

USART 8251

Terminales:

C/D' señal de Control/Datos (Ao)

RD' Read. Señal de lectura

WR' Write. Señal de Escritura

TxC' Transmitter Clock Output.

TxD Transmitter Data (salida serie)

TxRDY Transmitter Ready

RxC' Receiver Clock Input.

RxD Receiver Data (entrada serie)

RxRDY Receiver Ready

DSR' Data Set Ready

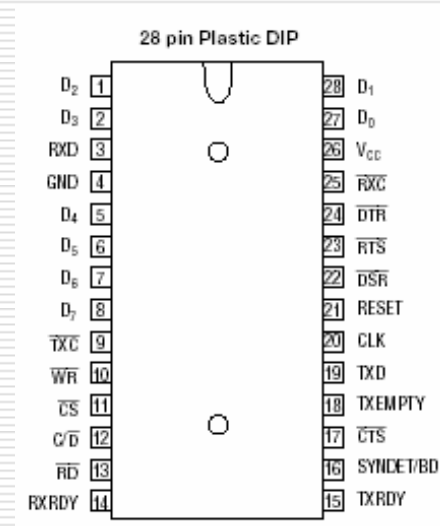
DTR' Data Terminal Ready

Syndet/BD Detector de sincronía/Detector de velocidad (rapidez de envío)

RTS' Request to Send. Petición de envío de datos

CTS' Clear to Send. Borrarse para enviar datos

TxE Transmitter Empty. Transmisor vacío.



El puerto serie: USART 8251A

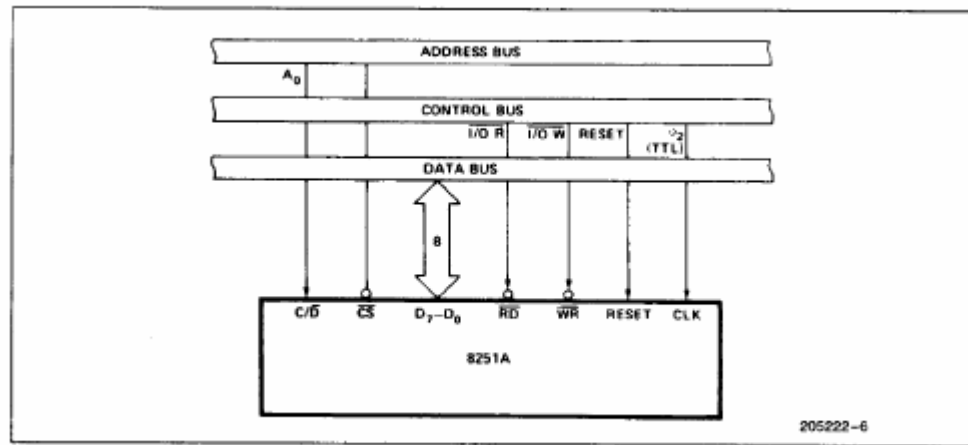


Figure 6. 8251A Interface to 8080 Standard System Bus

Table 1 Operation between MSM82C51A and CPU

CS	C/D	RD	WR	
1	x	x	x	Data Bus 3-State
0	x	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

El puerto serie: USART 8251A

For Example:

If Baud Rate equals 110 Baud,
 TxC equals 110 Hz in the 1x mode.
 TxC equals 1.72 kHz in the 16x mode.
 TxC equals 7.04 kHz in the 64x mode.

For Example:

Baud Rate equals 300 Baud, if
 RxC equals 300 Hz in the 1x mode;
 RxC equals 4800 Hz in the 16x mode;
 RxC equals 19.2 kHz in the 64x mode.
 Baud Rate equals 2400 Baud, if
 RxC equals 2400 Hz in the 1x mode;
 RxC equals 38.4 kHz in the 16 mode;
 RxC equals 153.6 kHz in the 64 mode.

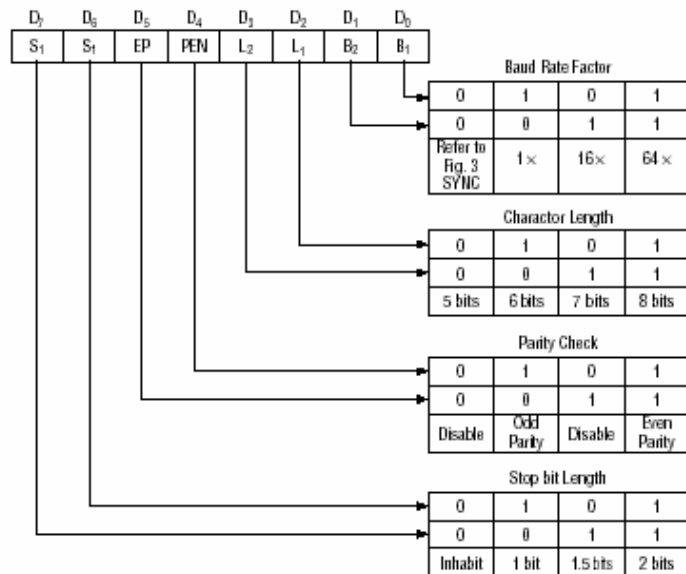


Fig. 2 Bit Configuration of Mode Instruction (Asynchronous)

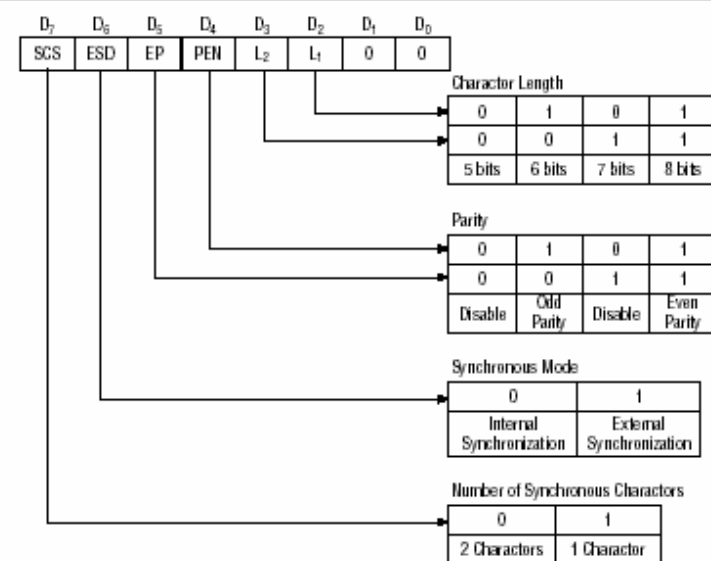


Fig. 3 Bit Configuration of Mode Instruction (Synchronous)

Programación 8251A

```
; Palabra de Modo basura
MOV AL,XX
OUT CTRL,AL
; Palabra de Comando con RESET
MOV AL,40H
OUT CTRL,AL
; Palabra de Modo
; 2bits paro, sin paridad, 7 bits
; factor 1
MOV AL,11X01001B
OUT CTRL,AL
; Palabra de comando
MOV AL,
OUT CTRL,AL
```

For Example:

If Baud Rate equals 110 Baud,
TxCl equals 110 Hz in the 1x mode.
TxCl equals 1.72 kHz in the 16x mode.
TxCl equals 7.04 kHz in the 64x mode.

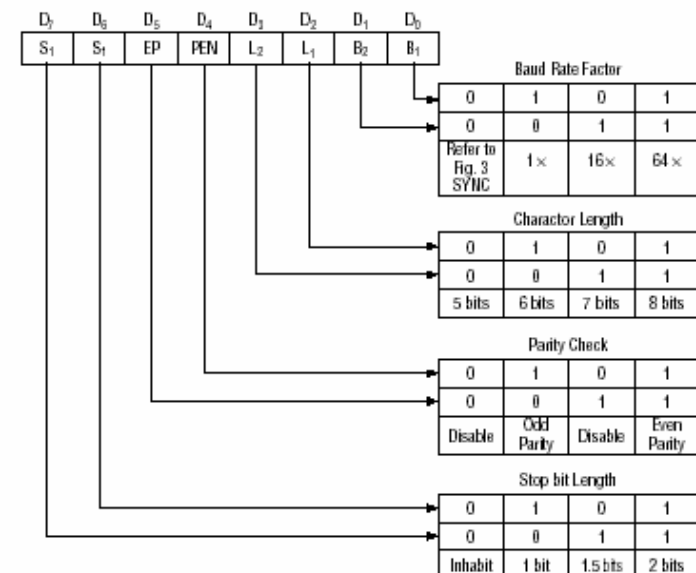
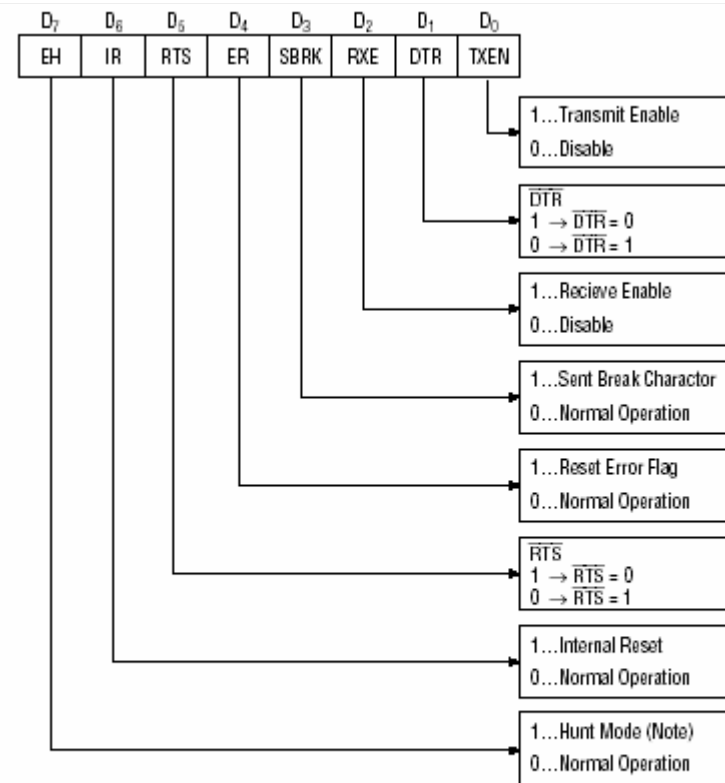


Fig. 2 Bit Configuration of Mode Instruction (Asynchronous)

Programación 8251A

```
; Palabra de Modo basura
MOV AL,XX
OUT CTRL,AL
; Palabra de Comando con RESET
MOV AL,40H
OUT CTRL,AL
; Palabra de Modo
; 2bits paro,sin paridad,7 bits
; factor 1
MOV AL,11X01001B
OUT CTRL,AL
;Palabra de comando
MOV AL,X0X101X1B
OUT CTRL,AL
```



Note: Search mode for synchronous characters in synchronous mode.

Fig. 4 Bit Configuration of Command

Programación 8251A

POLEO

```

; Palabra de Modo basura
MOV AL,XX
OUT CTRL,AL
; Palabra de Comando con RESET
MOV AL,40H
OUT CTRL,AL
; Palabra de Modo: 2bits paro, sin
; paridad, 7 bits factor 1
MOV AL,11X01001B
OUT CTRL,AL
; Palabra de comando
MOV AL,X0X101X1B
OUT CTRL,AL
; Poleo Transmisión
XX: IN AL,CTRL
    AND AL,01
    JZ XX
; -----
TX: MOV AL, DATO
    OUT PTO_DATOS,AL

```

; Poleo Recepción

```

YY: IN AL,CTRL
    AND AL,02
    JZ YY
; -----
RX: IN AL,PTO_DATOS

```

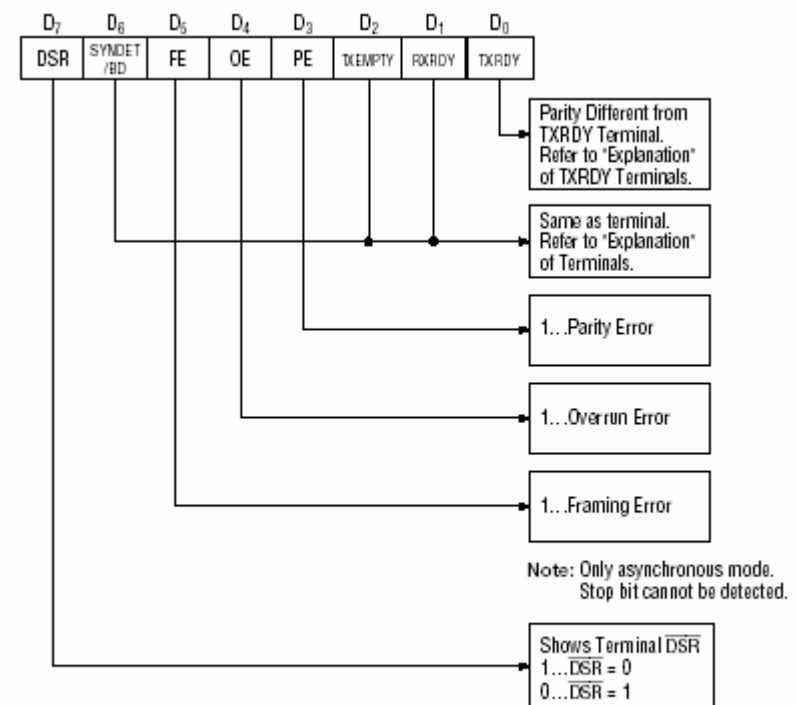


Fig. 5 Bit Configuration of Status Word

Programación 8251A

INTERRUPCIONES

```
;CODIGO CAMBIO DE VECTOR TX
;CODIGO CAMBIO DE VECTOR RX
;Palabra de Modo basura
MOV AL,XX
OUT CTRL,AL
;Palabra de Comando con RESET
MOV AL,40H
OUT CTRL,AL
;Palabra de Modo: 2bits paro, sin
; paridad,7 bits factor 1
MOV AL,11X01001B
OUT CTRL,AL
;Palabra de comando
MOV AL,X0X101X1B
OUT CTRL,AL
```

.
.
.

```
;Subrutina de TX
SUB_TX PROC
    MOV AL,DATO
    OUT PTO_DATOS,AL
    IRET
SUB_TX ENDP
```

```
;Subrutina de RX
SUB_RX PROC
    IN AL, PTO_DATOS
    IRET
SUBRX ENDP
```

El puerto serie: USART 8251A

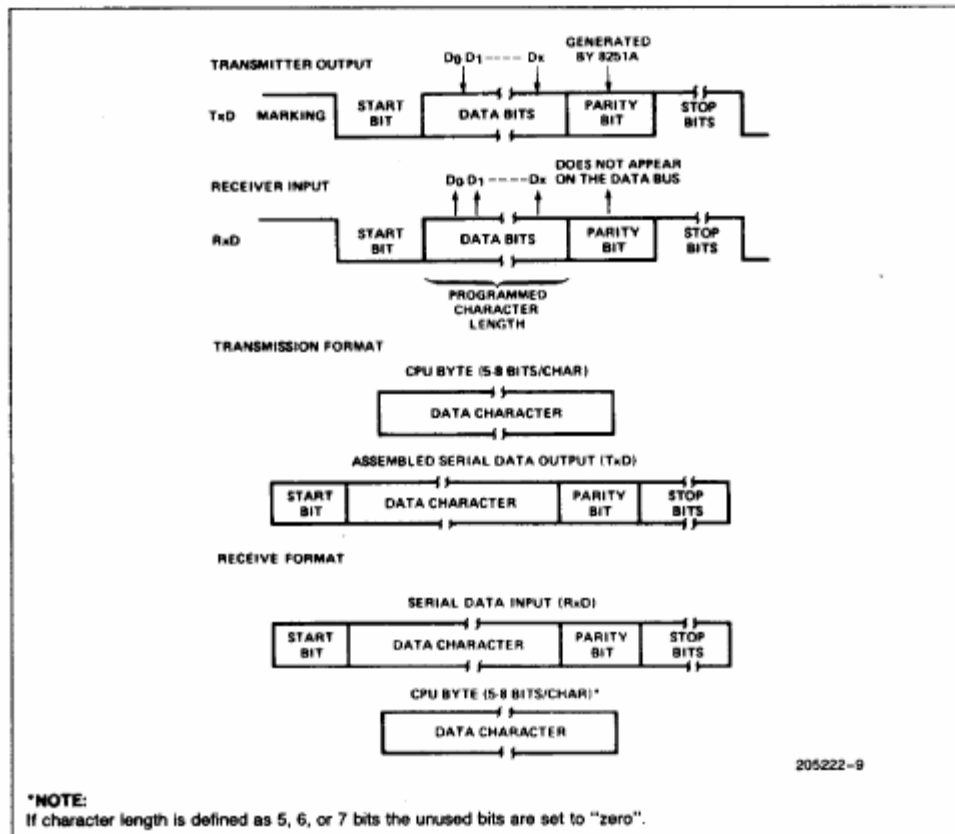


Figure 9. Asynchronous Mode

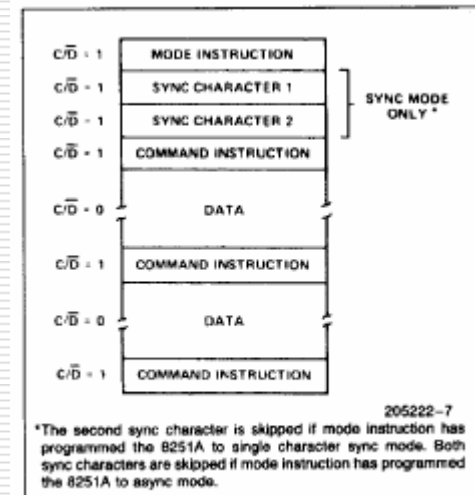


Figure 7. Typical Data Block

El puerto serie: USART 8251A

□ Aplicaciones

APPLICATIONS OF THE 8251A

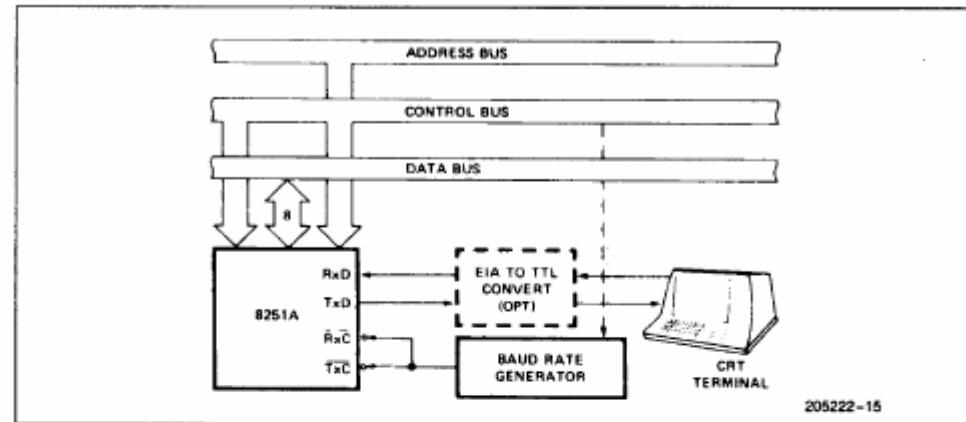


Figure 14. Asynchronous Serial Interface to CRT Terminal, DC—9600 Baud

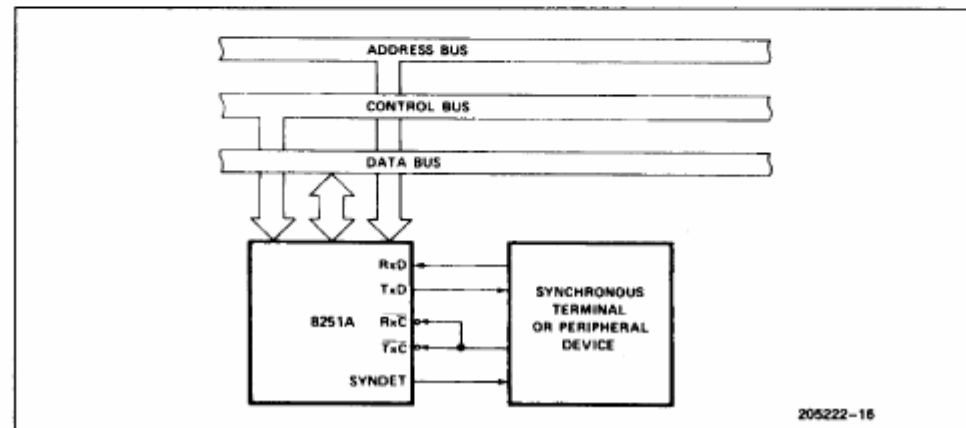
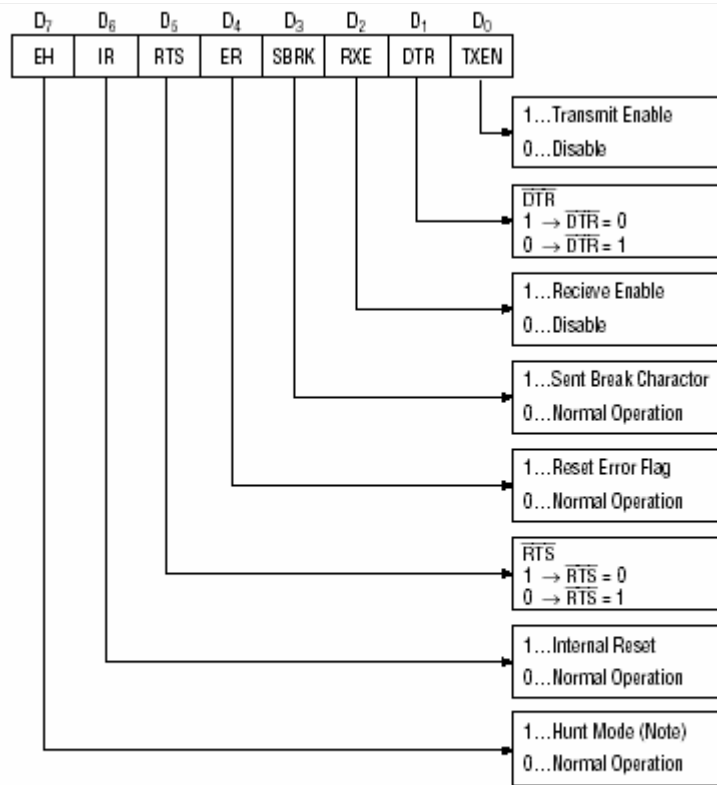


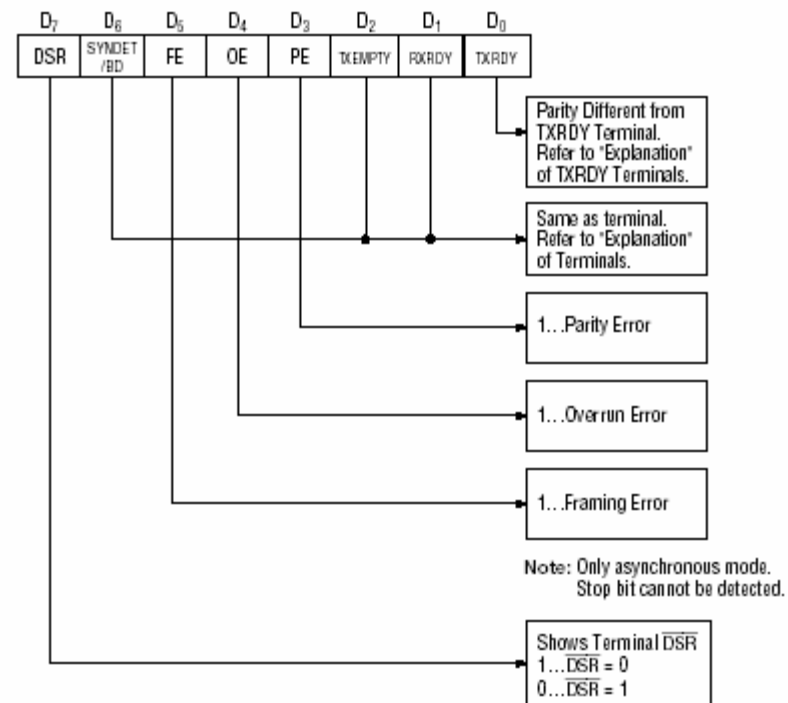
Figure 15. Synchronous Interface to Terminal or Peripheral Device

El puerto serie: USART 8251A



Note: Search mode for synchronous characters in synchronous mode.

Fig. 4 Bit Configuration of Command



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

Fig. 5 Bit Configuration of Status Word