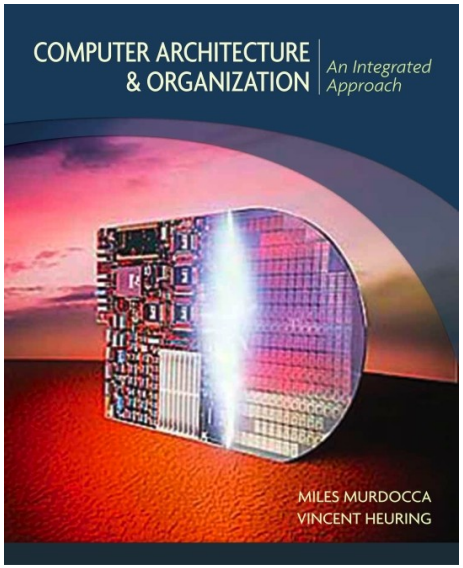
Organización de las Computadoras

Leonardo Giovanini



Puertos de la computadora

Contenidos

Puertos de una computadora

Puertos de una computadora

En términos informáticos, un ***puerto es una interfaz*** a través de la cual se puede ***enviar y recibir los diferentes tipos de datos***. Es decir, refiere a la parte de una computadora disponible para la conexión a periféricos, como dispositivos de entrada y salida.

En la capa física, un puerto de computadora es una ***salida especializada*** al que se conecta un dispositivo, proporcionando un método y un medio para transferir señales.

Se denomina ***puerto lógico*** a una zona o localización de la memoria de la computadora que se asocia con un puerto físico o un canal de comunicación, y que proporciona un espacio para el almacenamiento temporal de la información que se va a transferir entre la localización de memoria y el canal de comunicación.

En el ámbito de Internet, un puerto es ***el valor que se usa***, en el modelo de la capa de transporte, para distinguir entre las múltiples aplicaciones que se pueden conectar al mismo host, o puesto de trabajo.

Puertos de una computadora

En la capa física, un puerto de computadora es una salida especializada al que se conecta un conector y proporcionan un método para transferir señales entre dispositivos.

Electrónicamente, los puertos se pueden dividir según la transferencia de señal

- **Puertos serie -** envían y reciben un bit a la vez a través de un par de cables;
- **Puertos paralelos -** envían y reciben varios bits a través de conjuntos de cables.

Una vez que se conectan los puertos, requieren un protocolo de enlace, donde se comparte el tipo de transferencia, la velocidad de transferencia y otra información necesaria.

Los puertos externos de una computadora son

1 – Puerto para PS/2 para ratón;
2 – Puerto de salida SPDIF;
3 – Puerto de entrada SPDIF;
4 – Puerto VGA (adaptador de vídeo);
5 – Puerto IEEE 1394a (puerto FireWire)
6 – Puerto de red local (LAN)
7 – Audio - Altavoz central;
8 – Audio: Salida del altavoz posterior;
9 – Audio: Entrada de línea;
10 – Audio: Salida de línea;
11 – Audio: Micrófono;
12 – Salida de altavoz lateral;
13 – Puertos USB;
14 – Puerto PS/2 para teclado.

Puertos paralelo

Puerto paralelo

Un ***puerto paralelo*** es una interfaz entre un computador y un periférico, cuya principal característica es que los bits de datos viajan juntos, enviando un paquete de byte a la vez. Es decir, se implementa un cable o una vía física para cada bit de datos formando un bus.

A través del puerto paralelo podemos controlar periféricos externos como motores y reles, entre otros dispositivos.

El puerto paralelo de las computadoras, de acuerdo a la ***norma Centronics***, está compuesto por un bus de comunicación ***bidireccional de 8 bits de datos***, además de un ***conjunto de líneas de protocolo***. Las características eléctricas del puerto son

Tensión de nivel alto 5 V;

Tensión de nivel bajo 0 V.

Las líneas de comunicación cuentan con un latch que mantiene el último valor que les fue escrito hasta que se escribe un nuevo dato,

Puerto paralelo

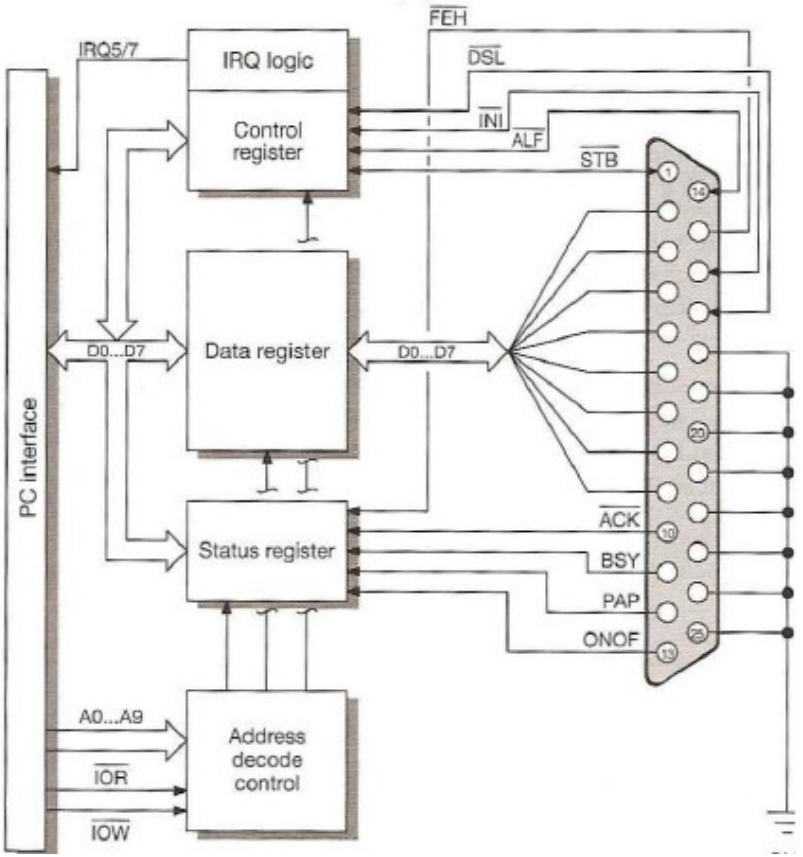
Componentes

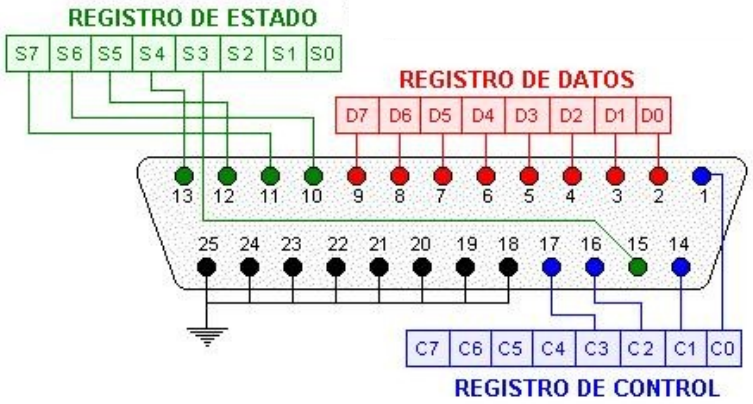
Los componentes del puerto paralelo son tres registros de ocho bits cada uno

- **Datos** – dirección base + 00h
- **Estado** – dirección base + 01h
- **Control** – dirección base + 02h

La computadora puede modificar el registro de control y responde a las interrupciones.

La computadora puede leer el registro de estado para comprobar el estado del puerto.





El **registro de datos** almacena los datos que deben ser transferidos. Este registro puede ser tanto leído como escrito.

Puerto paralelo

Componentes – Registro de estado

El **registro de estado** es de solo lectura y permite al computador comprobar el estado del dispositivo.

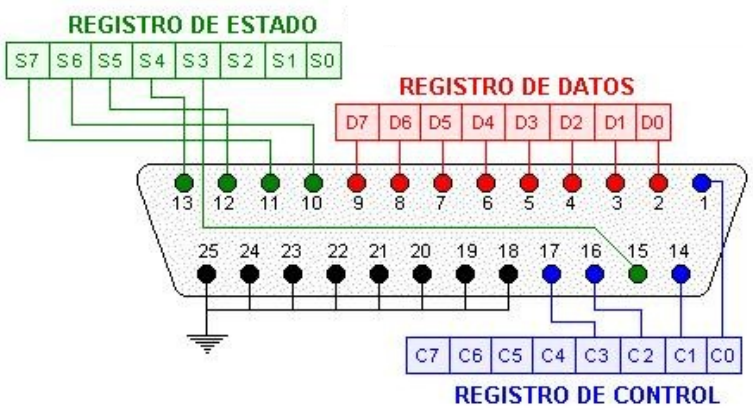
Esta compuesto por seis líneas

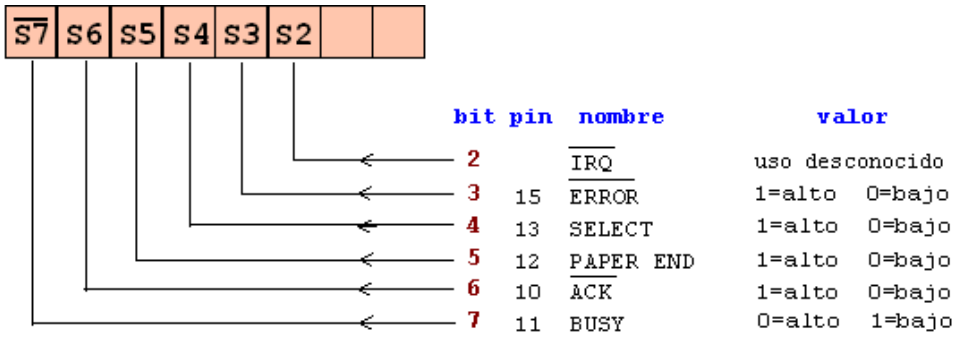
Ack – un pulso bajo indica que se han recibido datos en el dispositivo y que el mismo está preparado para recibir más datos;

Busy – en alto indica que el dispositivo está ocupado;

Paper End – en alto indica que no hay papel;

Select – en alto para seleccionar un dispositivo;





Error – en bajo indica la ocurrencia de un error en el dispositivo;

Select – en bajo indica que el dispositivo ha sido seleccionado (en gral. no se usa, ya que SelectIn se fija a alto);

IRQ – en bajo indica la solicitud de interrupción.

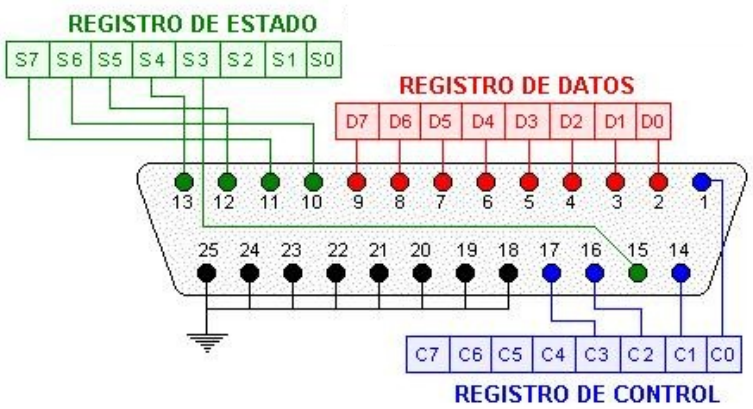
Puerto paralelo

Componentes– Registro de control

El **registro de control** permite a la computadora gestionar el comportamiento del dispositivo.

Por otro lado, también permite al dispositivo generar interrupciones para solicitar la atención de la computadora.

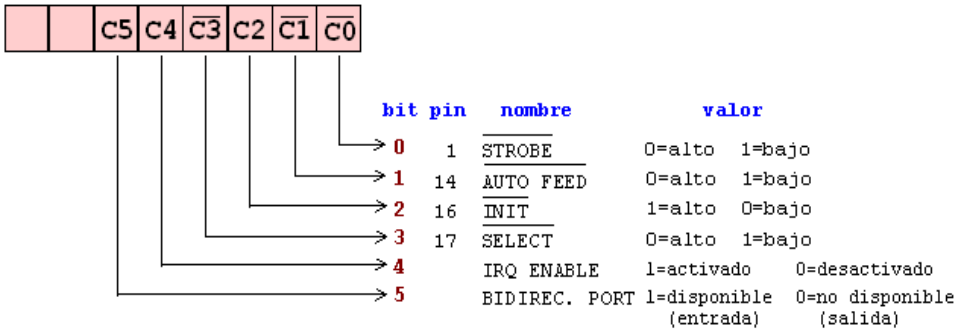
Esta compuesto por seis líneas



Strobe – si está bajo habilita al dispositivo que reciba los datos enviados;

AutoFeed – si está bajo, el papel se mueve una línea tras la impresión;

Init – si se envía un pulso en bajo se reinicia el dispositivo;



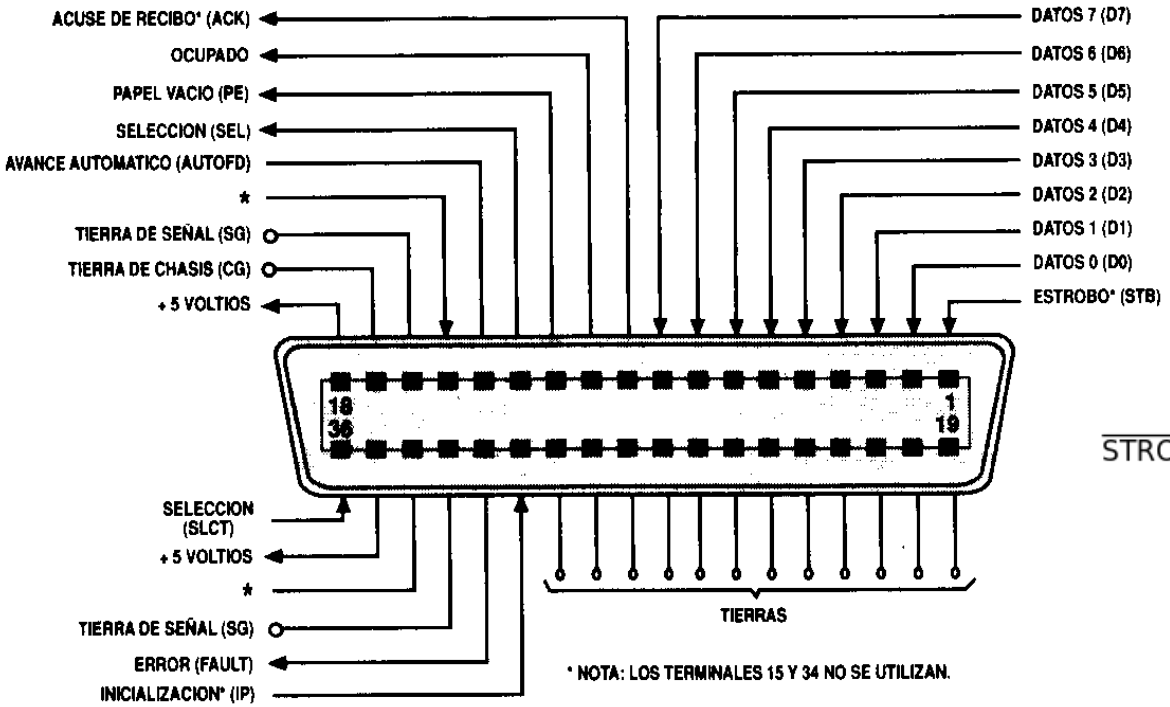
Select – en bajo indica que el dispositivo ha sido seleccionado (en gral. no se usa, ya que SelectIn se fija a alto);

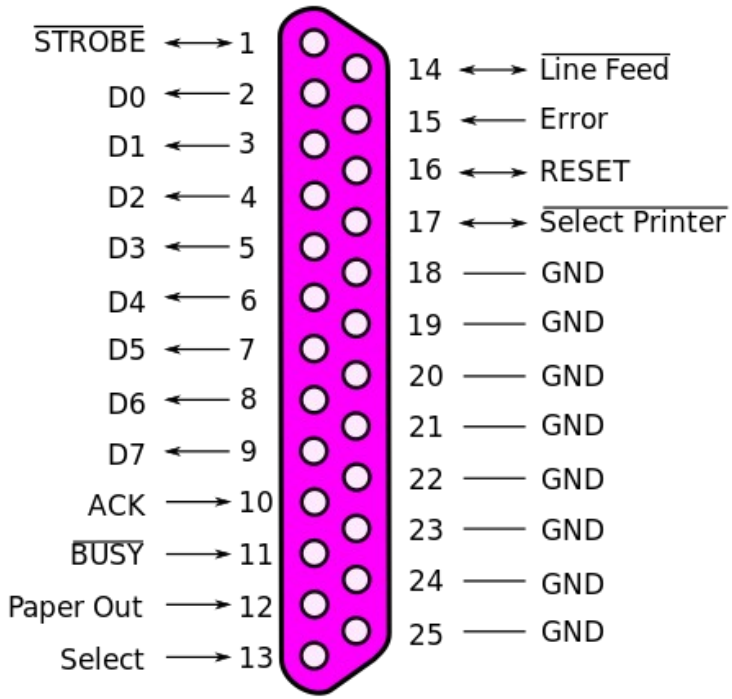
IRQ Enable – en uno indica que se pueden utilizar las interrupciones del dispositivo;

Bidirec – en uno indica que el puerto bidireccional esta habilitado.

Puerto paralelo

Componentes – Conectores





Puerto paralelo

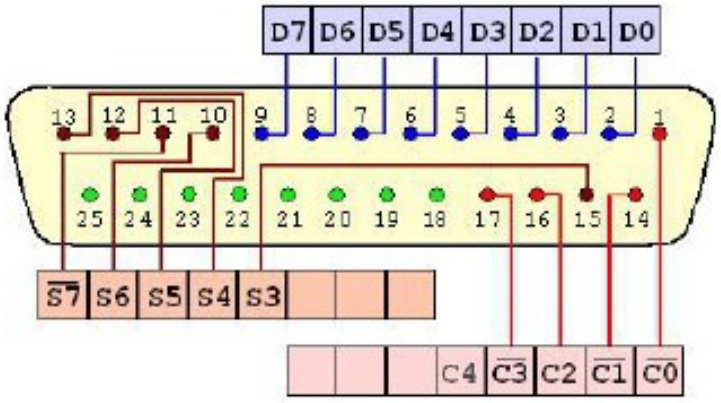
Protocolo de comunicación

Un puerto paralelo toma un ciclo ISA para leer o escribir. Una lectura se puede hacer aproximadamente cada 1.5 ms, en la práctica pueden ser desde 1.2 ms a 2 ms. Algunos puertos soportan un modo "*turbo*" que elimina los estados de espera de la CPU, con lo que la velocidad de lectura/escritura del puerto se duplica.

Las características eléctricas del puerto limitaba la longitud de los cables externos a un máximo de 1,5 metros.

En total, el protocolo de comunicación cuenta con 17 líneas

- **Datos:** 8 líneas de salida;
- **Estado:** 5 líneas de entrada (una invertida);
- **Control:** 4 líneas de salida (tres invertidas).



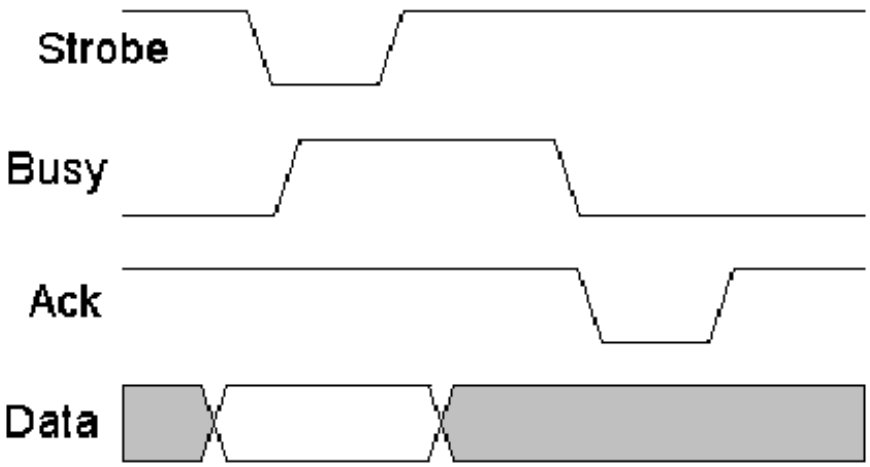
	Signal Name	Adapter Pin Number	
	←- Strobe	1	
E	← +Data Bit 0	2	P
X	← +Data Bit 1	3	A
T	← +Data Bit 2	4	R
E	← +Data Bit 3	5	A
R	← +Data Bit 4	6	L
N	← +Data Bit 5	7	L
A	← +Data Bit 6	8	E
L	← +Data Bit 7	9	L
	-Acknowledge →	10	
D	+Busy →	11	A
E	+Paper End →	12	D
V	+Select →	13	A
I	← -Auto Feed	14	P
C	-Error →	15	T
E	← -Initialize	16	E
	← -Select Input	17	R
	Ground	18-25	

Puerto paralelo

Protocolo de comunicación

El protocolo necesario para que una computadora transmita un byte a la impresora utiliza principalmente las señales *Strobe*, *Ack* y *Busy*, y viene dado por

1. Esperar a que el dispositivo no esté ocupada (Busy =0);
2. Activar *Strobe* (Strobe=0) para que el dispositivo acepte el dato;
3. Introducir el byte a enviar en el registro de datos;
4. La impresora activa *Busy* (Busy=1) para indicar que está procesando el dato;
5. La impresora activa *Ack* (Ack=0) para indicar que ha terminado y se puede regresar al primer paso.



Las otras señales sirven para verificar el estado de la impresora (*Error*, *PaperEnd*), para reiniciarla (*Init*) y para configurarla (*AutoFeed*, *Select*).

Se debe habilitar el buffer que conecta *ACK* con *IRQ* seteando el bit 4 del registro de control. Luego se debe preparar una rutina de atención de la interrupción enviando la señal EOI al registro de control del controlador de interrupciones al salir de la rutina. Finalmente se habilita la interrupción IRQ5 (o IRQ7).

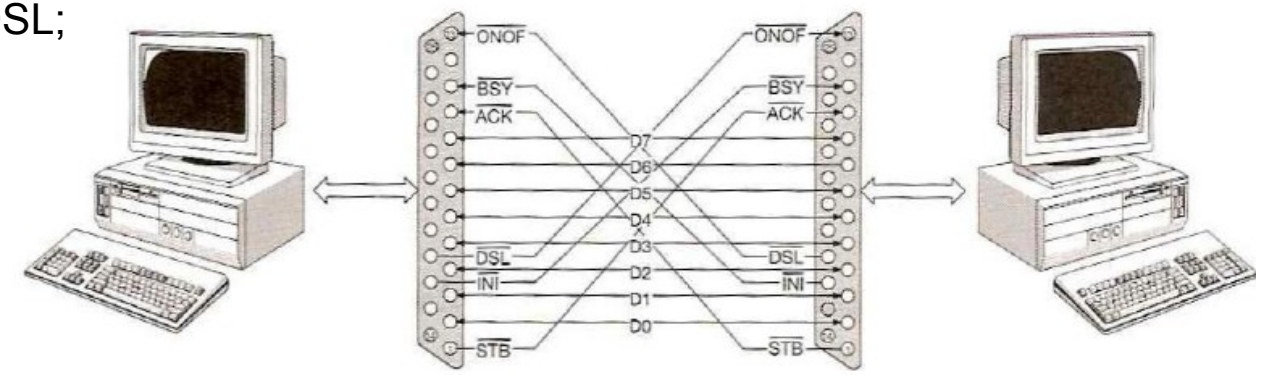
Puerto paralelo

Protocolo de comunicación

Es posible realizar un intercambio de datos entre dos computadores usando el puerto paralelo.

El protocolo de comunicación es el siguiente

1. El emisor activa DSL para indicar que quiere enviar datos;
2. El receptor contesta activando INI;
3. El emisor carga el dato y realiza un pulso de STB, lo que genera una interrupción en el receptor;La rutina de gestión de interrupción del receptor adquiere el dato;
4. El receptor realiza un pulso de STB, lo que genera una interrupción en el emisor;
5. La interrupción en el emisor devuelve el control a la rutina que repetirá los pasos 3, 4 y 5 hasta que el último dato se transmita;
6. El transmisor desactiva DSL;
7. El receptor desactiva INI.



Puerto paralelo IEEE 1284

El estándar IEEE 1284 permite un flujo de datos bidireccional y mas eficiente, con un rendimiento teórico máximo de 4 MB/s. En la practica, el ancho de banda es de alrededor de 2 MB/s dependiendo del hardware utilizado.

Este estándar define cinco modos de transferencia de datos, proveyendo métodos de transferencias de datos entre la computadora y el periférico. Los modos definidos son:

Modo de Compatibilidad (Standard Parallel Port) - es una interfaz unidireccional con unas pocas diferencias con el puerto paralelo original;

Modo Nibble - es una interfaz bidireccional que permite transferir nibbles, utilizando las cuatro lneas de estado del modo compatibilidad para transferir datos;

Modo de Byte - es una interfaz bidireccional half-duplex que transmite bytes utilizando las lineas de datos;

Modo Mejorado (Enhanced Parallel Port) - es una interfaz bidireccional half-duplex que permite transferencias grandes cantidades de datos a alta velocidad gestinada por el procesador;

Modo Capacidades Extendidas (Extended Capability Port) - es una interfaz bidireccional half-duplex similar al modo mejorado peroque utiliza aceso directo a memoria para gestionar la transferencias.

El puerto IEEE-1284 requiere que la comunicación del dispositivo siempre se inicie en el modo Nibble. Si el host no recibe respuesta en este modo, asume que el dispositivo es un heredado y entrará en el Modo de SPP. De lo contrario, el modo que se admite se negocia entre los dispositivos host y cliente mediante un intercambio de mensajes en modo Nibble.

Puerto paralelo IEEE 1284

Las líneas utilizadas por los cinco modos de operación son las mismas, pero con diferentes nombres y funciones.

Pin	Direction	Compatible	Nibble	Byte	ECP	EPP
1	PC to device	Strobe	–	HostClk	HostClk	Write
2	PC to/from device	Data 1	–	Data 1	Data 1	AD1
3	PC to/from device	Data 2	–	Data 2	Data 2	AD2
4	PC to/from device	Data 3	–	Data 3	Data 3	AD3
5	PC to/from device	Data 4	–	Data 4	Data 4	AD4
6	PC to/from device	Data 5	–	Data 5	Data 5	AD5
7	PC to/from device	Data 6	–	Data 6	Data 6	AD6
8	PC to/from device	Data 7	–	Data 7	Data 7	AD7
9	PC to/from device	Data 8	–	Data 8	Data 8	AD8
10	Device to PC	Acknowledge	PtrClk	PtrClk	PeriphClk	Interrupt
11	Device to PC	Busy	Data 3, 7	PtrBusy	PeriphAck	Wait
12	Device to PC	Paper Error	Data 2, 6	AckDataReq	AckRevers	UserDefin1
13	Device to PC	Select	Data 1, 5	–	Xflag	UserDefin3
14	PC to device	Autofeed	HostBusy	HostBusy	HostAck	DataStrb
15	not defined					
16	Logic GND					
17	Chassis GND					
18	Peripheral Logic High					
19	GND Strobe					
20	GND Data 1					
21	GND Data 2					
22	GND Data 3					
23	GND Data 4					
24	GND Data 5					
25	GND Data 6					
26	GND Data 7					
27	GND Data 8					
28	GND Paper Error, Select, Acknowledge					
29	GND Busy, Fault					
30	GND Auto Feed, Select In, Init					
31	PC to device	INIT	–	–	RevRequest	Reset
32	Device to PC	Fault	Data (0, 4)	DataAvail	PeriRequest	UserDefin2
33	not defined					
34	not defined					
35	not defined					
36	PC to device	SelectIn	1284Active	1284Active	1284Active	AddressStrb

Byte de control

D7	D6	D5	D4	D3	D2	D1	D0	
				0		0		reserved
	1	0	0				0	EPP mode
	0	0	1				0	ECP mode without RLE
	0	1	1				0	ECP mode with RLE
	0	0	0				1	Byte mode
	0	0	0				0	Nibble mode
					0			no device ID requested
					1			device ID requested
0								without extensibility link
1								with extensibility link

Puerto paralelo IEEE 1284

Puerto Paralelo Estandar – Modo byte

El **modo de Byte** es un modo de transferencia half-dúplex que permite que el dispositivo transmita un byte a la vez utilizando las mismas líneas de datos que se usan para la otra dirección.

Utiliza el bit 5 del registro de control (C5) para definir la dirección de las transferencias:

Si C5=1 el buffer de salida se pone en alta impedancia, *desconectando* los pines del puerto (D0 a D7). Si se escribe en el registro de datos, se escribe en el buffer pero no en la salida, lo cual permite que al leer el puerto se lea la entrada.

Si C5=0 el puerto retorna al modo salida, su estado por defecto.

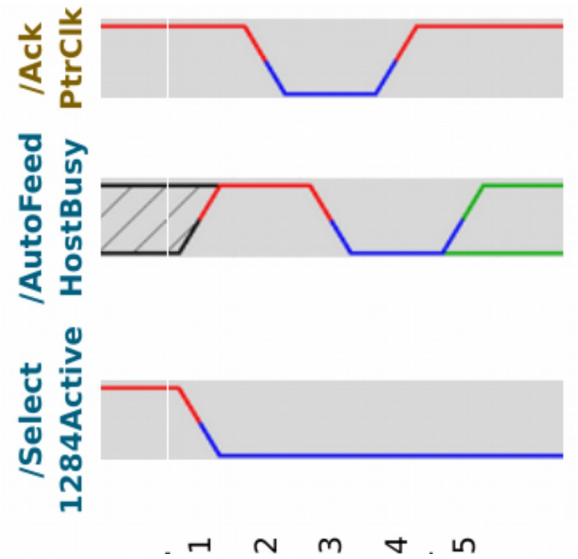
El valor del bit correspondiente del registro de control debe establecerse antes de cada ciclo de transferencia.

7	6	5	4	3	2	1	0
Ptr Busy	Ptr Clk	AckData-req	-	$\overline{\text{Data Available}}$	-	-	-

Status register in byte mode (base address +1)

7	6	5	4	3	2	1	0
-	-	Data Direction	IRQ	1284 Active	-	Host busy	Host Clk

Control register in byte mode (base address +2)



Puerto paralelo IEEE 1284

Puerto Paralelo Estandar – Modo nibble

El **modo nibble** es un modo de transferencia que permite al dispositivo transmitir un nibble por muestra, utilizando las cuatro líneas de estado del modo SPP.

En primer lugar se envía la primera mitad del byte. Una vez que ha sido procesada, se envía la segunda mitad.

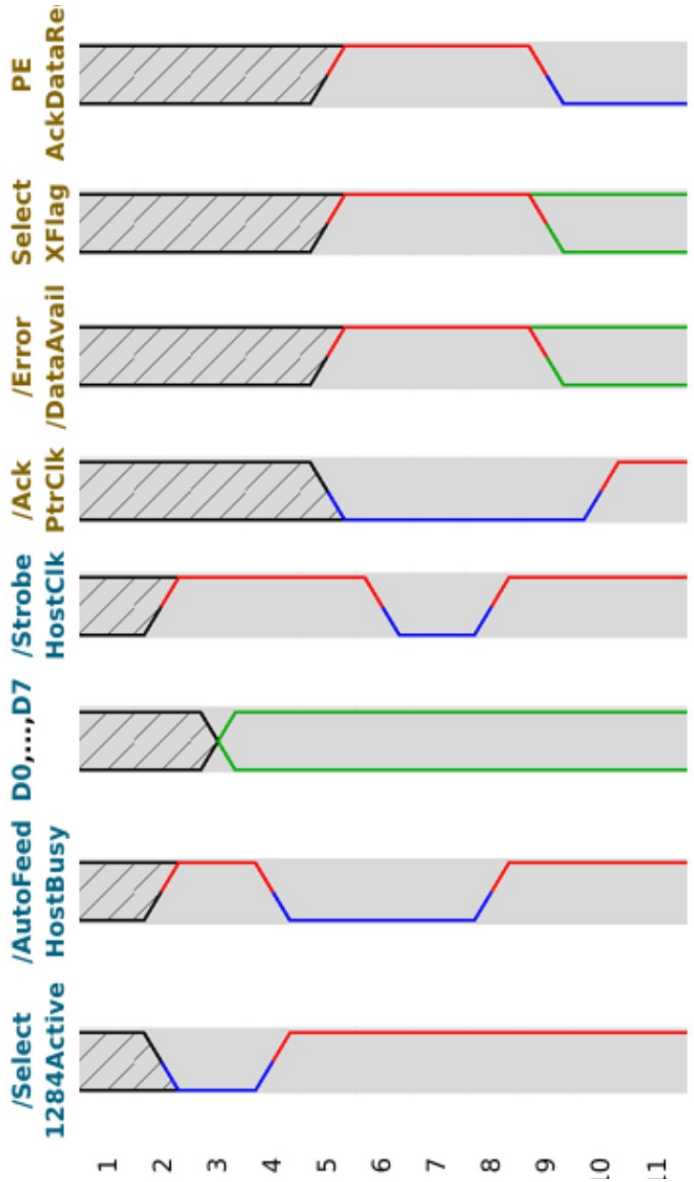
Un dispositivo periférico compatible con 1284 debe implementar la secuencia de negociación, porque así se activan los modos avanzado y de canal inverso.

7	6	5	4	3	2	1	0
Data 3, 7	Ptr Clk	Data 2, 6	Data 1, 5	Data 0, 4	-	-	-

Status register in nibble mode (base address +1)

7	6	5	4	3	2	1	0
-	-	-	IRQ	1284 Active	-	Host busy	-

Control register in nibble mode (base address +2)

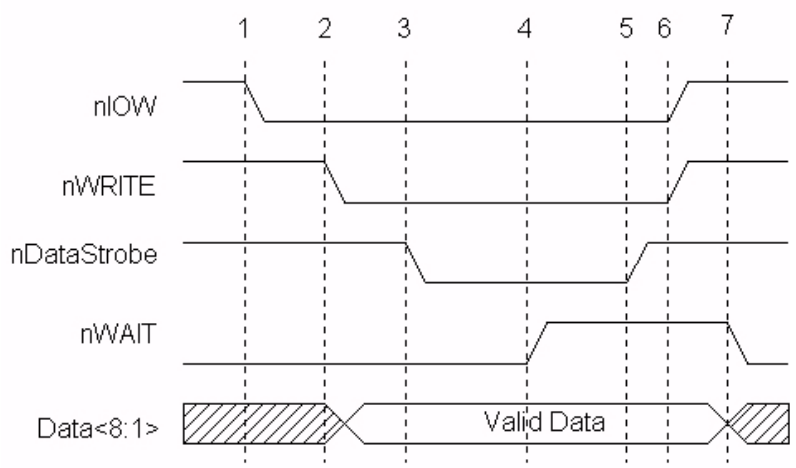


Puerto paralelo IEEE 1284

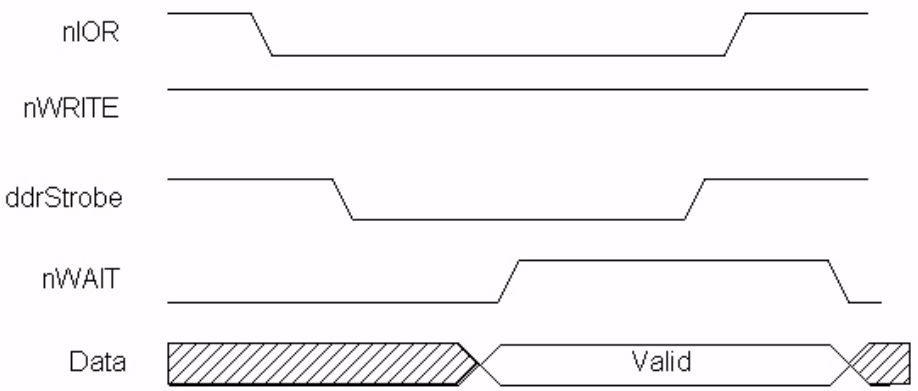
Puerto Paralelo Extendido

El **extended parallel port** (EPP) es una interfaz bidireccional half-duplex contrlada por el procesador diseñada para cambiar la dirección del canal y transmitir mucha información.

SEÑAL SPP	NOMBRE	In/Out	DESCRIPCIÓN
STROBE	WRITE	OUT	Inactivo indica una operación de escritura. Activo un ciclo de lectura.
AUTOFEED	DATASTB	OUT	Inactivo Operación de lectura o escritura de datos que esta en proceso.
SELECTIN	ADDRSTB	OUT	Inactivo Operación de lectura o escritura de direcciones, que esta en proceso.
INIT	RESET	OUT	Inactivo resetea periférico
ACK	INTR	IN	El periférico genera una interrupción al ordenador
BUSY	WAIT	IN	Inactivo indica OK para comenzar el ciclo Activo indica OK para finalizar el ciclo
D[8:1	AD[8:1	BI-DI	Fluyen bi-direccionalmente direcciones y datos
PE	definido por usuario	IN	Diferentes usos según periférico
SELECT	definido por usuario	IN	Diferentes usos según periférico
ERROR	definido por usuario	IN	Diferentes usos según periférico



Puede alcanzar anchos de banda de hasta 2 MB/s a partir de transferencias de bloques de 256 bytes. Se utiliza para periféricos como dispositivos de almacenamiento, adaptadores de red, escaneres, etc.



Puerto paralelo IEEE 1284

Puerto Paralelo Extendido

Este modo permite conectarse con periféricos como las unidades de almacenamiento, escaneres, placas digitalizadoras, entre otras.

Las velocidades pueden variar desde 500 kbps hasta 2 Mbps dependenindo de la longitud del cable de conexión (entre 6 metros y 10 metros).

- Dispone de los 3 registros estándar, lo que permite compatibilidad con el modo SPP;
- Además hay otros 4 registros adicionales;
- Al contrario que los modos anteriores, el protocolo de señales es controlado por hardware

Offset	Description	Access	Function
00h	SPP Data	Write	Standard data register
01h	SPP Status	Read	Standard status register
02h	SPP Control	Write	Standard control register
03h	EPP Address	Read/Write	EPP Address Register
04h	EPP Data 0	Read/Write	EPP data register (8-bit, 16-bit, 32-bit)
05h	EPP Data 1	Read/Write	EPP data register (16-bit, 32-bit)
06h	EPP Data 2	Read/Write	EPP data register (32-bit)
07h	EPP Data 3	Read/Write	EPP data register (32-bit)

Puerto paralelo IEEE 1284

Puerto con Capacidades Mejoradas

El **enhanced capability port** (ECP) es una interfaz bidireccional half-dúplex que utiliza un canal de acceso directo a memoria y un buffer FIFO para proporcionar un ancho de banda de hasta 2.5 MB/s.

Incorpora una compresión de datos en tiempo real basada en Run Lengh Encoding (RLE) para lograr compresiones de hasta 64:1. El RLE es un sistema simple que comprime secuencias largas utilizando un código de dos bytes. Este método funciona sobre cadenas de hasta 128 bytes.

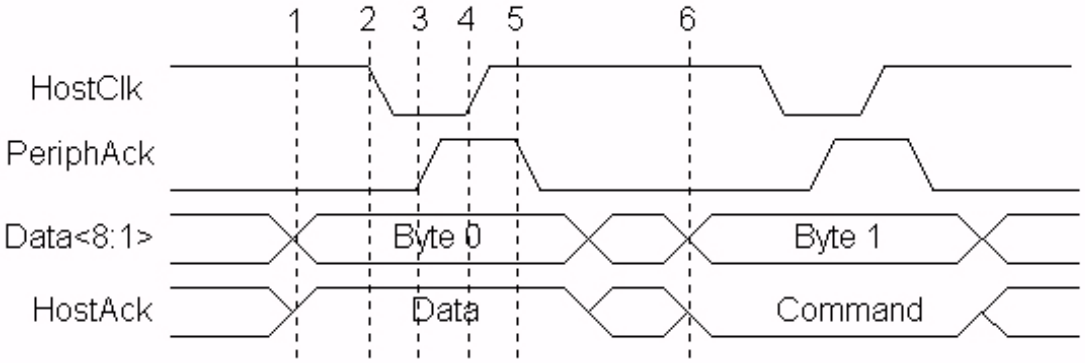
El puerto acepta múltiples dispositivos en un puerto, utilizando un mecanismo de direccionamiento basado en direcciones de canal por el bus de datos.

Offset	Description	Access	ECP-Mode	Function
000h	Data	Write/read	0-1	Data register
000h	ECP-A-FIFO	Write/read	3	ECP address FIFO
001h	DSR	Write/read	all	Status register
002h	DCR	Write/read	all	Control register
400h	C-FIFO	Write/read	2	Parallel port data FIFO
400h	ECP-D-FIFO	Write/read	3	ECP data FIFO
400h	T-FIFO	Write/read	6	Test FIFO
400h	CNFG-A	Read	7	Configuration register A
401h	CNFG-B	Write/read	7	Configuration register B
402h	ECR	Write/read	all	Extended control register

SEÑAL SPP	NOMBRE	In/Out	DESCRIPCIÓN
STROBE	HostClk	OUT	Usado con PeriphAck para transmitir datos o direcciones en la dirección directa.
AUTOFEED	HostAck	OUT	Proporciona estado de datos y de comando en la dirección directa. Usado con PeriphClk transfiere datos en la dirección inversa.
SELECTIN	1284Active	OUT	Cuando el ordenadores esta en el modo de transmisión 1284 se activa.
INIT	ReverseRequest	OUT	Se desactiva para colocar el canal en dirección inversa.
ACK	PeriphClk	IN	Usado con HostAck para transmisión de datos en la dirección inversa.
BUSY	PeriphAck	IN	Usado con HostClk para transmisión de información de datos o direcciones en la dirección directa. Proporciona estado de comandos y datos en la dirección inversa.
PE	AckReverse	IN	Desactivado para reconocer Reverse Request.
SELECT	Xflag	IN	Flag de extensibilidad.
ERROR	PeriphRequest	IN	Desactivado por el periférico para indicar que es posible la transferencia inversa.
Data[8:1	Data[8:1	BI-DI	Usado para proporcionar datos entre el periférico y el ordenador.

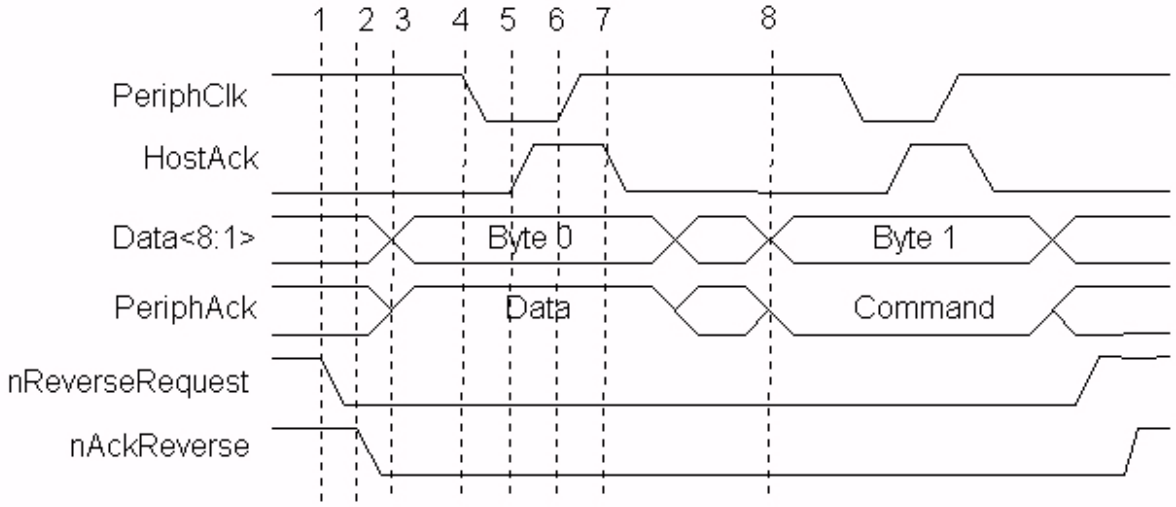
Puerto paralelo IEEE 1284

Puerto con Capacidades Mejoradas



Durante una escritura la señal HostAck esta activado indica que un ciclo de datos se esta llevando a cabo. Cuando HostAck esta desactivado se lleva a cabo un ciclo de comandos y los datos representan un calculo de RLE o un canal de direcciones. El bit 8 se usa para indicar una RLE o una dirección. Si es cero los bits restantes son un calculo de la longitud de cadena y si es 1 son una dirección.

Los cambios en la dirección de transferencias de datos deben ser negociados. Para realizar una lectura, el ordenador debe pedir desactivar ReverseRequest y esperar que el periférico reconozca la señal desactivando AckReverse. Solo entonces una lectura puede llevarse a cabo.



Puertos serie

Puerto serie

Un puerto serie o puerto en serie es una interfaz de comunicaciones de datos digitales donde la información es transmitida un solo bit a la vez; en contraste con el puerto paralelo que envía varios bits simultáneamente.

Mientras que las interfaces tales como Ethernet, FireWire y USB envían datos como un flujo en serie, el término ***puerto serie*** generalmente identifica al hardware compatible con el estándar RS-232, destinado a interactuar con un módem o con un dispositivo de comunicación similar .

El puerto serie estandar es muy común en los tipos de microcontroladores y se utiliza con dispositivos especializados, .

- Configuración y administración de equipos de red;
- Receptores GPS; escáneres de códigos de barras y otros dispositivos comerciales;
- Equipos de prueba y medición;
- Actualización de firmware en varios dispositivos de consumo masivo;.
- Controladores de Controles Numéricos Computarizados (CNC);
- Depuradores de software que se ejecutan en una segunda computadora; y
- Buses industriales; entre otros.

Dado que las señales de control para un puerto serie pueden activarse y desactivarse fácilmente mediante un interruptor, algunas aplicaciones utilizan las líneas de control de un puerto serie para monitorear dispositivos externos, sin intercambiar datos en serie.

Puerto serie

El puerto paralelo:

- Es teóricamente más rápido
- Los datos no necesitan ser preprocesados
- Es menos flexible
- Es más costoso
- Tiene predisposición a sufrir errores en distancias largas

El puerto serie:

- Es teóricamente más lento
- Los datos deben ser preprocesados (serializados/deserializados)
- Es más flexible
- Es mucho menos costoso por su menor número de líneas
- Tiene una menor probabilidadsición a sufrir errores, incluso en distancias largas

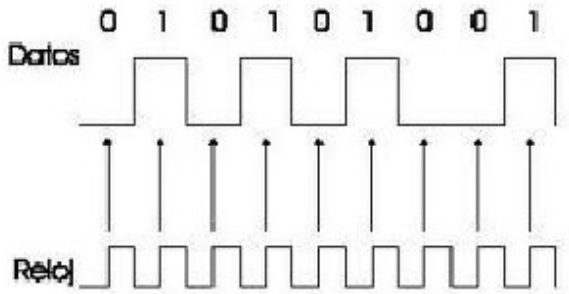
Puerto serie

Transmisión síncrona

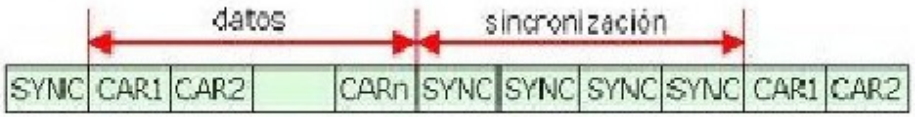
En una transferencia serial síncrona se intercambian una o varias señales de control, junto con los paquetes de informacion, entre emisor y receptor.

Las señales de control determinan cuando hay un bit de datos válido en la línea de transmisión

- Puede existir una señal de reloj que controle la sincronización; y/o
- La sincronización se realiza a través de un protocolo solicitud – reconocimiento o estructuras de información predeterminadas.



La transmisión es controlada emisor. La información se transmite entre dos grupos de datos denominados ***delimitadores***.

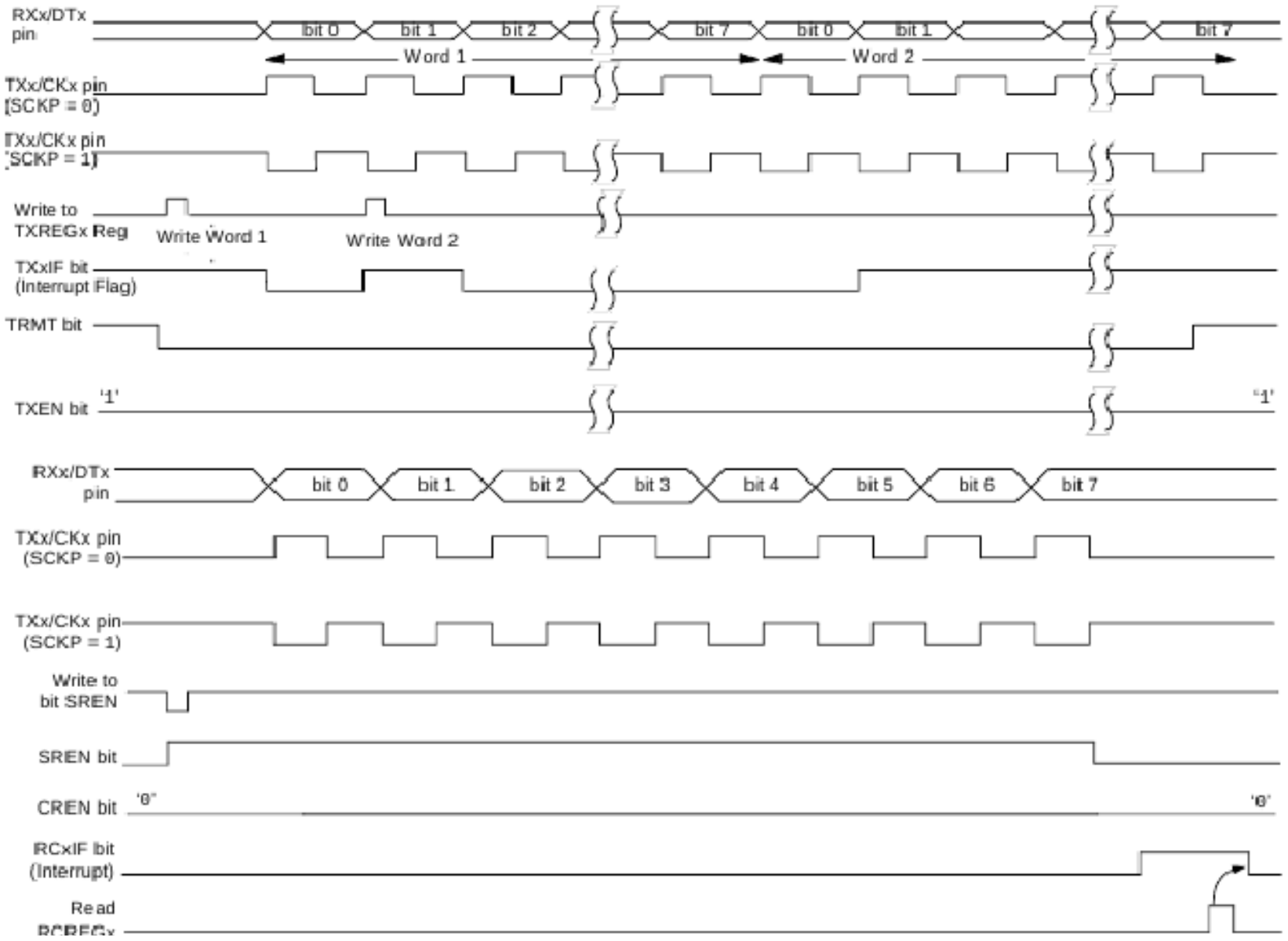


Ventajas Posee un alto rendimiento;
Operan a altas velocidades;
El flujo de datos es regular.

Desventajas Los equipamientos son más complejos y de costosos.

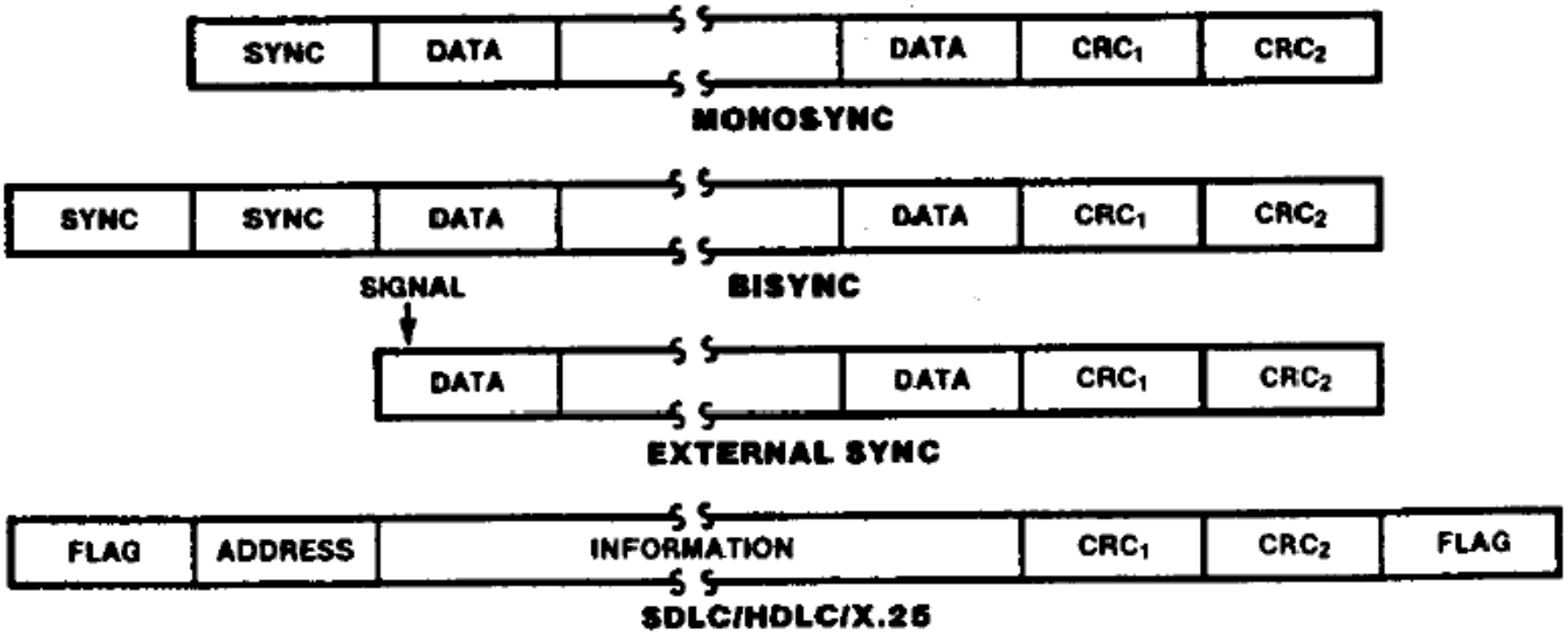
Puerto serie

Transmisión síncrona



Puerto serie

Transmisión síncrona

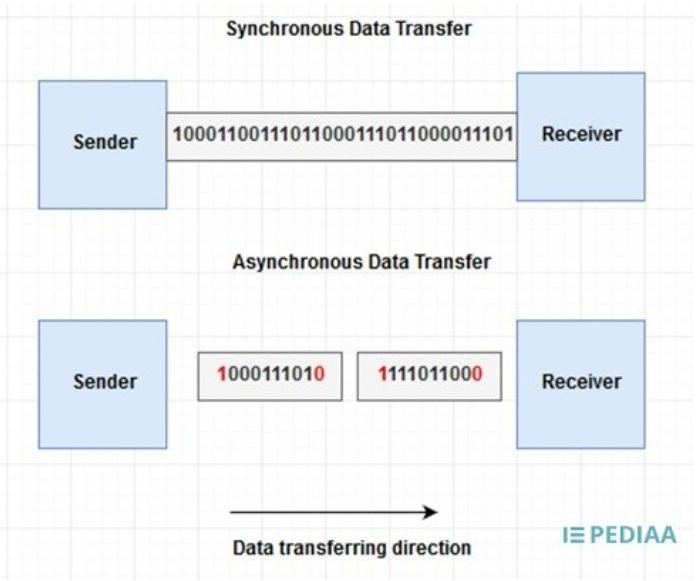


Puerto serie

Transmisión asíncrona

Una transferencia asíncrona ocurre cuando no hay relac transmisor y el receptor. La sincronización se realiza en c estructuras de datos.

El ritmo de presentación de la información en el recepto transmisor. No es necesario garantizar un ancho de bar que se disponga.



Por ello es necesario que

- Los datos contengan la información de temporización;
- El receptor muestre la línea a intervalos regulares y may llegada de datos.

Para implementar un sistema asicrónico se deben acordar los parámetros de señalización

- Tipo de operación (full o half-duplex) a realizar;
- Tamaño del dato (cantidad de bits por carácter) y el orden en que van a ser enviados;
- La velocidad de transmisión;
- El código de detección-corrección de errores utilizado; y
- La estructura de información de sincronización.

Puerto serie

Transmisión asíncrona

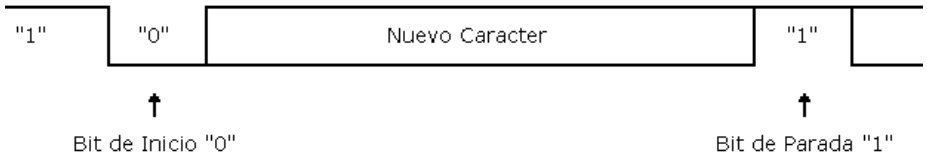
El proceso de transmisión serial toma los datos organizados en bytes y los transmite en forma de una secuencia de bits. En el destino, los bits transmitidos datos son reorganizados en bytes. Cada unidad de transmisión contiene un registro de desplazamiento, el cual es el método de conversión paralelo-serie.

La transmisión serie de información digital (bits) a través de un cable, o cualquier otro medio, es menos costoso que la transmisión paralela.

La comunicación puede ser ***simplex*** (en una única dirección, sin disposición para que el dispositivo receptor envíe información de vuelta al dispositivo de transmisión), full duplex (ambos dispositivos reciben y transmiten al mismo tiempo) o half duplex (cada dispositivo se toma su turno para transmitir y recibir).

La transmisión se realiza a nivel de bloques de datos (SDU)

- Un bit de comienzo (start) indica el principio de un SDU
- Uno o dos bits de final (stop) indica que el SDU ha terminado
- Se pueden añadir bits para el control de errores



Puerto serie

Transmisión asíncrona

La transferencia síncrona

- Permite mayores velocidades de transmisión;
- Permite que el receptor pueda interactuar con emisores de frecuencia de reloj variada;
- Permite interconectar una menor variabilidad de dispositivos, ya que emisor y receptor deben cumplir con el mismo protocolo de transmisión.

La transferencia asíncrona

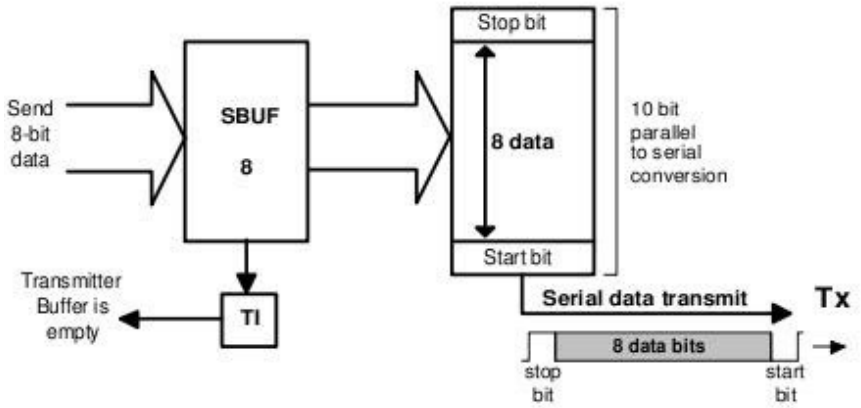
- Es más lenta debido a la información adicional y los de captura;
- En caso de errores se recupera la transmisión;
- Bajo rendimiento de transmisión;
- Equipamo económico y robusto;
- Aplicable a transmisiones con flujo de datos irregular; y
- Permite interconectar una mayor variabilidad de dispositivos.

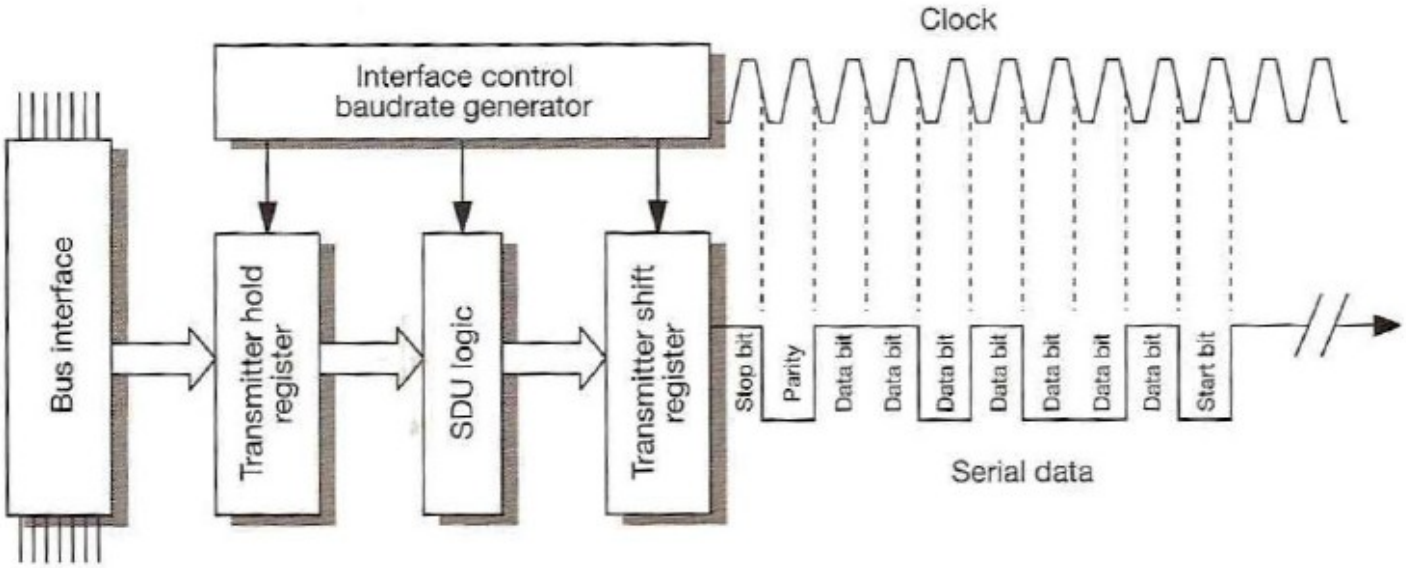
Puerto serie

Transmisión asíncrona - Transmisión

Dado un dato, la circuitería de procesamiento de datos organiza el paquete de datos (SDU) colocando la informacion necesaria (el bit de comienzo, paridad, dato y bits de final).

El bloque de datos generado se almacena en un buffer que alimenta un registro de desplazamiento que actúa de acuerdo con una señal de reloj.



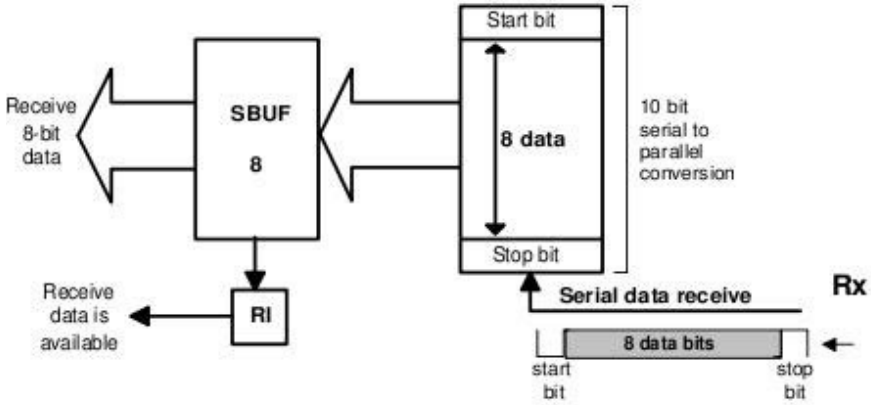


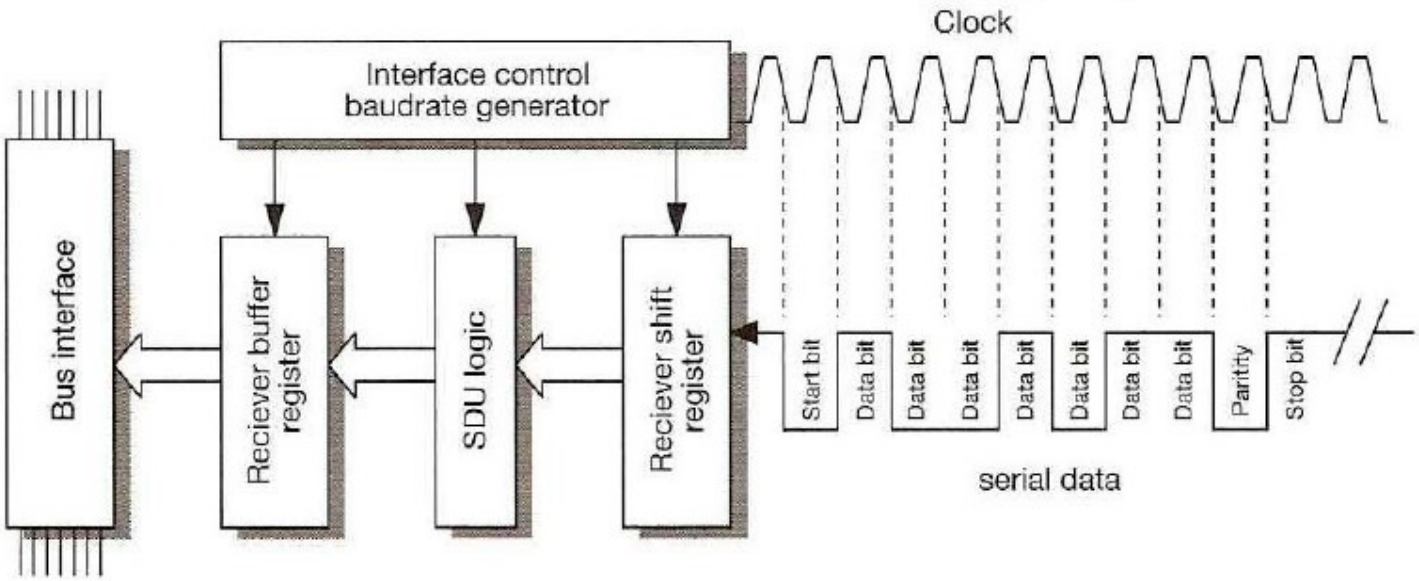
Puerto serie

Transmisión asíncrona - Recepción

Cuando se recibe el bit de comienzo, un registro de desplazamiento recompone el dato a partir de los bits recibidos siguiendo el ritmo del reloj.

Una vez que el dato completo está disponible, la circuitería SDU obtiene los bits de datos, comprueba que no haya errores y lo transfiere al buffer.





Puerto serie

Transmisión asíncrona - Errores

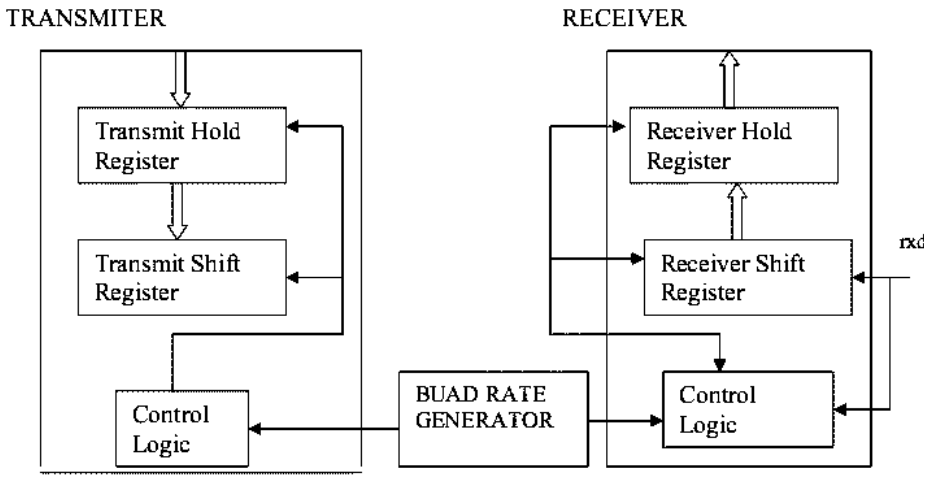
Durante la transmisión de un pueden darse diversos tipos de errores

- **Error de estructura (frame):** si el receptor detecta un bit de final incorrecto significa que el bloque de datos recibido no se ha ajustado a la estructura pactada al inicio de la transmisión.
- **Error de rotura (break):** si el receptor detecta que la línea está a un valor distinto del por defecto durante más tiempo que la duración de un bloque de datos significa que la conexión se ha roto.
- **Error de sobrescritura (overrun):** si los datos llegan demasiado rápido, un bloque de datos recompuesto pero no leído puede ser sobrescrito por uno posterior.
- **Error de paridad (parity):** si ninguno de los errores anteriores ha ocurrido, el bloque de datos ha llegado correctamente; sin embargo, todavía es necesario comprobar si no hay en el bloque mismo.

La unidad Universal Serie Síncrona/Asícrona de Transmisión/Recepción (*USART*)

Una unidad universal serie de transmisión/recepción sincróna/asíncrona (USART) es un dispositivo de hardware de computadora para la comunicación en serie en el que el formato de datos y velocidades de transmisión configurables.

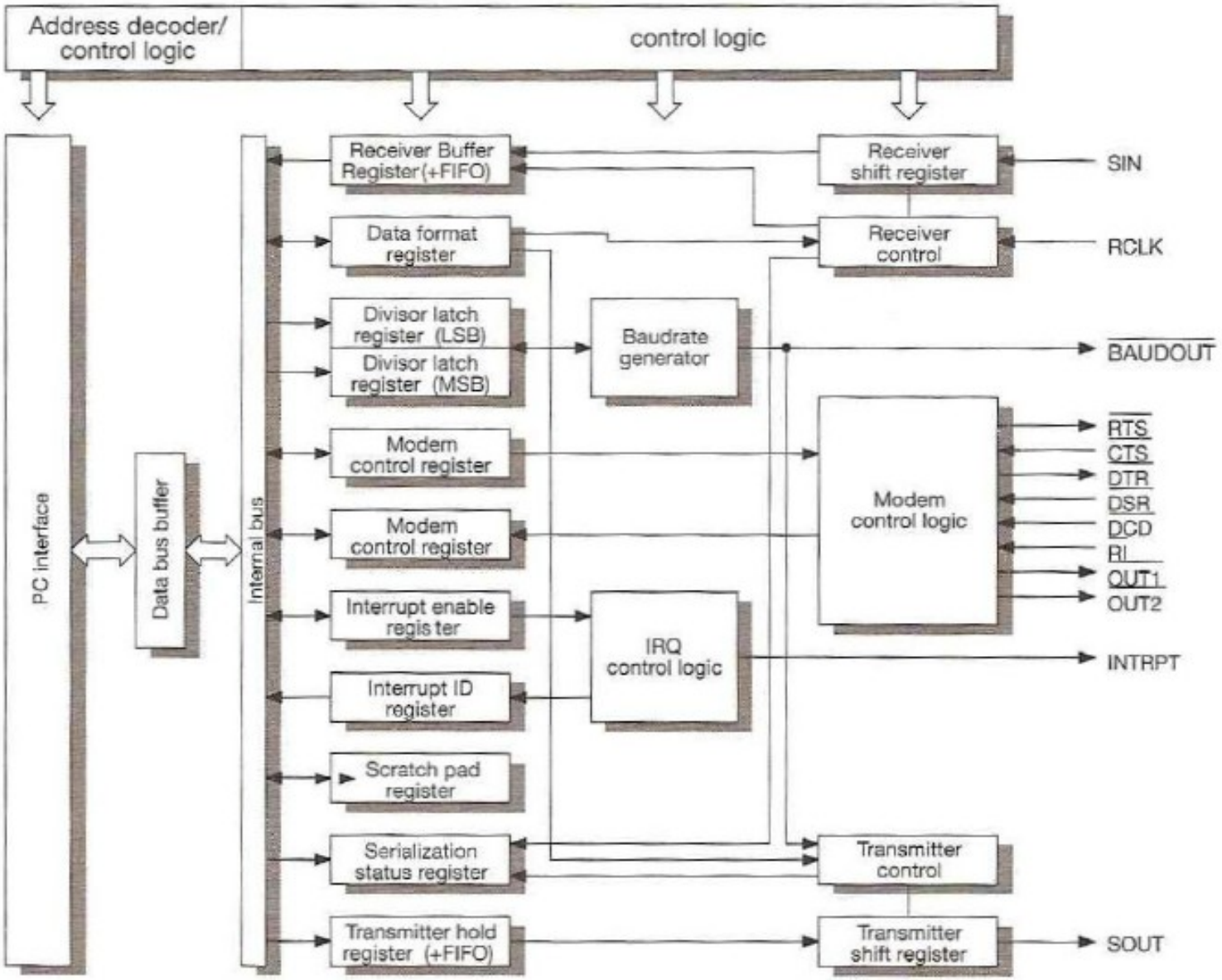
Las capacidades síncronas de USART estan destinadas a soportar protocolos síncronos, comunicaciones síncronas binarias, el control de enlace de datos síncrono (SDLC) y el control de enlace de datos de alto nivel estándar ISO (HDLC). La transmisión síncrona utiliza solo un poco más del 80% del ancho de banda de la transmisión asíncrona.



Una USART es un emisor/receptor programable cuyas funciones son

- **Conversión paralelo-serie -** serializar datos internos del computador para transmitirlos a través de una línea serie;
- **Conversión serie-paralelo -** recibir transmisiones serie y restituir los datos a su forma original;
- **Generacion del reloj** utilizado para la transmision y recepcion de los datos;
- **Generacion y validacion de los protocolos** de transmision.

La unidad Universal Serie de Transmisión/Recepción Síncrona/Asícrona (*USART*)

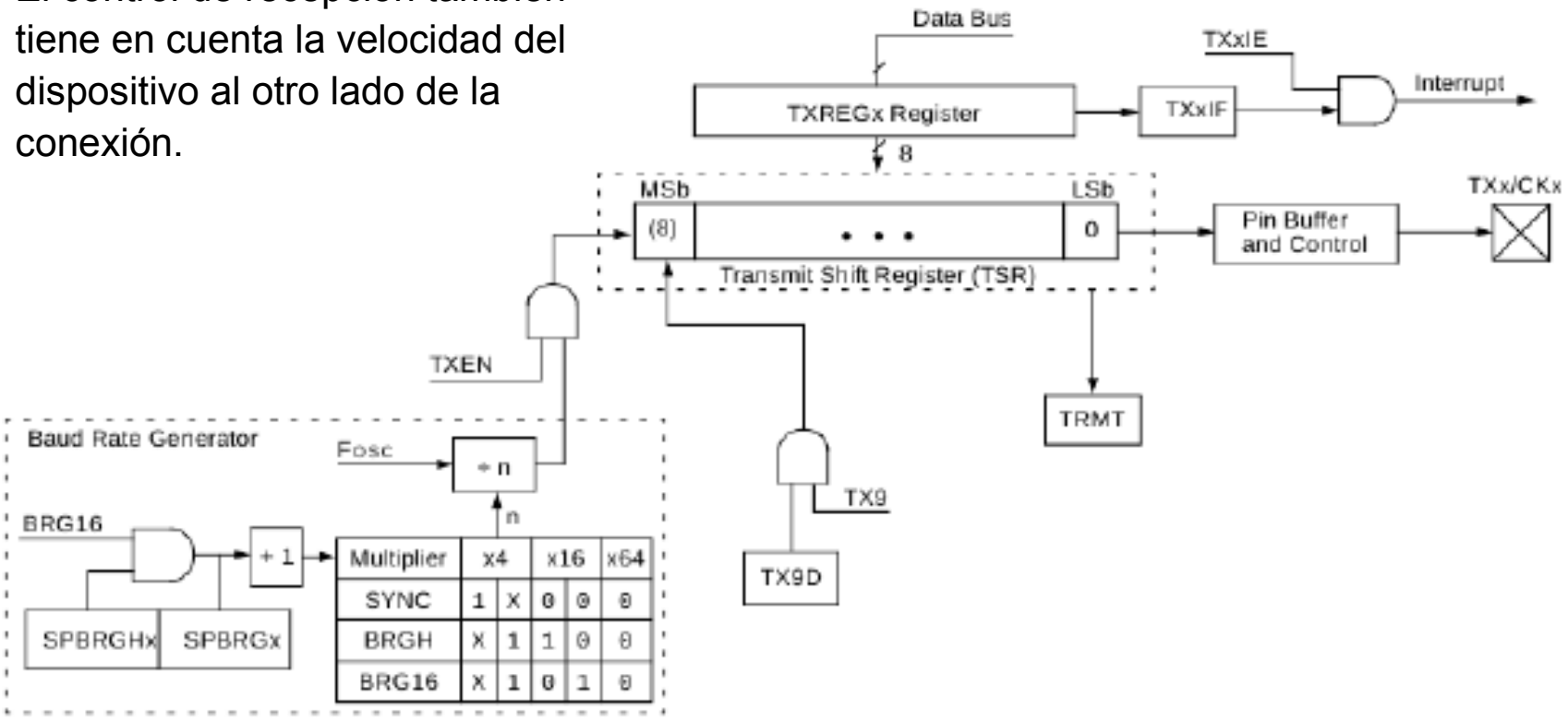


La USART

Bloque de transmisión

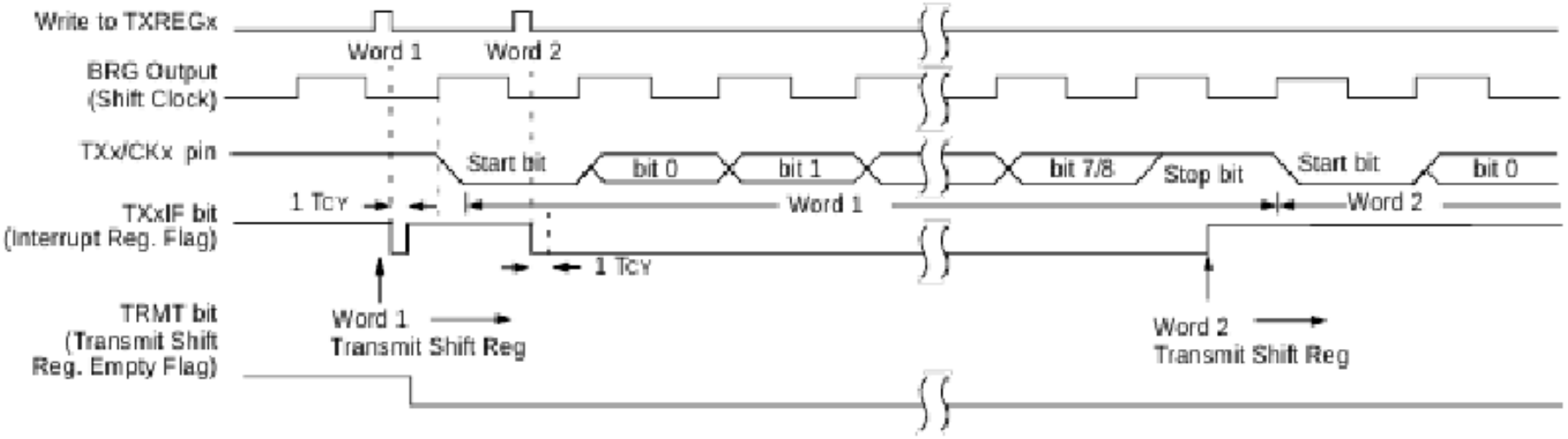
El control de transmisión toma el dato almacenado en el registro o FIFO

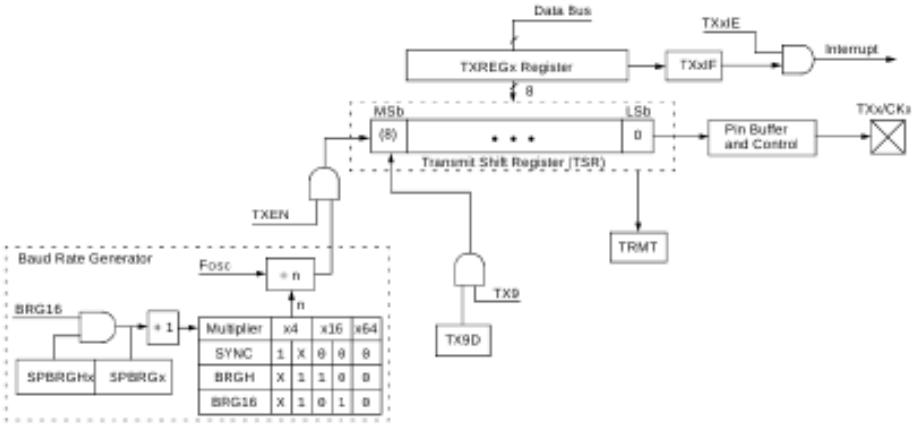
- Se incorporan los bits de comienzo, fin y paridad
- Un registro de desplazamiento permite serializar el dato para su transmisión
- La velocidad de emisión condiciona el funcionamiento del control de emisión.
- El control de recepción también tiene en cuenta la velocidad del dispositivo al otro lado de la conexión.



La USART

Bloque de transmisión



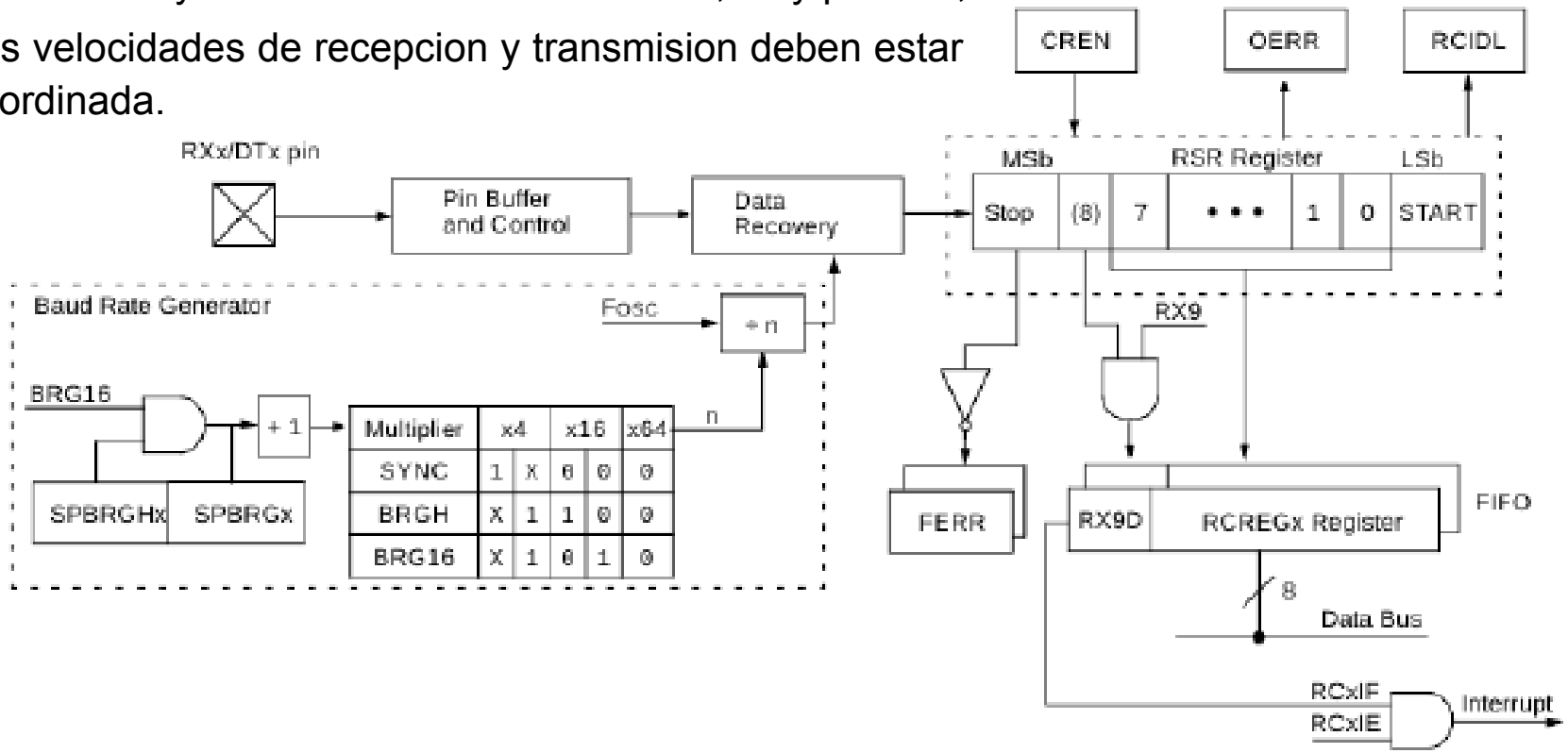


La USART

Bloque de recepción

El control de recepción toma el dato almacenado en el registro de emisión y lo transfiere al registro o FIFO.

- Un registro de desplazamiento permite paralelizar el dato para su recepción;
- Se detectan y extraen los bits de comienzo, fin y paridad;
- Las velocidades de recepcion y transmision deben estar coordinada.



La USART

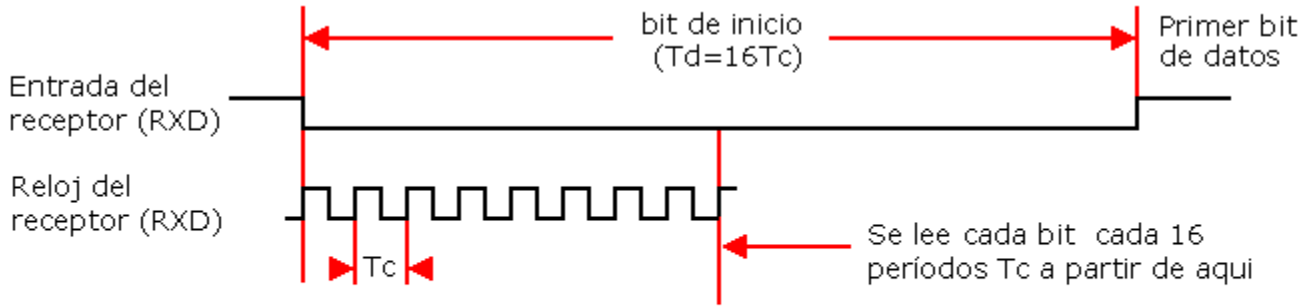
Bloque de transmisión

Para realizar la sincronización del dato recibido se debe comprobar el bit en la mitad del intervalo del tiempo que dura para evitar la lectura de falsas transiciones producto del ruido en la línea. Para la sincronización se utiliza un reloj externo de período TC que cumple la relación

$$Td = K * Tc.$$

donde Td es el periodo de un bit transmitido y Tc es el perodo del reloj.

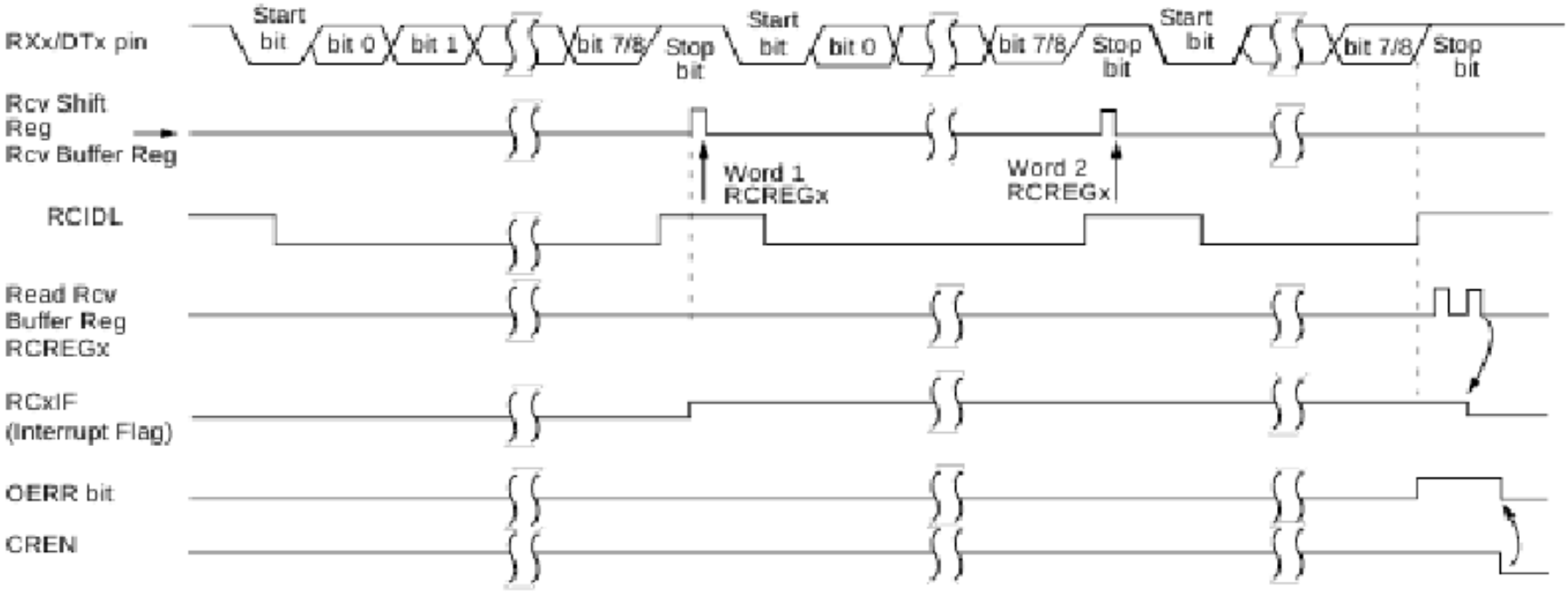
Para lograr la sincronización entre el transmisor y el receptor tanto Tc como K deben ser el mismo para ambos, ello permitirá que el bit de datos se compruebe en el momento preciso sin necesidad de conectar una línea adicional de reloj para lograr el sincronismo.

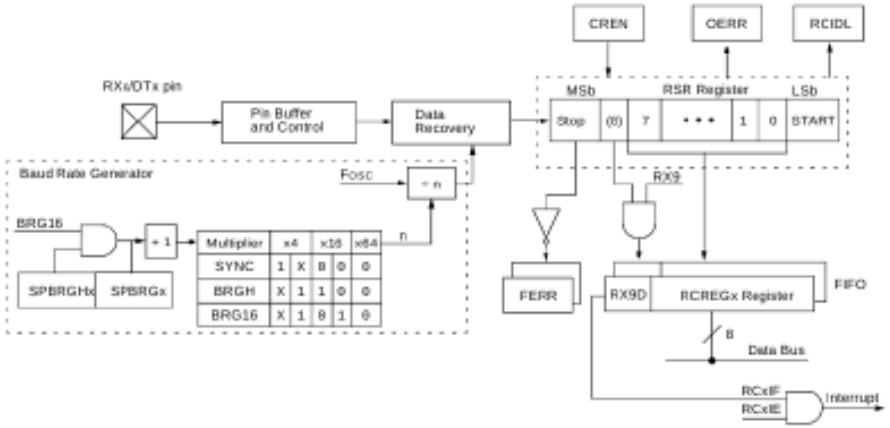


Deteccion del bit de inicio.

La USART

Bloque de recepción



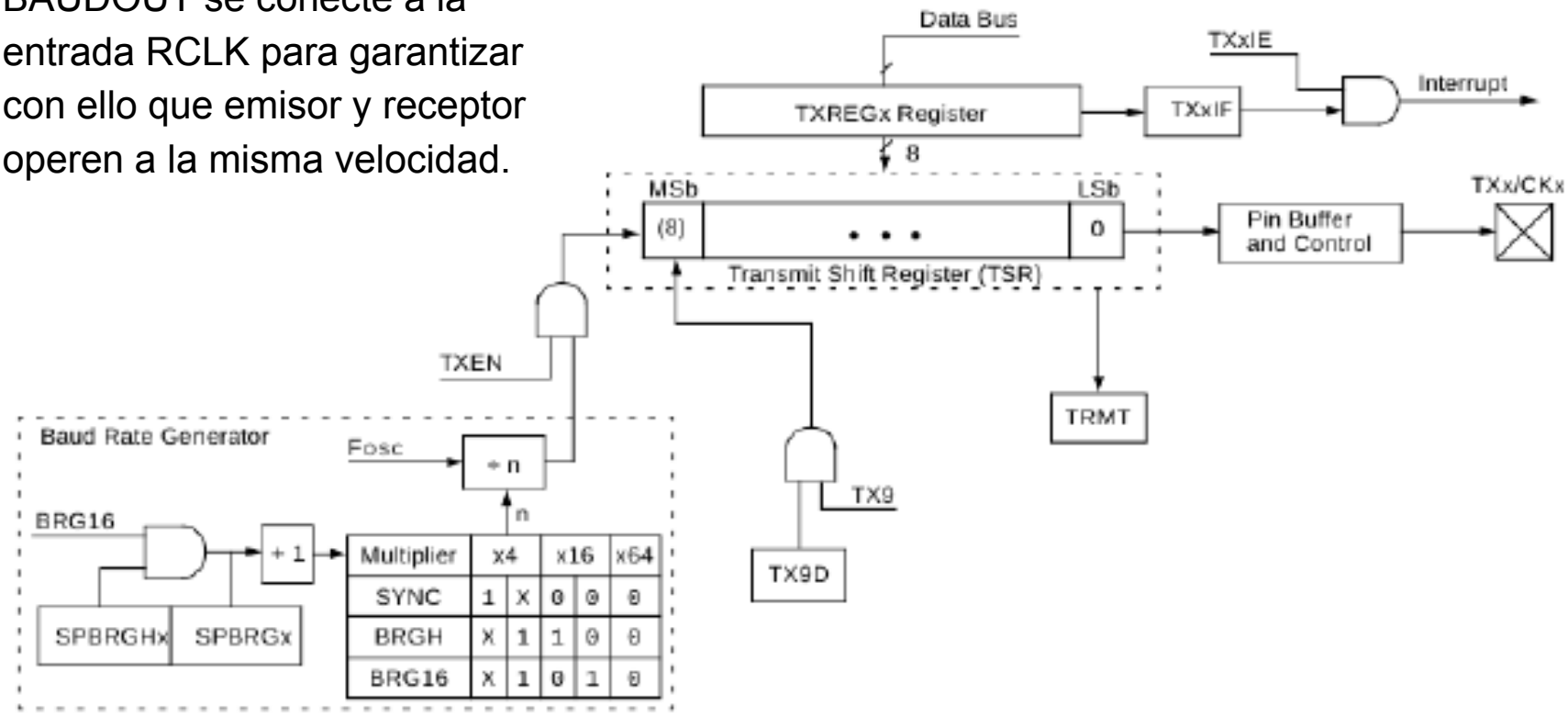


La USART

Bloque de temporización

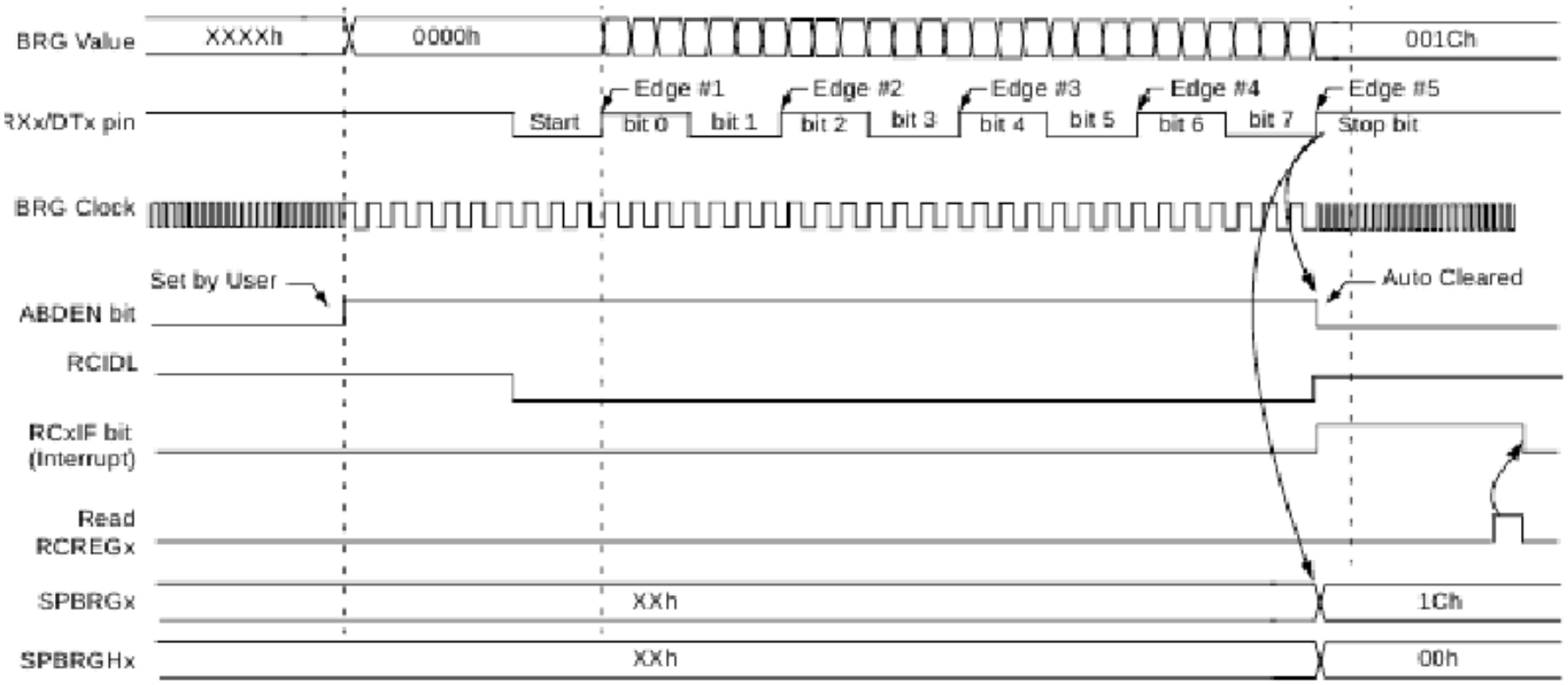
La frecuencia de referencia proviene de un oscilador externo y se divide por el contenido del registro divisor.

- El registro divisor permite configurar la UART para operar a distintas velocidades.
- Es frecuente que la salida BAUDOUT se conecte a la entrada RCLK para garantizar con ello que emisor y receptor operen a la misma velocidad.



La USART

Bloque de temporización

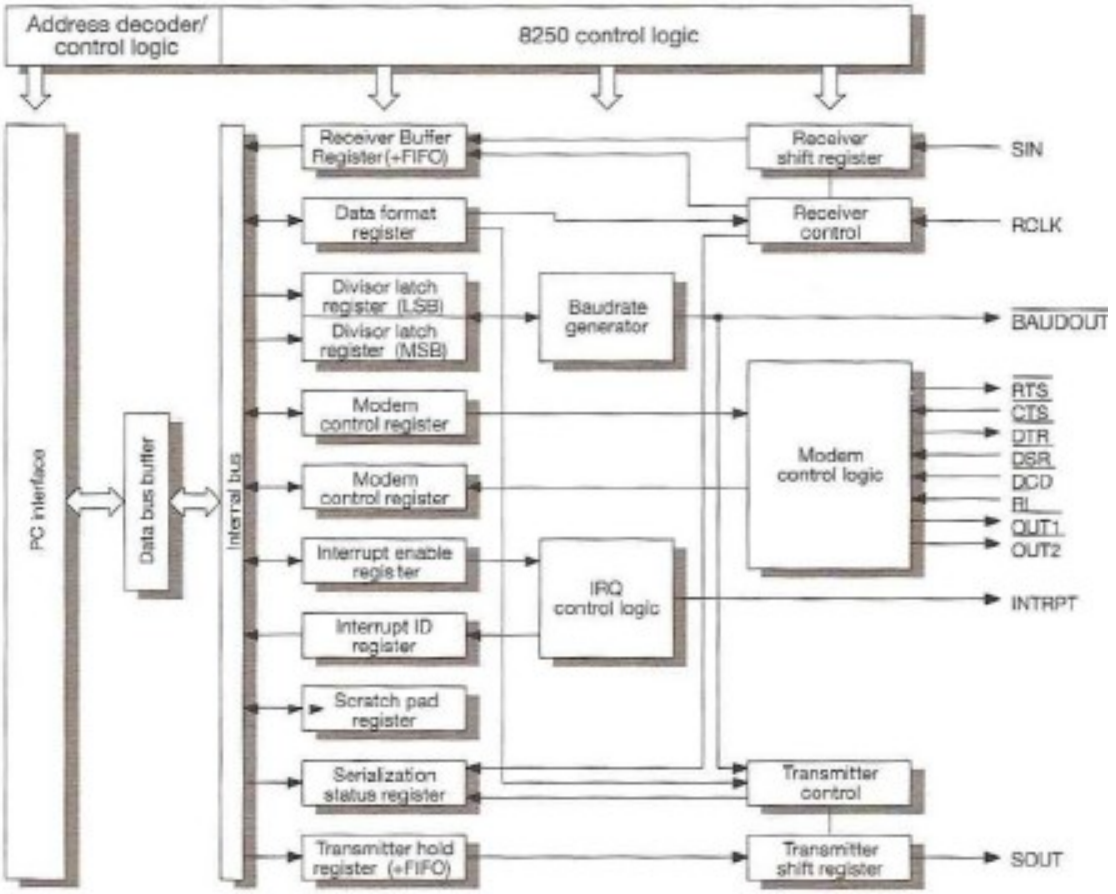


La USART

Bloque de control de modem

La lógica de control del módem se encarga de gestionar las señales de los protocolos de comunicación RS.

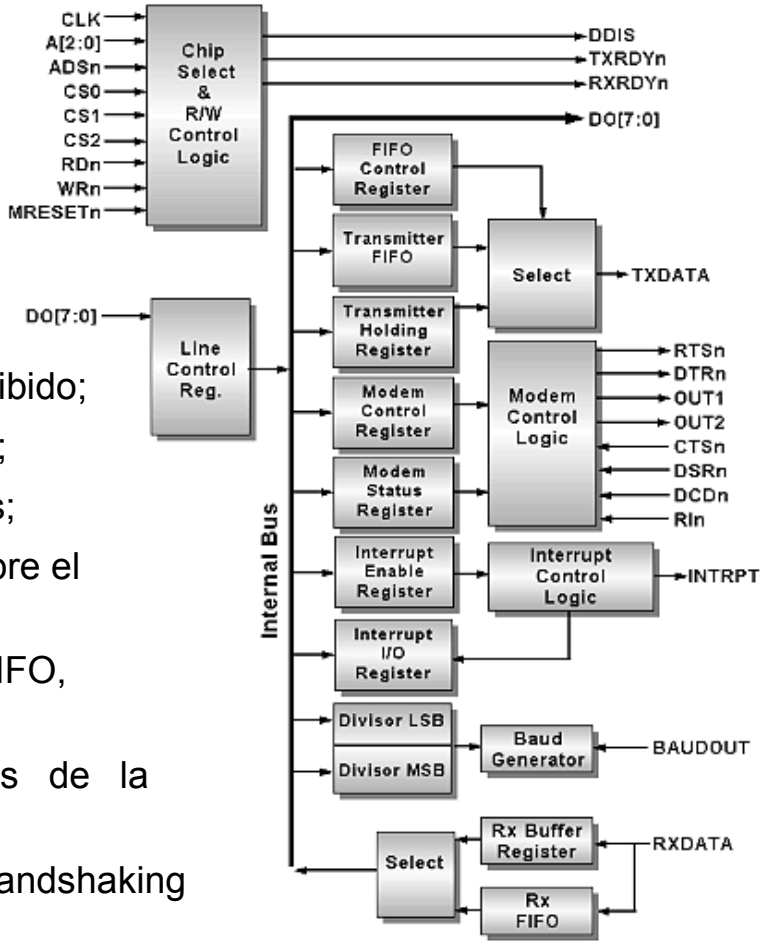
- Sin y Sout se corresponden con RD y TD;
- DTR, DSR, DCD RTS, CTS y RI implementan las señales de control del protocolo.



8250 - UART de INTEL

El UART 8250 es un circuito integrado diseñado para implementar la interfaz de las comunicaciones seriales asincronas. Incluye un generador de bit rate programable, permitiendo el uso tanto de los bit rates comunes como los de propósitos especiales.

Los registros disponibles son



Buffer de recepción (RBR) – contiene el último carácter recibido;

Buffer de transmisión (THR) – almacena el carácter saliente;

Habilitación de interrupción (IER) – habilita las interruciones;

Identificación de interrupción (IIR) – provee informacion sobre el estado de la UART y las causas de la interrupciones;

Control de FIFO (FCR) - controla el comportamiento de la FIFO, incluyendo la seleccion de modos de operacion;

Control de la línea (LCR) – establece los parametros de la comunicacion;

Control del modem (MCR) - se utiliza para las acciones de handshaking con el dispositivo asociado;

Estado de la línea (LSR) – muestra el estado de la comunicación, reflejando los errores;

Estado del modem (MSR) – muestra el estado de las líneas de estado entrantes. Los cuatro bits mas siginificativos indican el estado mientras que los otros indican los cambios en las lineas;

Registro de scratch (SCR) – se utiliza paraalmacenar informacion;

Registros de divisor (DLL, DLM) – se utiliza para almacenar el divisor utilizado por el generador de reloj.

Z8440 - USART de Zilog

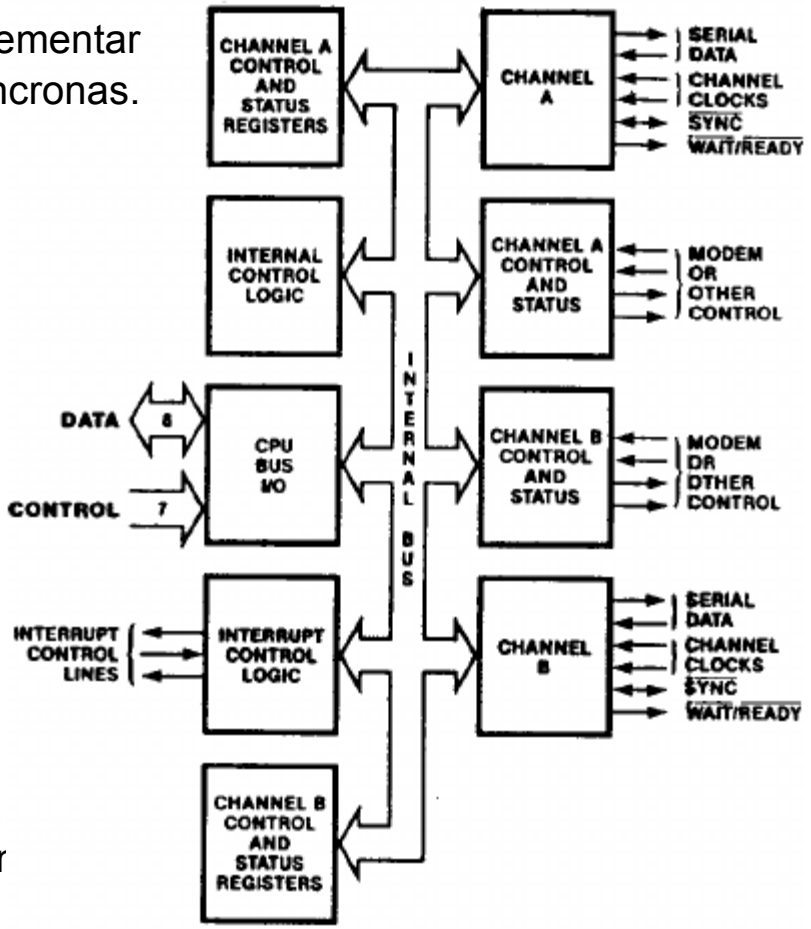
El Z8440 es un circuito integrado diseñado para implementar la interfaz de comunicaciones seriales síncronas y asíncronas.

Tiene dos canales independientes que pueden operar

Los ***modos asíncronos*** de transmisión y recepción pueden ejecutarse de manera independiente en cada canal con caracteres de longitud entre cinco y ocho bits, ademas de la paridad.

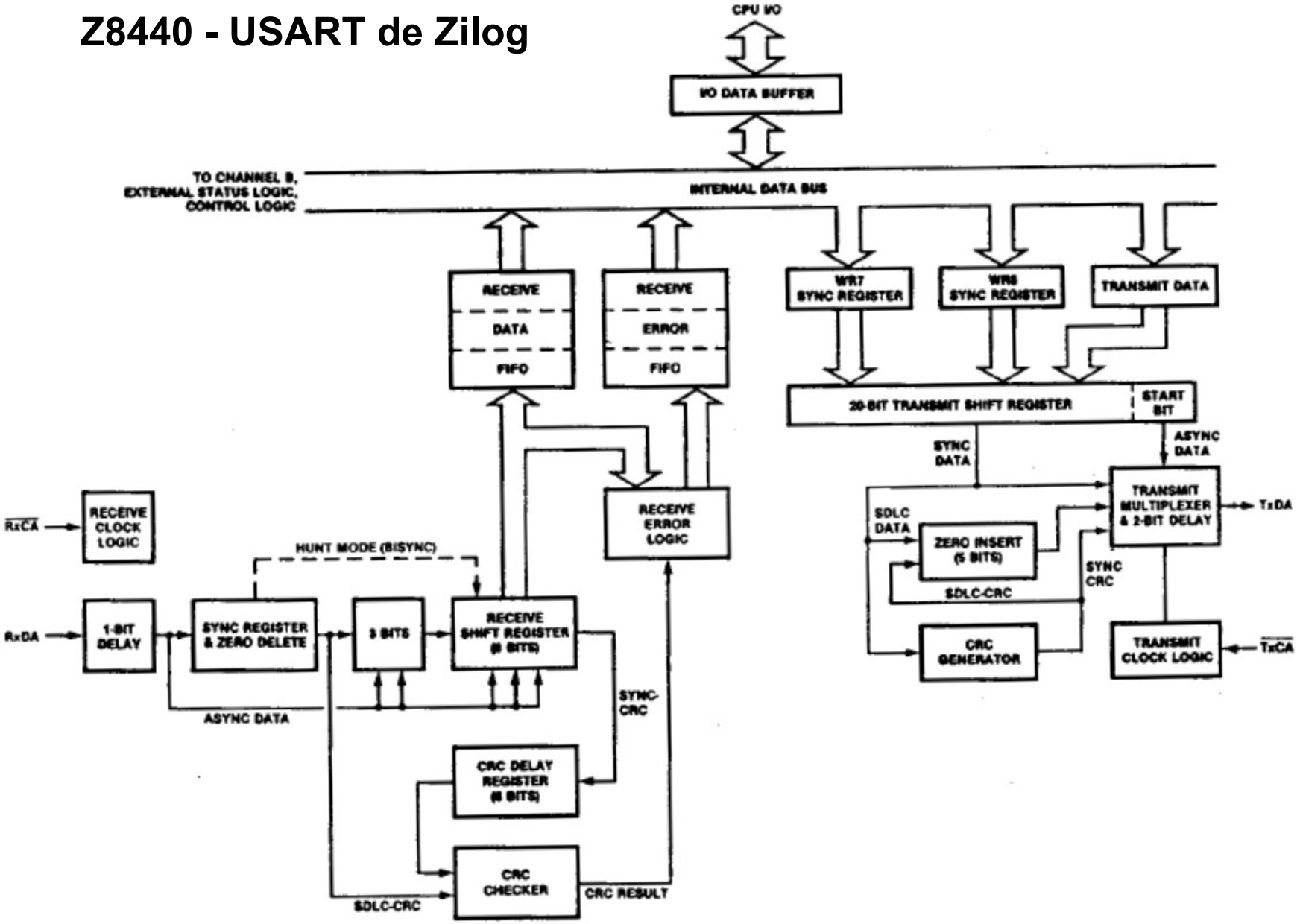
La transmisión pude incluir uno o dos bits de stop por carácter, e interrumpir la transmisión en cualquier momento. La recepción está protegida contra ruidos por un mecanismo de filtrado, e incluye la lógica de detección de interrupción de transmisión, asi como los mecanismos de deteccion de errores en el frame.

Los ***modos síncronos*** soportan protoclos orientados a bytes pueden manejar diferentes modos de sincror detección de errores.



El dispositivo incluye detección automáticas de banderas de transmisión, lo cual permite implementar protocolos orientados a bits. Un comando especial permite abrtar la transmisión.

Z8440 - USART de Zilog



Z8440 - USART de Zilog

El Z8440 de once registros por cada canal, donde ocho de ellos son solo escritura (WR0-WR7) y los tres restantes son de lectura (RR0-RR2).

El grupo de registros incluye ocho registros de control, dos de caracteres de sincronismos y dos de estados, dos de vector de interrupción.

Read Register Functions	
RR0	Transmit/Receive buffer status, interrupt status and external status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only)

Write Register Functions	
WR0	Register pointers, CRC initialize, and initialization commands for the various modes.
WR1	Transmit/Receive interrupt and data transfer mode definition.
WR2	Interrupt vector (Channel B only)
WR3	Receive parameters and control
WR4	Transmit/Receive miscellaneous parameters and modes
WR5	Transmit parameters and controls
WR6	Sync character or SDLC address field
WR7	Sync character or SDLC flag

Z8440 - USART de Zilog

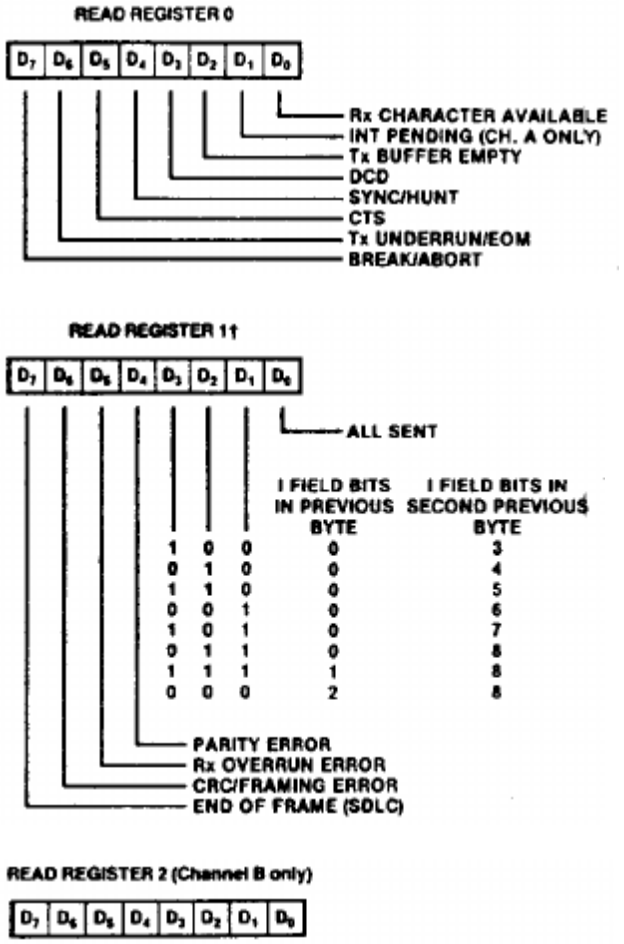
Los ***registros de lectura*** permiten obtener la información sobre el estado de cada canal serie. El puerto A tiene solo dos registros (RR0 y RR1), mientras que el puerto B tiene un registro extra (RR2), el cual contiene el vector de interrupciones que es modificado por el puerto de acuerdo a la condición que produce la interrupción.

La información del estado de cada canal incluye condiciones de error, vector de interrupción y las señales estandard para implementar interfaces de comunicación.

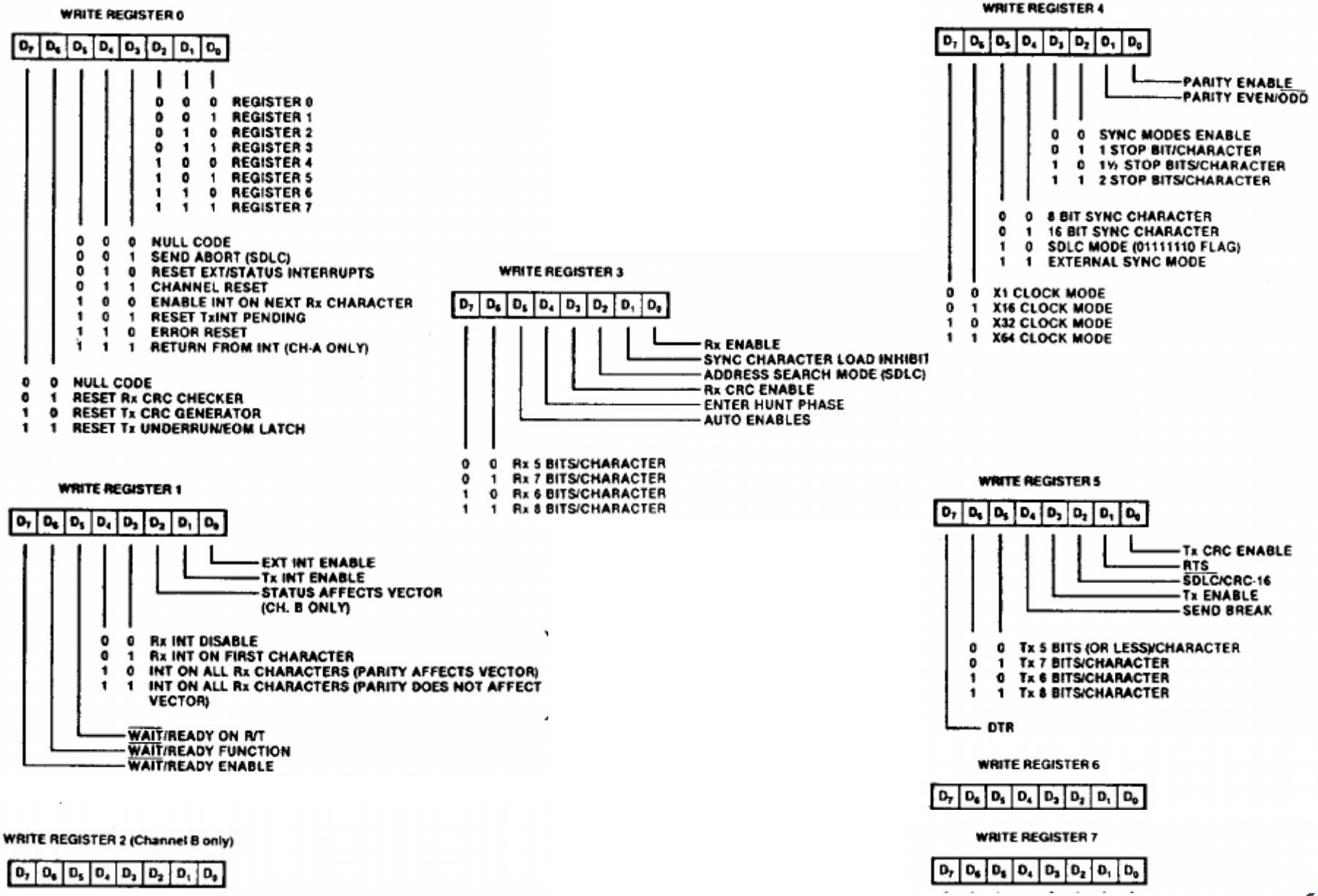
Para leer el contenido de culauier registro primero se debe escribir el puntero de registros en WR0 (D0-D2), luego se ejecuta la instrucción de lectura.

Los bits de estados en RR0 y RR1 están agrupados de manera de simplificar el monitoreo de las diferentes condiciones.

Los ***registros de escritura*** permiten configurar cada canal de manera independiente. El registro WR2 contiene el vector de interrupciones para ambos canales y los registros WR6-WR7 son los caracteres de sincronismos. Para ejecutar cualquier operación se necesitan dos bytes: uno para direccionar el registro (WR0) y despues el dato.



Z8440 - USART de Zilog



Puerto Firewire - IEEE 1394

El puerto Firewire

El puerto IEEE 1394 (Firewire) es una interface estandar para puertos series de comunicación para la transferencia de datos isocrona y en tiempo real.

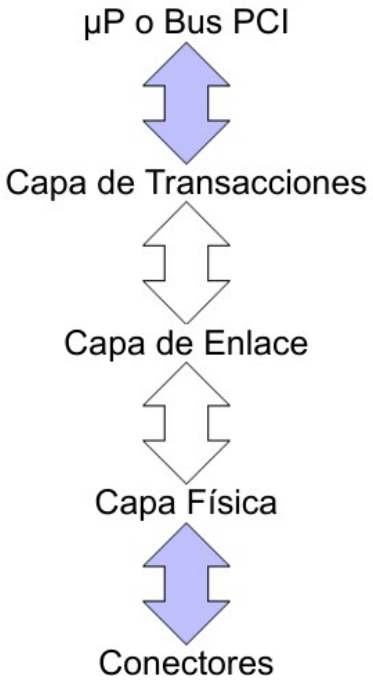
Está basado en el estandar ISO/IEC 13213:1994 (ANSI/IEEE 1212) que describe la arquitectura de comunicaciones entre buses de microcomputadoras.

El puerto Firewire fue diseñado para obtener un alto desempeño, especialmente en aplicaciones críticas en tiempo, cuyas mejoras en el desempeño se obtienen a partir del uso de técnicas de bajo nivel como acceso directo a memoria.

El Firewire soporta el modelo peer-to-peer, en la que cualquier dispositivo puede comunicarse directamente can cualquier otro, siempre que utilicen los mismos protoclos. Los dispositivos IEC 13213 incorporan una ROM de configuración con detalles sobre sus posibilidades, funcionalidades, protocolos que soportan, entre otros, de forma que pueden comunicar esta información a otros dispositivos conectados al bus.

Las especificaciones sobre los protocolos, y los formatos de datos correspondientes, se definen en una serie de estandares. Las facilidades plug-and-play, hot-swapping y la memoria de configuración permiten la coexistencia transparente de distintos protocolos, donde la única limitación es el ancho de banda disponibles.

El puerto Firewire



El tráfico de información en sistemas de tiempo real es muy sensible al tiempo, por este motivo, el protocolo FireWire dispone de dos modos de transmisión

- **Modo asíncrono** - se utiliza para comunicar dispositivos que no necesitan ancho de banda elevados. En este modo se envían cantidades arbitrarias de datos e información de la capa de transacción al nodo destino, seguido por una confirmación del este nodo; y
- **Modo isócrono** - garantiza una tasa de transferencia predeterminada para cada dispositivo sin que se produzcan interrupciones en el flujo de datos. En este modo se envía una cantidad arbitraria de datos a intervalos regulares a un nodo destino y sin requerir confirmación.

El protocolo FireWire se basa en tres capas

- **Capa de transacciones** - maneja las transferencias entre dos dispositivos a través del bus. El bus utiliza 64 bits de direccionamiento, los 16 bits más significativos son código de identificación del dispositivo. Este campo se divide en 10 bits para identificación de bus y 6 bits de desplazamiento.
- **Capa de enlace** - gestiona la transferencia de la información, la cual se se desarrolla en tres fases: Durante ***el arbitraje*** un dispositivo envía una petición de acceso al bus a la capa física. Durante ***la transmisión*** el dispositivo transmite un paquete de datos que contiene información de formato y transacción, la dirección de los dispositivos fuente y destino y los datos. ***El reconocimiento*** consiste en el envío de un código de confirmación indicando que los datos fueron correctamente recibidos.
- **Capa física** - convierte los niveles lógicos de la capa de enlace en señales eléctricas para el cable y viceversa, determina la configuración eléctrica y mecánica de la conexión y actúa como un árbitro que garantiza que los dispositivos accedan al bus cuando deseen enviar datos.

El puerto Firewire

Direccionamiento

Los dispositivos FireWire y sus recursos internos se seleccionan mediante un direccionamiento directo y jerárquico con 64 bits estructurado de la siguiente manera

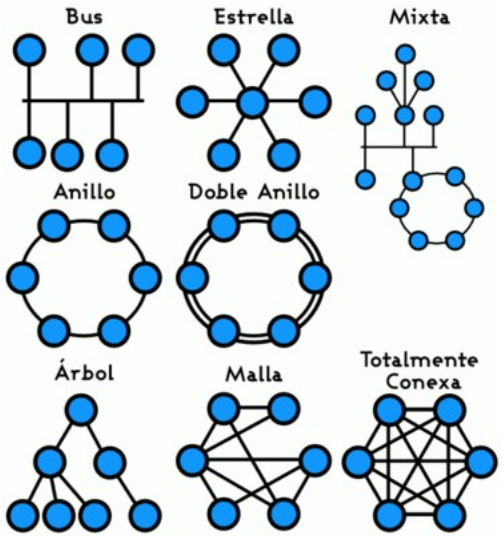
- **Dirección del nodo** (NodeID) identifica a nodo particular del bus, está conformado por 16 bits distribuidos de la siguiente manera
  - **Dirección del bus** (BusID) identifica a todos los nodos que pertenecen a los distintos buses del sistema, está conformada por 10 bits; y
  - **Dirección de nodo** (Physiscal-ID) identifica a un nodo pertenenciente a un bus particular, está conformada por 6 bits. Esta dirección es asignada a cada nodo durante el proceso de inicialización del bus.
- **Espacio Inicial de Nodo** (NodeOffset) es el espacio disponibles en cada nodo que se direcciona con 48 bits. Este espacio se divide en: i) **Espacio Inicial de Memoria**; ii) **Espacio Privado**; iii) **Espacio Inicial de Registro** y iv) **Espacio Inicial de Unidad**.

La existencia de varios buses en un mismo sistema necesita de eleementos (Bridges) para controlar y gestionar el flujo de información entre los mismos. Los Bridges pueden utilizarse para aumentar el número de nodos o para dividir el tráfico en segmentos.

Cada vez que se añade o quita un nodo al bus se asignan direcciones físicas a cada nodo. Los dispositivos no disponen de conmutadores de configuración, y además soporta conexión en caliente (hot-plug). La parte física que gestiona la interfaz en los dispositivos se denomina PHY, que imlementa la Configuración Automática de la Red.

El puerto Firewire

Topología



FireWire puede conectar hasta 63 periféricos por bus en una topología ***árbol*** o ***cadena*** (daisy-chain) que permite la comunicación entre dispositivos de igual a igual sin tener que usar la memoria del sistema o el procesador.

Una red en cadena puede tomar dos formas básicas

Topología lineal – conecta dispositivos sucesivos a través de un enlace bidireccional ;

Topología en anillo – es una topología lineal a la que se conectan los dispositivos extremos. Una ventaja del anillo es la reducida cantidad de transmisores y receptores y la tolerancia a falla.

Cada dispositivo tiene una identificación única (Physical-ID) que se asigna durante el proceso de identificación, el que ocurre cada vez que se reinicia el bus. El orden en que se asignan las Physical-ID es equivalente a atravesar el árbol primero en profundidad. Uno de los nodos es elegido nodo raíz y se le asigna la Physical-ID más alta.

FireWire es capaz de operar con seguridad sistemas críticos debido a la forma en los dispositivos interactúan con el bus y cómo el bus asigna ancho de banda. FireWire es capaz de realizar métodos de transferencia asíncronos e isócronos a la vez. FireWire dedica un 80% del bus a ciclos isócronos, dejando a los datos asíncronos un mínimo del 20% del bus.

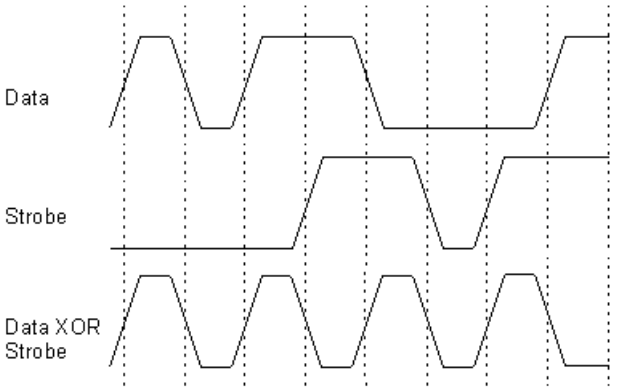
El puerto Firewire

Señalización

FireWire utiliza codificación Data / Strobe que utiliza dos señales para transmitir los datos con alta confiabilidad. La señal NRZ es procesada con una señal del reloj a través de una puerta XOR, creando la señal Strobe. Esta señal se proces a través de una XOR junto con la señal de datos para reconstruir el reloj. Esto a su vez actúa como una señal de referencia de un bucle enganchado en fase para fines de sincronización.

La ***codificación data-strobe*** es un esquema de codificación para transmitir datos en circuitos digitales que utiliza dos líneas de señal : **Datos** y **Strobe**. Estas señales tienen la propiedad de que cambian su valor lógico en cada ciclo de reloj, pero nunca en ambos. Los datos se transmiten tal cual y la señal Strobe cambia su estado si y solo si los datos permanecen constantes entre dos bits consecutivos. Esto permite una fácil recuperación del reloj con una buena tolerancia a la fluctuación de fase.

Hay una forma equivalente de especificar la relación entre Datos y Strobe. Para un número par de bits de datos, Strobe es lo opuesto a datos. Para un número impar de bits de datos, Strobe es lo mismo que datos. A partir de esta definición, es más obvio que la función XOR entre los datos y Strobe producirá una señal de reloj. Además, especifica el mecanismo más simple para generar la señal Strobe para un flujo de datos dado.



El puerto Firewire

Protocolo

El FireWire utiliza una técnica de multiplexado de paquetes. Cuando un dispositivo tiene un paquete para transmitir, arbitra por el bus y una vez gana el acceso transmite el paquete entero.

Existen dos tipos de transferencias

- **Asíncrona** – sigue la interfaz Load – Store tradicional en los sistemas que utilizan mapeado en memoria. Las peticiones de Datos se envian a direcciones de memorias concretas y ésta devuelve una indicación de Reconocimiento; y
- **Isócrona** – garantiza un ancho de banda y latencia para cada dispositivo a partir de enviar una cantidad arbitraria de datos a intervalos regulares a un nodo destino y sin requerir confirmación. Firewire soprta hasta 64 canales isócronos por sistema. Un canal isócrono es una relación entre nodos que foprman un grupo en el que hay transmisores (Talkers) y receptores (Listeners). Cada grupo se identifica por un nodo entre 0 y 63.

Cuando se (re)inicia el sistema Firewire se lige que nodos cumplen con las siguientes funciones

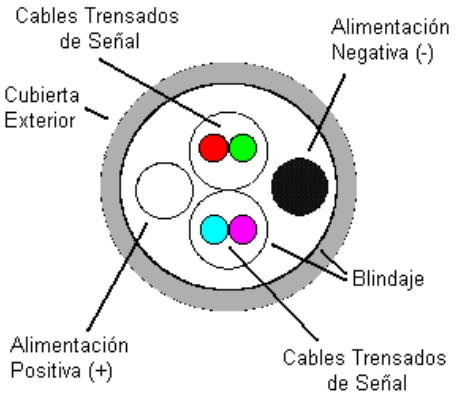
- **Root Node** – inicia ciclos en el bus cada 125 μS. El inicio de ciclo se indica mediante el paquete especial **cycle start pcket**. En cada ciclo se realizan primero las transferencia isócronas y el resto del ciclo a las transferencias asíncronas;
- **Bus Manager** – realiza la gestión de la alimentación del bus y publica la topología del bus y la velocidad máxima a la que pueden transmitir los nodos (**Mapa de Velocidad**); y
- **Isochronous Resource Manager** – realiza la asignación de los recursos isócronos.

El puerto Firewire

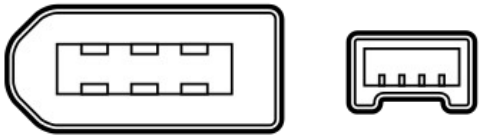
El cable

FireWire utiliza un cable de 6 hilos de cobre: 2 cables de alimentación y los otros 4 son de señal formando dos parejas de cables entrelazados. Cada pareja está apantallada al igual que el cable en su conjunto.

Los cables de alimentación transportan de 8V a 40V y proporcionan hasta 1.5A. Alimentanr directamente dispositivos conectados sin necesidad de una fuente de alimentación externa.



El conector es pequeño, flexible y de gran duración. El contacto eléctrico se establece en su interior, lo que evita riesgos.



Pin 1	TPB- (4 y 9 pines); Poder (6 pines)
Pin 2	TPB+ (4 y 9 pines); Tierra (6 pines)
Pin 3	TPA- (4 y 9 pines); TPB- (6 pines)
Pin 4	TPA+ (4 y 9 pines); TPB+ (6 pines)
Pin 5	TPA- (6 pines); A-shield (9 pines)
Pin 6	TPA+ (6 pines); Tierra (9 pines)
Pin 7	Sin conexión
Pin 8	Poder (9 pines)

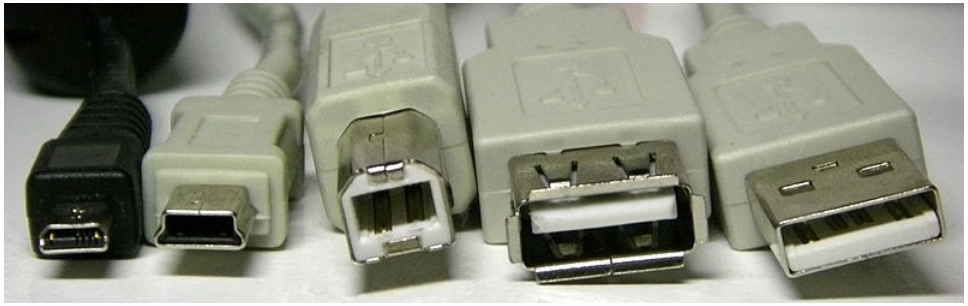
Cada dispositivo puede separarse de otro hasta 4.5 m, pudiendo aumentar esta distancia por medio de repetidores. Los dispositivos conectados al bus pueden conectarse y retirarse en cualquier momento. El bus se configura de forma automática (Plug & Play) lo que elimina la necesidad de intervención del usuario.

Puerto USB

Puerto USB

El Bus Universal en Serie (en inglés: Universal Serial Bus – USB) es un bus de comunicaciones que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre computadoras, periféricos y dispositivos electrónicos.

Este bus fue desarrollado para simplificar y mejorar la interfaz entre las computadoras personales y los dispositivos periféricos, en comparación con las interfaces existentes anteriormente. En este sentido, ha desplazado a conectores como el puerto serie, puerto paralelo, puerto de juegos, Apple Desktop Bus y PS/2.



El puerto USB es utilizado como estándar de conexión de periféricos como teclados, ratones, memorias, joysticks, escáneres, cámaras digitales, teléfonos móviles, reproductores multimedia, impresoras, dispositivos multifuncionales, sistemas de adquisición de datos, módems, tarjetas de red, tarjetas de sonido, tarjetas sintonizadoras de televisión, grabadoras de DVD y discos duros externos, entre otros.

Los conectores USB han ido reemplazando cada vez más otros tipos de cargadores de batería de dispositivos portátiles.

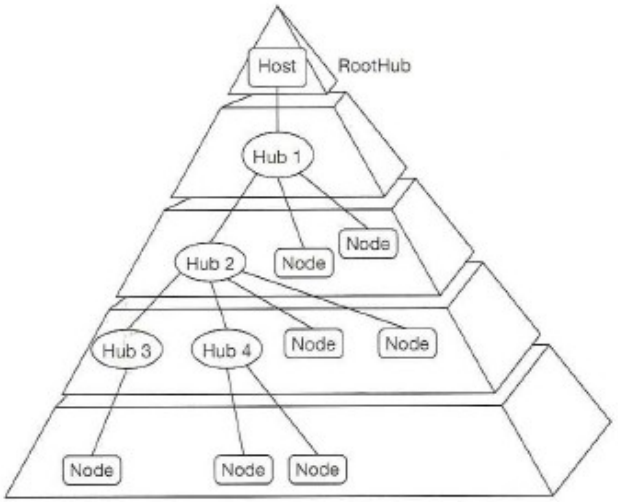
Puerto USB

Estructura

A USB system consists of a host with one or more downstream ports, and multiple peripherals, forming a tiered-star topology. Additional USB hubs may be included, allowing up to five tiers. A USB host may have multiple controllers, each with one or more ports.

USB devices are linked in series through hubs. The hub built into the host controller is called the root hub.

- Se ***reduce el número de conexiones*** necesarias;
- ***Facilita la configuración*** al permitir la conexión en caliente del dispositivo, identificando y configurando automaticamente los dispositivos conectados;
- ***Reduce la cantidad de recursos*** utilizados para gestionar los dispositivos (canales E/S, canales DMA, interrupciones, memoria, etc.); y
- ***Amplia la estructura*** del sistema y permite la conexión de un mayor número de dispositivos a partir de añadir hubs adicionales; y

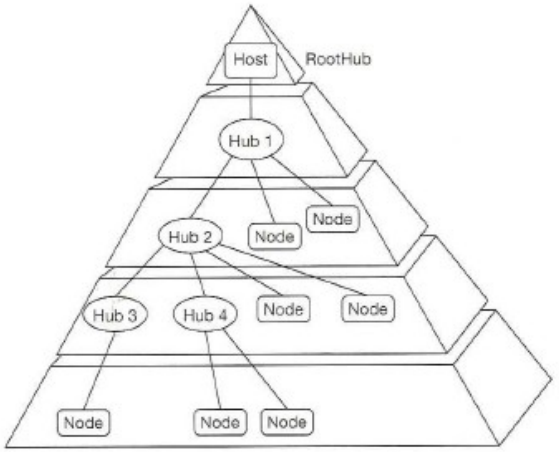


- ***Facilita su ampliación*** del sistema al poder conectar diferente tipos de dispositivos (síncronos/asíncronos, diferentes velocidades de transferencia) y proporciona alimentación; y

Puerto USB

Estructura

El bus USB tiene una estructura estratificada con forma de árbol. A USB device may consist of several logical sub-devices that are referred to as device functions. A composite device may provide several functions, for example, a webcam (video device function) with a built-in microphone (audio device function). An alternative to this is a compound device, in which the host assigns each logical device a distinct address and all logical devices connect to a built-in hub that connects to the physical USB cable.



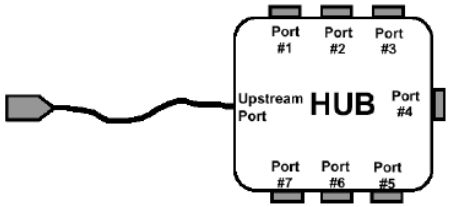
- La conexión de los dispositivos sigue un esquema encadenado (hardware polling);
- La gestión del bus es centralizada y se realiza desde el controlador integrado en el computador (host);
- Cada dispositivo USB tiene su propia dirección en el sistema
  - El controlador inicia todas las actividades y se comunica con el computador por medio de interrupciones
  - Ningún dispositivo USB puede iniciar una transacción por sí mismo para evitar sobrescribir datos presentes en el bus

Puerto USB

Estructura – El Hub

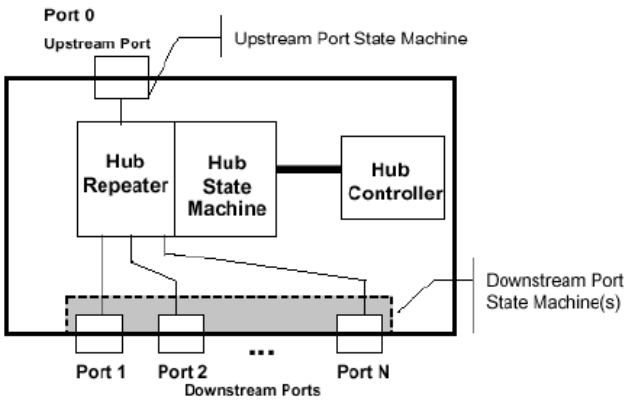
La principal funcion de un hub es extender el sistema proporcionando nuevos puertos de conexión.

Cada hub proporciona un puerto de conexión con el host (upstream) y varios puertos de conexión con dispositivos u otros hubs (downstream).



Al igual que cualquier otro dispositivo USB, un hub debe ser configurado, recibiendo su propia dirección.

Un hub USB no es sólo un distribuidor de datos, también dispone de cierta inteligencia.



- **Repetidor:** distribuye el tráfico entre el host (puerto upstream) y los dispostivisos USB (puertos downstream);
- **Controlador:** se encarga de regular el tráfico de datos de acuerdo con los dispositivos USB conectados;

El controlador del hub proporciona la interfaz de configuración al exterior.

A USB device may consist of several logical sub-devices that are referred to as device functions. A composite device may provide several functions, for example, a webcam (video device function) with a built-in microphone (audio device function). An alternative to this is a compound device, in which the host assigns each logical device a distinct address and all logical devices connect to a built-in hub that connects to the physical USB cable.

Puerto USB

Estructura – El cable

El cable USB contiene cuatro lineas, si el conector es estandar, o cinco líneas si el conector es mini o micro.

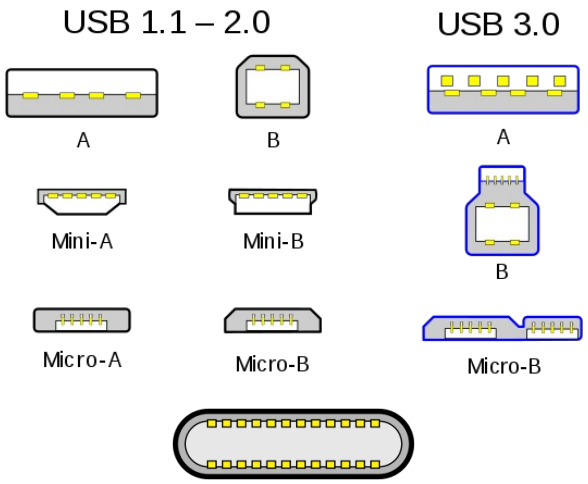
Pin	Nombre	Color de cable	Descripción
1	VCC	Rojo	+5 V
2	D−	Blanco	Data −
3	D+	Verde	Data +
4	GND	Negro	Tierra

Pin	Nombre	Color de cable	Descripción
1	VCC	Rojo	+5 V
2	D−	Blanco	Data −
3	D+	Verde	Data +
4	ID	Ninguno	Permite la distinción de Micro-A y Micro-B ▪ Tipo A: conectado a tierra ▪ Tipo B: no conectado
5	GND	Negro	Tierra y retorno o negativo

The USB 1.1 standard specifies that a standard cable can have a maximum length of 5 meters with devices operating at full speed (12 Mbit/s), and a maximum length of 3 meters with devices operating at low speed (1.5 Mbit/s).

USB 2.0 provides for a maximum cable length of 5 meters for devices running at high speed (480 Mbit/s).

The USB 3.0 standard does not directly specify a maximum cable length, requiring only that all cables meet an electrical specification: for copper cabling with AWG 26 wires the maximum practical length is 3 meters.



Puerto USB

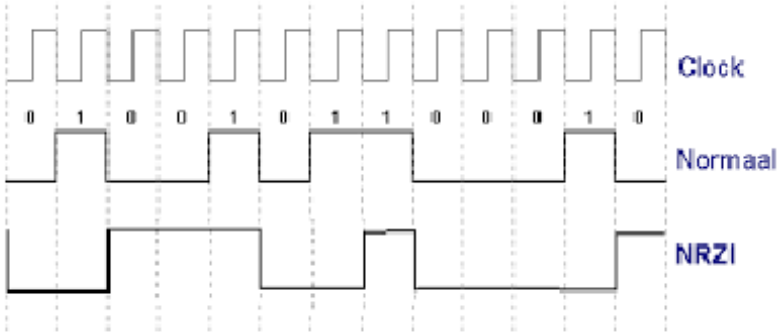
Estructura – El cable

Las señales del USB se transmiten en un cable de par trenzado con impedancia característica de 90 Ω, denominanados D+ y D–, que utilizan señal diferencial en half dúplex.

El USB 3.0 utiliza un segundo par de hilos, también con señal diferencial, para realizar una comunicación en full dúplex. La razón por la cual se realiza la comunicación en modo diferencial es simple, reduce el efecto del ruido electromagnético en enlaces largos.

La sincronización se obtiene a partir de los propios datos utilizando condificación de Manchester.

Además, para mejorar el desempeño y minimizar los efectos del cable en detección de los datos, los datos se codificacon un código de no retorno a cer invertido (NRZI - Non Return to Zero Inverted).



Los dispositivos USB se clasifican en cuatro tipos según su velocidad de transferencia de datos:

- **Baja velocidad** (1.0) - tasa de transferencia de hasta 1,5 Mbit/s, es utilizada por dispositivos de interfaz humana;
- **Velocidad completa** (1.1) - tasa de transferencia de hasta 12 Mbit/s, los dispositivos dividen el ancho de banda de la conexión entre ellos;
- **Alta velocidad** (2.0) - tasa de transferencia de hasta 480 Mbit/s, dispone de cuatro líneas, un par para datos, y otro par de alimentación;
- **Velocidad superalta** (3.0) - tiene una tasa de transferencia de hasta 4,8 Gbit/s;
- **Velocidad superalta** + (3.1) - duplica la velocidad de transferencia de datos máxima a 10 Gbit/s;

Puerto USB

Estructura – Transmisión de datos

Las transferencias de datos se realizan estableciendo canales de comunicación virtuales (pipes).

- Los canales son creados por el host;
- Cada canal ocupa parte del ancho de banda disponible.

Los canales terminan en un punto de final

- Cada dispositivo puede soportar varios puntos de final y, por tanto, varios canales distintos;
- La prioridad de los dispositivos viene dada por la conexión en cadena.

Podemos distinguir dos tipos distintos de canales

Los **canales de mensaje** tienen un formato concreto

- El esquema que siguen es: Petición – Dato – Estado;
- Cada petición debe ser completamente resuelta antes de pasar a la siguiente;
- Implican un movimiento de datos bidireccional.

Los **canales de flujo** (stream) no tienen un formato

- Los datos se envían de forma secuencial;
- El movimiento de datos es unidireccional.

Puerto USB

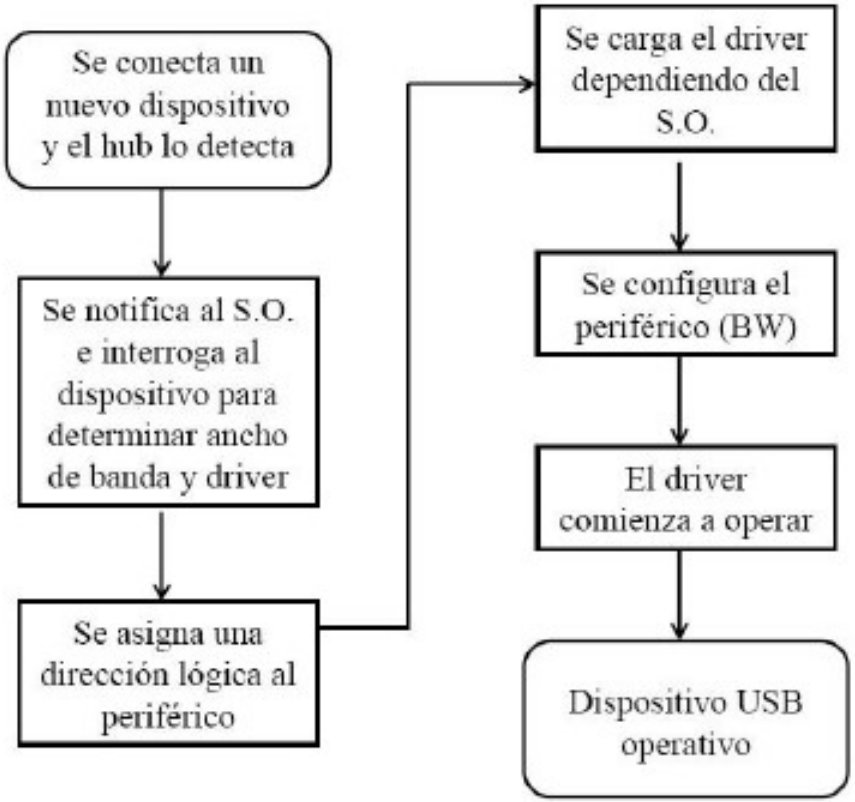
Estructura – Transmisión de datos

Cuando se conecta un dispositivo, el canal de control por defecto se establece con el punto de final 0.

Durante la inicialización, el host determina:

- El formato de datos que soporta el dispositivo conectado;
- El tipo de dispositivo y la dirección de la transferencia;
- Los requerimientos de frecuencia y latencia de bus;
- El ancho de banda necesario;
- El tamaño máximo de los paquetes.

Finalizado este proceso, el host asigna una dirección al dispositivo



Puerto USB

Estructura – Transmisión de datos

Las transmisiones se dividen en tramas de tiempo (frame). Durante las tramas se producen transacciones compuestas por paquetes

- Existen tres tipos de paquetes:
  - ✓ Inicialización (token)
  - ✓ Datos (data)
  - ✓ Protocolo (handshake)
- Cada paquete comienza con un campo de sincronización (SYNC) que maximiza el número de transiciones en la línea
- El tipo de paquete se diferencia con un identificador (PID)

Token:

PID								8 bit	3 bit	5 bit
0	1	x	x	1	0	x	x	ADDR	ENDP	CRC5

Data:

PID								0- n bit	16 bit
0	0	x	x	1	1	x	x	DATA	CRC16

Handshake:

PID							
1	0	x	x	0	1	x	x

PID	Meaning	PID (Hex)	Type
STALL	Error	0x1e	Handshake packet
SETUP	Initialization	0x2d	Token packet
PRE	Preamble	x	Special packet, only for low-speed devices
DATA1	Data packet 1	0x4b	Data packet
NACK	Not Acknowledge	0x5a	Handshake packet
IN	Receive	x	Token packet
SOF	Start of Frame	0x5a	Token packet
DATA0	Data packet 0	0xc3	Data packet
ACK	Acknowledge	0xd2	Handshake packet
OUT	Send	0x1e	Token packet

Puerto USB

Estructura – Transmisión de datos

En los canales de flujo se definen cuatro tipos posibles de transferencias

Control (canal mensaje) - utilizadas para configurar los dispositivos que se conectan

- Se garantiza la correcta emisión/recepción de datos;
- No se garantiza la latencia o el ancho de banda.

Masivas (canal flujo) - transferencias esporádicas de grandes cantidades de datos que pueden esperar (impresoras, escáners...)

- Se garantiza la correcta emisión/recepción de datos;
- No se garantiza la latencia o el ancho de banda.

Por interrupciones (canal flujo) - transferencias esporádicas de pocos datos que requieren atención inmediata (teclado, ratón...)

- Se garantiza la correcta emisión/recepción de datos;
- Se garantiza la latencia y el ancho de banda.

Alta velocidad (canal flujo) - grandes cantidades de información que se transmiten de forma continua (audio/video en tiempo real...)

- No se garantiza la correcta emisión/recepción de datos;
- Se garantiza una latencia y un ancho de banda constante.

La funcionalidad de un dispositivo USB se define mediante un código de clase enviado a un host USB. Esto permite que el host cargue módulos de software para el dispositivo y admita nuevos dispositivos de diferentes fabricantes. Las clases de dispositivo incluyen:

Class	Usage	Description	Examples, or exception
00h	Device	Unspecified[41]	Device class is unspecified, interface descriptors are used to determine needed drivers
01h	Interface	Audio	Speaker, microphone, sound card, MIDI
02h	Both	Communications and CDC Control	Modem, Ethernet adapter, Wi-Fi adapter, RS-232 serial adapter. Used together with class 0Ah *(CDC-Data, below)*
03h	Interface	Human interface device (HID)	Keyboard, mouse, joystick
05h	Interface	Physical Interface Device (PID)	Force feedback joystick
06h	Interface	Image (PTP/MTP)	Webcam, scanner
07h	Interface	Printer	Laser printer, inkjet printer, CNC machine
08h	Interface	Mass storage (MSC or UMS)	USB flash drive, memory card reader, digital audio player, digital camera, external drive
09h	Device	USB hub	Full bandwidth hub
0Ah	Interface	CDC-Data	Used together with class 02h *(Communications and CDC Control, above)*
0Bh	Interface	Smart Card	USB smart card reader
0Dh	Interface	Content security	Fingerprint reader
0Eh	Interface	Video	Webcam
0Fh	Interface	Personal healthcare device class (PHDC)	Pulse monitor (watch)
10h	Interface	Audio/Video (AV)	Webcam, TV
11h	Device	Billboard	Describes USB-C alternate modes supported by device
DCh	Both	Diagnostic Device	USB compliance testing device
E0h	Interface	Wireless Controller	Bluetooth adapter, Microsoft RNDIS
EFh	Both	Miscellaneous	ActiveSync device
FEh	Interface	Application-specific	IrDA Bridge, Test & Measurement Class (USBTMC),[42] USB DFU (Device Firmware Upgrade)[43]
FFh	Both	Vendor-specific	Indicates that a device needs vendor-specific drivers