

(6)



Unidad 6

Programación de los recursos hardware básicos del PC

SISTEMAS BASADOS EN MICROPROCESADORES

**Grado en Ingeniería Informática
Doble Grado en Ingeniería Informática y Matemáticas**

EPS - UAM

6. Programación de recursos hardware básicos del PC.

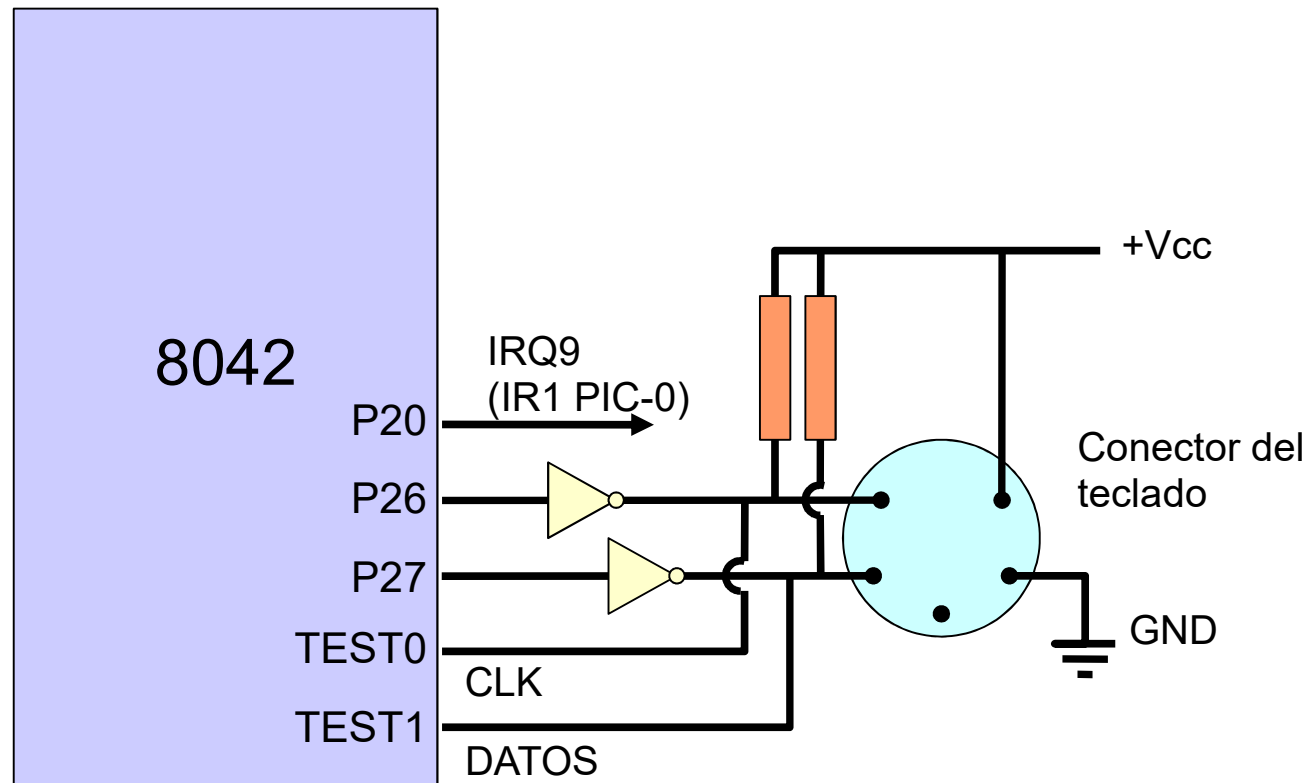
- 6.1. Teclado.
- 6.2. *Timer*.
- 6.3. Reloj de Tiempo Real (RTC).
- 6.4. Controladora de Vídeo y Pantalla.
- 6.5. Puerto Paralelo. Impresora.
- 6.6. Puerto Serie Asíncrono (UART 8250).

(6)

6.1. Teclado (I)

● Basado en un integrado de Intel (8042)

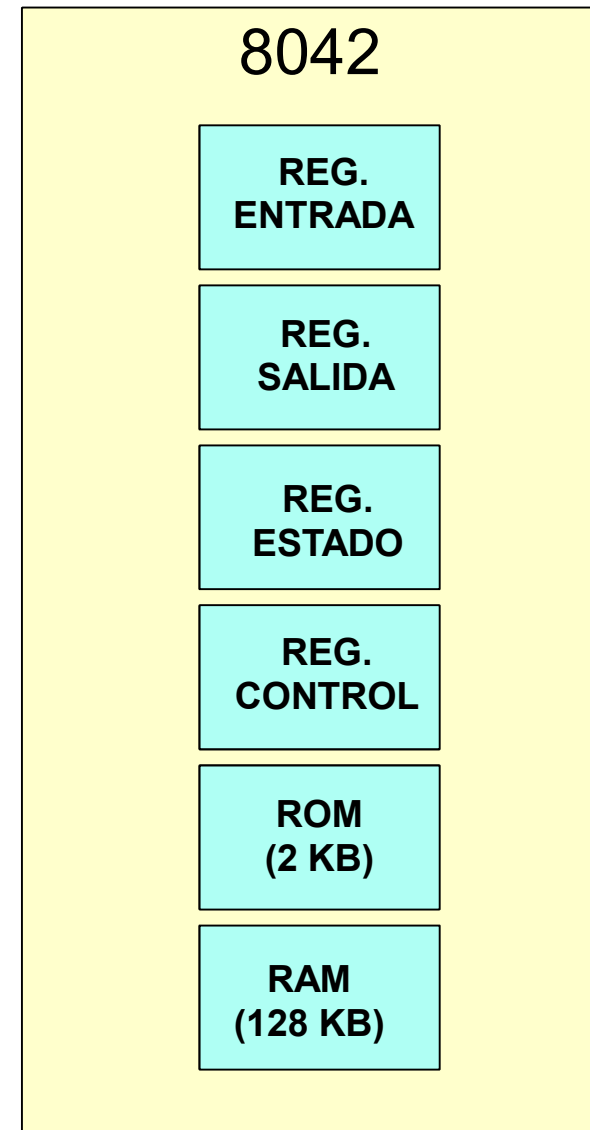
- El controlador de teclado 8042 tiene 2 puertos paralelos E/S (8 bits) programables y 2 entradas serie programables (TEST0, TEST1).
- Los pines P26 y P27 son para enviar CLK y DATOS al teclado.
- Las entradas serie TEST0 y TEST1 son para recibir el CLK y los DATOS enviados por el teclado.



(6)

6.1. Teclado (II)

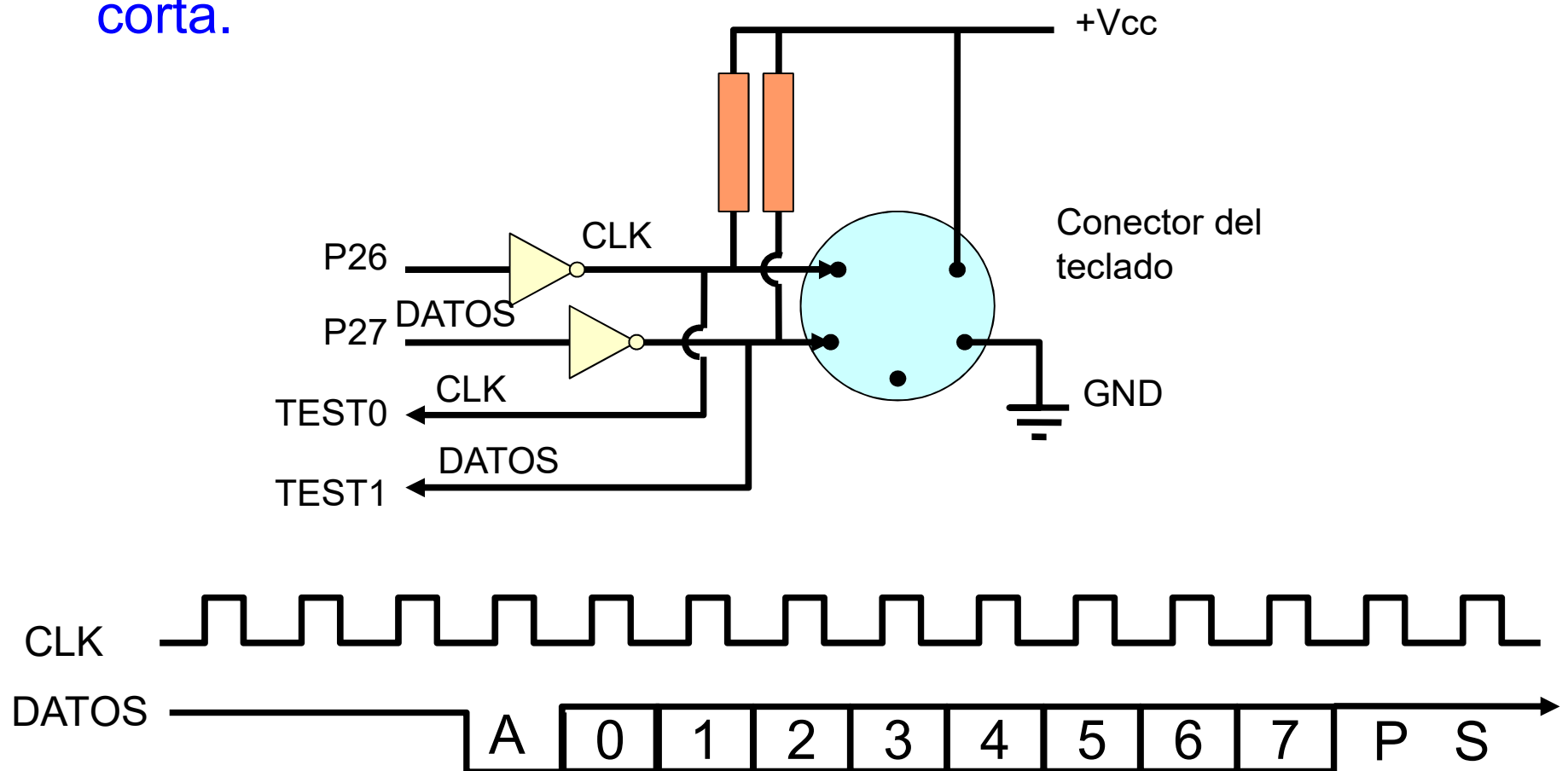
- El 8042 es un microcontrolador compatible con las familias MCS-48 (8048) y MCS-51 (8051) de Intel.
- Se comporta como un PPI (Interfaz de Periférico Programable) que permite implementar interfaces a medida para distintos periféricos.
- Dispone de 2 KB de memoria ROM (8042) o EPROM (8742), 128 KB de memoria RAM, 2 puertos prog. de E/S de 8 bits, 2 entradas serie prog., contador interno de 8 bits, reloj de 12 MHz, y otras señales de control y especiales
- Direcciones: 60h (REG. ENTRADA y SALIDA), 64h (REG. CONTROL y ESTADO)



(6)

6.1. Teclado (III)

- Al transmitir señal de reloj (CLK) aparte de los datos, no es necesario un controlador de línea serie (UART).
- No hay problemas de transmisión del reloj por ser línea muy corta.



(6)

6.1. Teclado (IV)

- Teclado del PC tiene 83 teclas divididas en 3 grupos:
 - Teclas de funciones.
 - Teclado alfanumérico.
 - Teclado numérico.
- Teclado se comunica con BIOS mediante INT 09h (PIC-0, IR1).
 - Cada vez que se pulsa/suelta tecla se genera **INT 09h**.
 - Rutina de servicio obtiene código de tecla (**SCAN CODE**) leyendo el puerto 60h del controlador de teclado.
 - Código de tecla al soltar es el mismo que al pulsar pero con bit de mayor peso a 1.
 - Cada vez que se pulsa tecla, la rutina de servicio almacena dos bytes en un buffer en memoria del BIOS:
 - Código ASCII correspondiente o 00h en caso de tecla especial (F1,...,F12, shift, flechas, etc.).
 - Identificador de tecla pulsada (SCAN CODE).

(6)

6.1. Teclado (V)

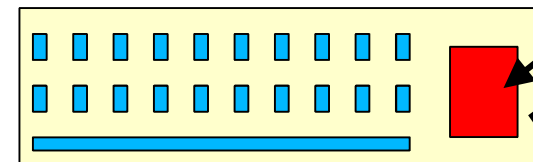
- Se pueden pulsar **combinaciones de teclas**. Normalmente teclas de tipo modificador (ALT, MAYÚSCULAS, CONTROL) acompañadas de otras teclas.
- Algunas combinaciones no son reconocidas por las rutinas BIOS y no generan ningún tipo de código.
- Teclas con significado especial:
 - **Alt-Ctrl-Del** (carga del sistema).
 - **Ctrl-Pausa** (llama a INT 1Bh del BIOS).
 - **Pausa** (detención de un programa hasta pulsar tecla).
 - **Impr Pant** (llama a INT 5h para imprimir pantalla por impresora)

(6)

6.1. Teclado (VI)

• Teclado

- Al pulsar o soltar tecla se envía código KSCAN (KEYBOARD SCAN CODE) que codifica posición física de la tecla en el teclado.



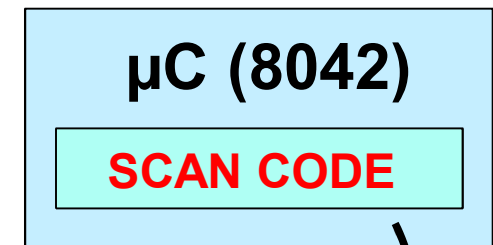
83 teclas

μC (8048, 8051)
Explora matricialmente el teclado y envía el KSCAN al PC

TEST0,1

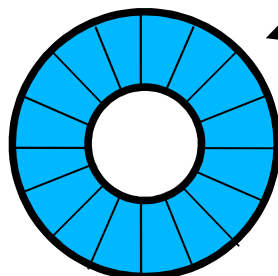
• Controlador teclado (8042)

- Programa de la ROM traduce KSCAN a SCAN CODE.
- Teclas F1 a F12, flechas, shift, etc. no generan código ASCII.

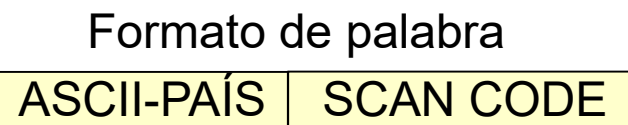


• Buffer del teclado (DOS) • BIOS (Int 09h)

PIC-0
(INT 09h)



Buffer de 16 palabras



Formato de palabra

BIOS
(INT 09h)

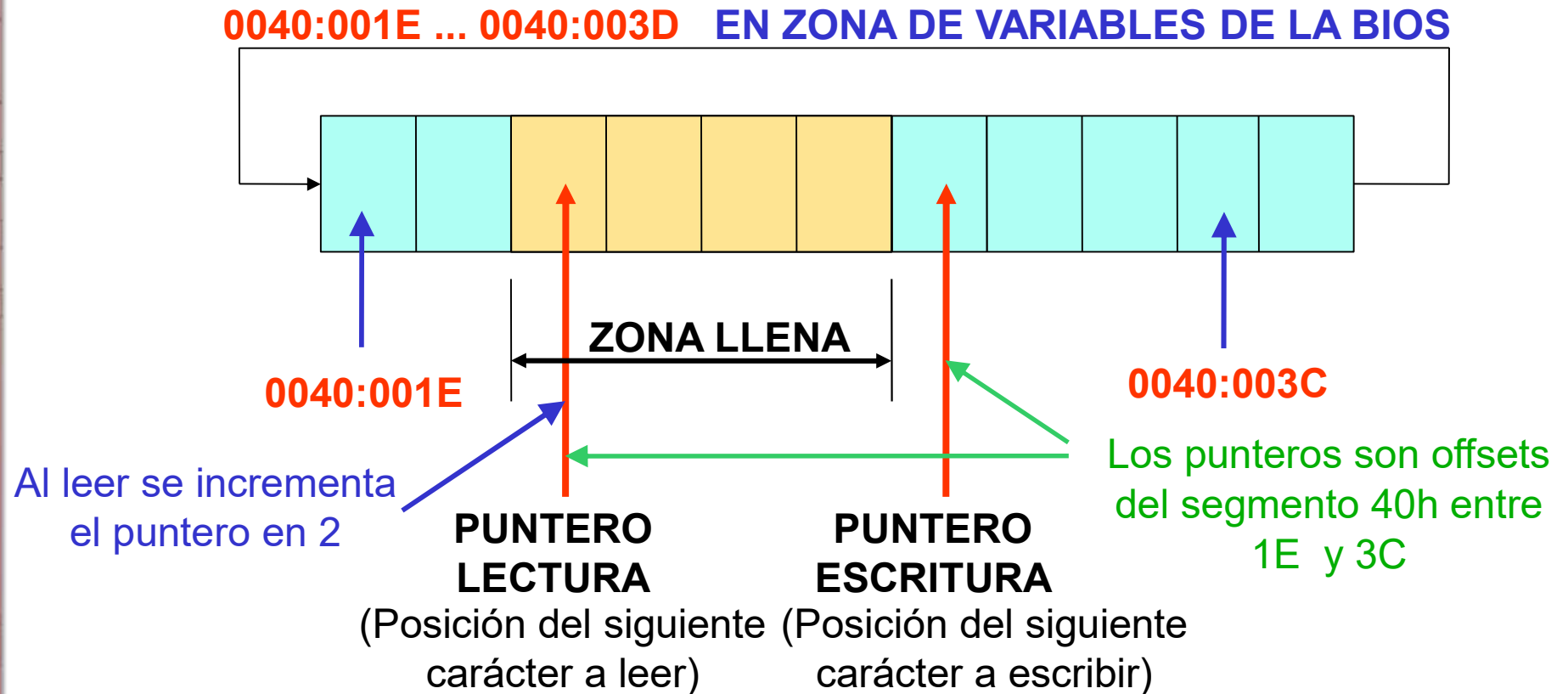


KEYB.COM

(6)

6.1. Teclado (VII)

Buffer cíclico



Se deja libre una posición (word). Se pueden guardar hasta 15 caracteres (words) en el buffer de 16 words

ASCII PAÍS (1 ^{er} byte)	SCAN CODE (2 ^o byte)
--------------------------------------	------------------------------------

(6)

6.1. Teclado (VIII)

INT 16h (BIOS)

AH	Función
00h y 10h	<p>Lee del buffer de teclado los códigos asociados a la tecla o combinación de teclas y avanza el puntero del buffer al carácter siguiente. Si el buffer está vacío espera a que se pulse una tecla.</p> <p>Salidas: AH = Identificador de la tecla (SCAN CODE) AL = código ASCII del carácter</p>
01h y 11h	<p>Devuelve el estado del buffer del teclado</p> <p>Salidas: ZF = 1 si buffer vacío ZF = 0 si buffer no vacío AH = Identificador de la tecla AL = código ASCII del carácter</p>
02h y 12h	<p>Estado de pulsación de distintas teclas</p> <p>Salidas: Devuelve en AL el byte de estado de pulsación de distintas teclas: 7: Ins, 6: Bloq. Mayúsculas, 5: Bloq. Numérico, 4: Bloq. Desplazamiento, 3: Alt, 2: Ctrl, 1: Mayúscula Izqda., 0: Mayúscula Dcha.</p>

(6)

6.1. Teclado (IX)

INT 16h (BIOS)

AH	Función
12h	Estado de pulsación de distintas teclas.
	Salidas: Devuelve en AH el byte de estado de pulsación de distintas teclas: 7: Impresión Pantalla, 6: Bloq. Mayúsculas, 5: Bloq. Numérico, 4: Bloq. Desplazamiento, 3: Alt Dcho., 2: Ctrl Dcho., 1: Alt Izqdo., 0: Ctrl Izqdo.
05h	Escribe el carácter indicado en el buffer del teclado.
	Entradas: CH = Identificador de la tecla (SCAN CODE) CL = Código ASCII del carácter. Salidas: AL = 1 si buffer lleno.

(6)

6.1. Teclado (X)

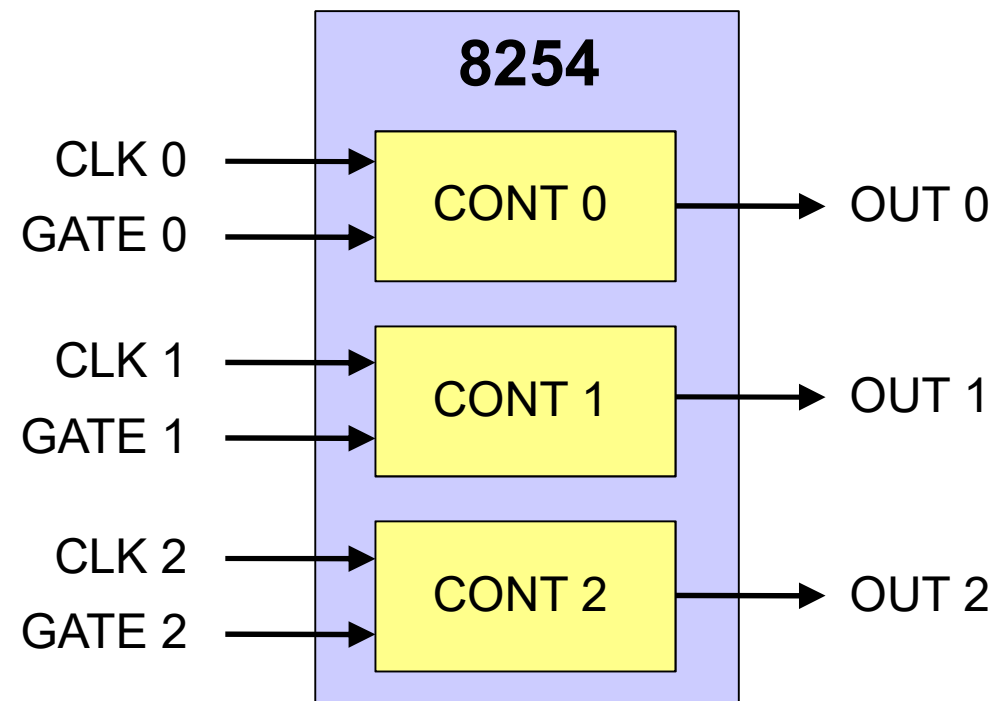
INT 21h (DOS)

AH	Función
01h	Espera para leer un carácter del teclado y lo escribe por pantalla
	Salidas: AL = carácter tecleado
06h	Lee un carácter del teclado sin salida por pantalla (DL = FFh)
	Salidas: AL = carácter tecleado si hay disponible, AL = 0 si no disponible
07h	Espera para leer un carácter del teclado sin escribirlo por pantalla
	Salidas: AL = carácter tecleado si hay disponible
0Ah	Lee caracteres del teclado sin escribirlos por pantalla
	Entrada: DS:DX = dirección del buffer de memoria El primer byte del buffer debe indicar el número máximo de caracteres a leer (incluido el retorno de carro) Salidas: El segundo byte del buffer indica el número de caracteres leídos sin incluir el retorno de carro. Los códigos ASCII de los caracteres tecleados se guardan a partir del tercer byte.
0Bh	Lee el estado del teclado
	Salidas: AL = FFh si carácter disponible, AL = 0 si no disponible.

(6)

6.2. *Timer (I)*

- Temporizador de intervalo programable de Intel (8254).
- Contiene 3 contadores independientes de 16 bits.
- Cada contador se decrementa en una unidad en cada flanco de bajada del reloj (entrada CLK) si está activada su señal de habilitación (entrada GATE).
- La salida de cada contador (señal OUT) se activa en función del modo en que el contador se haya programado previamente.
- 6 modos de funcionamiento.



(6)

6.2. *Timer* (II)

- El *timer* tiene 4 puertos de E/S.
- Puerto 43h (sólo de escritura)
 - Registro de control del *timer*.
 - Permite definir la configuración inicial de un contador y enviar un comando de memorización del valor de conteo actual.
- Puerto 40h (CONT 0)
 - En escritura modifica valor de conteo inicial de CONT 0.
 - El contador decrementa a partir de ese valor inicial.
 - Al llegar a cero, el contador puede volverse a cargar con el mismo valor inicial según el modo de funcionamiento.
 - En lectura retorna el valor memorizado de CONT 0.
- Puertos 41h (CONT 1) y 42h (CONT 2)
 - Igual comportamiento que 40h para los otros contadores.

(6)

6.2. *Timer (III)*

- **Palabra de control**

- Se escribe en el puerto 43h el siguiente valor:

SC1	SC0	RW1	RW0	M2	M1	M0	BCD
-----	-----	-----	-----	----	----	----	-----

(6)

6.2. *Timer (IV)*

- **Palabra de control**

- Se escribe en el puerto 43h el siguiente valor:

SC1	SC0	RW1	RW0	M2	M1	M0	BCD
-----	-----	-----	-----	----	----	----	-----

- Selección del contador

SC1	SC0	Acción
0	0	Selecciona CONT 0
0	1	Selecciona CONT 1
1	0	Selecciona CONT 2

- Configuraciones del resto de la palabra de control aplicadas sólo al contador seleccionado.

(6)

6.2. *Timer (V)*

- **Palabra de control**

- Se escribe en el puerto 43h el siguiente valor:

SC1	SC0	RW1	RW0	M2	M1	M0	BCD
-----	-----	-----	-----	----	----	----	-----

- Comando de memorización del valor actual del contador
+ Configuración de Lectura/Escritura en contador

RW1	RW0	Acción
0	0	Comando de memorización del valor actual (campos M y BCD sin efecto).
0	1	Lectura/Escritura en puerto del contador sólo afecta al byte bajo.
1	0	Lectura/Escritura en puerto del contador sólo afecta al byte alto.
1	1	Primera Lectura/Escritura en puerto del contador afecta a byte bajo y segunda al byte alto.

(6)

6.2. *Timer (VI)*

- **Palabra de control**

- Se escribe en el puerto 43h el siguiente valor:

SC1	SC0	RW1	RW0	M2	M1	M0	BCD
-----	-----	-----	-----	----	----	----	-----

- Configuración de modo de funcionamiento

M2	M1	M0	Acción
0	0	0	Modo 0: Interrupción al final del conteo.
0	0	1	Modo 1: Monoestable programable.
X	1	0	Modo 2: Generador de frecuencia.
X	1	1	Modo 3: Generador de onda cuadrada.
1	0	0	Modo 4: Pulso <i>Strobe</i> iniciado por software.
1	0	1	Modo 5: Pulso <i>Strobe</i> iniciado por hardware.

(6)

6.2. *Timer* (VII)

- **Palabra de control**

- Se escribe en el puerto 43h el siguiente valor:

SC1	SC0	RW1	RW0	M2	M1	M0	BCD
-----	-----	-----	-----	----	----	----	-----

- Configuración de codificación del contador

BCD	Acción
0	Contador binario de 16 bits.
1	Contador BCD de cuatro dígitos.

- Binary-Coded Decimal (**BCD**): Cada cuatro bits codifican un dígito entre 0 (**0000b**) y 9 (**1001b**).

(6)

6.2. *Timer (VIII)*

- **Configuración inicial:** Se envía palabra de control al puerto 43h indicando contador y su configuración inicial (**RW \neq 00b**).
- **Valor de conteo inicial de un contador:**
 - Se envía al puerto del contador (**40h, 41h o 42h**).
 - Se envía byte bajo, byte alto o byte bajo seguido de byte alto según campo RW de la palabra de control.
- **Ejemplo:** Configurar contador 2 (**SC = 10b**) en modo 3 (**M = 011b**), con codificación binaria (**BCD = 0**), e inicializarlo a 1234h (**RW = 11b**).
 - **mov al, 10110110b** ; Palabra de control: **SC | RW | M | BCD**
out 43h, al ; Envía palabra de control
mov al, 34h
out 42h, al ; Envía byte bajo de valor de conteo inicial.
mov al, 12h
out 42h, al ; Envía byte alto de valor de conteo inicial.

(6)

6.2. *Timer (IX)*

- **Lectura del valor actual de un contador:**

- (1) Enviar comando de memorización de valor de conteo actual (RW=00b) al puerto 43h, especificando el contador.
- (2) Leer valor memorizado de conteo actual desde el puerto del contador especificado (40h, 41h o 42h):

Se lee byte bajo, byte alto o byte bajo seguido de byte alto según campo RW de configuración inicial.

- **Ejemplo:** Leer valor de conteo actual de contador 2 (SC=10b), que fue inicializado con RW=11b.

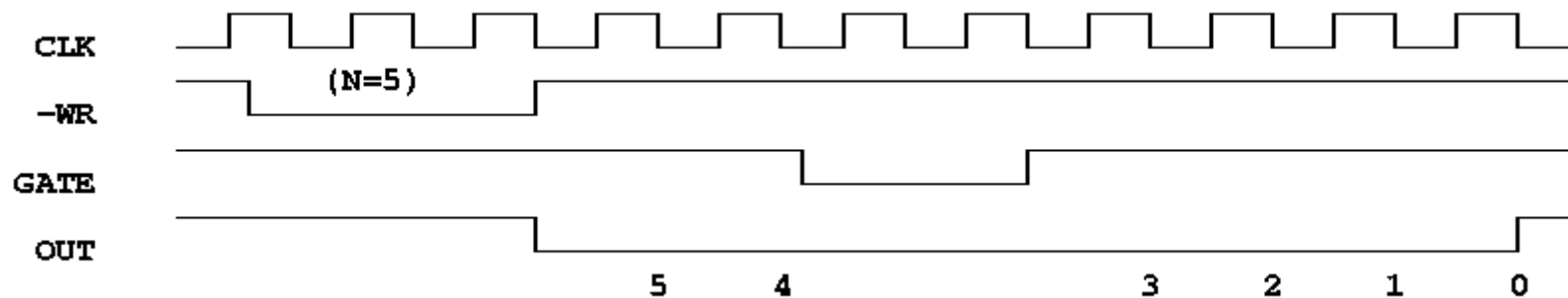
- | | |
|--------------------------|--|
| mov al, 10000000b | ; SC RW M BCD |
| out 43h, al | ; Memoriza valor de conteo actual |
| in al, 42h | ; Lee byte bajo de valor de conteo actual. |
| mov bl, al | |
| in al, 42h | ; Lee byte alto de valor de conteo actual. |
| mov bh, al | ; BX con valor memorizado de conteo actual |

El valor memorizado puede no coincidir con el valor de conteo actual: el contador sigue decrementando tras memorizar.

(6)

6.2. *Timer (X)*

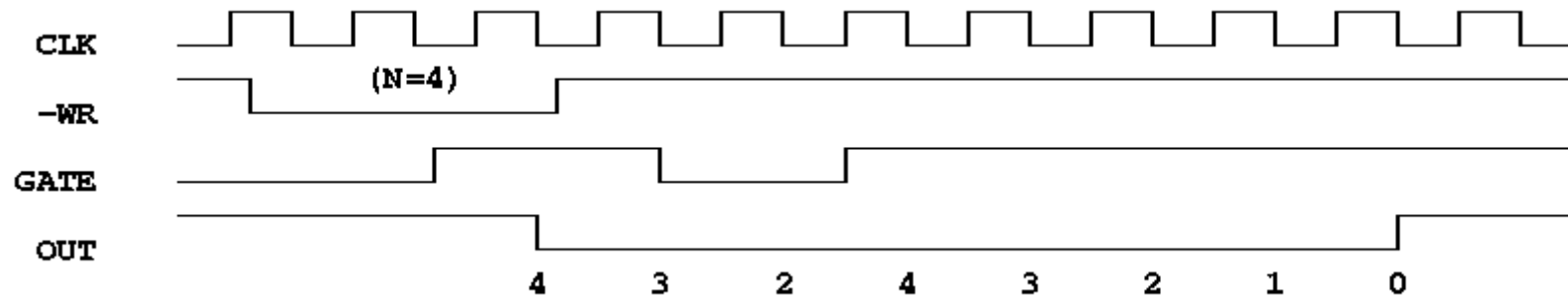
- **Modo 0: Interrupción al final de conteo.**
 - OUT pasa de 0 a 1 cuando contador llega a 0.
 - Contador decreuenta mientras GATE=1.
 - Contador reinicia conteo cuando se le envía nuevo valor de conteo inicial o nueva palabra de control.



(6)

6.2. *Timer (XI)*

- **Modo 1: Monoestable programable.**
 - OUT pasa de 0 a 1 cuando contador llega a 0.
 - Contador reinicia conteo con valor inicial cuando GATE pasa de 0 a 1.

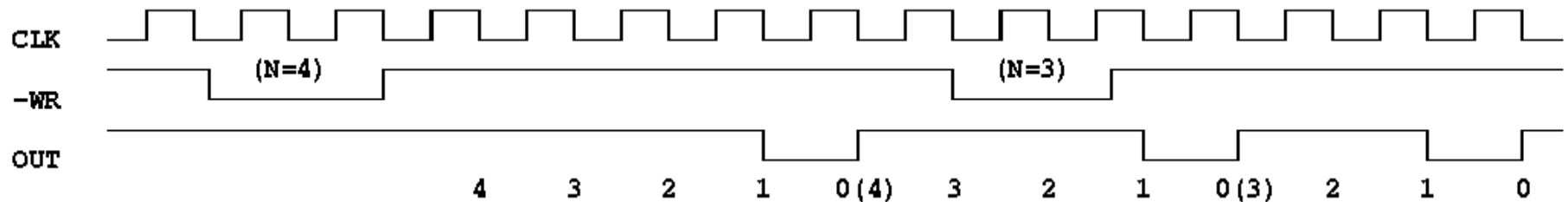


(6)

6.2. *Timer (XII)*

● Modo 2: Generador de frecuencia.

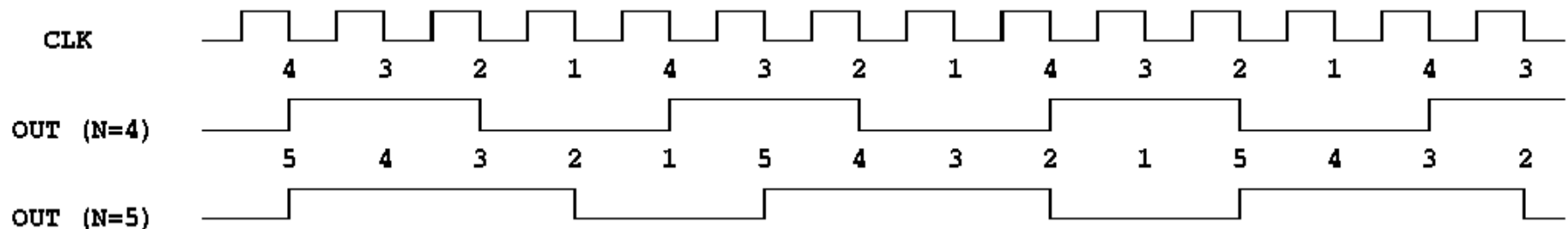
- OUT pasa de 1 a 0 cuando contador llega a 1 y de 0 a 1 cuando llega a 0.
- Contador decrementa mientras GATE=1.
- Contador reinicia conteo con valor inicial cuando GATE pasa de 0 a 1 y cuando contador llega a 0.



(6)

6.2. *Timer (XIII)*

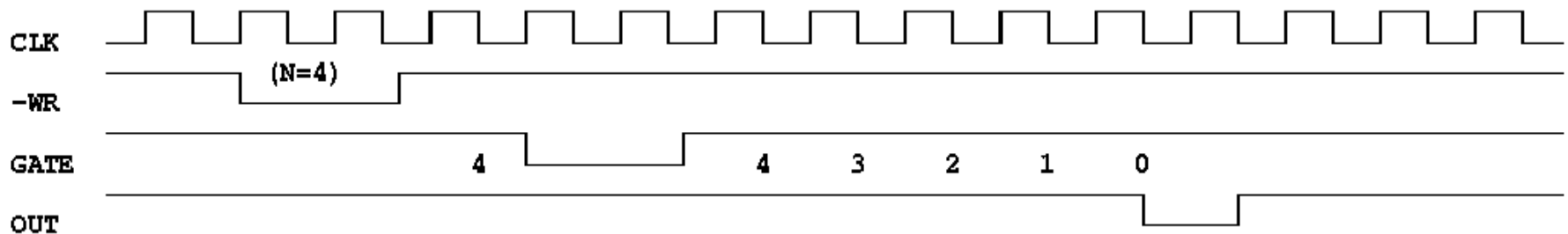
- **Modo 3: Generador de onda cuadrada.**
 - Modo habitual.
 - OUT está a 1 en la primera mitad del conteo y a 0 en la segunda mitad.
 - Contador decreuenta mientras GATE=1.
 - Contador reinicia conteo con valor inicial cuando GATE pasa de 0 a 1 y cuando contador llega a 0.



(6)

6.2. *Timer (XIV)*

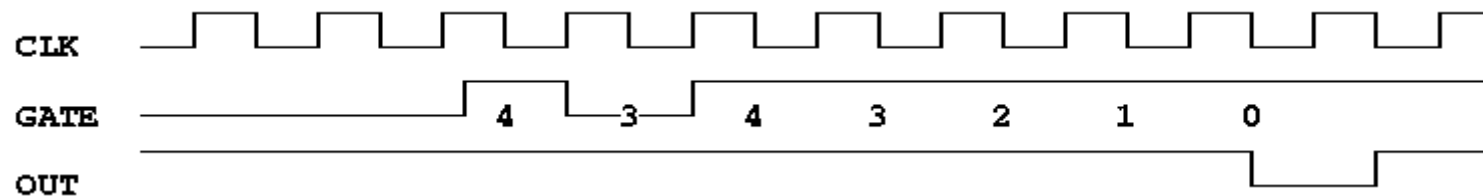
- **Modo 4: Pulso *Strobe* iniciado por software.**
 - OUT está a 0 durante un ciclo de reloj cuando contador llega a 0.
 - Contador decrementa mientras GATE=1.
 - Contador reinicia conteo con valor inicial cuando GATE pasa de 0 a 1.
 - Contador reinicia conteo cuando se le envía nuevo valor de conteo inicial o nueva palabra de control (sincronización por software)



(6)

6.2. *Timer (XV)*

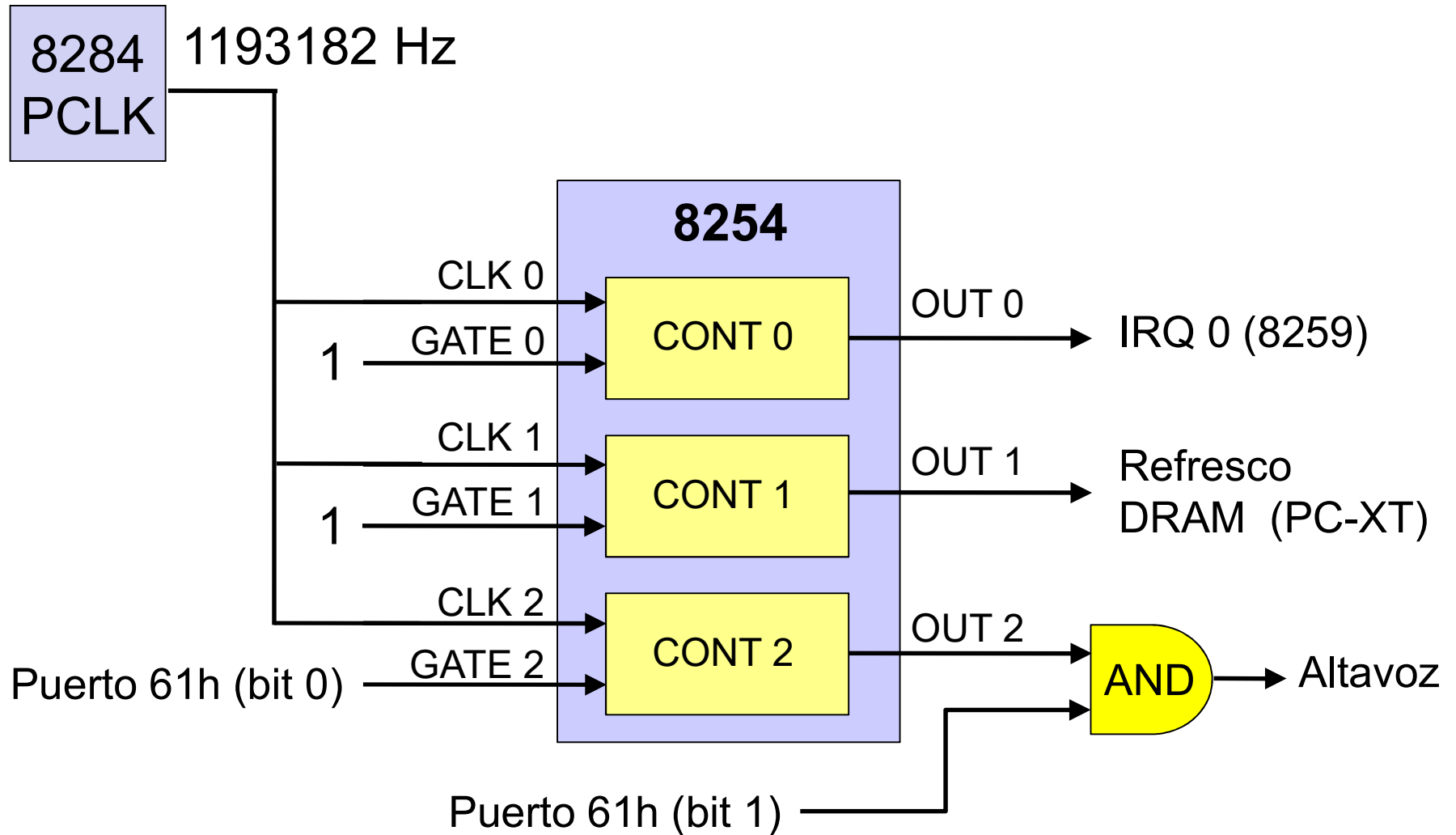
- **Modo 5: Pulso *Strobe* iniciado por hardware.**
 - OUT está a 0 durante un ciclo de reloj cuando contador llega a 0.
 - Contador reinicia conteo con valor inicial cuando GATE pasa de 0 a 1.
 - Contador reinicia conteo tras llegar a 0 cuando se le envía nuevo valor de conteo inicial o nueva palabra de control.



(6)

6.2. *Timer (XVI)*

- Conexión del *timer* en el PC

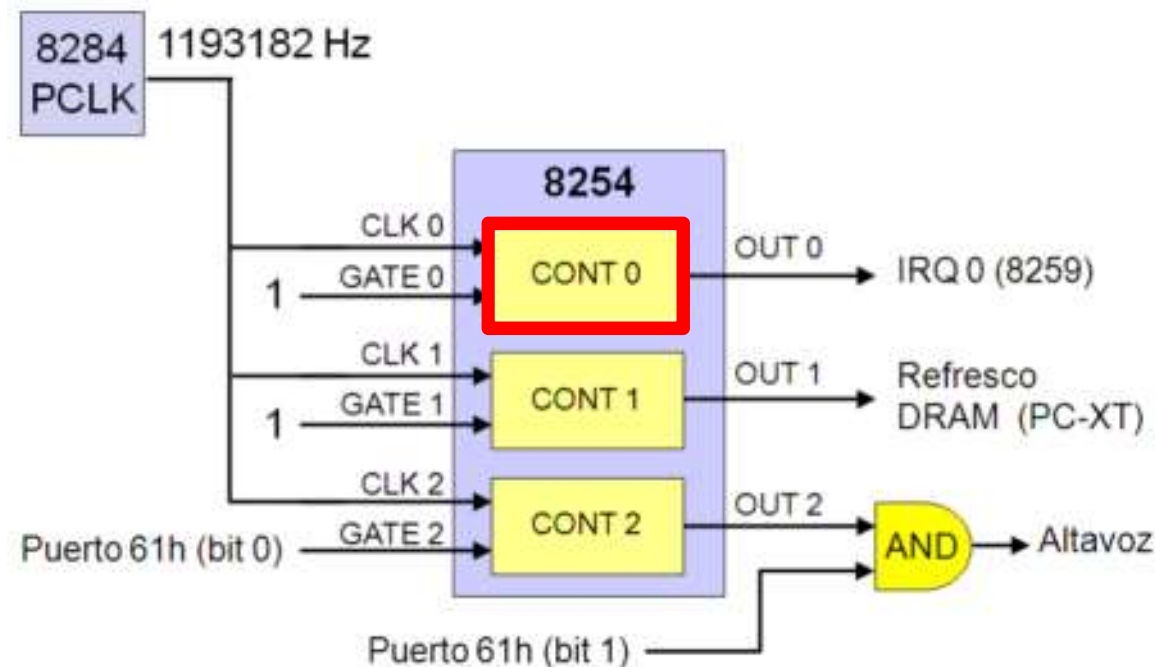


(6)

6.2. *Timer (XVII)*

• Conexión del *timer* en el PC

- CONT 0 genera interrupción enmascarable INT 8.
- Configurado en modo 3 (onda cuadrada)
- Valor inicial de conteo = 0 ($\equiv 65536$)
- Frecuencia de onda = $1193182 / 65536 = 18,2$ (Hz)

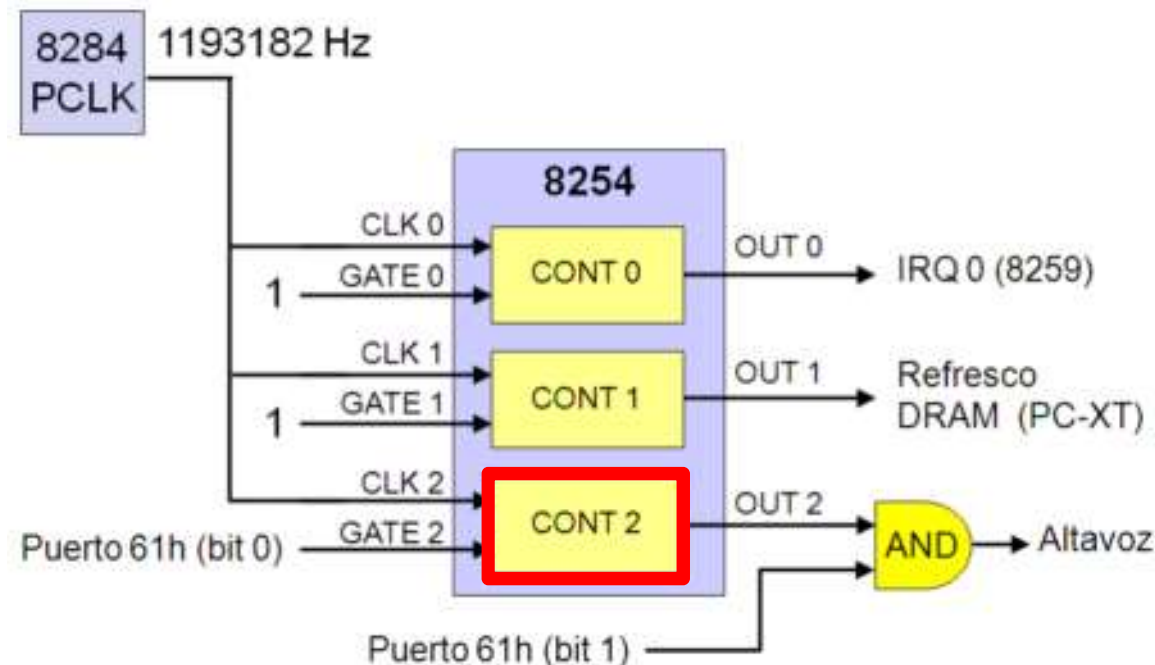


(6)

6.2. *Timer (XVIII)*

• Conexión del *timer* en el PC

- CONT 2 libre para usuario.
- Salida OUT conectada al altavoz del PC si bit 1 del puerto 61h vale 1.
- Entrada GATE conectada a bit 0 del puerto 61h.
- Configuración habitual en modo 3 (onda cuadrada)
- Valor inicial = $1193182 / \text{Frecuencia deseada (Hz)}$



(6)

6.3. Reloj de Tiempo Real (RTC) (I)

- IBM incluyó en el PC-AT el chip RTC Real Time Clock (MC146818 de Motorola) alimentado por batería.
- Tecnología CMOS de bajo consumo (idónea para baterías).
- Tiene memoria RAM estática de 64 bytes para la configuración del sistema:
 - 6 bytes de hora (Segundos, Minutos, Horas) del reloj y de la alarma.
 - 4 bytes de fecha (Día de la semana, Día del mes, Mes, Año).
 - 4 bytes para registros de control (A, B, C, D)
 - 50 bytes libres para el usuario.
- Puede generar interrupciones periódicas, alarmas y señales hardware.

(6)

6.3. Reloj de Tiempo Real (RTC) (II)

• Lectura del RTC:

- Se escribe (OUT) en el puerto 70h la dirección de la posición que se desea leer.
- Se realiza una lectura (IN) del puerto 71h.

• Escritura del RTC:

- Se escribe (OUT) en el puerto 70h la dirección de la posición en la que se desea escribir.
- Se escribe (OUT) en el puerto 71h el valor que se desea escribir.

(6)

6.3. Reloj de Tiempo Real (RTC) (III)

- **Registro A** (enviar un valor 0Ah al puerto 70h)
 - Leer o escribir sobre el puerto 71h el valor dado por:

UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0
-----	-----	-----	-----	-----	-----	-----	-----

(6)

6.3. Reloj de Tiempo Real (RTC) (IV)

- **Registro A** (enviar un valor 0Ah al puerto 70h)
 - Leer o escribir sobre el puerto 71h el valor dado por:

UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0
-----	-----	-----	-----	-----	-----	-----	-----

- **Update_In_Progress** (sólo lectura): Cuando está a 0 indica que se puede leer/escribir en los puertos del reloj sin que interfiera con actualizaciones internas.

(6)

6.3. Reloj de Tiempo Real (RTC) (V)

- **Registro A** (enviar un valor 0Ah al puerto 70h)
 - Leer o escribir sobre el puerto 71h el valor dado por:

UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0
-----	-----	-----	-----	-----	-----	-----	-----

DV2...DV0	Frecuencia del oscilador
000	4.193404 MHz
001	1.048576 Mhz
010	32.768 kHz

(6)

6.3. Reloj de Tiempo Real (RTC) (VI)

- **Registro A** (enviar un valor 0Ah al puerto 70h)
 - Leer o escribir sobre el puerto 71h el valor dado por:

UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0
-----	-----	-----	-----	-----	-----	-----	-----

- Cálculo de RS a partir de la frecuencia deseada de interrupciones periódicas del reloj:

$$RS = 1 + \log_2 \frac{32768 \text{ (Hz)}}{\text{Frecuencia (Hz)}}$$

- Ejemplo: si se desea una frecuencia de 512 Hz
 - **RS = 7 = 0111b**

(6)

6.3. Reloj de Tiempo Real (RTC) (VII)

- **Registro B** (enviar un valor 0Bh al puerto 70h)
 - Leer o escribir sobre el puerto 71h el valor dado por:

SET	PIE	AIE	UIE	SQWE	DM	12/24	DSE
-----	-----	-----	-----	------	----	-------	-----

- **DSE:** (Daylight Saving Enable) Se activa con 1.
- **12/24:** Se activa a 24 horas con un 1.
- **DM:** (Date Mode) 0 es BCD, 1 es Decimal.
- **SQWE:** (Square Wave Enable) Habilita onda de salida del reloj.
- **UIE:** Se habilita una interrupción tras el cambio de hora o de fecha (actualización del reloj).
- **AIE:** Se habilita una interrupción en la hora de la alarma.
- **PIE:** Se habilitan las interrupciones periódicas.
- **SET:** Cuando está a 0 el reloj funciona. Cuando se pone a 1 se para el reloj y se pueden cambiar los 14 bytes de configuración del reloj.

(6)

6.3. Reloj de Tiempo Real (RTC) (VIII)

- **Registro C** (enviar un valor 0Ch al puerto 70h)
 - Leer (**sólo lectura**) sobre el puerto 71h el valor dado por:

IRQF	PF	AF	UF	----	----	----	----
------	----	----	----	------	------	------	------

- Cuando están a 1 determinan el tipo de suceso que ha provocado la interrupción.
 - **IRQF: Petición de interrupción**
 - **PF: Interrupción periódica**
 - **AF: Alarma**
 - **UF: Actualización de la hora/fecha**

(6)

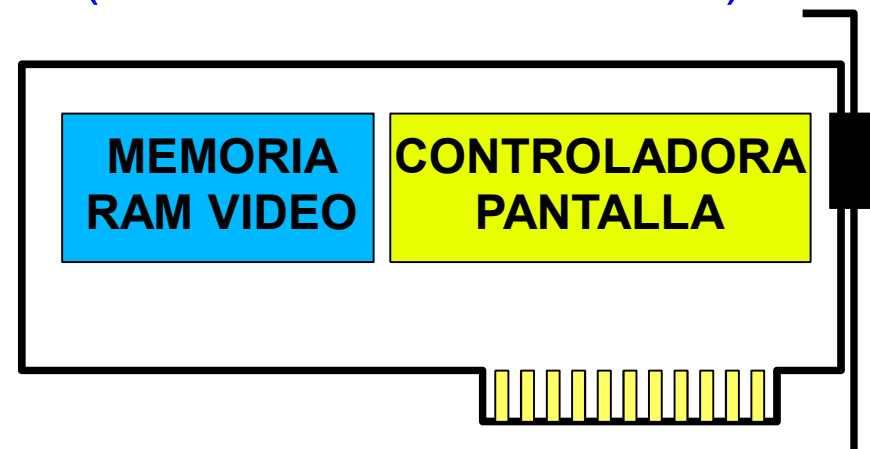
6.3. Reloj de Tiempo Real (RTC) (IX)

- El RTC no genera peticiones de interrupción por defecto. Es necesario programarlo para ello.
- Se usa el registro B para determinar qué evento va a producir la interrupción (alarma, interrupción periódica o cambio de hora).
- La rutina de atención a la interrupción ha de comprobar si el evento que ha generado la interrupción es el deseado leyendo el registro C.
- Al final de la rutina se debe mandar el EOI correspondiente a los PICs (8259) esclavo y maestro.
- Genera la interrupción 70h y está conectado al IRQ 0 del esclavo.

(6)

6.4. Controladora de Vídeo y Pantalla (I)

- Componentes de la controladora de vídeo:
 - Memoria RAM de Vídeo:
 - Almacena la información mostrada por pantalla.
 - Mapeada en espacio de direcciones del PC.
 - Organizada en páginas.
 - Controladora de pantalla:
 - Lee periódicamente el contenido de la página activa desde la RAM de vídeo, generando la señal de salida a la pantalla.
 - La página que está activa se puede cambiar en cualquier momento por programa (técnica de **doble buffer**).



(6)

6.4. Controladora de Vídeo y Pantalla (II)

- **Píxel**

- Unidad básica de información gráfica de la pantalla, que está organizada como matriz 2D de píxels.

- **Resolución de pantalla**

- N° píxels verticales (**filas**) x N° píxels horizontales (**columnas**)

- **En pantallas en color, cada pixel consta de 3 colores:**

- Rojo, Verde y Azul (RGB).

- **Tamaño del punto (*Dot Pitch*)**

- Tamaño del pixel en mm. Factor de calidad de la pantalla. Las mejores suelen tener actualmente un punto de 0.18 mm y las peores de 0.25 mm.

(6)

6.4. Controladora de Vídeo y Pantalla (III)

- Evolución de las primeras controladoras de vídeo
 - Modo MDA (Monochrome Display Adapter)
 - Primer modo texto monocromo.
 - Modo CGA (Color Graphics Adapter)
 - Primer modo gráfico color. Poca resolución y colores.
 - Modo EGA (Enhanced Graphics Adapter)
 - Aparece en 1.985. Muy costosa. No tiene aceptación.
 - Modo HERCULES
 - Mejora la calidad de los gráficos CGA pero es monocromo (720 x 348 pixels). Soporta el modo MDA (texto).
 - Modo VGA (Video Graphics Adapter)
 - Aparece en 1.987. Sustituye al EGA al ser mucho más asequible.
 - Modo SVGA (Super VGA)
 - Mejora el VGA.

(6)

6.4. Controladora de Vídeo y Pantalla (IV)

MODO	TIPO	RESOL.	COLOR	# COLORES	CGA	EGA	VGA	SVGA
0	T	40x25	M	16	S	S	S	S
1	T	40x25	C	16	S	S	S	S
2	T	80x25	M	16	S	S	S	S
3	T	80x25	C	16	S	S	S	S
4	G	320x200	C	4	S	S	S	S
5	G	320x200	M	4	S	S	S	S
6	G	640x200	M	2	S	S	S	S
7	T	80x25	M	2		S	S	S
13	G	320x200	C	16		S	S	S
14	G	640x200	C	16		S	S	S
15	G	640x350	M	2		S	S	S
16	G	640x350	C	16		S	S	S
17	G	640x480	M	2			S	S
18	G	640x480	C	16			S	S
19	G	320x200	C	256			S	S

(6)

6.4. Controladora de Vídeo y Pantalla (V)

- En el mapa de memoria del PC hay 128 KB reservados para la controladora de vídeo.
- Lecturas/Escrituras en esas direcciones de memoria del PC son mapeadas a memoria de vídeo.

A000h:0000h

...

B000h:FFFFh

Modo de vídeo	Dirección inicial	Dirección final	Tamaño
Modo 7 (EGA, VGA)	B000:0000	B000:7FFF	32 KB
Modos 0 a 6 (CGA)	B800:0000	B800:7FFF	32 KB
Modos 13 ... (EGA, VGA, SVGA)	A000:0000	B000:FFFF	128 KB

(6)

6.4. Controladora de Vídeo y Pantalla (VI)

• Modos de texto (alfanuméricos)

- Necesario definir atributo y código ASCII del carácter.
- Necesario calcular la dirección de memoria donde escribir un carácter que aparezca en la posición (X,Y) de la pantalla.
- El cursor indica la posición activa de la pantalla. Es físico en modo texto.
- Se puede modificar y consultar la posición del cursor mediante el BIOS.
- En modo gráfico no existe cursor físico \Rightarrow los programas deben crearlo.

(6)

6.4. Controladora de Vídeo y Pantalla (VII)

- Unidad de información \Rightarrow carácter (2 bytes consecutivos)
- Cada carácter está definido por una matriz de píxeles de tamaño fijo (todos ocupan el mismo espacio).

7	6	5	4	3	2	1	0
Código ASCII del carácter							
Dirección par							

- **P** : Parpadeo
- **R** : Rojo
- **G** : Verde
- **B** : Azul
- **I** : Intensidad

7	6	5	4	3	2	1	0
Atributo del carácter							
Dirección impar							
P	R	G	B	I	R	G	B
Fondo				Texto			

(6)

6.4. Controladora de Vídeo y Pantalla (VIII)

- Dirección de carácter en posición (X,Y):

$$@ = \text{Buff} + \text{Pag} * \text{TamPágina} + 2 * Y * \text{Rx} + 2 * X + B$$

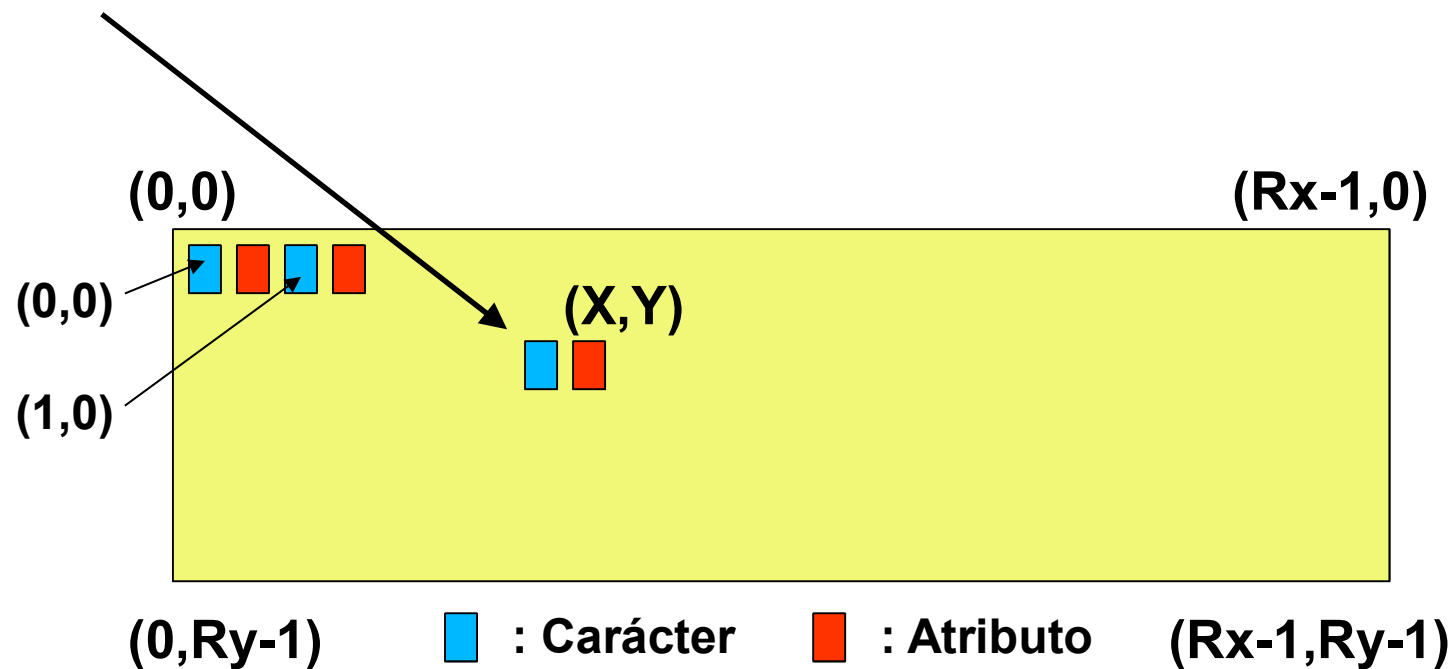
- **Buff** : Dirección inicial en el PC donde está mapeada la memoria RAM de vídeo en modo texto.
- **Pag** : N° de página
- **TamPágina**: Número de bytes de una página de pantalla
- **B** : 0 si byte es código ASCII, 1 si byte es atributo
- **X** : Columna
- **Y** : Fila
- **Rx** : resolución en columnas (n° columnas / fila)

(6)

6.4. Controladora de Vídeo y Pantalla (IX)

- Dirección de carácter en posición (X,Y):

$$@ = \text{Buff} + \text{Pag} * \text{TamPágina} + 2 * Y * \text{Rx} + 2 * X + B$$



(6)

6.4. Controladora de Vídeo y Pantalla (X)

Modo CGA de texto

- **Matriz carácter** : 8 x 8 píxels.
- **Carácter** : 7 x 7 pixels.
- **Resolución** : 80 x 25 caracteres.
- **Colores** : 16 para carácter y 16 para fondo.
- **Tamaño página** : 2000 car x 2 bytes/car.
- **Tamaño buffer vídeo** : 16 KB (4 páginas).
- **Segmento inicial del buffer de vídeo**: B800h
- **Pantalla**:
 - 80x25x2=4000 bytes (0FA0h) posiciones de memoria de vídeo.
 - En el buffer de vídeo caben cuatro pantallas distintas.

I	R	G	B	Color
0	0	0	0	Negro
0	0	0	1	Azul
0	0	1	0	Verde
0	0	1	1	Cyan
0	1	0	0	Rojo
0	1	0	1	Magenta
0	1	1	0	Marrón
0	1	1	1	Blanco
1	0	0	0	Gris
1	0	0	1	Azul claro
1	0	1	0	Verde claro
1	0	1	1	Cyan claro
1	1	0	0	Rojo claro
1	1	0	1	Magenta claro
1	1	1	0	Amarillo
1	1	1	1	Blanco intenso

(6)

6.4. Controladora de Vídeo y Pantalla (XI)

Modo CGA de texto

- Sólo puede haber una página activa en cada momento y es la que se visualiza.

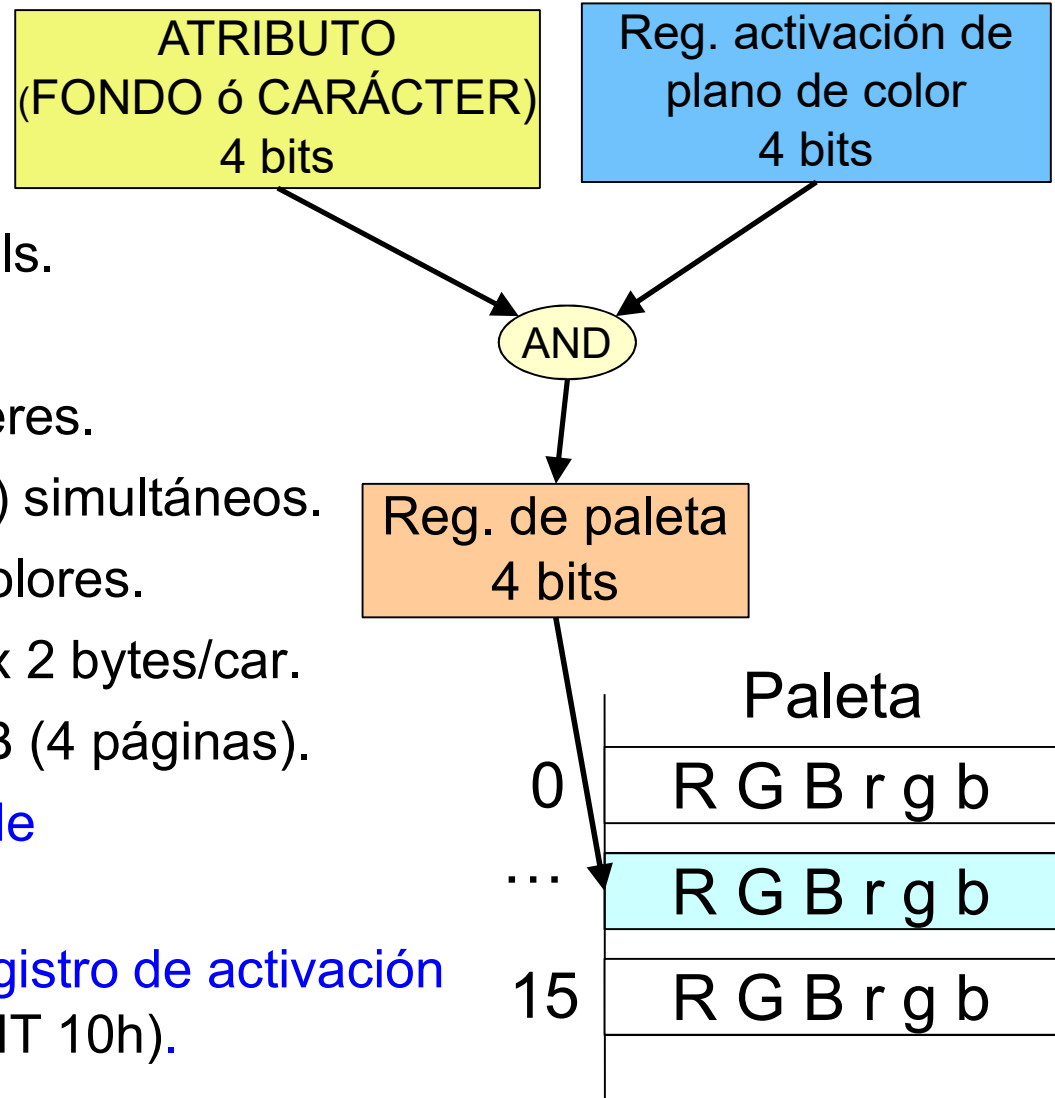
B800:0000	4000 (FA0h) bytes	Página 0 Activa	4 KB
B800:0F9F	96 (60h) bytes	No usada	
B800:1000	4000 (FA0h) bytes	Página 1	4 kB
B800:1F9F	96 (60h) bytes	No usada	
B800:2000	4000 (FA0h) bytes	Página 2	4 kB
B800:2F9F	96 (60h) bytes	No usada	
B800:3000	4000 (FA0h) bytes	Página 3	4 KB
B800:3F9F	96 (60h) bytes	No usada	
B800:3FFF			

(6)

6.4. Controladora de Vídeo y Pantalla (XII)

Modo EGA de texto

- **Matriz carácter** : 8 x 14 pixels.
- **Carácter** : 7 x 7 pixels.
- **Resolución** : 80 x 25 caracteres.
- **Colores** : 64 (2^6) con 16 (2^4) simultáneos.
- **Paletas** : 4 paletas de 16 colores.
- **Tamaño página** : 2000 car x 2 bytes/car.
- **Tamaño buffer vídeo** : 16 KB (4 páginas).
- **Segmento inicial del buffer de vídeo** : B800h.
- **Contenido de paleta y de registro de activación** definidos mediante BIOS (INT 10h).



(6)

6.4. Controladora de Vídeo y Pantalla (XIII)

Modo EGA de texto

- Paleta por defecto compatible con modo CGA.
- Registro de control de la controladora puede programarse para activar parpadeo: 1 en bit más significativo del byte de atributo produce parpadeo.
El índice de color de fondo se reduce a 3 bits de menor peso (8 colores simultáneos).

Índice	R	G	B	r	g	b	Color
0	0	0	0	0	0	0	Negro
1	0	0	0	0	0	1	Azul
2	0	0	0	0	1	0	Verde
3	0	0	0	0	1	1	Cyan
4	0	0	0	1	0	0	Rojo
5	0	0	0	1	0	1	Magenta
6	0	0	0	1	1	0	Marrón
7	0	0	0	1	1	1	Blanco
8	1	1	1	0	0	0	Gris
9	0	0	1	0	0	1	Azul claro
10	0	1	0	0	1	0	Verde claro
11	0	1	1	0	1	1	Cyan claro
12	1	0	0	1	0	0	Rojo claro
13	1	0	1	1	0	1	Magenta claro
14	1	1	0	1	1	0	Amarillo
15	1	1	1	1	1	1	Blanco intenso

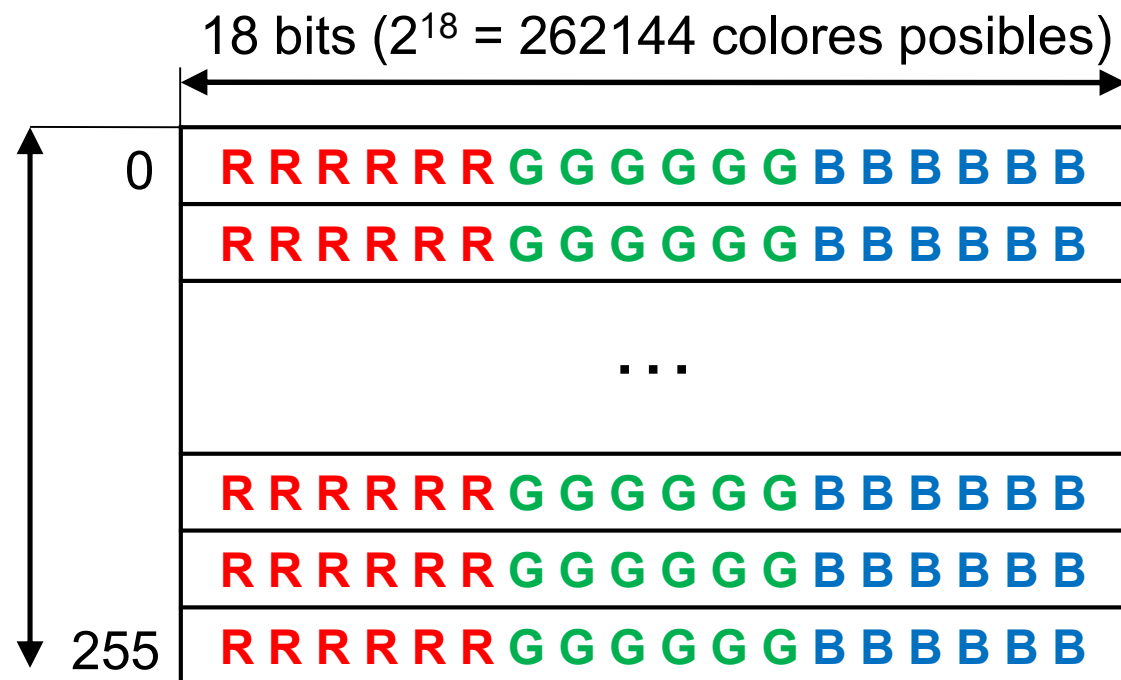
(6)

6.4. Controladora de Vídeo y Pantalla (XIV)

Modo VGA de texto

- Memoria de vídeo 256 KB.
- Se envía a la pantalla una **señal analógica** generada mediante un conversor digital a analógico (DAC).
- 256 registros de color de 18 bits.

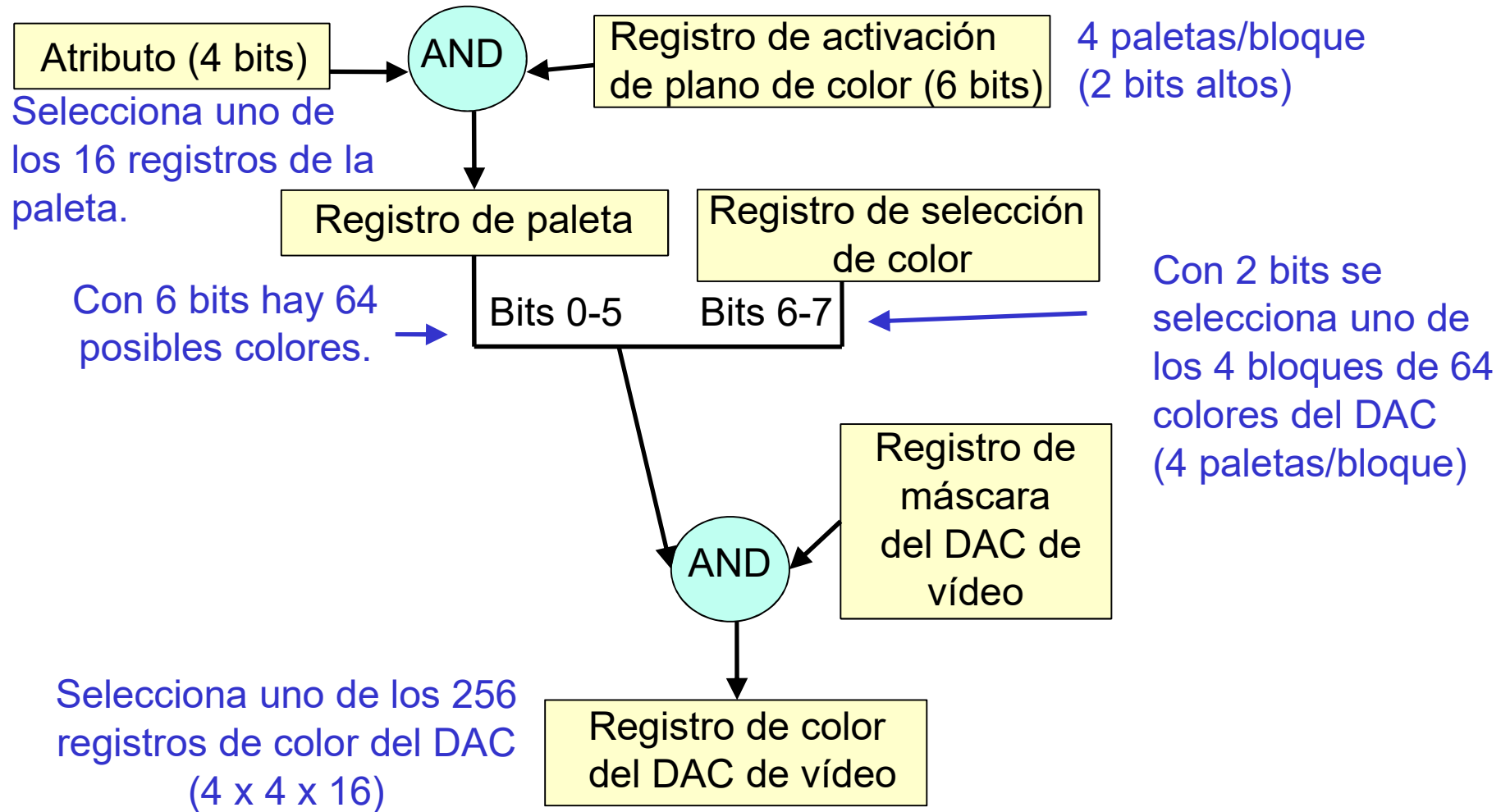
256 entradas =
4 bloques x
4 paletas/bloque x
16 colores/paleta



(6)

6.4. Controladora de Vídeo y Pantalla (XV)

Modo VGA de texto



Registros de color del DAC y registros de activación, selección y máscara definidos mediante BIOS (INT 10h).

(6)

6.4. Controladora de Vídeo y Pantalla (XVI)

• Modos gráficos

- La unidad representable es el píxel.
- Cada píxel tiene un color asociado.
- Los píxeles no parpadean.
- El color de cada píxel depende del contenido de un grupo de bits de una determinada posición en la memoria RAM de vídeo.
- El número de colores que pueden visualizarse simultáneamente depende del número de bits asociados a cada píxel.

• **Ejemplo:** Modo 13h \Rightarrow VGA (320x200, 256 colores)

(6)

6.4. Controladora de Vídeo y Pantalla (XVII)

VGA (320x200, 256 colores)

- Cada uno de los píxels viene definido por un atributo de 8 bits que se almacena en una posición de la RAM de vídeo.
- Un píxel puede visualizarse en uno de 256 colores (8 bits) diferentes.
- El valor del atributo de cada píxel indexa uno de los 256 registros de color del DAC.
- La memoria de vídeo es lineal, empezando en A000:0000 y terminando en A000:FA00.

(6)

6.4. Controladora de Vídeo y Pantalla (XVIII)

AH	Función INT 10h (BIOS)
00h	Establecer modo de pantalla
	Entrada: AL=00h, 40 x 25 texto, grises; AL=01h, 40 x 25 texto, color; AL=02h, 80 x 25 texto, grises; AL=03h, 80 x 25 texto, color; AL=04h, 320 x 200 gráficos, color; AL=05h, 320 x 200 gráficos, grises; AL=06h, 640 x 200 gráficos, B y N; AL=07h, 80 x 25 texto, B y N; AL=10h, 640 x 350, gráficos, 16 colores; AL=12h, 640 x 480 gráficos, 16 colores; AL=13h, 320 x 200, gráficos, 256 colores
01h	Establecer tamaño del cursor
	Entradas: CH = línea inicial del cursor (0-15). Si es 0, la línea superior parpadea. CL = línea final del cursor (0-15)
02h	Posicionar el cursor
	Entradas: DH = fila (0-24), DL = columna (0-79), BH = número de página

(6)

6.4. Controladora de Vídeo y Pantalla (XIX)

AH	Función INT 10h (BIOS)
07h	Desplazamiento hacia abajo de una ventana en la página activa. Antes de utilizarla debe fijarse la página activa.
	Entrada: AL= número de líneas (las líneas de la parte superior de la ventana se borran; si AL=0 se borra toda la ventana); CH = fila superior, CL = columna izquierda, DH = fila inferior, DL = columna derecha, BH = atributo a utilizar en la línea en blanco
08h	Lee carácter y atributo de la posición actual del cursor (modo alfanumérico)
	Entradas: BH = número de página Salidas: AL = carácter leído, AH = atributo del carácter leído
09h	Escribe carácter y atributo en posición actual del cursor (ambos modos)
	Entradas: BH = número de página, BL = atributo del carácter, CX = número de caracteres a escribir, AL = carácter a escribir

(6)

6.4. Controladora de Vídeo y Pantalla (XX)

AH	Función INT 10h (BIOS)
0Ah	Escribe carácter en posición actual de cursor (ambos modos)
	Entradas: BH=número de página, BL=atributo, AL=carácter, CX= no. veces
0Bh	Establece color de fondo y paleta (gráficos, 320x200)
	Entradas: BH = 0 y BL = color del fondo; BH = 1 y BL = paleta (0-1)
0Ch	Escribir punto (modo gráfico)
0Dh	Leer punto (modo gráfico)
0Eh	Escribe carácter en posición actual del cursor y avanza el cursor
	Entradas: AL = carácter a escribir BH = número de página activa
0Fh	Leer estado actual de la pantalla.
	Entradas: BH = número de página Salidas: AL = número del modo de pantalla AH = número de columnas en la pantalla, BH = número de la página activa

(6)

6.4. Controladora de Vídeo y Pantalla (XXI)

AH	AL	Función INT 10h (BIOS)
10h	10h	Actualiza el registro de color del DAC (VGA)
		Entradas: BX = no. registro de color (0..255), CH = valor del verde, CL = valor del azul, DH = valor del rojo (valores de color: 0..63)
10h	12h	Actualiza un bloque de registros de color del DAC (VGA)
		Entradas: BX=primer registro a actualizar, CX=número de registros A actualizar, ES:DX=Tabla de valores rojo-azul-verde
10h	13h	Establecer el espacio de selección
		Entradas: BL= 0 y BH=0 \Rightarrow 4 de 64, BL=0 y BH=1 \Rightarrow 16 de 16
10h	15h	Lee el registro de color del DAC (VGA)
		Entradas: BL=número de registro de color (0..255) Salidas: CH=valor del verde, CL=valor del azul, DH=valor del rojo
10h	17h	Lee un bloque de registros de color del DAC
		Entradas: BX=primer registro a leer, CX=número de registros a leer, ES:DX=Tabla de valores rojo-azul-verde Salidas: Tabla actualizada
10h	18h	Actualiza registro máscara del DAC
		Entradas: BL=nuevo valor
10h	19h	Lee registro máscara del DAC
		Salidas: BL= valor del registro máscara

(6)

6.4. Controladora de Vídeo y Pantalla (XXII)

AH	Función INT 21h (DOS)
02h	Escribir un carácter por pantalla en posición del cursor. Entradas: DL = código ASCII de carácter.
09h	Escribir una cadena de caracteres en posición del cursor. La cadena debe terminar con el carácter '\$'. Entradas: DS:DX = Dirección de la cadena.

(6)

6.5. Puerto Paralelo. Impresora (I)

- El PC soporta hasta 4 puertos paralelos (LPT1, LPT2, LPT3, LPT4)
- Cada puerto paralelo se maneja mediante tres registros accesibles en direcciones de E/S consecutivas:
 - **Registro de datos:** en el que la CPU envía datos o lee datos si el puerto es bidireccional.
 - **Registro de estado:** en el que la CPU lee el estado del puerto paralelo (ej.: impresora apagada, sin papel, etc.).
 - **Registro de control:** en el que la CPU envía señales de control al puerto paralelo (ej.: validación de dato, inicialización de la impresora, etc.)

(6)

6.5. Puerto Paralelo. Impresora (II)

- Tipos de puerto paralelo (seleccionados por BIOS):
 - Puerto paralelo estándar (SPP)
 - Registro de datos de salida (8 bits).
 - Registro de estado de entrada (4 bits)
 - CPU envía cada byte de datos y gestiona protocolo.
 - Puerto bidireccional (PS/2 o extendido)
 - Registro de datos de entrada o salida (8 bits).
 - Registro de estado de entrada (4 bits)
 - CPU envía/**recibe** cada byte de datos y gestiona protocolo.
 - Puerto paralelo mejorado (EPP)
 - Registro de datos de entrada o salida (8 bits).
 - CPU envía/recibe cada byte de datos.
 - **Puerto gestiona protocolo.**
 - Configuración de señales diferente a SPP y bidireccional.
 - Puerto con capacidades mejoradas (ECP)
 - Igual que EPP pero las **transferencias de bytes de datos se realizan por DMA.**

(6)

6.5. Puerto Paralelo. Impresora (III)

Tipo de Puerto Paralelo	Modo de Entrada	Modo de Salida	Velocidad
SPP (Estándar)	Nibble (4 bits) (Reg. Estado)		50 KByte/s
SPP (Estándar)		Compatible (8 bits)	150 KB/s
Bidireccional (PS/2 o Extendido) (1987)	Byte (8 bits)		150 KB/s
Bidireccional (PS/2 o Extendido) (1987)		Compatible (8 bits)	150 KB/s
EPP (P. Paralelo Mejorado) (IEEE 1284)(1991) (Periféricos)	EPP (8 bits)		500 KB/s – 2 MB/s
EPP (P. Paralelo Mejorado) (IEEE 1284)(1991) (Periféricos)		EPP (8 bits)	500 KB/s – 2 MB/s
ECP (P. Capacidades Mejoradas) (IEEE 1284) (DMA) (1992)	ECP (8 bits)		500 KB/s – 2 MB/s
ECP (P. Capacidades Mejoradas) (IEEE 1284)(DMA) (1992)		ECP (8 bits)	500 KB/s – 2 MB/s

(6)

6.5. Puerto Paralelo. Impresora (IV)

- Durante el arranque del PC, las rutinas del BIOS se encargan de:
 - Detectar los puertos paralelos instalados.
 - Almacenar direcciones base y *time-out*.

0000h:0408h	0000h:040Ah	0000h:040Ch	0000h:040Eh
Dirección base	Dirección base	Dirección base	Dirección base
LPT1	LPT2	LPT3	LPT4

0000h:0478h	0000h:0479h	0000h:047Ah	0000h:047Bh
Time-out LPT1	Time-out LPT2	Time-out LPT3	Time-out LPT4

- **Time-out** : segundos que se esperará como máximo a que suceda algún evento. En el caso de la impresora, es el tiempo que se puede esperar hasta determinar que la impresora no está disponible (apagada, off-line, sin papel, ...)

(6)

6.5. Puerto Paralelo. Impresora (V)

- Para acceder a los puertos es necesario usar IN o OUT especificando el registro.
- Direcciones habituales de registros (dependen del fabricante de la placa base):

Puerto	Registro de datos	Registro de estado	Registro de control
LPT1	03BCh	03BDh	03BEh
LPT2	0378h	0379h	037Ah
LPT3	0278h	0279h	027Ah

- El LPT4 sólo se usa en algunos casos.

(6)

6.5. Puerto Paralelo. Impresora (VI)

- **Registro de datos:** Dir. base.
 - Registro de 8 bits que se corresponden con los pines 2 a 9 del conector externo (DB-25) del puerto paralelo.
 - En modo SPP es un puerto de salida para enviar datos a la impresora.
 - En modo bidireccional permite enviar o recibir datos (no simultáneamente \Rightarrow *half duplex*).

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

(6)

6.5. Puerto Paralelo. Impresora (VII)

- **Registro de estado:** Dir. base + 1
 - Sólo de entrada (se usan 5 bits).

7	6	5	4	3	2	1	0
<u>BUSY</u> (inv)	<u>ACK</u>	PE	SLCT	<u>ERROR</u>	----	----	----

- **BUSY (inv):** 0 indica impresora ocupada. En situaciones de error también se pone a 0.
- **ACK:** 0 indica que impresora ha recibido un dato y que está preparada para recibir otro.
- **PE:** 1 indica impresora sin papel.
- **SLCT:** 1 indica que la impresora está *on-line*.
- **ERROR:** 0 indica error en la impresora (sin papel, mal funcionamiento, etc.).
- (inv) Señal invertida por hardware en el conector DB-25.

(6)

6.5. Puerto Paralelo. Impresora (VIII)

- **Registro de control:** Dir. base + 2.
 - Sólo de salida (se usan 6 bits).

7	6	5	4	3	2	1	0
----	----	BIDIR	IRQEN	SLCT IN (inv)	<u>INIT</u>	AUTOFD (inv)	STROBE (inv)

- **BIDIR:** 1 indica puerto de entrada. 0 (defecto) indica salida. Hay que restaurarlo a 1 cada vez que se recibe un dato.
- **IRQEN:** 1 permite que señal ACK active IRQ7 del PIC maestro.
- **SLCT IN (inv):** 1 indica a la impresora que ha sido seleccionada.
- **INIT:** Normalmente a 1. Cuando se envía un pulso a 0, se provoca un *reset* en la impresora.
- **AUTOFD (inv):** Cuando este bit es 1, la impresora hará un salto de línea (line feed) al recibir un carácter de *retorno de carro* (13).
- **STROBE (inv):** La transición 0-1 en este bit indica a la impresora que el dato es válido y puede ser leído.

(6)

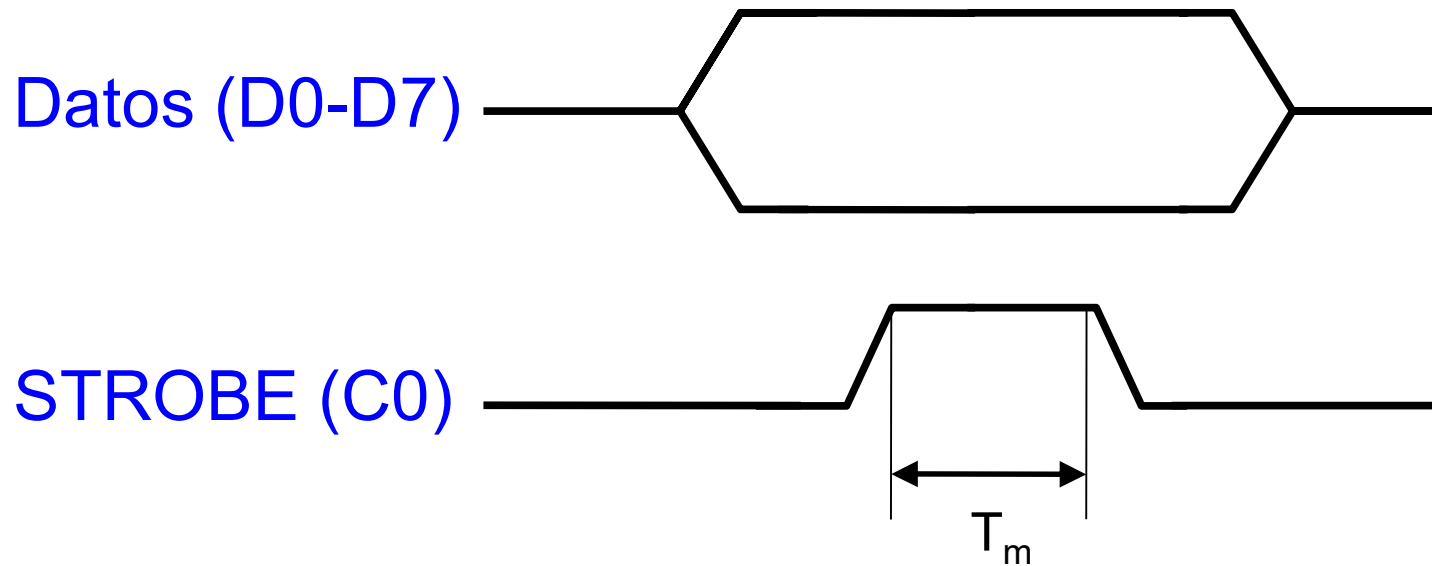
6.5. Puerto Paralelo. Impresora (IX)

- Necesario ejecutar protocolo para enviar un byte y esperar a que la impresora lo acepte:
 - Protocolo STROBE (más simple)
 - Protocolo BUSY (con realimentación y espera activa)
 - Protocolo ACK (basado en interrupciones).

(6)

6.5. Puerto Paralelo. Impresora (X)

Protocolo STROBE

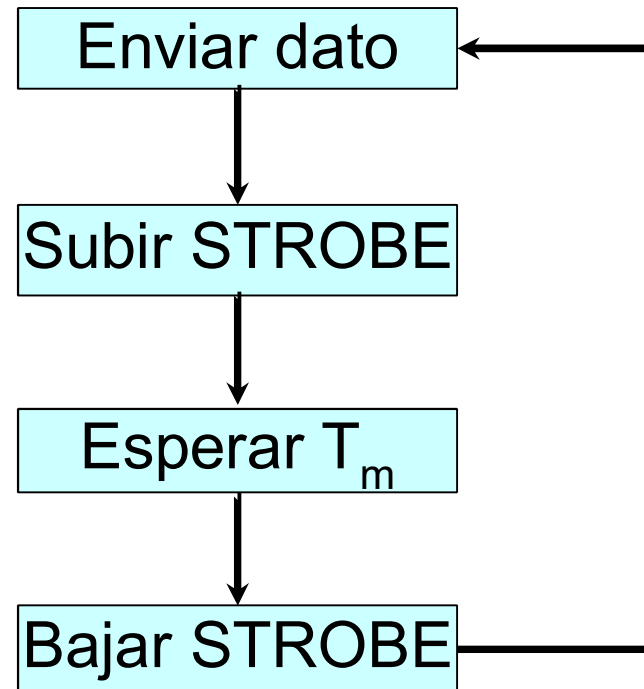


T_m = tiempo mínimo para que STROBE sea efectivo

(6)

6.5. Puerto Paralelo. Impresora (XI)

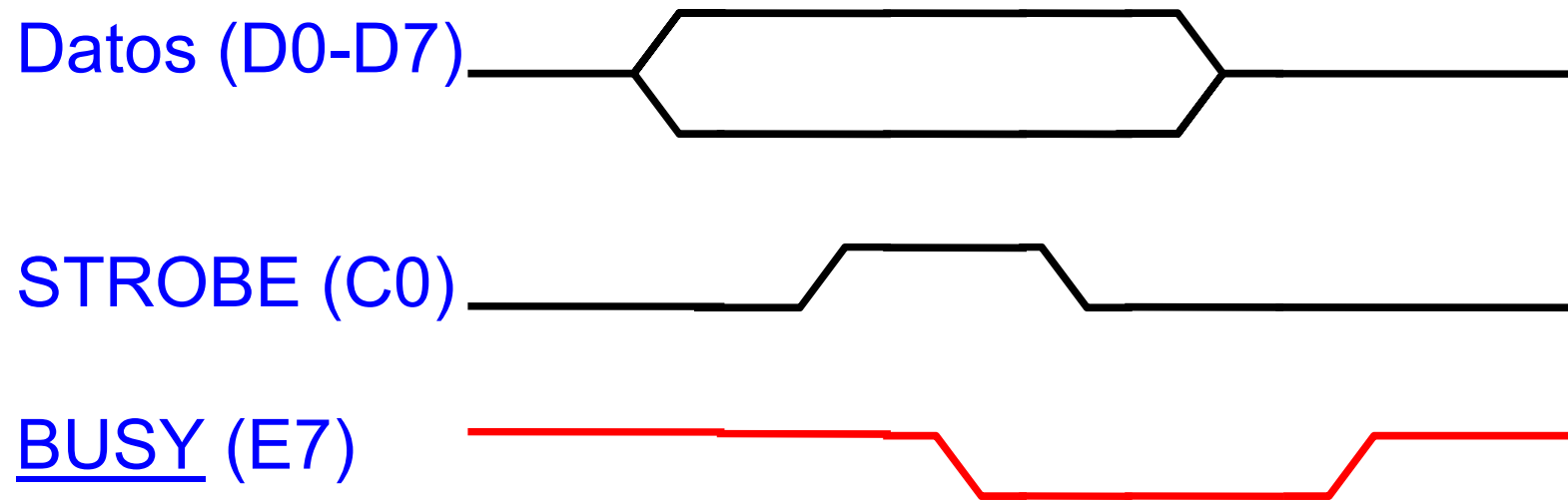
Protocolo STROBE



(6)

6.5. Puerto Paralelo. Impresora (XII)

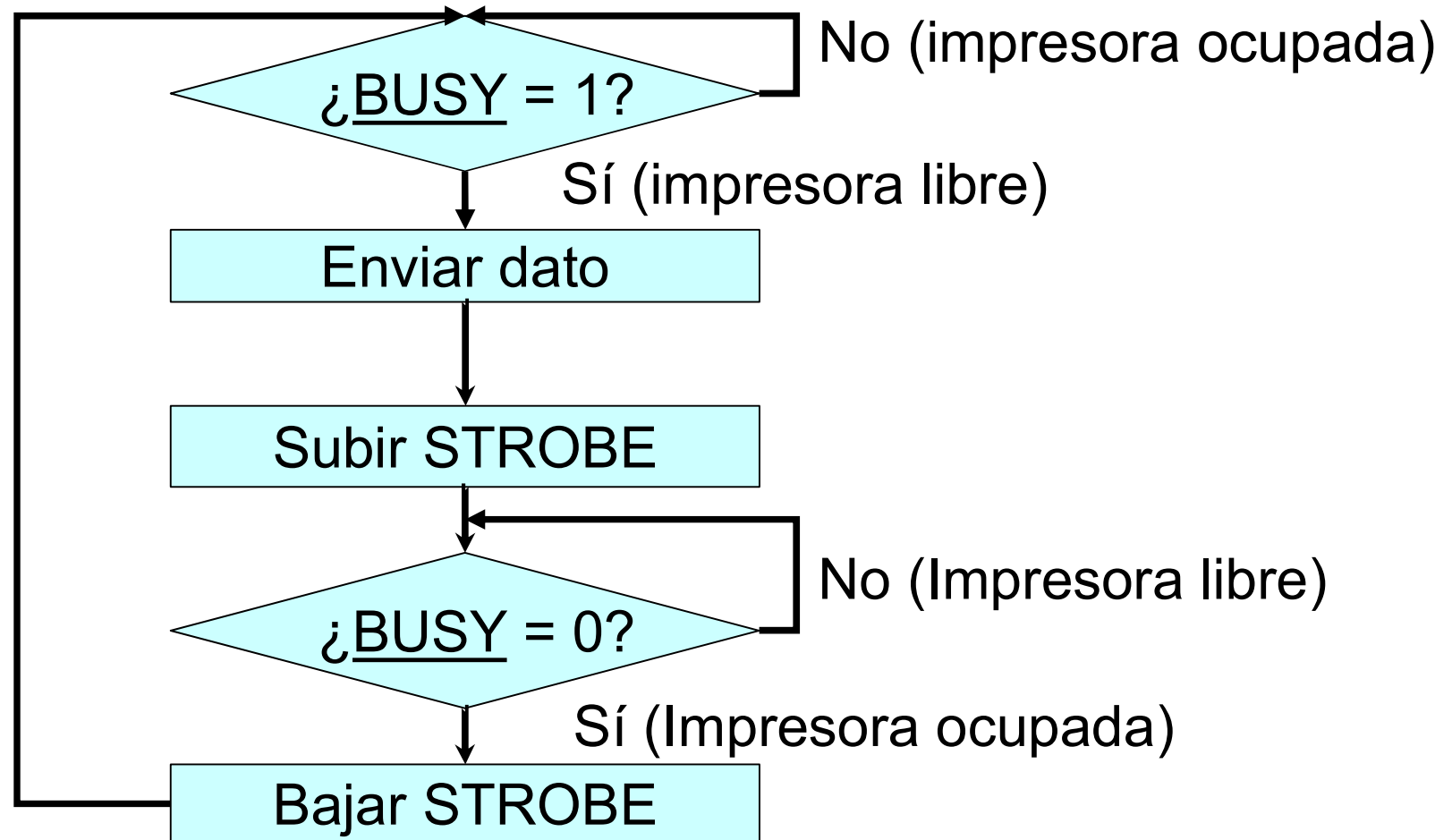
Protocolo BUSY



(6)

6.5. Puerto Paralelo. Impresora (XIII)

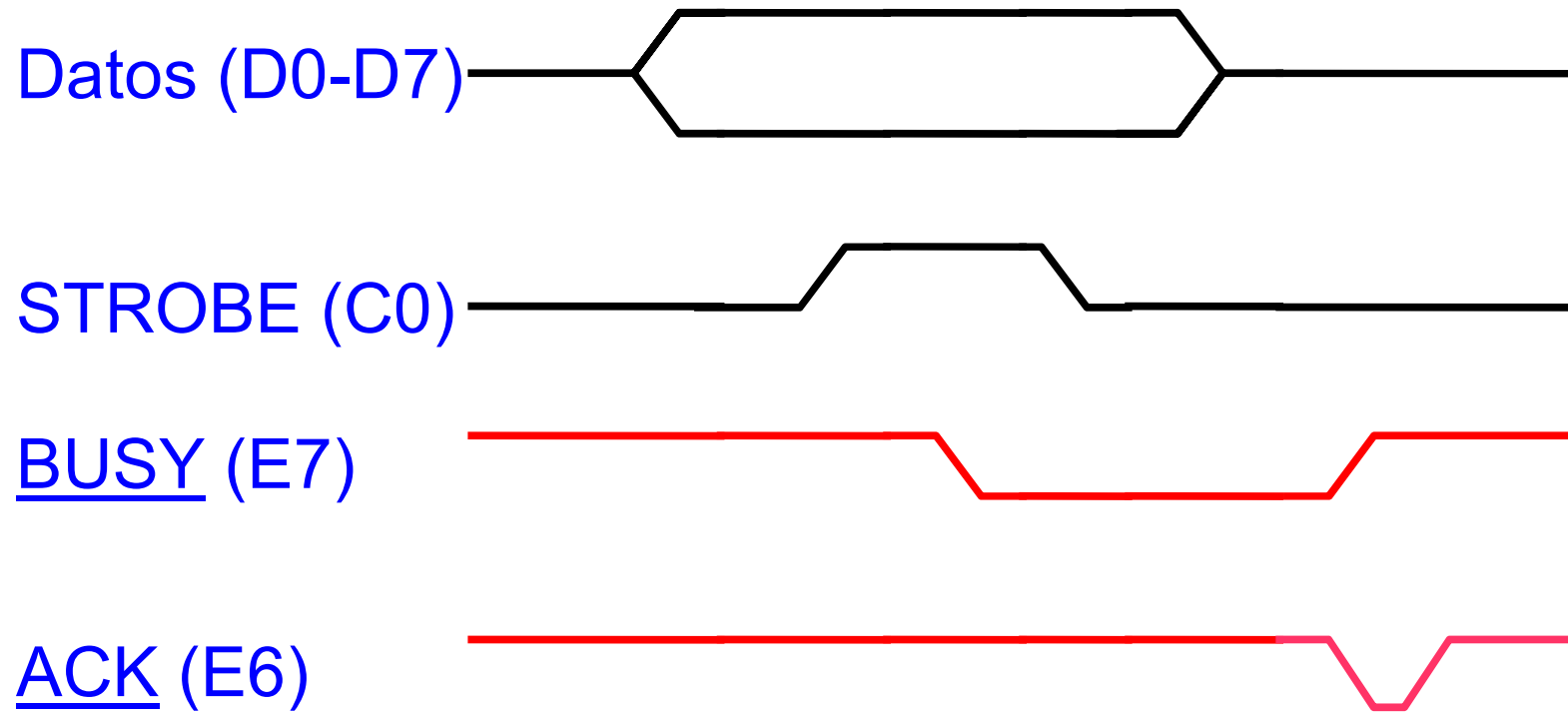
Protocolo BUSY



(6)

6.5. Puerto Paralelo. Impresora (XIV)

Protocolo ACK

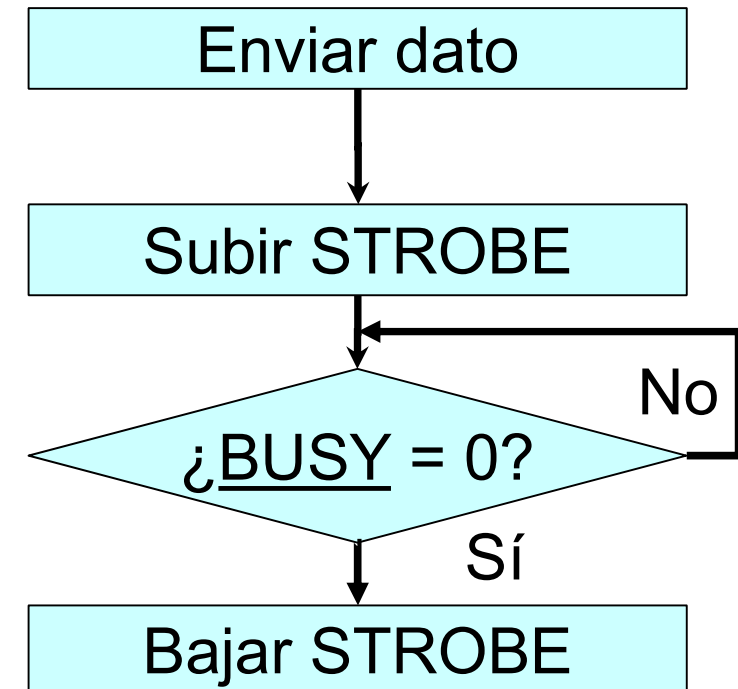


(6)

6.5. Puerto Paralelo. Impresora (XV)

Protocolo ACK

- ACK activa INT 0Fh.
- RSI inicia envío
- No es necesario esperar subida de BUSY al principio porque siempre que se genera un ACK es que BUSY ha subido.
- Primer dato se envía mediante interrupción software.
- Resto de datos se envían por interrupción *hardware*.



(6)

6.5. Puerto Paralelo. Impresora (XVI)

• Relación con pines del conector DB-25

• Registro de datos

7	6	5	4	3	2	1	0
PIN 9	PIN 8	PIN 7	PIN 6	PIN 5	PIN 4	PIN 3	PIN 2

• Registro de control

7	6	5	4	3	2	1	0
				<u>PIN 17</u>	PIN 16	<u>PIN 14</u>	<u>PIN 1</u>

• Registro de estado

7	6	5	4	3	2	1	0
<u>PIN 11</u>	PIN 10	PIN 12	PIN 13	PIN 15			

PIN n indica señal invertida en el conector mediante puerta inversora.

(6)

6.5. Puerto Paralelo. Impresora (XVII)

● Interrupciones asociadas a la impresora

INT	AH	Función	I/O	
5h		Imprimir pantalla		Se llama desde el teclado
17h	0h	Imprimir carácter	Entradas	AL : ASCII a imprimir DX : impresora a utilizar (1,2,3)
			Salida	AH : byte de estado
	1h	Inicializar impresora	Entrada	DX : impresora a utilizar (1,2,3)
			Salida	AH : byte de estado
	2h	Leer estado de impresora	Entrada	DX : impresora a utilizar (1,2,3)
			Salida	AH : byte de estado
21h	5h	Imprimir carácter	Entrada	DL : carácter a imprimir

(6)

6.6. Puerto Serie Asíncrono (I)

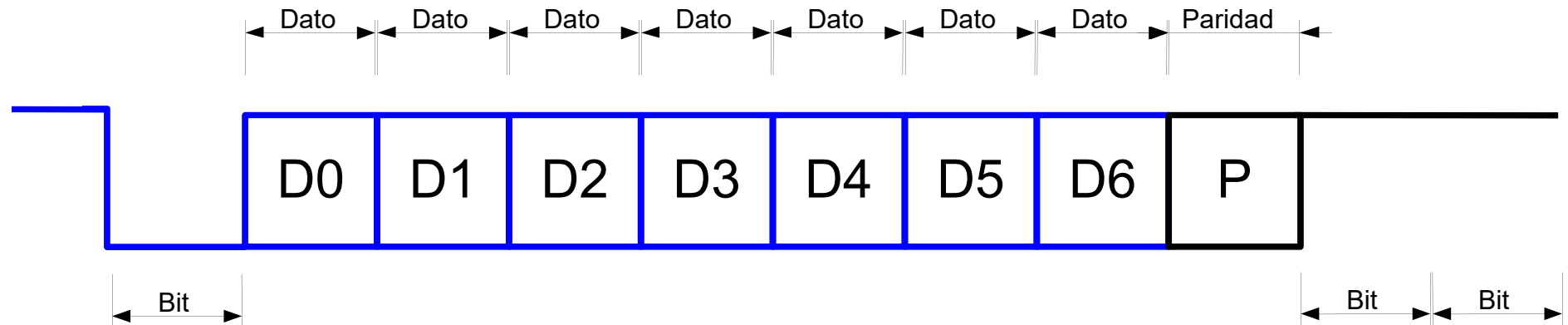
- Basado en un circuito integrado de *National Semiconductor* (UART 8250).
- Sirve para la transmisión de datos en serie.
- Permite la transmisión a mayores distancias que en paralelo (ej. 15 metros usando codificación EIA RS-232-C).
- Los bits de datos y de control se transmiten de forma sucesiva a través de una única línea.
- La anchura en tiempo de cada bit depende de la velocidad de transmisión, que se expresa en bits por segundo (bps).
- Tanto el emisor como el receptor deben estar configurados con los mismos valores de longitud de dato, tipo de paridad, velocidad de transmisión y número de bits de parada.

(6)

6.6. Puerto Serie Asíncrono (II)

● Transmisión asíncrona

Bit de paridad



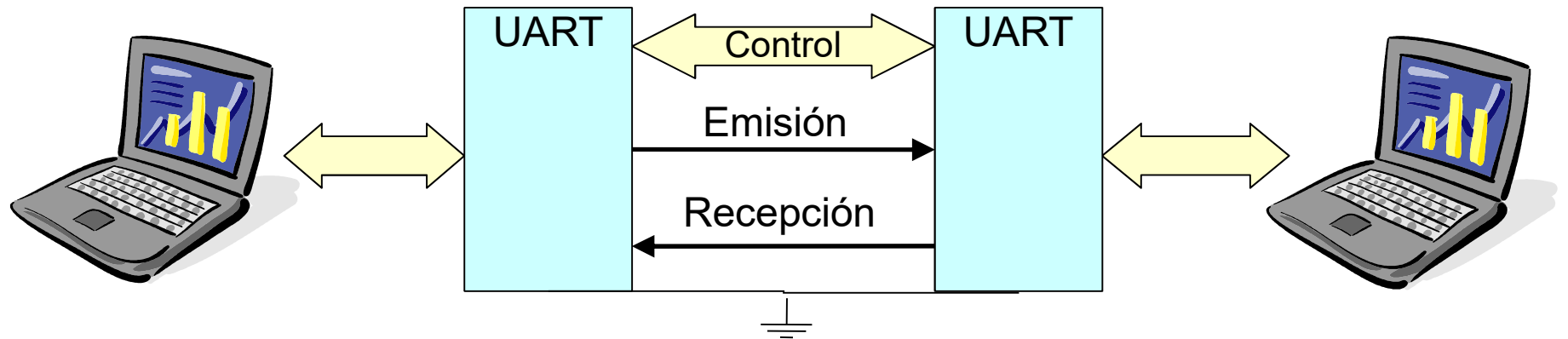
Bit de inicio

1, 1.5 o 2 bits
de parada

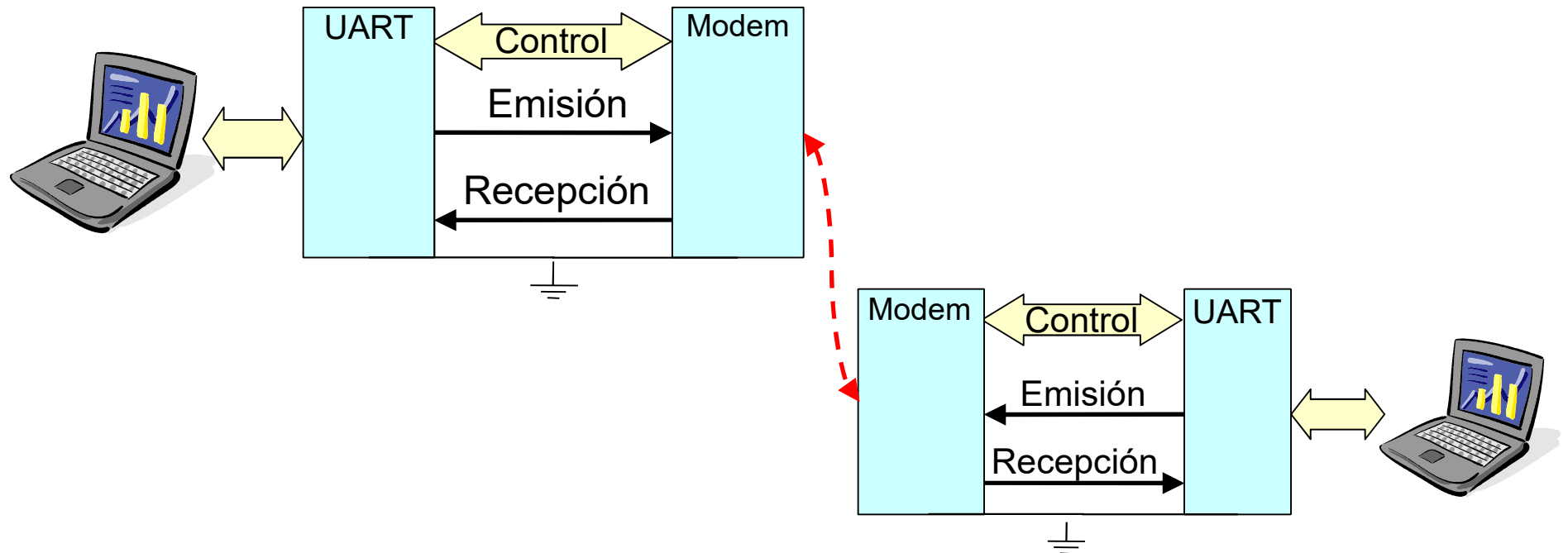
(6)

6.6. Puerto Serie Asíncrono (III)

- Transmisión serie directa:



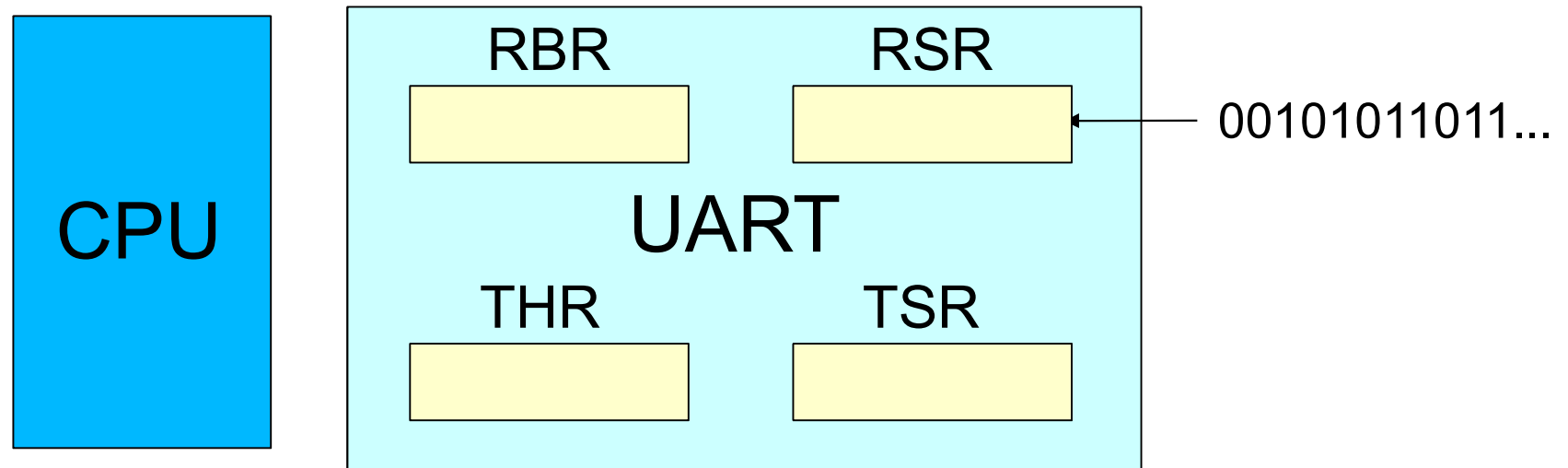
- Transmisión serie usando línea telefónica:



(6)

6.6. Puerto Serie Asíncrono (IV)

- Entrada en transmisión serie directa:



RBR (Receiver Buffer Register)

RSR (Receiver Shift Register)

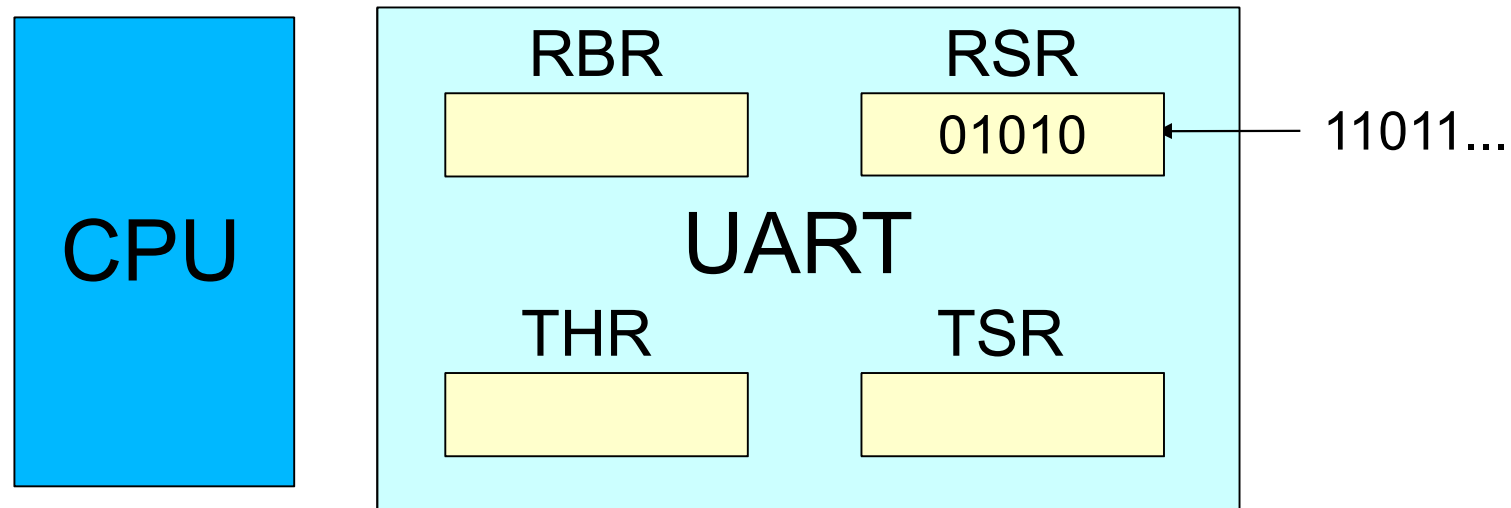
THR (Transmitter Holding Register)

TSR (Transmitter Shift Register)

(6)

6.6. Puerto Serie Asíncrono (V)

- Se elimina el bit de inicio y los bits van siendo cargados en el RSR uno a uno.
- Se reciben los bits de parada.



RBR (Receiver Buffer Register)

RSR (Receiver Shift Register)

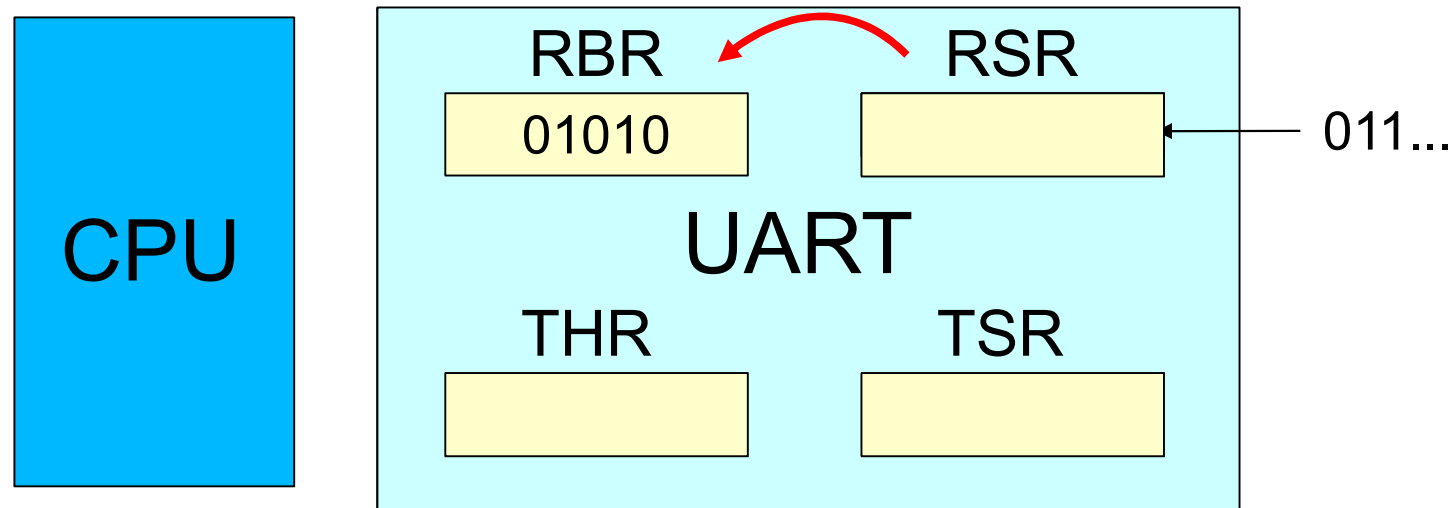
THR (Transmitter Holding Register)

TSR (Transmitter Shift Register)

(6)

6.6. Puerto Serie Asíncrono (VI)

- Cuando el RSR tiene todos los bits que corresponden al dato, se vuelcan en el RBR y se vacía RSR.



RBR (Receiver Buffer Register)

RSR (Receiver Shift Register)

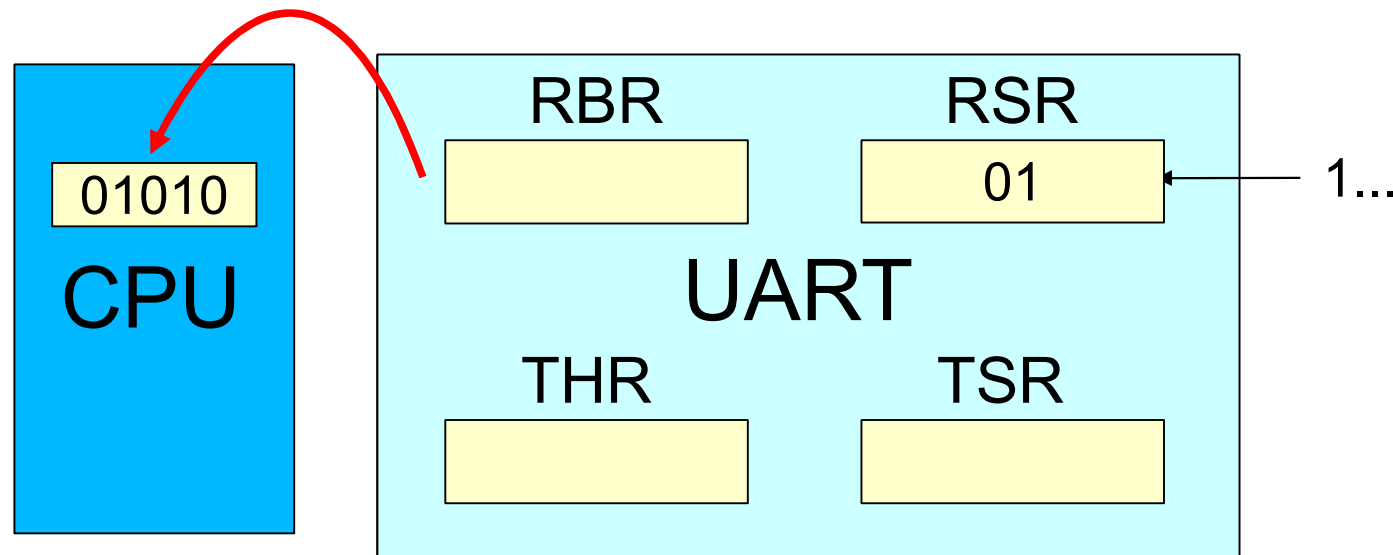
THR (Transmitter Holding Register)

TSR (Transmitter Shift Register)

(6)

6.6. Puerto Serie Asíncrono (VII)

- El RSR está vacío y sigue admitiendo nuevos datos.
- Se transmiten los datos a la CPU desde el RBR y éste queda vacío.



RBR (Receiver Buffer Register)

RSR (Receiver Shift Register)

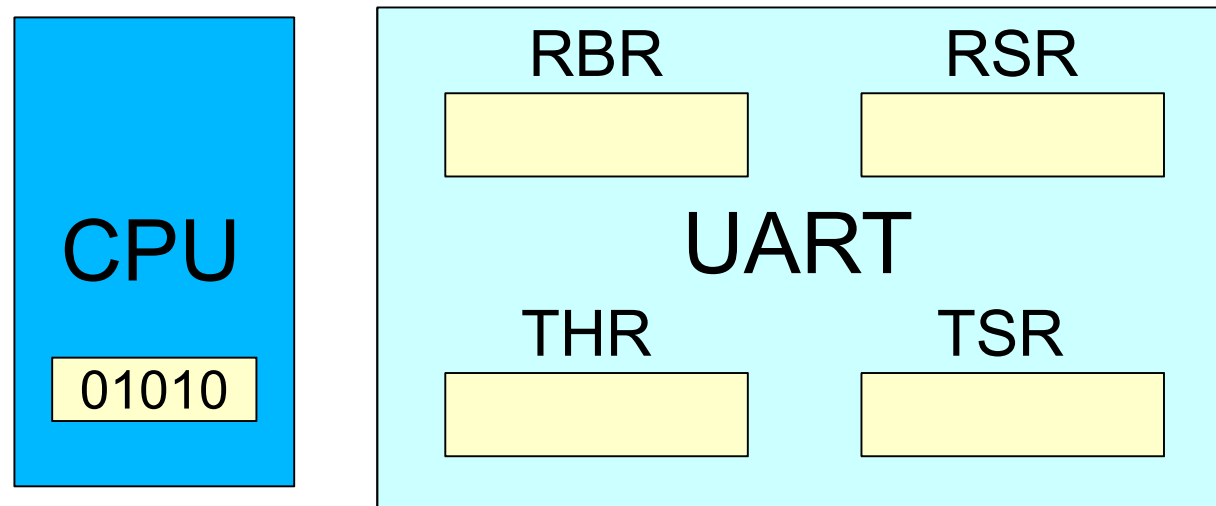
THR (Transmitter Holding Register)

TSR (Transmitter Shift Register)

(6)

6.6. Puerto Serie Asíncrono (VIII)

- Salida en transmisión serie directa:



RBR (Receiver Buffer Register)

RSR (Receiver Shift Register)

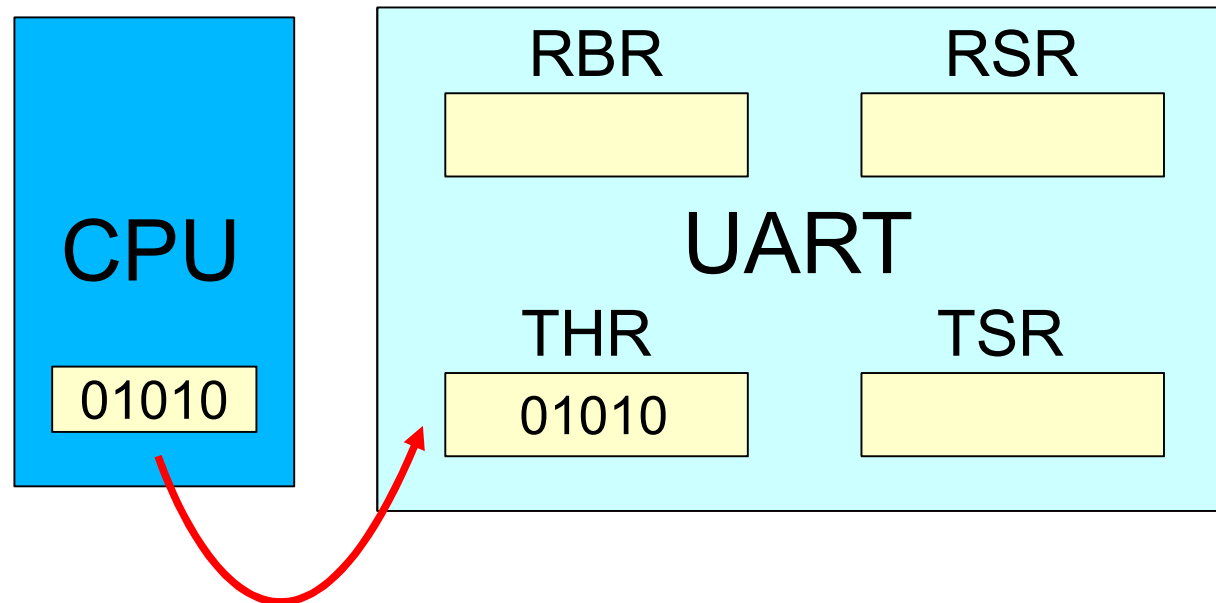
THR (Transmitter Holding Register)

TSR (Transmitter Shift Register)

(6)

6.6. Puerto Serie Asíncrono (IX)

- La CPU escribe el dato en el THR.



RBR (Receiver Buffer Register)

RSR (Receiver Shift Register)

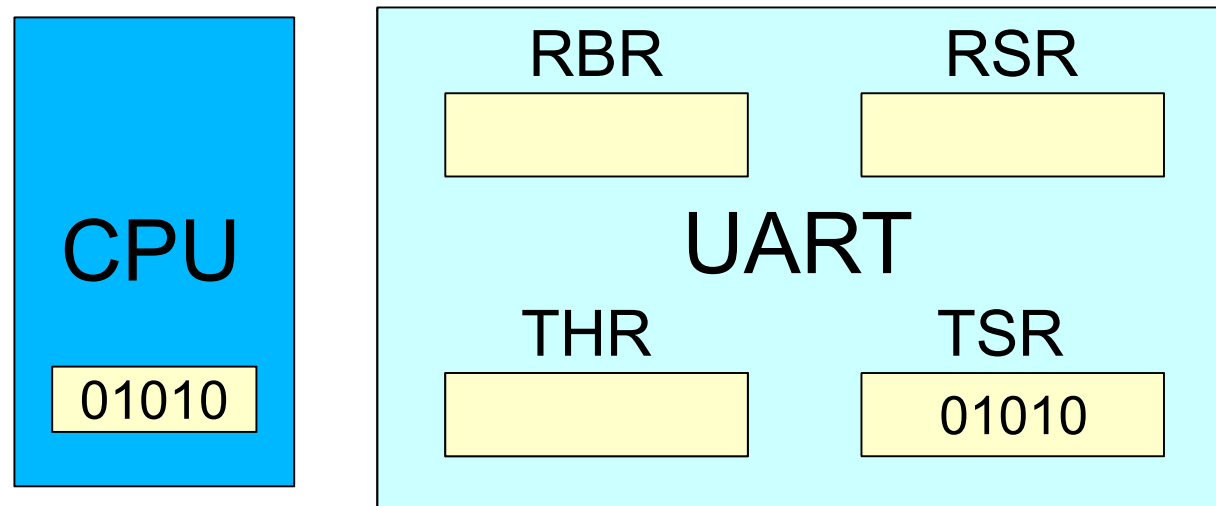
THR (Transmitter Holding Register)

TSR (Transmitter Shift Register)

(6)

6.6. Puerto Serie Asíncrono (X)

- El UART vuelca el dato automáticamente en el TSR.



RBR (Receiver Buffer Register)

RSR (Receiver Shift Register)

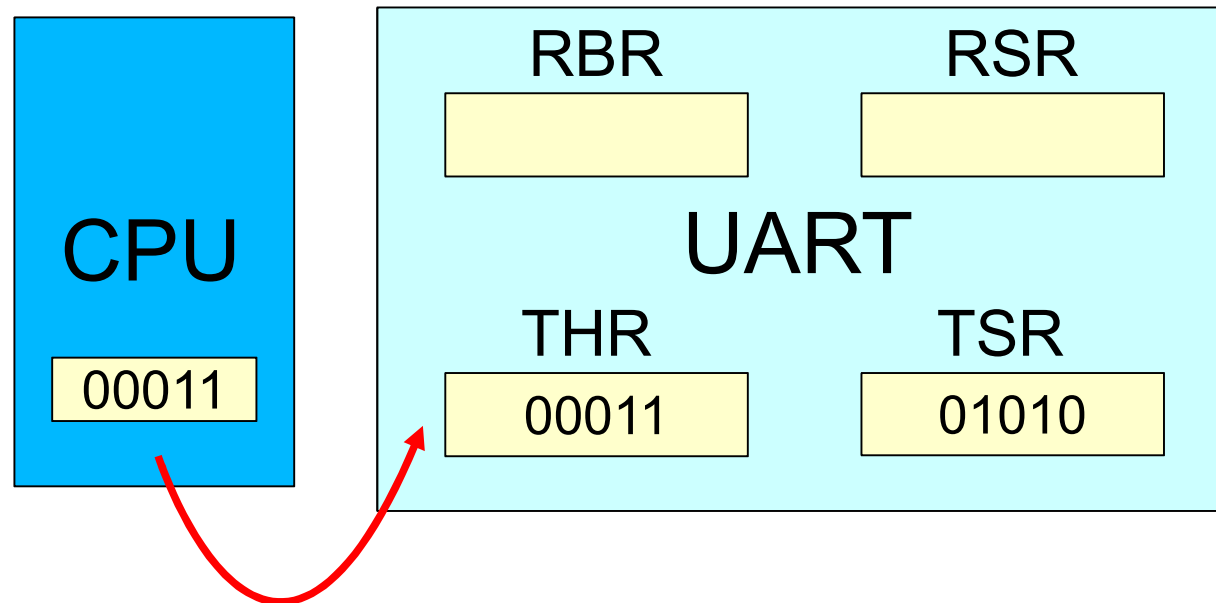
THR (Transmitter Holding Register)

TSR (Transmitter Shift Register)

(6)

6.6. Puerto Serie Asíncrono (XI)

- La CPU puede volver a escribir otro dato en el THR.



RBR (Receiver Buffer Register)

RSR (Receiver Shift Register)

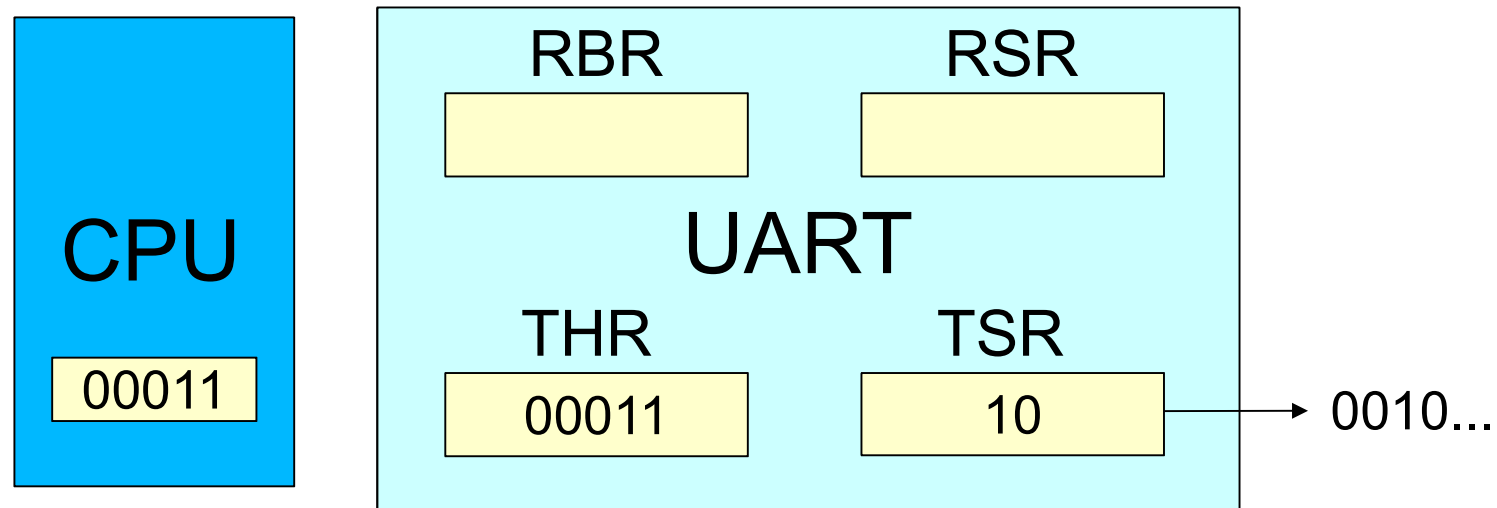
THR (Transmitter Holding Register)

TSR (Transmitter Shift Register)

(6)

6.6. Puerto Serie Asíncrono (XII)

- El UART envía el bit de inicio.
- A continuación envía todos los bits a partir del de menor peso.



RBR (Receiver Buffer Register)

RSR (Receiver Shift Register)

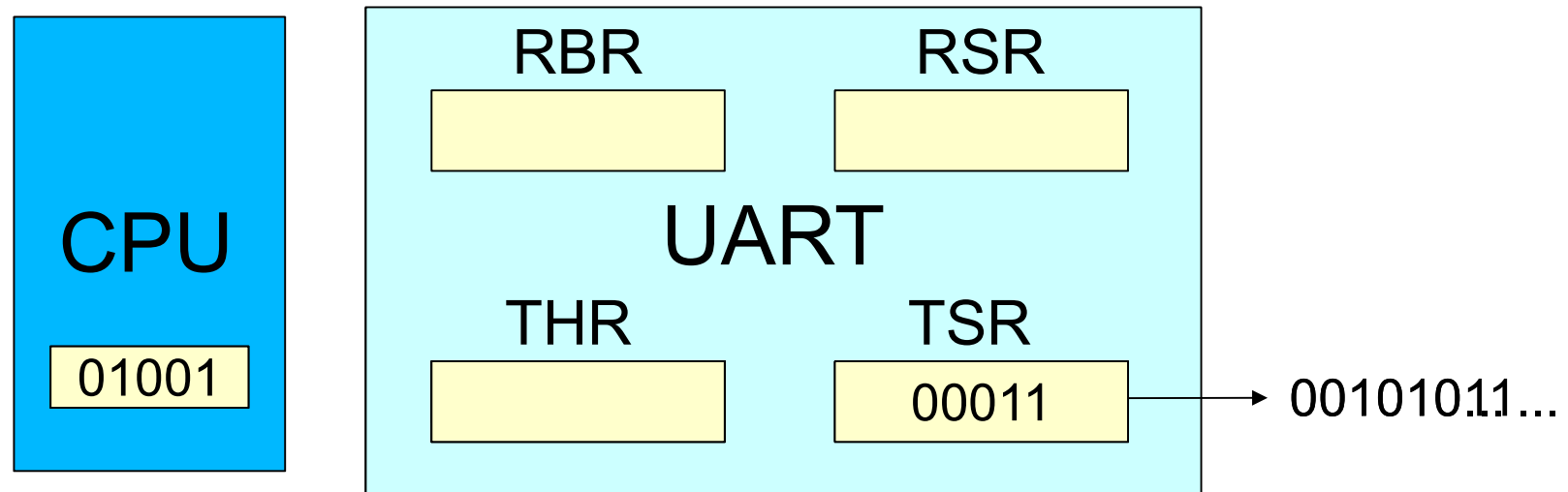
THR (Transmitter Holding Register)

TSR (Transmitter Shift Register)

(6)

6.6. Puerto Serie Asíncrono (XIII)

- Cuando ha enviado dato, el UART añade el bit de paridad.
- Al quedar vacío TSR, el UART escribe nuevo dato en TSR.
- La CPU puede volver a escribir en THR.



RBR (Receiver Buffer Register)

RSR (Receiver Shift Register)

THR (Transmitter Holding Register)

TSR (Transmitter Shift Register)

(6)

6.6. Puerto Serie Asíncrono (XIV)

- Líneas de comunicación del UART con el exterior:

Líneas	Significado
SOUT	Línea de transmisión serie (*)
SIN	Línea de recepción serie (*)
<u>OUT1</u>	Salida digital
<u>OUT2</u>	Salida digital

- Las líneas con * tienen niveles lógicos TTL y es necesario un chip (buffer) para convertirlos a niveles EIA antes de salir por la línea de transmisión serie.
- En norma EIA RS232-C, un 1 lógico se convierte a $-12V$ y un 0 lógico a $+12V$.

(6)

6.6. Puerto Serie Asíncrono (XV)

- Líneas de protocolo externo del UART:

Líneas	Dirección	Significado
<u>DTR</u>	Salida	(Data Terminal Ready) El UART comunica al modem que está listo para la comunicación
<u>DSR</u>	Entrada	(Data Set Ready) El modem comunica al UART que está listo para la comunicación
<u>RTS</u>	Salida	(Request To Send) El UART comunica al modem que está listo para enviarle un dato
<u>CTS</u>	Entrada	(Clear To Send) El modem comunica al UART que está listo para enviar un dato a la línea
<u>RI</u>	Entrada	(Ring Indicator) Teléfono del modem está sonando
<u>DCD</u>	Entrada	(Data Carrier Detect) Teléfono remoto ha descolgado

(6)

6.6. Puerto Serie Asíncrono (XVI)

Registros	A2	A1	A0	DLAB	Acción
THR	0	0	0	0	Salida de datos (THR)
RBR	0	0	0	0	Entrada de datos (RBR)
DLL	0	0	0	1	Divisor de frecuencia (byte bajo)
IER	0	0	1	0	Habilitación de interrupciones
DLH	0	0	1	1	Divisor de frecuencia (byte alto)
IIR	0	1	0		Identificación de interrupción
LCR	0	1	1		Control de línea (paridad, longitud dato...)
MCR	1	0	0		Control de modem (DTR, RTS)
LSR	1	0	1		Estado de línea
MSR	1	1	0		Estado de modem (DSR, CTS, ...)
SCR	1	1	1		Uso libre ("Scratchpad")

● Direcciones base de puerto definidas en BIOS:

COM1: 0000h:0400h y 0000h:0401h

COM2: 0000h:0402h y 0000h:0403h

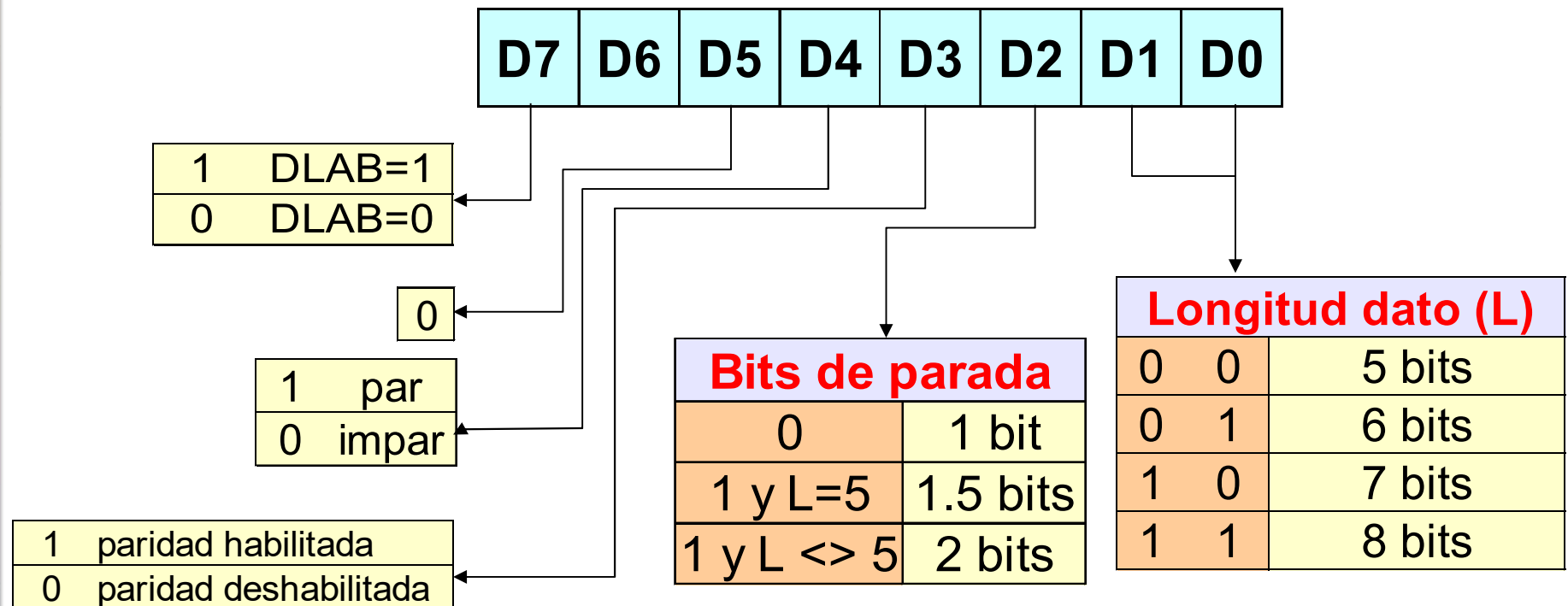
COM3: 0000h:0404h y 0000h:0405h

COM4: 0000h:0406h y 0000h:0407h

(6)

6.6. Puerto Serie Asíncrono (XVII)

- Registro de control de línea (LCR): Dir. base + 3
 - Registro de escritura.



- Cuando D6 se activa a 1, se genera en la línea de salida SOUT la condición de BREAK, que consiste en que SOUT se pone a 0 independientemente del estado en que se encuentre la transmisión. Este cambio puede ser detectado por receptor.

(6)

6.6. Puerto Serie Asíncrono (XVIII)

- Registros del divisor de frecuencia (DLL, DLH): Dir. base y Dir. base + 1 (DLAB=1)
 - Registros de escritura.
 - Dividen la frecuencia de transmisión.
 - La velocidad de transmisión en bps es:

$$v = \frac{1843200}{16 (256 DLH + DLL)}$$

$$DLH = (1843200 / 16v) \text{ div } 256$$

$$DLL = (1843200 / 16v) \text{ mod } 256$$

- Ejemplo: **DLH = 6** y **DLL = 0**

$$v = \frac{1843200}{16 (256 \times 6 + 0)} = 75 \text{ bps}$$

(6)

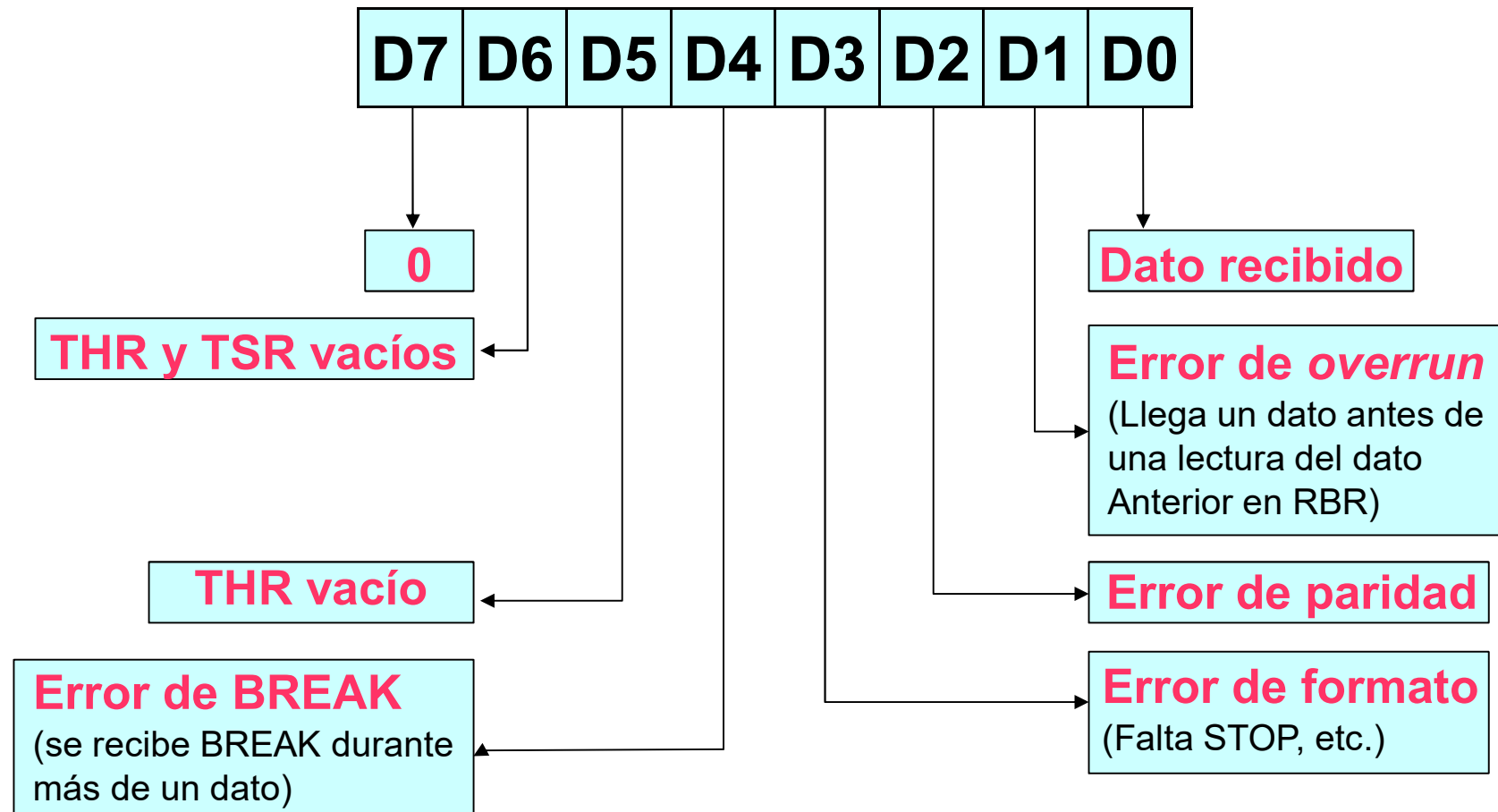
6.6. Puerto Serie Asíncrono (XIX)

- Registros de lectura (RBR) y escritura (THR) de datos:
Dir. base (DLAB=0).
 - Si se accede en **lectura** corresponde al RBR.
 - Si se accede en **escritura** corresponde al THR.

(6)

6.6. Puerto Serie Asíncrono (XX)

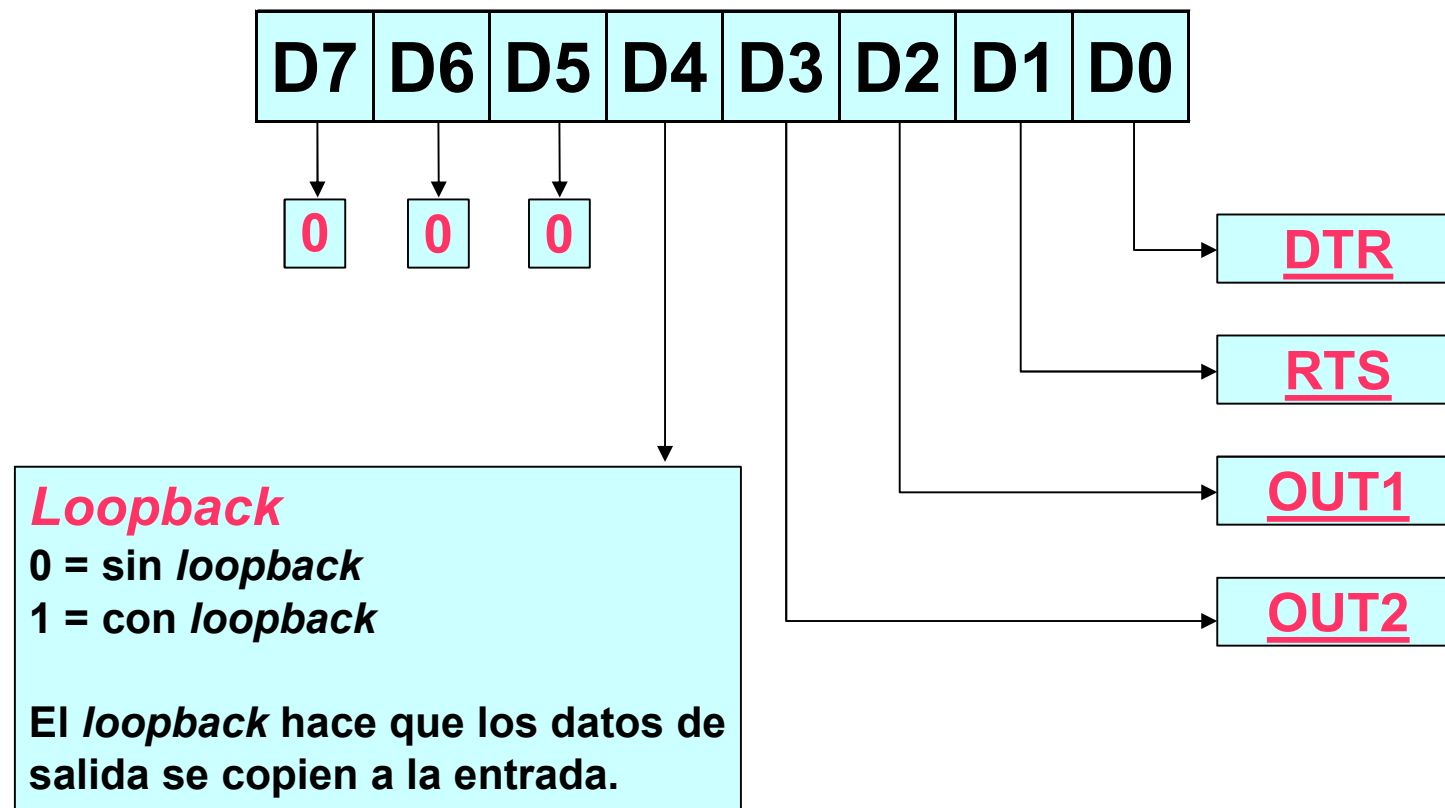
- Registro de estado de línea (LSR): Dir. base + 5
 - Registro de lectura.



(6)

6.6. Puerto Serie Asíncrono (XXI)

- Registro de control del modem (MCR): Dir. base + 4.
 - Registro de escritura.



(6)

6.6. Puerto Serie Asíncrono (XXII)

- Registro de estado del modem (MSR): Dir. base + 6.
 - Registro de lectura.

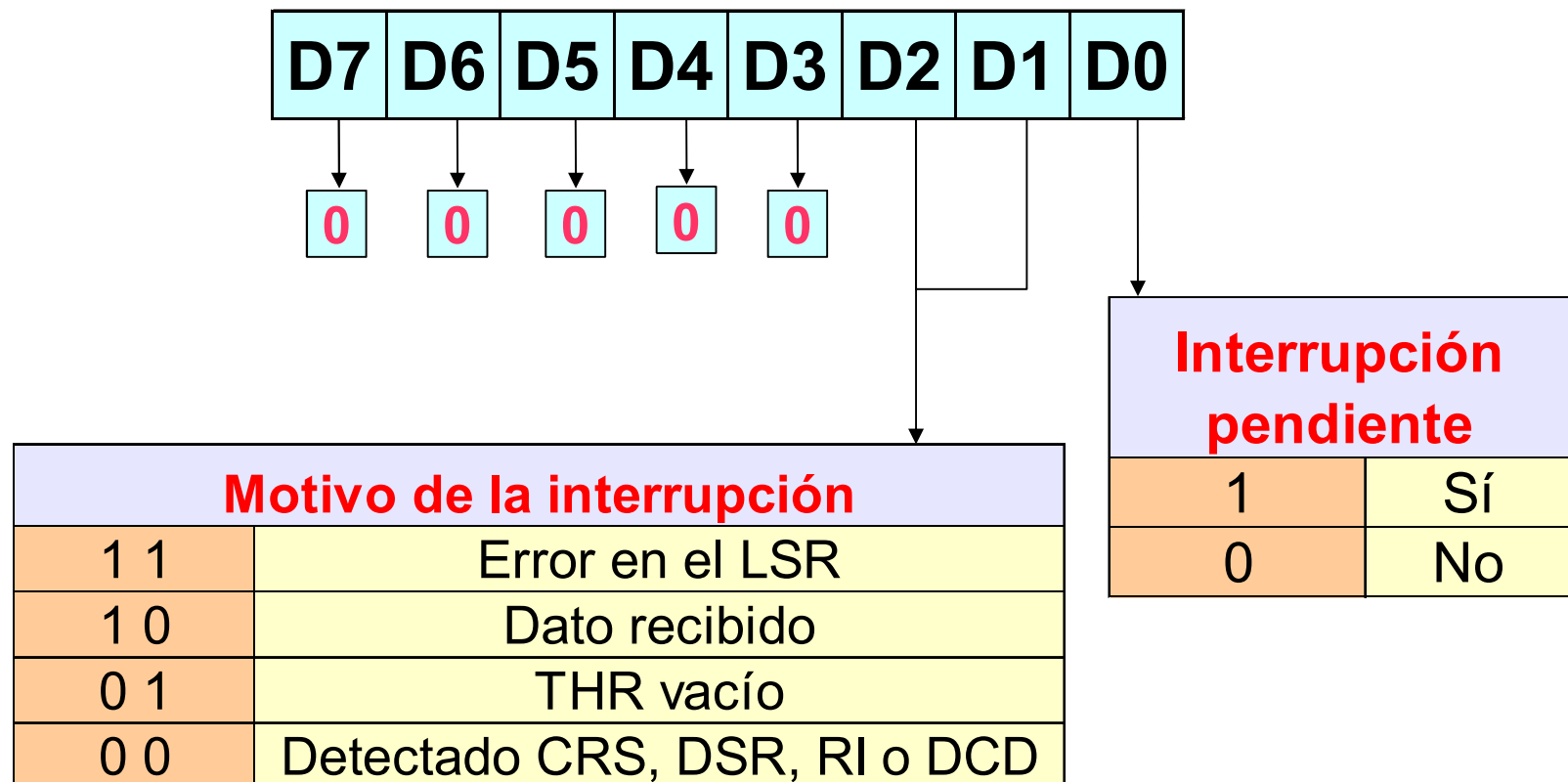
D7	D6	D5	D4	D3	D2	D1	D0
DCD	RI	DSR	CTS	DDCD	DRI	DDSR	DCTS

- Estado lógico complementado de los terminales.
- Informan de un cambio de estado desde la última vez que se leyó el MSR.
 - Cuando se lee MSR, se ponen a 0 y sólo indican transiciones de 0 a 1.

(6)

6.6. Puerto Serie Asíncrono (XXIII)

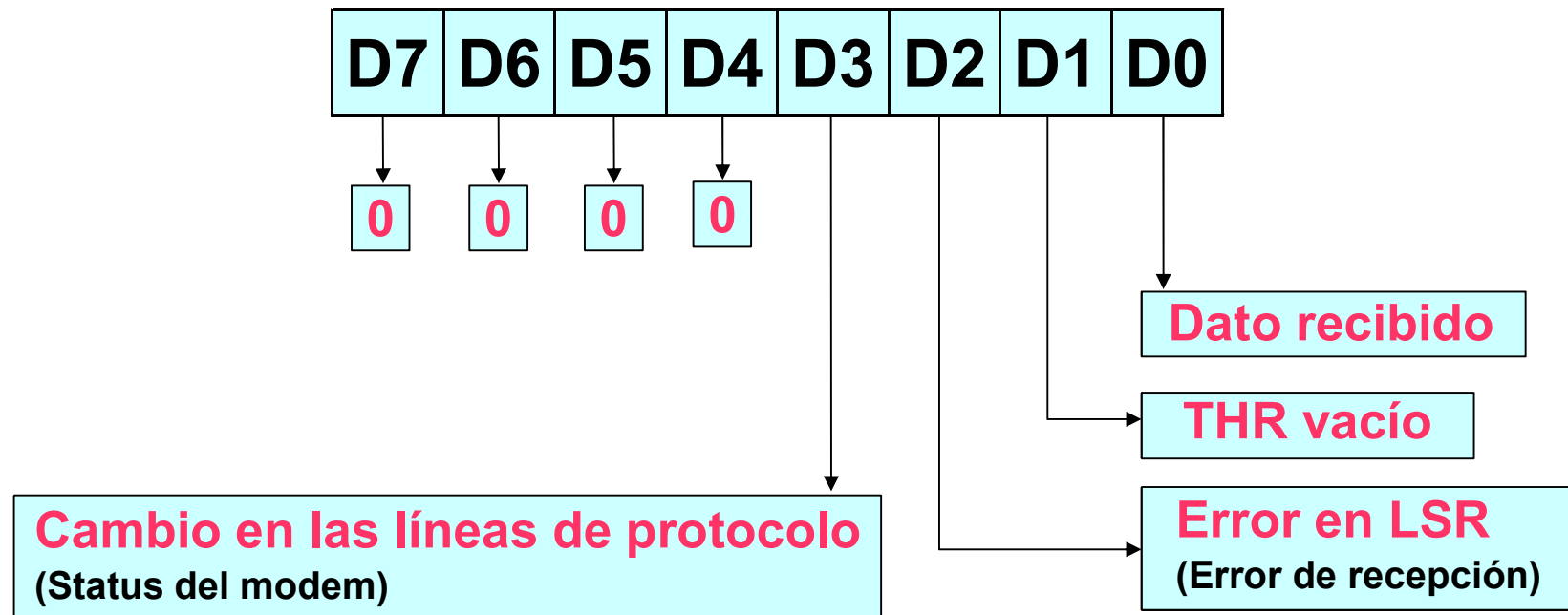
- Registro de identificación de interrupción (IIR): Dir. base+2
 - Registro de lectura.



(6)

6.6. Puerto Serie Asíncrono (XXIV)

- **Registro máscara de interrupciones (IER):** Dir. base + 1 (DLAB=0).
 - Registro de escritura.
 - Si se ponen a 1 habilitan la interrupción correspondiente y a 0 las inhabilitan.



(6)

6.6. Puerto Serie Asíncrono (XXV)

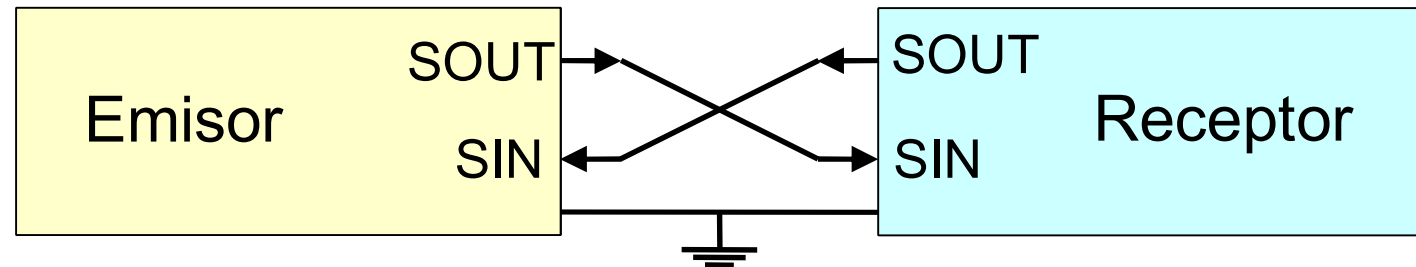
- Registro de scratch (SCR): Dir. base + 7.
 - Registro de lectura y escritura.
 - No realiza ninguna función sobre el UART.
 - Sirve para guardar algún dato de interés para el programador.

(6)

6.6. Puerto Serie Asíncrono (XXVI)

● Protocolo XON/XOFF

- Control de flujo usado cuando la transmisión es unidireccional (ej. entre un ordenador y una impresora serie).
- No se usan líneas de control (DTR, RTS, ...).
- Conexión serie más simple (3 hilos):

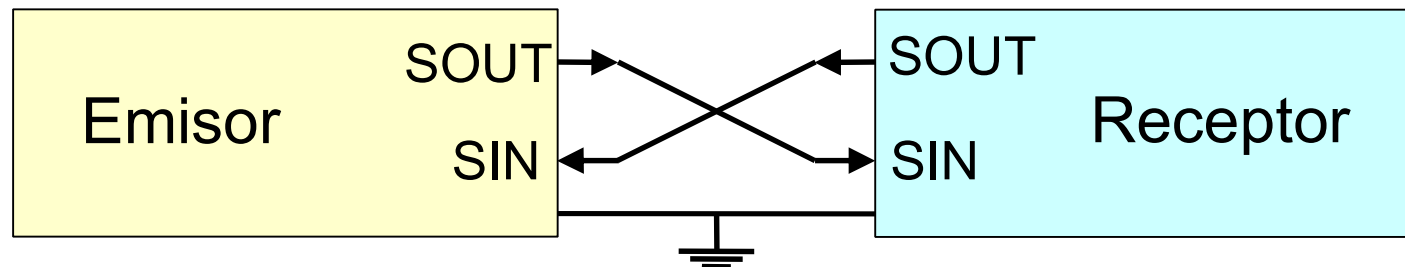


(6)

6.6. Puerto Serie Asíncrono (XXVII)

● Protocolo XON/XOFF

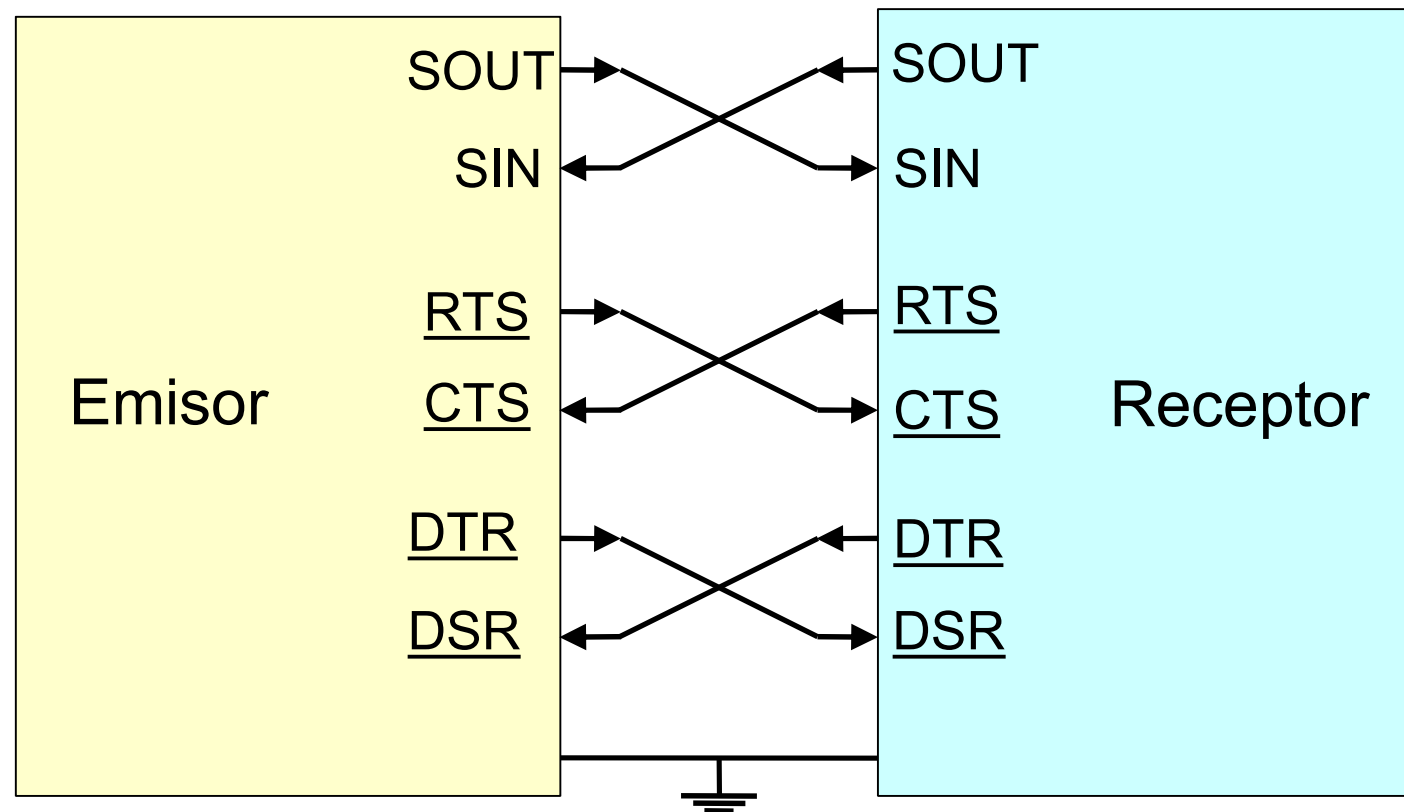
- Emisor envía datos.
 - Si receptor no puede aceptar datos al ritmo impuesto por el emisor, envía al emisor un Control-S (carácter XOFF).
 - Emisor deja de enviar datos.
 - Cuando receptor está listo para recibir datos envía un Control-Q (carácter XON).
 - Emisor reanuda el envío.
- Antes de escribir un dato en el THR, el emisor debe:
- Comprobar que no se ha recibido un Control-S.
 - Comprobar que el THR está vacío.



(6)

6.6. Puerto Serie Asíncrono (XXVIII)

- Protocolo usando líneas de control.
 - La transmisión puede ser Simplex (unidireccional) o Duplex (bidireccional)



Conexión NULL Modem

(6)

6.6. Puerto Serie Asíncrono (XXIX)

Protocolo Simplex

• Emisor

- Pone DTR a 0 indicando que está listo para la comunicación.
- Pone RTS a 0 indicando que desea enviar un byte.
- Comprueba que DSR y CTS estén ambas a 0 (las tiene que activar el receptor).
- Envía el byte.

• Receptor

- Pone DTR a 0 indicando que está listo para la comunicación.
- Pone RTS a 0 indicando que puede recibir un byte.
- Si el receptor desea parar el envío por algún motivo, pondrá RTS a 1. Para reanudarlo de nuevo pondrá RTS a 0.

(6)

6.6. Puerto Serie Asíncrono (XXX)

Protocolo Duplex

- Ambos equipos actúan como emisores / receptores
 - Ambos ponen DTR a 0 indicando que están listos para la comunicación.
 - Ambos ponen RTS a 0 indicando que pueden enviar datos.
 - Ambos deben comprobar que DTS y CTS están a 0.
 - Si cualquiera de los dos desea parar el envío, pondrá RTS a 1. Para reanudarlo de nuevo pondrá RTS a 0.

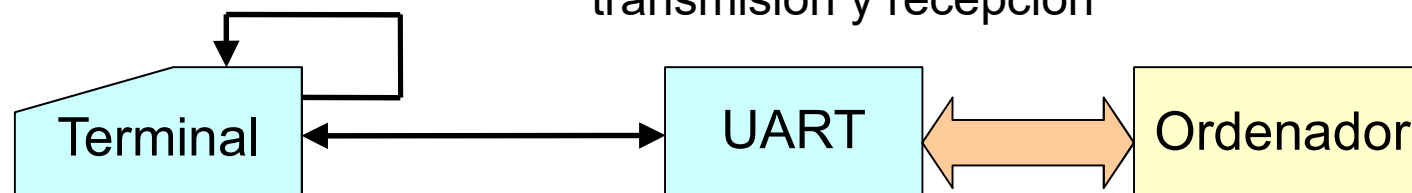
(6)

6.6. Puerto Serie Asíncrono (XXXI)

- Half Duplex (envío y recepción no simultáneos, alternados)

El carácter se visualiza
inmediatamente

En la línea debe alternarse
transmisión y recepción



- Full Duplex (envío y recepción simultáneos)

