TEMA 5: Entrada-salida física

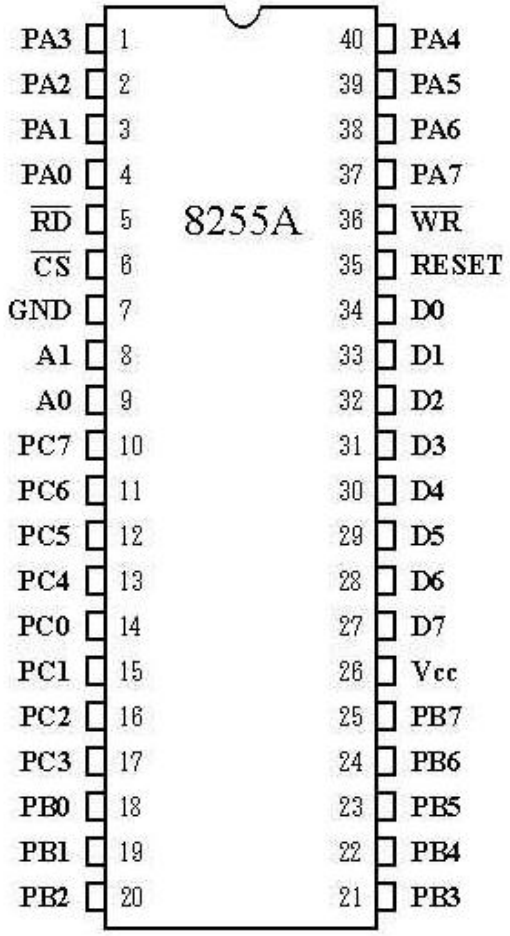
Contenidos

1. Sincronización I. Señales de sincronización
2. Sincronización II. Contadores internos.
3. Sincronización III. Instrucciones de sincronización
4. Interfaces de E/S I. Introducción
5. Interfaces de E/S II. Circuitos básicos
6. Interrupciones. Gestión. PIC y APIC

Objetivo: Métodos y procedimientos de E/S. Buses de comunicación internos y externos.

5. Interfaces de E/S II. Circuitos Básicos

PPI 8255 (Parallel Periferical Interface) Enlace



PA/B/C 0-7: Puertos A, B y C

D0-D7: Bus de datos bidireccional de 3 estados.

RESET: Esta señal borra el registro de control y todos los puertos (A, B y C) son colocados en modo entrada.

RD: Utilizada por la CPU para leer información de estado o datos procedentes del 8255.

WR: Utilizada por la CPU para enviar palabras de control o datos al 8255.

A0-A1: Líneas de dirección: permiten seleccionar uno de los tres puertos o el registro de control.

CS: un nivel bajo en esta entrada habilita la intercomunicación entre el microprocesador y el periférico.

VCC y GND: Alimentación

Modos de operación 8255

MODO 0: Esta configuración implementa simples funciones de entrada/salida para cada bit de los 2 puertos de 8 bits y los 2 puertos de 4 bits.

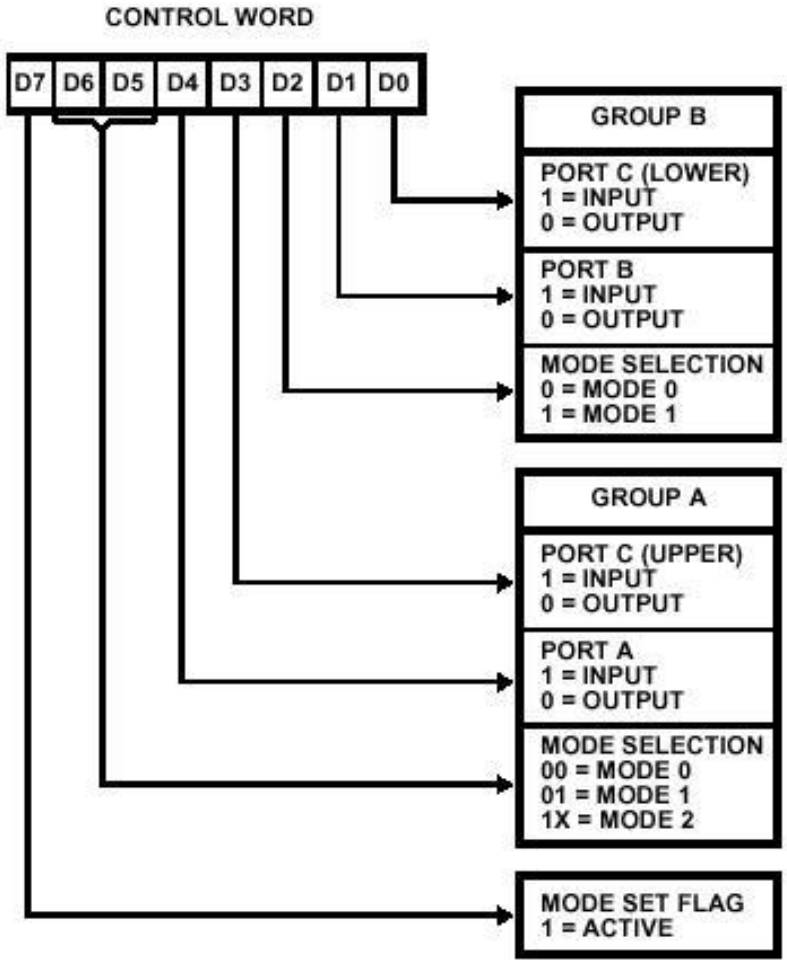
MODO 1: Existen dos grupos (A y B) formados por los puertos A y B más el puerto C, que es repartido a la mitad entre ambos grupos para gestionar las señales de control.

MODO 2: En este modo se constituye un bus bidireccional de 8 bits, por el que los datos pueden ir en un sentido o en otro, siendo el flujo regulado de nuevo por señales de control a través del puerto C. Este modo sólo puede operar en el Grupo A. Tanto las entradas como salidas son almacenadas en latch.

Tabla de selección de registros

A1	A0	SELECCION
0	0	PORT A
0	1	PORT B
1	0	PORT C
1	1	CONTROL

Registro de control:



El modo de funcionamiento de cada una de las puertas queda programado por software. La palabra de control que escribe el microprocesador sobre el periférico contiene la información del modo, bit activado, etc, y todo ello sirve para inicializar a dicho periférico.

Cada uno de los bloques de control acepta comandos de programación para las puertas asociadas a él.

- Control grupo A: puerta A y puerta C alta (C7 a C4).
- Control grupo B: puerta B y puerta c baja (C3 a C0).

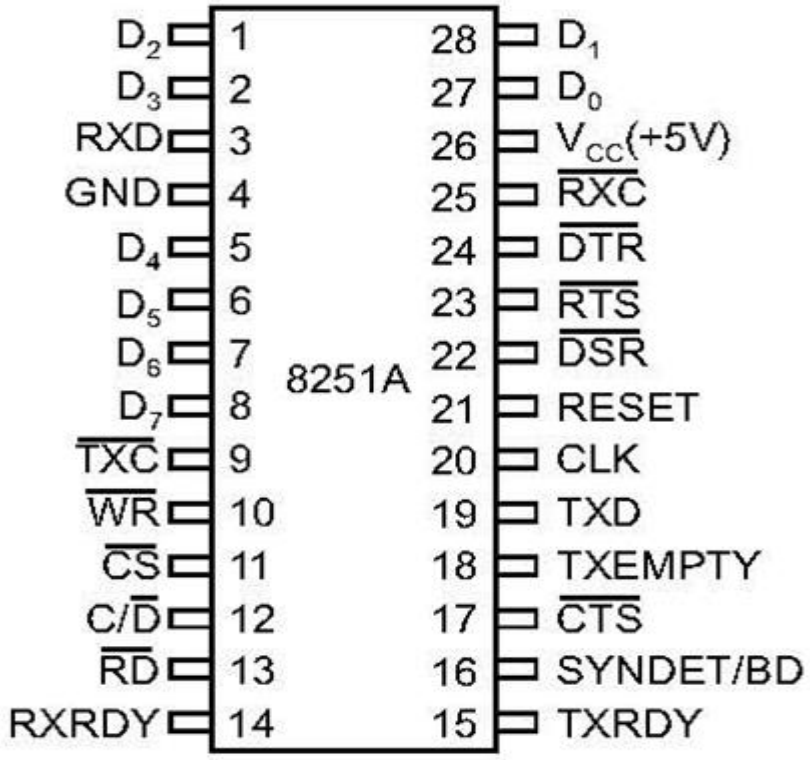
En el registro de Control de palabra solamente se puede escribir y no se permiten operaciones de lectura.

Simulador online de 8255:

http://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/50-rtlib/60-pio8255/pio.html

Necesita java y configurarlo con “paciencia”

USART: Universal Synchronous/Asynchronous Receiver Transmitter (Intel 8251) Enlace (¡45 pag!)



Descripción muy muy resumida!!

D7 a D0: bus bidireccional que conecta con la CPU

RESET: Un nivel alto produce la reinicialización del 8251.

CLK: Entrada en un reloj externo. Es independiente de RXC o TXC.

WR: Entrada de control que se utiliza para recibir datos desde la CPU

RD: Entrada activa a nivel bajo que permite leer datos y palabras de control, desde el 8251

CS: Entrada activa a nivel bajo que active o desactiva al circuito.

C/D: Entrada que recibe una señar de selección de datos o palabras de control o palabras de estado, cuando la CPU accede al 8251.

TXD: Se trata de un terminal de salida para la transmisión de datos desde el que se envían los datos de serie convertidos.

TXRDY: Este es un terminal de salida que indica que el 8251 está listo para aceptar un carácter transmitido.

TXEMPTY: Se trata de un terminal de salida que indica que el 8251 ha transmitido a todos los caracteres.

RXD terminal de recepción de datos

TXC: Esta es una señal de entrada de reloj que determina la velocidad de transferencia de datos transmitidos.

RXRDY: Se trata de un terminal de salida que indica que el 8251 contiene un carácter que está listo para leer.

RXC: Entrada de reloj que determina la frecuencia de recepción de datos

SYNDET/BD: Es bidireccional, cuya función cambia de acuerdo con el modo.

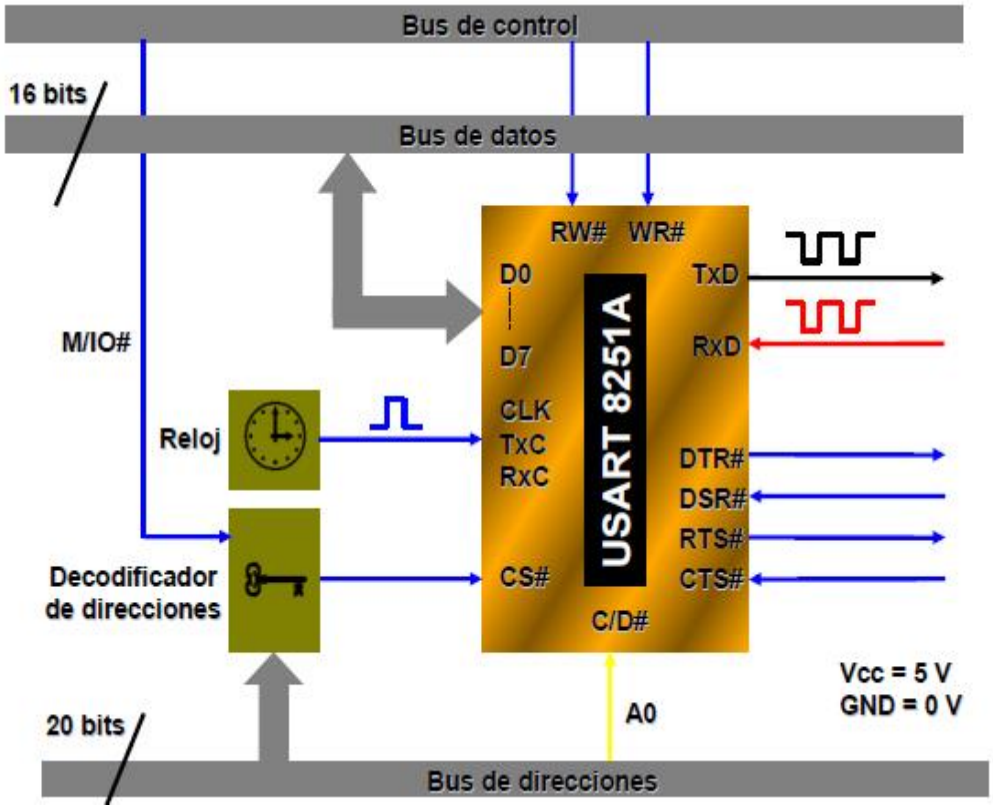
DSR: En una señal de entrada para la interface de módem. Data Send Ready

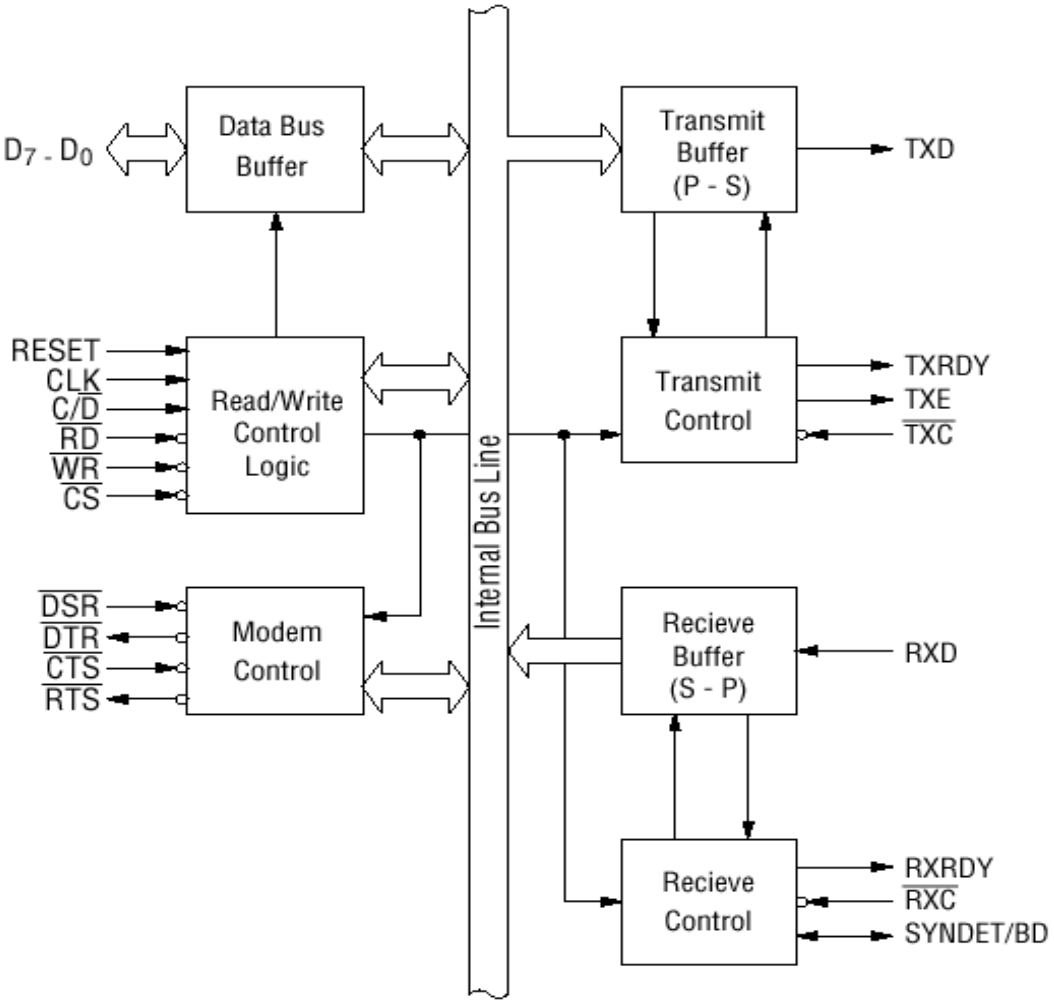
DTR: En una señal de salida para la interface de módem. Data to Transmit Ready

CTS: Terminal de entrada para la interface de módem. Se utiliza para controlar el circuito de transmisión. Actúa si el dispositivo está configurado en "TX Habilitar".

RTS: En una señal de salida para la interface de módem. Ready To Send

Conexión al sistema

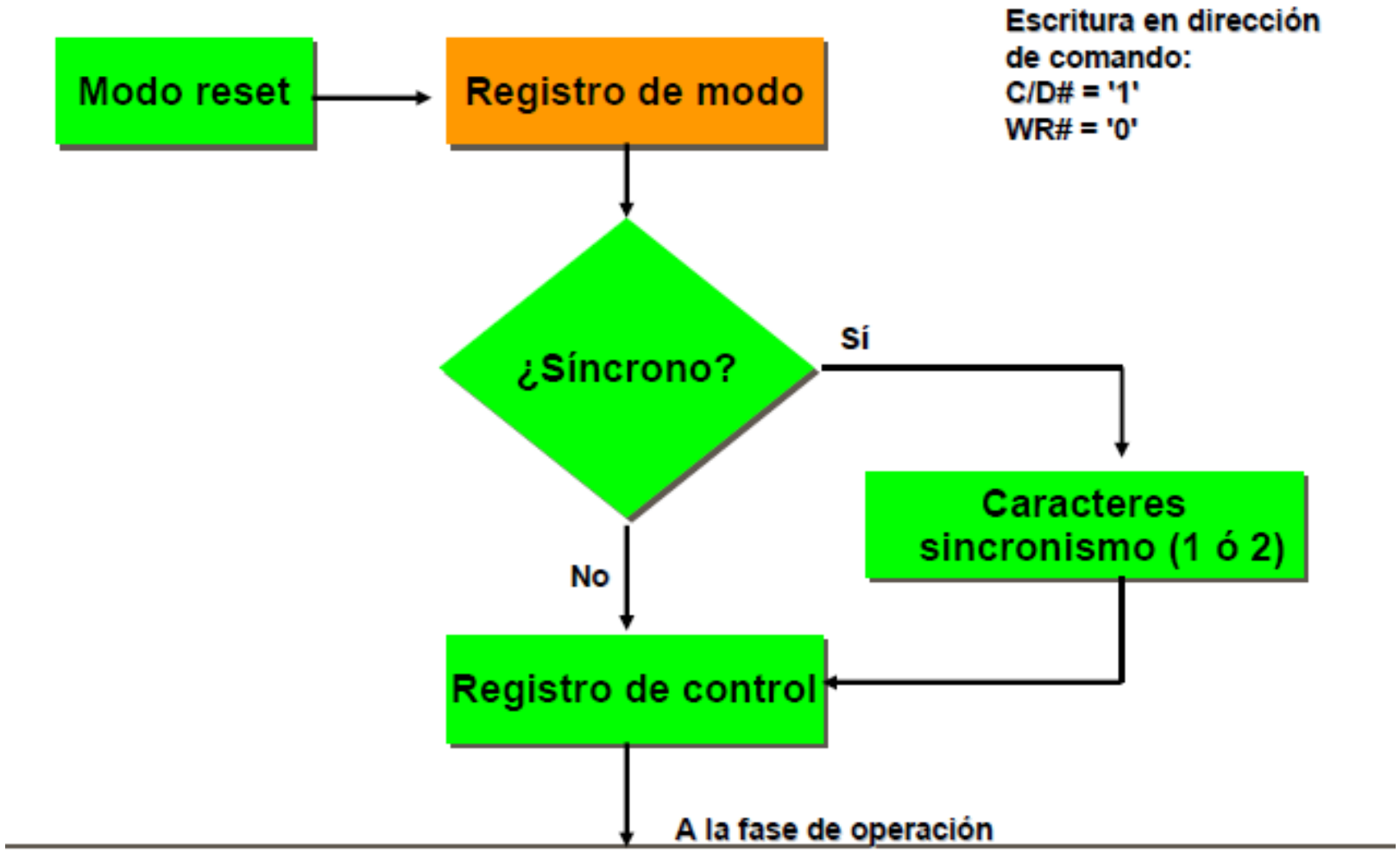




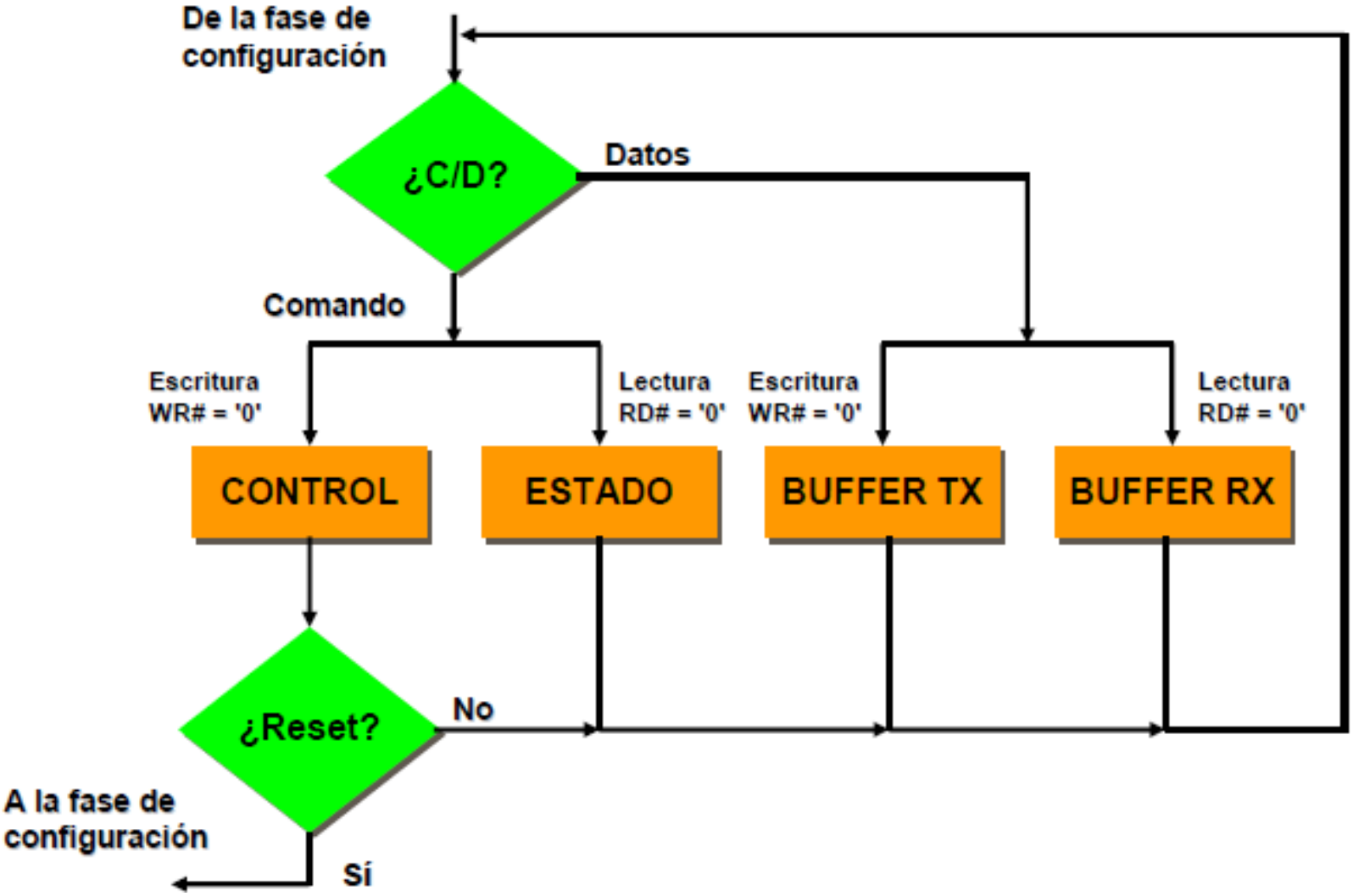
Registros internos: Siete registros accesibles por la CPU a través de dos direcciones de puertos. Secuencia de programación. Patilla C/D (Command / Data) conectada a la línea A0 del bus de direcciones para diferenciar entre las dos direcciones.

Registro	$C/\overline{D}$	Tipo
Modo	1	Escritura
Control	1	Escritura
Estado	1	Lectura
1er carácter sincronismo	1	Escritura
2º carácter sincronismo	1	Escritura
Registro de transmisión	0	Escritura
Registro de recepción	0	Lectura

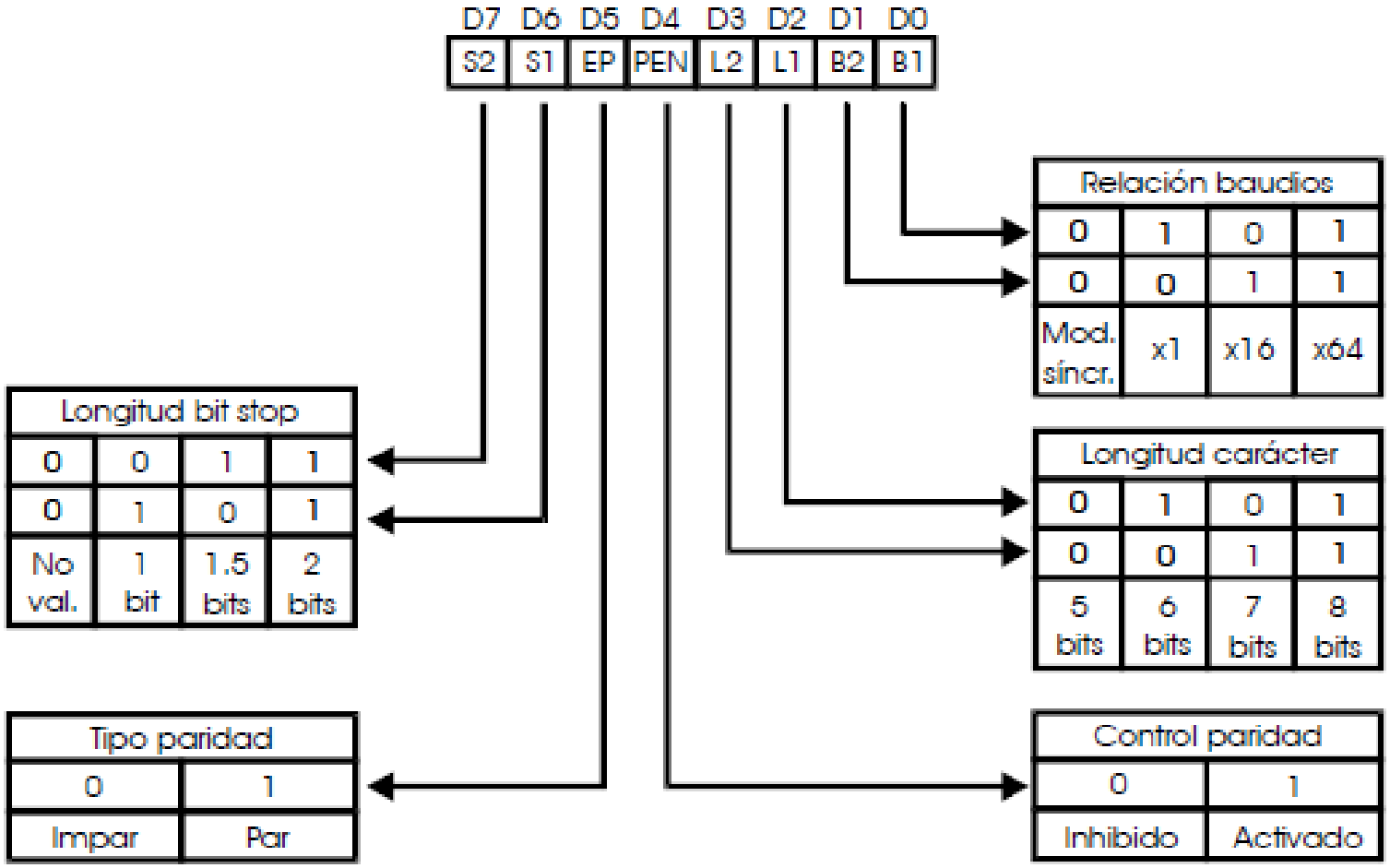
Fase de configuración



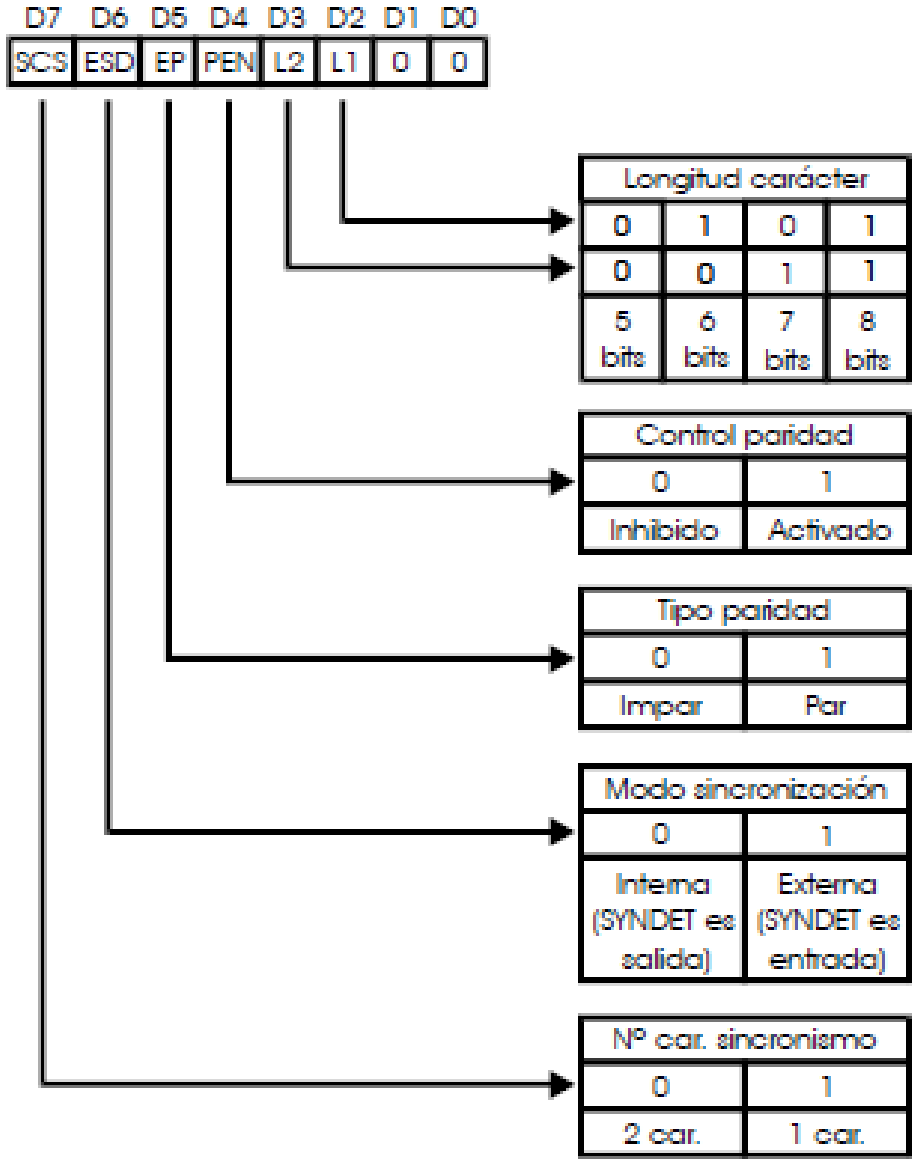
Fase de operación



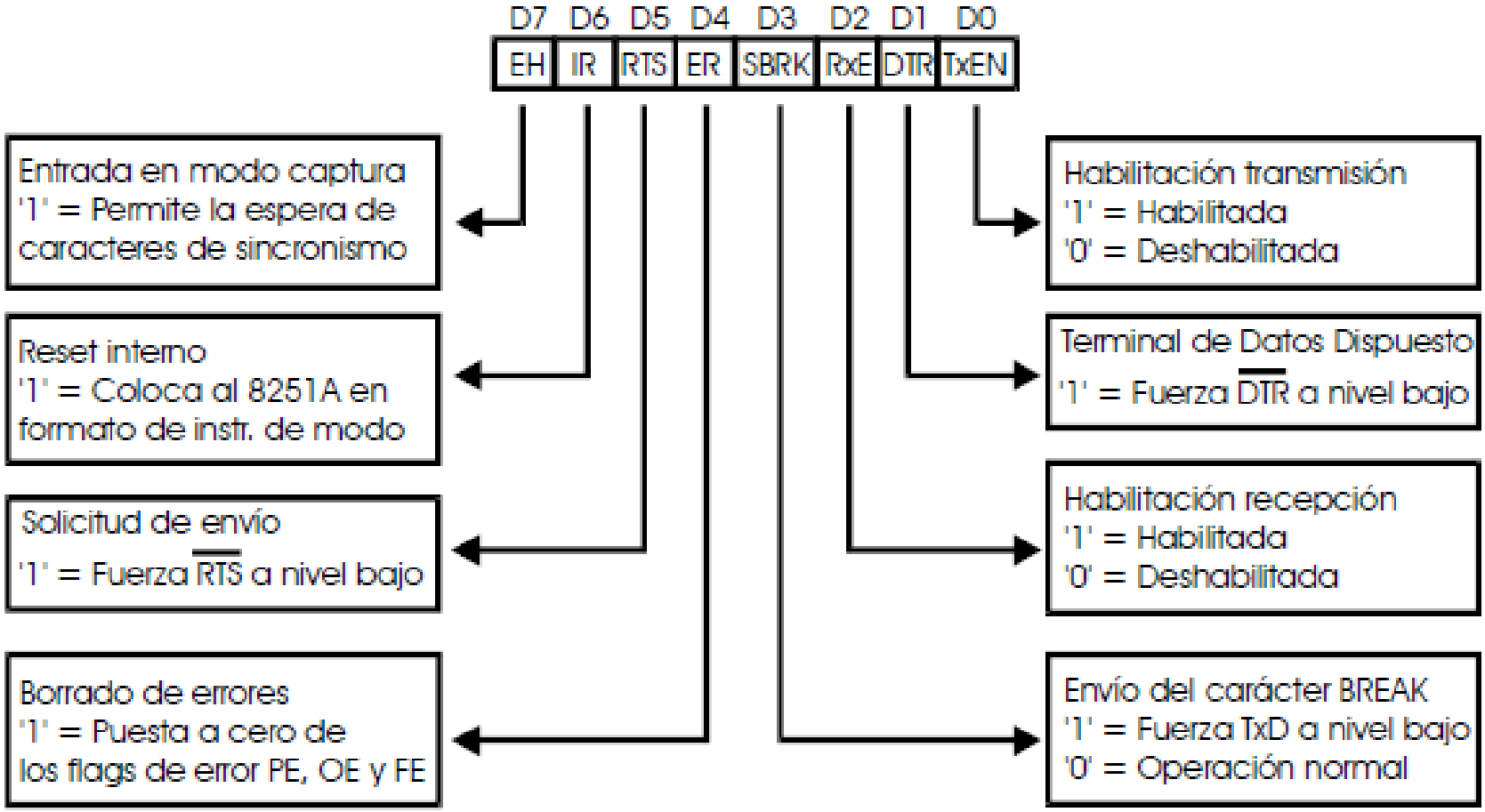
Modo asíncrono (Registro de modo)



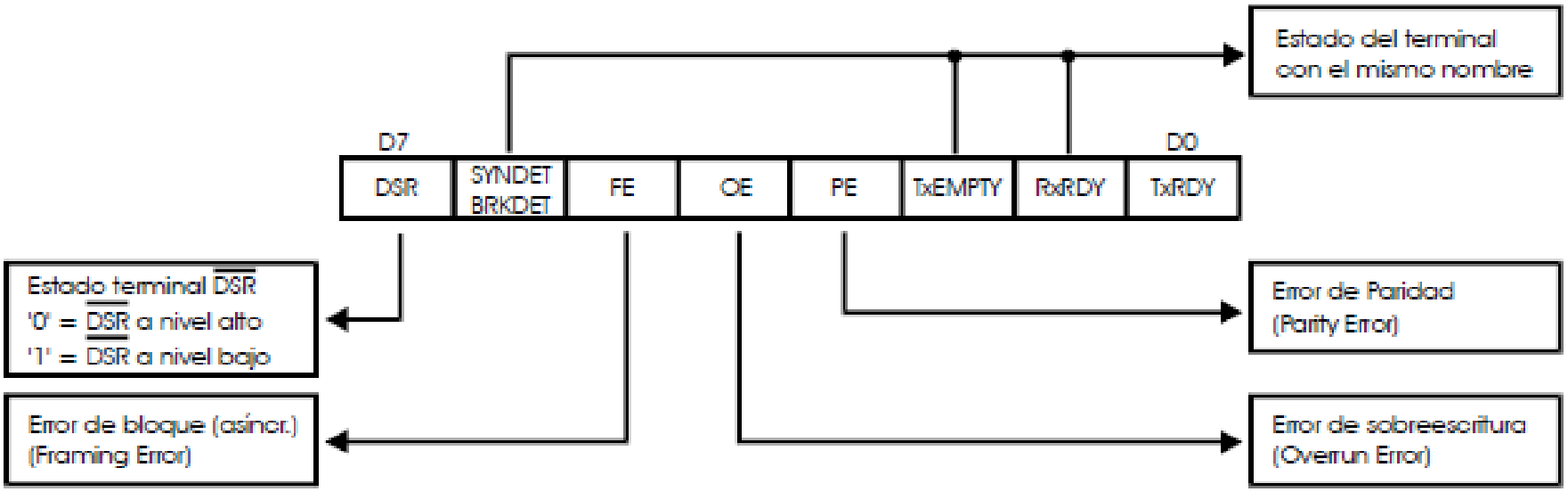
Modo síncrono (Registro de modo)



Formato del registro de control



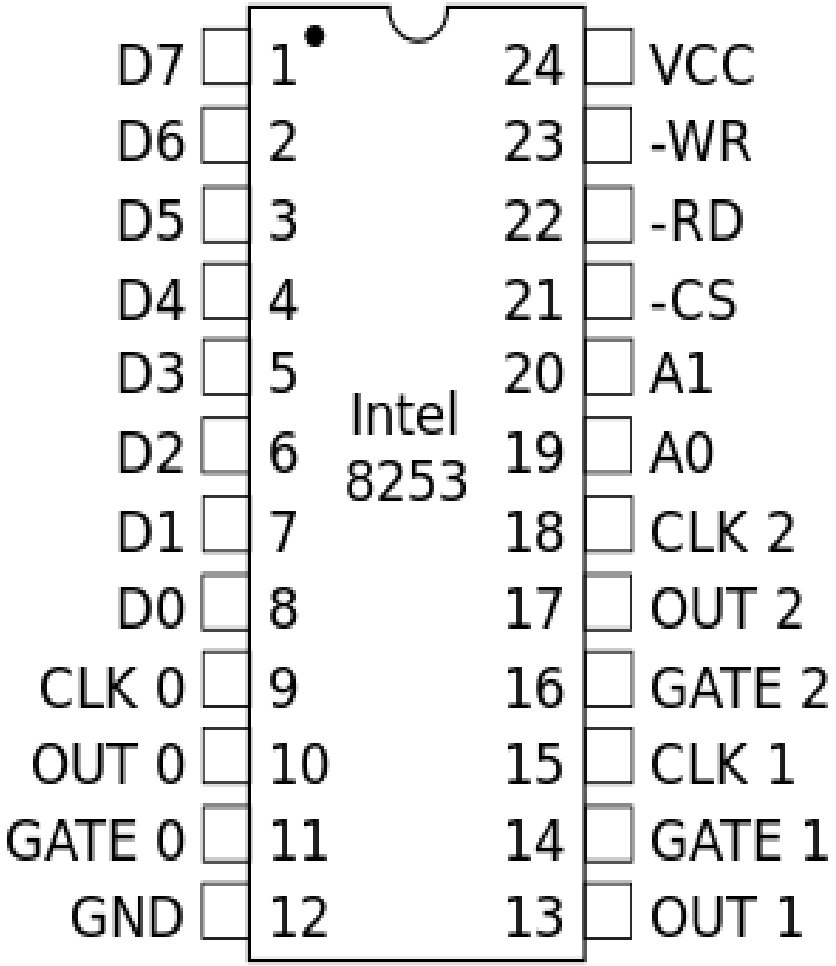
Formato del registro de estado

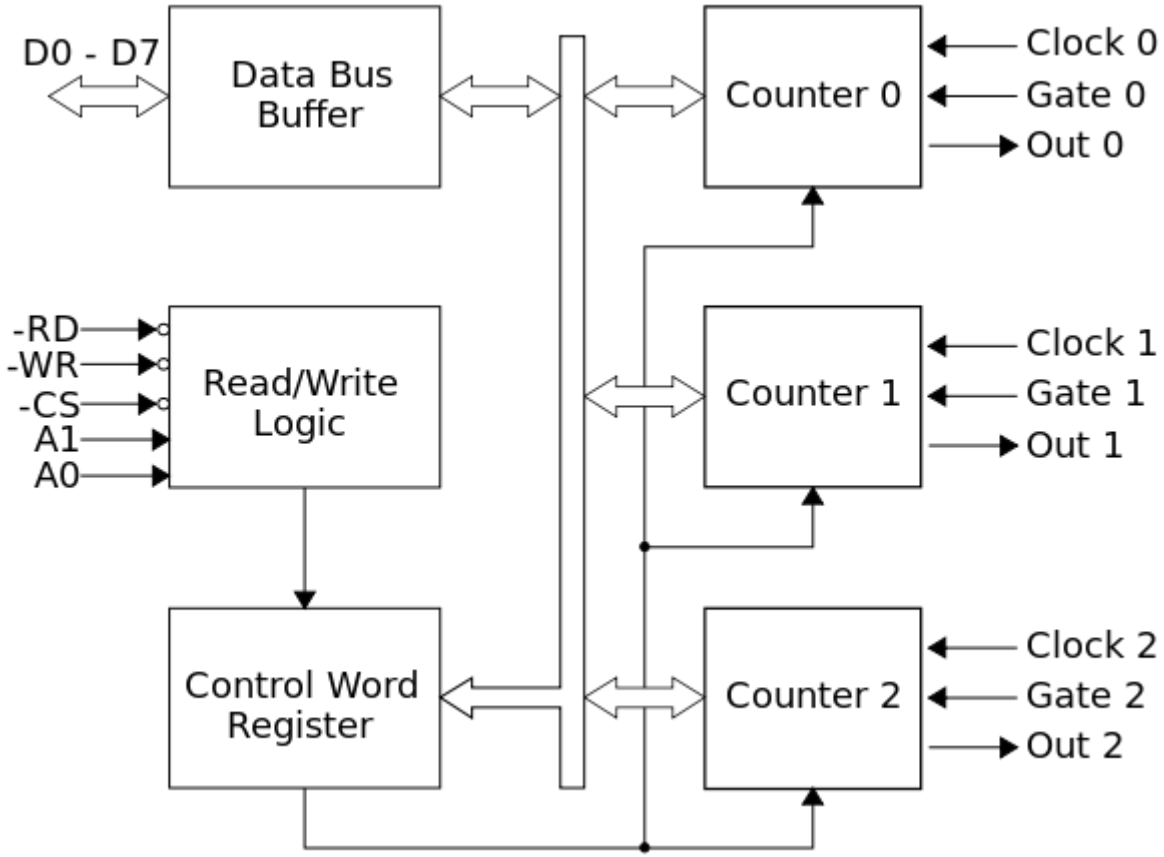


¿Cómo funciona el 8251? Simulador ->

http://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/50-rtlib/65-usart8251/usart.html

PPI 8253/4 (Programmable Interval Timer)





Tareas a realizar: AT

- Pensemos en tres interfaces de E/S para incluir en el proyecto final de la asignatura: paralelo, serie y timer:

  - ¿Cómo limitaríamos el diseño para que sea viable de realizar en Logisim? Indicar 2-3 limites en cada uno de ellos
  - ¿Qué características podemos implementar en cada uno de ellos? Citar 3 o 4 características para cada uno de ellos. Funcionalidad y sencillez como objetivo.
  - Proponer un diagrama de bloques, para cada uno de ellos, que haga posible abordar las características enunciadas
  - ¿Algún otro interface que proponer?

6. Interrupciones

- E/S utilizando la técnica de "polling" es ineficiente, ya que la CPU pasa demasiado tiempo de espera.
- E/S por medio DMA, no es capaz de sincronizar de forma eficiente el sistema

La solución, gestión a **nivel de evento (concurrencia)**

- Se necesita de un protocolo de handshaking para funcionar correctamente: señales INTRQ, INTACK
- La CPU atenderá la solicitud según **una secuencia predeterminada y cuando le sea posible**.
- La pregunta más importante es ¿Dónde está Wally? ¿Dónde está la rutina de gestión de cada interrupción?

- Se llama vector de interrupción a la dirección de memoria donde está localizada la rutina de gestión correspondiente. Están contenidos en una tabla. Ej:

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19–31	(Intel reserved, do not use)
32–255	maskable interrupts

- El dispositivo puede indicar de varias formas el vector de interrupción asociado a la interrupción que solicita: tabla fija, índice o vector completo. MMU nos dirá al dirección física

- Las interrupciones pueden coincidir en el tiempo, por lo que es necesario establecer una **prioridad** en su gestión

- Existen interrupciones deshabilitables (enmascarables) o imposibles de deshabilitar (No enmascarables)

- La aceptación de las interrupciones se realiza dentro del ciclo de ejecución de las instrucciones. En concreto al final de la ejecución de la instrucción en curso.

- Son controladas por el Chipset y requieren un control específico del DMA.

Tipos de interrupciones/excepciones:

- Asíncronas al programa (interrupciones)

  Fuentes externas a la CPU. Las producen los periféricos o dispositivos externos para solicitar que la CPU los atienda. No están relacionadas directamente con la ejecución.

- Síncronas al programa(excepciones)

  - Fallos o errores. Hay que gestionarlos ¿Qué hacer?
  - Traps. marcas de depuración/contención predefinidas.
  - Excepciones programadas: Requerimientos para ejecución de rutinas de S. O. (software int / syscalls)

Tablas de vectores de interrupción

- Son las encargadas de "traducir" el requerimiento de un dispositivo a una dirección física (Chipset, DMA, etc). En esa dirección física está la rutina de gestión
- Cada tabla debe ser inicializada para poder empezar a aceptar interrupciones
- Se controla por medio del registro llamado IDTR
  - Dirección de inicio de la rutina
  - Límite de espacio de intercambio
- Hoy en día nos encontramos con jerarquías de tablas.

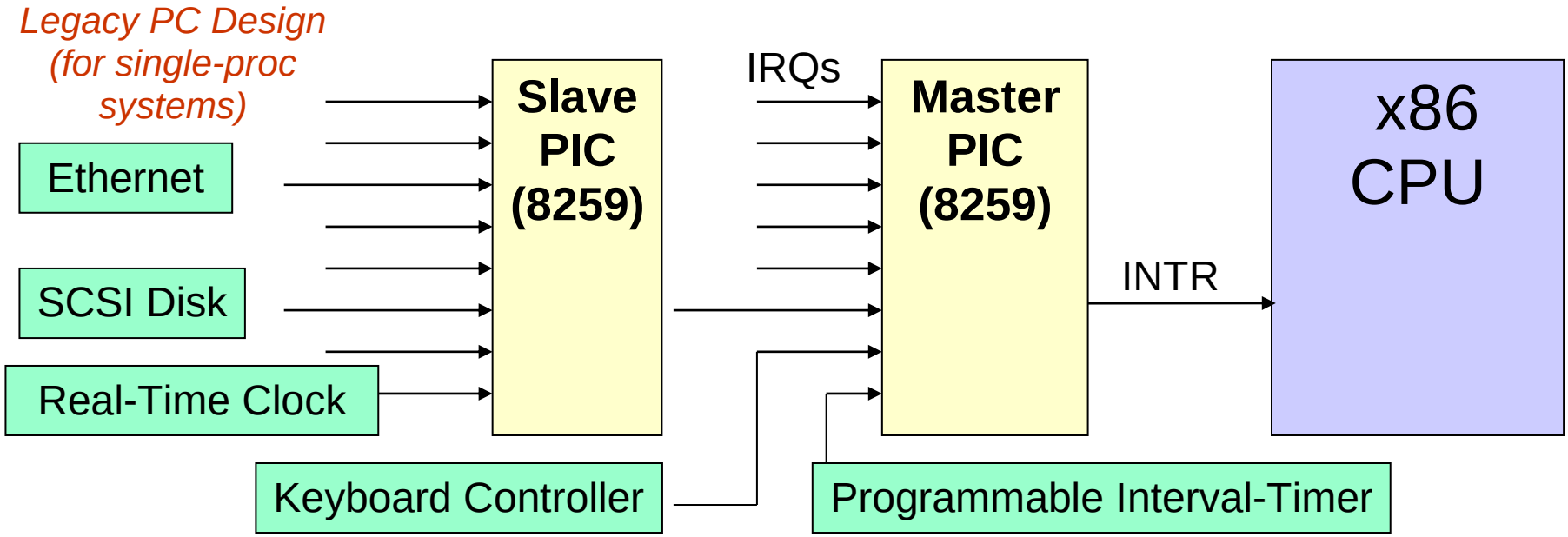
Secuencia de gestión de una interrupción (Una vez aceptada. Intel)

- Se determina el número de interrupción (identificar dispositivo)
- Se determina el tipo interrupción (prioridad, acceso, etc.)
- Se obtiene la rutina de gestión de interrupción (vector)
- Se determina si la rutina está disponible
- Se determina si los privilegios son correctos
- Se salvan los registros a utilizar por la rutina de gestión
- Se almacena un código de error en la pila, por si es necesario gestionar el mal funcionamiento de la rutina
- Se cargan lo valores adecuados en los registros
- Se salta a la rutina de gestión
- Se ejecuta la rutina
- Tras terminar su empleo se borra el código de error de la pila, se recuperan los valores previos de los registros y se vuelve al flujo normal del programa. Esto se realiza por medio de una instrucción (IRET)

Líneas de INT y PICs

- Líneas de INT, son los terminales que se utilizan para que los dispositivos soliciten interrupción (INT RQ)

- PIC (Programable Interruption Control) Chips procesadores de interrupciones

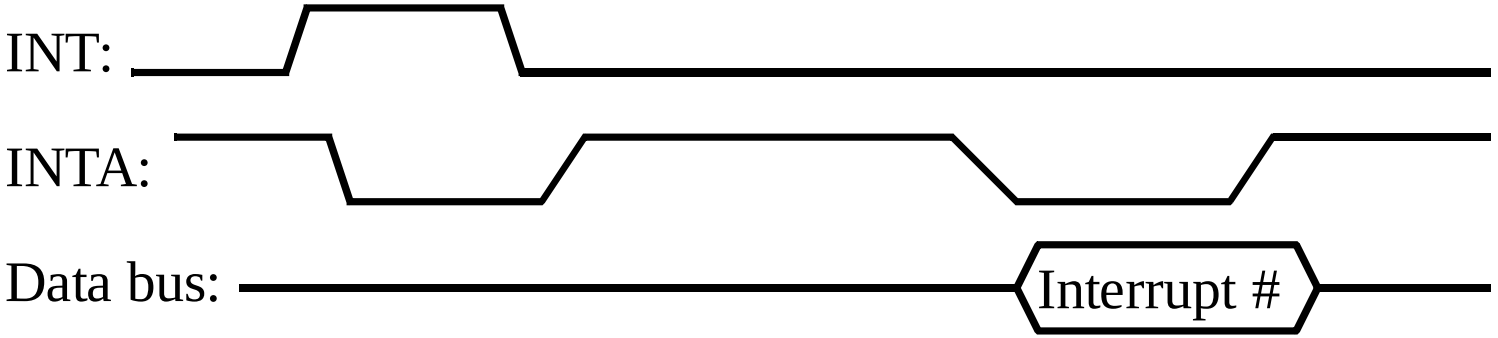


Controlador de interrupciones (PIC)

- Gestor dispositivos y de vectores de interrupción
- Almacena el vector en un registro interno
- Gestiona el Handshaking (RQ – ACK)
- Gestiona la prioridad de los dispositivos
- Gestiona la habilitación/deshabilitación de interrupciones
- El chip PIC "más estándar" es el 8259A

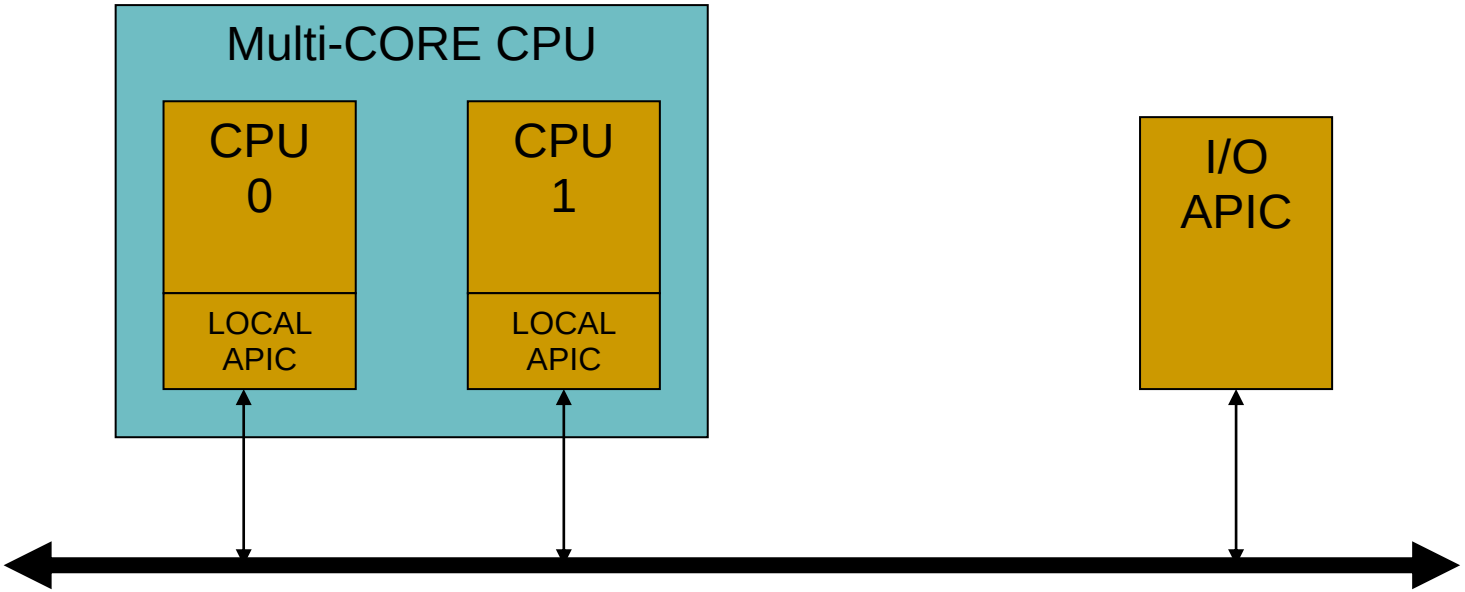
Ejemplo Intel 386

- Tiene una entrada de interrupción (INT) y una de aceptación (INTA).
- Recibe el vector de interrupción por medio del bus de datos del sistema



¿Qué sucede en CPUs con varios núcleos?

- Hay que encontrar formas de coordinar el uso de interrupciones.
- Diversidad de soluciones. Ej: En cada núcleo existe un PIC Local



Procesadores avanzados de interrupción: (L) APIC, IO-APIC

- Advanced PIC (APIC) para sistemas multiprocesadores simétricos
  - Es lo más utilizado hoy en día
  - Las interrupciones se dirigen a las CPUs a través del sistema de buses internos
  - Se dispone de **IPI**s: inter-processor interrupt
- Local APIC (LAPIC) y IO-APIC
  - Los dispositivos se conectan al IO-APIC (externo)
  - IO-APIC se comunica, a través del bus, con el APIC (Local)

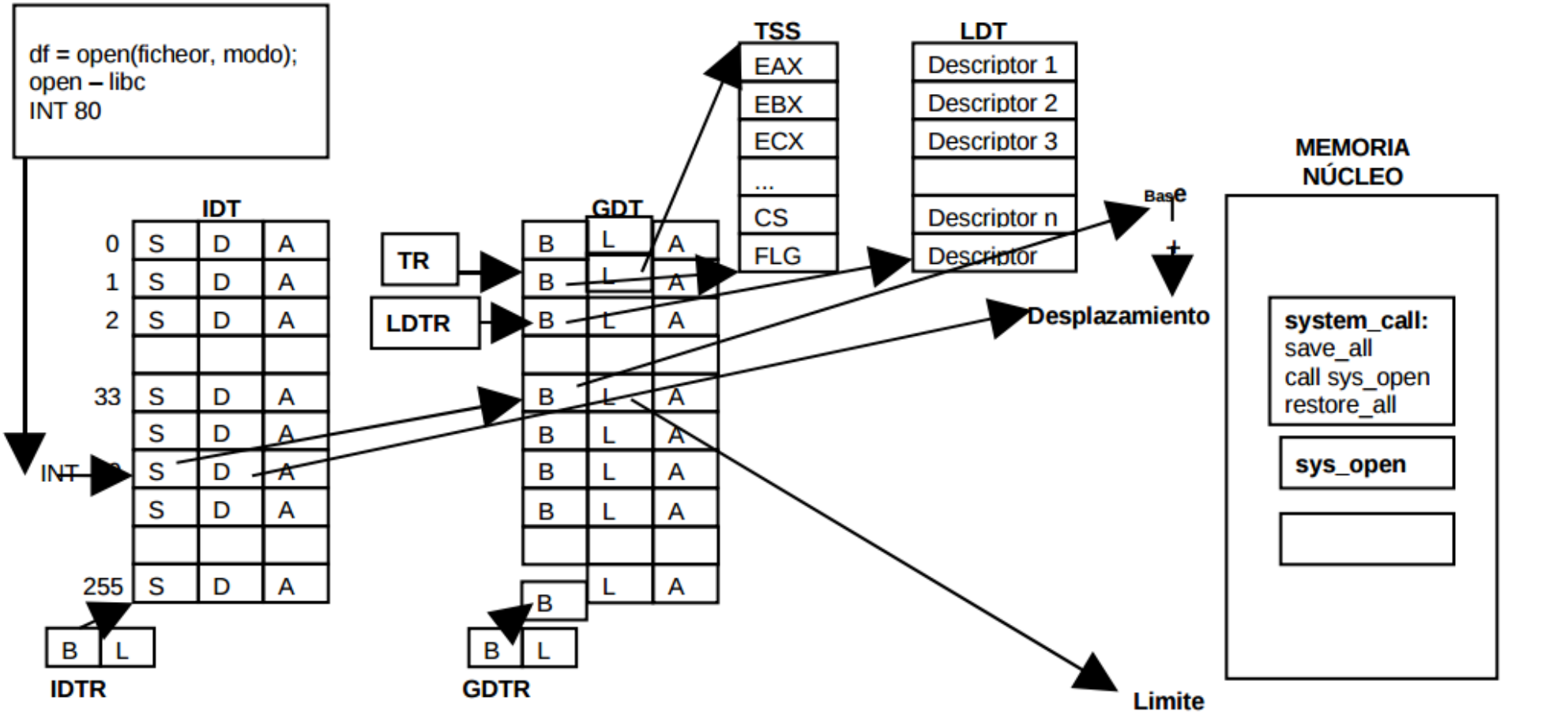
- Ruteo de las interrupciones
  - El sistema debe comunicar o seleccionar la ruta que deben seguir las interrupciones en función de la tarea que está ejecutando cada núcleo.
  - Debe ser capaz de distribuir la tareas de interrupción a los procesadores locales
  - Debe ser capaz de gestionar la prioridad de las interrupciones
  - Utiliza un registro especial: Task Priority Register (TPR) para arbitrar el acceso a los procesadores locales, cuando existan problemas o conflictos de prioridad o carga de interrupciones

Asignando IRQs a los dispositivos

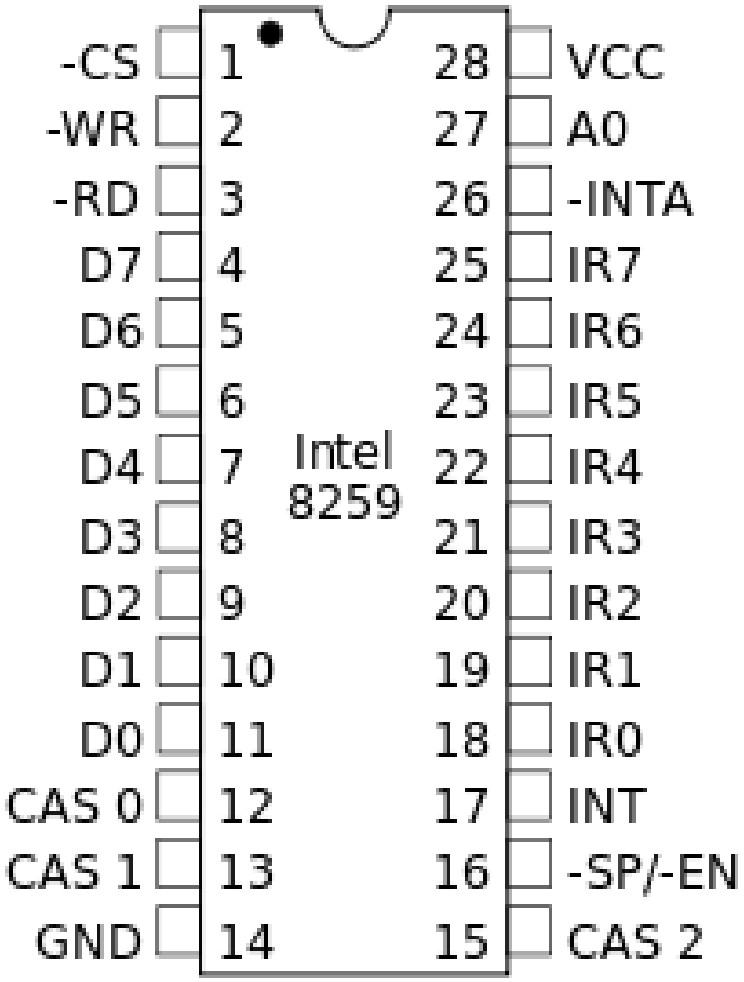
- La asignación de IRQ es dependiente del hardware
- La asignación se puede programar de forma lógica o física.
- Ej: El Bus PCI asigna las IRQs desde el arranque del sistema
- Algunas IRQs están prefijadas por el sistema
  - IRQ0: Interval timer. IRQ2: Cascade pin for 8259A
- La asignación dinámica de IRQs es especialmente útil para los drivers que se cargan en caliente (buses PnP)
- La asignación de vectores a las IRQs. Ejemplo:
  - Vectores, índice (0-255) dentro de la tabla de vectores
  - Vectores se asignan a las IRQs superiores a 32
  - Por debajo de estos 32, se reservan para NMI y excepciones
  - Las interrupciones enmascarables se asignan libremente
  - El Vector 128 -> llamadas al sistema (syscall)
  - Los Vectores 251-255 se utilizan para el IPI

Tabla de interrupciones (IDT)

- La dirección física de la rutina de gestión de la interrupción está localizada en el tabla de descriptores de interrupción (IDT)
- IDT: "descriptores"
  - Segment selector + offset de la rutina
  - Descriptor Privilege Level (DPL)
  - Gates
    - Task gate: incluye TSS (pila) para transferencias
    - Interrupt gate: deshabilita las interrupción
    - Trap gate: para interrupciones anidadas
- GDT (Tabla Global de Descriptores)
  - Esta tabla es única para el sistema y contiene información de los descriptores de segmentos del sistema.


df = open(ficheor, modo);
open – libc
INT 80
IDT
0 S D A
1 S D A
2 S D A
33 S D A
S D A
INT S D A
S D A
255 S D A
B L
IDTR
TR
LDTR
GDT
B L A
B
B L
GDTR
TSS
EAX
EBX
ECX
...
CS
FLG
LDT
Descriptor 1
Descriptor 2
Descriptor 3
Descriptor n
Descriptor
Base
+
Desplazamiento
Limite
MEMORIA NÚCLEO
system_call:
save_all
call sys_open
restore_all
sys_open
S – Selector
D – Desplazamiento
A – Atributos
B – Base
L - Limite

Controlador de Interrupciones 8259: Enlace (24 pág.)

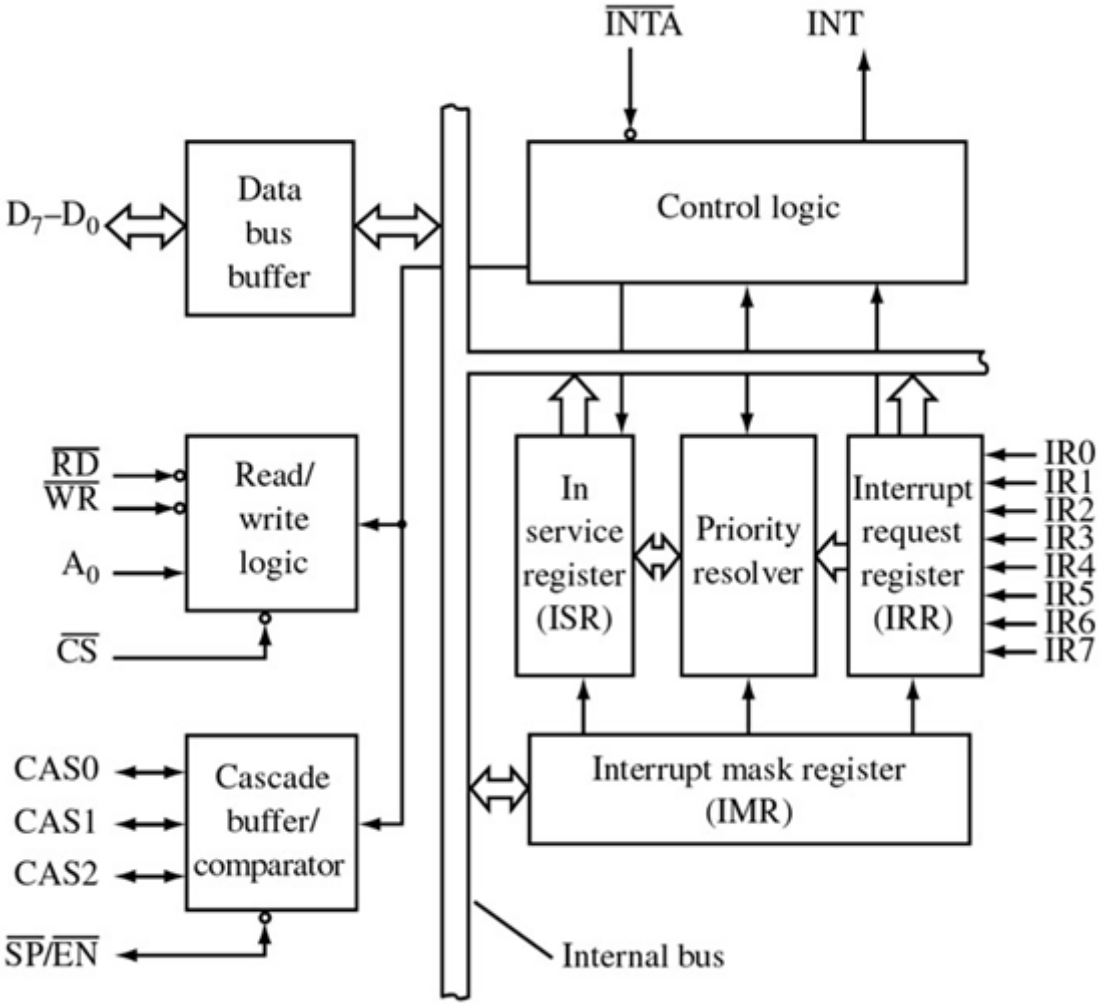


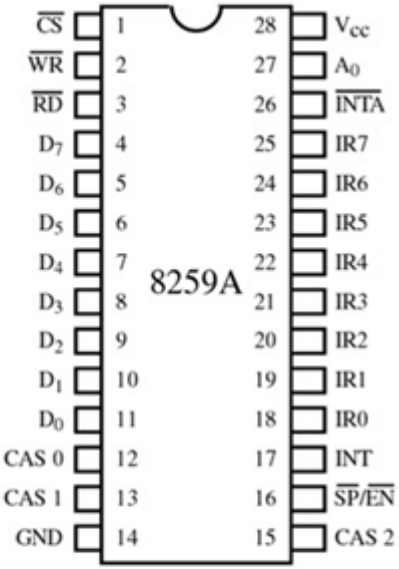
ESTRUCTURA (conexión)

- IRQ0 – IRQ7: peticiones de interrupción de los periféricos, de mayor a menor prioridad.
- INT: petición de interrupción al procesador gestionada por el i8259.
- INTA (Int ACK): reconocimiento / aceptación de la interrupción por parte del procesador.
- CS (chip select): para leer y escribir en los registros del PIC -> utilizada para programar PIC.

- CAS2 – CAS0: líneas para la conexión en cascada de varios i8259. Actúan como salida del PIC maestro y como entrada de los PIC esclavos.
- EN: indica si el PIC actúa como maestro o como esclavo cuando hay varios encadenados (en cascada).
- RD, RW (read, write): permiten leer o escribir en los registros de control del i8259.
- A0: única línea del bus direcciones usada para seleccionar los registros de control.
- Bus de datos: intercambio de datos entre el PIC y el resto de componentes de un computador (memoria y procesador).

Estructura interna





Secuencia de Reconocimiento de una Interrupción

1. Una o más líneas IRQ son activadas por los periféricos conectados al PIC y esto activa los correspondientes bits del IRR.
2. El 8259 evalúa la prioridad de estas interrupciones (mediante el codificador de prioridad) y solicita una interrupción a la CPU activando la línea INT.
3. Cuando la CPU reconoce la interrupción envía la señal INTA.
4. Entonces el PIC, recibida la señal INTA, activa el bit correspondiente a la interrupción de mayor prioridad (la que va a ser procesada) en el ISR y borra ese mismo bit en el IRR. En este ciclo, el 8259 aún no controla el bus de datos.

5. Cuando la CPU envía un segundo ciclo INTA, el 8259 deposita en el bus de datos un valor de 8 bits que indica el número de vector de interrupción (Ejemplo: offset o base de interrupciones del PIC + nº de IRQ). La CPU multiplica este valor por 4 para buscar en esa dirección de memoria la dirección de comienzo de la RTI.(Rutina de gestión de interrupción)

6. En el modo AEOI (Advances End Of Interrupt) del PIC, el bit de la interrupción en el registro ISR es borrado automáticamente nada más acabar el segundo pulso INTA. En caso contrario, este bit permanece activo hasta que la CPU envíe el mandato EOI (End of Interrupt) al final de la rutina que trata la interrupción (esto es lo más normal).

Rotación de prioridades:

- Hay sistemas en que varios periféricos tienen el mismo nivel de prioridad, en los que no interesa mantener un orden de prioridades en las líneas IR.
- La solución consiste en asignar el menor nivel de prioridad a la interrupción recién atendida para permitir que las demás pendientes se procesen también.
- Para ello se envía un EOI que rote las prioridades: si, por ejemplo, se había procesado una IRQ3, IRQ3 pasará al menor nivel de prioridad e IRQ4 al mayor.
- Existe también una asignación específica de prioridades, a través de OCW2.

APIC (Advanced PIC: 82093) Gestión avanzada de int.

- No funciona si en el sistema no hay al menos un IO APIC
  - Tiene que configurarse en la BIOS.
  - Funciona conjuntamente con el/los IO APIC superando las limitaciones del esquema tradicional con dos 8259 en cascada.

- Cada IO APIC permite conectar entre 24 y 61 dispositivos.

- Además, la conexión con los Local APIC de todos los procesadores del sistema permite que cuando hay Hyperthreading o se trata de un multiprocesador, la interrupción llegue al procesador adecuado.

Tareas a realizar: AT

No vamos a incluir interrupciones en el bus que estamos diseñando, pero pensemos en la posibilidad de incluirlas.

- ¿Cómo podemos aplicar las interrupciones al bus que estamos diseñando?
- ¿Qué elemento de bus tendría que gestionar las interrupciones?
- Pensar en dos opciones de gestión de interrupciones: síncronas y asíncronas. ¿En qué cambiaría la gestión?
- ¿Cómo se podrían incluir prioridades en las peticiones de INT?
- ¿Cómo se gestionarían las INT en estructura jerárquica de buses?