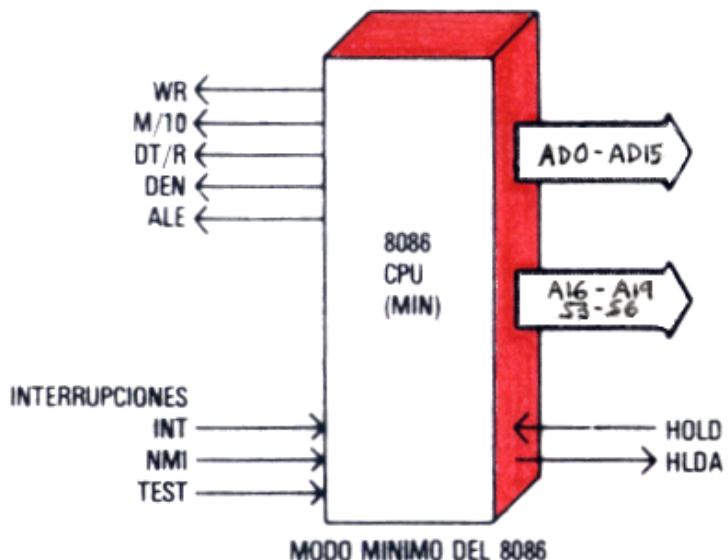


# Señales del 8086

## Dos modos de funcionamiento:

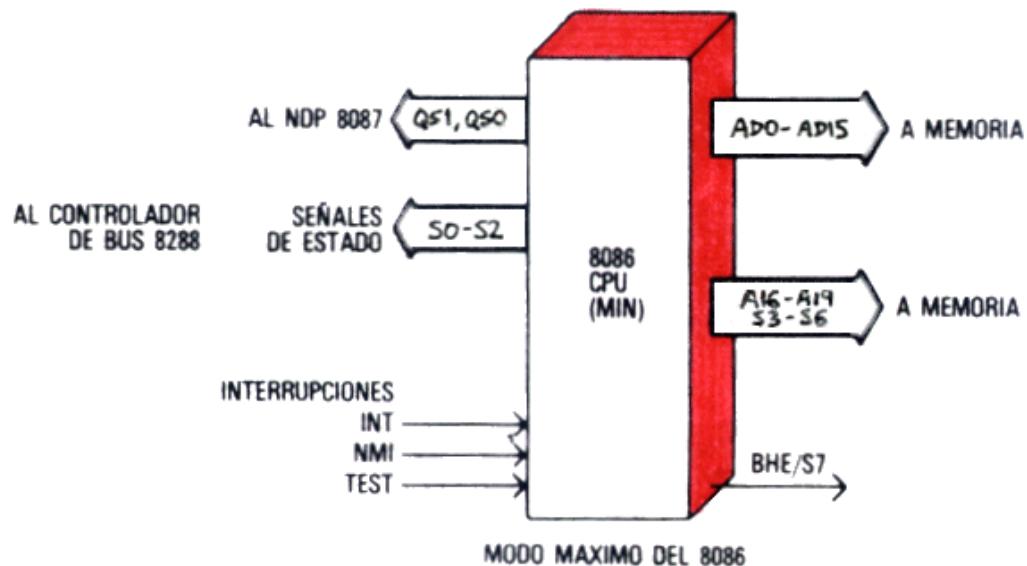
- MODO MÍNIMO:

- Para pequeños sistemas.
- El 8086 genera las señales de control por sí mismo, trabajando de forma más autónoma.



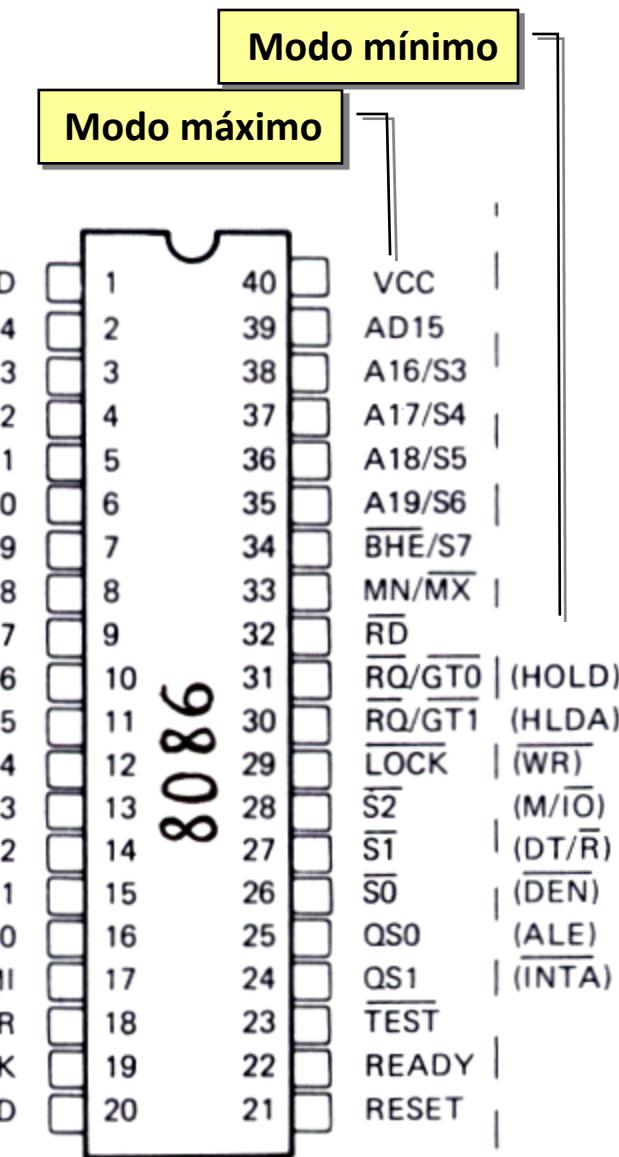
- MODO MÁXIMO:

- Para trabajar en colaboración con el procesador numérico 8087 y el procesador de E/S 8089.
- El 8086 depende del controlador de bus 8288 para generar el conjunto completo de señales de control del bus.



# Señales del 8086

- La elección del modo mínimo o máximo se hace a través de la patilla MN/MX#, uniéndola respectivamente a +5V o a masa.
- Algunas señales del 8086 coinciden en los dos modos, y otras son específicas de un modo.



# Señales del 8086

## ■ Señales comunes a los dos modos del 8086.

### ■ **VCC, GND:**

- Alimentación (+5V).
- Tierra.

### ■ **AD0 - AD15:**

- Bus de direcciones (Address) / datos (Data) multiplexado en el tiempo.
- En el 8088 sólo hay 8 señales de datos AD0-AD7.

### ■ **A16 / S3, A17 / S4, A18 / S5, A19 / S6:**

- 4 líneas más significativas del bus de direcciones, multiplexadas en el tiempo con 4 señales de estado.
- S3 y S4 indican qué registro de segmento dirección de este ciclo del bus.
- S5 es una copia del bit de estado IF.
- S6 está a nivel 0.

Multiplexación en el tiempo:

En cierto momento (estado T1) los terminales llevan información sobre la dirección, y posteriormente (estados T2, T3, TW y T4) llevan datos e información sobre el estado.

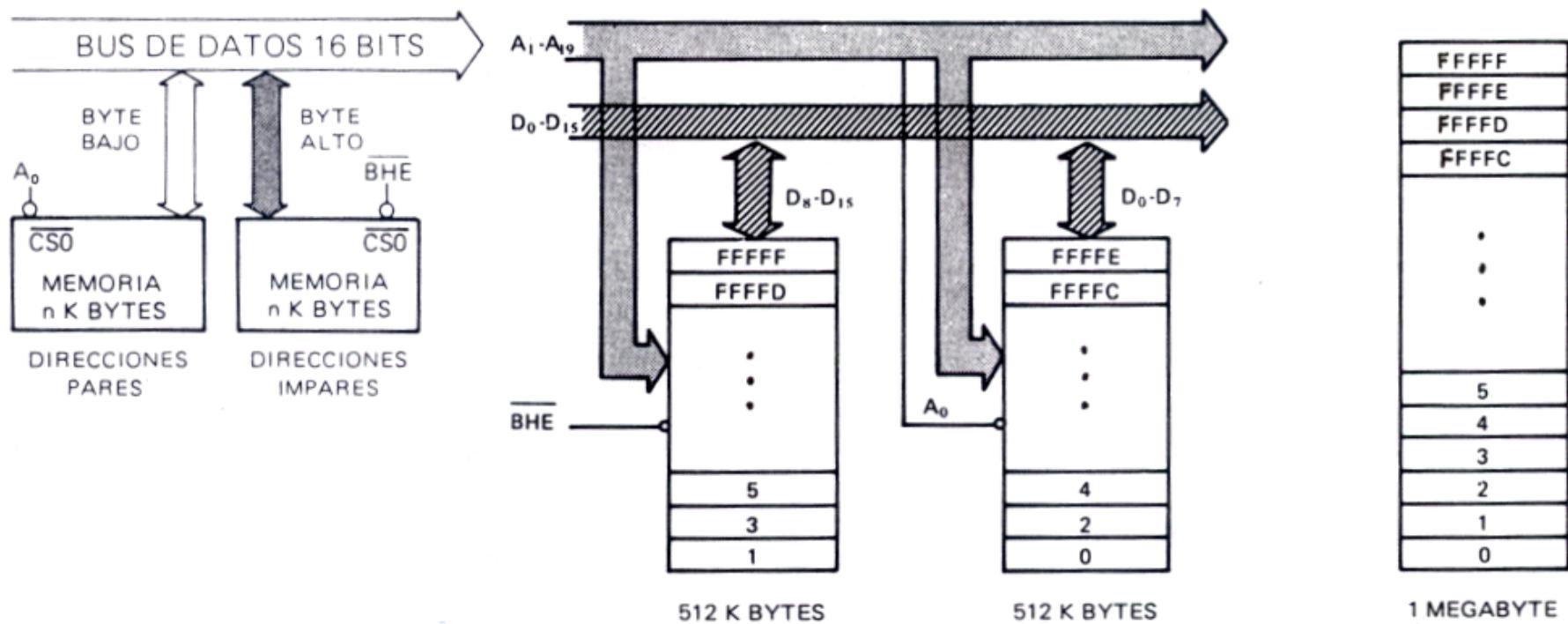
S4	S3	Segmento
0	0	EXTRA
0	1	STACK
1	0	CODE
1	1	DATA

# Señales del 8086

## BHE / S7

- BUS HIGH ENABLE (validación del byte más significativo del bus de datos), multiplexada en el tiempo con la señal de estado S7, que no se usa en el 8086 (estaba prevista para otras versiones).

BHE#	A0	Dato transferido
0	0	Palabra completa en dirección par
0	1	Byte alto desde/a dirección impar
1	0	Byte bajo desde/a dirección par
1	1	Nada



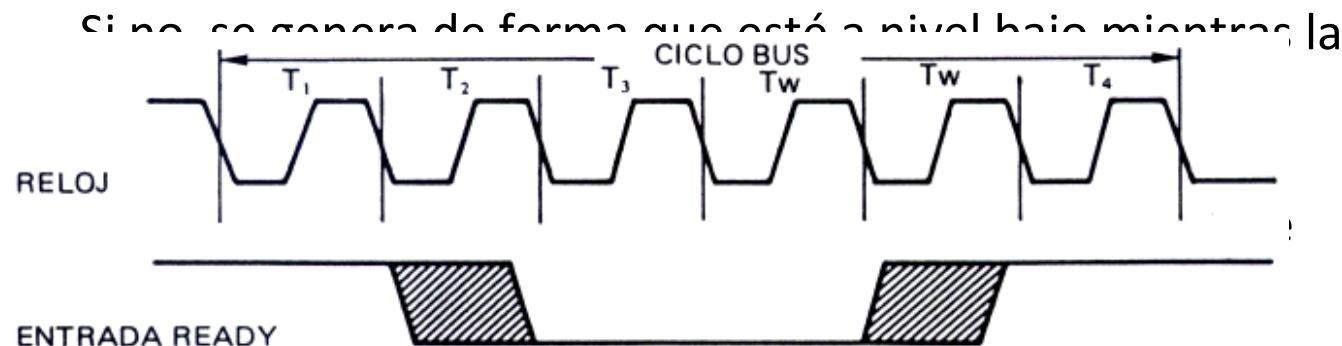
# Señales del 8086

## ■ RD

- Indica que el 8086 está realizando un ciclo de lectura
  - de memoria, o
  - de un periférico.
- dependiendo de M/IO# - S2#
- Se puede activar (=0) durante los estados T2, T3 y TW.

## ■ READY

- Para efectuar intercambios asíncronos entre procesador y mem.-E/S.
  - Si la memoria / E/S son lo bastante rápidos, se conecta a +5V.



# Señales del 8086

## ■ CLK

- Entrada de reloj que proviene del 8284 (circuito generador de pulsos de reloj).

## ■ RESET

- La activación de esta señal durante al menos 4 pulsos de reloj inicializa al 8086:

- Borra la cola de instrucciones.
- Borra los indicadores de estado.
- Borra DS, SS y ES.
- Borra el IP.
- Fija CS a FFFFh.

} Se salta a FFFF0h



# Señales del 8086

## ■ NMI

- NON MASKABLE INTERRUPT (Interrupción no enmascarable).
- Debido a la transición de nivel bajo a alto de esta señal, se activa la interrupción correspondiente al vector 2 al final de la instrucción actual.
- No existe ninguna forma “software” de inhabilitarla.
- Es prioritaria sobre INTR.

## ■ INTR

- INTERRUPT REQUEST (Petición de interrupción).
- Se comprueba al final de cada instrucción si está a nivel alto, en cuyo caso se entra en el reconocimiento de la interrupción.
- Es vectorizada.
- Puede ser inhabilitada por software poniendo a 0 el indicador de estado IF.
- Se suele conectar al controlador de interrupciones 8259.

# Señales del 8086

## ■ TEST

- Utilizada para la sincronización con un procesador especializado con el 8086.
- Es examinada por la instrucción WAIT.
  - Si TEST# = 0 la ejecución continúa.
  - Si TEST# = 1 el 8086 espera.

## ■ MN / MX

- Selecciona modo mínimo (=1) o modo máximo (=0).

# Señales del 8086

## ■ Señales específicas del modo mínimo.

### ■ M / IO:

- Distingue entre un acceso a memoria (=1) y un acceso a un periférico (=0).
- Es equivalente a S2# en modo máximo.

### ■ WR:

- Indica que el 8086 está realizando un ciclo de escritura en memoria o en un periférico, dependiendo de M/IO#.
- Se puede activar (=0) durante los estados T2, T3 y TW.

### ■ INTA:

- INTERRUPT ACKNOWLEDGE (Reconocimiento o autorización de interrupción).
- El 8086 genera dos impulsos en esta salida si la petición de interrupción procede de INTR. En respuesta, el 8259 escribe el número del vector de interrupción en el bus de datos.

# Señales del 8086

## ■ ALE

- ADDRESS LATCH ENABLE (en el *latch 8282*).
- Es un pulso a nivel alto durante el estado T1.

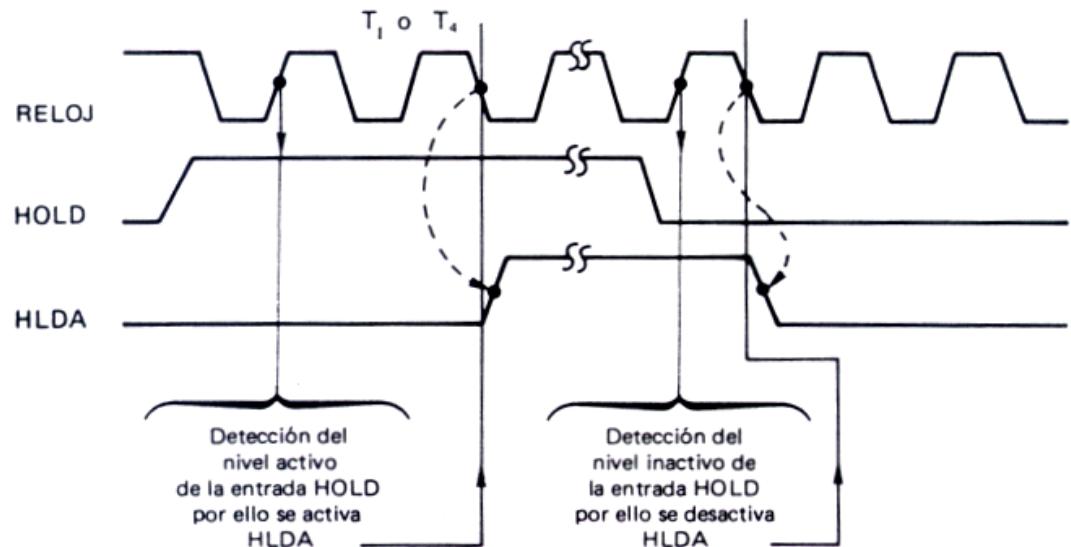
## ■ DT / $\bar{R}$ y DEN

- DATA TRANSMIT / RECEIVE y DATA ENABLE.
- Las señales de datos del 8086 pueden no ser lo suficientemente potentes si hay muchos circuitos en el bus.
- Se utilizan circuitos de interfaz de potencia (“**transceivers**” o transceptores), como el 8286, capaz de amplificar 8 señales.
  - Cuando no se utiliza el bus, salida en alta impedancia.
  - Cuando se va a utilizar el bus de datos, el 8086 activa DEN# (en los accesos a memoria o E/S y en ciclos INTA#).
- La señal DT / R# selecciona el sentido de la transferencia:
  - =0  $\Rightarrow$  hacia el 8086 (entrada).
  - =1  $\Rightarrow$  hacia la memoria o un periférico (salida).

# Señales del 8086

## ■ HOLD y HLDA

- Petición y concesión del bus.
- En acceso directo a memoria, HOLD es activada por el controlador de DMA o el procesador de E/S 8089.
- El 8086 activa HLDA y se aisla (alta impedancia) de:
  - Bus de direcciones / datos.
  - 4 bits de mayor peso del bus de direcciones.
  - Señales de control BHE#/S7, RD#, M/IO#, WR#, INTA#, DT/R# y DEN#.

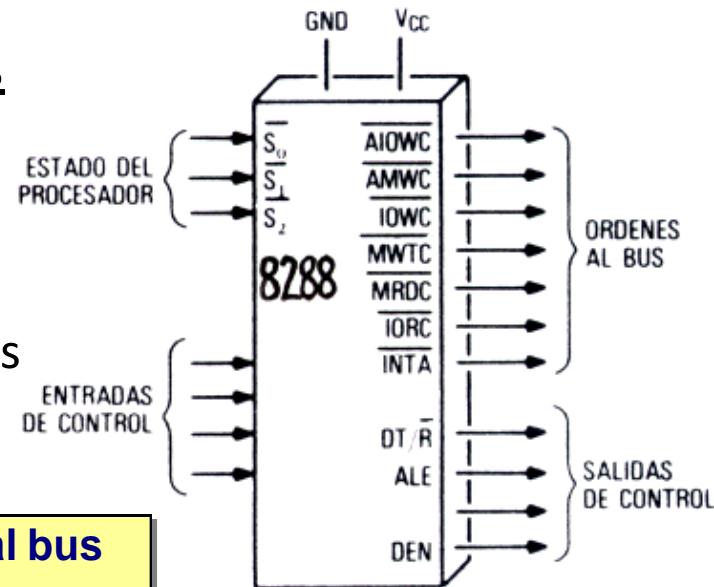


# Señales del 8086

## ■ Señales específicas del modo máximo.

- S0, S1, S2:

- Señales de estado suministradas por el 8086 y usadas por el controlador del bus 8288 para generar todas las señales de control de acceso a memoria y E/S.

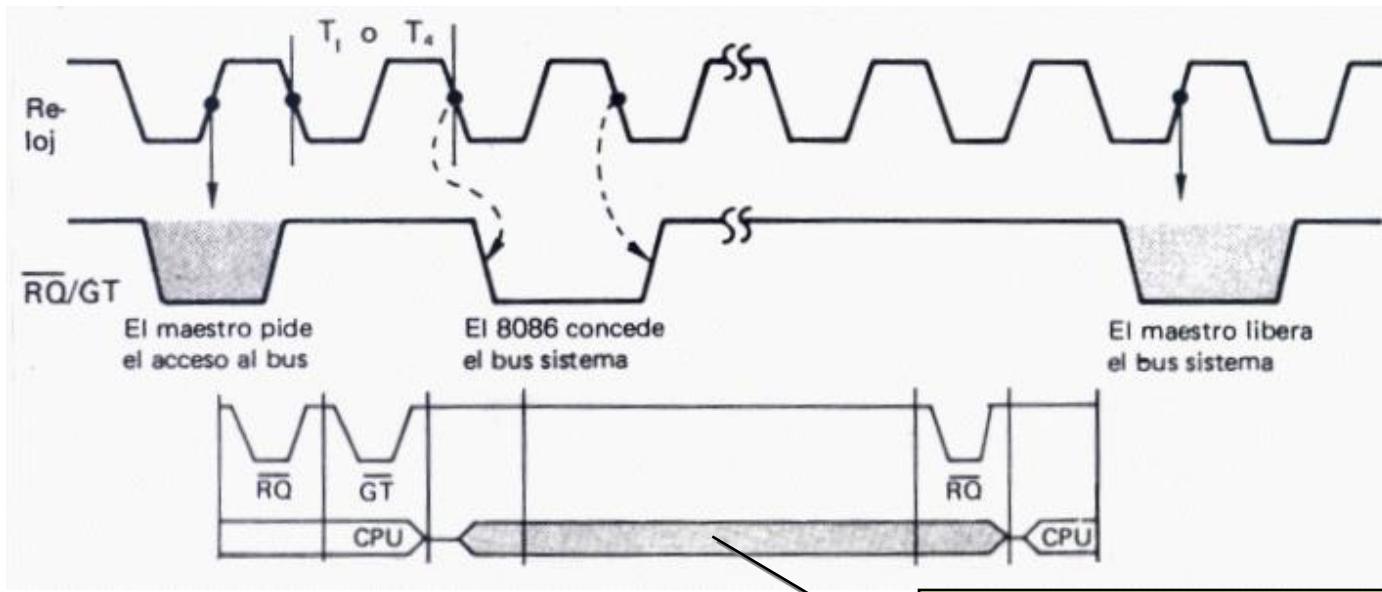


S2	S1	S0	Estado del procesador (tipo de transferencia en curso)	Órdenes al bus del 8288
0	0	0	reconocimiento de interrupción	INTA#
0	0	1	lectura de puerto de E/S	IORC#
0	1	0	escritura en puerto de E/S	IOWC#, AIOWC#
0	1	1	HALT	ninguna
1	0	0	búsqueda de instrucción	MRDC#
1	0	1	lectura de memoria	MRDC#
1	1	0	escritura en memoria	MWTC#, AMWC#
1	1	1	pasivo	ninguna

# Señales del 8086

## ■ RQ / GT<sub>0</sub> y RQ / GT<sub>1</sub>

- REQUEST / GRANT. Petición de acceso al bus y su autorización.
- Patillas usadas por otros posibles maestros del bus para forzar al procesador a liberar el bus al final del ciclo en curso.
- Cada patilla es bidireccional.
- RQ# / GT<sub>0</sub># tiene mayor prioridad que RQ# / GT<sub>1</sub>#.



# Señales del 8086

## ■ **LOCK**

- Se activa (=0) durante la ejecución de las instrucciones precedidas del prefijo LOCK (y durante T2 del primer ciclo INTA).
- Prohibe el acceso al bus a cualquier otro circuito distinto del 8086.

## ■ **QS0, QS1**

- QUEUE STATUS (Estado de la cola).
- Estas dos salidas definen la actividad de la cola de instrucciones del 8086.
- Están destinadas a los coprocesadores, como el 8087, que las utilizan para coordinarse con el 8086.

QS1	QS0	Estado de la cola de espera
0	0	ninguna operación en curso
0	1	el primer byte de una instrucción se ha extraído de la cola de espera
1	0	la cola de espera está vacía
1	1	otro byte distinto del primero de una instrucción se ha extraído de la cola de espera

# Señales del 8086

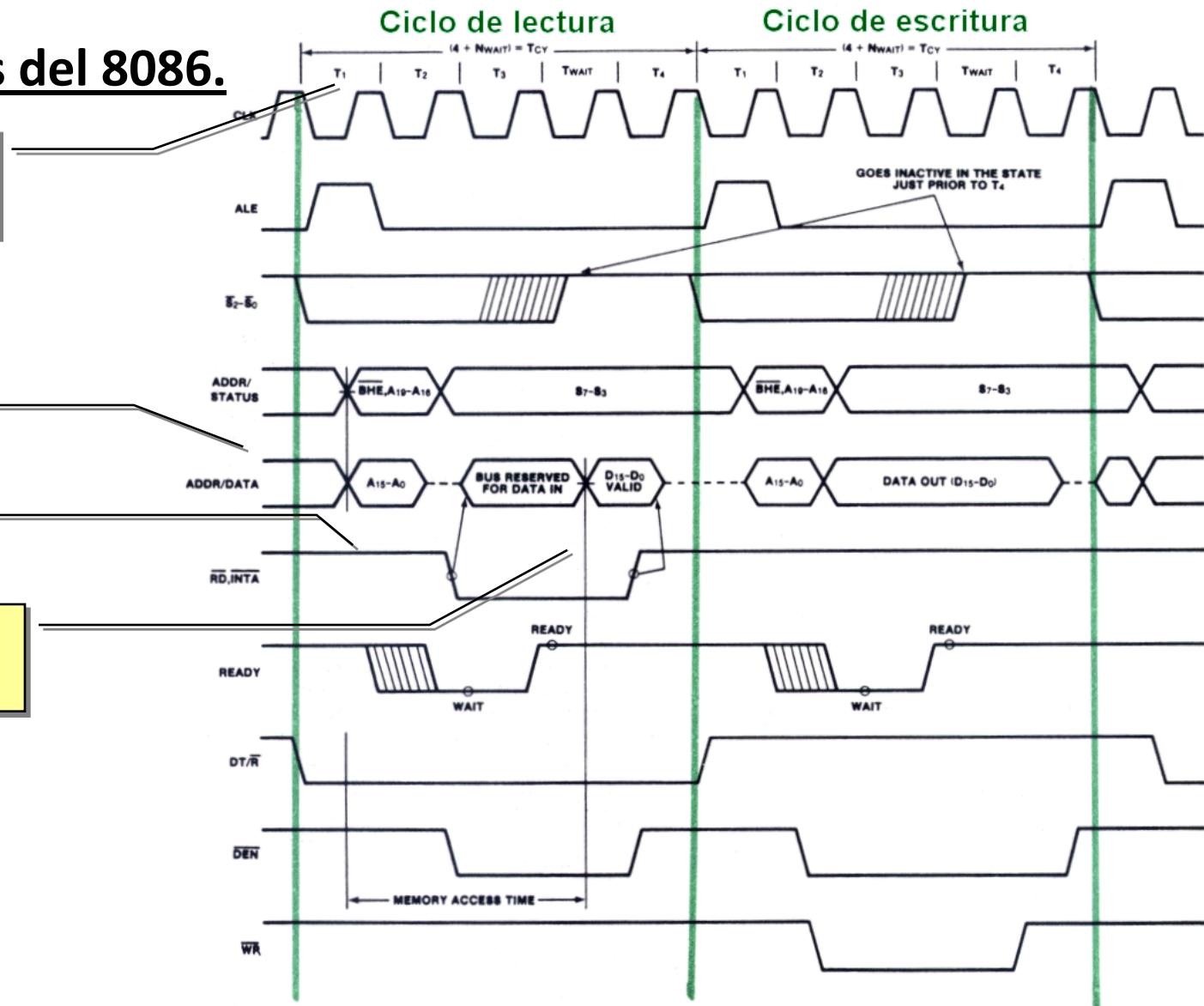
## ■ Ciclo del bus del 8086.

Al menos 4 períodos  
de reloj

1. Dirección  
emitida en T<sub>1</sub>

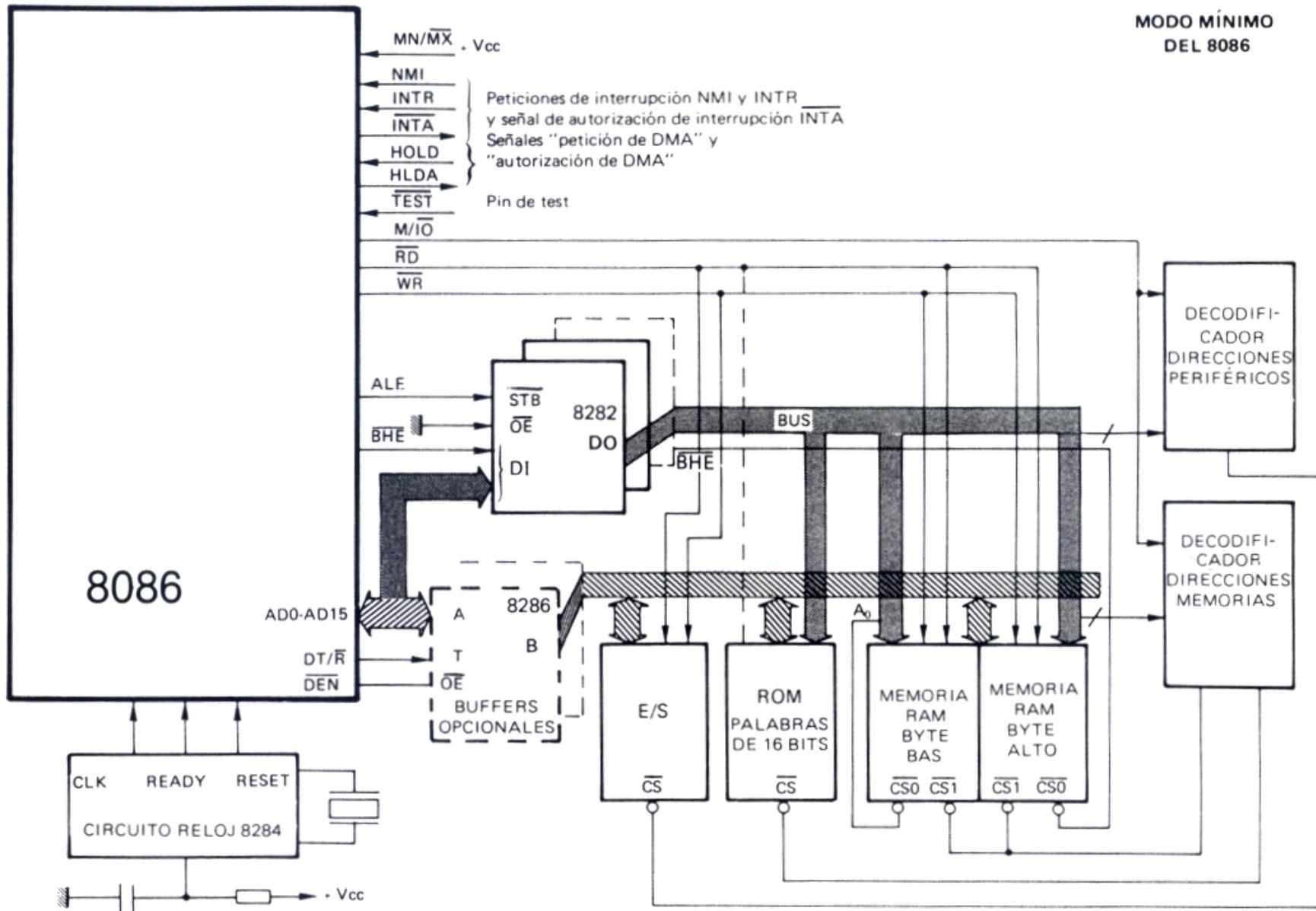
2. En T<sub>2</sub> o T<sub>3</sub> se  
activa RD o WR

3. Los datos leídos se  
estabilizan en T<sub>3</sub> o T<sub>4</sub>



# Señales del 8086

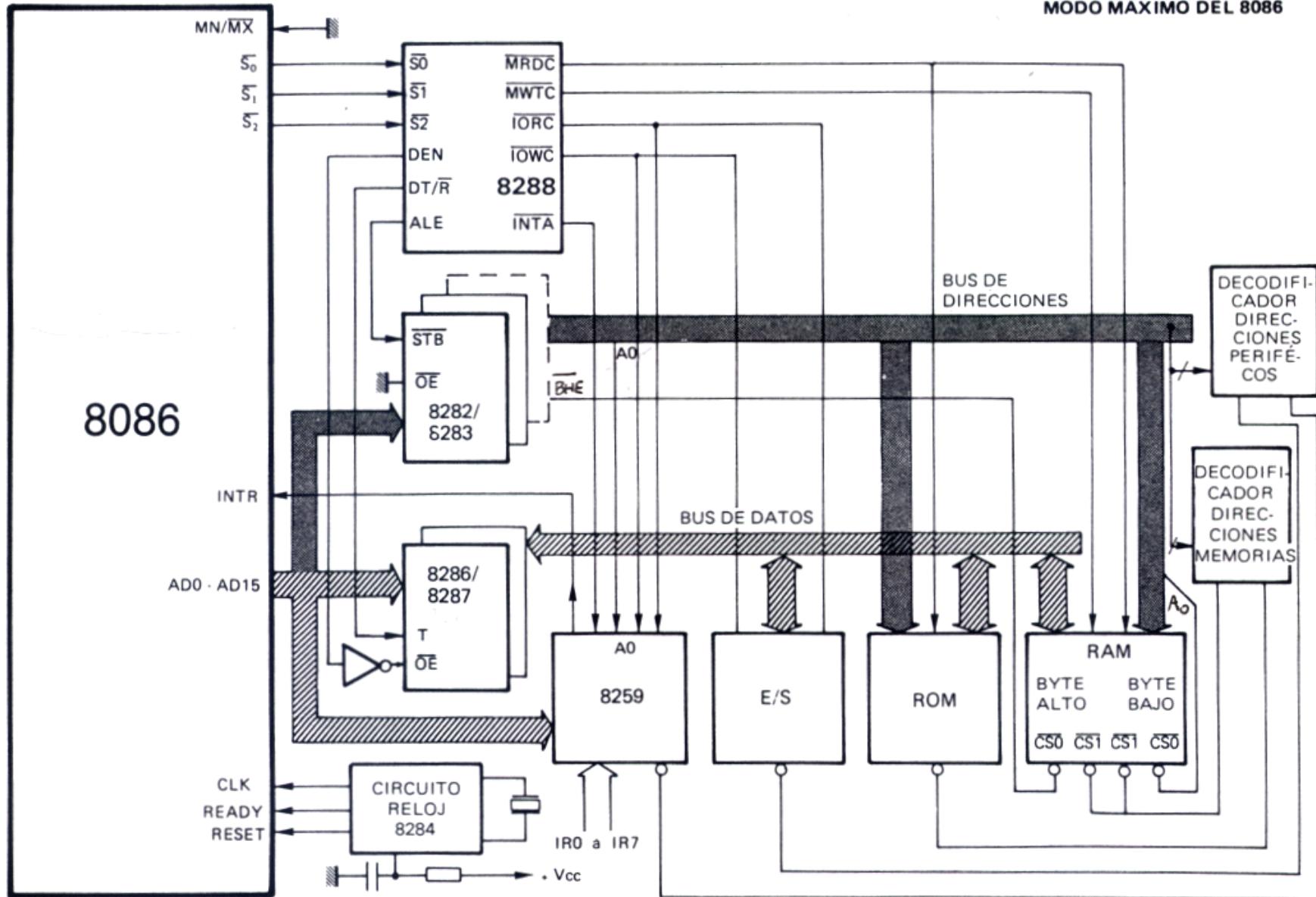
Organización externa en modo mínimo



# Señales del 8086

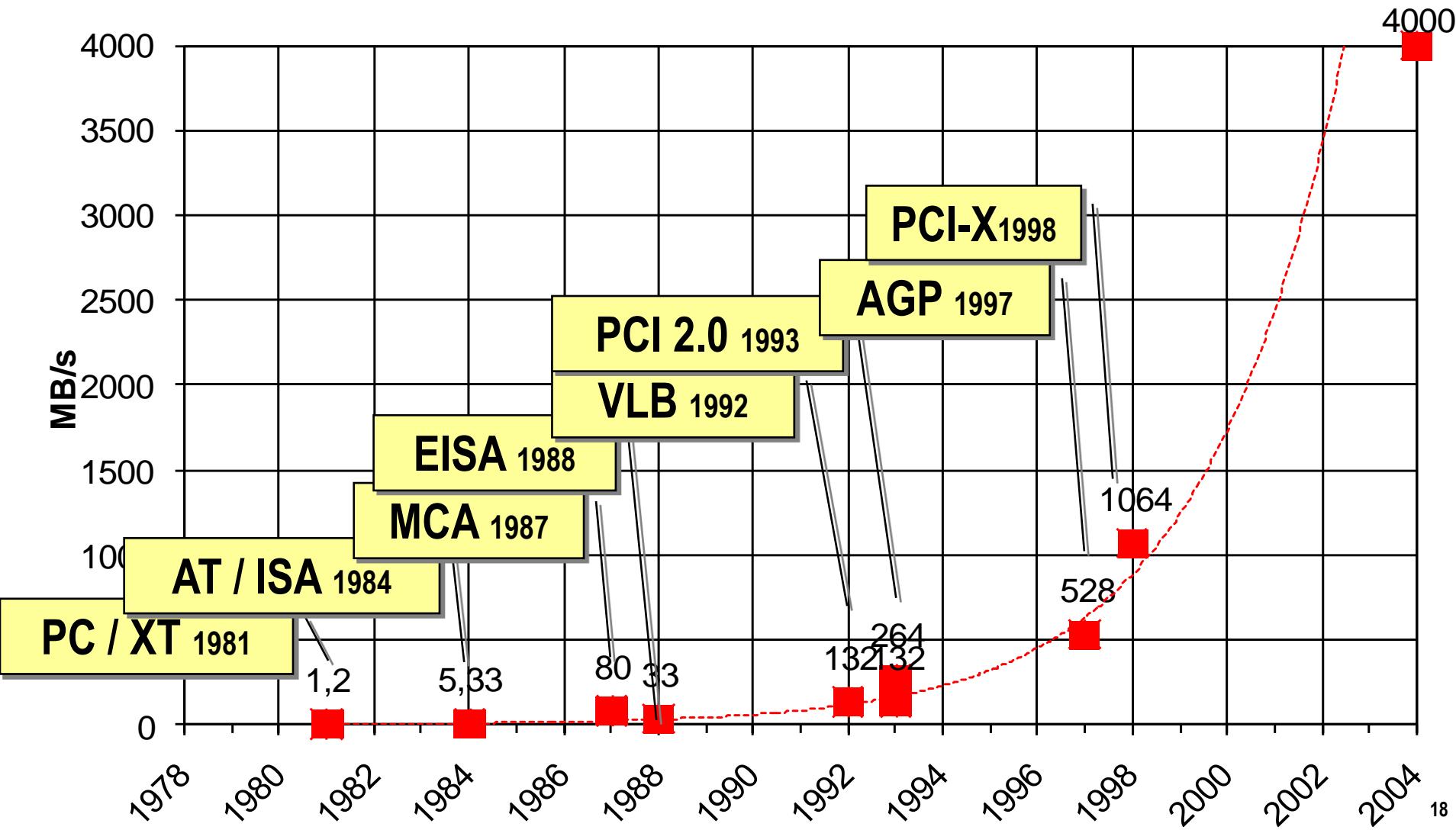
Organización externa en modo máximo

MODO MÁXIMO DEL 8086



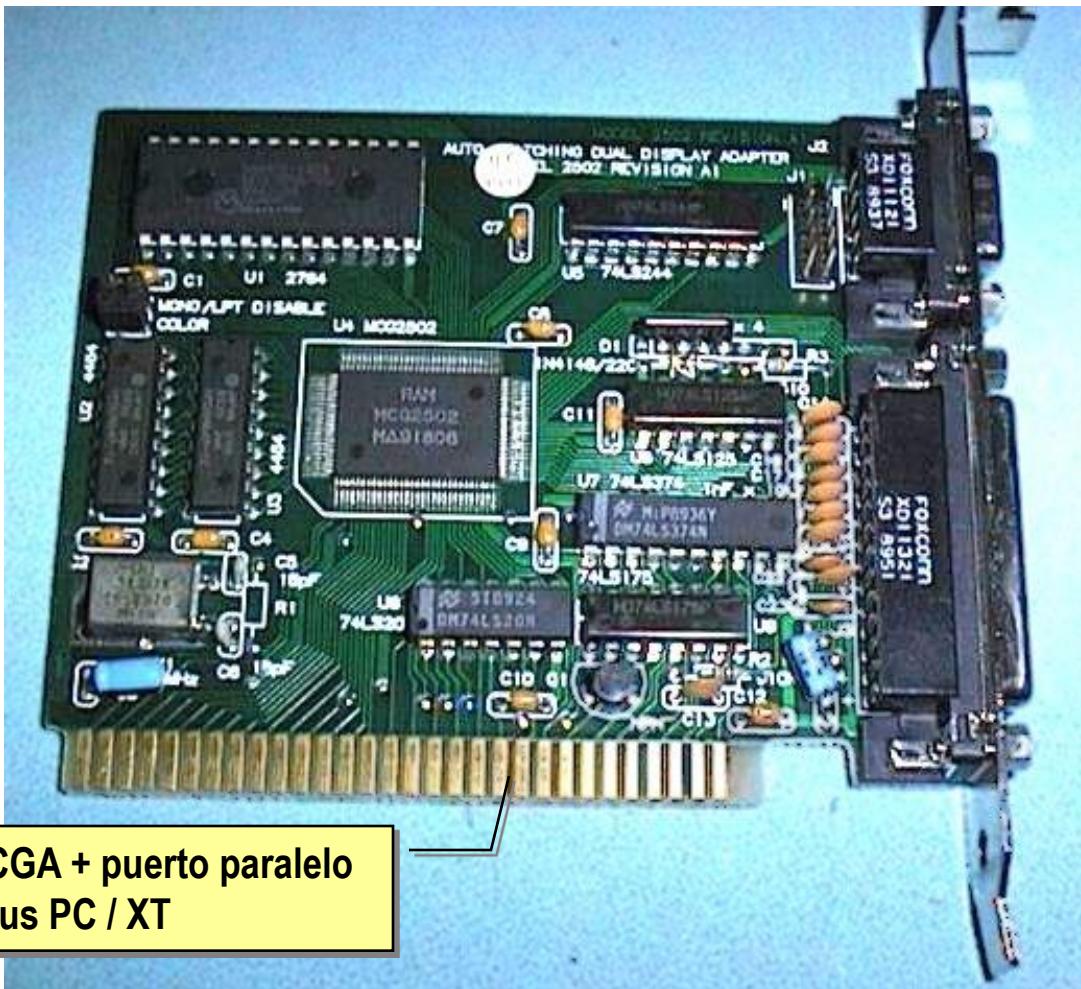
# Buses de placa madre del PC

PCI Express 2004



# Bus PC/XT

## ■ Bus PC / XT (eXtended Technology)



Tarjeta de vídeo CGA + puerto paralelo  
para bus PC / XT

# Bus PC/XT

- Introducido en IBM PC (1981) e IBM XT (1983)

- Conector de 62 contactos.

- 53 señales

- 8 bits de datos.
    - 20 bits de direcciones.
    - 6 interrupciones disparadas por flanco (IRQ2 - IRQ7).
    - 4 canales de DMA:

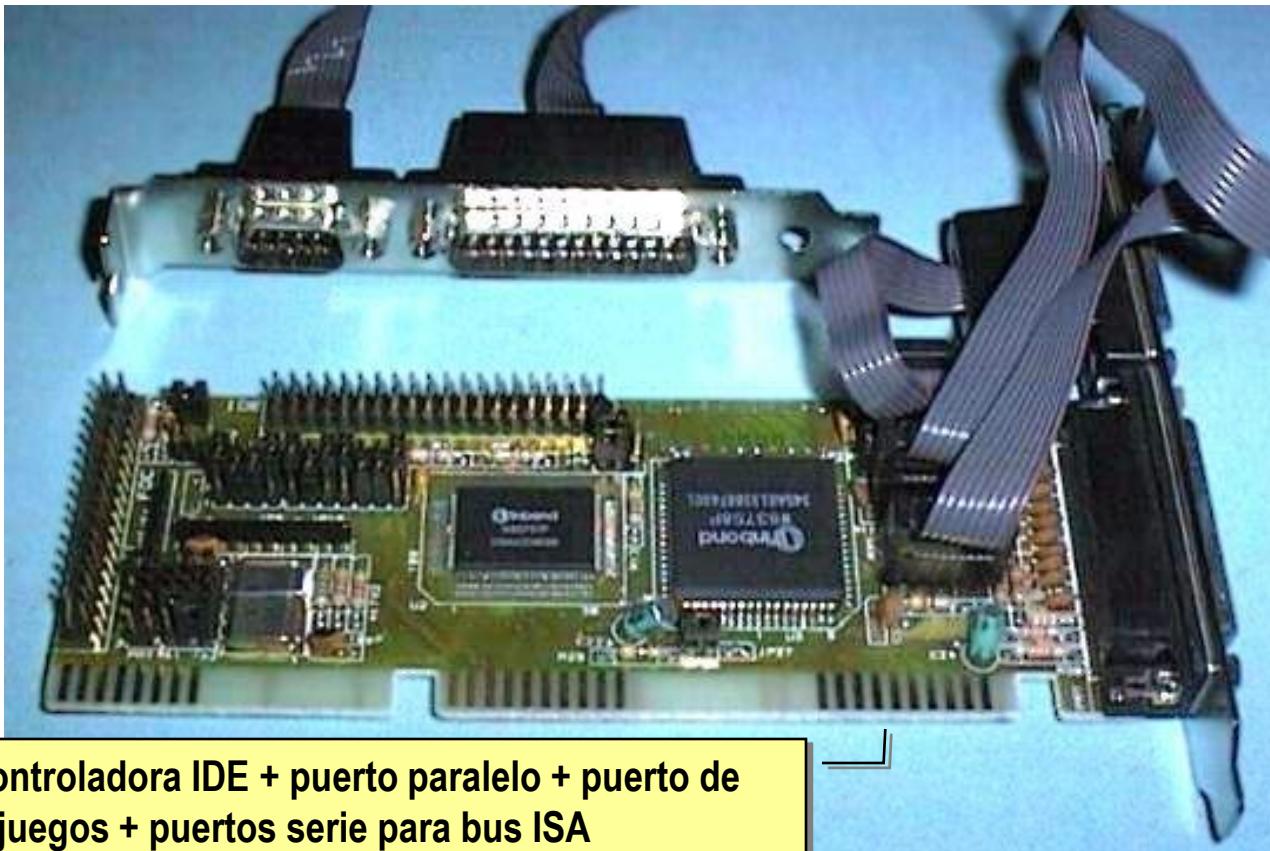
} separados

Canal	¿En bus expansión?	Función estándar
0	No	refresco DRAM
1	Sí	otras tarjetas
2	Sí	controlador FD
3	Sí	controlador HD

- 8 líneas alimentación / tierra.
- 4.77 MHz ⇒ <1.2 MB/s máx.
  - Misma frecuencia de reloj que el 8088 ⇒  placa madre más fácil de diseñar y más barata.

# Bus AT/ISA

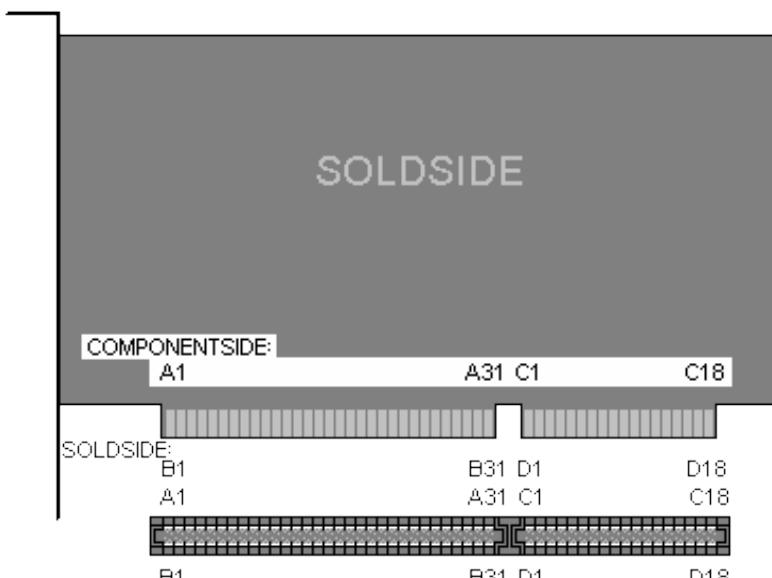
## ■ Bus AT / ISA (Industry Standard Architecture)



Ranura ISA

# Bus AT/ISA

- Introducido en IBM AT 286 (1984)
- Conector de 98 contactos:
  - 62 (=bus PC/XT) + 36 (extensión a 16 bits). Para mantener compatibilidad con tarjetas PC / XT de 8 bits.



- **8/16 bits de datos.**
- 24 bits de direcciones (16 MB) } separados

# Bus AT/ISA

- 6 MHz (IBM AT original), 8 MHz (estándar ISA) ⇒ hasta 5.33 MB/s.
  - 3 ciclos de reloj para una transferencia de 16 bits
  - $(8 \text{ M ciclos/s}) / (3 \text{ ciclos/transf.}) \times (2 \text{ bytes/transf.}) = 5.33 \text{ MB/s}$
  - IBM no hizo ninguna especificación acerca de la velocidad.
  - Posteriores versiones clónicas de 10 - 12 MHz ⇒ problemas con algunas tarjetas.
- Bus único compartido por procesador (único maestro), mem. y E/S.
- 15 interrupciones disparadas por flanco.
  - No compartidas por más de un dispositivo.
- 7 canales de DMA.
- Usado actualmente
  - en PC “de casa” (conjuntamente con bus más moderno y rápido), para tarjetas antiguas, o tarjetas sencillas y baratas.
  - en PC industriales, para tareas de control en fábricas, laboratorios...

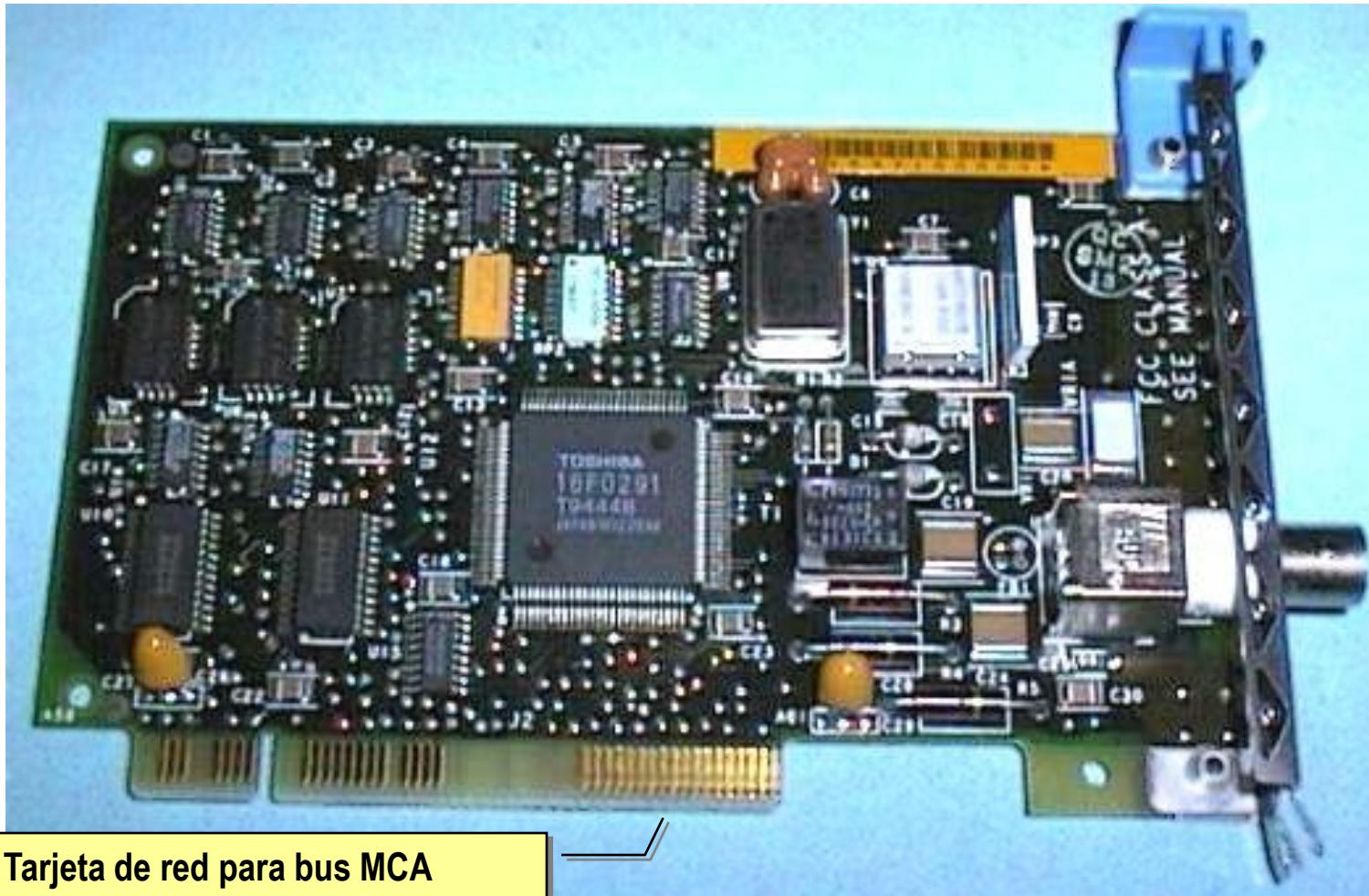
# Buses de placa madre del PC

## ■ ¿Por qué surgen buses estándares de 32 bits?

- Muchos PC tenían parte de su RAM en tarjetas de expansión.
- Intel lanza el 80386 ⇒ posibilidad y deseo de que el procesador accediera a memoria...
  - ...con un ancho del bus de datos de 32 bits,
  - ...a velocidades superiores a 8 MHz.
- Soluciones:
  - Poner la memoria en la placa madre y acceder a través de un bus local de 32 bits.
  - Muchas compañías decidieron colocar ranuras de expansión propietarias, de 32 bits, para tarjetas de memoria.  
     Imposibles de encontrar transcurridos unos meses.
  - Surge la **necesidad de buses estándares de 32 bits** y mayor velocidad que ISA: MCA y EISA.

# Bus MCA

## ■ Bus MCA (Micro Channel Architecture)



# Bus MCA

- Introducido en IBM PS/2 (1987). Bus **propiedad de IBM**.
  - Usado en estaciones de trabajo RS/6000 y en AS/400 y 370.
- Conector de 178 contactos:
  - **8/16/32 bits de datos.**
  - **24/32 bits de direcciones.**
- **10 MHz / 20 MHz**  $\Rightarrow$  hasta 80 MB/s.
  - Asíncrono para transferencias básicas.
  - Síncrono para transferencias en ráfagas (bloque).
- Interrupciones disparadas por nivel
  - Varias tarjetas pueden compartir una misma línea.
- **Autoconfigurable:** registros POS (*Programmable Option Select*)
  - Permite al software identificar los requisitos de una tarjeta y resolver conflictos potenciales en las IRQ, DMA, direcciones...
  - Las tarjetas se configuran automáticamente con un programa en lugar de usar conmutadores y “jumpers”.

# Bus MCA

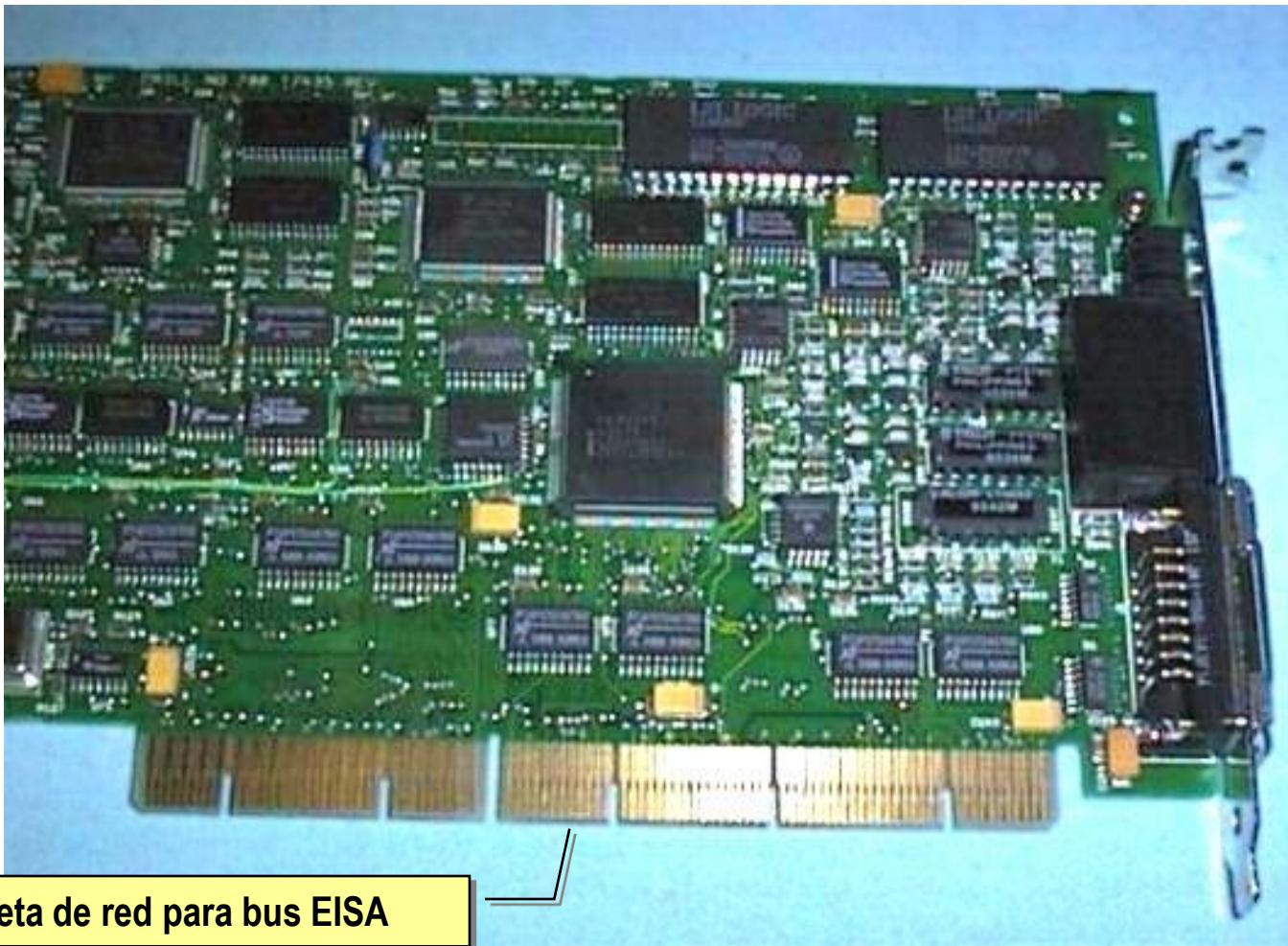
- Capacidad para **tarjetas “busmaster”**.
  - Una tarjeta “busmaster” es «inteligente»: tiene su propio controlador de DMA, con capacidad de generar una secuencia de direcciones de memoria a/de las que transferir datos directamente, sin intervención del procesador.
- Hasta 8 tarjetas en un bus.
- ✖ **No compatible con ISA.**
  - Una tarjeta ISA no se puede insertar en un slot MCA.



- Alto coste de las tarjetas comparado con ISA.
  - Debido a su pequeño tamaño comparado con ISA o EISA.

# Bus EISA

## ■ Bus EISA (*Enhanced Industry Standard Architecture*)

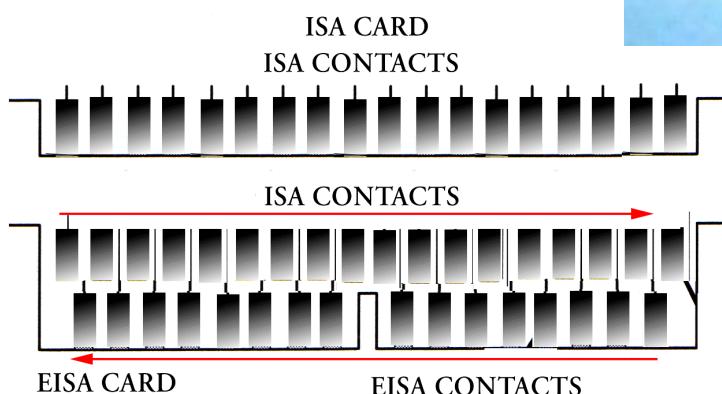


# Bus EISA

- Competencia de IBM, 1988
  - Desarrollado como respuesta al bus MCA.
- Conector de 188 contactos:
  - 98 (=bus ISA) + 90 (extensión).
  - **8/16/32 bits de datos.**
  - **32 bits de direcciones (4 GB).**
- **8.33 MHz** ⇒ hasta **33 MB/s**
  - Para mantener compatibilidad con ISA.
- Interrupciones disparables...
  - por flanco,
  - o por nivel: varias tarjetas pueden compartir una única línea.
- **Autoconfigurable.**
- Capacidad para **tarjetas “busmaster”**.

# Bus EISA

- **Compatible con ISA.**
  - ✓ Gran ventaja frente a MCA.
  - Un slot EISA contiene la interfaz ISA y conectores para las nuevas señales.



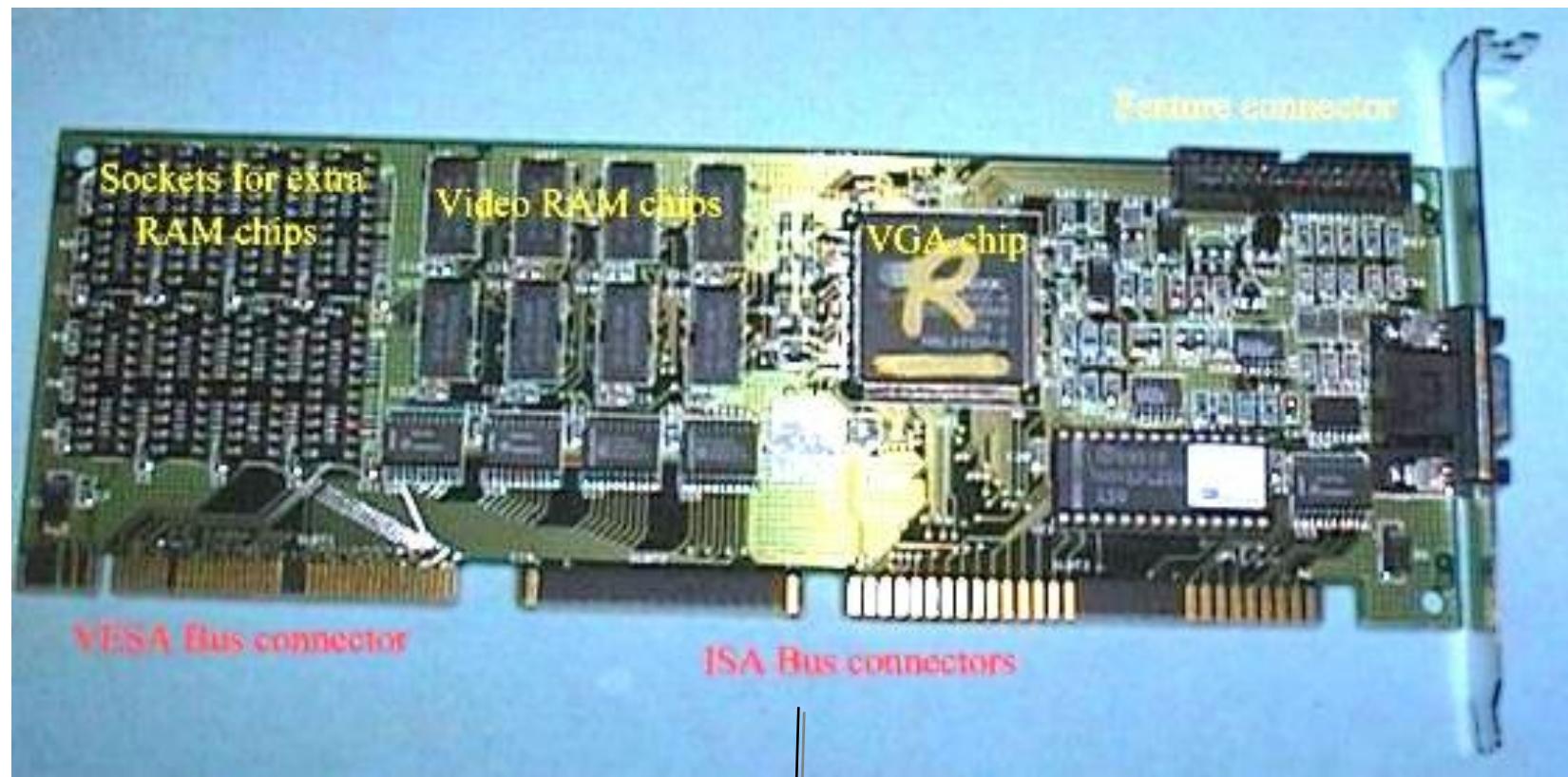
- ✗ Alto coste de las tarjetas comparado con ISA.
  - Sin demasiado éxito comercial.

# Buses de placa madre del PC

- Relativo fracaso comercial de MCA y EISA.  
    ⇒ Continuidad del uso de ISA.
- Microsoft lanza Windows 3.1
  - Usuario demanda **tarjetas gráficas** de alta resolución y más de los 16 colores de una VGA.
- Paralelamente crece la necesidad de mayor velocidad de comunicación con discos duros rápidos y tarjetas de red.
- Bus ISA no suficientemente rápido.
- Soluciones:
  - Algunos fabricantes colocan circuitería de vídeo en placa madre, en el rápido bus local procesador ↔ memoria.
    - ✖ Dificultad o imposibilidad de actualización.
  - Otros incluyen ranuras de expansión propietarias para el vídeo.
    - ✖ El mismo problema que con las de expansión de memoria.
  - Bus local VESA.

# Bus VLB

## ■ Bus VLB (VESA Local Bus)



Tarjeta de vídeo SVGA para bus VESA

# Bus VLB

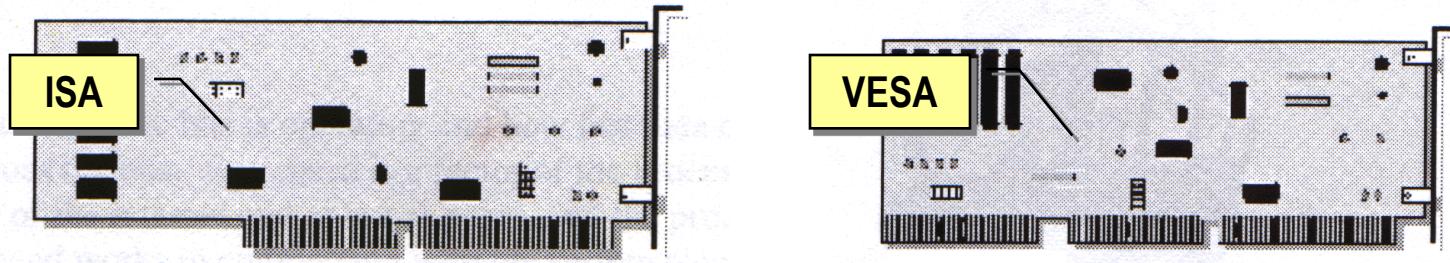
- VESA: *Video Electronics Standards Association*, 1992.
  - Consorcio de compañías para definir estándares de pantallas y controladoras de vídeo.
  - Definió estándares para la temporización de las señales y para las resoluciones de la Super VGA.
- 32 bits de datos
- 32 bits de direcciones } separados
  - Diseñado como una extensión del bus del 486.
- 25, 33, 40, 50 MHz
- ✖ Limitado a 2 ó 3 tarjetas.
  - 2 ranuras + 2 dispositivos placa madre para 33 MHz (132 MB/s)
  - 1 ranura para 40 MHz (160 MB/s)
  - Ninguna ranura para 50 MHz.
  - Usado para tarjetas gráficas y controladoras de disco.

...

# Bus VLB

✓ Coste moderado.

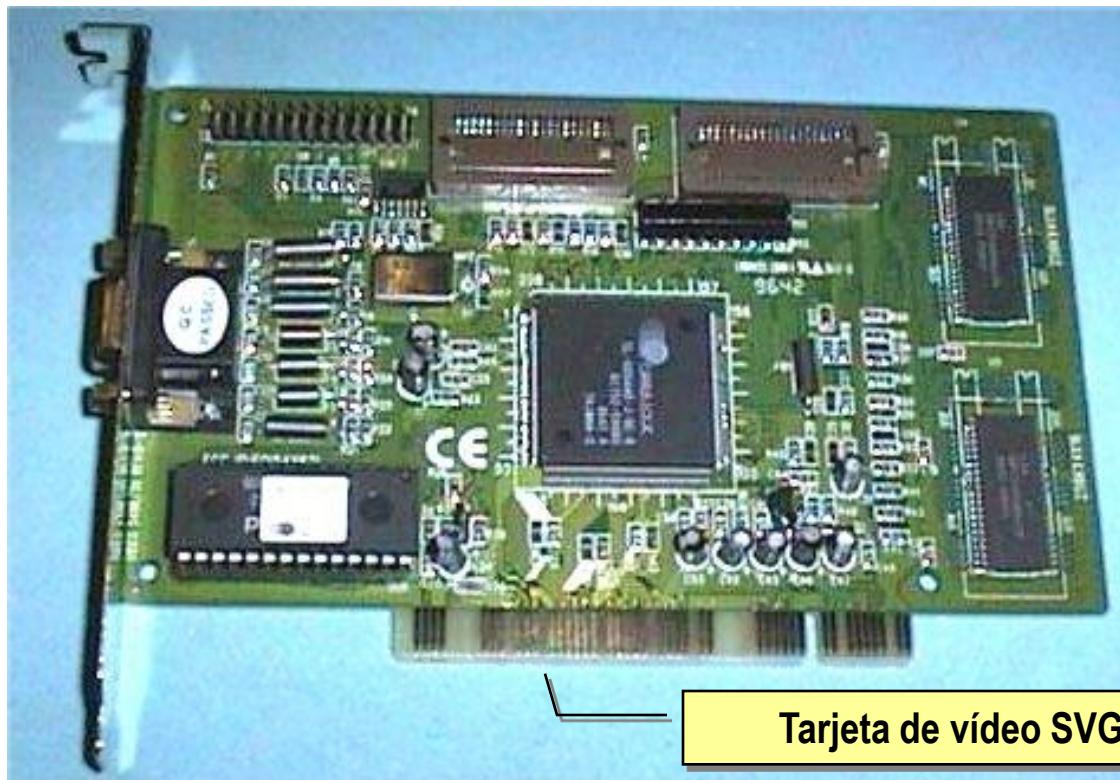
- Para conseguirlo, era un requisito de las especificaciones que el bus se basara en la tecnología y *chip sets* existentes.
- Compatible con ISA.
  - Conector Micro Channel + conector ISA (usado en dispositivos VESA sólo para alimentación).



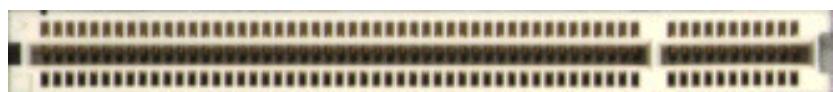
- Una misma tarjeta puede incorporar un dispositivo VESA (ej. interfaz IDE) y uno ISA (ej. *Floppy Disk Controller*).
- Coexiste en la misma placa con ISA / EISA.
- Pronto obsoleto (unos 2 años), sobrepasado por éxito del bus PCI.

# Bus PCI

## ■ Bus PCI (*Peripheral Component Interconnect*)



Tarjeta de vídeo SVGA para bus PCI



Ranura PCI (distingible por su color blanco)

# Bus PCI

- Intel comienza la especificación (1991).
- PCI SIG (*Special Interest Group*)
  - Asociación de fabricantes.
  - Continúa la especificación de nuevos estándares: PCI 2.0 (1993)
- Estándar independiente del procesador:
  - No exclusivo de PC: usado en Macintosh (PowerMac), PowerPC, DEC Alpha y otros ordenadores RISC (HP, Sun).
  - Tarjetas desarrolladas para PC pueden usarse en otro hardware, requiriendo únicamente el *driver* correspondiente.
- Coexiste en la misma máquina con ISA / EISA / MCA, no con NuBus en Mac.
- Indicado para tarjetas que requieran un gran ancho de banda: red, vídeo, adaptadores SCSI.
- ✓ Coste moderado.

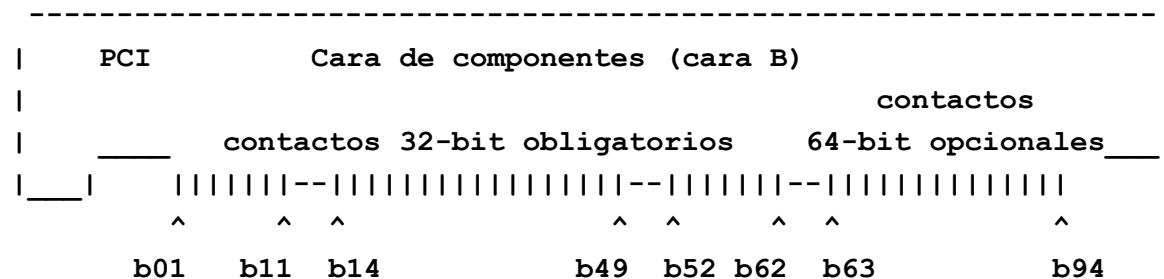
# Bus PCI

- Conector de 124 (188 opc.) contactos:

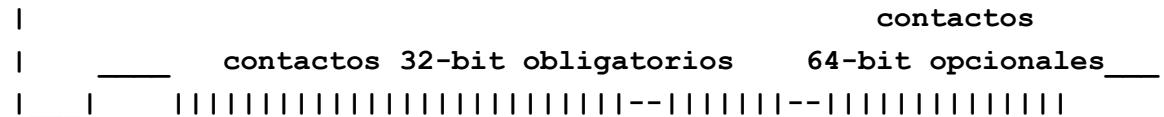
- 32 bits de datos (64 opc.)
  - 32 bits de direcciones (64 opc.)
- } Multiplexados (para reducir costes)

- Admite tarjetas de 5V y de 3.3 V.

Tarjeta PCI Universal 32/64 bit



Tarjeta PCI 5V 32/64 bit



Tarjeta PCI 3.3V 32/64 bit



- **33 MHz** normalm. (hasta 66 MHz en la versión 2.1, sólo para 3.3V).
- Hasta **132 MB/s** a 33 MHz, hasta 264 MB/s para 64 bits.

# Bus PCI

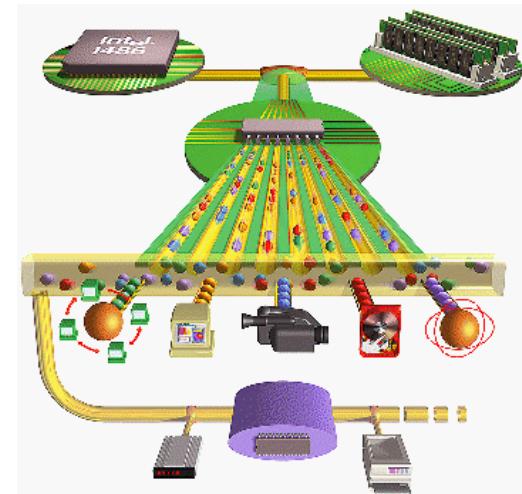
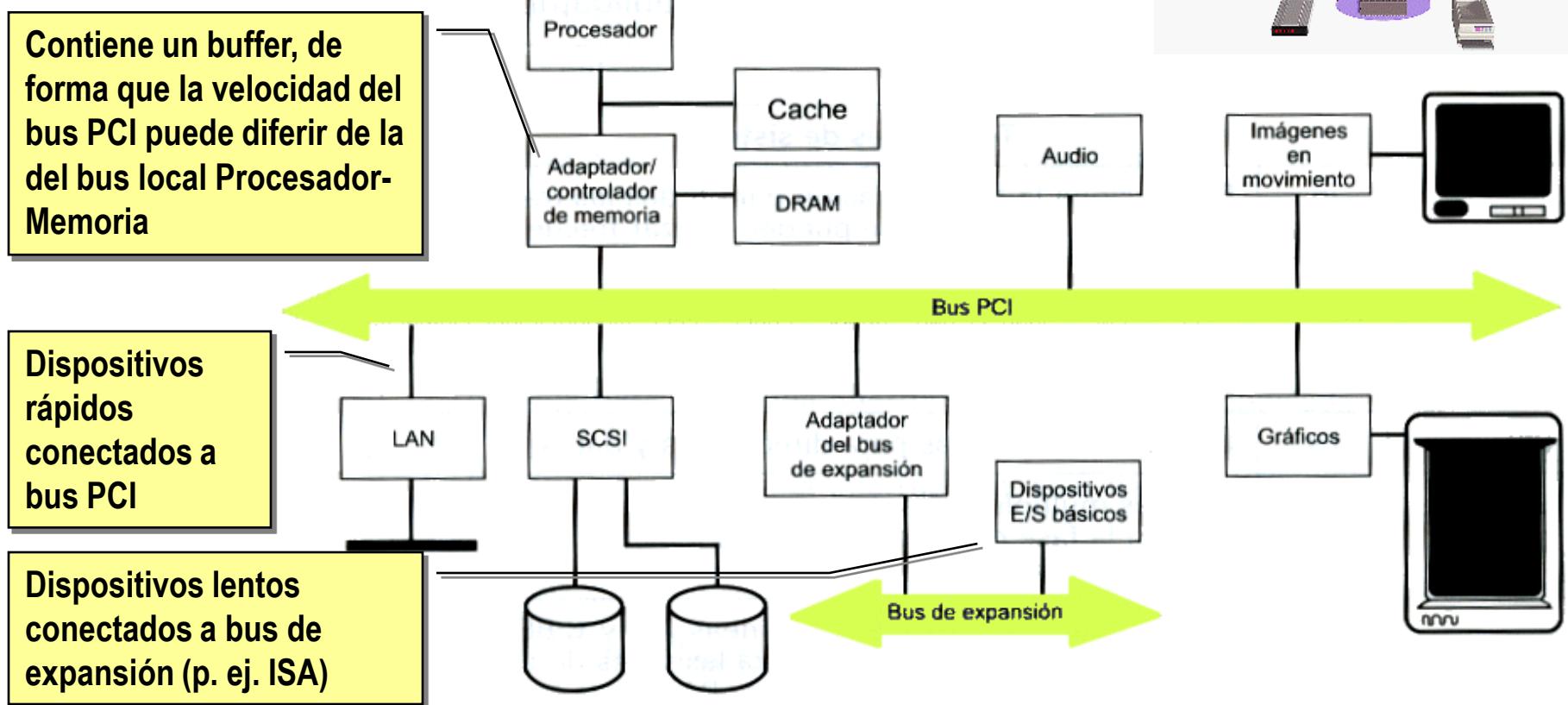
- Temporización síncrona.
- El estándar limita a 4 tarjetas por bus (para 33 MHz)
  - $f = 33 \text{ MHz} \Rightarrow T = 30 \text{ ns}$ , de los cuales 10 ns se emplean en la propagación de las señales por el bus.
  - Se puede aumentar el número de tarjetas
    - reduciendo la frecuencia a 25 MHz  $\Rightarrow T = 40 \text{ ns} \Rightarrow$  el tiempo de propagación de la señal puede ser de hasta 20 ns  $\Rightarrow$  se pueden conectar el doble de tarjetas,
    - añadiendo un adaptador PCI a PCI (*bridge*).
- Capacidad para tarjetas “busmaster”.
- Esquema de arbitraje centralizado.
- “*Busmaster pre-emption*” (Prioridad de los maestros del bus):
  - Un maestro del bus PCI tiene un tiempo límite de posesión del bus cuando otro maestro requiere su uso, prefijado en su temporizador de latencia.

# Bus PCI

- Autoconfigurable:
  - PCI define tres espacios de direcciones:
    - memoria
    - E/S
    - información de configuración
      - » Cada dispositivo PCI está equipado con un área de memoria de 256 bytes, que incluye:
        - » detalles sobre el dispositivo y su tipo (por ej. Adaptador de red, controladora de disco, adaptador de gráficos...)
        - » registros de control y de estado del bus PCI
        - » direcciones de posible ROM de expansión
        - » registros de dirección base
        - » temporizador de latencia
      - » Cuando el sistema arranca, un chipset PCI comprueba esta área para cada dispositivo. A cada uno se le asigna una dir. base y un nº de interrupción únicos.

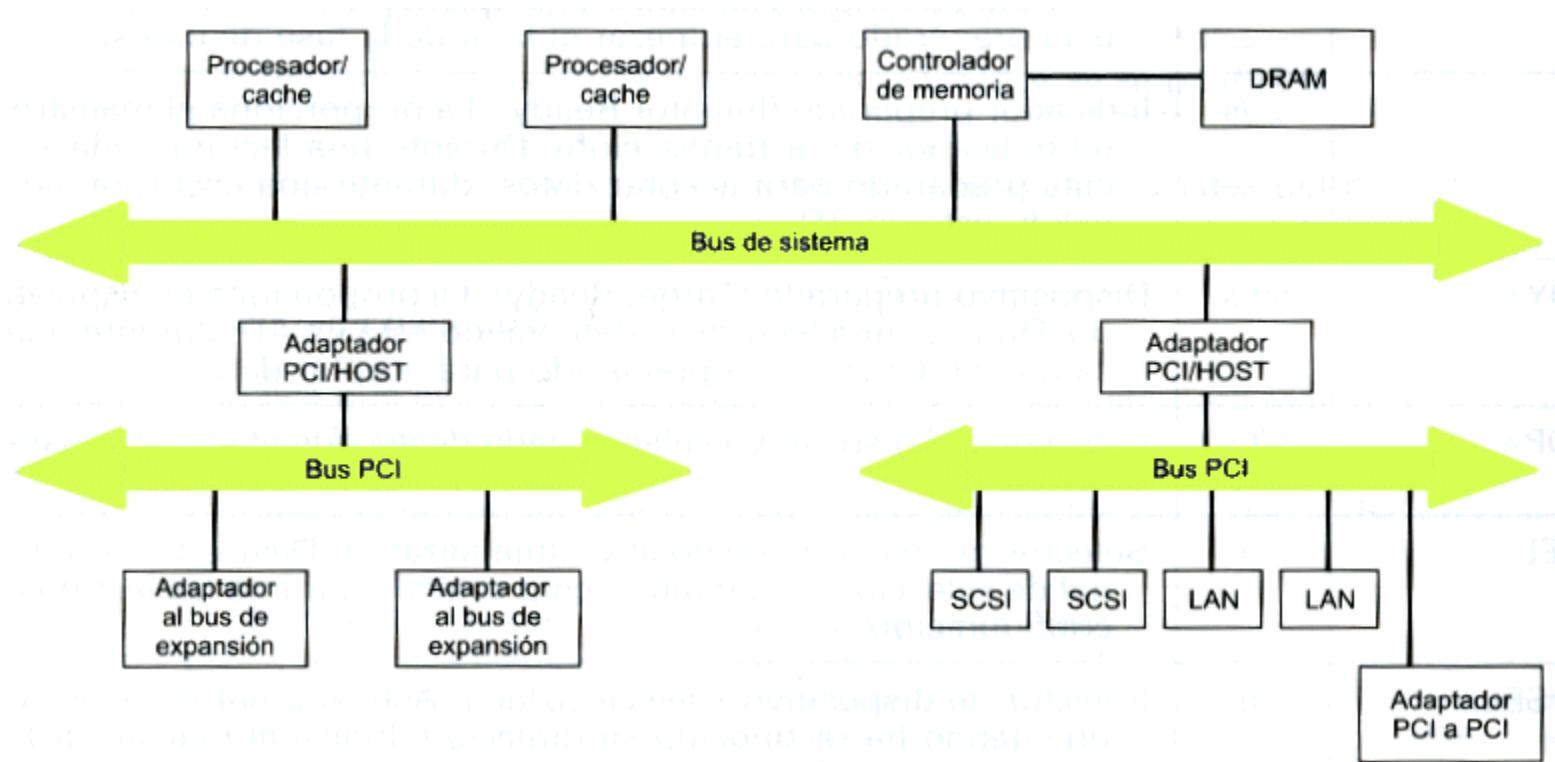
# Bus PCI

- Diseñado para adaptarse a una gran variedad de configuraciones, mono y multiprocesador.
  - Configuración típica **monoprocesador** (sistema de sobremesa):



# Bus PCI

- Configuración típica **multiprocesador** (servidor):



# Bus PCI

- Señales obligatorias (50 líneas)
  - *Líneas del sistema:*
    - **CLK** Proporciona temporización para todas las transacciones.
    - **RST#** Fuerza el sistema a un valor inicial. Reinicia todas las señales del bus.
  - *Líneas de direcciones y datos:*
    - **AD[31:0]** Líneas multiplexadas para direcciones y datos.
    - **C/BE[31:0]#** Líneas multiplexadas para órdenes y habilitación de bytes (durante la fase de datos, indican cuáles de los 4 bytes de datos son válidos).
    - **PAR** Paridad sobre las líneas AD y C/BE#.

# Bus PCI

- *Líneas de control de interfaz (controlan temporización de transacc. y proporcionan coordinación entre iniciadores y destinos):*
  - **FRAME#** Fijada por maestro del bus actual para indicar comienzo y duración de transacción de bloque de datos.
  - **IRDY# (Initiator Ready)**. Fijada por el maestro actual (iniciador de la transacción):
    - » En lectura: el maestro está preparado para aceptar datos.
    - » En escritura: hay datos válidos en AD.
  - **TRDY# (Target Ready)**. Fijada por el esclavo (objetivo de la transacción):
    - » En lectura: hay datos válidos en AD.
    - » En escritura: el objetivo está preparado para aceptar datos.
  - **STOP#** Indica que el objetivo actual desea que el iniciador detenga la transacción actual.
  - **LOCK#** Indica que la operación (puede requerir múltiples transacciones) ha de ser atómica (no interrump. por otro maestro).

# Bus PCI

- **IDSEL** (*Initialization Device Select*). Usado como “chip select” en transacciones de lectura y escritura de configuración (selecciona los registros de configuración).
- **DEVSEL** Fijada por el objetivo cuando reconoce su dirección. Indica al iniciador si se ha seleccionado algún dispositivo.
- *Líneas de arbitraje (cada maestro tiene su par de líneas que lo conectan directamente al árbitro del bus):*
  - **REQ#** Línea punto a punto que le indica al árbitro que un dispositivo requiere uso del bus.
  - **GNT#** Línea punto a punto que le indica al dispositivo demandante que el árbitro le ha concedido acceso al bus.
- *Líneas de información sobre errores:*
  - **PERR#** (*Parity Error*). Indica que un error de paridad ha sido detectado por un objetivo durante una fase de escritura de datos, o por un iniciador en lectura de datos.
  - **SERR#** (*System Error*). Cualquier dispositivo puede mandar un pulso para indicar cualquier error crítico.

# Bus PCI

- Señales opcionales (50 líneas)
  - *Líneas de interrupción:*
    - **INTA#, INTB#, INTC#, INTD#** Usadas para pedir una interrupción. Cada dispositivo puede tener su línea (INTA#) o sus líneas (INTA#, INTB#, ...) de petición de interr. punto a punto.
  - *Líneas de cache:*
    - **SBO#, SDONE** Permiten protocolos de “*snoopy cache*”, para coherencia de cache en sistemas multiprocesador.
  - *Líneas de extensión del bus a 64 bits:*
    - **AD[63:32]** Líneas multiplexadas para direc./datos 64 bits.
    - **C/BE[31:0]#** Líneas multiplexadas para órdenes adicionales y habilitación de los 4 bytes más significativos.
    - **REQ64#** Usada para pedir una transferencia de 64 bits.
    - **ACK64#** Indica que el destino va a realizar la transfer.
    - **PAR64** Paridad sobre las líneas AD y C/BE# extendidas.
  - *Líneas de test:*
    - **TCK, TDI, TDO, TMS, TRST#**

# Bus PCI

## ■ Órdenes PCI

- Actividad del bus transcurre en forma de transacciones entre

Iniciador (maestro)

transacción

Objetivo

Cuando adquiere el control del bus, determina el **tipo de transacción** durante la fase de direcciones, usando las líneas C/BE#

- Reconocimiento de interrupción. El controlador de interrupciones lee el identificador de interrupción (vector).
- Ciclo especial. Enviar mensaje a varios destinos (*broadcast*).
- Lectura de E/S. Maestro lee del esclavo.
- Escritura en E/S. Maestro escribe en el esclavo.
- Lectura de memoria.
- Lectura de una línea de memoria.
- Lectura de memoria múltiple.
- Escritura en memoria.
- Escritura en memoria e invalidación. *Write-back* en cache.
- Lectura de configuración.
- Escritura de configuración.
- Ciclo de direccionamiento dual. Cuando el iniciador usa direccionamiento de 64 bits.

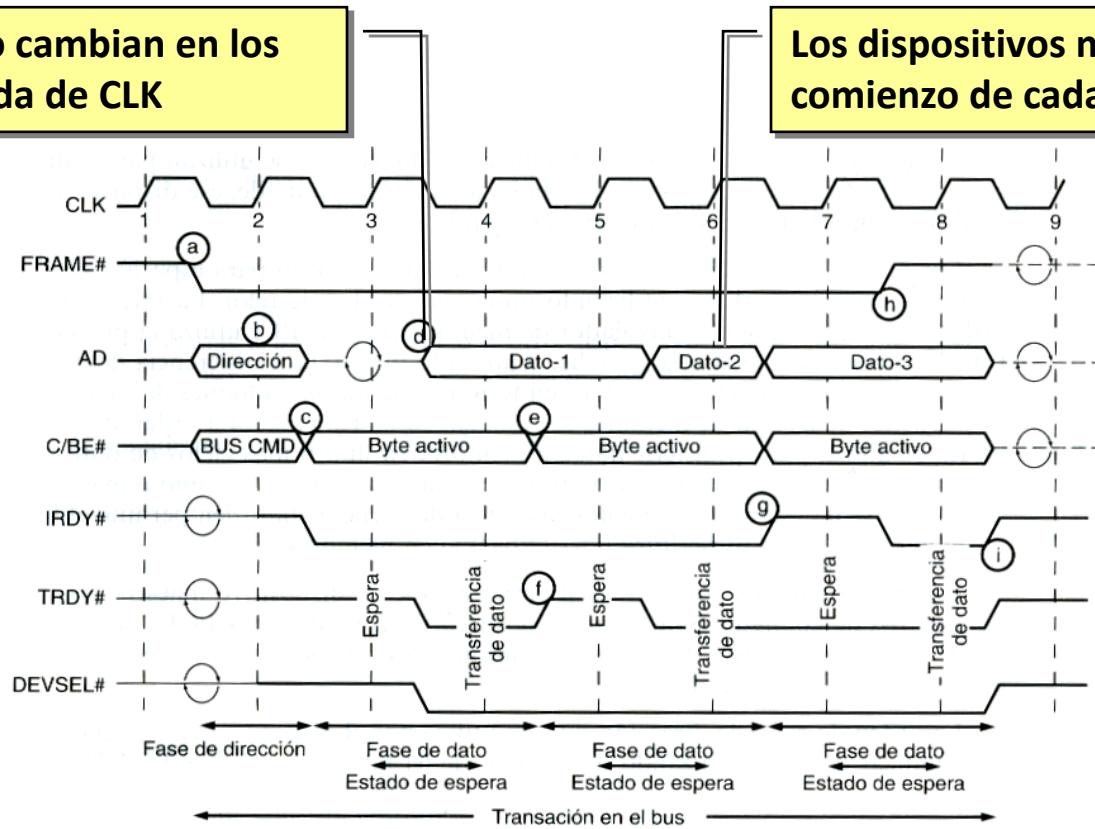
} Se diferencian en el tamaño del bloque leído

# Bus PCI

- Transferencias de datos
  - Cada transferencia de datos es una transacción que consiste en una fase de dirección seguida por una o más fases de datos.
  - Ejemplo: **operación de lectura.**

Las señales sólo cambian en los flancos de bajada de CLK

Los dispositivos muestrean el bus al comienzo de cada periodo de CLK

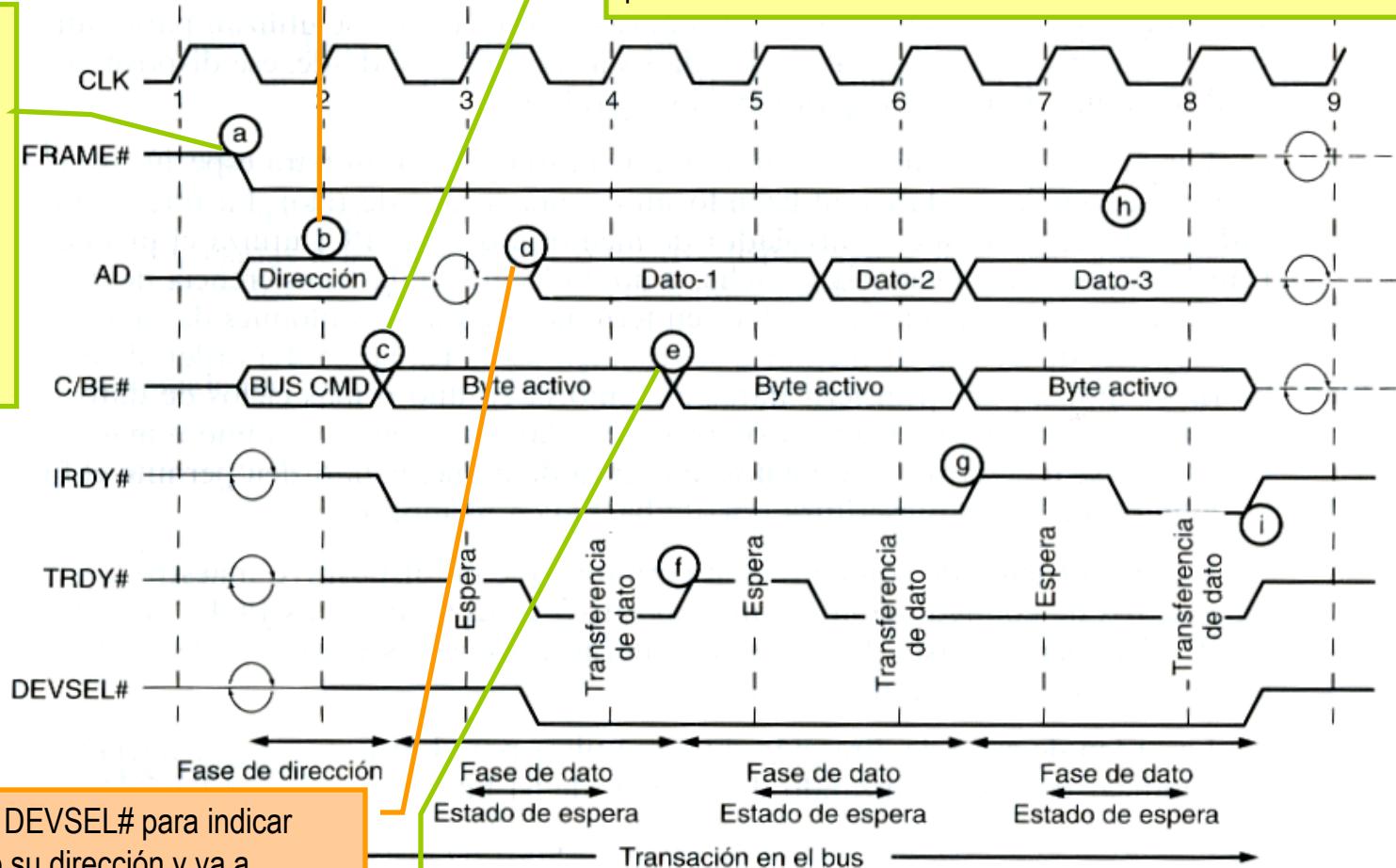


# Bus PCI

a) El iniciador comienza la transacción bajando FRAME#. Pone la dirección inicial en AD y la orden de lectura en C/BE#

b) El objetivo reconoce su dirección.

c) El iniciador cambia las líneas C/BE# para indicar qué líneas AD se usarán para transferir los datos direccionalizados (de 1 a 4 bytes). También baja IRDY# para indicar que está listo para el primer dato.

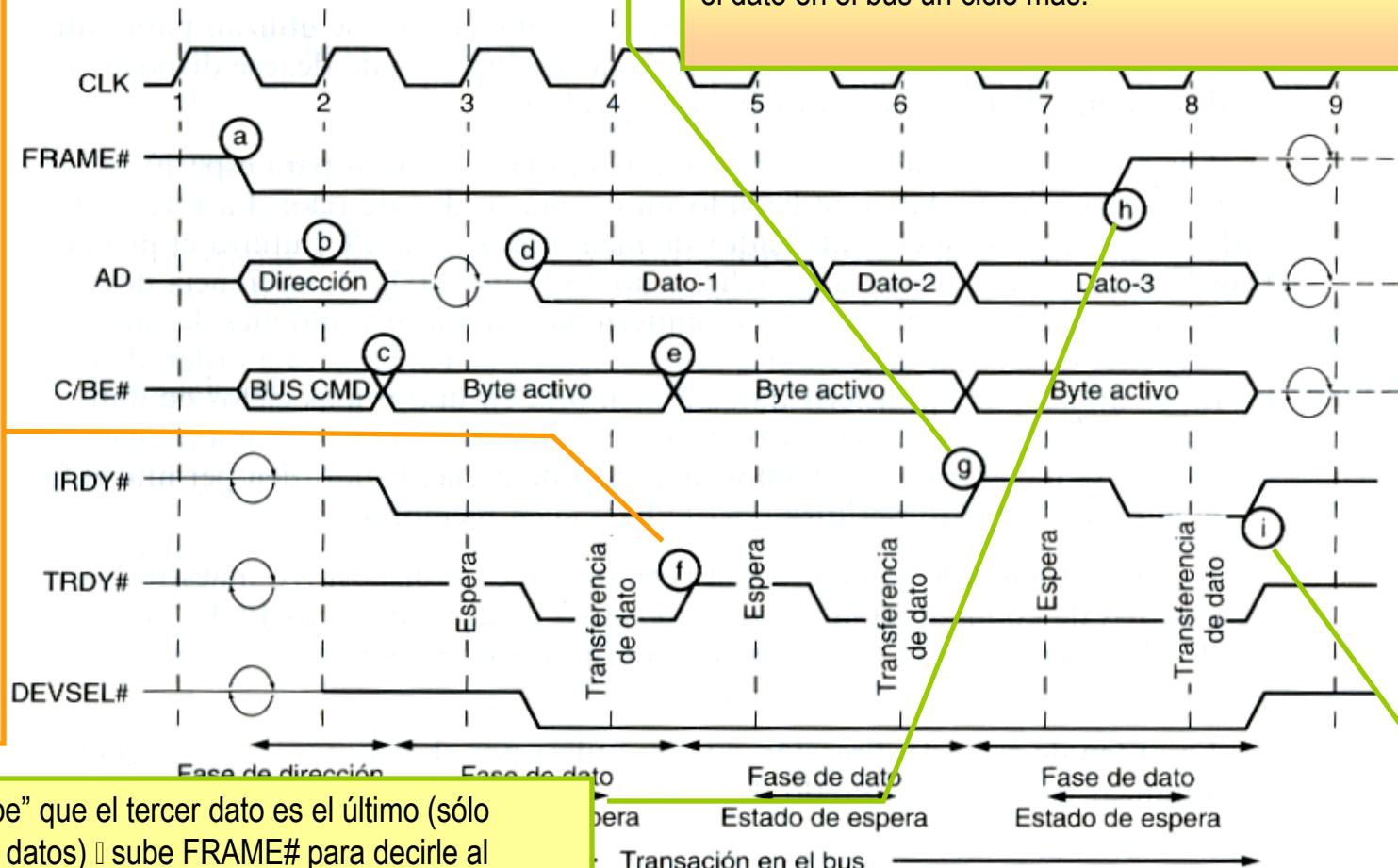


d) El objetivo baja DEVSEL# para indicar que ha reconocido su dirección y va a responder. Pone los datos pedidos en AD y baja TRDY# para indicar datos válidos en el bus.

e) El iniciador lee los datos al comienzo del ciclo 4. Cambia C/BE# si es necesario para la siguiente lectura.

# Bus PCI

f) En este ejemplo, el objetivo necesita algún tiempo para preparar el segundo dato a transmitir → sube TRDY# para decirle al iniciador que no habrá nuevos datos durante el ciclo que comienza. Los datos no se leen al comienzo del ciclo 5, sino del 6.



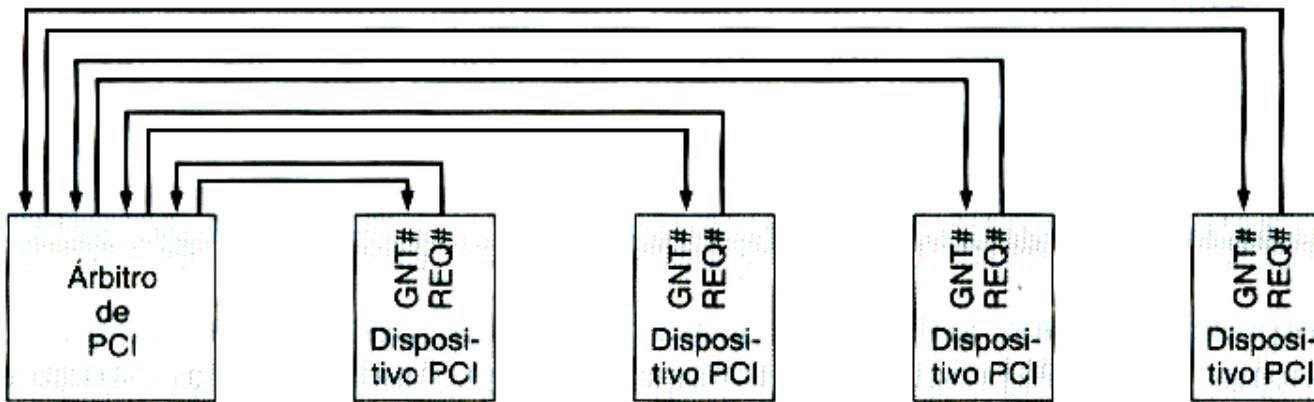
h) El iniciador “sabe” que el tercer dato es el último (sólo quería transmitir 3 datos) → sube FRAME# para decirle al destino que no tiene que transferir más datos. También baja IRDY# para decirle que está listo para completar la transferencia.

g) Durante el ciclo 6, el objetivo pone el tercer dato en el bus. Sin embargo, en este ejemplo, el iniciador no está listo para leer el dato → sube IRDY#. El objetivo mantiene el dato en el bus un ciclo más.

i) El iniciador sube IRDY#. El bus retorna al estado inactivo. El destino sube TRDY# y DEVSEL#.

# Bus PCI

- Arbitraje
  - Esquema de arbitraje centralizado.
  - Cada maestro tiene una señal REQ# y una señal GNT#.
    - Estas líneas llegan a un árbitro central del bus.



- El estándar no especifica el algoritmo de arbitraje:
  - primero en llegar, primero en ser servido,
  - rotación de turnos,...
- Ejemplo: **secuencia de arbitraje de dos dispositivos A y B.**

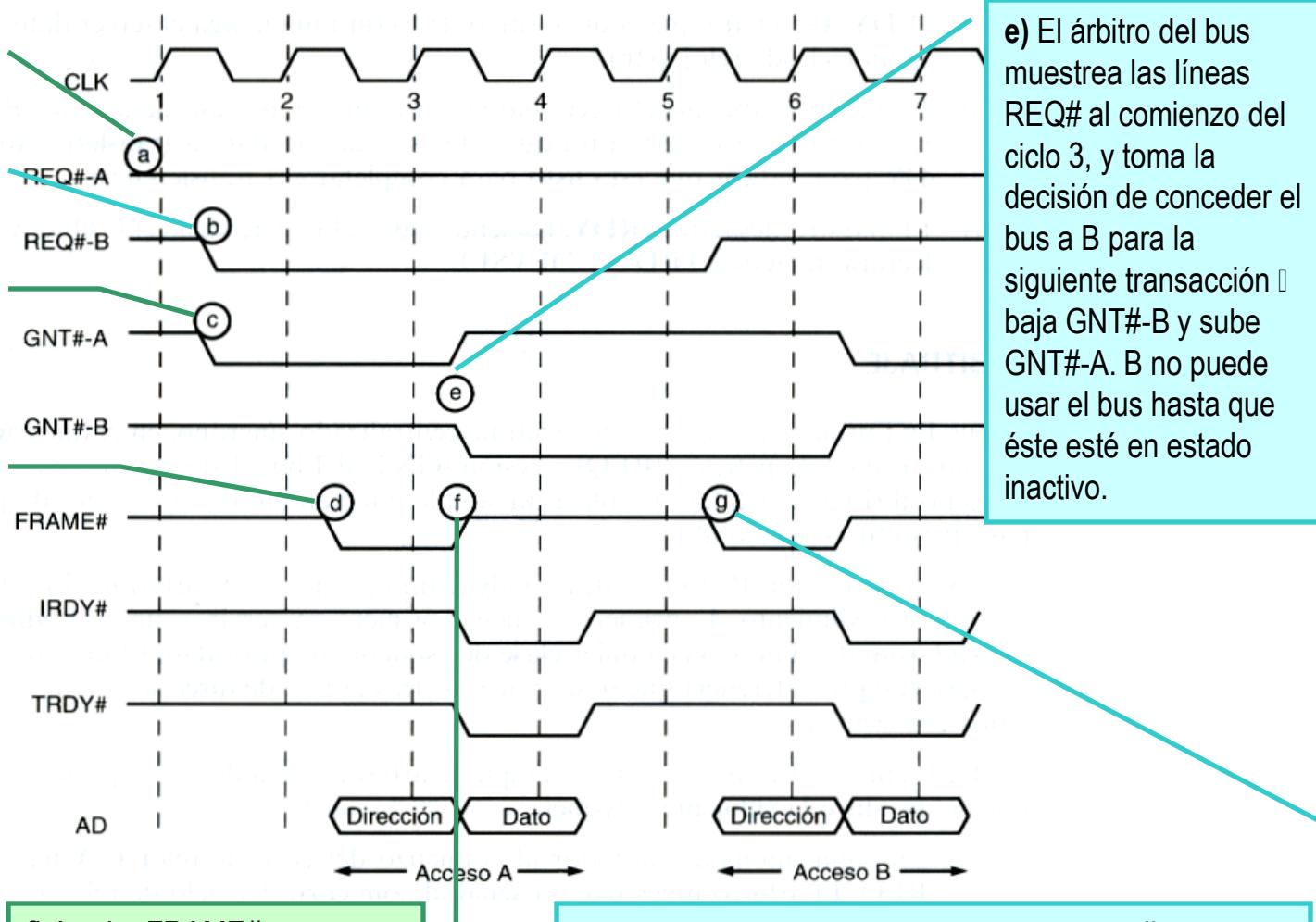
# Bus PCI

a) A tiene activada su línea de petición

b) B pide el uso del bus.

c) Al mismo tiempo, el árbitro concede el bus a A.

d) A muestrea GNT#-A al comienzo del ciclo 2 y sabe que se le ha concedido el bus. Realiza una transacción, pero sigue con REQ#-A a nivel bajo, porque quiere realizar una segunda transacción.



f) A sube FRAME# para indicar que la última (única) transferencia de datos está en progreso.

g) Al comienzo del ciclo 5, B encuentra IRDY# y FRAME# desactivados. Toma el control del bus bajando FRAME#. También sube REQ-B# porque sólo quiere hacer una transacción.

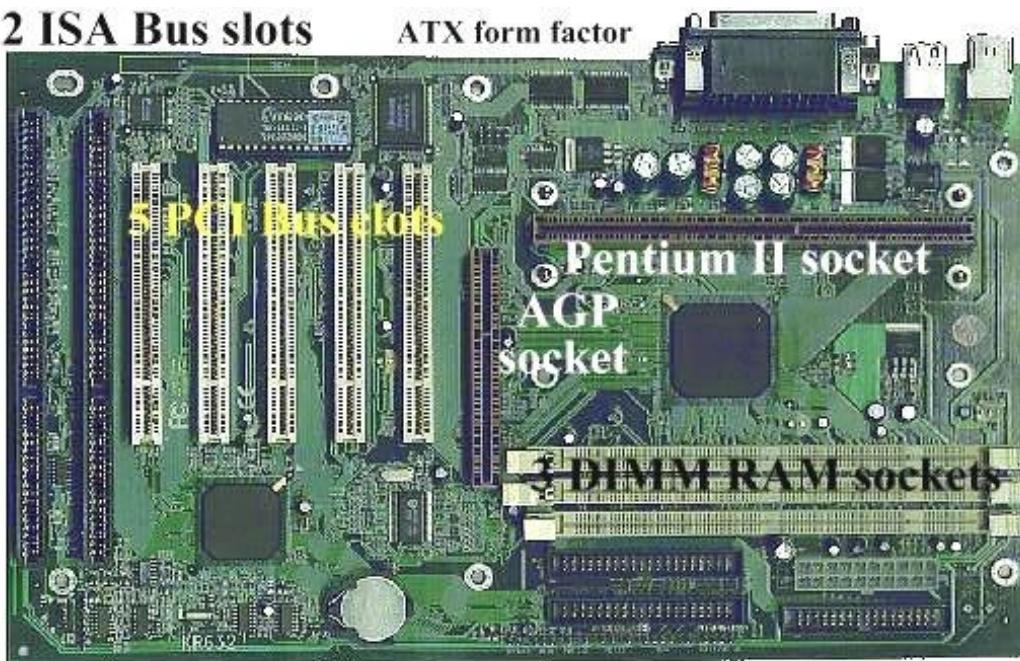
e) El árbitro del bus muestrea las líneas REQ# al comienzo del ciclo 3, y toma la decisión de conceder el bus a B para la siguiente transacción. Baja GNT#-B y sube GNT#-A. B no puede usar el bus hasta que éste esté en estado inactivo.

# Bus AGP

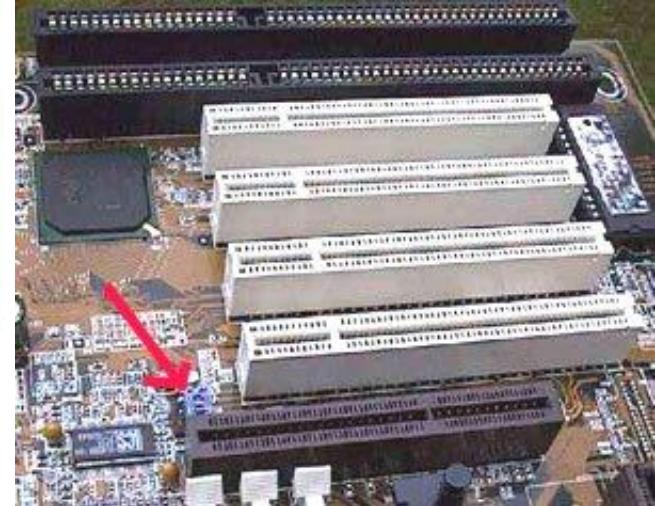
## ■ Bus AGP (Accelerated Graphics Port)

2 ISA Bus slots

ATX form factor



Placas Pentium II con ranura AGP



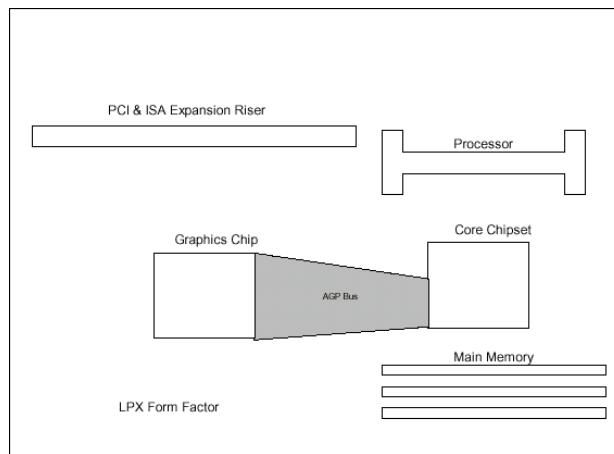
Ranura AGP

# Bus AGP

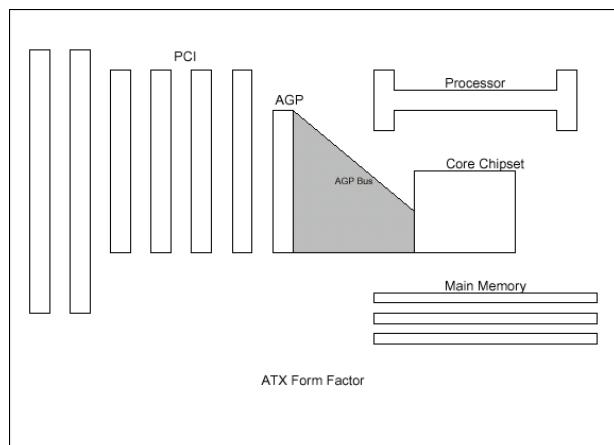
- Introducido por Intel en 1997 junto con el Pentium II.
- Se llama “*Port*” en lugar de “*Bus*” por tener un uso particular, en lugar de ser un slot genérico.
  - Diseñado exclusivamente para **hardware de gráficos**.
- Objetivos:
  - Aumentar el ancho de banda entre procesador-memoria y el sistema de vídeo.
    - Aceleración 3D.
    - Reproducción de vídeo en tiempo real.
  - Liberar el bus PCI del trabajo con gráficos,
    - ✓ lo que permite dedicarlo a otros usos (LAN 100 Mb/s, HD, DVD).

# Bus AGP

- El hardware de gráficos puede encontrarse
  - en la placa madre:

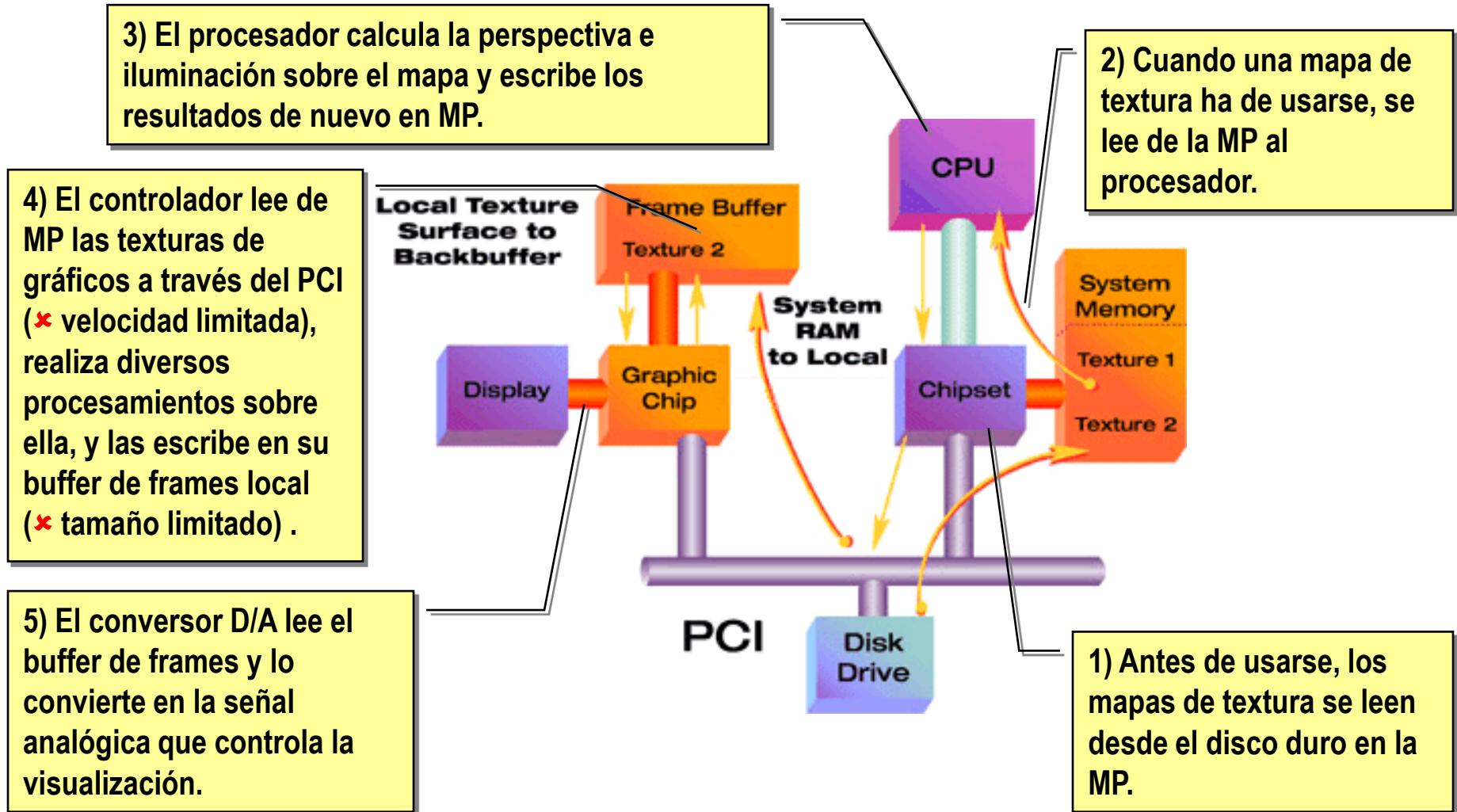


- o en una tarjeta de vídeo (en ranura AGP):



# Bus AGP

- Así funciona un sistema de gráficos 3D basado en PCI:



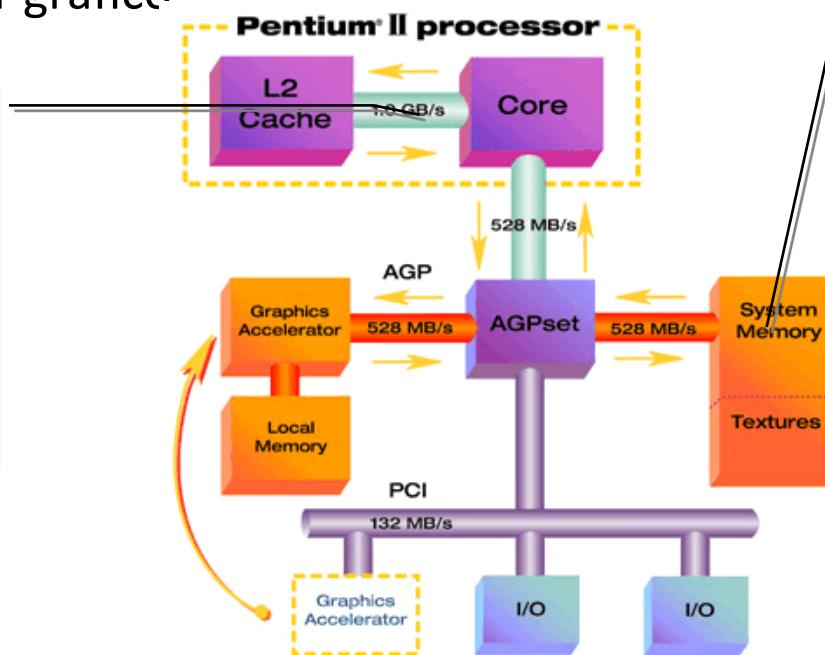
# Bus AGP

- Problemas de este sistema basado en PCI:
  - ✖ Las texturas deben almacenarse tanto en MP como en el buffer de *frames* ⇒ copias redundantes, **uso poco eficiente de memoria.**
  - ✖ Almacenar las texturas en la memoria local supone un **límite al tamaño de las texturas.**
    - Demanda de texturas con un mayor detalle ⇒ **necesidad de una memoria local cada vez mayor.**
    - ✖ Pero la **memoria de vídeo es más cara** que la MP.
  - ✖ El ancho de banda de 132 MB/s del bus PCI **limita la velocidad** a la que los mapas de textura se transfieren al subsistema gráfico.
    - ✖ Además varios dispositivos de E/S en el bus PCI pueden **compartir el ancho de banda disponible.**

# Bus AGP

- AGP es un bus dedicado de alta velocidad entre MP y el controlador gráfico

**Arquitectura Dual Independent Bus (DIB) del Pentium II: dos buses independientes pueden trabajar al mismo tiempo.**



El SO reserva dinámicamente segmentos de MP para ser usados por el controlador gráfico (“memoria AGP” o memoria de vídeo no local).

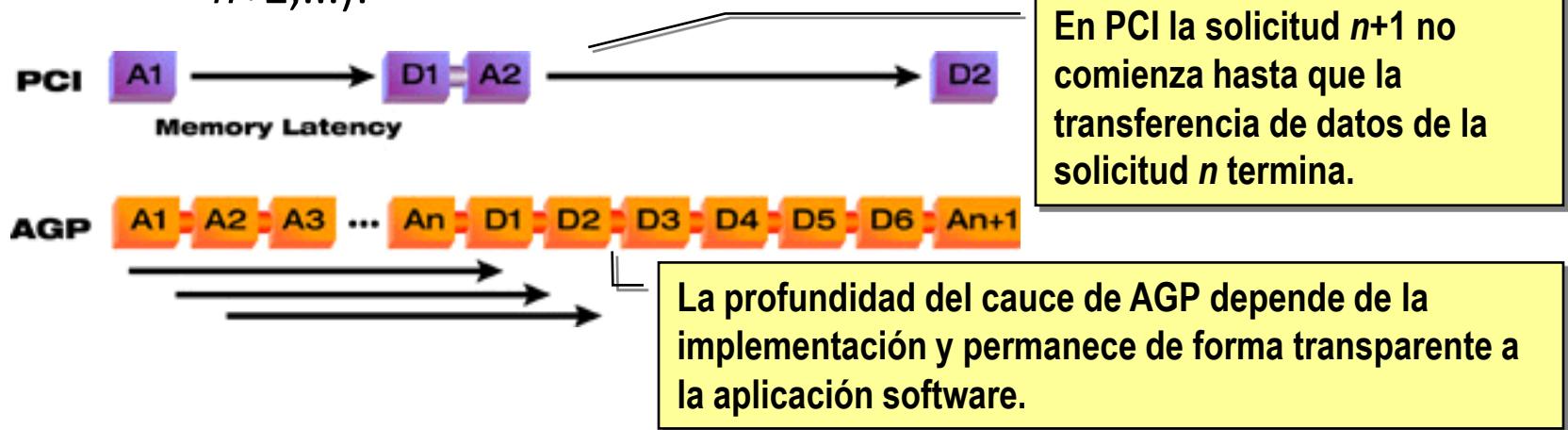
- **Gran ancho de banda** entre el acelerador gráfico y la MP.  
⇒ Permite **DIME** (“*Direct Memory Execute*”)
  - Los aceleradores gráficos usan la MP (además de la memoria de la tarjeta de vídeo) para almacenar texturas utilizadas en 3D y acceden a ellas durante la presentación.

# Bus AGP

- Las texturas más frecuentemente usadas o las menores pueden residir en la memoria local, mientras que **las mayores y menos frecuentemente usadas podrían residir en MP.**
- El controlador gráfico puede **acceder a grandes estructuras de datos**, como mapas de bits de textura (típicamente de 1 KB a 128 KB), **como una única entidad**.
  - El hardware **GART** (“*Graphics Address Remapping Table*”), incorporado en el chipset AGP, traduce direcciones virtuales a físicas de MP de manera similar al paginado hardware del procesador.
- Los **accesos del procesador a MP** pueden realizarse **concurrentemente con lecturas de memoria AGP** por parte del controlador gráfico.
- AGP opera concurrente, e **independientemente**, con las transacciones PCI.

# Bus AGP

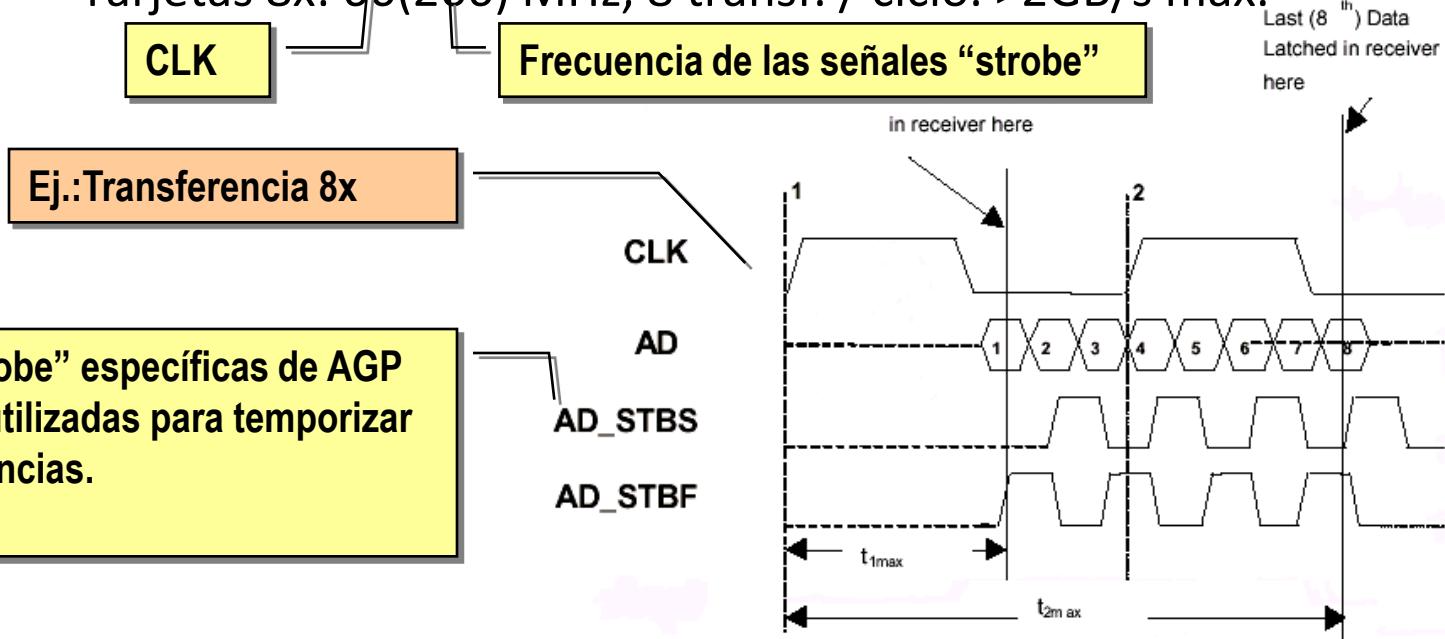
- Basado en PCI 2.1 a 66 MHz (32 bits), con tres extensiones:
  - 1 Operaciones de lectura/escritura en memoria segmentadas.
    - Se solapan los tiempos de acceso a la memoria para una solicitud  $n$  con la emisión de las siguientes solicitudes ( $n+1, n+2, \dots$ ):



- 2 Demultiplexación de direcciones y datos en el bus.
- 3 Se transmiten datos en los flancos de subida y de bajada del reloj (excepto en modo 1x).

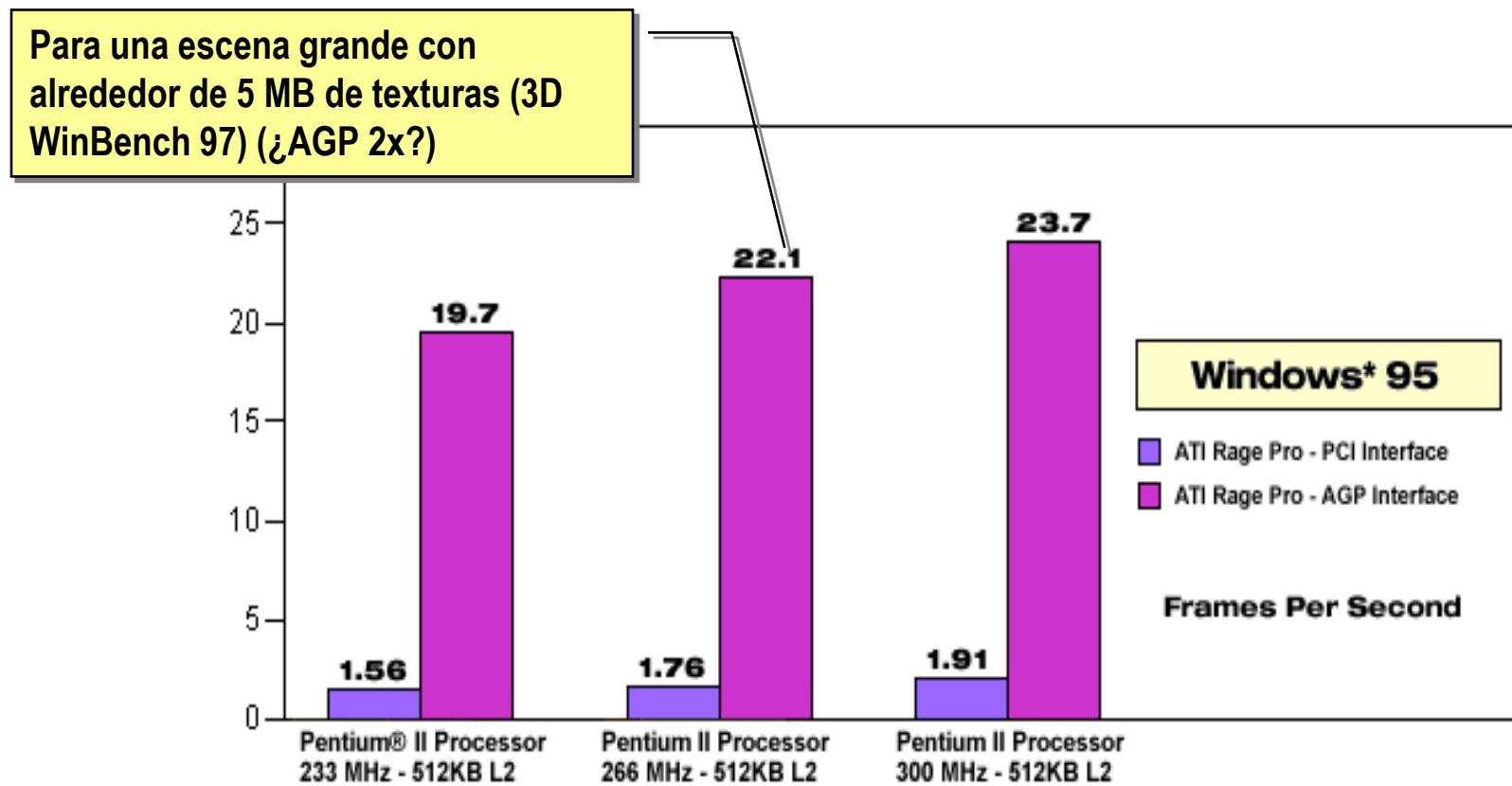
# Bus AGP

- Velocidad de transferencia pico teórica de 528 MB/s entre la MP y el acelerador de gráficos, frente a 132 MB/s de PCI.
  - Tarjetas 1x: 66 MHz, 1 transf. / ciclo, 264 MB/s máx.
  - Tarjetas 2x: 66(66) MHz, 2 transf. / ciclo, 528 MB/s máx.
  - Tarjetas 4x: 66(133) MHz, 4 transf. / ciclo, >1GB/s máx.
  - Tarjetas 8x: 66(266) MHz, 8 transf. / ciclo. >2GB/s máx.



# Bus AGP

- En la práctica se obtienen velocidades alrededor del 50-80% de los valores máximos.
  - ¡Procesador y E/S también tienen que acceder a memoria!
- Comparación de prestaciones PCI/AGP.



# Buses paralelo para periféricos

## ■ ATA:

- IDE.
- EIDE.
- Ultra DMA-33,66,100,133.

## ■ SCSI:

- SCSI-1.
- SCSI-2.
- SCSI-3.

# ATA



Western Digital  
Compaq



ANSI X3.221-1994

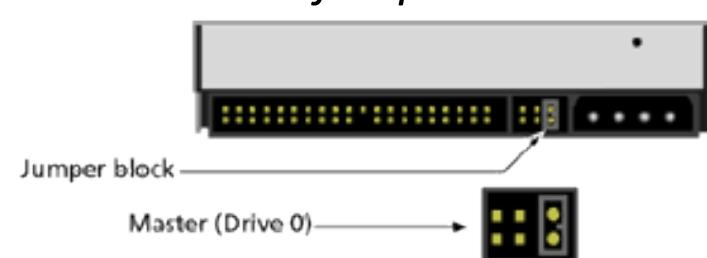
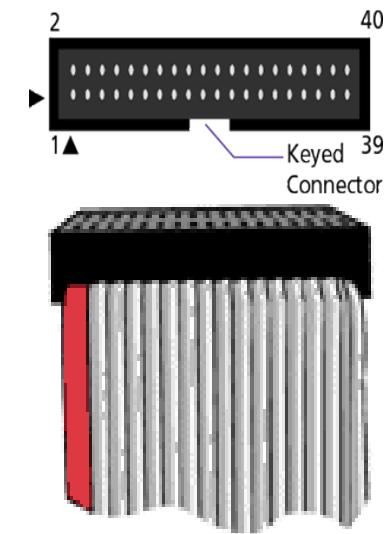
## ■ Bus IDE (*Integrated Device Electronics*), o ATA (*AT Attachment*)

<http://www.ata-atapi.com>

- Interfaz de conexión de **discos duros** (únicamente).
- ✓ Bajo coste.
- ✓ Sencillez de instalación y configuración.
- Hasta 4,1 MB/s en transferencia mediante DMA.
- Controlador disponible: **en la placa madre**, o **en una tarjeta** en el bus de expansión.
- “*Integrated Device Electronics*”
  - ≡ El controlador de disco forma parte del dispositivo
    - antes de IDE el controlador estaba en la tarjeta y la transmisión (analógica) era más sensible a ruidos y a interferencias.
- ATA (“*AT Attachment*” ≡ “Accesorio (conexión) tipo AT”):
  - estándares ANSI para solucionar problemas de compatibilidad encontrados a la hora de conectar dos dispositivos IDE de distintos fabricantes a un mismo bus.

# ATA

- Un disco IDE se conecta (a la placa madre o a la tarjeta controladora) usando un conector plano (“flat ribbon”)
  - 40 hilos (con el número 1 coloreado).
  - Longitud máxima = 18”
- Permite conectar **dos discos duros**.
  - Tres conectores en el cable:
    - uno en cada extremo,
    - otro entre ambos (distancia no simétrica).
      - » los dos más cercanos se suelen conectar a los discos.
  - Se llaman **principal** o **maestro**, y **secundario** o **esclavo**, aunque su única diferencia es la prioridad en el bus.
    - Los discos deben configurarse cambiando los *jumpers* o puentes que los acompañan.



# ATA

- Más que inventar una nueva interfaz, las señales del cable **IDE simplemente duplican la actividad del bus ISA.**
    - 16 bits de datos.
  - Soporte lógico en servicios del BIOS (INT 13h)
    - Transparente a las aplicaciones.
    - Discos de tamaño  $\leq$  504 MB
      - debido al diseño original del BIOS que utilizaba direccionamiento físico CHS (*Cylinder Head Sector*), y limitaba
        - » el número de cilindros a 1024
        - » el número de cabezas a 16
        - » el número de sectores por pista a 63 (512 B/sector)
- $$16 * 1024 * 63 * 512 = \mathbf{528} \text{ } 482 \text{ } 304 \text{ } (/2^{20} = 504 \text{ M})$$

# ATA

Western Digital

ANSI X3.279-1996

Seagate,  
Quantum

## ■ Bus EIDE (Enhanced IDE), ATA-2, o Fast-ATA

- IDE con varias extensiones:

- Nuevos modos de transferencia más rápidos (hasta 16 MB/s).
- Permite conectar **4 dispositivos**: discos duros, cintas, lectores de CD-ROM y DVD, grabadoras de CD y DVD.
  - A través de **dos interfaces o canales** separados (actúa básicamente como dos controladores IDE: **IDE primario** e **IDE secundario**), conectando dos dispositivos en cada uno mediante un cable IDE.



# ATA

- Cada canal IDE pide su propia IRQ
  - » normalmente IDE primario=IRQ 14, IDE secundario=IRQ 15.

Cuando se instala un disco duro y una unidad de CD-ROM, es importante, de cara a las prestaciones, conectarlos en dos canales IDE diferentes:

cuando se envía una orden al CD-ROM (lento en comparación con el disco duro), el otro dispositivo del mismo canal queda bloqueado, tiene que esperar hasta que la orden haya terminado.



# ATA

- Gestión de discos duros de más de 504 MB.
  - LBA (*Logical Block Addressing*)
    - » Dirección lógica de 28 bits vista como un único número para especificar el sector. Límite teórico de 128 GB.

$$2^{28} * 512 = 2^{37} = \mathbf{137\ 438\ 953\ 472} \quad (/2^{30} = 2^7 = 128 \text{ GB})$$

- » La mayoría de las BIOS sólo utilizan 24 de los 28 bits.  
Hasta 7,8 GB.

$$255 \text{ (cabezas)} * 1024 * 63 * 512 = \mathbf{8\ 422\ 686\ 720} \quad (/2^{30} = 7,8 \text{ G})$$

- Nueva orden “*Identify Drive*” que permite al software preguntar al disco por su geometría y características (Plug ‘n’ Play).
- Compatibilidad con IDE:
  - la misma especificación de señales en los cables.

# ATA

## ■ ATA-3

- Pequeña revisión de ATA-2 que mejora la fiabilidad de las transferencias más rápidas introducidas en ATA-2.
  - Integridad de datos mediante CRC (*Cyclical Redundancy Check*)
    - Para cada bloque el “host” y el controlador de disco calculan el CRC y lo almacenan en sendos registros.
    - Al final de la transmisión del bloque, el “host” envía el contenido de su registro CRC al controlador de disco, que lo compara con el suyo.
    - Si el controlador informa de error, el “host” reintenta la orden que lo produjo.

# ATA

ANSI NCITS.317-1998

## ■ Bus Ultra DMA-33, Ultra-ATA-33, o ATA/ATAPI-4

- Se añaden y modifican:
  - Órdenes ATAPI (*ATA Packet Interface*).
    - Subconjunto de órdenes SCSI a nivel lógico,
    - para dispositivos como CD-ROM, DVD, unidades de cinta,
    - y dispositivos desmontables como ZIP o JAZZ.
  - Cambia muchas órdenes y características ATA obsoletas.
  - Añade un protocolo de transferencia denominado “Ultra DMA”
    - con velocidades de transferencia mayores (hasta 33 MB/s).
  - “*Overlapped feature set*” permite que un dispositivo lento libere el bus de manera que lo use otro dispositivo mientras el primero concluye una operación.
  - Posibilidad de usar un cable de 80 hilos (no obligatorio).

# ATA

## ■ Bus Ultra DMA-66, Ultra-ATA-66, o ATA/ATAPI-5

- Añade modo Ultra DMA/66
  - 66 MB/s máx.

## ■ Bus Ultra DMA-100, Ultra-ATA-100, o ATA/ATAPI-6 (2001)

- Se incrementa tamaño de LBA de 28 a 64 bits (límite teórico de  $2^{33}$  TB), y número de sectores de una transferencia pasa de usar 8 a 16 bits.

$$2^{64} * 512 = 2^{73} = (/2^{40} = 2^{33} \text{ T})$$

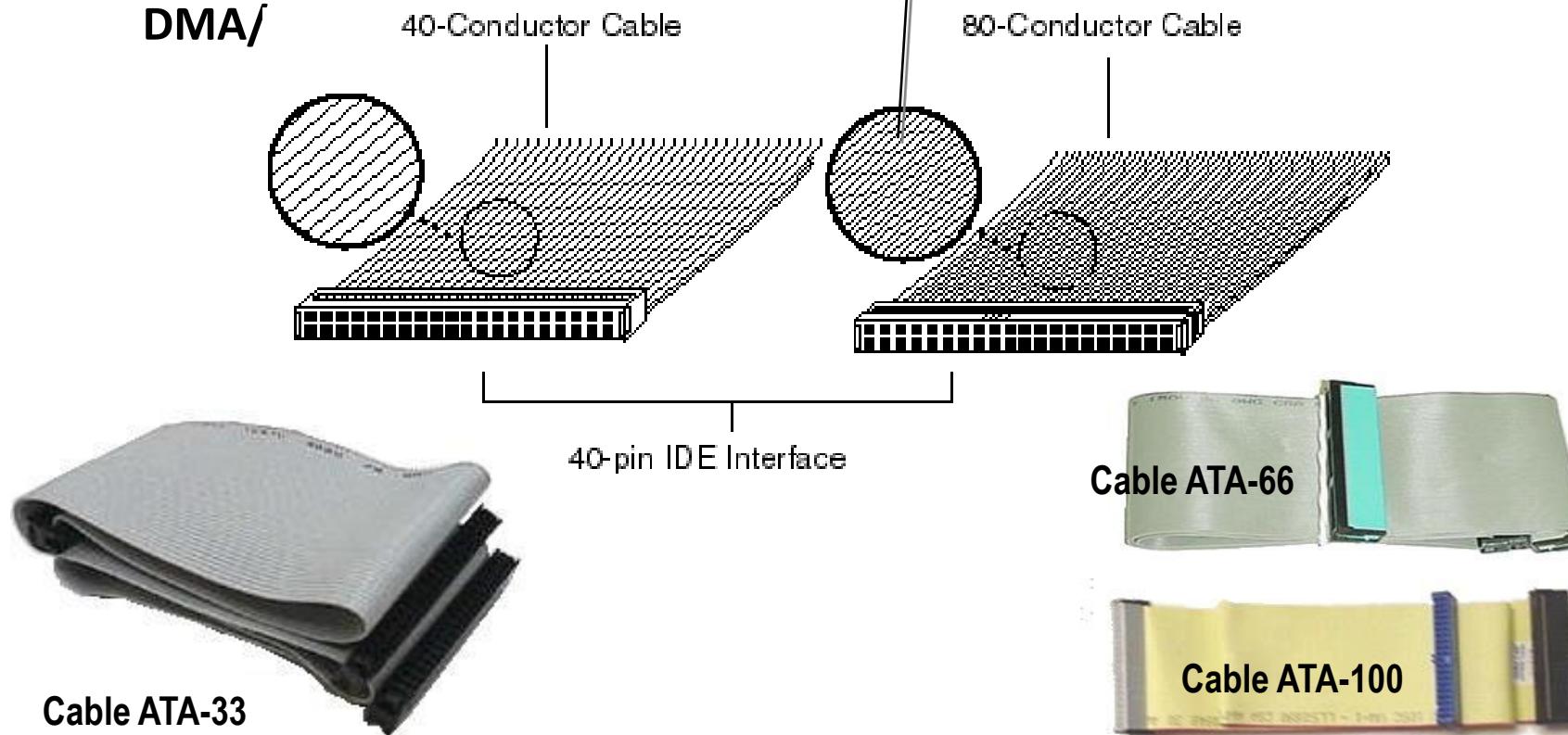
- Se añade modo Ultra DMA/100
  - 100 MB/s máx.

## ■ Bus Ultra DMA-133, Ultra-ATA-133, o ATA/ATAPI-7 (2002)

# ATA

- Obligatoriedad a partir de ATA-66 de utilizar cable de 80 hilos para los modos más rápidos que Ultra DMA/

Para poder manejar tiempos de ciclo menores, se incluyen líneas de tierra que sirven de apantallamiento reduciendo el "crosstalk".



# ATA

## ■ Resumen de modos de transferencia:

### ① PIO (*Programmed Input Output*):

- **El procesador interviene en la transferencia de datos** desde (o hacia) el dispositivo de almacenamiento hacia (o desde) la MP a través de puertos de E/S.
- Hoy en desuso.

Modos PIO	Velocidad transf. (MB/s)	Especificación
0	3,3	ATA-1
1	5,2	ATA-1
2	8,3	ATA-1
3	11,1	ATA-2
4	16,6	ATA-2

# ATA

## ② DMA (*Direct Memory Access*):

- El trabajo de transferir los datos es encargado a un **controlador del sistema con “bus-mastering” (controlador DMA)**.
- El **procesador sólo ha de inicializar** (programar) el controlador DMA, lo que ahorra tiempo de CPU.
- La principal diferencia entre Ultra DMA y los antiguos modos “*singleword*” y “*multiword*” es que **en el modo Ultra DMA se transfieren datos dos veces por ciclo de reloj** (en los flancos de bajada y de subida) doblando el ancho de banda.

Modos DMA	Velocidad transf. (MB/s)	Especificación
Single Word 0	2,1	ATA-1
Single Word 1	4,2	ATA-1
Single Word 2	8,3	ATA-1
Multi Word 0	4,2	ATA-1
Multi Word 1	13,3	ATA-2
Multi Word 2	16,7	ATA-2
Ultra DMA 0	16,7	ATA/ATAPI-4
Ultra DMA 1	25,0	ATA/ATAPI-4
Ultra DMA 2 o UDMA33	33,3	ATA/ATAPI-4
Ultra DMA 3	44,4	ATA/ATAPI-5
Ultra DMA 4 o UDMA66	66,7	ATA/ATAPI-5
Ultra DMA 5 o UDMA100	100,0	ATA/ATAPI-6

# ATA

## ■ Resumen de especificaciones ANSI ATA:

	ATA o ATA-1	ATA-2/3	ATA-4	ATA-5	ATA-6
<b>Alias</b>	IDE	EIDE Enhanced IDE Fast ATA-2	ATAPI Ultra ATA-33 Ultra DMA-33	Ultra ATA-66 Ultra DMA-66	Ultra ATA-100 Ultra DMA-100
<b>Velocidad transf. máx.</b>	8,3 MB/s	16,7 MB/s	33,3 MB/s	66,7 MB/s	100 MB/s
<b>Conexión</b>	Cable y conector de 40 contactos	Cable y conector de 40 contactos	Cable y conector de 40 contactos	Cable de 80 hilos y conector de 40 contactos	Cable de 80 hilos y conector de 40 contactos

# SCSI ◀

## ■ Bus SCSI (Small Computer System Interface)

Pronunciado  
“skuzzy” ('skʌzɪ)  
(≡“cutre” en inglés).



Más tarde Seagate Technology

- SASI (*Shugart Associates Systems Interface*).
  - Interfaz estándar para discos duros.
- SCSI (1981): colaboración entre NCR y Shugart Associates.
- Introducido en Apple Macintosh en 1984.
- SCSI-1: estándar ANSI (1986).
  - ✖ Mayoría de implementaciones no compatibles entre sí.
- SCSI-2: estándar ANSI (1990).
  - Estándar incluso a nivel físico y eléctrico. Plug'n'play.
- SCSI-3.

# SCSI ◀

- Muy extendido, multitud de plataformas y fabricantes.
- Admite **mayor variedad de periféricos** (y de órdenes para controlarlos) que cualquier otro estándar:
  - Discos duros, disquetes, unidades de cinta, RAID (discos redundantes), CD-ROM, unidades ZIP, impresoras, escáneres, ...
- 8 bits (*Narrow SCSI*) / 16 bits (*Wide SCSI*) de datos
- Permiten respectivamente conectar **hasta 8 ó 16 dispositivos** (incluyendo el adaptador o controlador) encadenados a un mismo puerto mediante “**daisy-chain**”.
  - Internos (cable plano “*flat ribbon*”).



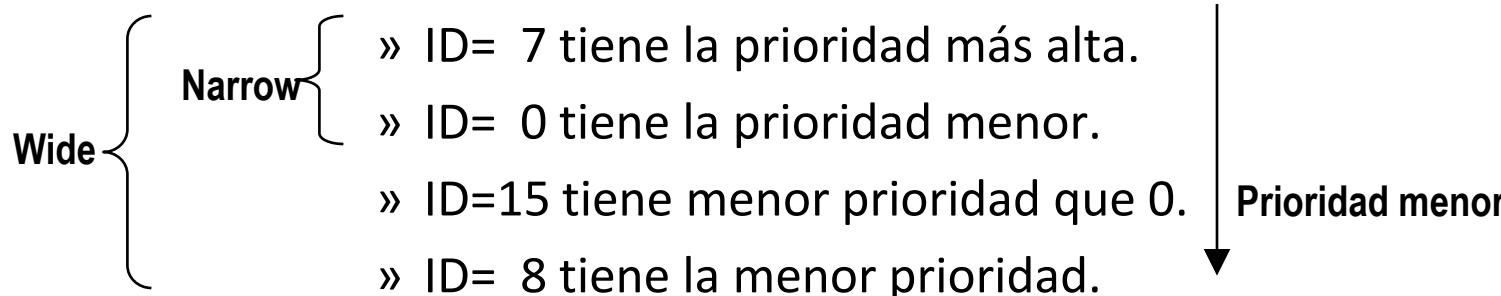
✓ Externos (varios metros de cable).

# SCSI ◀

- Los dispositivos conectados al bus SCSI “hablan” al ordenador a través del controlador o adaptador.
  - es uno de los dispositivos SCSI, en este caso conectado además a uno de los buses de datos del ordenador.
    - Es una **tarjeta** conectada en una ranura (PCI, por ej.),
    - o bien forma **parte de la circuitería de la placa madre**.
- Puede haber varios controladores, para permitir más dispositivos.
- Para cada controlador:
  - Una única línea de interrupción, arbitraje gestionado por el controlador.  
⇒ El ordenador “ve” todos los dispositivos conectados a ese controlador como uno solo.
- Todos los intercambios en SCSI se producen entre un dispositivo iniciador y un dispositivo seleccionado.
  - El iniciador normalmente es el controlador, pero no es obligatorio. Por ejemplo un disco duro puede guardar su contenido en una cinta sin intervención del ordenador.

# SCSI ◀

- Cada dispositivo tiene cableado un número identificador, llamado dirección SCSI o ID.
  - Cada dispositivo se direcciona usando una línea de datos  
⇒ nº bits de datos = nº. máx. de dispositivos.  
Narrow SCSI ⇒ 8 dispositivos, del 0 al 7. Controlador = 7
- Dos alternativas:
  - Fijado manualmente.
  - SCAM (*SCSI Configured Auto-Magically*). Se asigna ID dinámicamente.
- Dos propósitos:
  - **Identifica únicamente** cada dispositivo.
  - **Determina la prioridad** de cada dispositivo.





- Permite transferencias:
  - **Asíncronas:**
    - *Handshake* petición/reconocimiento. Señales de temporización REQ y ACK.
    - Cada byte transmitido necesita un reconocimiento.
  - **Síncronas:**
    - Sincronizadas con una señal de reloj.
    - Se pueden hacer varias peticiones antes de recibir los reconocimientos.
    - $\approx 3$  veces más rápidas que asíncronas.
      - » SCSI1: 2 MB/s  $\rightarrow$  5 MB/s
      - » SCSI2: 3 MB/s  $\rightarrow$  10 MB/s

# SCSI ◀

- Permite transferencias:
  - **PIO (Programmed Input/Output):**
    - Con la intervención del procesador .
    - Sólo un dispositivo SCSI puede transmitir.
    - Lenta comparada con *bus mastering*.
  - **Bus Mastering:**
    - El controlador incorpora un DMA para manejar transferencias por DMA, sin intervención del procesador.
    - Permite transferencias
      - » dispositivo a dispositivo.
      - » dispositivo a memoria.
      - » memoria a dispositivo.
    - Varios dispositivos pueden operar simultáneamente.



- Principales fases en el bus:
  - **Bus libre:** Ningún dispositivo está utilizando el bus.
  - **Arbitraje:** Permite que un dispositivo (iniciador) tome el control del bus, de manera que pueda iniciar o reanudar un proceso de E/S.
  - **Selección:** El iniciador selecciona un dispositivo para realizar una operación, por ejemplo una lectura o una escritura.
- seguidas de varias secuencias de:
  - **Orden:** El dispositivo seleccionado solicita orden al iniciador.
  - **Datos:** Se envían / reciben los datos.
  - **Estado:** Se envía información de estado indicando que se ha completado la transferencia desde el dispositivo seleccionado al iniciador.
  - **Mensaje:** Se transfiere un mensaje entre iniciador y seleccionado: orden completada, desconexión, error, abortar, iniciar transferencia síncrona.



- Transferencias partidas / *multi-threading*:
  - Los dispositivos comparten el ancho de banda del bus conectándose sólo cuando tienen datos para enviar.
    - Un dispositivo puede liberar el bus tras recibir una orden, para realizar operaciones que consuman tiempo y no requieran uso del bus.
    - El bus queda libre para cualquier otro dispositivo hasta que el primero esté listo y se vuelva a conectar para terminar la orden.
  - Soporte software desde Windows 95 en adelante.



- Encolado y reordenación de órdenes (introducidos en SCSI-2):
  - En SCSI-1 o IDE/ATA sólo se puede enviar una única orden a la vez a un dispositivo.
    - Si una orden se ha enviado a un dispositivo, las siguientes no se pueden enviar hasta que no finalice la primera.
  - El **encolado de órdenes** permite aceptar hasta 256 órdenes, incluso provenientes de diferentes dispositivos.
  - La **reordenación de órdenes** permite a un dispositivo que tenga órdenes pendientes, terminarlas “sin orden”, es decir, en un orden distinto al que fueron recibidas.
  - Tienen sentido en servidores multitarea / multidispositivo.
    - Se permite aceptar peticiones simultáneas de distintos usuarios y completarlas del modo más eficiente.
    - Por ej. si en un disco las órdenes se procesan según se reciben, puede perderse tiempo cuando los componentes mecánicos se saltan un espacio de datos cercano que sea necesario acceder una o dos órdenes después.

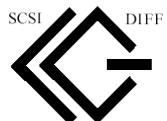
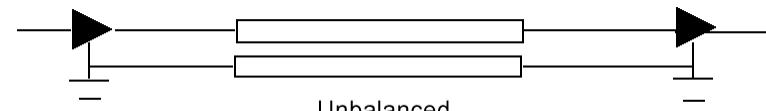
# SCSI




- Especificaciones eléctricas:

- **Single-Ended (SE)** o no balanceada:

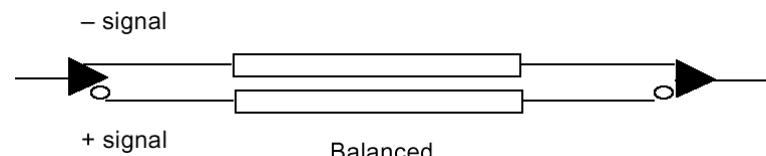
- Para distancias cortas ( $\leq 6 \text{ m}$ ).
    - Para cada señal (1 bit) hay un cable.
    - El receptor detecta el voltaje respecto a tierra.
      - » Nivel TTL (0 y 5V).



- **Diferencial** (balanceada) o **HVD (High Voltage Differential)** (IEEE485):

- Para mayores distancias ( $\leq 25 \text{ m}$ ) o mayor velocidad.
    - Para cada señal (1 bit) hay una pareja de cables:
      - » uno lleva la misma señal que en *single-ended*.
      - » el otro lleva la negación lógica.
    - El receptor calcula la diferencia de los dos cables.
      - » Si  $(V+) - (V-) > 0 \Rightarrow 1 \text{ lógico}$ .
      - » Si  $(V+) - (V-) < 0 \Rightarrow 0 \text{ lógico}$ .

$\Rightarrow$  Inmunidad al ruido ↑.



# SCSI ◀



- **LVD (*Low Voltage Differential*)**

- Diferencial, pero a un voltaje menor.
  - »  $(V+) - (V-) > 60 \text{ mV} \Rightarrow 1 \text{ lógico.}$
  - »  $(V+) - (V-) < -60 \text{ mV} \Rightarrow 0 \text{ lógico.}$
- ✓ Permite integrar el “*transceiver*” en el chip controlador.



- **Multimodo LVD/SE**

- Un dispositivo multimodo conmuta automáticamente entre LVD y SE cuando detecta otro dispositivo en la cadena funcionando en LVD o SE.
  - » Si hay algún dispositivo SE, todos conmutan a SE.

# SCSI ◀

- Conectores:

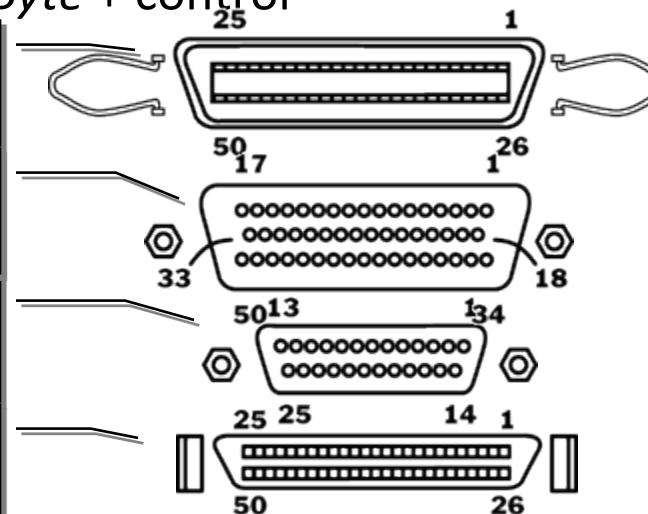
- **Narrow:** 50 hilos: *Low Data Byte + control*

**Conecotor Centronics de 50 patillas (5 MB/s)**

**Conecotor de 50 patillas de antiguas SUN y DG**

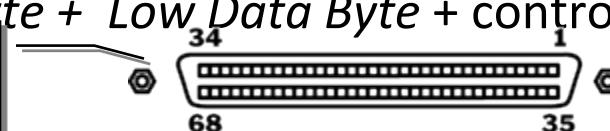
**Conecotor DB-25 (Apple Mac y algunas SUN antiguas)**

**Conecotor de alta densidad de 50 patillas (10 MB/s)**



- **Wide:** 68 hilos: *High Data Byte + Low Data Byte + control*

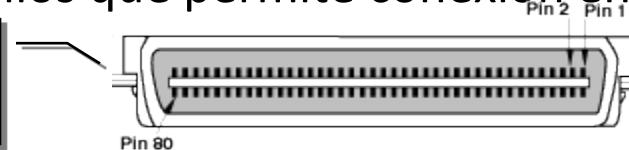
**Conecotor de alta densidad de 68 patillas**



- **SCA (Single Connector Assembly):**

- Combina en un único cable conector SCSI y alimentación.
- SCA-2: Conector de 80 hilos que permite conexión en caliente.

**Conecotor SCA-2 de 80 patillas**



# SCSI ◀

- Requiere terminadores (el primero y el último en la cadena):
  - Evitan la reflexión causante de errores.
  - Diferentes tipos:
    - internos / externos.



- *single-ended* / diferenciales.
- pasivos / activos.
  - » pasivos: para cada señal se usa un divisor de tensión (pareja de resistencias). Adecuados para SCSI-1 a 5 MHz.
  - » activos: resistencias conectadas a una fuente de alimentación. Consiguen menor reflexión ⇒ para mayor distancia y/o velocidad (obligatorio en SCSI-3).



## Evolución de los estándares SCSI:

Todos mantienen compatibilidad hacia atrás.

- **SCSI-1** (1986):
  - ≤ 8 dispositivos (principalmente discos).
  - Longitud cable SE=6 m, HVD=25 m.
  - **8 bits** de datos, 5 MHz⇒**5 MB/s** pico.
- **SCSI-2** (1990):
  - Mejoras en el conjunto de órdenes para dispositivos ≠ discos.
  - Posibilidad de encolar hasta 256 órdenes para cada dispositivo.
  - Longitud cable SE=3 m, HVD=25 m.
  - **Fast SCSI**:
    - ≤ 8 dispositivos
    - **8 bits, 10 MHz, 10 MB/s.**
- **Fast Wide SCSI**:
  - ≤ 16 dispositivos.
  - Requiere conector de alta densidad de 68 contactos.
  - **16 bits, 10 MHz, 20 MB/s.**
  - Sin embargo las órdenes, estado y arbitraje se siguen transmitiendo a 8 bits por compatibilidad.

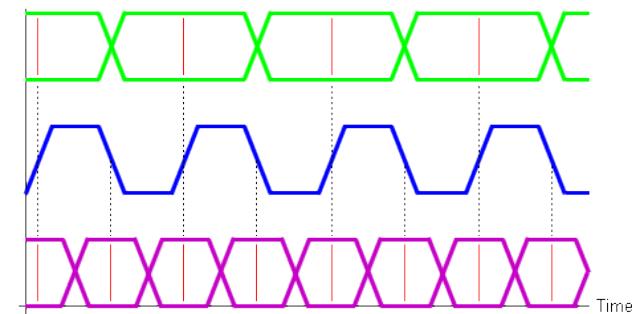
# SCSI ◀

- **SCSI-3:**

- Familia de estándares presentes y futuros.
  - Paralelo y serie, cables más largos, autoconfiguración de las direcciones de dispositivos (SCAM), adopción de nuevas tecnologías como fibra óptica.
- **Ultra SCSI o Fast20 SCSI (1996):**
  - $\leq 8$  dispositivos.
  - Longitud cable SE=1.5 m / 3 m ( $\leq 4$  dispositivos), HVD=25 m.
  - 8 bits, 20 MHz, 20 MB/s.
- **Wide Ultra SCSI:**
  - $\leq 16$  dispositivos.
  - Longitud cable SE=1.5 m ( $\leq 8$  dispositivos) / 3 m ( $\leq 4$  dispositivos), HVD=25 m.
  - 16 bits, 20 MHz, 40 MB/s.
- **Ultra2 SCSI o Fast40 SCSI:**
  - $\leq 8$  dispositivos.
  - Longitud cable HVD=25 m, LVD=12 m.
  - 8 bits, 40 MHz, 40 MB/s.

# SCSI ◀

- **Wide Ultra2 SCSI:**
  - $\leq 16$  dispositivos.
  - Longitud cable HVD=25 m, LVD=12 m.
  - 16 bits, 40 MHz, 80 MB/s.
- **Ultra3 SCSI o Ultra160 SCSI:**
  - $\leq 16$  dispositivos.
  - Longitud cable LVD=12 m.
  - 16 bits, 40 MHz doble transferencia.
    - » En los flancos de subida y bajada:
  - 160 MB/s.
- **Ultra320 SCSI:**
  - $\leq 16$  dispositivos.
  - Longitud cable LVD=12 m.
  - 16 bits, 80 MHz doble transferencia, 320 MB/s.
- **Ultra640 SCSI:**
  - 640 MB/s.



# SCSI ◀

- Ventajas de SCSI sobre IDE/EIDE:
  - ✓ Mayor velocidad (>frecuencia, transfer. partidas, encolado de órdenes).
  - ✓ Mayor fiabilidad.
  - ✓ Mayor número de dispositivos conectables: 7 ó 15 frente a 4 en EIDE.
  - ✓ Más tipos de dispositivos conectables: Escáneres, impresoras, ...
  - ✓ Posibilidad de conectar dispositivos externos (EIDE sólo internos).
  - ✓ Nuevas tecnologías se desarrollan en primer lugar con interfaz SCSI.
- Desventajas:
  - ✗ Precio alto de las tarjetas adaptadoras (IDE es estándar en los chipset Pentium, pero hay que comprar una tarjeta controladora SCSI por separado).
  - ✗ Precio alto de las unidades (complejidad del controlador de dispositivo para admitir protocolos complejos).
- En comparación con ATA-4/5:
  - No es ventajoso en un PC normal.
  - Sí cuando se desean conectar muchos dispositivos (servidor).

# Buses serie para periféricos



**USB.**



**IEEE 1394.**



## (Universal Serial Bus)

- Desarrollado por Compaq, DEC, IBM, Intel, Microsoft, NEC y Northern Telecom.
- Se espera que en breve reemplace completamente a los puertos serie y paralelo en PCs y Mac.
- Permite conectar encadenados hasta 127 dispositivos

Por ejemplo se puede conectar el ratón al teclado, el teclado al monitor y el monitor al ordenador





- Velocidad

- 1.5 ó 12 Mb/s
    - Puerto serie ↗150 Kb/s
    - Puerto paralelo ↗1.2 Mb/s



Dispositivos de baja velocidad, interactivos (10-100 Kb/s)

- teclado, ratón, monitor (configuración), periféricos de juegos
  - Transmisión USB a 1.5 Mb/s (*Low-speed*)



Dispositivos de media velocidad (500 Kb/s - 10Mb/s)

- módem, escáneres, cámaras digitales.
  - Transmisión USB a 12 Mb/s (*Full-speed*)



NO dispositivos de alta velocidad (25-500 Mb/s)

- vídeo digital
  - (sí en USB 2.0: 480 Mb/s).



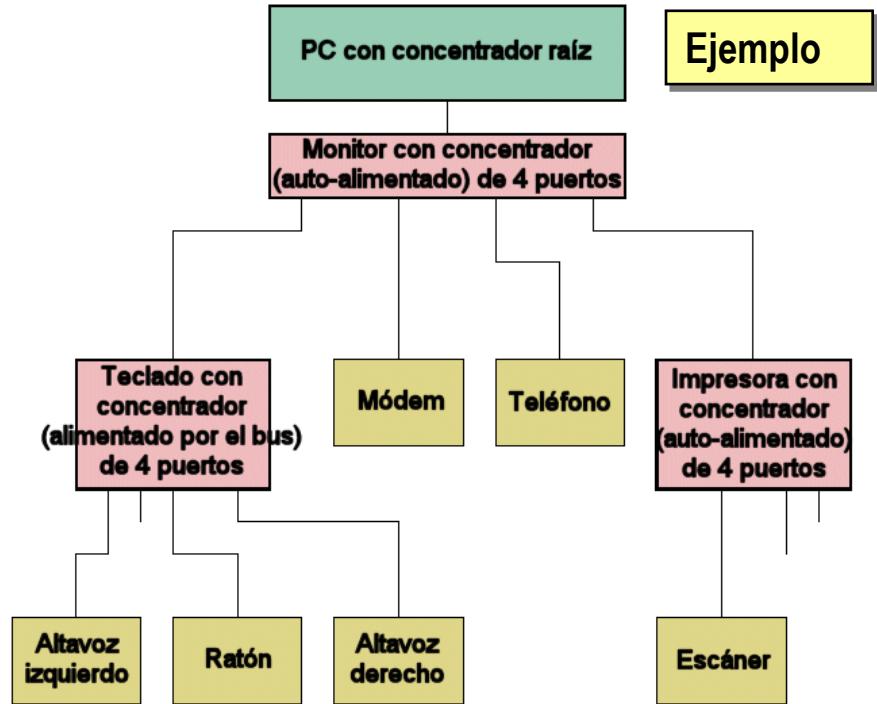
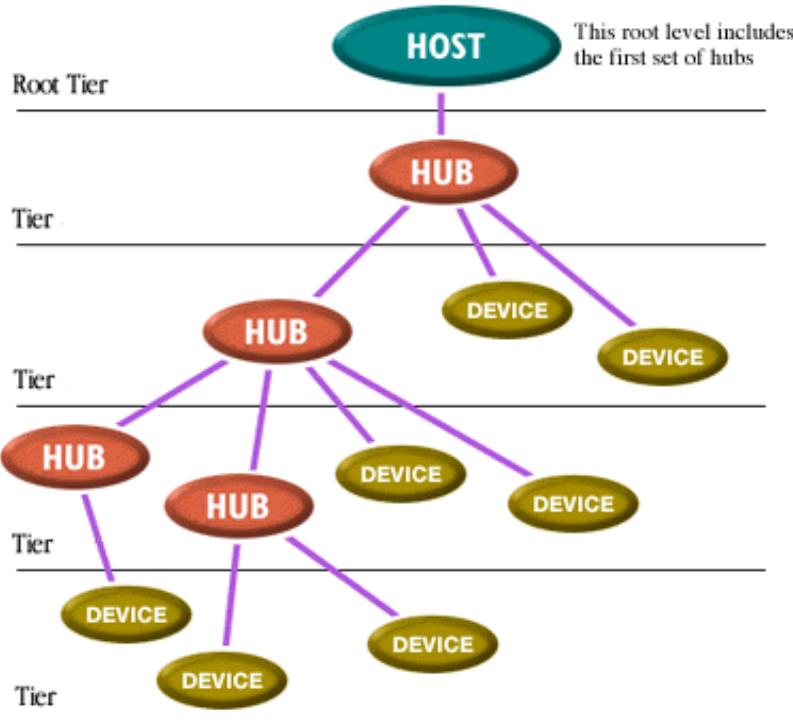
- Clases de dispositivos y topología.



- La especificación USB define cinco clases de dispositivos:
  - Concentradores (*hubs*):
    - ① alimentados por el bus.
    - ② auto-alimentados.
  - Funciones o dispositivos:
    - ① de bajo consumo, alimentados por el bus.
    - ② de alto consumo, alimentados por un concentrador.
    - ③ auto-alimentados.



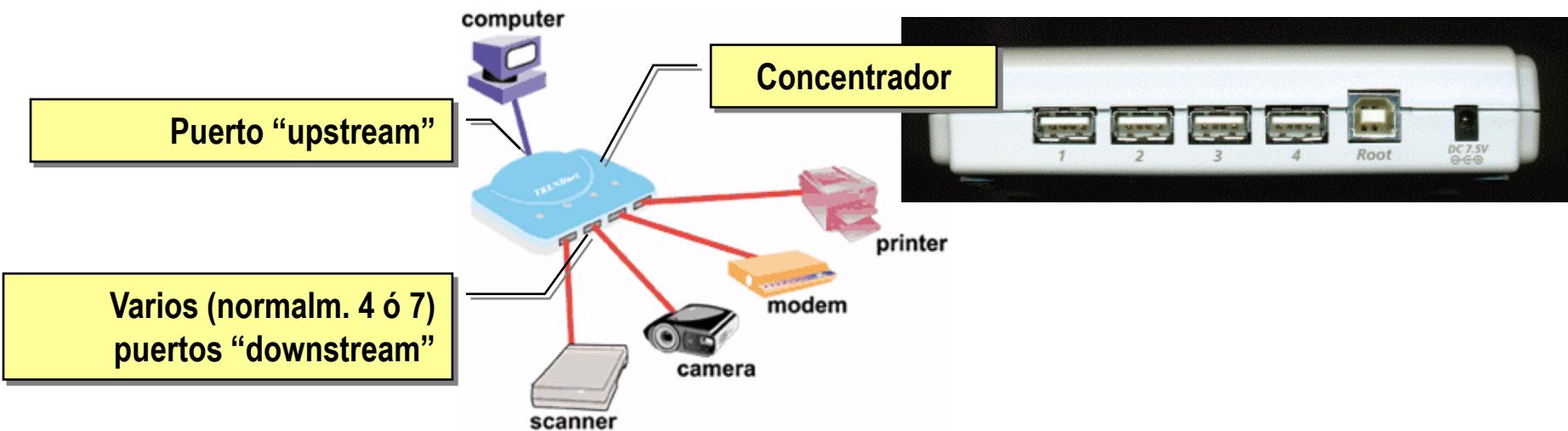
- Se pueden conectar con una topología en estrella de hasta 6 capas lógicas (*tiers*), incluyendo la capa raíz.
  - Sólo se pueden conectar hasta 127 dispositivos al mismo tiempo (incluyendo el concentrador del *host* y los otros concentradores).





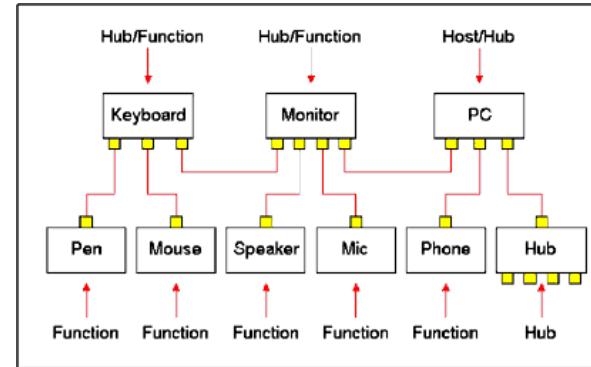
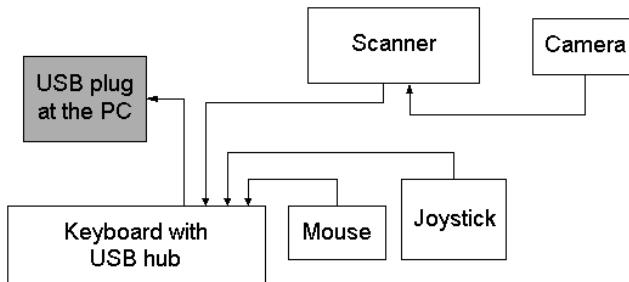
- Concentradores (*Hubs*):

- Permiten expandir el número de dispositivos conectables.
- Conectan
  - un puerto “*upstream*” (“río arriba”), que se conecta directamente al ordenador o a otros concentradores,
  - con varios puertos “*downstream*” (“río abajo”).





- Pueden ser:
  - alimentados por el bus.
    - 100 mA +5V en cada puerto *downstream*.
    - ⇒máximo 4 puertos *downstream*.
  - auto-alimentados (por una fuente de alimentación externa, por ej. un adaptador AC).
    - 500 mA +5V en cada puerto *downstream*.
- Pueden ser:
  - aislados o independientes,
  - o integrados en perif. USB ("dispositivos compuestos")
    - el controlador los reconoce como concentradores con una función conectada.





- Funciones:

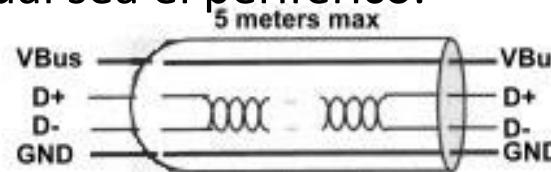
- Dispositivos que se conectan al bus (ratón, teclado, altavoces, conexión RDSL, etc).
- Sólo pueden conectarse a concentradores.
- Cada función se subdivide en “terminaciones” (*endpoints*)
  - Cada terminación se refiere a algún tipo de operación específica que puede implementar el dispositivo USB.
    - Ej. Impresora + fotocopiadora + fax.
  - Existen terminaciones de entrada y de salida, y puede haber un total de 16 terminaciones de un tipo o de otro en una misma función.
  - La terminación 0 de entrada y la terminación 0 de salida están reservadas y se usan para que el dispositivo pueda ser configurado tras ser conectado al bus y proporcionar información sobre su estado, respectivamente.



- Cable:

- Un único cable estándar. sea cual sea el periférico.

- 4 hilos:



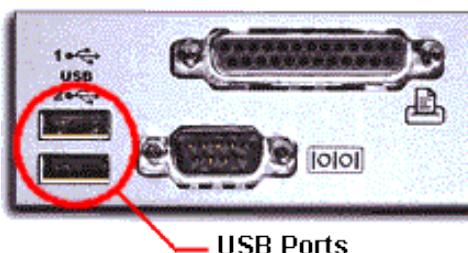
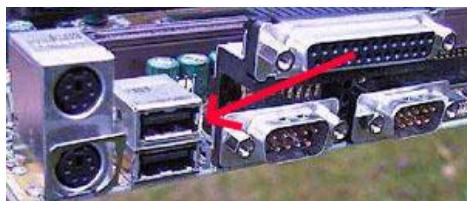
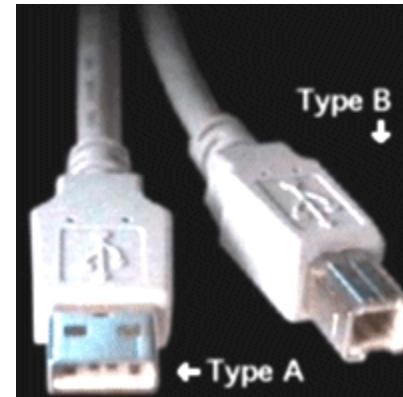
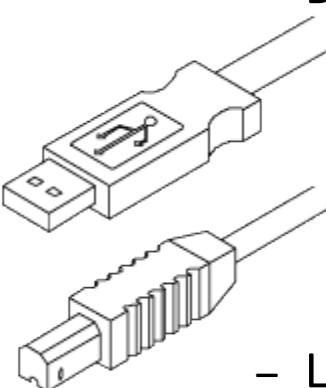
- Par trenzado (señal diferencial D+, D-).
      - Codifican la información junto con el reloj usando NRZI (no retorno a cero invertido). Transición ↗ 0 lógico.
    - El cable también envía alimentación a periféricos de bajo consumo a través de VBus (+5V) y GND.
      - ✖ Problema en los portátiles cuando están alimentados con la batería.
    - Cada segmento de cable es una conexión punto a punto entre el controlador y un concentrador o función, o bien entre dos concentradores, o bien entre un concentrador y una función.
      - 1, 2, 3 y 5 m (máximo).





## ■ Conectores:

- Dos conectores estándar:
  - Tipo A o “*downstream*”.
    - Dirección a los periféricos.
  - Tipo B o “*upstream*”
    - Dirección al ordenador.
- La mayoría de PC tienen un único sistema USB con dos puertos.



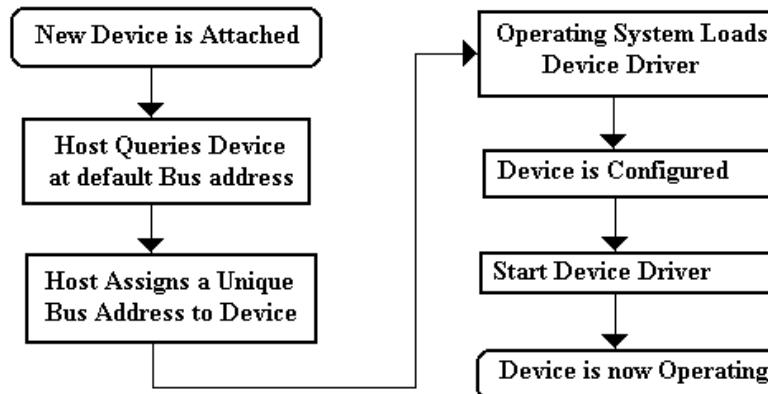
- Si el computador no tiene puertos USB:
  - se puede añadir una tarjeta adaptadora en una ranura PCI (o CARD en portátiles).





- Configuración automática.

- “*Hot-swapping*”: Plug-and-play verdadero.
  - No hay que apagar ni reiniciar para añadir o quitar perifér.
  - Los concentradores detectan cuándo se conecta un nuevo dispositivo a uno de sus puertos *downstream*, notificándolo al controlador y permitiendo configuración.
    - El controlador determina automáticamente el *driver software* necesario.
- Detección automática de velocidad de funcionamiento de cada dispositivo y control automático de transmisión en cada modo.





- Cuatro tipos de transferencia:

- ① Transferencia de control.

- Paquetes de control para configurar los dispositivos cuando se conectan por 1<sup>a</sup> vez, o para fines específicos.

- ② Transferencia de datos “bulk”.

- Método usado con la mayoría de los dispositivos.
    - Para enviar o recibir una gran cantidad de datos de manera secuencial. Ej. Grabadora de CD, escáner.

- ③ Transferencia de datos mediante interrupciones.

- Usado para transmitir unidireccionalmente (desde el dispositivo) muy poca información y poco frecuentemente. Por ej. un ratón, un teclado, o una palanca de juegos necesitan el bus en un momento dado para una transferencia breve.



④ Transferencia de datos isócrona.

crinos = tiempo

Iso = igual

- Flujo unidireccional (desde el controlador o desde el dispositivo) sin reconocimiento, con velocidad de envío = a la de recepción.
- Para dispositivos multimedia de tiempo real (audio y vídeo). Por ej. para teléfono, altavoces o micrófonos.



- Protocolo de transmisión.

- Cada transferencia de datos comprende tres fases:

- ① Paquete testigo (*token*):

- El controlador envía a todos (*broadcast*) un paquete que describe si es una operación de entrada, de salida, o de configuración, la dirección del dispositivo USB, y el número de terminación que debe intervenir.
      - El periférico cuya dirección (asignada en el momento del arranque o de la conexión) coincida, puede disponer del bus y aceptar o enviar datos.

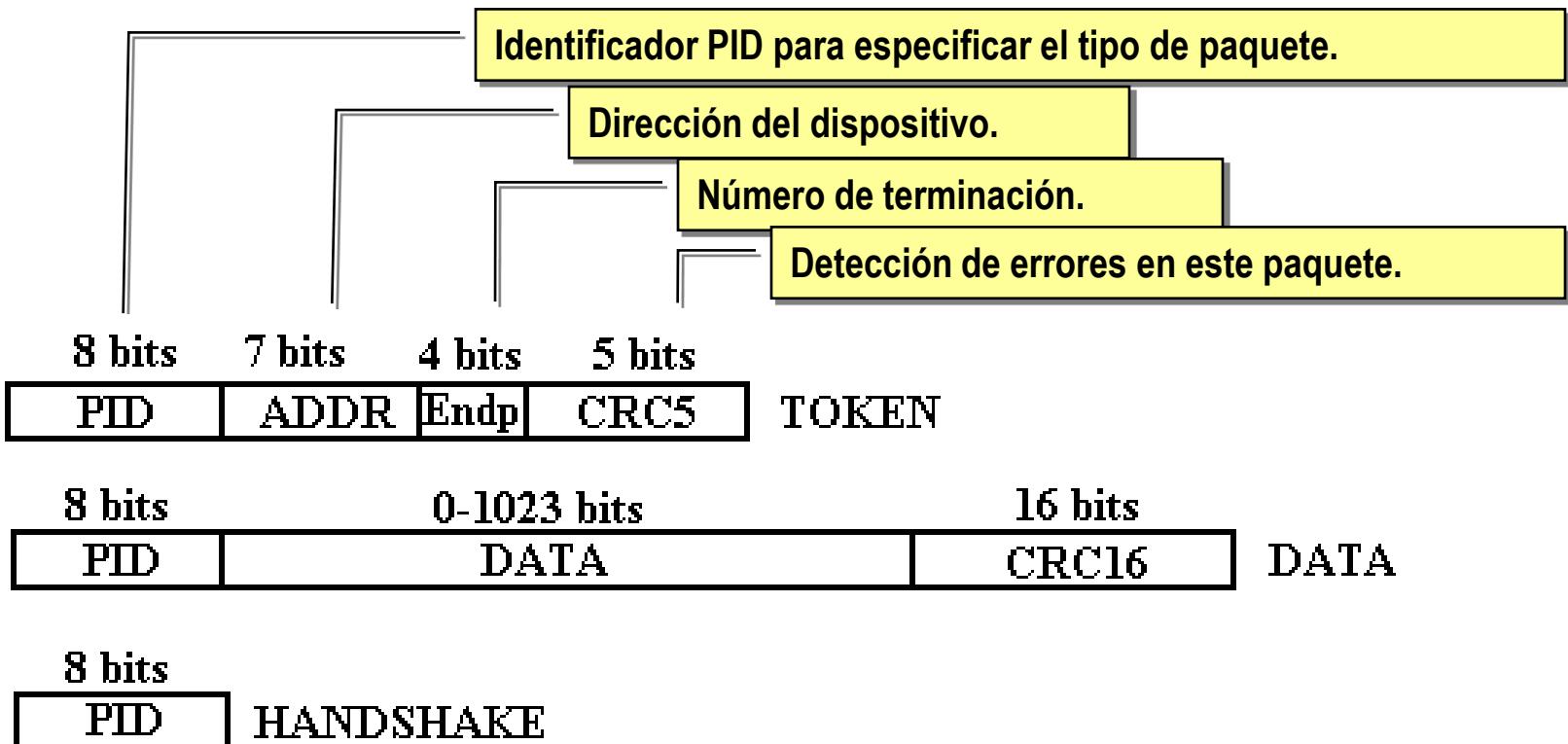
- ② Paquete de datos:

- Si el controlador ha pedido datos, el periférico responde con su propio paquete de datos.
      - Si no, espera paquete de datos del controlador.



### ③Paquete opcional de reconocimiento (*handshake*):

- Despues de recibir un paquete de datos, el dispositivo o el controlador envian un paquete de reconocimiento, indicando exito o fallo.





✖ Limitación USB: ancho de banda limitado cuando se conectan varios dispositivos a 12 Mb/s. Ej. videoconferencia.

- Soluciones:

- ☺ Añadir múltiples canales USB en el controlador (*host*).

- Por ej. 4 puertos *downstream* de 12 Mb/s → 48 Mb/s.

- ☺ USB 2.0 (2000)

- 480 Mb/s (40 veces la velocidad de USB 1.1)

- Imágenes de alta resolución, conectar discos duros, CD-ROM...

- Diversas mejoras:

- mayor tamaño de paquete.

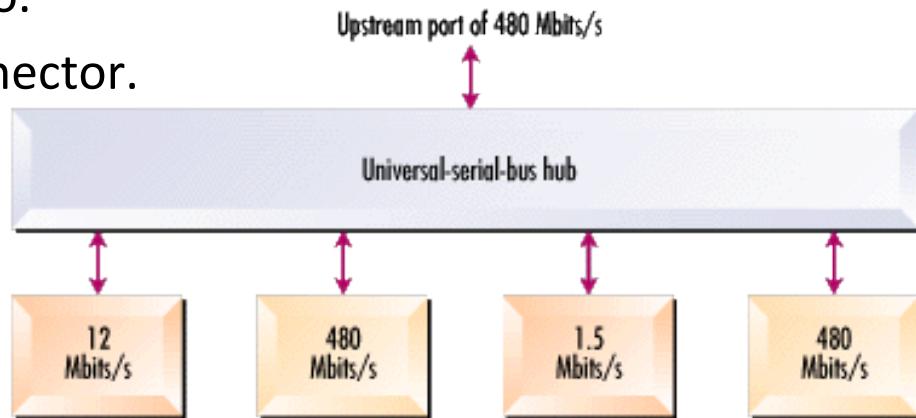
- nuevas órdenes.

- importantes mejoras eléctricas en el bus → mayor inmunidad a ruido, necesaria debido a menor voltaje y mayor frecuencia.

- nuevos modos de test (permiten comprobar un producto antes de sacarlo al mercado).



- 3 modos de funcionamiento:
  - *Low-speed*: 1.5 Mb/s
  - *Full-speed*: 12 Mb/s
  - *High-speed*: 480 Mb/s. 400 mV.
- Compatibilidad hacia atrás con USB 1.1 completamente transparente al usuario.
  - Mismo cable y conector.



Cada puerto del controlador o concentrador detecta y se adapta a la velocidad del dispositivo conectado. Si el dispositivo es *high-speed* (480 Mb/s), el concentrador se ajusta a alta velocidad. Si es *full-speed* (12 Mb/s) o *low-speed* (1.5 Mb/s) decrementa la velocidad a 12 Mb/s o 1.5 Mb/s, respectivamente. Si a un controlador de 12 Mb/s se conecta un dispositivo *high-speed*, éste aparecerá de cara al controlador como un dispositivo *full-speed* y las transacciones irán a 12 Mb/s.



- Nuevo protocolo que evita perder tiempo del bus en transferencias de salida hacia periféricos que temporalmente no pueden aceptar datos nuevos.

#### ✗ USB 1.1

- Envía paquete testigo que indica operación de salida.
- Envío de paquete de datos.
- Respuesta mediante paquete “*handshake*”.
- Se pierde mucho tiempo cuando el dispositivo no puede aceptar los datos. Con los mayores paquetes de USB 2.0 el problema es más grave.

#### ✓ USB 2.0

- El controlador determina si el dispositivo puede recibir datos, enviando un pequeño paquete testigo “*ping*”.
- El dispositivo responde con paquete “*handshake*”.
- Si no puede aceptar datos, el controlador espera algún tiempo antes de enviar otro paquete “*ping*”.

# IEEE 1394



Es un bus serie estándar, de bajo coste, no propietario, e independiente de la plataforma, para la integración de los equipos multimedia de consumo incluyendo audio y vídeo, a la vez que ordenadores y periféricos.

## ■ Bus IEEE 1394 (FireWire)

- Desarrollado originalmente por Apple como intento de mejorar el bus SCSI.
- Estándar IEEE desde 1995.
- Es una alternativa **serie** a la conexión paralela SCSI.
  - Sobre distancias cortas la comunicación paralela es mejor que la serie, sin embargo tiene un problema para mayores distancias:
    - ***skew*** (sesgar, torcer):
      - » Aunque los bits de datos se envían de forma paralela, no todos llegan al mismo tiempo.
      - » Debido a impurezas en los cables.
      - ✖ Limita la velocidad máxima posible.
  - ✓ Diseñado para ser sencillo y de bajo coste.
  - Mismas órdenes que SCSI.

# IEEE 1394



- Para **dispositivos de alta velocidad**
  - Discos duros, DVD, vídeo y audio digital, cámaras digitales, impresoras y escáneres.
    - No sólo utilizado en computadores, también en electrónica de consumo para transmitir vídeo digital.
- Tan veloz como SCSI, pero la interfaz es más sencilla.
- Hasta 400 Mb/s.
  - 30 veces más rápido que USB 1.1.
- Permite conexión/desconexión “en caliente”
  - Los dispositivos pueden ser conectados o desconectados del sistema sin necesidad de apagar el ordenador.
- Dos posibles configuraciones:
  - backplane.
  - cable.

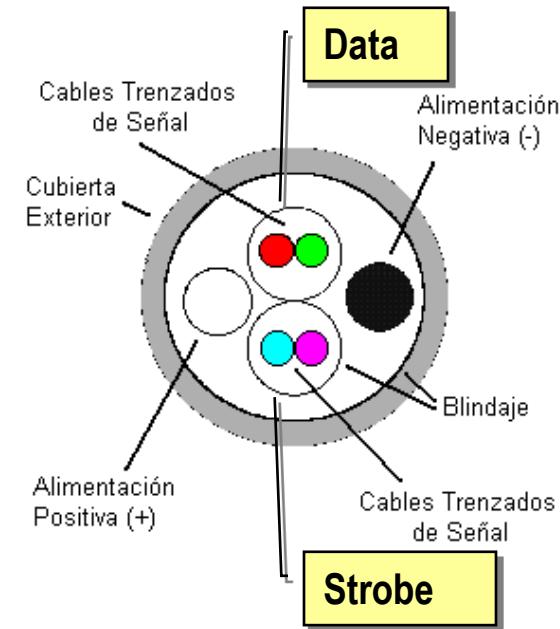


# IEEE 1394

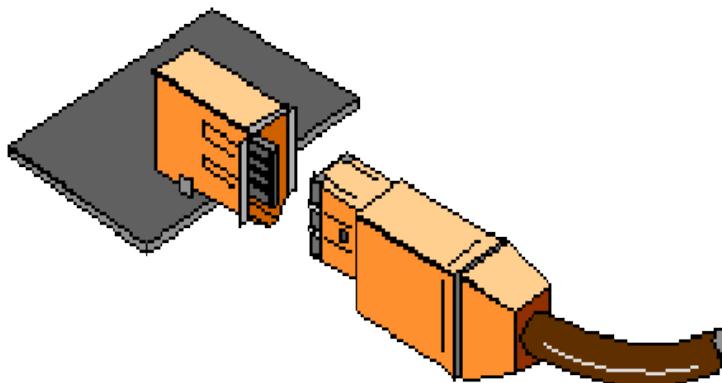


- Cable:

- Dos pares de hilos trenzados y apantallados para dos señales diferenciales:
  - ① Data
  - ② Strobe
- Un par de hilos (opcional en IEEE-1394a) para proporcionar energía a equipos de bajo consumo.

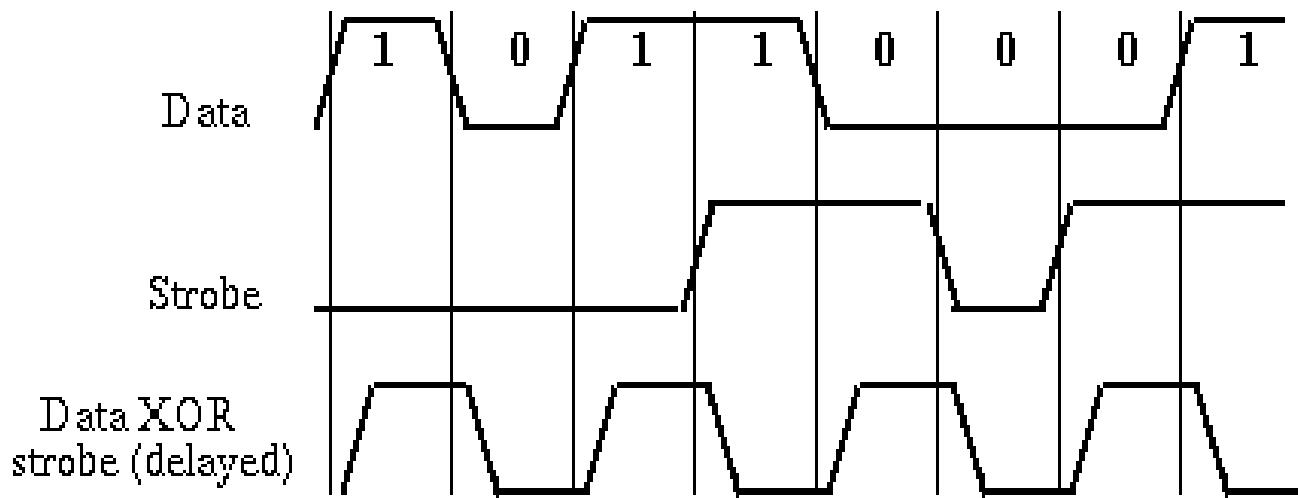


- Conector:



# IEEE 1394

- Técnica de señalización “*Data-Strobe*”:
  - Evita la necesidad de transmisión de la señal de reloj.
  - Transmisión de un bit  $\Rightarrow$  cambio en la línea *Data* o en la *Strobe*, pero nunca en ambas.
  - El reloj se recupera mediante la XOR de ambas señales (detección automática de la velocidad de transmisión).



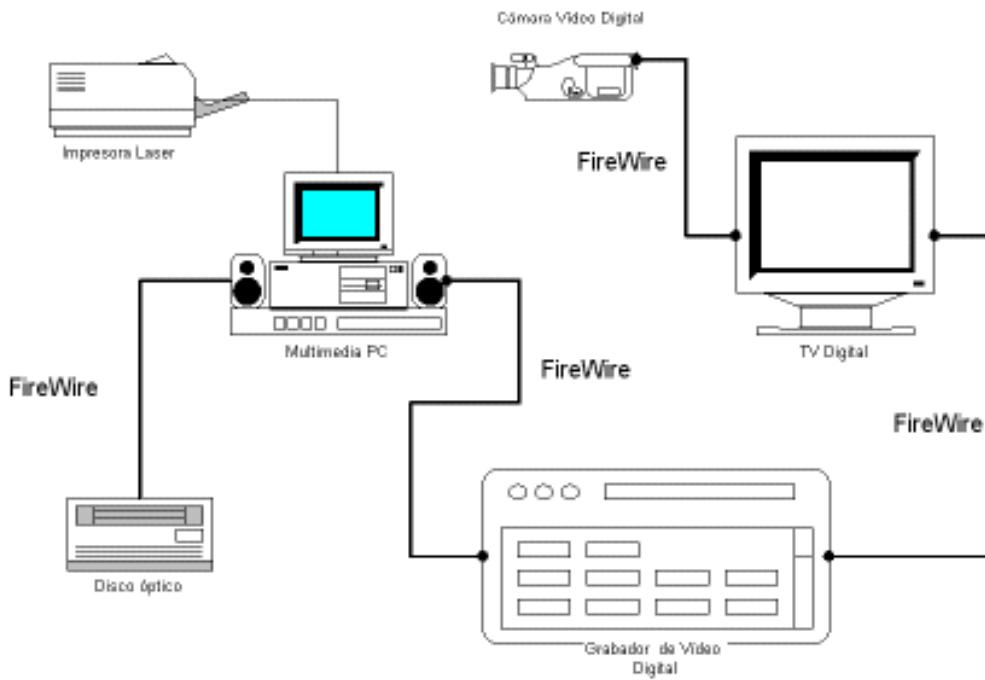
# IEEE 1394



- Direccionamiento de 64 bits:
  - **Node ID** (código de identificación de dispositivo), 16 bits:
    - **Bus ID** (dirección de bus), 10 bits.
      - » Con adaptadores (*bridges*) pueden conectarse hasta 1023 segmentos de bus.
      - » Normalmente una controladora PCI es el único bridge.
    - **Physical ID** (dirección de nodo físico), 6 bits.
      - » Se asigna a cada nodo como parte del proceso de inicialización del bus. Los nodos no se configuran manualmente.
      - » Hasta 63 dispositivos en el mismo segmento de bus (sin necesidad de concentradores).
  - **Node Offset**, 48 bits.
    - 256 TB de espacio direccionable por nodo (denominado "*espacio inicial de nodo*").

# IEEE 1394

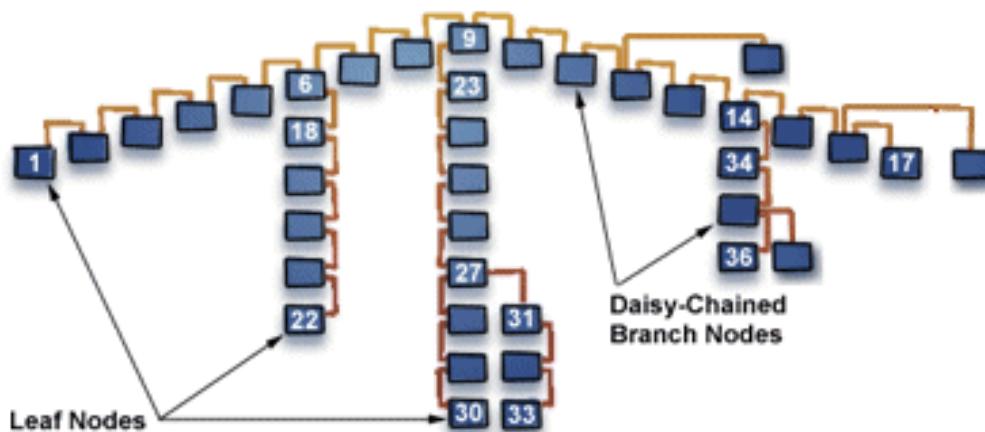
## ■ Topología



- en cadena (*daisy-chain*) (requiere  $\geq 2$  conectores por dispositivo),
- o en árbol (requiere  $\geq 3$  conectores por dispositivo).
- El estándar prevé entre 1 y 16 conectores por dispositivo (típicamente 3).
  - Dispositivos con 1 conector (sólo al final de cadena).

# IEEE 1394

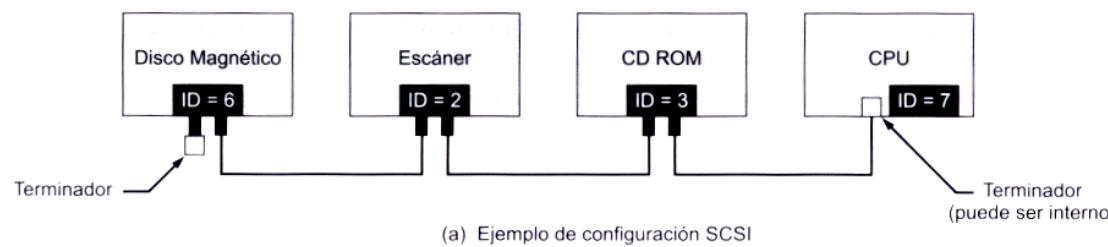
- Enlaces punto a punto de hasta 4.5 m. Puede aumentarse esta distancia por medio de repetidores (*bridges*) entre dos buses.
- Al actuar los repetidores como paso entre un bus y otro, el tráfico de un bus no interfiere en el del otro bus, salvo que un paquete deba pasar de un bus a otro ↗ ancho de banda total ↗.
- Distancia máxima entre cualesquiera dos dispositivos no puede ser mayor que 16 cables (15 dispositivos entre ellos, hasta 72 m).



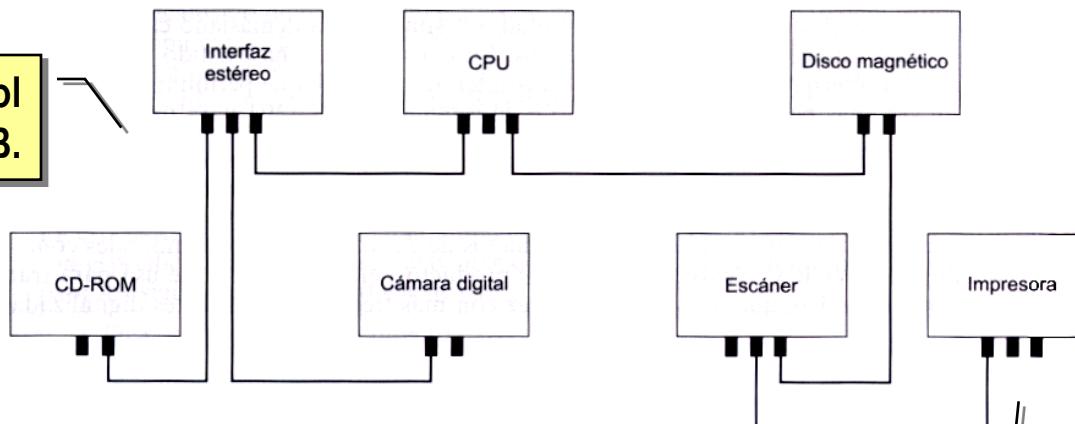
No more than 16 cables or 72 meters between any two leaf nodes  
(for example, between nodes 1 and 17 and between nodes 33 and 22)

# IEEE 1394

- Comparación de topologías SCSI y FireWire.



(a) Ejemplo de configuración SCSI



(b) Ejemplo de configuración FireWire

Topología en árbol similar a la de USB.

En FireWire no hay terminadores.

# IEEE 1394

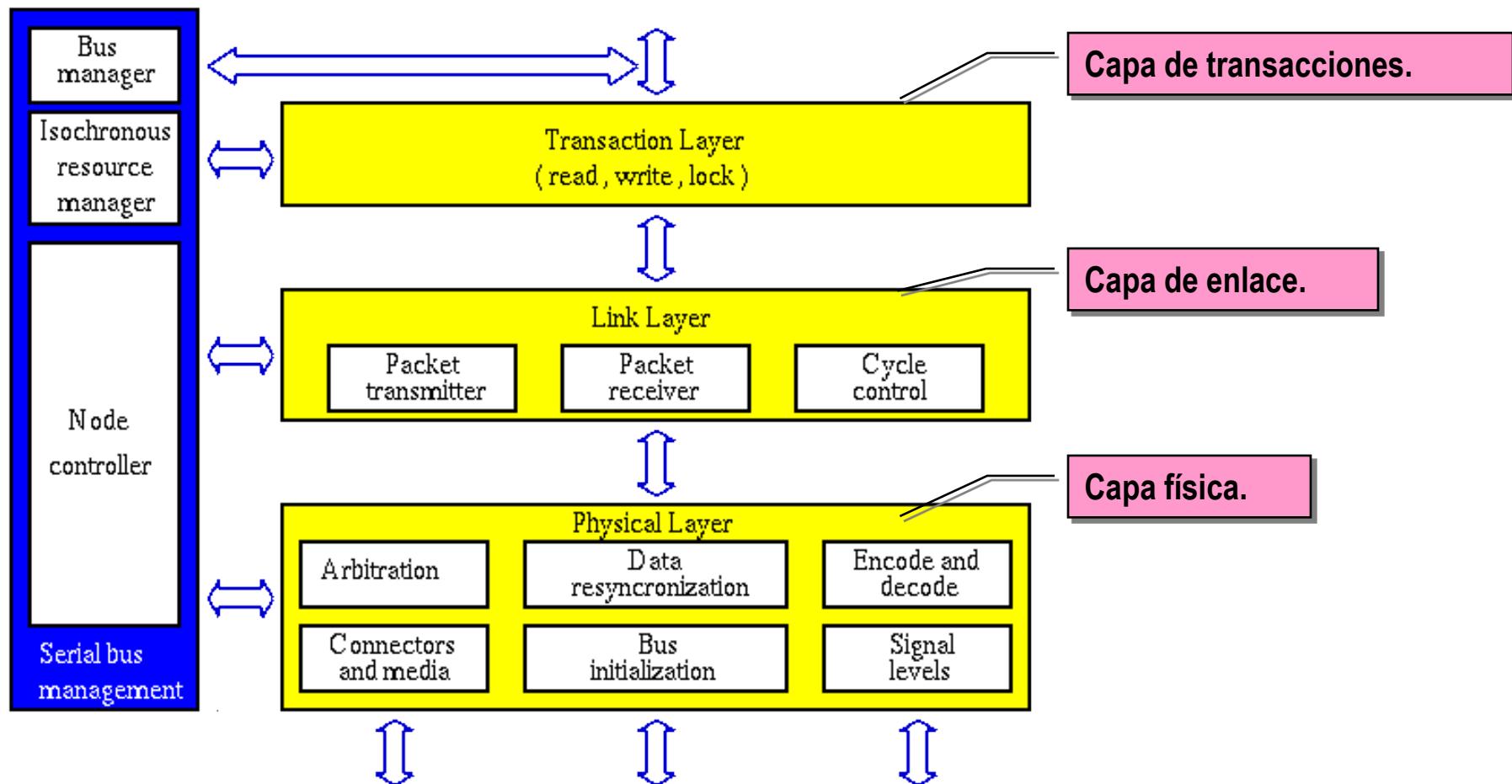


- Velocidad:
  - 3 opciones:
    - S100:  $\approx 100$  Mb/s ( 98 304 b/s)
    - S200:  $\approx 200$  Mb/s (196 608 b/s)
    - S400:  $\approx 400$  Mb/s (393 216 b/s)
      - » Por ej. esta velocidad permite transferir vídeo descomprimido en 640x480 y 30 frames por segundo y a la vez audio estéreo de 16 bits y alta calidad.
  - Mayor velocidad en IEEE 1394b:
    - S800: 800 Mb/s (100 MB/s)
    - S1600: 1.6 Gb/s (200 MB/s)
    - S3200: 3.2 Gb/s (400 MB/s)
  - El bus utiliza la velocidad del dispositivo más lento.

# IEEE 1394



- Protocolo en 3 capas:



# IEEE 1394



- **Capa de transacción:**

- Define los tipos de transacciones (lectura, escritura).
  - Oculta a las aplicaciones los detalles de las capas inferiores.

- **Capa de enlace:**

- Describe la transmisión de datos mediante paquetes.
  - Dos tipos de transmisión:
    - Asíncrona.
    - Isócrona.

- **Capa física:**

- Define los medios de transmisión que permite el FireWire, y las características eléctricas y señales de cada uno.
  - Convierte los datos binarios en las señales eléctricas y viceversa.
  - Proporciona el servicio de arbitraje.

# IEEE 1394



## ■ Arbitraje:

- Uno de los nodos se designa como la raíz del árbol, y los demás se organizan mediante relaciones padre/hijo.
- Una vez que se ha establecido la configuración, el **nodo raíz** actúa como **árbitro central**.
  - Procesa las solicitudes de acceso al bus de manera que la primera en llegar es la primera en atenderse (“*first-come-first-served*”).
  - Ante solicitudes simultáneas, se concede el bus al dispositivo con mayor prioridad (mayor cercanía al nodo raíz y en caso de igualdad menor valor de identificación).

# IEEE 1394



- **Arbitraje equitativo:**

- Evita que uno o más dispositivos de prioridad alta puedan monopolizar el uso del bus.
  - » El tiempo se organiza en **intervalos de equidad** (*Fairness Interval*).
  - » Al principio del intervalo se registran los nodos que están luchando por acceso al bus. El intervalo termina cuando todos los que competían por el bus han sido atendidos. Entonces comienza un nuevo intervalo.
  - » Cuando un nodo gana el acceso al bus, no puede competir de nuevo durante el intervalo actual.

- **Arbitraje urgente:**

- Algunos dispositivos se pueden configurar para que tengan una prioridad urgente.
  - » Pueden ganar el acceso al bus varias veces en un intervalo de equidad.

# IEEE 1394



- **Paquete:**

- Cabecera:
  - Identificadores de la fuente y el destino.
  - Tipo de paquete.
  - CRC.
  - Parámetros del tipo específico de paquete.
- Bloque de datos (opcional):
  - Datos.
  - CRC.

# IEEE 1394



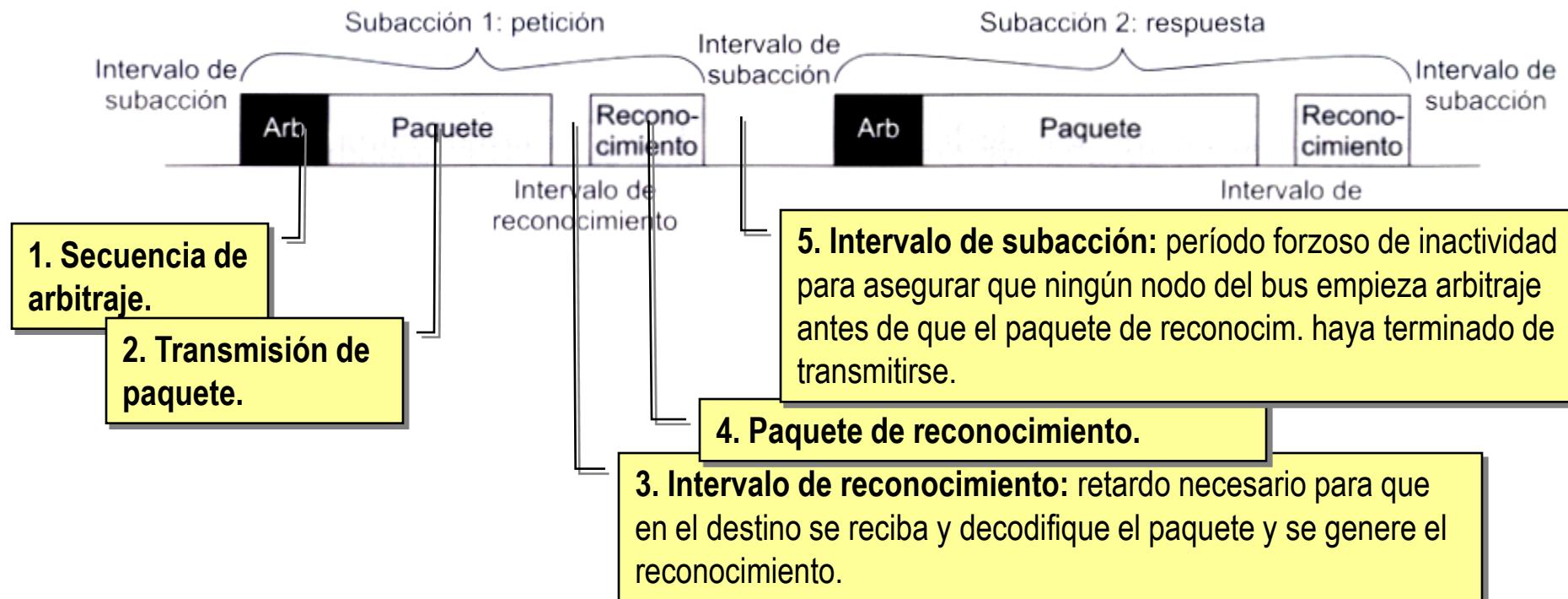
- **Transmisión asíncrona:**

- Se transmite un paquete con una cantidad variable de datos.
- Se devuelve un paquete de reconocimiento.
  - Confirmación por parte del dispositivo destino indicando que los datos se recibieron correctamente.
- Para dispositivos que no necesitan tener una velocidad de transferencia fija. Ej.: impresora, módem.

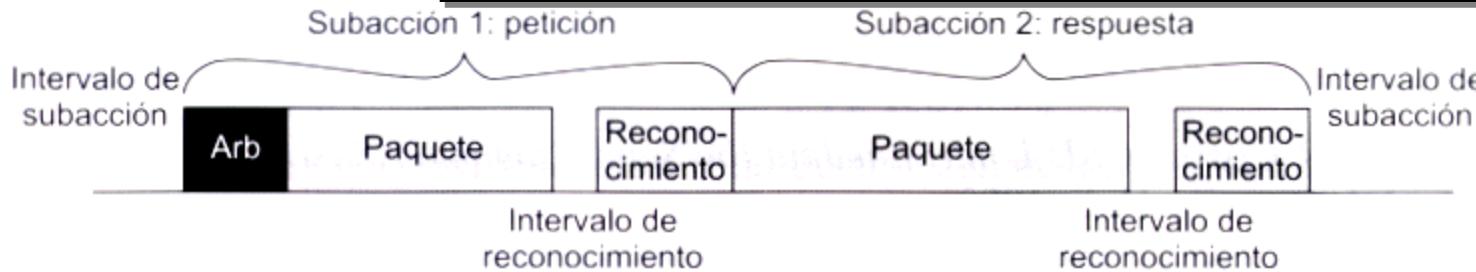
# IEEE 1394



**Subacción** → proceso de envío de un paquete



En el momento que se envía el reconocimiento, el nodo que lo envía tiene el control → puede transmitir inmediatamente el paquete de respuesta sin tener que realizar arbitraje



# IEEE 1394



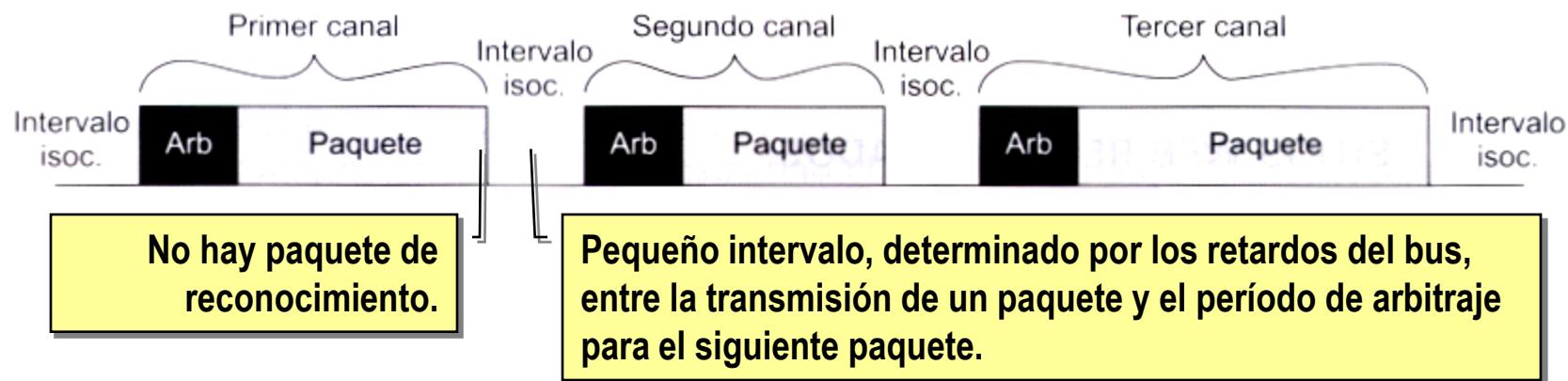
- **Transmisión isócrona:**

- **Garantiza un ancho de banda fijo sin que se produzcan interrupciones en el flujo de datos.**
- Se transmite una cantidad variable de datos mediante una **secuencia de paquetes de tamaño fijo, a intervalos regulares.**
- Para dispositivos multimedia de audio y vídeo digital que generan o consumen datos de manera regular.

...

# IEEE 1394

- Uno de los nodos se designa como **maestro del ciclo**.
  - Periódicamente (cada 125  $\mu$ s, 8000 ciclos/s) genera un **paquete de comienzo de ciclo** isócrono.
- Durante un ciclo isócrono sólo se envían paquetes isócronos.
- Cada fuente de datos isócronos interviene en el arbitraje.
- Nodo ganador transmite un paquete.



- Despues de que todas las fuentes isócronas hayan transmitido, las fuentes asíncronas pueden competir por acceder al bus. Podrán utilizar el bus hasta el comienzo del siguiente ciclo isócrono.

# IEEE 1394



- Diferentes estándares:
  - IEEE 1394 (1995): Estándar original.
  - IEEE 1394a
    - Incorpora protocolo de gestión energética *Suspend and Resume*, que permite a los nodos entrar en un modo de bajo consumo y ser despertados mediante una señal enviada por otro nodo.
    - Incorpora conector de 4 contactos para nodos no alimentados desde el bus.
  - IEEE 1394b
    - 800 Mb/s, 1.6 Gb/s, 3.2 Gb/s
      - » con fibra óptica de hasta 100 m por conexión.
    - Latencia por debajo de los 125 µs.
    - Arbitraje concurrente con la transmisión del paquete.

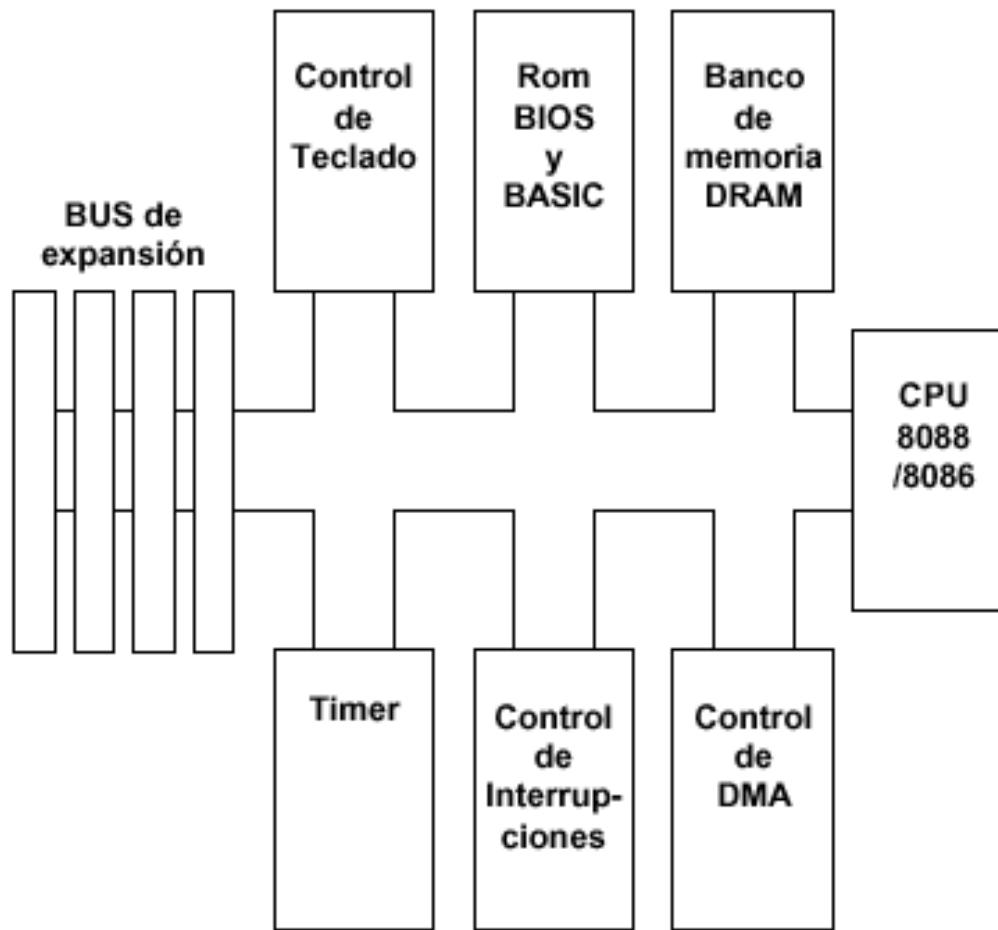
# Buses serie para periféricos

## ■ Comparación entre USB y FireWire

	USB 2.0	IEEE 1394b
<b>Similaridades</b>		
Conexión en caliente	Sí	Sí
Cables	Similares: delgados, flexibles, y con conectores simples y robustos.	
Topología	Estrella en capas	Estrella
<b>Diferencias</b>		
Comunicación dispositivo a dispositivo	No	Sí
Conector	A al host, B al dispositivo	El mismo para todos los dispositivos
Ancho de banda	480 Mbps (60 MB/s) en USB 2.0	800 Mbps (100 MB/s) 1.6 Gbps (200 MB/s) 3.2 Gbps (400 MB/s)
Conexión de periféricos internos	No	Sí (discos, por ejemplo)
Nº máx. dispositivos	127	63
Periféricos	Teclado, ratón, palanca de juegos, CD-ROM, módem, cámara digital, impresora, escáner	Cámara digital, HDTV, disco duro, DVD-ROM
Coste de implementación	Muy bajo	Bajo

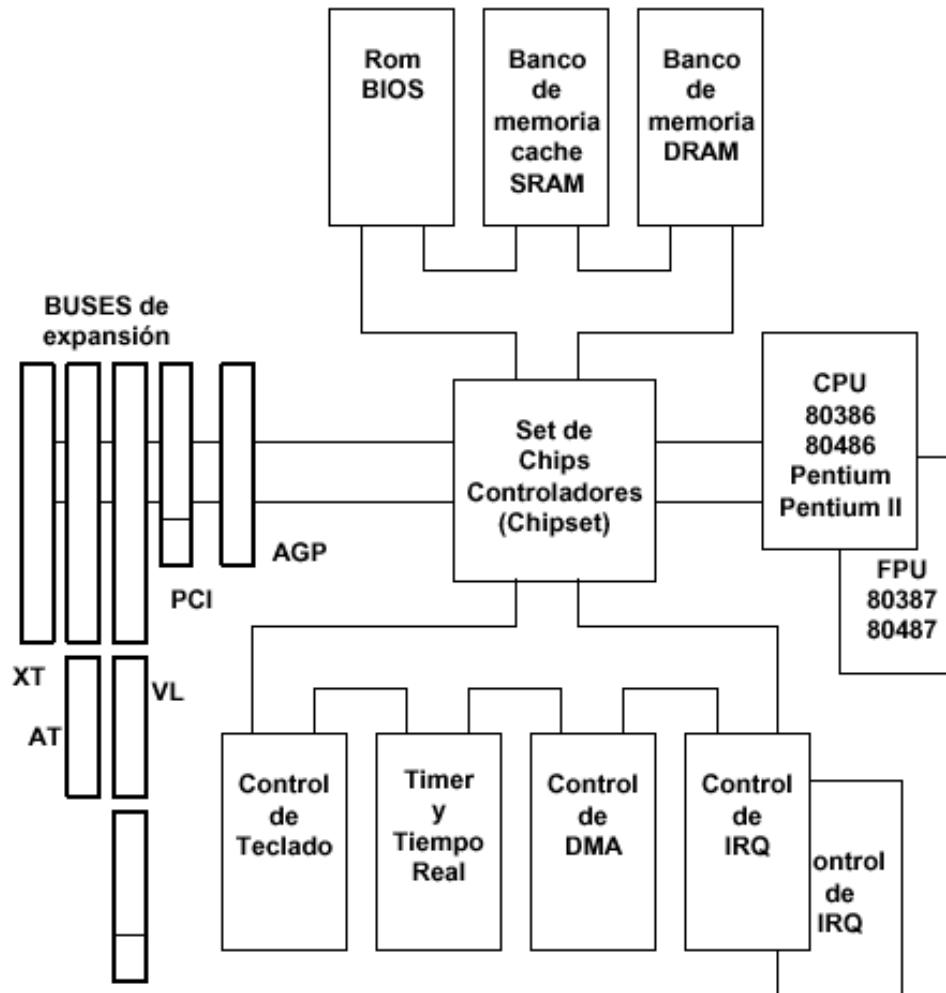
# Resumen de buses en el PC

- Arquitectura del PC/XT original:



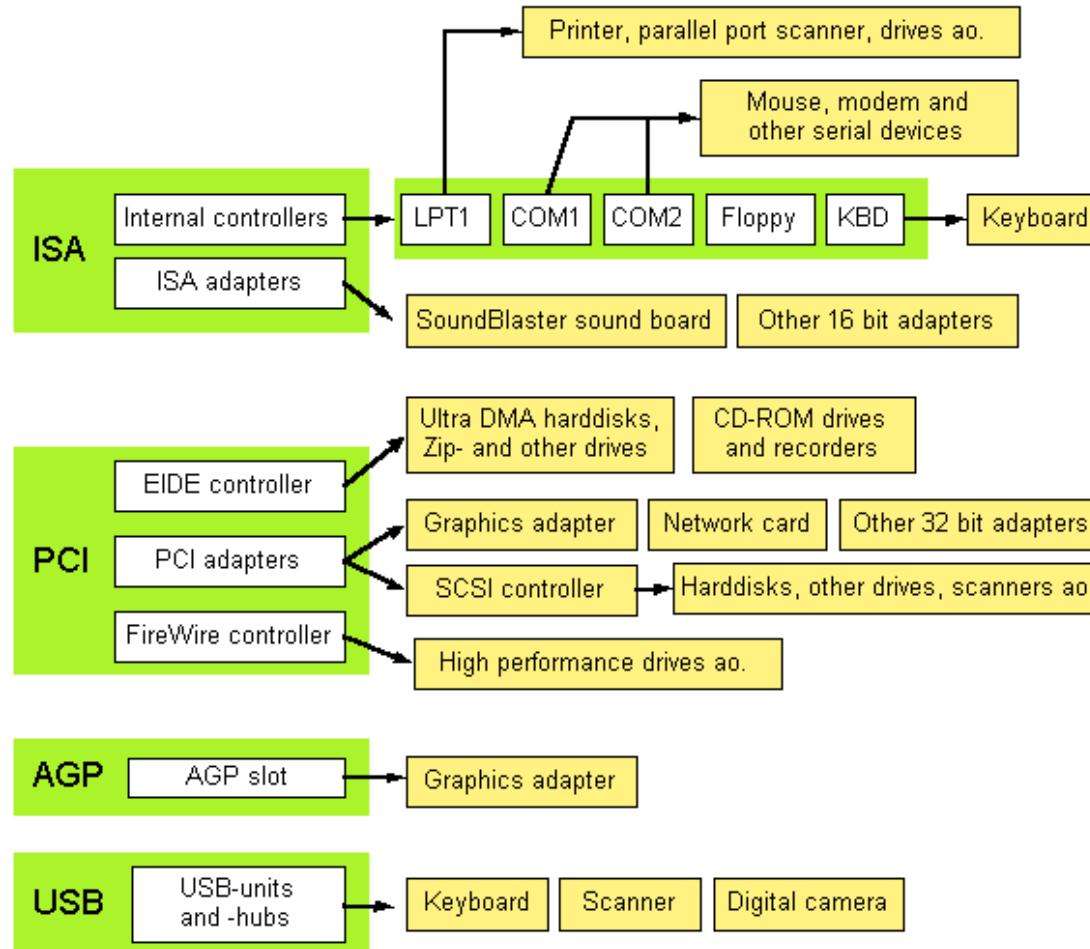
# Resumen de buses en el PC

- Arquitectura de un PC más reciente:



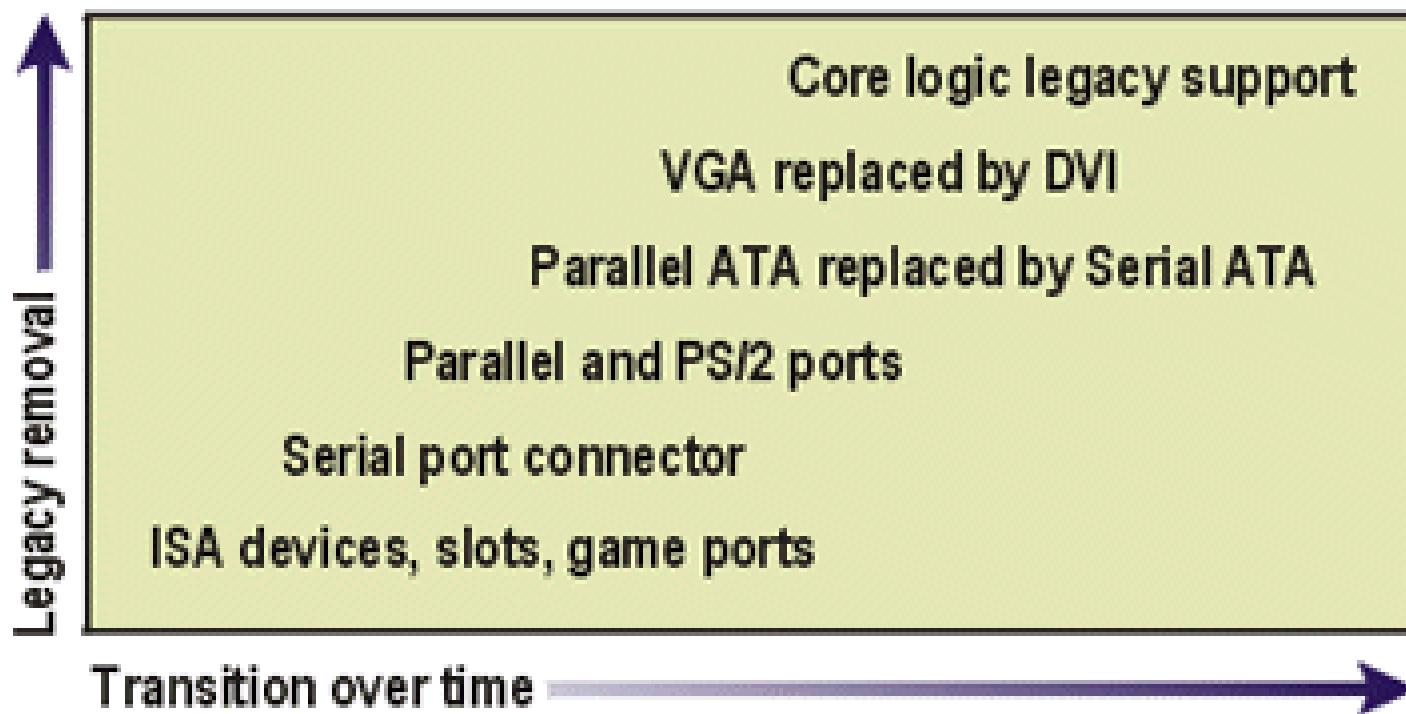
# Resumen de buses en el PC

## ■ Diferentes buses en el PC más reciente:



# Resumen de buses en el PC

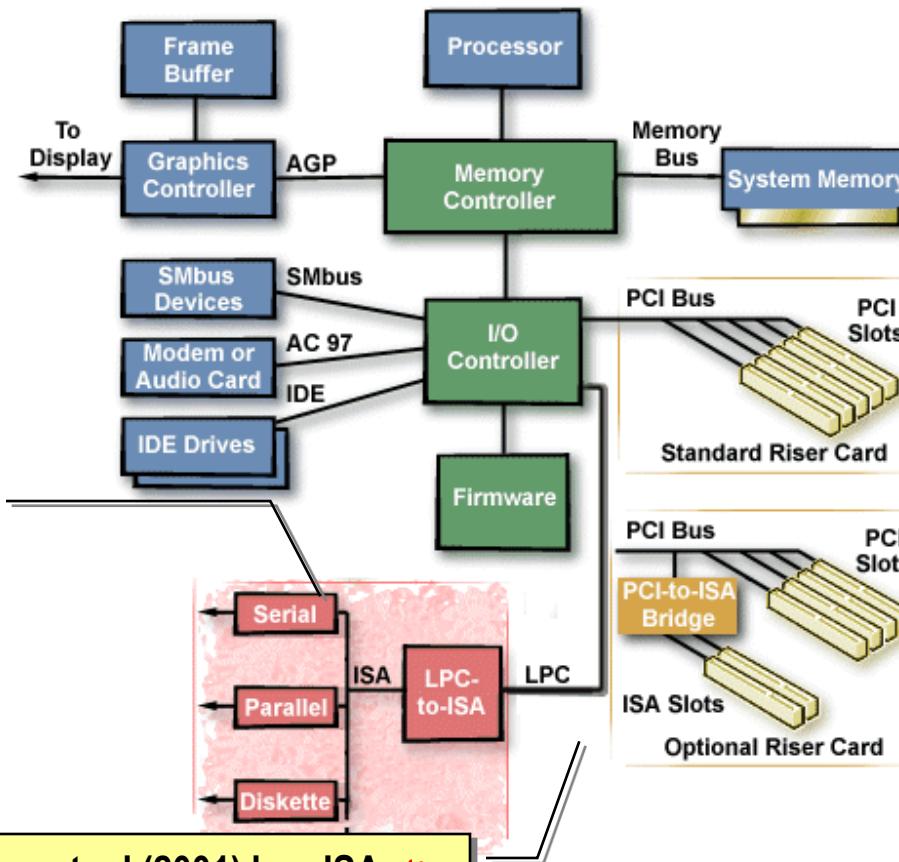
- Hacia el “*Legacy-Free PC*” (PC libre de herencia del pasado):



# Resumen de buses en el PC

## ■ Hacia la desaparición del bus ISA (I):

### ■ Bus LPC



**Bus LPC (Low Pin Count):** preserva las características principales del bus ISA, pero con menos contactos, menor espacio en la placa madre y menor consumo.

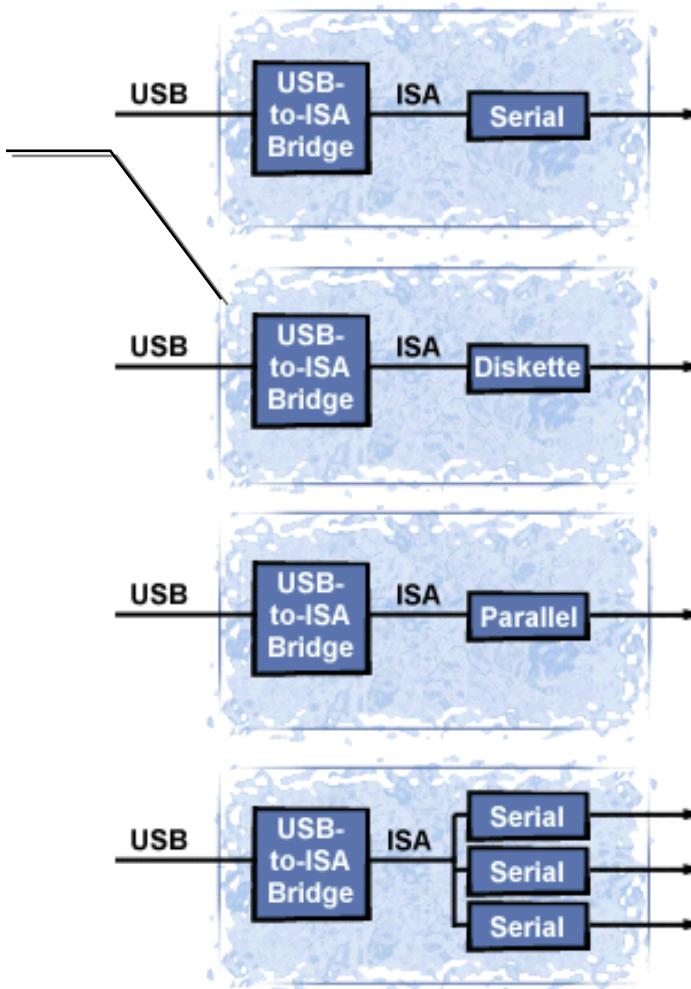
Configuración actual (2001) bus ISA. ❌  
Usa ancho de banda del bus PCI.  
Tiende a desaparecer.

# Resumen de buses en el PC

## ■ Hacia la desaparición del bus ISA (II):

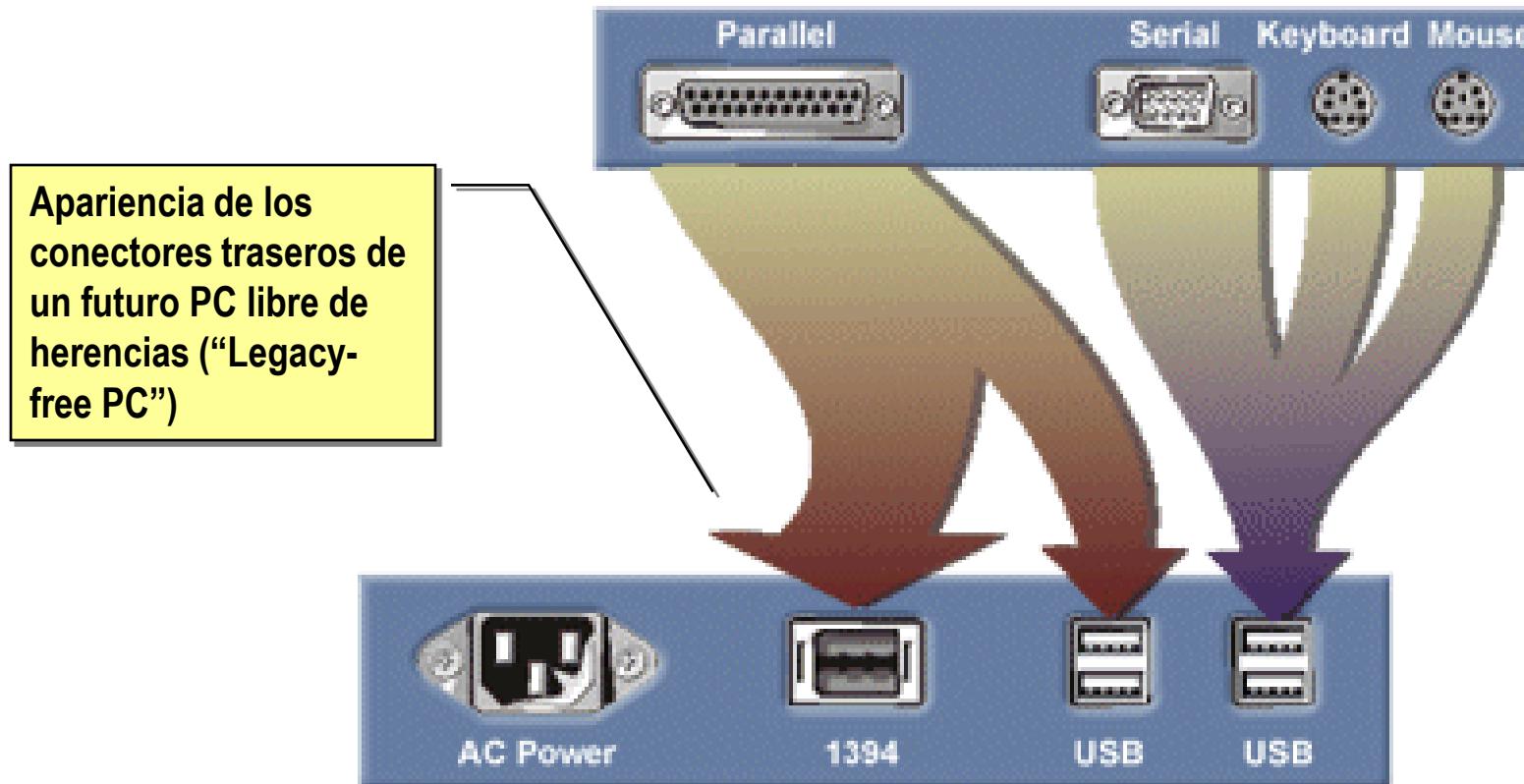
### ■ *USB Mappers:*

Son dispositivos externos o internos que se conectan a USB por un lado y a periféricos “obsoletos” por el otro.



# Resumen de buses en el PC

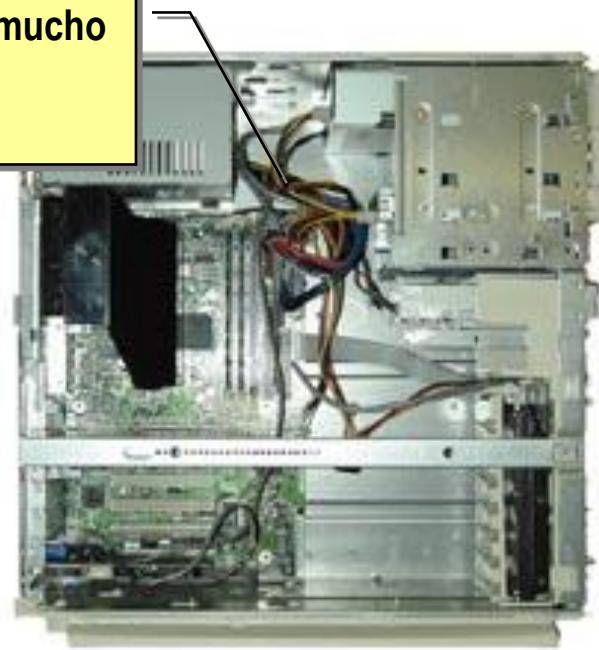
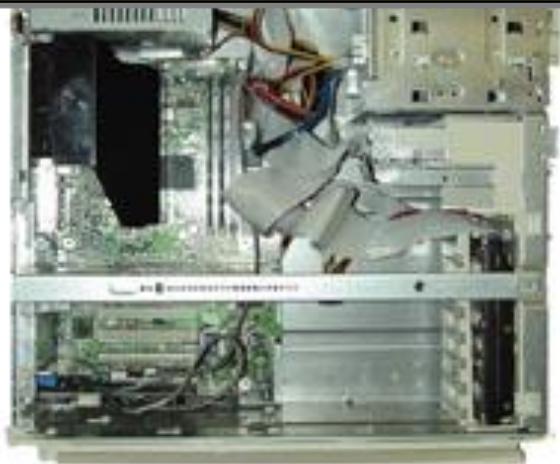
- Hacia la desaparición de los puertos paralelo, serie, ... :



# Resumen de buses en el PC

## ■ Transición de ATA paralelo a ATA serie:

**Bus ATA serie:** menor voltaje, menor número de contactos, cable mucho más pequeño y más largo (hasta 1 m). El software es compatible.  
Velocidad a partir de 150 MB/s.



Adaptadores para compatibilidad hardware.

