



Arquitecturas de Almacenamiento

Tema II – Parte 1

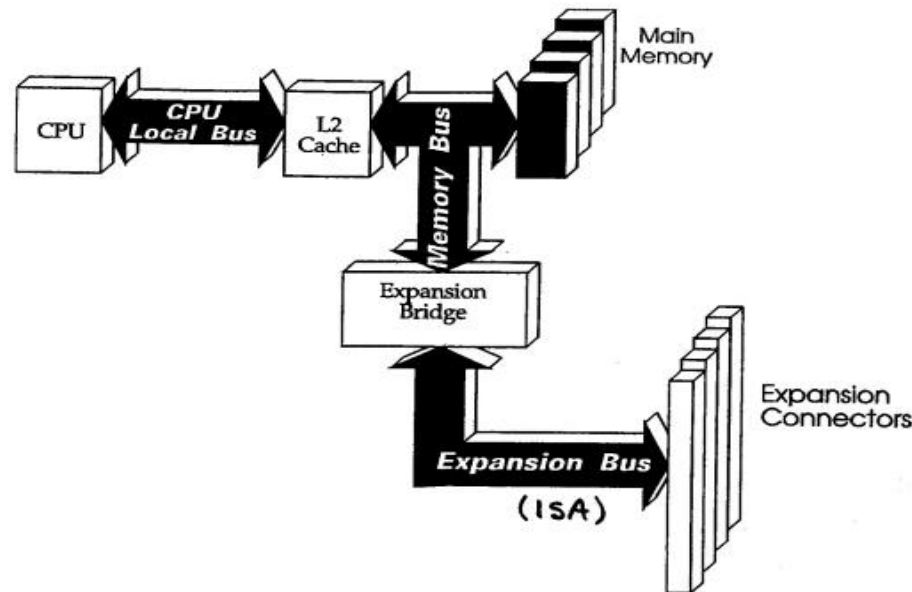
Análisis y evolución del bus PCI

Contenidos

- Bus PCI: Arquitectura y operación
- Bus PCI-X: Evolución del PCI para aumentar el rendimiento
- Bus PCI-Express: Serialización del bus e introducción de un modelo de capas

I – Bus PCI: Introducción

- Bus PCI aparece a finales de los 80 para solucionar limitación en el rendimiento de la E/S de la arquitectura del PC
- Durante la década de los 80, la arquitectura dominante en el PC era la ISA
 - Arquitectura basada en bus local CPU-Memoria, más un bus ISA (AT) para E/S



Bus PCI: Introducción

■ Capacidad del bus de E/S muy limitada:

- 24 bits direcciones, 16 bits datos
- Bus síncrono, CLK = 8.33 MHz
- No soporta modo ráfaga
 - Cada transferencia de datos va precedida de una fase de direccionamiento (incluso para DMA)
 - Por tanto, incluso para los dispositivos más rápidos (0 WS = 0 estados de espera), se requiere un mínimo de 2 CLK/transferencia

■ Velocidad máxima de transferencia ISA es sólo 8.33 MBytes/seg

- Num.transferencias/seg = $8.33 \text{ MHz} / 2 = 4.165 \text{ Mtransf/seg}$
- Bytes/transf = 16 bits = 2 bytes
- Luego $V_{\text{tmax}} = 4.165 \text{ Mtransf/s} * 2 \text{ bytes/transf} = 8.33 \text{ MB/s}$

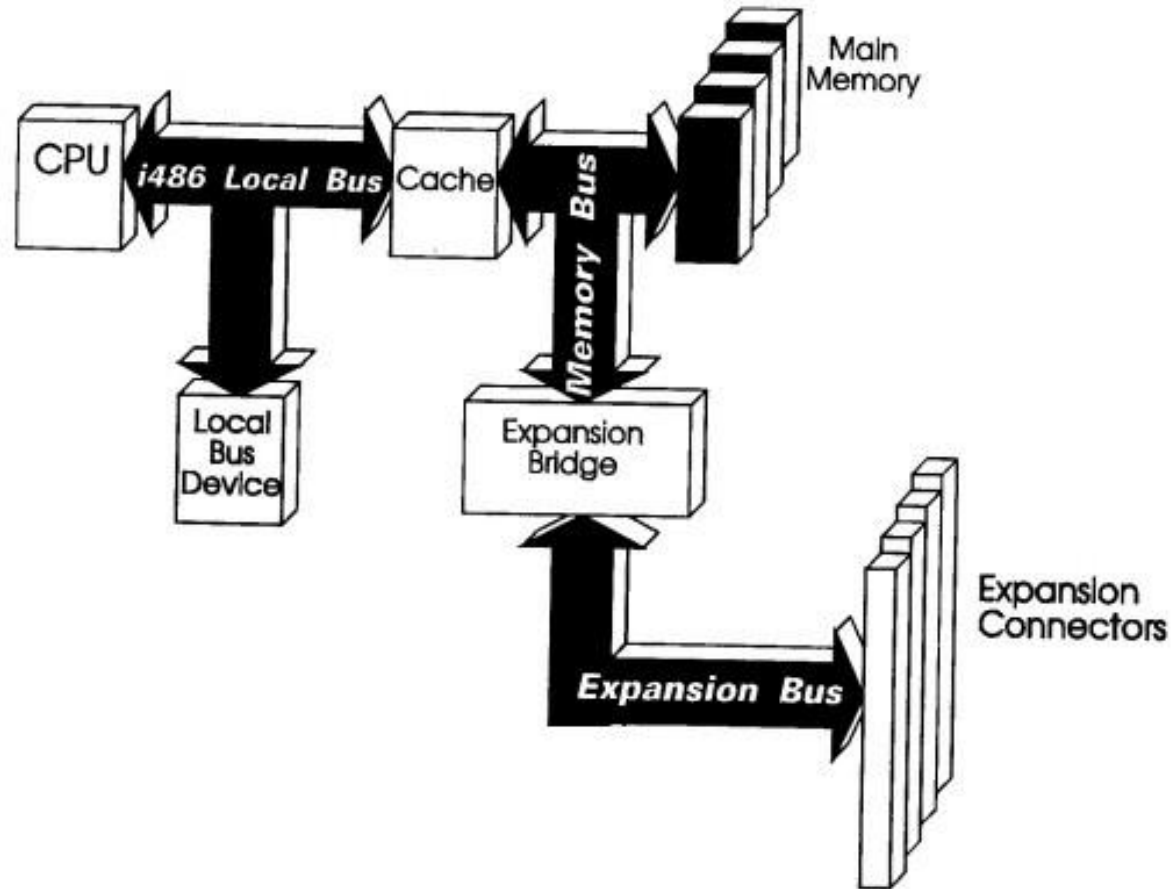
■ Sin embargo, arbitración de bus y frecuentes estados de espera (no ráfaga, y bus bloqueante) hacían que V_t real fuera muy inferior a la nominal

Arquitectura VESA Local Bus

- A finales de los 80, capacidades gráficas de PC (para juegos) requerían imperiosamente aumento capacidad bus de E/S
- A iniciativa de la asociación de estándares para dispositivos de video (*VESA = Video Electronic Standards Association*), aparece una arquitectura de alta velocidad, llamada *VESA Local Bus*
 - Usada para procesadores Intel 386 y 486
 - Orientada casi exclusivamente a resolver necesidades de ancho de banda para adaptadores gráficos (tarjetas de video)

Arquitectura VESA Local Bus

- En arquitectura VESA Local Bus, el dispositivo de E/S se conecta directamente al bus local CPU-Memoria



Arquitectura VESA Local Bus

- Características del VESA Local Bus mejores que las de ISA:
 - Bus de 32 bits (4 bytes) de datos
 - Bus síncrono, con típicamente CLK = 33 MHz para i486
 - Soporta transferencias en modo ráfaga, con 1 CLK/transferencia
 - Por tanto, soporta 33 Mtransferencias/s para CLK = 33 MHz
- Velocidad máxima de transferencia muy superior a ISA
 - $V_{\text{tmax}} = 33 \text{ Mtransf/seg} * 4 \text{ bytes/transf} = 132 \text{ MBytes/s}$

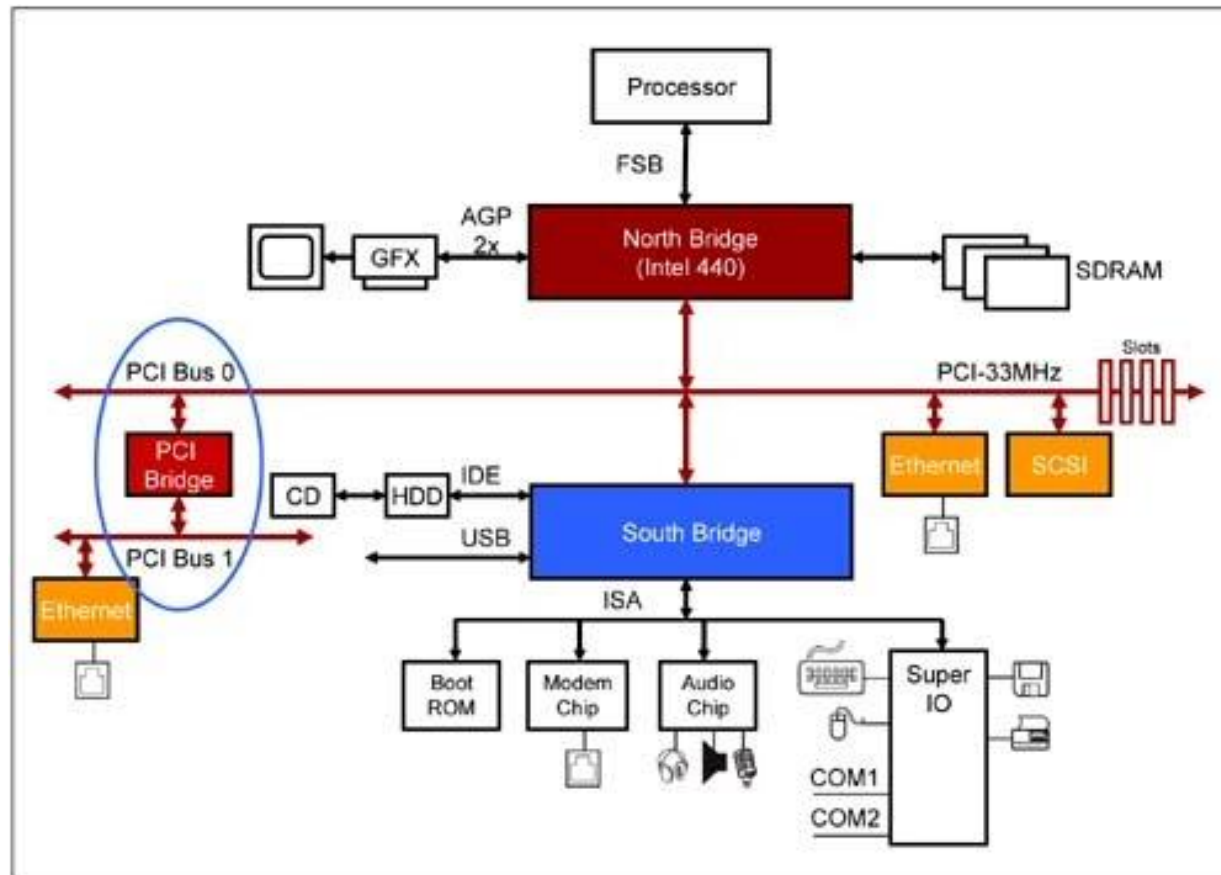
Arquitectura VESA Local Bus

- Arquitectura VESA Local Bus tenía, sin embargo, serios problemas:
 - Presencia dispositivo en bus local CPU-M producía frecuentes problemas de estabilidad del sistema
 - CPU debe detener su funcionamiento durante transferencias DMA a dispositivos VESA LB
 - Limitado a uso en procesadores 386/486
 - Forzosamente trabaja a la misma frecuencia de CLK a que esté operando la CPU
 - Esto producía problemas de estabilidad si se hacía *overclocking* (es decir, configurar placa madre para forzar CPU a trabajar por encima de la frecuencia nominal del procesador)
- Se necesitaba arquitectura con la misma V_t , pero más flexible y estable
 - Esta necesidad motivó la aparición de la arquitectura PCI

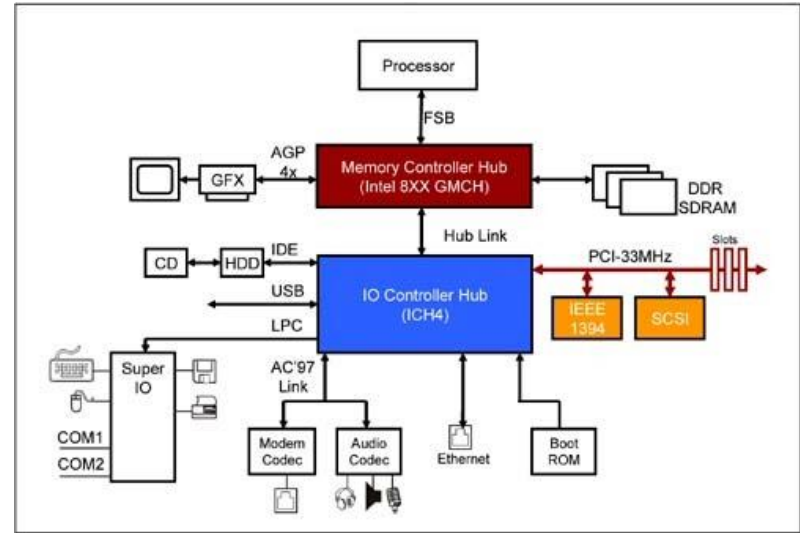
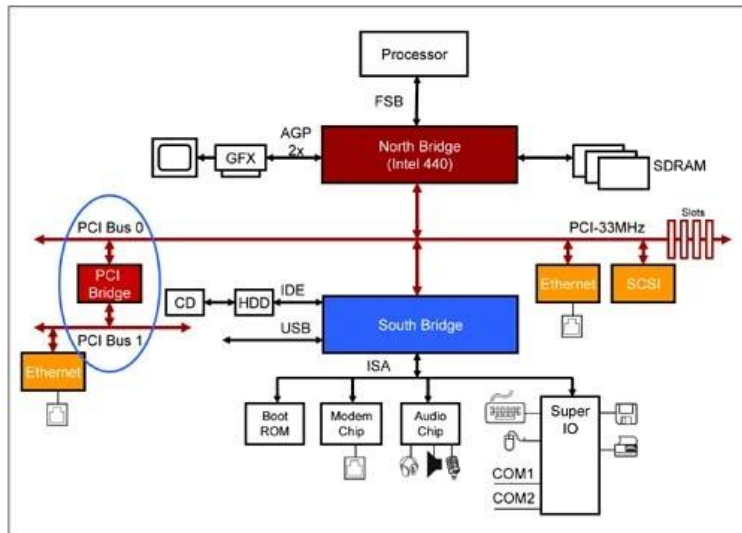
Arquitectura PCI

■ Propuesta por Intel en 1991

- Desarrollada por consorcio PCI SIG (Special Interest Group) como bus abierto de uso público

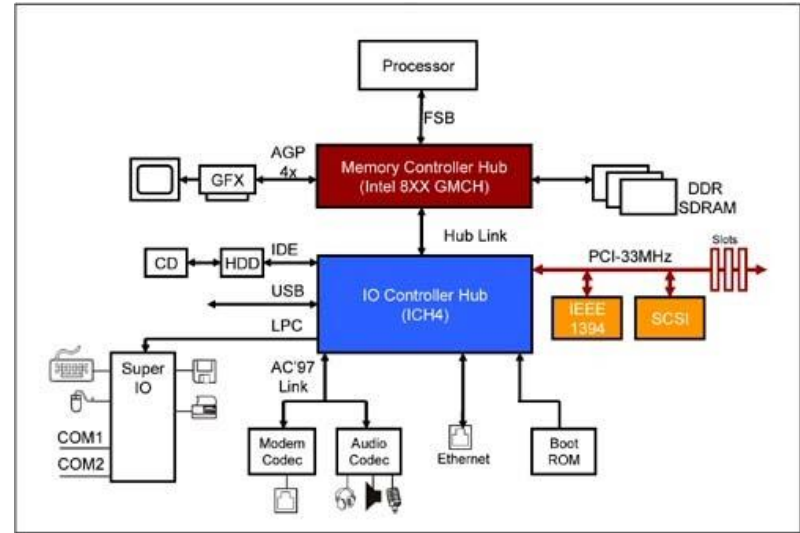
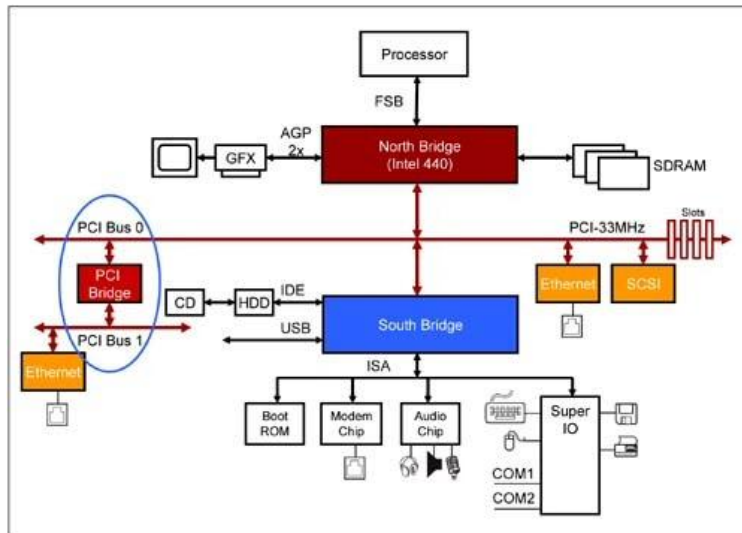


Arquitectura PCI



- En arquitectura PCI, circuito controlador de memoria desacopla CPU y RAM en el bus local, y hace de puente al bus de E/S
 - Permite acceder a memoria concurrentemente desde CPU y E/S
 - Hace al bus PCI independiente de la marca, tipo y modelo de procesador

Arquitectura PCI



■ Bus PCI es un bus jerárquico

- Puede soportar en cascada otros buses PCI, hasta 256
- Cada bus, a su vez, soporta hasta 32 dispositivos físico y 256 dispositivos lógicos
- Permite también conectar en cascada otros buses de E/S, usando el HBA (Host Bus Adapter) correspondiente

Arquitectura PCI

■ Características técnicas y capacidad del bus PCI:

- Bus síncrono, con CLK = 33 o 66 MHz
- Puede usar 32 (4 bytes) o 64 (8 bytes) de direcciones/datos
- Usa por defecto transferencias en modo ráfaga de longitud arbitraria, con 1 CLK/transferencia
- Soporte completo de busmastering, con arbitración centralizada paralela
- Velocidad máxima de transferencia típicamente de 132 Mbytes/s, y puede llegar a 528 Mbytes/s

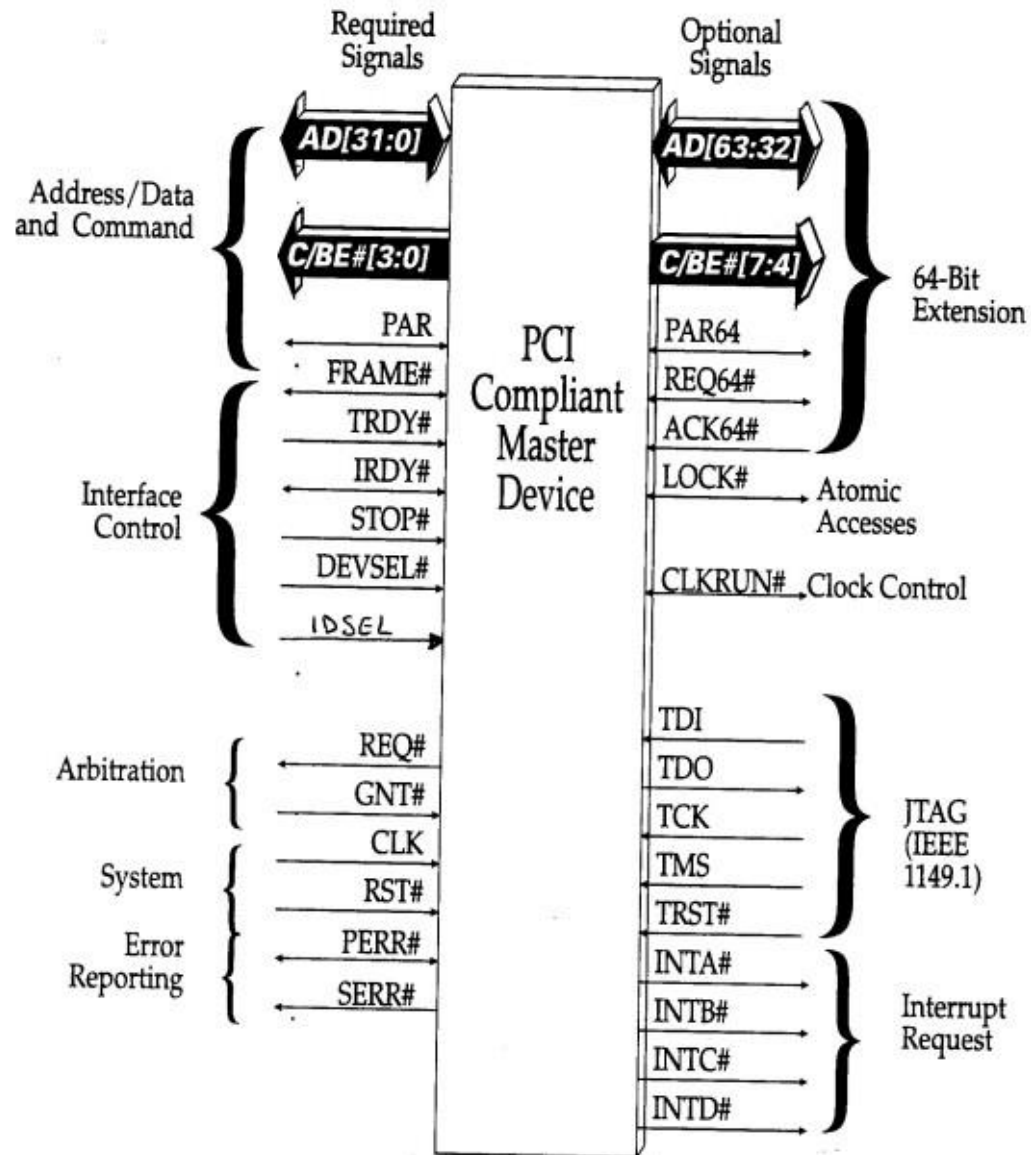
$$V_{\text{tmax}} (33 \text{ MHz}/32 \text{ bits}) = 33 \text{ Mtransf/seg} * 4 \text{ bytes/transf} = 132 \text{ Mbytes/s}$$

$$V_{\text{tmax}} (66 \text{ MHz}/64 \text{ bits}) = 66 \text{ Mtransf/seg} * 8 \text{ bytes/transf} = 528 \text{ Mbytes/s}$$

- Soporta transacciones divididas, y permite finalizar prematuramente las transacciones
 - Porcentaje de uso efectivo del bus es muy alto, y también, por tanto, la V_t sostenida

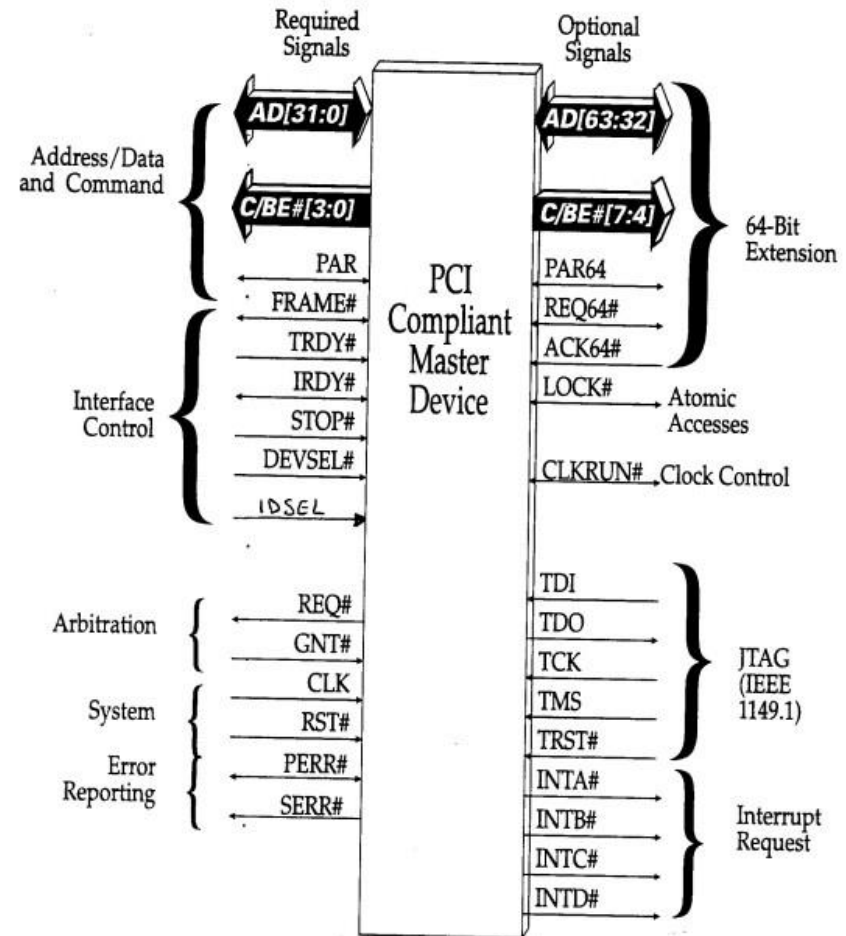
Control del bus: Iniciador y diana

- En el bus PCI cada transacción de bus involucra a un iniciador y una diana
 - Iniciador ("Initiator"): es un maestro de bus, es decir, un dispositivo capaz de comenzar una transacción de bus
 - Genera para ello una fase de direccionamiento, en la que da valores a las líneas de direcciones y a los bits identificadores del tipo de transacción
 - Diana ("Target"): es un esclavo de bus, que espera pasivamente a ser direccionado por un iniciador
 - No puede generar fases de direccionamiento, ni compete por el bus en el proceso de arbitración



Señales del bus PCI

- La figura muestra las señales correspondientes a un dispositivo iniciador
 - Las de la diana son las mismas, excepto las líneas de arbitración
- NOTA: El estándar PCI utiliza el sufijo “#” para indicar que una línea es activa en baja (es decir, el “1” lógico está representado por un “0” físico, y viceversa)



Señales del bus PCI

■ Las señales son:

- CLK: Reloj del bus síncrono. Las señales del bus se leen e interpretan en su flanco de subida
- FRAME#: Usada por los iniciadores para indicar que el bus está en uso; delimita el inicio y final de una transacción
- AD[31:0] (opcionalmente AD[63:32]): Bus multiplexado de 32 (opcional 64) bits de direcciones y datos
- C/BE#[3:0] (opcionalmente C/BE#[7:4]): En la fase de direccionamiento contienen un código de 4 bits con el tipo de transacción. En cada fase de datos indican qué bytes del bus de datos tienen contenido válido

Señales del bus PCI

- IRDY#: Indica que el iniciador está listo para intercambiar un ítem de datos con la diana
 - Su desactivación permite al iniciador solicitar la inserción de ciclos de espera
- TRDY#: Indica que la diana está lista para intercambiar un ítem de datos con el iniciador
 - Su desactivación permite a la diana solicitar la inserción de ciclos de espera
- DEVSEL#: Activada por una diana cuando reconoce (decodifica) su dirección durante la fase de direccionamiento
- STOP#: Activada por la diana para forzar la terminación prematura de una transacción de bus
- REQ# y GNT#: Líneas de arbitración paralela centralizada, presentes sólo en los iniciadores (maestros de bus)
 - REQ# permite pedir al árbitro el bus
 - GNT# permite al árbitro indicar la cesión del bus

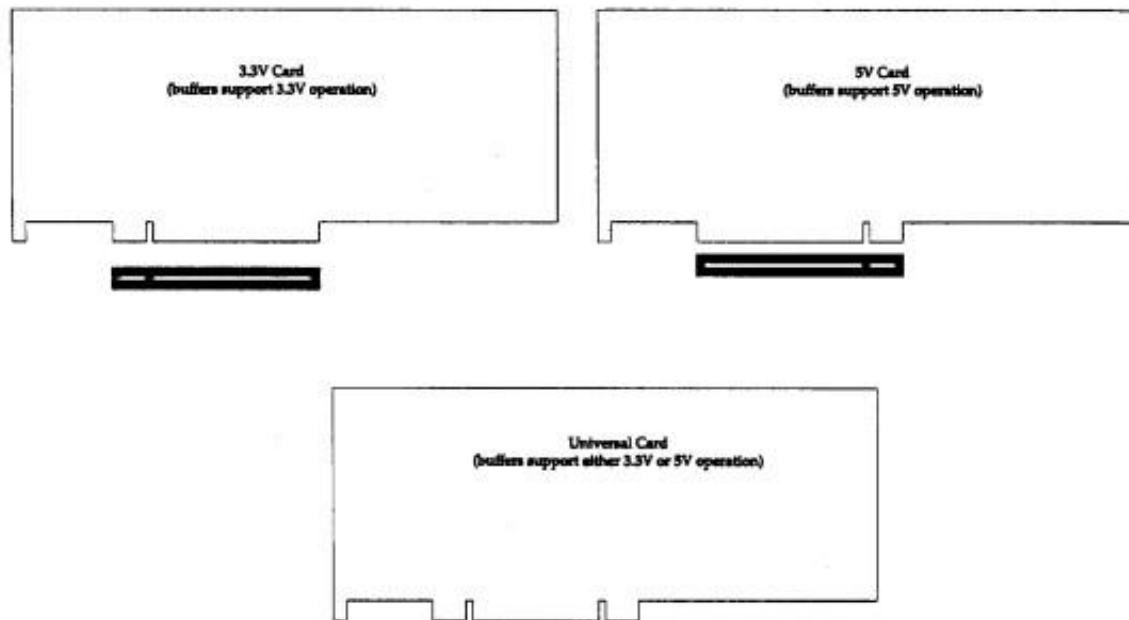
Señales del bus PCI

- INTA# a INTD#: Líneas de petición de interrupción
- IDSEL#: Usada en la autoconfiguración “Plug & Play” para acceder al espacio de configuración PCI del dispositivo

Tarjetas PCI de 5V ("Legacy") y 3.3V

■ Estándar PCI soporta señalización a 5V y a 3.3V

- Para evitar daños, es físicamente imposible insertar tarjetas de 5V en slots de 3.3V, y viceversa
- Tarjeta universal puede ser insertada en cualquiera de los dos slots
- OJO: "PCI Legacy" es un término que identifica a un slot PCI con señalización a 5V



Transacciones de bus PCI más relevantes

■ Reglas generales:

- Valores en las líneas del bus PCI sólo son efectivas en el flanco de subida de CLK
- Las transferencias de datos PCI se realizan en modo ráfaga, pero:
 - la fase de direccionamiento PCI no contiene información sobre la longitud de la ráfaga
 - por tanto, la diana no conoce a priori cuantos datos se van a transferir
- Iniciador usa FRAME# para indicar que el bus está ocupado, y para señalar a la diana la longitud de la ráfaga
 - Iniciador activa FRAME# al inicio de la transacción para indicar bus ocupado
 - Iniciador desactiva FRAME# justo antes de la última transferencia de datos, para indicar a diana el fin de la ráfaga

Transacción de lectura

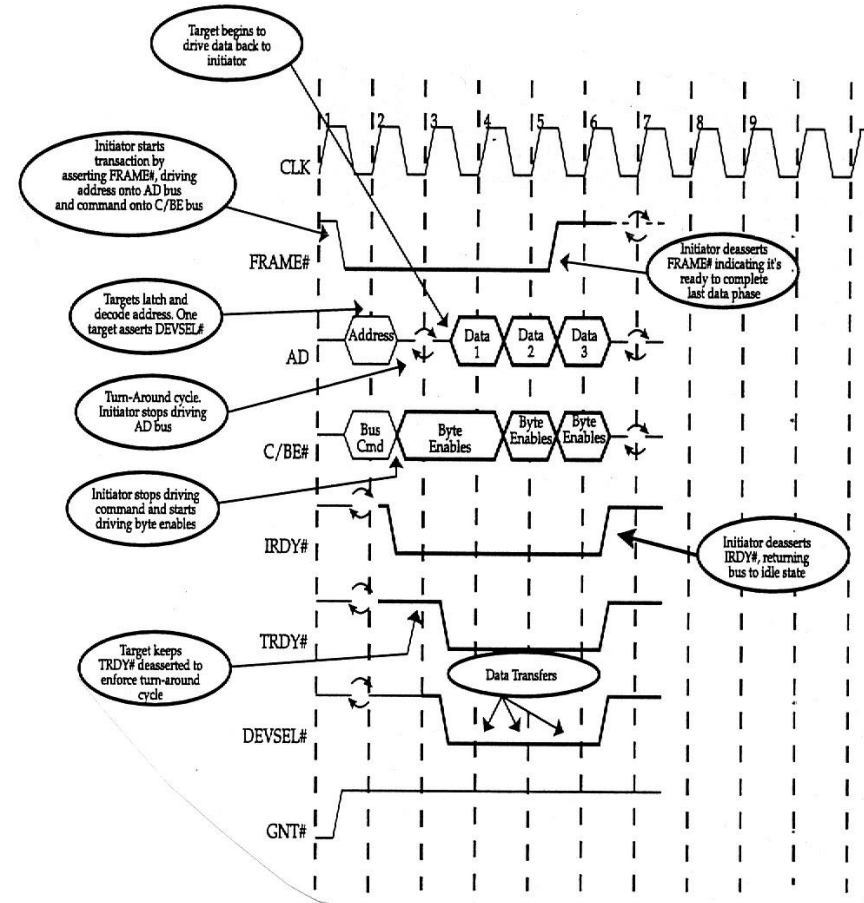
Secuencia de eventos (entre paréntesis, ciclo de reloj en que se produce):

(1) Iniciador gana arbitración, GNT# está activa

(1) Iniciador detecta bus libre (inactivas FRAME#, IRDY# y TRDY#)

(2) Iniciador activa FRAME# y toma control del bus

(2) Iniciador genera fase de direccionamiento, dando valores a AD (dirección de diana) y C/BE# (código de tipo de transacción)



Transacción de lectura

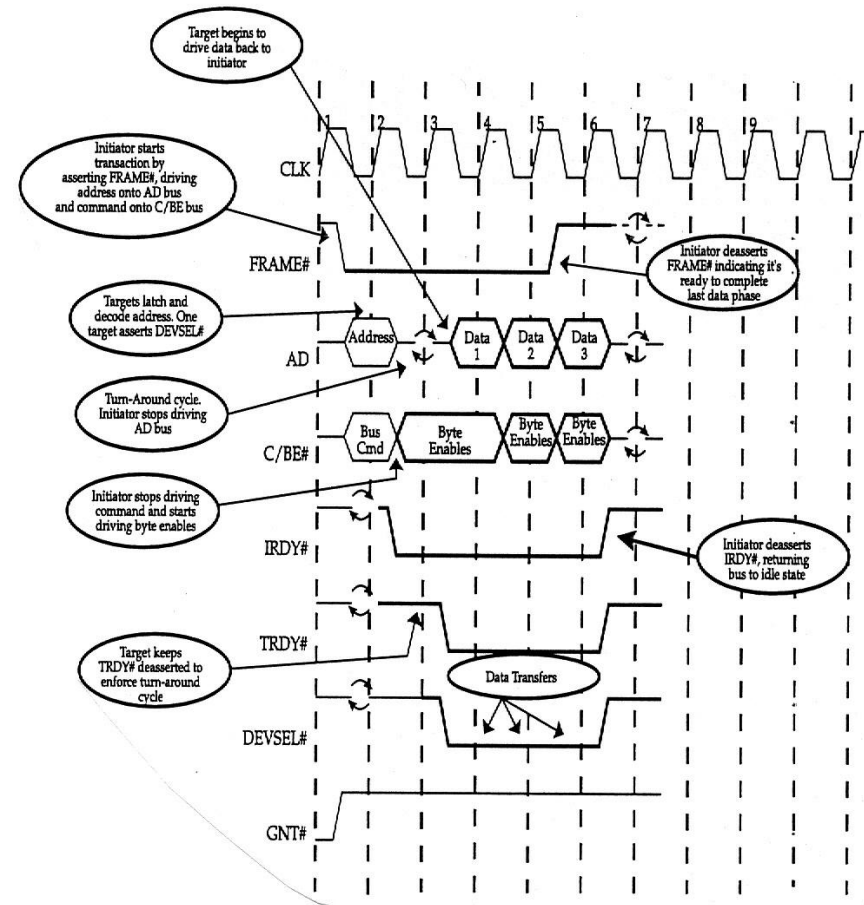
(4) Si diana ha contestado (DEVSEL#) y ambos extremos están listos (IRDY#, TRDY# activas), comienza ráfaga, y diana escribe datos en el bus

- En cada CLK se transfieren 4 bytes de datos al iniciador

(6) Iniciador desactiva FRAME# para indicar a diana fin de ráfaga. Se transfiere el último ítem de datos

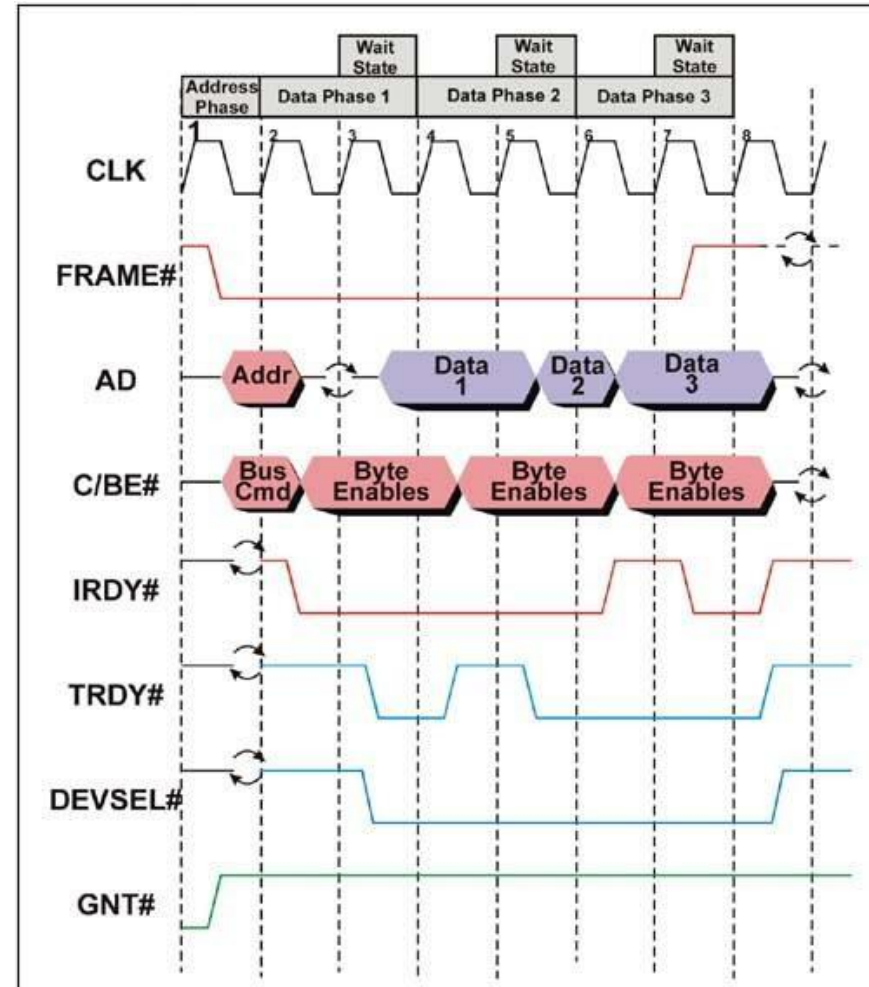
(7) Iniciador y diana desactivan IRDY#, TRDY#, DEVSEL#, y dejan el bus libre

La transacción de escritura es similar, cambiando él sentido de la transferencia de datos



Transacción de lectura

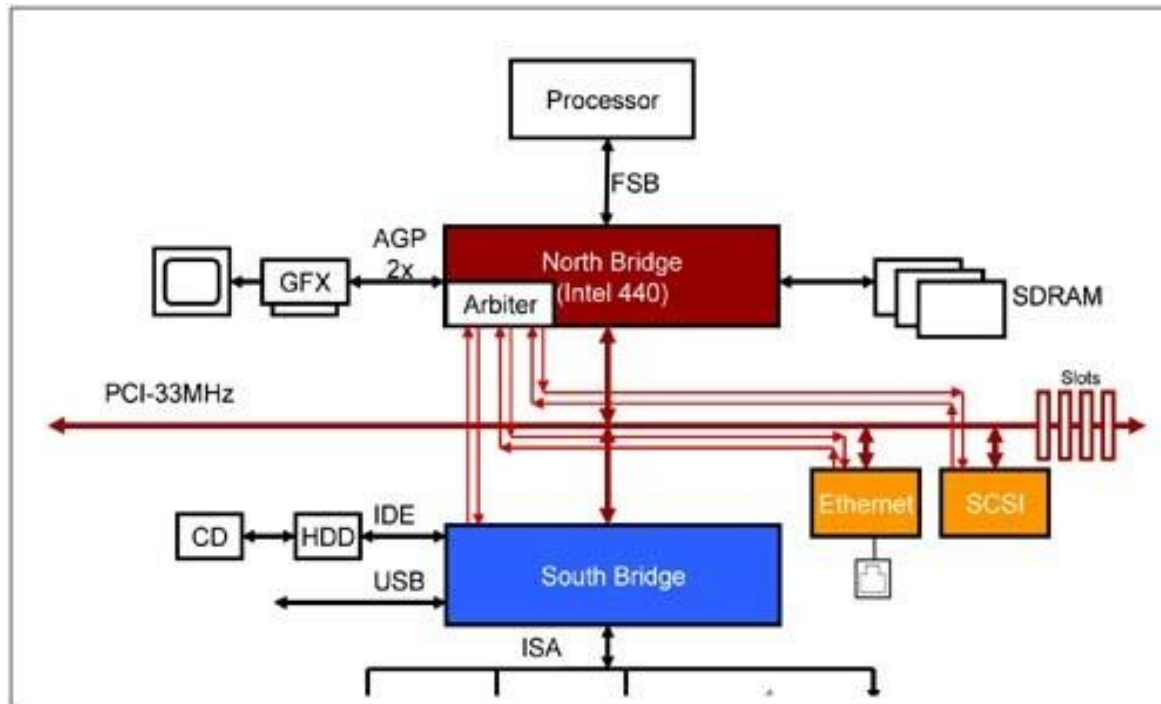
- La inserción de estados de espera durante la ráfaga se hace desactivando IRDY# o TRDY#



Arbitración

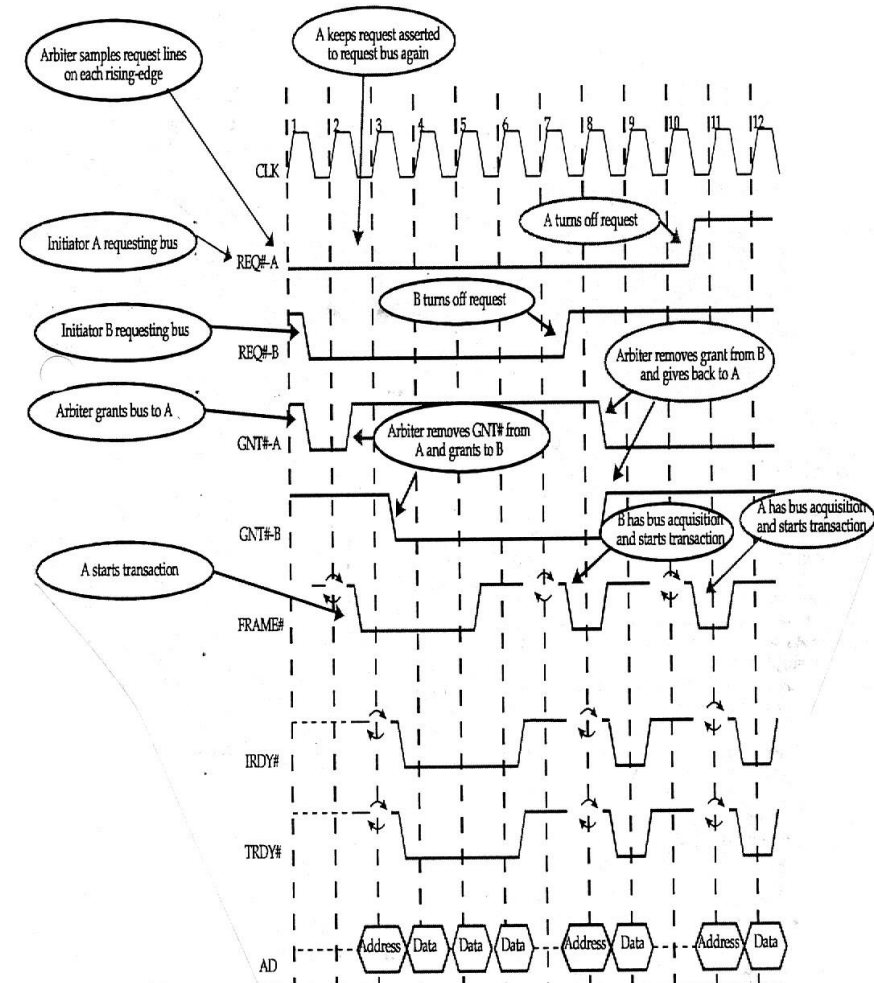
■ PCI usa arbitración centralizada paralela

- Proceso de arbitración se realiza al mismo tiempo que transferencia de datos, ganando eficiencia
- Arbitración usa esquemas de prioridad e imparcialidad (garantía de mínimos para acceso al bus) para decidir qué maestro de bus tomará el control en cada transacción



Arbitración

- Figura muestra arbitración con dos iniciadores, A y B, siendo B el más prioritario
- Importante: una vez que un iniciador toma el control del bus debe terminar su transacción, incluso aunque el árbitro le retire la cesión de bus



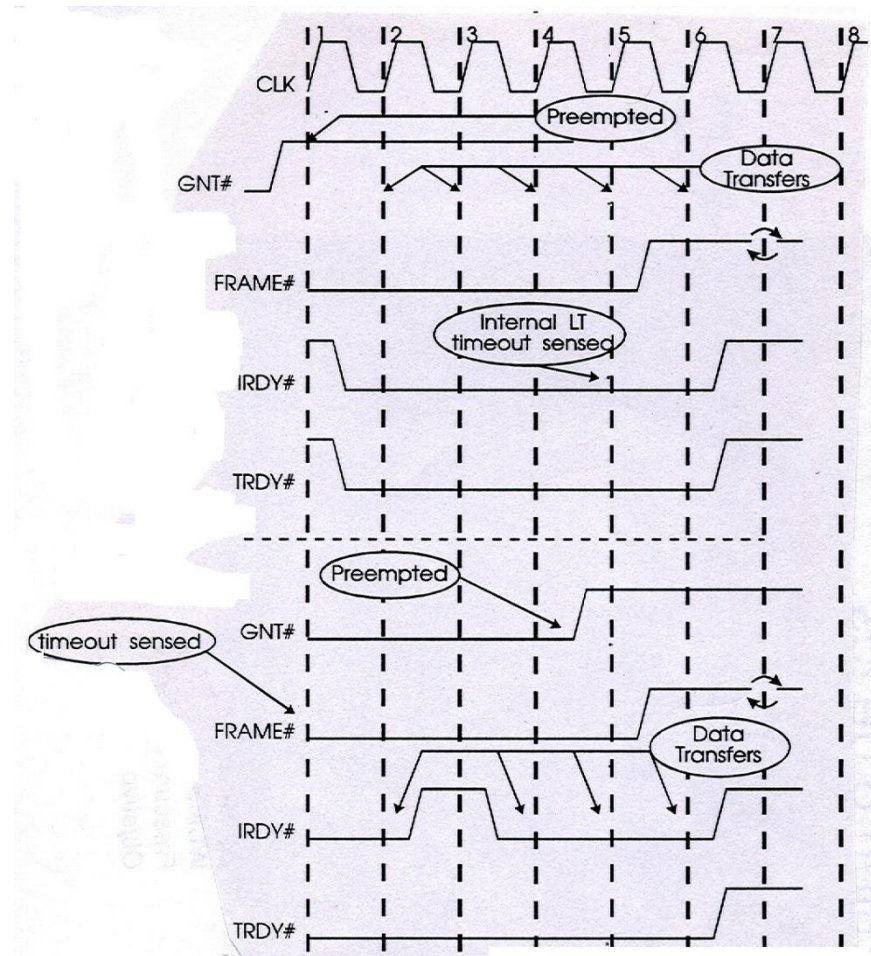
Terminaciones prematuras desde iniciador

- PCI contempla un mecanismo para evitar que una ráfaga muy larga monopolice el uso del bus por un único iniciador
- Iniciador contiene un contador de cuenta atrás, llamado *Master Latency Timer* (LT)
 - Latency Timer (LT) se carga a un valor preconfigurado al tomar el control del bus
 - En cada fase de datos se decrementa en 1 el LT
 - Cuando LT llega a 0, iniciador debe terminar la transacción tan pronto el árbitro le quite la cesión de bus, o inmediatamente si ya se la había quitado

Terminaciones prematuras desde iniciador

■ Figura muestra operación del LT

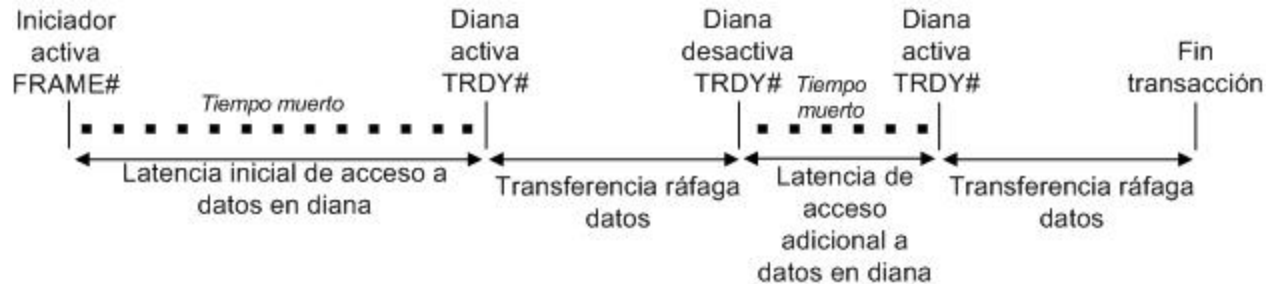
- En el primer caso, el árbitro retira la cesión de bus, pero iniciador no cede el bus hasta que no expira su LT
- En el segundo caso el iniciador cede el bus inmediatamente tras serle retirada la cesión, porque el LT ya había expirado previamente



Pérdidas de eficiencia por latencias en diana

- Llamamos *latencias* a los tiempos muertos durante una transacción de bus
 - Debidos a que un dispositivo (iniciador o diana) está esperando al otro
 - Si iniciador retiene el bus durante una latencia, se pierde eficiencia en el sistema
- Formas de evitar estas pérdidas de eficiencia:
 - Evitar, si es posible, la aparición de latencias
 - Liberar rápidamente el bus si se produce una latencia de larga duración

Pérdidas de eficiencia por latencias en diana



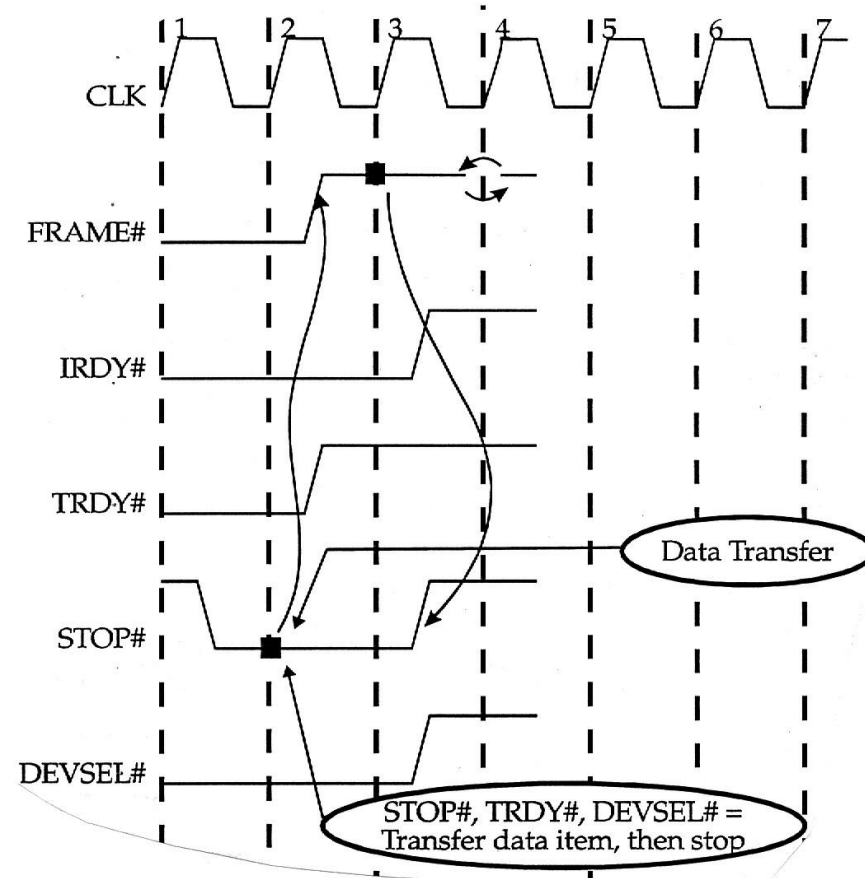
- Figura muestra latencias debidas a una diana en transacción de bus PCI
- Puede necesitarse un tiempo importante al principio para preparar buffer inicial de datos en diana
 - Tiempo usado para obtener datos de la media física (lectura) o preparar buffer en memoria (escritura)
- Adicionalmente, pueden necesitarse tiempos adicionales de espera durante la transferencia de datos
 - Tiempo para pasar nuevos datos desde media física a buffer (lectura) o para guardar contenido del buffer a media física (escritura)

Pérdidas de eficiencia por latencias en diana

- En un bus bloqueante, iniciador debe retener bus durante tiempos muertos
 - Ningún otro dispositivo podrá usar el bus, aunque esté inactivo
 - Se desperdicia tiempo, y se penaliza el rendimiento global del sistema
- Solución: usar transacciones divididas, y hacer que iniciador libere el bus durante los tiempos muertos
 - Diana debe señalar explícitamente a iniciador la necesidad de liberar el bus
 - *Desconexión*: Liberación del bus por el iniciador a petición de la diana
 - *Reconexión*: Iniciador comienza una nueva transacción de bus, retomando la transferencia de datos en el punto donde se había quedado
 - Requiere una nueva fase de direccionamiento
- Concepto de “desconexión” y “reconexión”, a nivel de comandos, también existe en el interfaz SCSI

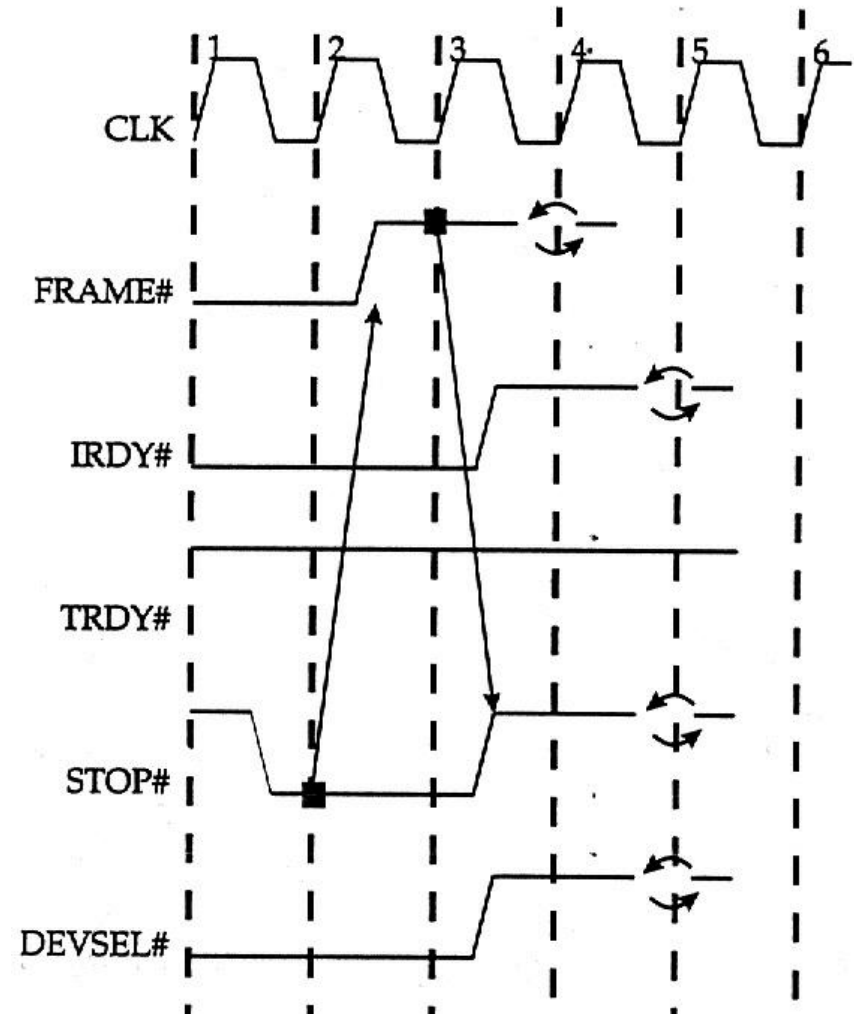
Terminaciones prematuras desde diana

- Desconexión es terminación prematura cuando ya se ha iniciado transferencia de datos
- Diana señala necesidad de desconexión activando señal STOP#
- Cuando iniciador ve activarse STOP#, transfiere un último ítem de datos y libera el bus
- Iniciador responsable de, pasado un tiempo > 8 CLK, volver a solicitar el bus y retomar transacción desde donde se quedó



Terminaciones prematuras desde diana

- Reintento es terminación prematura sin que se llegue a producir transferencia de datos
- Diana activa STOP# para solicitar desconexión debida a latencia inicial
 - No llegan a transferirse datos
- Iniciador se limita a liberar rápidamente el bus y volver a repetir la misma transacción unos pocos ciclos después
 - Más eficiente que bloqueo, pues permite a otros iniciadores usar el bus
 - No es solución óptima, porque iniciador volverá a intentar la transacción sin saber si la diana ya tiene listos los datos



Autoconfiguración PCI: Plug & Play

- Para que los periféricos de un ordenador funcionen sin conflictos, es necesario asignar de forma ordenada:
 - Líneas (IRQs) y vectores de interrupción
 - Puertos de E/S
 - Direcciones de E/S mapeadas por memoria
 - Recursos de DMA
- Antes de la aparición del bus PCI era necesario hacer manualmente la asignación de estos recursos
 - Precisaba manipular jumpers o interruptores en los periféricos
 - Requería experiencia y familiaridad con la arquitectura del ordenador
 - Por tanto, para el común de los usuarios, era una frecuente fuente de conflictos hardware, de difícil diagnóstico

Autoconfiguración PCI: Plug & Play

- Para solucionar este problema, estándar PCI introdujo un mecanismo de autoconfiguración
- Objetivo: Asignación automatizada de recursos, realizada por el propio ordenador en forma libre de conflictos
 - Realizada durante el arranque del ordenador, tras encenderlo o reiniciarlo
 - Ordenador hace una búsqueda sistemática para localizar todos los dispositivos PCI del sistema, y les va asignando recursos conforme los encuentra
 - No requiere intervención ninguna del usuario
- Como estrategia de márketing, a este mecanismo de autoconfiguración se le llamó *Plug & Play*

Autoconfiguración PCI: Plug & Play

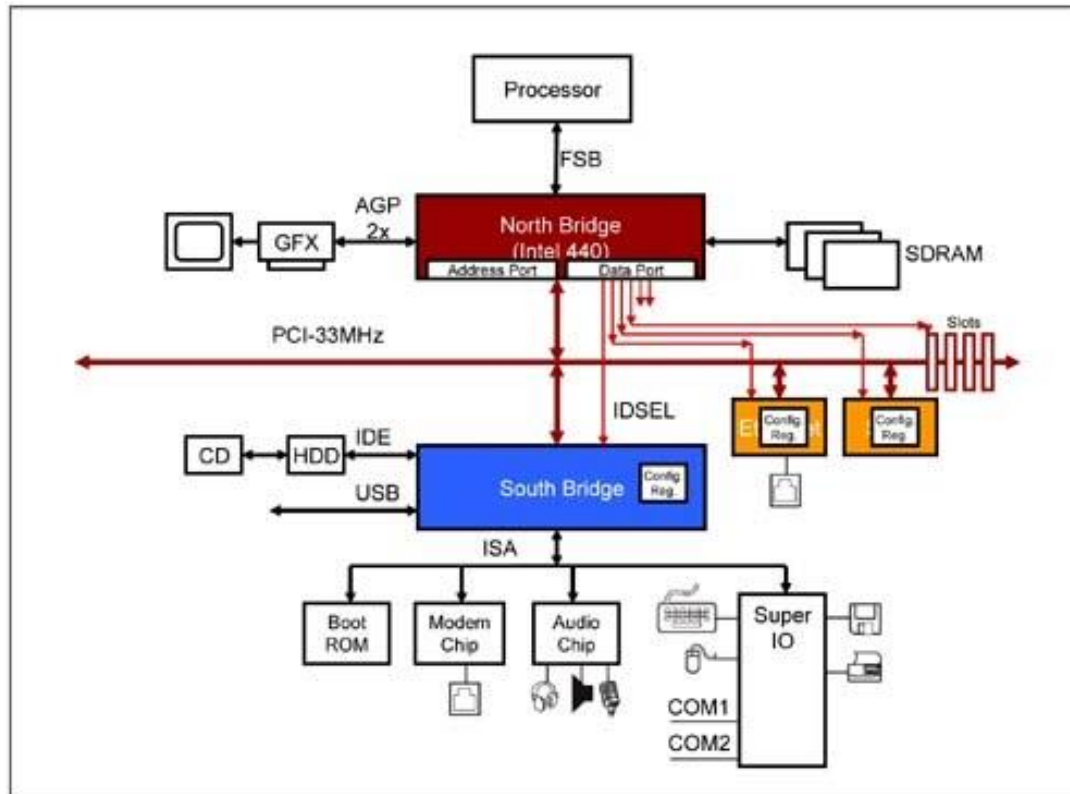
- La autoconfiguración del bus PCI se basa en tres elementos fundamentales:
 - 1) La existencia en cada dispositivo PCI de un espacio individual de configuración
 - 2) Las transacciones de configuración PCI, que proporcionan:
 - el mecanismo de búsqueda de dispositivos, y
 - el mecanismo de acceso al espacio de configuración de cada dispositivo
 - 3) El uso de una BIOS Plug & Play, que proporciona el acceso por software a las funciones de autoconfiguración

Espacio de configuración

- Mecanismo de autoconfiguración requiere que cada dispositivo PCI implemente un cierto número de registros, dedicados a contener información de configuración
 - Acceso a estos registros desde mecanismo de autoconfiguración debe, obviamente, ser independiente de que exista una asignación de recursos y direcciones de E/S
- Solución adoptada fue crear en cada dispositivo un espacio de direcciones individual, dedicado exclusivamente a registros de configuración
 - Este espacio de direcciones contiene un máximo de 256 bytes
 - Organizado en hasta 64 dobles palabras (4 bytes/doble palabra)

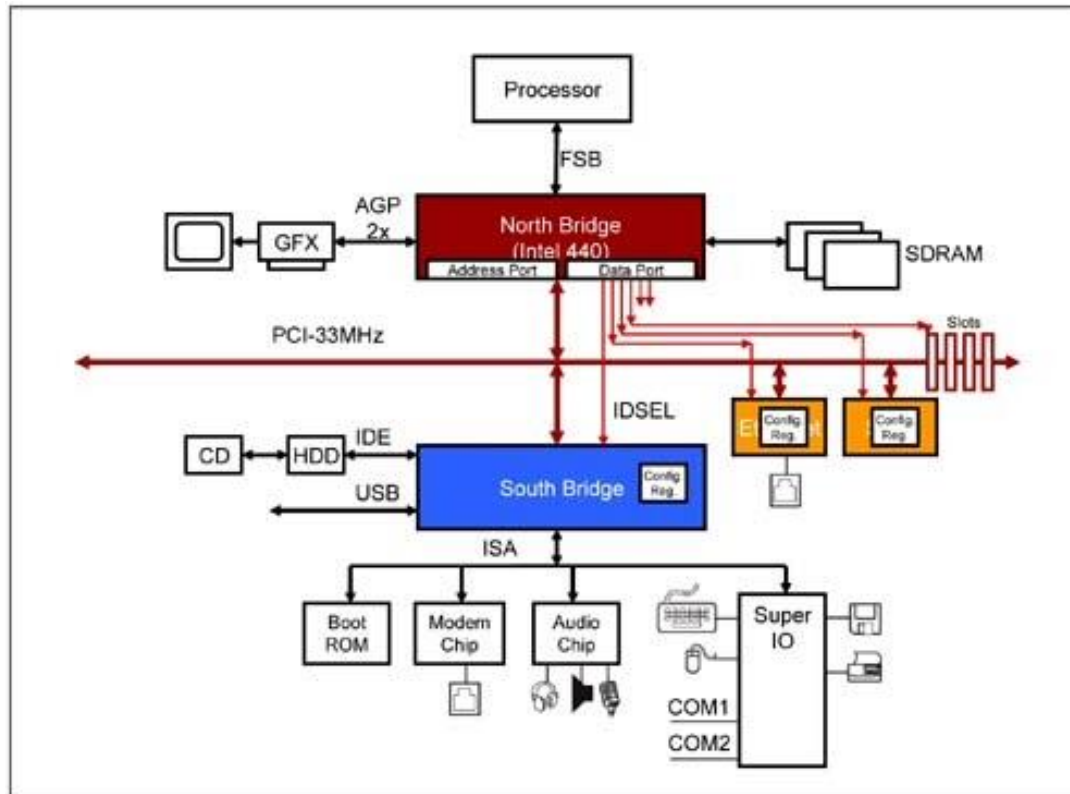
Espacio de configuración

- **Espacio de configuración es accedido activando la línea IDSEL**
 - Cada dispositivo PCI tiene su propia línea de selección IDSEL
 - Líneas IDSEL son activadas desde puente norte (o desde un puente PCI-PCI)

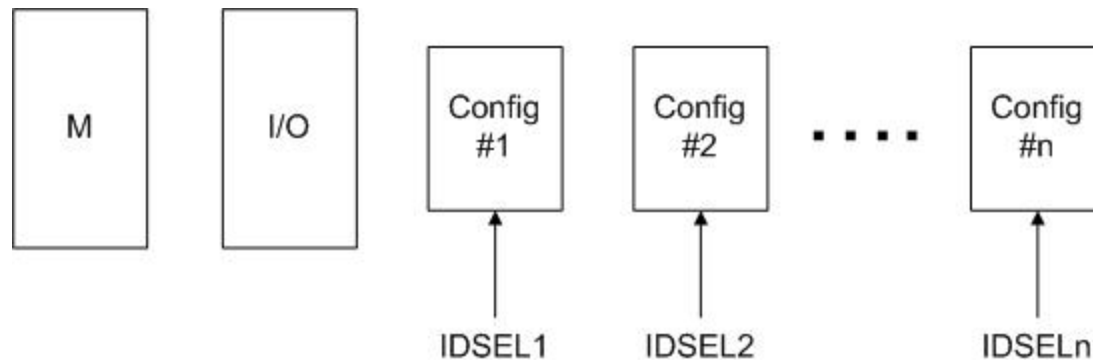


Espacio de configuración

- Existen dos posibles implementaciones de las líneas IDSEL
 - Líneas IDSEL separadas, enrutadas en la placa madre
 - Las líneas AD[31:16], combinadas con el código C/BE#[3:0] que indica transacción de configuración

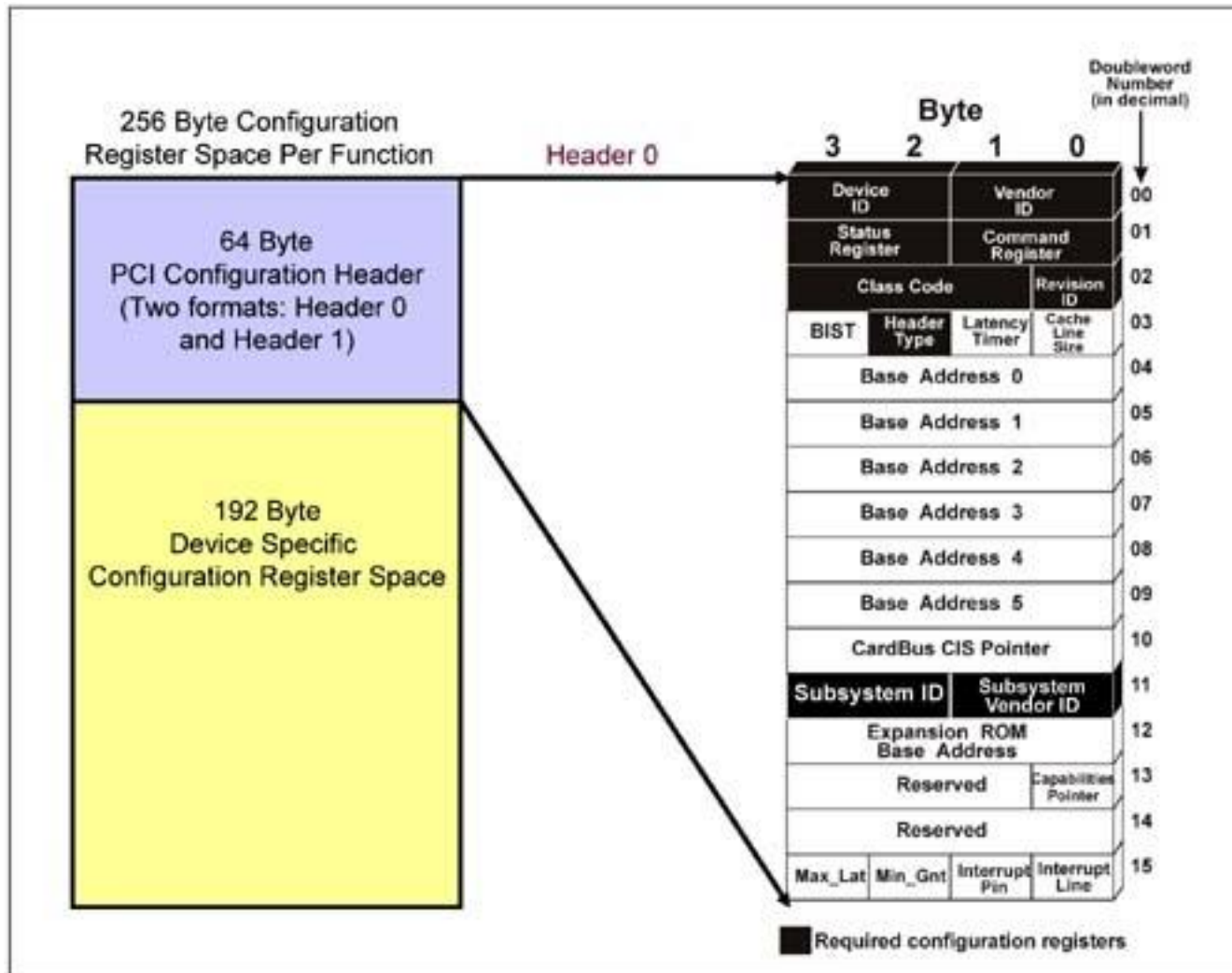


Espacio de configuración



- En sistema PCI con n dispositivos existirán, por tanto, tres tipos distintos de espacios de direcciones:
 - Memoria: Direcciones accesibles con transacciones de lectura o escritura de memoria; su tamaño es de 2^{32} bytes (4 Gbytes), o de 2^{64} bytes
 - I/O: Direcciones accesibles con transacciones de lectura o escritura de E/S; su tamaño es de 2^{32} bytes (4 Gbytes)
 - Configuración: n espacios distintos de direcciones, cada uno con hasta 256 bytes. Direcciones accesibles con transacciones de lectura o escritura de configuración, siempre que se active al mismo tiempo la línea IDSEL específica del dispositivo a cuyo espacio de configuración se quiere acceder

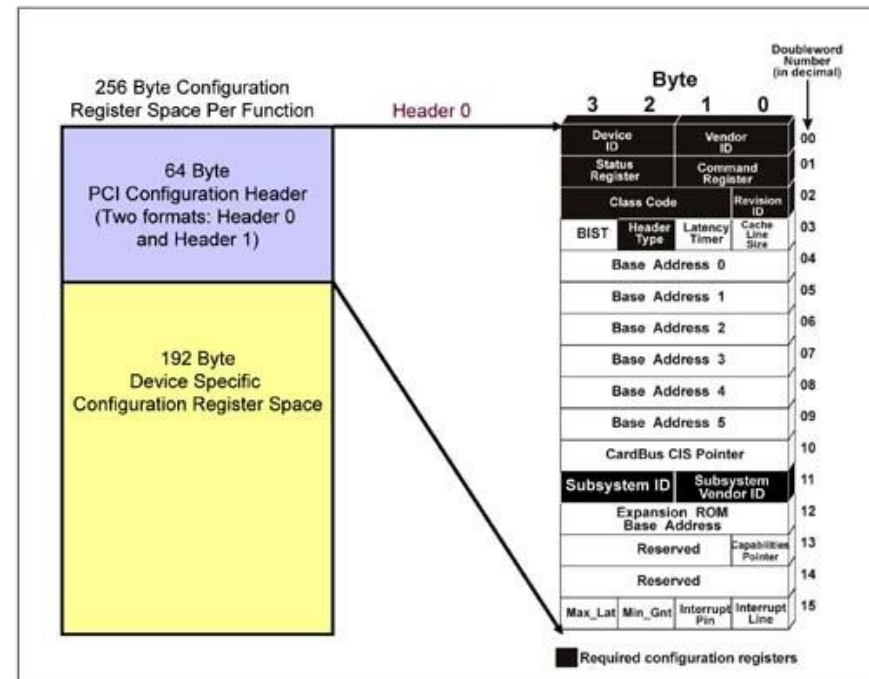
Espacio de configuración



Espacio de configuración

Espacio de configuración de cada dispositivo está organizado en:

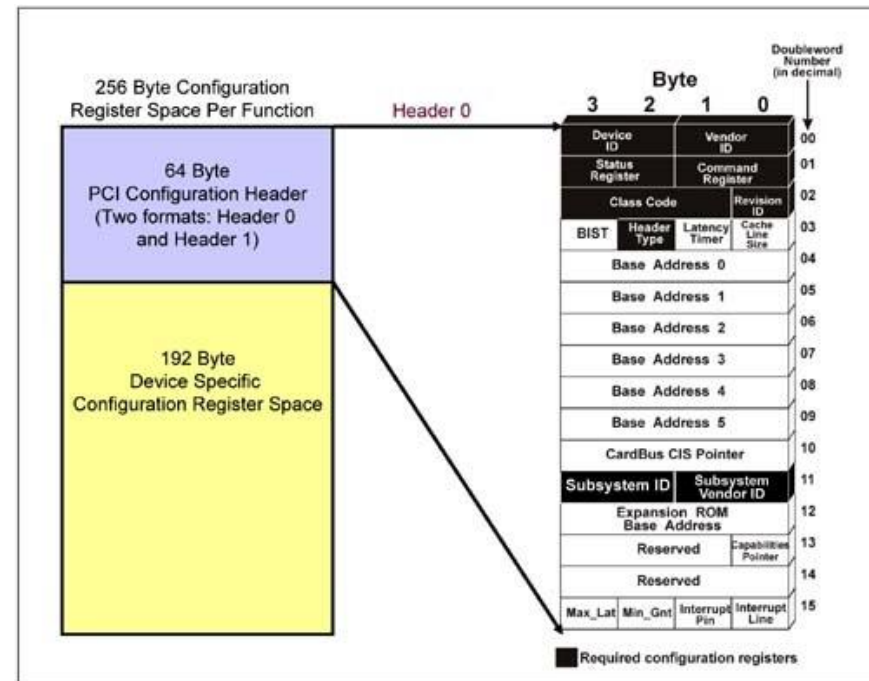
- 1) Una cabecera de hasta 16 doubles palabras (64 bytes), llamada *PCI Configuration Header*, con los registros que contienen la información crítica para identificar al dispositivo, su funcionalidad y sus requerimientos de recursos
- 2) Un espacio opcional de hasta 48 doubles palabras (192 bytes) con información de configuración específica del dispositivo, que será usada por su driver



Espacio de configuración

Los registros más relevantes son:

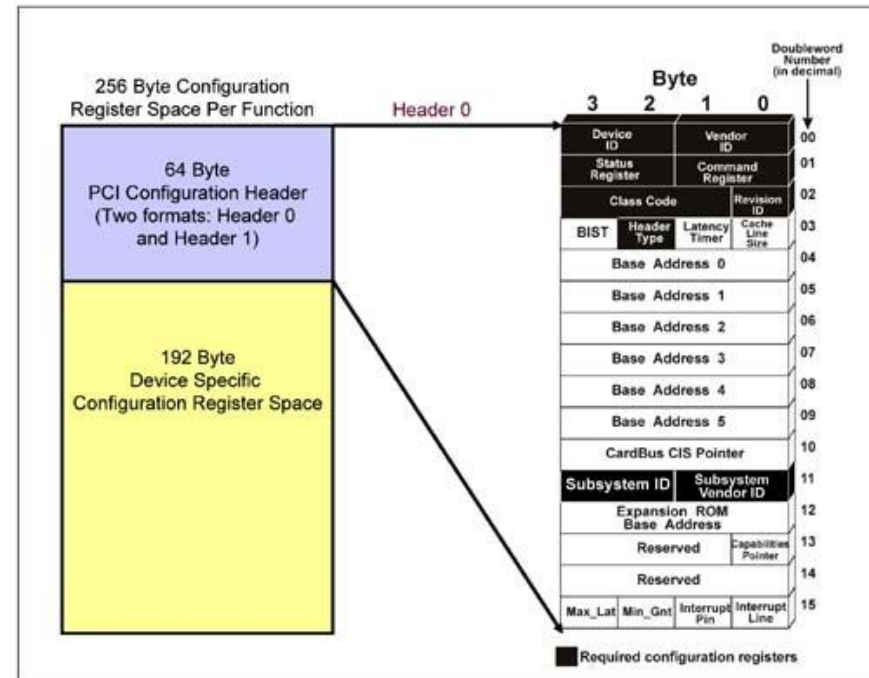
- Vendor ID, Device ID, Revision ID: Signaturas que identifican de forma única al fabricante, modelo y versión del dispositivo
- Class code: Código que identifica el tipo de funcionalidad del dispositivo (p.e, tarjeta de red Ethernet, controladora de disco SCSI, etc)
- Latency Timer, Max Lat, Min Gnt: Para dispositivos con capacidad de operar como maestros de bus, permiten configurar el sistema de arbitración y DMA (definiendo la prioridad del dispositivo, y la longitud mínima de ráfaga para conseguir eficiencia)



Espacio de configuración

Los registros más relevantes son:

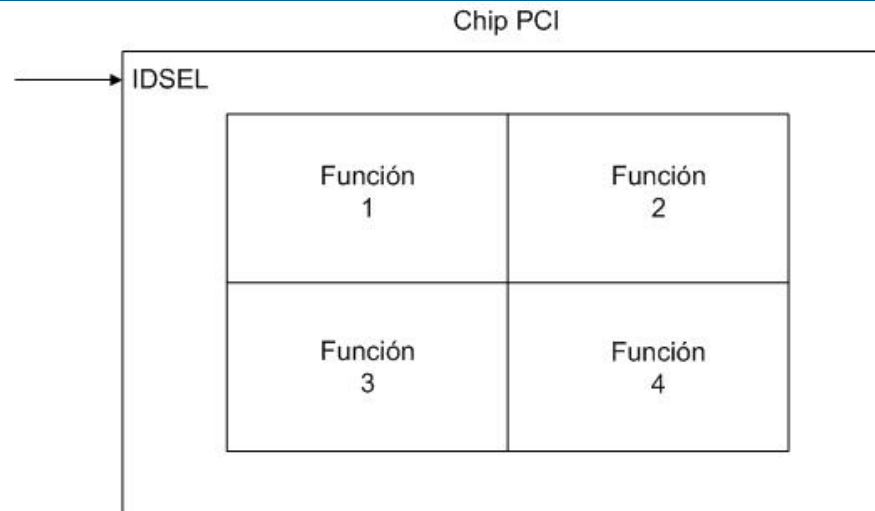
- Interrupt Pin, Interrupt Line: Permiten configurar si el dispositivo genera interrupciones (y qué línea PCI usará para hacerlo), así como el vector de interrupción (petición IRQ) que se le ha asignado
- Base Address x: A través de estos registros el dispositivo informa al sistema de sus requerimientos de puertos de E/S o de direcciones de E/S mapeada por memoria



Transacción de configuración

- El espacio de configuración de un dispositivo PCI sólo puede ser accedido a través de una transacción de configuración
 - Transacción de lectura o escritura, identificada por código en bits C/BE#[3:0] y activación de IDSEL
 - Consta de una única fase de datos, en que se leerá o escribirá una doble palabra completa del espacio de configuración
 - En su fase de direccionamiento, la transacción de configuración debe proporcionar un identificador no ambiguo del dispositivo a acceder, para garantizar la activación de la señal IDSEL correcta

Transacción de configuración



- Identificación dispositivo PCI destino de transacción requiere considerar que un mismo dispositivo físico PCI puede tener implementada más de una función
 - Cada función tiene su propio espacio de configuración
 - Activación de IDSEL habilita el acceso a la vez a todas las funciones
 - Necesario, por tanto, especificar el número de función destino

Transacción de configuración

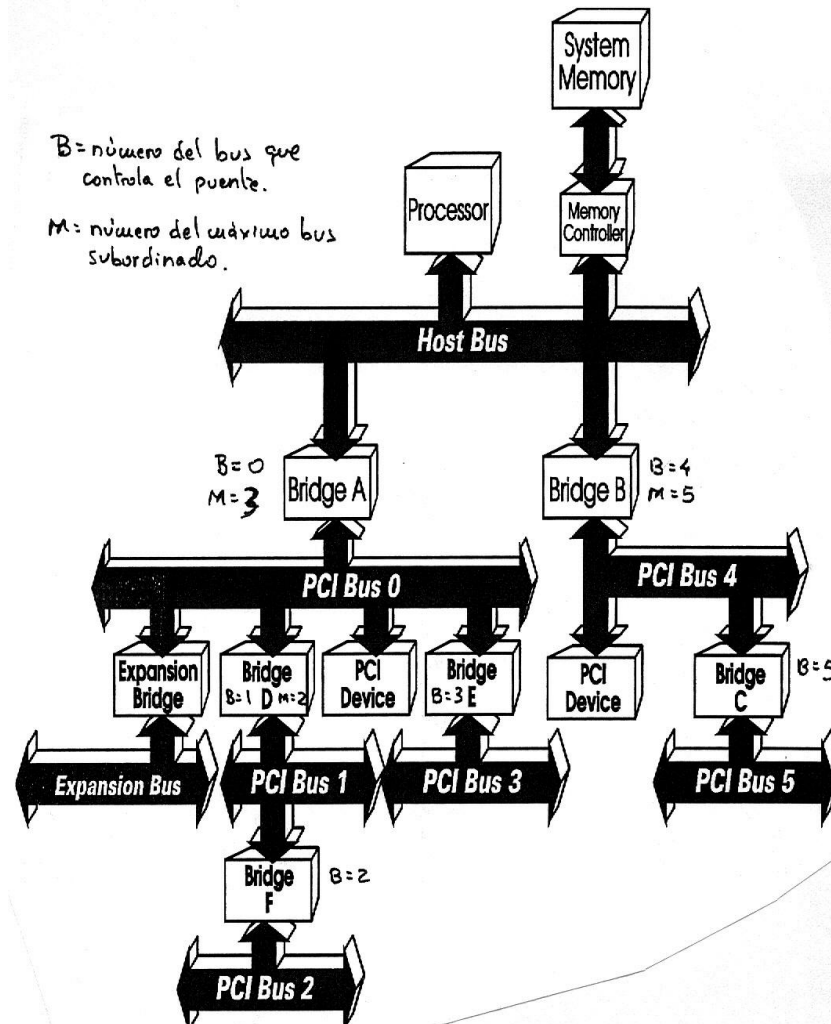
- Identificación dispositivo PCI debe también considerar que bus PCI es jerárquico
 - Pueden existir múltiples buses PCI en un mismo sistema, interconectados en cascada
 - ID de dispositivo (número de 0 a 31) es, obligatoriamente, único dentro de un bus PCI
 - Sin embargo, un mismo ID se puede repetir en distintos buses
- Por tanto, la identificación completa del dispositivo PCI debe comprender su número de bus, número de dispositivo y número de función

Propagación del acceso de configuración

- En los sistemas PCI puede (y suele) existir una jerarquía de buses PCI interconectados
 - El paso de un bus a otro se realiza a través de dispositivos PCI especiales, llamados *punte PCI-a-PCI*
 - Cada bus tiene asignado un identificador numérico único
 - Asignación de identificadores de bus se hace durante el diseño de la placa madre
 - Asignación se realiza siguiendo un orden riguroso, recorriendo el árbol de buses en orden “profundidad primero” (*depth-first*)

Propagación del acceso de configuración

- Cada puente PCI-a-PCI guarda en sus registros el ID de su bus y el máximo ID de sus buses subordinados



Propagación del acceso de configuración

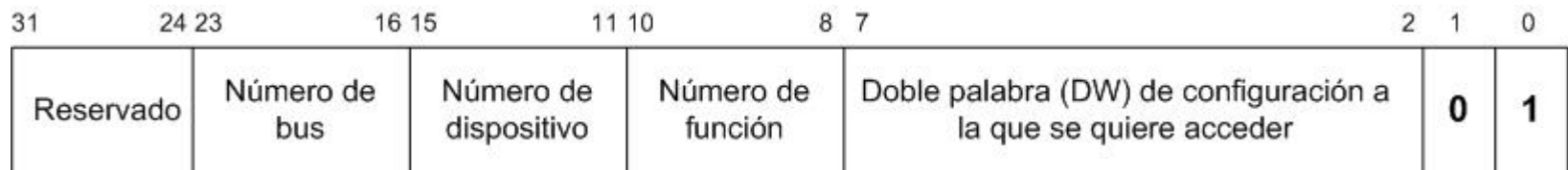
- Accesos a espacios de configuración de los dispositivos provienen de CPU, pues son generados desde la BIOS
- Propagación del acceso de configuración es, por tanto, un proceso en dos pasos:
 1. Acceso de configuración debe ser propagado desde el puente norte al puente que da acceso al bus PCI donde esté el dispositivo
 - Esta propagación se realiza usando una transacción de configuración de tipo 1
 - No es necesaria si dispositivo destino está en el bus PCI 0 (el que cuelga del propio puente norte)
 2. Una vez alcanzado el bus destino, acceso de configuración es señalizado en el bus al tiempo que puente activa línea IDSEL del dispositivo destino
 - Esta fase se realiza usando una transacción de configuración de tipo 0

Transacción de configuración de tipo uno

■ Es una transacción de lectura o escritura en la que:

- El código C/BE#[3:0] indica ciclo de lectura, o escritura, de configuración
- No hay activación de la línea IDSEL
- Los 32 bits de la dirección se descomponen en los campos mostrados en la figura

➤ El nombre de “tipo uno” procede de que el bit 0 estará fijo al valor 1



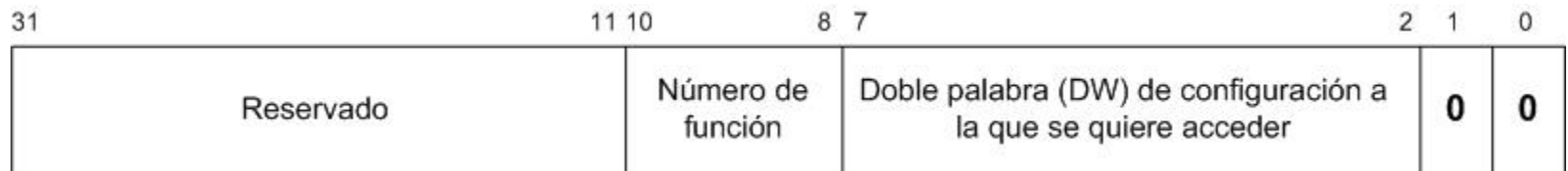
Transacción de configuración de tipo uno

- Transacciones de configuración tipo uno son ignoradas por todos los dispositivos PCI, excepto puentes PCI-a-PCI
- Al recibir transacción, puente compara ID de bus en el campo correspondiente de la dirección con el suyo propio y con el de sus buses subordinados
 - Si bus direccionado no es ninguno de ellos, puente ignora transacción
 - Si bus direccionado es uno de sus buses subordinados, el puente replica la transacción tipo uno en su propio bus
 - Así, la enruta hacia los niveles inferiores de su jerarquía de buses
 - Si bus direccionado es el controlado por el puente, intercepta la transacción
 - Usa ID de dispositivo para activar IDSEL correspondiente
 - Genera en el bus una transacción de configuración de tipo cero

Transacción de configuración de tipo cero

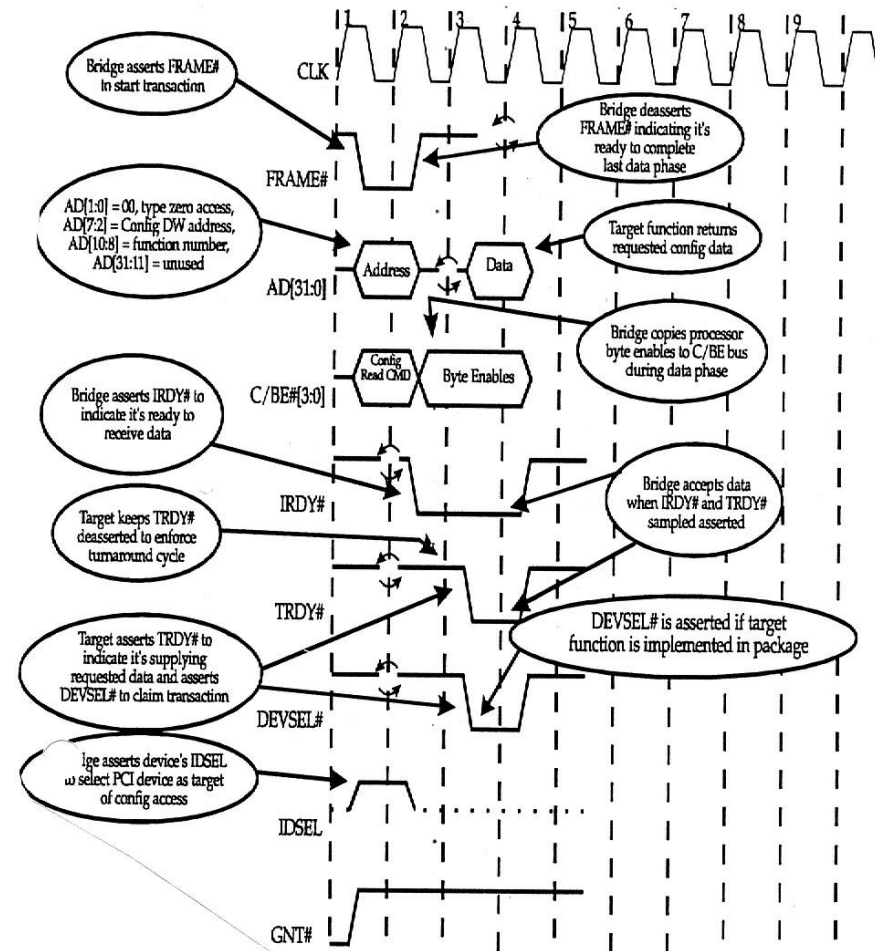
■ Es una transacción de lectura o escritura en la que:

- El código C/BE#[3:0] indica ciclo de lectura, o escritura, de configuración
- Se activa la línea IDSEL del dispositivo destino
- Los 32 bits de la dirección se descomponen en los campos mostrados en la figura



Transacción de configuración de tipo cero

- Figura muestra cronograma de una transacción de configuración de tipo cero



BIOS Plug & Play

- Acceso a espacio de configuración PCI es proporcionado por funciones implementadas en la BIOS del sistema
 - Es posible, por tanto, acceder programáticamente en cualquier momento a un espacio de configuración PCI, llamando a la función correspondiente en la BIOS
 - Estos accesos son usados por núcleo del sistema operativo para:
 - Obtener IDs de dispositivos PCI del sistema
 - Determinar de estos IDs los drivers a cargar
 - Modificar, si es necesario, la asignación de recursos hecha durante el arranque

BIOS Plug & Play

- La propia BIOS usa sus funciones Plug & Play para realizar una asignación inicial de recursos durante el POST (Power-On Self-Test)
 - Realiza un barrido sistemático de todos los buses PCI que pueden existir en el sistema
 - En cada bus, realiza un barrido sistemático de cada posible dispositivo que puede existir en el bus
 - Conforme va encontrando dispositivos, les asigna recursos (puertos de E/S, direcciones de E/S por memoria, DMA, interrupciones) de acuerdo a las necesidades declaradas por el dispositivo
 - Al terminar el barrido, el sistema conoce exactamente los dispositivos presentes, y tiene ya una asignación de recursos libre de conflictos con la que cargar el sistema operativo

BIOS Plug & Play

- Sistemas x86 mapean acceso a espacio configuración a través de puertos E/S
 - CPU escribe a Address Port dirección de transacción tipo uno
 - CPU lee o escribe a través de Data Port valor de doble palabra de configuración

