

CAPACIDADES Y FUNCIONALIDADES DEL BUS PCI

John Alexander Cortés Romero*
Edwin Camargo Bareño**
Carlos Ivan Camargo***

1. Introducción

Cuando la compañía INTEL decidió abandonar definitivamente el estándar VESA VL adoptó una opción que pudiera asegurar que el mercado no se atestara con varias permutaciones de arquitecturas peculiares de bus a buses de procesadores específicos.

Intel empezó a trabajar en el bus PCI en 1990, considerando que sus sistemas se basaban en el procesador Pentium. Luego la empresa cedió las patentes al dominio público y promovió la creación de una asociación industrial, la PCI SIG (Special Interest Group), para continuar con el desarrollo y mantener la compatibilidad de las especificaciones del PCI. El resultado es que este ha sido ampliamente adoptado y su uso se está incrementando en computadores personales, estaciones de trabajo y servidores de sistema. La versión actual (el PCI 2.2) apareció en 1998; dado que las especificaciones son de dominio público y están soportadas por una amplia banda de la industria de procesadores y periféricos, los productos PCI fabricados por compañías diferentes son compatibles entre sí.

El bus de interconexión de componentes periféricos (PCI Peripheral Component Interconnect) tiene un elevado ancho de banda con un protocolo *plug and play*, diseñado para cumplir con las más altas demandas de funcionamiento de los componentes periféricos de alto desempeño de PCs y estaciones de trabajo, y para tener gran variedad de aplicaciones. Este estándar se ha convertido rápidamente en el más aceptado en la industria de los computadores y es ampliamente utilizado para aplicaciones tales como video, sistemas de audio, tarjetas aceleradoras de gráficos, adaptadores de red, tarjetas de adquisición de datos y dispositivos de almacenamiento de datos.

* Ingeniero Electricista, Magister en Automatización Industrial Universidad Nacional de Colombia. Profesor de tiempo completo adscrito a la Facultad Tecnológica de la Universidad Distrital Francisco José de Caldas. Director del Proyecto de Investigación en Instrumentación Virtual

** Estudiante de 10º semestre de Ingeniería Electrónica de la Universidad Santo Tomas de Aquino, coinvestigador Proyecto de Investigación en Instrumentación Virtual

*** Ingeniero Electricista Universidad Nacional de Colombia, Magister en Electrónica y Computadores de la Universidad de los Andes. Profesor de tiempo completo adscrito a la Facultad Tecnológica de la Universidad Distrital Francisco José de Caldas, coinvestigador Proyecto de Investigación de Instrumentación Virtual

La especificación actual permite el uso de hasta 64 líneas de datos a 66 MHz para una velocidad de transferencia máxima de 528 Mbytes por segundo. No obstante, su elevada velocidad no sólo es la característica que le hace atractivo; el dispositivo ha sido diseñado específicamente para adaptarse económicamente a los requisitos de entrada / salida de los sistemas actuales, se implementa con muy pocos circuitos integrados y permite su interconexión con otros buses.

El desarrollo de los productos compatibles requiere un estricto cumplimiento de las especificaciones PCI (PCI Local Bus Specification); sin embargo también debe tenerse en cuenta que la continua evolución de sus especificaciones y las necesidades específicas de cada aplicación requieren de una solución flexible.

Algunas de las características del bus PCI son opcionales a la hora de implementar una aplicación dada; por esto es necesario implementar aplicaciones en dispositivos flexibles a futuros cambios. Dado los avances actuales de las técnicas de diseño digital, el uso de dispositivos lógicos programables es la solución más adecuada para la implementación de cualquier tipo de interfaz compatible PCI.

El Grupo de Investigación en Instrumentación Virtual que funciona en la Facultad Tecnológica de la Universidad Distrital F.J.C. pretende desarrollar dos instrumentos virtuales. Uno es un Osciloscopio y el otro un generador de señales. Ambos involucran un proceso de adquisición de datos por el bus PCI.

2. Principales Características del Bus PCI

CARACTERÍSTICAS	DESCRIPCIÓN
Independencia del procesador	Los componentes diseñados para el bus PCI son específicos del estándar del mismo nombre, de tal modo que se aísla el diseño del dispositivo con respecto a posibles actualizaciones del procesador
Soporte de hasta aproximadamente 80 funciones PCI por bus PCI	Una implementación típica del bus PCI soporta aproximadamente 10 cargas eléctricas, y cada uno de los dispositivos presenta una carga al bus; cada dispositivo puede contener hasta 8 funciones PCI
Soporte hasta de 256 buses	La especificación provee soporte hasta de 256 buses PCI
Bajo consumo de potencia	Uno de los mayores objetivos de diseño de las especificaciones PCI es crear un sistema de diseño que maneje la menor corriente posible
Todas las transacciones de escritura y de lectura pueden ejecutarse en modo "ráfaga" (burst mode)	Un bus PCI de 32 bits soporta una velocidad máxima de transferencia de datos de 132 Mbytes/seg. Para un bus PCI de 64 bits la velocidad máxima corresponde a 264 Mbytes/seg. Si se trabaja con una frecuencia de 66 MHz se pueden obtener velocidades de hasta 528 Mbytes/seg. para buses PCI de 64 bits.
Velocidad del bus	La revisión 2.2 da soporte para velocidades de 33 MHz y 66 MHz
Ancho del bus de 64 bits	Total definición para la extensión de 64 bits
Tiempo de acceso	60ns par una velocidad del bus de 33 MHz
Operación concurrente del bus	Los puentes (bridges) soportan total concurrencia con el bus del procesador, el bus PCI (o buses PCI) y la expansión del bus simultáneamente en uso.

1 Modo de transferencia de datos en el cual se optimizan todas las operaciones necesarias para la transacción de tal forma que se minimice el tiempo de ejecución de la transferencia.

CARACTERÍSTICAS	DESCRIPCIÓN
Soporte de bus maestro	Total soporte de maestros de bus PCI, lo cual permite el acceso total al bus, acceso a la memoria principal y dispositivos de expansión a través de los puentes PCI-PCI. Adicionalmente un maestro ² PCI puede acceder un dispositivo que reside sobre otro bus PCI de menor jerarquía
Arbitración del bus oculta	La arbitración del bus puede estar ejecutándose mientras otro maestro del bus está realizando una transferencia sobre el bus PCI
Pocos terminales	Para la implementación PCI "Target" se requieren 47 terminales; para el PCI "Initiator" se requieren 49 terminales. Esto indica un uso económico de las señales del bus
Chequeo integral de todas las transacciones	Se realiza un chequeo de la paridad sobre el bus de direcciones datos y sobre el bus de comandos de forma integral.
Tres espacios de direcciones	Memoria, entradas/salidas y configuración de direcciones
Autoconfiguración	La especificación tiene en cuenta una configuración total de los registros necesarios para soportar detección y configuración automática de dispositivos
Transparencia del software	Los "drivers" utilizan el mismo conjunto de comandos y definición de status cuando se comunican con dispositivos PCI o sus correspondientes dispositivos de expansión
Tarjetas para el bus PCI	La especificación incluye una definición de los conectores y de las tarjetas para el bus PCI

Tabla 1. Características del Bus PCI. Tomada con adaptaciones de SHAN 99, pp. 8-10

3. Arquitectura PCI

La base fundamental de la Arquitectura PCI, y en general de un computador, es el *ChipSet*. Este se encarga de servir como puente de información entre la CPU y los Buses ISA y PCI, que son los encargados de manejar la comunicación con los dispositivos de almacenamiento (discos duros, unidades de CD ROM, unidades SCSI, etc.) y los periféricos (puertos USB, seriales, de impresora, teclado, mouse, etc.).

En los nuevos diseños de computadores se encuentra un *ChipSet* compuesto por dos módulos llamados *North Bridge* y *South Bridge*. Estos dos circuitos integrados son parte fundamental de la *MotherBoard*. En la Figura 1 se pueden apreciar dos de estos dispositivos, desarrollados por la compañía de semiconductores VIA.

El dispositivo *North Bridge* es el encargado de realizar las transacciones entre la CPU y la memoria principal. Además maneja el puerto AGP, por lo cual puede incrementar la velocidad de transferencia de video, al encontrarse en un nivel más cercano al de la CPU y a la vez tener el control sobre el Bus PCI. Todo esto puede apreciarse en la Figura 2.



Figura 1. ChipSet VT82C598AT (North Bridge), VT82C586B (South Bridge)

2 El maestro y esclavo que interactúan en una transacción se denominan dentro de la especificación como "initiator" y "target" respectivamente. En este escrito se utilizarán estas dos últimas denominaciones, puesto que los autores no consideran conveniente utilizar los términos en español

La versión 2.2. del Bus PCI es un sistema que puede trabajar a una frecuencia de hasta 66MHz, por lo cual se encarga de comunicarse con dispositivos de alta velocidad tales como tarjetas de red, dispositivos SCSI, modems, tarjetas de sonido, etc., además de controlar el tráfico de información entre el *South Bridge* y el *North Bridge*, lo cual lo hace la columna vertebral del computador.

El dispositivo denominado *South Bridge* es el encargado de manejar los puertos de mediana velocidad, como los IDE y USB; también se encarga de controlar la interfaz con el Bus ISA, que es uno de los más antiguos, el cual maneja los periféricos (mouse, teclado, impresoras, floppy disks y en algunos casos los dispositivos de multimedia). Cabe decir que el Bus ISA no aparece en los nuevos diseños de MotherBoards, ya que restan velocidad al computador.

4. Señales de la interfaz PCI

La especificación define 49 líneas de señal obligatorias del PCI, divididas en los siguientes grupos funcionales:

- **Terminales de sistema:** constituidas por los terminales de reloj y de inicio (reset)
- **Terminales de direcciones y datos:** incluye 32 líneas para datos y direcciones multiplexadas en el tiempo. Las otras líneas del grupo se utilizan para interpretar y validar las señales correspondientes a los datos y las direcciones
- **Terminales de control de la interfaz:** controlan la temporización de las transferencias y proporcionan coordinación entre los que las inician y los destinatarios
- **Terminales de arbitraje:** a diferencia de las otras líneas de señal del PCI, éstas no son líneas compartidas. Cada maestro del PCI tiene su propio par de líneas que lo conectan directamente al árbitro del bus PCI
- **Terminales para señales de error:** utilizadas para indicar errores de paridad u otros
- **Terminales de interrupción:** para los dispositivos PCI que deben generar peticiones de servicio. Igual que los terminales de arbitraje, no son líneas compartidas sino que cada dispositivo PCI tiene su propia línea o líneas de petición de interrupción a un controlador de interrupciones
- **Terminales de soporte de caché:** son necesarias para permitir memorias caché en el bus PCI asociadas a un procesador o a otro dispositivo. Estos terminales permiten el uso de protocolos de coherencia de caché de sondeo de bus (snoopy cache)
- **Terminales de ampliación a bus de 64 bits:** incluyen 32 líneas multiplexadas en el tiempo para direcciones y datos, combinadas con las líneas obligatorias de dirección y datos para constituir un bus de direcciones y datos de 64 bits. Hay otras líneas de este grupo que se utilizan para interpretar y validar las líneas de datos y direcciones. Por último, hay dos líneas que permiten que dos dispositivos PCI se pongan de acuerdo para usar los 64 bits
- **Terminales de Test (JTAG/Boundary Scan):** estas señales se ajustan al estándar IEEE 1149.1 para la definición de procedimientos de test.

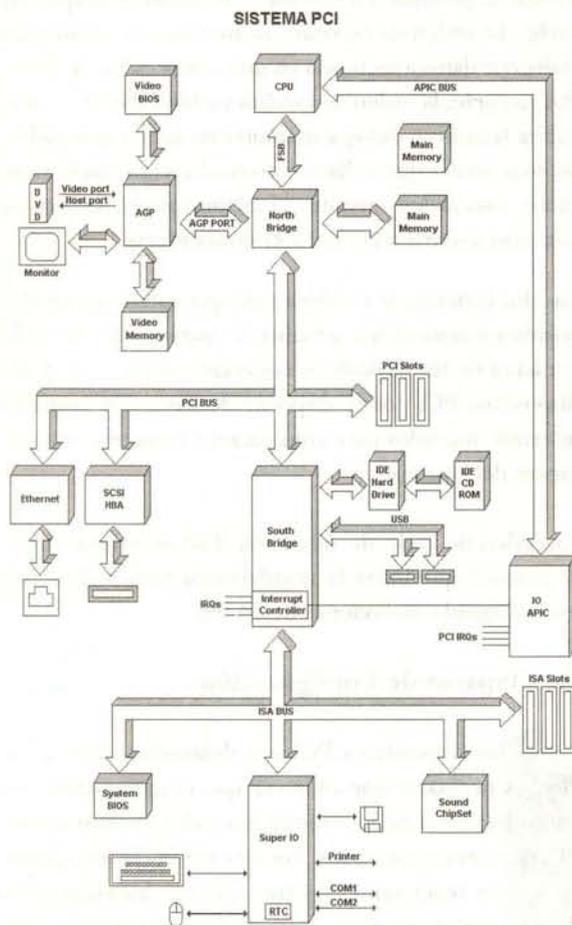


Figura 2. Arquitectura PCI

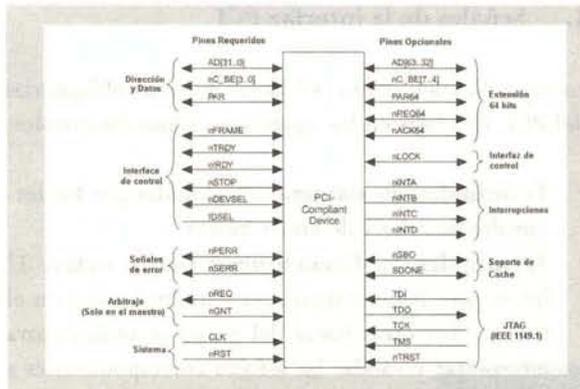


Figura 3. Señales de la Interfaz PCI

5. Comandos del PCI

La actividad del bus consiste en transferencias entre dos elementos. Se denomina *maestro* al que inicia la transacción. Cuando un maestro adquiere el control del bus determina el tipo de transferencia que se producirá a continuación. Durante la fase de direccionamiento de transferencia se utilizan las líneas C/BE para indicar el tipo de transferencia.

Los tipos de órdenes son:

- Reconocimiento de interrupción
- Ciclo especial
- Lectura de E/S
- Escritura en E/S
- Lectura de memoria
- Lectura de línea de memoria
- Lectura múltiple de memoria
- Escritura en memoria
- Escritura e invalidación de memoria
- Lectura de configuración
- Escritura de configuración
- Ciclo de dirección dual

El reconocimiento de interrupción es una orden de lectura proporcionada por el dispositivo que actúa como controlador de interrupciones en el bus PCI. Las líneas de direcciones no se utilizan en la fase de direccionamiento, y las de *byte activo* («*byte enables*») indican el tamaño del identificador de interrupción a devolver.

La orden de ciclo especial se utiliza para iniciar la difusión de un mensaje a uno o más destinos. Los órdenes de lectura de E/S y escritura en E/S se utilizan para intercambiar datos entre el módulo que inicia la transferencia y un controlador de E/S. Cada dispositivo de E/S tiene su propio espacio de direcciones, y las líneas de direcciones se utilizan para indicar un dispositivo concreto y para especificar los datos a transferir a, o desde, ese dispositivo.

Las órdenes de lectura y escritura en memoria se utilizan para especificar la transferencia de una secuencia de datos, utilizando uno o más ciclos de reloj. La interpretación de estas órdenes depende de si el controlador de memoria del bus PCI utiliza el protocolo PCI para transferencias entre memoria y caché o no. Si lo utiliza, la transferencia de datos a y desde la memoria normalmente se produce en términos de líneas o bloques de caché. La orden de escritura en memoria se utiliza para transferir datos a memoria en uno o más ciclos de datos. Por su parte, la orden de escritura e invalidación de memoria transfiere datos a memoria en uno o más ciclos; además, indica que se ha escrito en al menos una línea de caché. Esta orden permite el funcionamiento de la caché con post-escritura (*write back*) en memoria.

Las dos órdenes de configuración permiten que un dispositivo maestro lea y actualice los parámetros de configuración de un dispositivo conectado al bus PCI. Cada dispositivo PCI puede disponer de hasta 256 registros internos utilizados para configurarse durante la inicialización del sistema.

La orden de ciclo de dirección dual se utiliza por el dispositivo que inicia la transferencia para indicar que está utilizando direcciones de 64 bits.

6. Espacio de Configuración

Como los dispositivos PCI son denominados *Plug and Play*, es necesario que cada vez que el computador sea encendido o reiniciado reconozca todos los dispositivos PCI que tiene conectados. Por esta razón estos dispositivos deben tener unos registros que son los encargados de proporcionar información de la configuración, evitando de esta forma las configuraciones manuales que se

hacen por medio de interruptores (*jumpers*), en la forma como funcionan las tarjetas ISA. Los registros de configuración son mostrados en la Figura 4.



Figura 4. Registros de configuración

En la Tabla 2 se explican los registros de configuración más importantes.

Registro	Descripción
Vendor ID	Indica el fabricante del dispositivo
Device ID	Identificación del dispositivo que hace el SIG (Special Interest Group)
Command	Determina la respuesta del dispositivo a los diferentes comandos
Status	Guarda eventos y errores que sucedan en el Bus PCI
Revision ID	Especifica la revisión del dispositivo
Class code	Usado para identificar la función del dispositivo
Header type	Indica si el dispositivo tiene múltiples funciones
Base address Register (BAR)	Informa al Target la dirección existente entre el espacio de direcciones del dispositivo

Tabla 2. Registros de configuración PCI

A continuación se muestra un ejemplo de algunos de los datos que son almacenados en estos registros, los cuales fueron adquiridos con el programa PCISCAN.

PCI Scan 1.0 PCI Device Information file

Vendor ID	10EC
Product ID	8029
BUS Number	00
DEV Number	0D
FUNC Number	00
Vendor Name	Realtek Semiconductor
Device Name	RT8029(AS) 10Mb Ethernet Adapter
Base Class	PCI Network controller
Sub Class	Ethernet controller

Tabla 3. Registros de configuración de una tarjeta de red

7. Transferencia de Datos en el Bus PCI

Para explicar la forma de transferir datos sobre el Bus PCI se toma el caso en el cual dos dispositivos PCI diferentes, denominados A y B, solicitan el acceso al bus. Cuando el bus PCI es requerido por dos o más dispositivos simultáneamente es necesario hacer un control de tráfico sobre él; este procedimiento de control es realizado por un dispositivo denominado Árbitro (*Arbiter*). En la siguiente explicación las señales que terminan con el símbolo # son aquellas que son activadas en bajo.

El maestro informa al árbitro que va a utilizar el bus activando su señal REQ#; la forma como el árbitro otorga el bus a un maestro es activando su señal GNT#. No puede darse el caso que el árbitro seleccione dos dispositivos al tiempo, lo cual sí ocurre en los dispositivos ISA, en los cuales es posible dar una misma dirección a dos dispositivos diferentes por medio de "jumpers" generando colisión de información. Así, todos los dispositivos PCI deben tener dos señales que vayan directamente al árbitro, tal como se ilustra en la Figura 5.

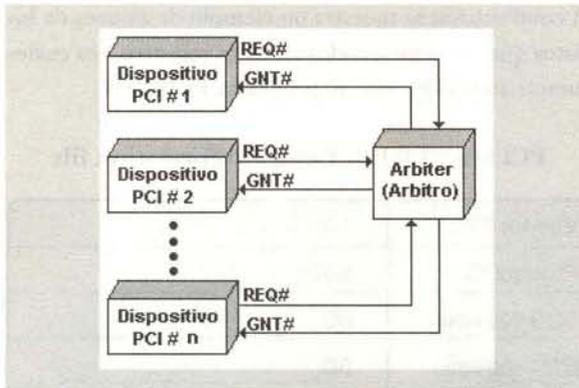


Figura 5. Esquema del Control de Tráfico de Información en el Bus PCI

En el ejemplo el dispositivo A requiere el Bus para hacer dos transacciones. La primera consta de tres fases de escritura de datos y la segunda de una sola fase de datos. El dispositivo B requiere hacer una transacción con una sola fase de datos. El diagrama de tiempos se muestra en la Figura 6; es importante notar que todas las señales son muestreadas con el flanco de subida del reloj (CLK).

El ejemplo de control de tráfico de la Figura 6 procede de la siguiente manera:

- **Pulso de reloj 1:** el árbitro nota que el *Initiator A* tiene su señal $REQ\#$ activada y, antes de que ocurra el siguiente pulso de reloj, activa la señal de $GNT\# A$ indicándole que tiene el control sobre el bus. En el mismo periodo de reloj el *Initiator B* activa su señal $REQ\#$ indicándole al árbitro que quiere utilizar el Bus
- **Pulso de reloj 2:** en el segundo flanco de subida del reloj el *Initiator A* muestrea la señal de $GNT\#$ activada, así como las señales $IRDY\#$ y $FRAME\#$ desactivadas, lo cual indica que el Bus está desocupado. En el periodo de reloj 2 el *Initiator A* activa su señal $FRAME\#$ para indicar que va a iniciar la transacción, colocando en el bus de datos y dirección (A/D Bus) la dirección inicial. Dado que el *Initiator A* continúa con su señal $REQ\#$ activada y el *Initiator B* también tiene activada su señal $GNT\#$, el árbitro debe asignar el bus al dispositivo que tenga mayor prioridad. En este caso se asume que el *Initiator B* tiene la prioridad sobre A, por lo cual en el periodo de reloj 2 el árbitro desactiva la señal $GNT\#A$ indi-

cando que cuando el *Initiator A* termine la transacción actual el bus será asignado a otro dispositivo.

- **Pulso de reloj 3:** en el tercer flanco de subida el árbitro activa la señal $GNT\#$ del dispositivo B indicando que él será el próximo propietario del Bus. En este periodo de reloj se activan las señales $IRDY\#$ y $TRDY\#$ indicando que tanto el *Initiator* como el *Target* están listos para llevar a cabo la siguiente fase de datos.
- **Pulsos de reloj 4 y 5:** en ellos se llevan a cabo las dos primeras fases de datos del dispositivo A. En el periodo de reloj 5 se desactiva la señal $FRAME\#$ para indicar que la próxima fase de datos (en este caso la tercera) será la última.
- **Pulso de reloj 6:** en el sexto flanco de subida del reloj el *Initiator A* completa su última fase de datos y en el mismo periodo de reloj desactiva su señal $IRDY\#$ indicando que está listo para desocupar el bus
- **Pulso de reloj 7:** en el séptimo flanco de subida del reloj el *Initiator B* muestrea las señales $IRDY\#$ y $FRAME\#$ desactivadas. Por lo tanto en el mismo periodo de reloj procede a activar su señal $FRAME\#$ y a colocar la dirección de inicio en el bus de datos y dirección, indicando que va a comenzar con la transacción. El árbitro desactiva la señal $GNT\# B$ indicando al *Initiator B* que cuando termine la transacción actual será despojado del Bus
- **Pulso de reloj 8:** en el octavo periodo de reloj el árbitro activa nuevamente la señal $GNT\# A$, con lo cual el *Initiator A* puede realizar su segunda transacción una vez el *Initiator B* libere el Bus. En el mismo periodo el *Initiator B* desactiva su señal $FRAME\#$, indicando que será su primera y única fase de datos. Además, las señales de $IRDY\#$ y $TRDY\#$ son activadas para indicar que tanto el *Initiator* como el *Target* están listos para llevar a cabo la fase de datos
- **Pulso de reloj 9:** en el noveno flanco de subida del reloj el *Initiator B* lleva a cabo su única fase de datos y en el mismo periodo de reloj desactiva su señal $IRDY\#$ indicando que va a desocupar el Bus

- **Pulso de reloj 10:** en el décimo flanco de subida del reloj el *Initiator A* nota que las señales *IRDY#* y *FRAME#* están desactivadas y toma posesión del bus para llevar a cabo la última transacción, la cual consiste en una sola fase de datos.

8. Conclusiones

El bus PCI se usa ampliamente para la comunicación con dispositivos periféricos, debido a su gran ancho de banda. Para aplicaciones que requieran altas especificaciones de funcionamiento éste se presenta como la mejor opción de interfaz a implementar.

El diseño de una interfaz PCI puede tomar varios meses-ingenero, trabajando de tiempo completo. Para su implementación se requiere de un grupo interdisciplinario con buenos conocimientos en conjunto de arquitectura de computadores, diseño digital y, preferiblemente, de técnicas avanzadas de diseño digital; además, de amplios conocimientos de sistemas operativos, programación a alto y bajo nivel, experiencia en programación de drivers software, instrumentación electrónica y acondicionamiento de señales.

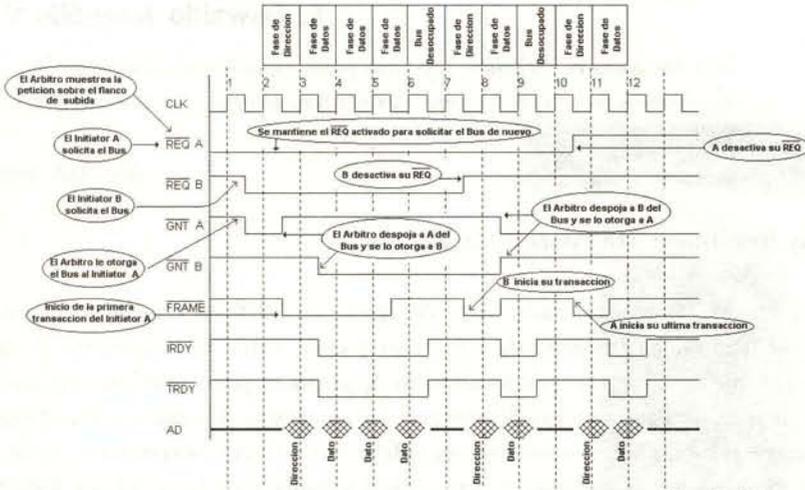


Figura 6. Diagrama de tiempos de una transacción PCI

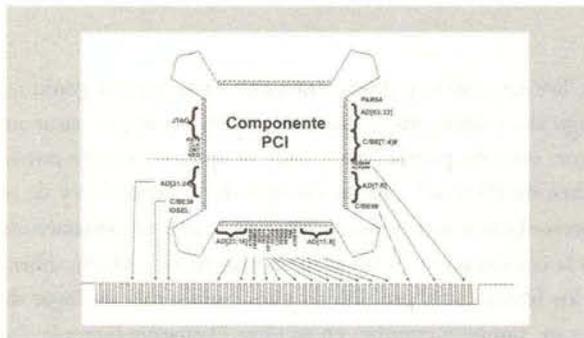


Figura 7. Disposición de terminales de salida recomendados

REFERENCIAS BIBLIOGRÁFICAS

- **ALEXANDRIS, N.** *Design of Microprocessor-Based Systems*. Englewood Cliffs, NJ: Prentice Hall, 1993.
- **SHANLEY, T., y Anderson, D.** *PCI System Architecture*. Addison Wesley, Mindshare Press, 1999.
- **SOLARI, E., y Wilse, G.** *PCI Hardware and Software: Architecture and Design*. San Diego, CA: Annabooks, 1994.
- **SIG PCI.** *PCI Specification v2.2*