

Contenido

Abstract.....	2
Introducción.....	2
Bus PCI – Descripción General.....	2
Características.....	3
Inicialización y Espacio de Direccionamiento.....	4
Configuración.....	5
CardBus.....	7
Investigación.....	15
Introducción.....	15
Inicialización.....	15
Configuración.....	16
Anexo.....	16
Tipos de Cabecera y BAR.....	16
Referencias.....	19

Driver PCI

Abstract. La presente investigación tiene como objetivo el desarrollo de un driver PCI¹ totalmente funcional, de forma tal que el driver pueda detectar los dispositivos conectados al bus PCI actuando de interfaz entre un dispositivo y la CPU². Actualmente SODIUM³ cuenta con un driver PCI pero este no cuenta con la integridad suficiente para centralizar la gestión correspondiente de los device drivers⁴ conectados al BUS. Se investigará como como el driver PCI se conecta con los dispositivos conectados a sus buses, como configura y administra los mismos.

Introducción

El propósito de este documento es explicar el driver PCI, como es su estructura, funcionamiento y pasos necesarios para que los device drivers puedan interactuar con la CPU. Se presupone que el lector tiene conocimientos de cómo surge el bus PCI y se posiciona en la Arquitectura de las computadoras, así como también conoce las especificaciones de hardware. Inicialmente se hará un breve resumen sobre la estructura y funcionamiento del bus PCI y luego se profundizará en la interacción que este tiene con el Sistema Operativo, abarcando su inicialización, métodos de polling⁵, y como organiza la configuración de los dispositivos conectados a sus buses.

Bus PCI – Descripción General

El bus local PCI es un bus de alta performance de 32 o 64 bits con líneas de dirección y de datos multiplexadas. Su uso se orienta como mecanismo de interconexión entre controladores de periféricos altamente integrados, placas periféricas de expansión y sistemas procesador/memoria.

Los componentes e interfaces agregables PCI son independientes del tipo de microprocesador, permitiendo una eficiente transición a futuras generaciones de procesadores y al uso de arquitecturas multi-procesador. Esta independencia permite que el bus local PCI sea optimizado para funciones I/O⁶, habilita la operación concurrente del bus local con el sub-sistema procesador/memoria y acomoda múltiples periféricos de altas prestaciones.

¹ Peripheral Component Interconnect - Interconexión de Componentes Periféricos.

² Unidad Central de Procesamiento.

³ Sistema Operativo del Departamento de Ingeniería de la Universidad de la Matanza.

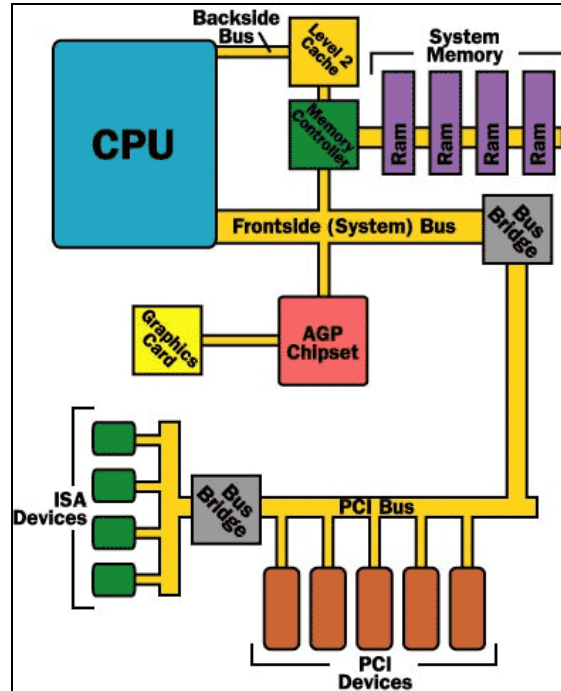
⁴ controlador de dispositivo, es un programa informático que permite al sistema operativo interactuar con un periférico.

⁵ Sondeo de los buses PCI

⁶ Input – Output / Entrada - Salida

Características

En la siguiente figura se muestra un diagrama lógico de un ejemplo de sistema basado en PCI.



Se observa que el bus PCI está delimitado por 2 bridges, llamados puente Norte y puente Sur. Hacemos un párrafo aparte para explicar la funcionalidad de estos 2 puentes.

- *Puente Norte:* También se conoce como controlador de memoria, se encarga de controlar las transferencias entre el procesador y la memoria RAM⁷. Se encuentra ubicado físicamente cerca del procesador. También se lo conoce como GMCH⁸.
- *Puente Sur:* También conocido como concentrador de controladores de entrada/salida, es un circuito integrado que se encarga de coordinar los diferentes dispositivos de entrada y salida y algunas otras funcionalidades de baja velocidad dentro de la tarjeta madre. Este puente no está conectado a la CPU y se comunica con ella indirectamente a través del Puente norte.

Aquí se observa que el sistema CPU se encuentra conectado al bus PCI a través de un "Puente Norte". Este puente provee de un camino de baja latencia y gran ancho de banda, por el cual el procesador puede acceder a cualquier dispositivo mapeado en memoria o en los espacios para I/O y permite que cualquier Master PCI tenga acceso directo a memoria. El puente puede, opcionalmente, incluir funciones como buffering/posting y funciones centrales PCI (por ej.: arbitraje).

La CPU y los dispositivos PCI tienen que acceder a la memoria que se comparte entre ellos. Esta memoria es utilizada por los controladores de dispositivos para el control de los dispositivos PCI y para pasar información entre ellos. Típicamente, la memoria compartida contiene registros de control y de estado para el

⁷ Memoria de Acceso Aleatorio

⁸ Concentrador de controladores gráficos y de memoria

dispositivo. Estos registros se utilizan para controlar el dispositivo y para leer su estado. Por ejemplo, el controlador de dispositivo SCSI⁹, PCI leería su registro de estado para determinar si el dispositivo SCSI está dispuesto a escribir un bloque de información en el disco SCSI. O puede escribir en el registro de control para iniciar el dispositivo que ejecuta después de que se haya encendido.

Inicialización y Espacio de Direccionamiento

En un bus PCI existen tres espacios de direccionamiento:

- Memoria
- Entrada - Salida
- Configuración

El *espacio de direcciones de memoria* tiene un tamaño de hasta 2^{32} dado que el bus de direcciones de PCI puede alcanzar los 32 bits. (64 si incluimos los ciclos de dirección dual) Observar que una dirección de memoria es en definitiva un identificador dentro del sistema de una localización, presumiblemente, donde almacenar un byte. Esto significa que disponemos de $2^{32/64}$ identificadores que podemos asignar a localizaciones de memoria o a dispositivos de entrada salida si mapeamos estos en memoria.

Del *espacio de Entrada/Salida* se puede decir tres cuartos de lo mismo que se ha dicho del espacio de direcciones de memoria, salvo que no es posible realizar ciclos de dirección dual a entrada-salida.

Situar un dispositivo o un módulo de memoria dentro de uno de estos dos espacios significa asignar una dirección a un dispositivo o localización de memoria de manera que siempre que se realice una transacción con esa dirección respondan ese dispositivo o esa localización y no otra. A esa asignación es a lo que llamamos configuración.

Mientras no se realice el proceso de configuración, ninguno de los dispositivos o localizaciones de memoria responderá ante ninguna transacción de memoria o entrada/salida. ¿Cómo es posible entonces acceder a algún dispositivo, puesto que no tienen identificador asignado, como mínimo para poder asignarle un identificador?

La respuesta es porque existe un tercer espacio de direcciones que es el de configuración. Este último espacio vamos a detallarlo un poco más en detalle.

Los sistemas deben proporcionar un mecanismo que permite el acceso al espacio de configuración PCI, como la mayoría de CPUs no tienen ningún mecanismo de este tipo. Esta tarea se realiza generalmente por el anfitrión como Puente PCI.

El *espacio de configuración*. Se trata de 256 bytes que son direccionables, podemos decir que por cada bus hay 8 bits para cada bus, donde 5 bits son para los dispositivos y los 3 bits restantes son para las funciones del mismo. Esto permite un máximo de 256 autobuses, cada uno con hasta 32 dispositivos, cada uno apoyando 8 funciones. Una sola tarjeta de expansión PCI puede responder como un dispositivo y debe implementar al menos número de función cero. Los primeros 64 bytes de espacio de configuración están estandarizados, y el resto están disponibles para los propósitos definidos por el proveedor.

Al iniciar, el sistema realiza la configuración de todos los dispositivos conectados al bus PCI. Parte de la información, presente en el espacio de configuración de cada dispositivo PCI, permite a la BIOS del sistema habilitar el dispositivo PCI y asignarle un espacio de memoria en la tabla de memoria del sistema elaborada por el software de arranque.

⁹ Small Computers System Interface (Interfaz de Sistema para Pequeñas Computadoras). Permite conectar otros dispositivos a la pc como ser: un escáner SCSI, un disco duro SCSI, etc.

Configuración

En esta sección vamos a explicar aspectos claves sobre el mecanismo de configuración, debido a que más adelante detallaremos como se configuran los dispositivos mediante el bus PCI mientras inicia el SO¹⁰.

Para la configuración del dispositivo se proveen 2 espacios de direcciones de E/S¹¹, que tienen un ancho de 32 bits. La primera ubicación (0xcf8) se denomina CONFIG_ADDRESS (Dirección), y la segunda (0xcfc) se llama CONFIG_DATA (Datos).

CONFIG_ADDRESS especifica la dirección de configuración que se requiere para tener accesos, mientras que los accesos a config_data en realidad generan el acceso a la configuración y se transferirán los datos hacia o desde el registro config_data.

El CONFIG_ADDRESS es un registro de 32 bits con el formato que se muestra en la figura siguiente.

31	30 – 24	23 – 16	15 – 11	10 – 8	7 – 2	1 – 0
Bit Habilitado	Reservado	Número del Bus ¹²	Numero de Dispositivo ¹³	Numero de Función ¹⁴	Número de Registro	00

- Bit 31 es una bandera de habilitación para determinar cuándo accesos a config_data deben ser traducidos a los ciclos de configuración.
- Los bits 23 a 16 permiten que el software de configuración para elegir un bus PCI específica en el sistema.
- Bits 15 a 11 seleccionar el dispositivo específico en el bus PCI.
- Bits 10 a 8 elegir una función específica en un dispositivo (si el dispositivo es compatible con múltiples funciones).
- Bits 7 a 2 seleccionar el área específica de 32 bits en el espacio de configuración del dispositivo.

Este espacio de direcciones de configuración se divide en un área de encabezado y en un área que depende del dispositivo. En general los dispositivos PCI tienen que soportar aquellos registros necesarios para su funcionamiento. El área de configuración tiene que estar accesible en cualquier momento, y no solo al inicio. Para cada dispositivo PCI se establece un área de encabezado de 64 Bytes, y los 192 Bytes restantes (a partir de la dirección 40h) dependen del dispositivo y pueden ser utilizados por los fabricantes para funciones propias. Por ejemplo en esta área se encuentran los registros para el control de memoria cache y RAM, bridges, etc.

Podemos tener diferentes cabeceras que identifican lo siguiente:

- PCI Standard
- Puente PCI-to-PCI
- Puente PCI a CardBus (Mas adelante vamos a explicar el concepto de CardBus)

Las siguientes descripciones de los campos son comunes a todos los tipos de cabecera:

¹⁰ Sistema Operativo.

¹¹ Entrada / Salida.

¹² Un número de bus, se pueden direccionar hasta 256 buses.

¹³ Un número de dispositivo, hasta 32 dispositivos por cada bus.

¹⁴ Un número de función, hasta 8 funciones por cada dispositivo.

31		16 15		0		
Device ID		Vendor ID				00h
Status		Command				04h
Class Code			Revision ID			08h
BIST	Header Type	Lat. Timer	Cache Line S.			0Ch
Base Address Registers						10h
						14h
						18h
						1Ch
						20h
Cardbus CIS Pointer						24h
Subsystem ID			Subsystem Vendor ID			28h
Expansion ROM Base Address						2Ch
Reserved				Cap. Pointer		30h
Reserved						34h
Reserved						38h
Max Lat.	Min Gnt.	Interrupt Pin	Interrupt Line			3Ch

Si hemos comprendido bien entenderemos que el espacio de configuración tiene por objeto identificar el conjunto de todos los registros de configuración que albergan todos los dispositivos conectados a la estructura de buses PCI.

Este conjunto de registros está distribuido en bloques de 64 registros de 32 bits uno por cada función implementada en cada dispositivo. La función de los registros es la de informar al sistema del tipo de dispositivo y qué requerimientos tiene con respecto a espacios de direcciones y vectores de interrupción, así como qué tipo de función realiza y características con que son realizadas. En muchos casos estos registros son cableados por el propio fabricante. Otros registros sirven para que el sistema configure el dispositivo para que encaje en el lugar (rango de direcciones de memoria o entrada-salida, vectores de interrupción, etc.) que el sistema le ha localizado. El sistema escribirá en dichos registros la información necesaria para que el dispositivo quede convenientemente instalado en el sistema.

Los registros de configuración podemos dividirlos en tres categorías: los que son de obligada inclusión en todo dispositivo PCI, con independencia de la función que realice, los que se incluirán dependiendo de la función que realice el dispositivo y otros que pueden ser definidos de forma específica.

- *ID Dispositivo*: Es un registro de 16 bits que identifica al fabricante del dispositivo. El identificador es suministrado por una autoridad central a cada fabricante. El valor FFFFh está reservado para indicar que dicha función no ha sido implementada en el dispositivo (Recuérdese que el dispositivo puede tener 'hueco' para ocho funciones; se conocerá que un 'hueco' no está ocupado porque su registro Vendor ID está a este valor)
- *ID Fabricante*: También es un registro de 16 bits. Identifica el fabricante del dispositivo. Los IDs válidos son asignados por PCI-SIG ¹⁵ para garantizar la singularidad, por lo tanto no hay 2 fabricantes con el mismo ID. 0xFFFF se retorna cuando se quiere acceder a un espacio de configuración de un dispositivo inexistente.
- *Estado*: A regístrase utilizado para registrar información sobre el estado de los eventos relacionados con el bus PCI.
- *Comando*: Proporciona control sobre la capacidad de un dispositivo para generar y responder a los ciclos PCI. La única funcionalidad garantizada con el apoyo de todos los dispositivos es cuando un 0

¹⁵ PCI-SIG es la organización para desarrollar y gestionar el estándar PCI

se escribe en este registro, el dispositivo se desconecta del bus PCI para todos los accesos a excepción del acceso al espacio de configuración.

- *Código de clase*: registro de solo lectura que especifica el tipo de función que realiza el dispositivo.
- *Subclase*: registro de sólo lectura que especifica la función específica del dispositivo realiza.
- *Prog SI*: Un registro de sólo lectura que especifica una interfaz de programación a nivel de registro del dispositivo tiene, si tiene alguno.
- *ID de revisión*: Especifica un identificador de revisión para un dispositivo en particular. Cuando IDs válidos son asignados por el vendedor.
- *BIST*: Representa que el estado y permite el control de dispositivos BIST (built-in auto-test).
- *Tipo de cabecera*: Identifica el diseño de la cabecera empezando en 0x10 bytes de la cabecera y también especifica si el dispositivo tiene múltiples funciones o tiene solo una función. El valor de 0x00 especifica un dispositivo en general, un valor de 0x01 especifica un puente PCI-PCI, y un valor de 0x02 especifica un puente CardBus. Si se establece el bit 7 de este registro, el dispositivo tiene múltiples funciones, de lo contrario, es un único dispositivo de función.
- *Temporizador de Latencia*: Especifica el tiempo de latencia en unidades de relojes del bus PCI.
- *Tamaño Caché-Line*: Especifica el sistema de caché tamaño de la cache-line (2 bytes). Un dispositivo puede limitar el número de tamaños de cache-lines que puede soportar, si un valor no admitido se escribe en este campo, el dispositivo se comporta como si el valor 0 se hubiera escrito en este registro.

Para ver la descripción y un breve resumen sobre tipos de cabecera ver *anexo Tipos de Cabecera y BAR*¹⁶
Realizada la breve introducción sobre la estructura y funcionamiento del bus PCI, en los próximos temas abarcaremos la interacción que el Bus tiene con el Sistema Operativo SODIUM, abarcando su inicialización, métodos de polling, y como organiza la configuración de los dispositivos conectados a sus buses.

CardBus

La **PCMCIA 1.0** (Peripheral Component MicroChannel Interconnect Architecture) intentó ser una versión internacional de la ya bien establecida especificación y arquitectura de la tarjeta de memoria JEDIA 4.0 (Japan Electronic Industries Development Association).

En 1991 los estándares JEDIA y PCMCIA se fusionaron para formar la especificación PCMCIA 2.0. Esta nueva versión iba más allá de las tarjetas de memoria, ya que agregaba funcionalidades de I/O, abriendo camino a otro tipo de hardware como módems, y placas de red.

En 1992 se lanzó la versión PCMCIA 2.01 en la que agregaron funcionalidades ATA (Advanced Technology Attachment) que proporcionaba un medio de almacenamiento con un controlador dedicado. Esto le permitía a la especificación PCMCIA operar también como si fuera un disco rígido (IDE).

En 1993 se lanzó la versión PCMCIA 2.1, que optimizaba las capacidades eléctricas y físicas de los dispositivos.

En 1995 PCMCIA pasó a llamarse **PC Card** y se produjo una revolución en la especificación, ya que proporcionaba una conexión estándar para dispositivos móviles a 33MHz con lo que pasó a ser un rival para PCI. La nueva especificación agregaba:

¹⁶ Registros de Dirección Base

- Capacidades de 32-bit Bus mastering (**Cardbus**).
- 3.3 volt
- Soporte APM (Advanced Power Management)
- Soporte DMA (Direct Memory Access) - Cardbus
- Capacidad multifunción para los dispositivos (p.e. modem y ethernet).

Resumiendo:

PC Card y PCMCIA

- Dispositivo de 16-bit
- No implementa DMA ni Bus mastering.

Cardbus

- Dispositivo de 32-bit
- Implementa DMA o Bus mastering.

Una tarjeta PCMCIA o PC Card se puede utilizar en un slot compatible con Cardbus, pero no al revés. Si bien físicamente son parecidas, la Cardbus tiene una franja dorada al lado de los conectores.

Dentro del espacio de configuración, tenemos que ver en el registro **HEADER TYPE**. Si es 02h entonces se trata de un puente Cardbus. Recordemos que 00h es para dispositivo PCI, y 01h es para puente PCI-PCI.

Otro registro importante es el **CARDBUS CIS POINTER**. Este registro, opcional, apunta al CIS (Card Information Structure) de una tarjeta.

La CIS es una estructura de datos que se acceden a través de Card Services, y que contiene identificación y configuración de tarjetas PC Card. Incluye información como velocidad del dispositivo, tamaño de datos, y recursos del sistema para el funcionamiento de la tarjeta. El driver de una tarjeta PC Card accede a la CIS durante la inicialización para determinar las opciones de configuración que soporta la tarjeta.

La CIS es una lista enlazada de tuplas de longitud variable que describen la función y características de una PC Card. Cada tupla tiene un byte que describe el tipo de tupla, y otro que indica el comienzo de la próxima tupla dentro de la lista enlazada.

La especificación PCMCIA se refiere a la combinación de todas las tuplas utilizadas para describir a una PC Card como metaformato. Gracias al metaformato, la CIS se puede leer por cualquier sistema operativo.

Puente PCI-Cardbus/PC Card

En esta sección analizamos el puente PCI-to-CardBus/Pc Card. Cada socket aceptará 16-bit PC Card o 32-bit CardBus. Cuando se inserta una de estas placas, el puente detecta el tipo de placa instalada y proporciona las funciones necesarias para dar soporte a dicha placa. Estas funciones incluyen el protocolo del bus, la lógica de configuración y el control de energía del socket.

El puente aparece en el primer bus PCI como un puente multifunción 16-bit/32-bit con uno o dos sockets compartidos. La Función 0 proporciona la interface de funcionalidad para el socket 0. La función 1 contiene la misma funcionalidad para el socket 1. Esta división de configuración y control para cada socket CardBus

sigue las convenciones definidas para el puente PCI-PCI. Siguiendo esta metodología el puerto se puede tratar como dos puentes separados e independientes.

Cuando se diseña un puente, no es posible saber cuántos de los sockets disponibles serán utilizados. Se debe contar con algún mecanismo que le diga al puente cuántos sockets se han implementado. Si no se proporciona tal mecanismo, o no se guarda el estado, el software asumirá y reportará incorrectamente recursos que de hecho no existen. El software no conoce la diferencia entre un socket vacío y un socket inexistente. Cuando se configura como un solo socket, sólo el espacio de configuración, y registros de control, para el socket implementado, deberían ser visibles para el software. El espacio de configuración para el socket no implementado debe permanecer invisible e inaccesible para el software. Esto permite al software de configuración tratar con la implementación de un sistema dado, sin información externa.

El protocolo del bus es responsabilidad del hardware. El puente debe usar el protocolo correcto, en su lado secundario, basado en el tipo de placa conectada. Cuando se conecta a un socket una placa PC Card, o una placa CardBus, el puente automáticamente utiliza el protocolo de comunicaciones con el bus. El software no es responsable de configurar esto.

Las placas se pueden mapear en memoria en cualquier lugar dentro del espacio de direcciones asignadas al puente. El puente proporciona dos pares de registros memoria base / limite, utilizados para el mapeo a memoria.

Se proporcionan dos pares de registros de mapeo I/O para cada socket. Esto permite fragmentar el espacio de I/O en cada placa, también permite intercalar espacio I/O con otros dispositivos I/O.

El puente determina el tipo de placa y los voltajes Vcc que requiere a través de los pines Card Detect y VS. Los pines Card Detect, tipo de placa y los voltajes requeridos/permitidos se guardan en el registro Status del socket del puente.

El puente controla la energía de cada socket. Corta automáticamente la energía a un socket una vez que la placa se extrae.

Mapeo de direcciones de memoria

El puente proporciona varios registros base-limite que determinan un rango de memoria. Los mismos sirven para mapear estos rangos en la memoria del sistema. Estos rangos se configuran por software, de acuerdo a la memoria disponible y requerimientos de la placa. Si un rango no está en el bus secundario, el puente asume que está en el bus primario.

Hay rangos de memoria prefetchable y rangos de memoria I/O. Cada rango es un espacio de direcciones contiguo. Los rangos separados no necesitan ser contiguos. El puente utiliza los valores de los registros de mapeo para determinar dónde se encuentra la dirección de destino. Si está en un rango del bus secundario, será reclamada por el puente y pasada al bus apropiado. Los accesos de una placa a una dirección que no está en su bus, se pasara al bus PCI, a menos que la dirección resida en el otro CardBus. En ese caso, pasará el acceso al bus secundario. Los controladores CardBus que proporcionen más de un slot, deben proporcionar transferencias placa-a-placa.

Configuración

El puente ve al bus PCI principal como dos buses secundarios separados que residen en un solo dispositivo. Cada socket tiene su propio espacio de configuración. Esto hace que el puente sea un dispositivo multifunción.

El espacio de configuración del puente es accesible solo desde el bus PCI primario.

Los comandos de configuración del bus PCI primario se manejan de la siguiente manera:

Tipo 0 (AD[1::0] = 00): Si es seleccionado por IDSEL el puente decodifica los bits 07::02 para determinar el correspondiente registro de configuración (DWord) seleccionado. Basado en el comando de configuración (Read/Write) y el C/BE[3::0] el puente proporciona datos del registro seleccionado o escribirá datos. Los datos leídos serán los 32 bits del registro con los datos solicitados. Los datos a escribir serán depositados en el registro seleccionado utilizando C/BE[3::0] para habilitar la escritura.

Tipo 1 (AD[1::0]=01): Si el número de bus (bits 23::16) coincide con el controlador del número del bus secundario, iniciará un ciclo de configuración Tipo 0 en el bus destino. Setea AD[1::0] = 00 y pasa a través de AD[10::2] sin cambios.

Los ciclos Tipo 1 que no coinciden con los puentes del bus secundario, pero que coinciden con el bus subordinado, serán pasados sin modificación.

Los ciclos de configuración Tipo 0 en el bus subordinado son ignorados por el controlador. Los comandos Tipo 1 en el bus secundario, también son ignorados, excepto en el caso cuando el número de dispositivo son todos 1.

Registros de Configuración

El puente tiene dos espacios de configuración separados, uno para cada socket.

Cuando se configura como un puente CardBus de dos slots, el puente tiene dos espacios de configuración. La función 0 es para el socket 0, la función 1 es para el slot 1. CardBus se definió para ser lo más parecido a PCI para permitir compartir circuitos. La definición de los espacios de configuración de Cardbus son parecidos a los definidos en puentes PCI-to-PCI. Las diferencias en los registros base I/O y registros límite son manejados por CardBus, requiriendo 4 bytes, mientras que PCI-to-PCI requieren 4 Kbytes.

A continuación se presentan los registros de cabecera tipo 2, puente PCI-Cardbus y luego una explicación de cada registro (muchos de ellos cumplen la misma función que los de cabecera tipo 0):

Device ID = nnnn		Vendor ID = nnnn		00H
Status		Command		04H
Class Code = 060700H			Revision ID = nn	08H
BIST	Header Type = 82H	Latency Timer	Cache Line Size	0CH
PC Card Socket Status and Control Registers Base Address				10H
Secondary Status		Reserved	Cap_Ptr	14H
CardBus Latency Timer	Subordinate Bus Number	CardBus Bus Number	PCI Bus Number	18H
Memory Base 0				1CH
Memory Limit 0				20H
Memory Base 1				24H
Memory Limit 1				28H
I/O Base 0 (Upper 16 Bits, optional)		I/O Base 0 (Lower 16 Bits)		2CH
I/O Limit 0 (Upper 16 Bits, optional)		I/O Limit 0 (Lower 16 Bits)		30H
I/O Base 1 (Upper 16 Bits, optional)		I/O Base 1 (Lower 16 Bits)		34H
I/O Limit 1 (Upper 16 Bits, optional)		I/O Limit 1 (Lower 16 Bits)		38H
Bridge Control		Interrupt Pin	Interrupt Line	3CH
Subsystem ID (optional)		Subsystem Vendor ID (optional)		40H
PC Card 16 Bit IF Legacy Mode Base Address (Optional)				44H
Reserved				48-7FH
User Defined				80-FFH

Vendor ID (0x00)

Identifica al fabricante del dispositivo. Lo asigna la PCI SIG.

Device ID (0x02)

Identifica al dispositivo particular. Lo asigna el fabricante.

Command Register (0x04)

Lo mismo que en la Especificación PCI Local Bus y PCI-to-PCI Bridge.

Status Register (0x06)

Lo mismo que en la Especificación PCI Local Bus, pero sólo para la interface primaria PCI.

Revision ID (0x08)

Seleccionada por el fabricante.

Class Code (0x09)

06 = bridge device; 07 = cardbus; 00 = interface de programación.

Cache Line Size (0x0C)

Lo mismo que en la Especificación PCI Local Bus.

Latency Timer (0x0D)

Lo mismo que en la Especificación PCI Local Bus. Solo aplica a la interface primaria.

Header Type (0x0E)

82 = Puente PCI-CardBus multifunción.

02 = Puente PCI-Cardbus simple.

BIST (0x0F)

Lo mismo que en la Especificación PCI Local Bus. Si no está implementado, este registro devuelve todos ceros.

PC Card Socket Status y Control Registers Base Address (0x10)

Este registro apunta la memoria mapeada I/O que contiene los registros de estado y control para PC Card y CardBus. Para CardBus estos registros empiezan en offset = 0x000, y para PC Card en offset = 0x800. Bits [31::11] son de R/W. Bits [11::00] son siempre ceros. Lo mismo que en la Especificación PCI Local Bus.

Cap_Ptr (Capabilities Pointer) (0x14)

Proporciona un offset al primer elemento de la lista enlazada de Capabilities Linked List. Es un registro Dword alineado de tal forma que los dos bits menos significativos son siempre 00. Tiene un valor mínimo de 0x80 y un valor máximo de 0xF8. Si la característica de New Capabilities no está implementada, devuelve 00.

Secondary Status (0x16)

Es similar a la función de Primary Status Register, pero contiene información referida a CardBus. Si el bit 14 es 1, indica que el puente ha detectado un SERR en la placa CardBus. Lo mismo que en la Especificación PCI Local Bus.

PCI Bus Number (0x18)

El número de bus primario identifica el número de bus PCI en la parte primaria del puente. Se configura por software.

CardBus Bus Number (0x19)

Identifica el número de placa CardBus conectada al socket. Se configura por PCI BIOS. Se denomina "Secondary Bus Number" en el caso de puentes PCI-PCI.

Subordinate Bus Number (0x1A)

Es un registro definido para puentes PCI-PCI. Contiene el número de bus en la parte menor de la jerarquía detrás del puente. Normalmente, un puente CardBus estará en la parte más baja de la jerarquía y contendrá el mismo valor que el registro CardBus Bus Number.

CardBus Latency Timer (0x1B)

Tiene la misma funcionalidad que el registro Latency Timer del bus PCI primario, pero aplica sólo a placa CardBus conectada a este socket específico. Configurado por el PCI BIOS.

Memory Base #0 (0x1C)

Define la dirección más baja de una ventana de memoria I/O mapeada. Los 20 bits más altos corresponden a los bits de dirección AD[31::12]. Los 12 bits más bajos son RO y devuelven ceros. Esta ventana se habilita con el bit 1 del registro Command. Hacerla prefetcheable se hace controlando el bit 8 del registro Bridge Control.

Memory Limit #0 (0x20)

Define la dirección más alta de una ventana de memoria I/O mapeada. Los 20 más altos corresponden a AD[31::12]. Los 12 bits más bajos son de RO y devuelven ceros. El puente asume que los bits [11:0] son 1 para determinar el rango definido. Por lo tanto, si Memory Base y Limit tienen el mismo valor, se ha definido una ventana de 4KB. Ambas ventanas de memoria se habilitan con el bit 1 del registro Command. Para deshabilitar la ventana individualmente, el registro Limit debe tener un valor debajo de Memory Base.

Memory Base #1 (0x24)

Lo mismo que Memory Base #0. Prefetching se hace con el bit 9 del registro Bridge Control.

Memory Limit #1 (0x28)

Lo mismo que Memory Limit #0.

I/O Base #0 (Lower 16 bits) (0x2C)

Define la dirección más baja del rango de direcciones utilizada por el puente cuando tiene que determinar si debe reenviar la transacción I/O al CardBus. Los bits en este registro corresponden a AD[15::0].

Los bits AD[1::0] se usan para indicar si el puente implementa direccionamiento de 16 o 32 bits (0x00: 16-bit; 0x01: 32-bit). Para la decodificación, si se implementa solo direccionamiento 16-bit el puente debe controlar que los 16 bits más altos sean ceros antes de aceptar el acceso.

I/O Base #0 (Upper 16 bits) (0x2E)

Opcional. Extensión. Define los bits AD[31::16]. Si estos y los 16-bit de I/O Limit Upper no están implementados, los dispositivos I/O detrás del puente serán mapeados debajo de 0x00010000 y el puente debe validar que los bits AD[31::16] sean todos ceros antes de aceptar el acceso.

I/O Limit #0 (Lower 16 bits) (0x30)

Define si la dirección más alta del rango de direcciones que se usará por el puente para determinar cuándo reenviar accesos I/O hacia el CardBus. Estos bits se corresponden con AD[15::00].

I/O Limit #0 (Upper 16 bits) (0x32)

Opcional. Extensión del registro I/O Limit.

Interrupt Line (0x3C)

Lo mismo que en la Especificación PCI Local Bus.

Interrupt Pin (0x3D)

Lo mismo que en la Especificación PCI Local Bus

Bridge Control Register (0x3E)

Es una extensión del registro PCI Command.

0: Habilitar respuesta de error de paridad. Si es cero los errores se ignoran. Si es uno, los errores se controlan y se reportan. De forma predeterminada están deshabilitados.

1: Habilitar SERR#. Si es uno el puente reenvía al bus PCI una indicación de SERR# en el cardbus. Default: deshabilitado.

2: Habilitar ISA.

3: Habilitar VGA. Modifica las respuestas del puerto hacia direcciones compatibles VGA. Si es uno, el puente reenvía transacciones a los siguientes rangos:

Memoria: 0A 0000 hasta 0B FFFF.

I/O: direcciones donde AD[9:0] están en los rangos 3B0 a 3BB y 3C0 a 3DF.

4: Reservado. Devuelve cero.

5: Modo Master Abort. Controla el comportamiento del puente cuando un master abort ocurre en la interface PCI o CardBus, cuando el puente esta en modo master.

6: CardBus Reset. Cuando es 1 el puente impone y mantiene CRST#. Cuando es cero quita el CRST#.

7: Habilita IREQ-INT. Si es uno habilita ruteo IRQ para PC Card 16-bit. Cuando es cero interrupciones IREQ son ruteadas al pin INT indicado por el registro Interrupt Pin.

8: Habilitar Memory 0 Prefetch.

9: Habilitar Memory 1 Prefetch.

10: Habilitar Write Posting.

11-15: Reservados. Devuelven ceros.

Subsystem Vendor ID (0x40)

Opcional. Lo mismo que en la Especificación PCI Local Bus. Si no está implementado devuelve ceros. Se asigna por la PCI SIG.

Subsystem ID (0x42)

Opcional. Lo mismo que en la Especificación PCI Local Bus.

PC Card 16 bit IF Legacy Mode Base Address (0x44)

Opcional. Apunta a los registros índice y datos que residen en 3E1 y 3E0 en el 82365. Lo mismo que en la Especificación PCI Local Bus.

Capabilities List (lista de capacidades)

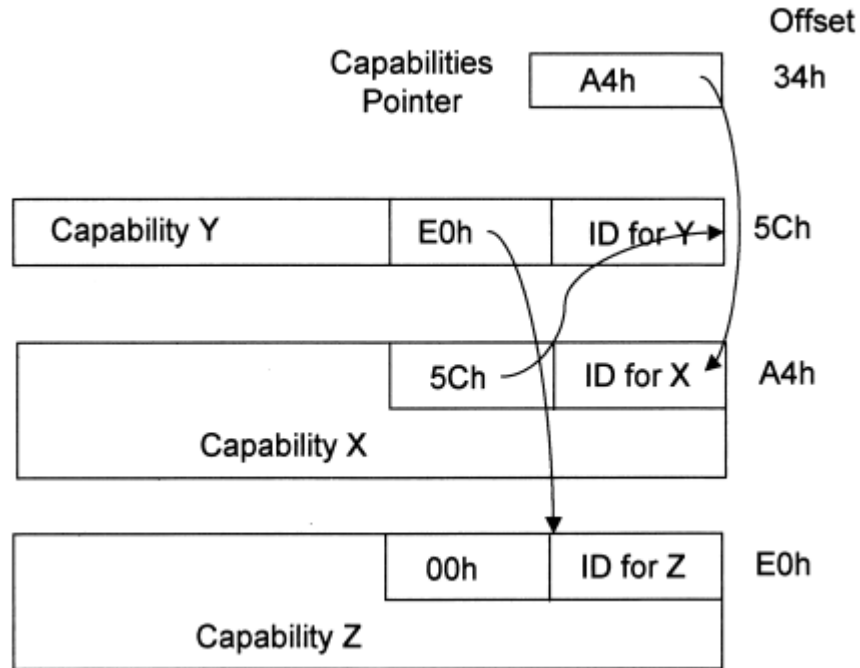
La lista de capacidades, es un nuevo mecanismo incorporado en la revisión 2.2 que soporta capacidades nuevas y opcionales de dispositivos PCI, y se presenta en forma de lista enlazada. Si el bit 4 del registro Status es 1, entonces el byte en el offset 0x34 de la cabecera, contiene el offset al primer elemento de la lista de capacidades. Esta lista reside en el espacio de configuración de la función específica del dispositivo.

Cada capacidad consiste en 8-bit de ID, asignado por la PCI SIG, 8-bit de offset al próximo elemento de la lista, y otros bytes adicionales que pueden ser RO o RW.

El campo offset del último elemento de la lista es cero.

Algunas capacidades ya están estandarizadas, por ejemplo:

- 0: Reservada.
- 1: PCI Power Management Interface.
- 2: AGP. Identifica una controladora gráfica utilizando características de AGP.
- 3: VPD. Soporte para Vital Product Data.
- 4: Slot Identification. Identifica un puente que proporciona capacidades de extensión externas.
- 5: Message Signaled Interrupts.
- 6: Compact PCI Hot Swap CSR.



Investigación

Introducción

Al iniciar, el sistema realiza la configuración de todos los dispositivos conectados al bus PCI. Parte de la información, presente en el espacio de configuración de cada dispositivo PCI, permite a la BIOS del sistema habilitar el dispositivo PCI y asignarle un espacio de memoria en la tabla de memoria del sistema elaborada por el software de arranque.

Inicialización

Al iniciar SODIUM los dispositivos PCI se encuentran en estado inactivos, para activar dichos dispositivos se debe hacer efectuar un método de *sondeo* por los buses PCI, de esta manera se puede detectar los mismos para asignarles recursos, inicializarlos y luego habilitarlos.

Hay distintos métodos de *sondeo* o *polling* para detectar y habilitar los dispositivos.

- *Fuerza Bruta*: Se chequea todos los buses, por cada bus se chequea los 32 dispositivos asociados al mismo y por cada dispositivo chequea las 8 funciones
- *Recursivo*: Como su nombre lo indica este método es recursivo. Un primer paso habría que desarrollar una función en la cual se chequea cada bus, luego se chequea el dispositivo, entonces si este es puente se procede a efectuar la recursividad ya que se llama nuevamente a la función del cheque del bus para recorrerlo.
- *Recursivo con configuración de bus*: Este método es muy similar al anterior pero tiene la posibilidad de configurar también el bus secundario.

SODIUM podría implementar un mecanismo de selección donde se podría elegir el método de sondeo para poder configurar los distintos dispositivos conectados al sistema.

Configuración

A continuación se va a hacer una breve explicación de cómo SODIUM configura los dispositivos conectados al bus PCI.

- Por cada bus PCI se crea una estructura con la capacidad suficiente para ir almacenando los datos relevantes de los posibles dispositivos a habilitar. En una primera instancia solo se puede salvar el número de bus escaneado.
- Luego se procede a escanear los posibles 32 dispositivos conectados al bus. Para ello por cada línea de dispositivo se lee el registro número 4 que indica el Tipo de cabecera del Dispositivo.
- Si es que se detecta un dispositivo o puente se procede a sondear las posibles 8 funciones del dispositivo encontrado.
- Luego se lee el registro 1 que indica el ID del dispositivo y el ID del fabricante, entonces si el dispositivo existe se procede a salvar la información necesaria de acuerdo a los registros generales y el tipo de cabecera
- Una vez obtenida la información se procede a la configuración del dispositivo PCI. Para ello esta se tienen que inicializar los recursos PCI del dispositivo, determinando sus espacios de entrada/salida, memoria, si los tuviera, y la línea de interrupción para poder establecer la comunicación del dispositivo con el resto del sistema.
- Luego se procede a guardar la estructura PCI para tener los datos de los dispositivos en memoria.
- Y por último se procede a llamar a una función que se encarga de buscar el driver específico para un dispositivo PCI.

Una vez que se realizaron los pasos indicados y el dispositivo se encuentra habilitado, los datos permanecen en memoria para que el dispositivo pueda ser accedido por el sistema.

Anexo

Tipos de Cabecera y BAR

Esta tabla es aplicable si el tipo de cabecera es 00h.

Registro	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
00	ID de dispositivo		Vendor ID	
04	Estado		Comando	
08	Código de clase	Subclase	Prog SI	Revisión ID

0C	BIST	Tipo de Cabecera	Temporizador de latencia	Caché Tamaño de línea
10	Dirección base # 0 (BAR0)			
14	Dirección de Base # 1 (BAR1)			
18	Dirección base # 2 (BAR2)			
1C	Dirección de Base # 3 (BAR3)			
20	Dirección base # 4 (BAR4)			
24	Dirección base n ° 5 (Bar5)			
28	Cardbus CIS Pointer			
2C	Subsistema ID		Subsistema Vendor ID	
30	Expansión dirección base ROM			
34	Reservado			Capacidades Pointer
38	Reservado			
3C	Max latencia	Min subvención	Interrumpir PIN	Interrumpir Línea

Esta tabla es aplicable si el tipo de cabecera es 01h (puente PCI-to-PCI)

registro	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
00	ID de dispositivo		Vendor ID	
04	Estado		Comando	
08	Código de clase	Subclase	Prog SI	Revisión ID
0C	BIST	Tipo de Cabecera	Temporizador de latencia	Caché Tamaño de línea
10	Dirección base # 0 (BAR0)			
14	Dirección de Base # 1 (BAR1)			
18	Temporizador de Latencia secundaria	Número Bus subordinado	Número Bus Secundaria	Número Bus Principal
1C	Estado Secundaria		I / O Limit	Base I / O
20	Límite de memoria		Base de memoria	
24	Límite de memoria prefetchable		Base memoria prefetchable	
28	Prefetchable base superior 32 Bits			
2C	Prefetchable límite superior de 32 Bits			
30	E / S de límite superior 16 Bits		I / O base superior 16 Bits	

34	Reservado	Indicador de Capacidad	
38	Expansión dirección base ROM		
3C	Puente de Control	Interrumpir PIN	Interrumpir Línea

Esta tabla es aplicable si el tipo de cabecera es 02h (puente PCI a CardBus)

registro	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
00	ID de dispositivo		Vendor ID	
04	Estado		Comando	
08	Código de clase	Subclase	Prog SI	Revisión ID
0C	BIST	Tipo de Cabecera	Temporizador de latencia	Caché Tamaño de línea
10	CardBus Socket / EXCA dirección base			
14	Estado de Secundaria		Reservado	Desplazamiento de la lista de capacidades
18	CardBus latencia temporizador	Número de bus subordinado	El autobús CardBus	Número de bus PCI
1C	Memoria Base Dirección 0			
20	Límite de memoria 0			
24	Dirección de memoria Base 1			
28	Límite de memoria 1			
2C	I / O Base Address 0			
30	I / O Limit 0			
34	Dirección E / S base 1			
38	I / O LIMIT 1			
3C	Puente de Control		Interrumpir PIN	Interrumpir Línea
40	Subsistema Vendor ID		Subsistema de ID de dispositivo	
44	16-bit PC Card legado dirección base el modo			

Los registros base (BARs), son los encargados de informar a la BIOS del número y tamaño de los espacios de memoria o de entrada / salida necesarios y de identificar una zona de memoria del sistema mapeada sobre un determinado dispositivo PCI. Este proceso se realiza en dos fases:

- En primer lugar, el contenido de cada BAR es leído para determinar si se relaciona con un espacio de memoria o de entrada / salida y qué tamaño precisa. El bit 0 de cada BAR indica si es un 0 que se trata de memoria y si es un 1, un espacio de entrada/salida. El peso binario del primer bit no nulo indica su tamaño según se trate de memoria o entrada / salida.
- Posteriormente, el sistema decodifica esta información y escribe en cada BAR la dirección de memoria que le ha asignado el software de arranque.

Referencias

http://es.wikipedia.org/wiki/Controlador_de_dispositivo

<http://www.ctr.unican.es/asignaturas/pib/PIB-TEMA-V-B1-2en1.pdf>

<http://www.ing.unlp.edu.ar/electrotecnia/procesos/pci1.pdf>

http://es.wikipedia.org/wiki/Peripheral_Component_Interconnect

<http://tarjetasmadre.wikispaces.com/Puente+Norte+y+Puente+sur>

<http://es.kioskea.net/contents/364-que-es-un-bus-informatico>

http://es.wikipedia.org/wiki/Small_Computer_System_Interface

http://en.wikipedia.org/wiki/PCI_configuration_space

http://xwindow.angelfire.com/page13_1.html

http://www.ele.uva.es/~jesman/BigSeti/ftp/Perifericos/PCI/transparencias_pci.pdf

<http://icaro.eii.us.es/asignaturas/tpbn/Transparencias/tema6.pdf>

<http://www.geocities.ws/xmezones/manuales/BusesPC.pdf>

Klaus Dembowski, Gran libro del hardware, Argentina, Editorial MARCOMBO, 2003, segunda edición, pp. 380-395.

Inside PC Card: Cardbus and PCMCIA Design

PC Card Standard Vol. 4 Metaformat Specification

PC Card Standard vol. 8 PC Card Host System Specification

PC Card Standard Vol. 9 Guidelines

PCI Local Bus Specification Rev. 3.0

PCI Bus Demystified

