“PCI BUS HANDLER”: UNA APLICACIÓN QUE FACILITA EL ESTUDIO DEL BUS PCI

Manuel A. Ortiz, Francisco J. Quiles, José I. Benavides, Miguel A. Montijano, Carlos D. Moreno

Universidad de Córdoba. el1orlom@uco.es

RESUMEN

El bus PCI se ha convertido en el bus de conexión de dispositivos periféricos más utilizado actualmente. Sin embargo la complejidad de sus ciclos y la configuración en cuanto al mapeo de los dispositivos que se conecten, dificultan el estudio de este bus para el alumno. Para intentar desmitificar el bus PCI y conseguir que el alumno conozca sus ciclos y funcionalidades básicas, presentamos aquí una simple aplicación que pretende facilitar el estudio y manejo de este bus, permitiendo realizar todas las operaciones necesarias para visualizar y configurar los dispositivos periféricos que estén conectados. No se trata solo de que el alumno conozca el manejo de la aplicación, sino que, además, comprenda como sus actuaciones afectan a los dispositivos periféricos conectados al bus.

1. INTRODUCCIÓN

Actualmente el Bus PCI (Peripheral Component Interconect Bus) se ha convertido en uno de los buses más utilizados para la interconexión de periféricos. Además de su utilización en los ordenadores personales es ampliamente utilizado en otros campos a través de sus variantes como PCIMG en el campo de equipos industriales, Compact PCI en sistemas empotrados, etc.

Aunque este bus resulta familiar al alumno, sin embargo, es un gran desconocido. Su conocimiento aún en los aspectos básicos no es fácil, ya que los dispositivos que se conecten a él deben configurarse antes de realizar transferencias y no es posible acceder a las placas periféricas si no se conoce este mecanismo de configuración. Esta característica de ser un bus configurable hace aún más tediosa y dura la labor del docente para mostrarlo a los alumnos.

Por las razonas anteriormente expuestas hemos emprendido en el Área de Arquitectura y Tecnología de Computadores de la Universidad de Córdoba una serie de trabajos que tienen como finalidad la creación de herramientas hardware y software encaminadas a facilitar el conocimiento básico de este Bus al alumnado. En la comunicación que sigue mostraremos una primera versión de la aplicación “PCI BUS HANDLER”, que pretende mostrar al alumno el bus PCI desde el punto de vista del programador de aplicaciones dejando para otras comunicaciones las placas que hemos desarrollado y que tienen como finalidad mostrar al alumno el bus PCI desde el punto de vista hardware. No sólo es importante que el alumno conozca la aplicación, sino también, como la aplicación accede y configura los dispositivos PCI, es decir, junto con la aplicación el docente debe mostrar como se trasladan las acciones que realiza el usuario al bus PCI.

Por otro lado esta aplicación resulta de gran utilidad a los alumnos de Proyecto Fin de Carrera que realizan placas para el bus PCI, facilitándole la programación de sus placas y un mínimo entorno de pruebas.

2. INTRODUCCIÓN AL BUS PCI

Está fuera del ámbito de esta comunicación describir el bus PCI en toda su extensión. Tampoco es necesario conocerlo profundamente en su aspecto hardware para una lectura comprensiva del resto de la comunicación, por lo que haremos una breve introducción. Si el lector desea una información exhaustiva puede consultar [1] o directamente las especificaciones [2] [3].

En la figura 1 se muestra la estructura típica de buses en un computador. En ella puede observarse el bus PCI que se conecta localmente a la CPU y al sistema de memoria. Al bus PCI se conectan los sistemas periféricos a los que se les da el nombre de dispositivos. Estos dispositivos pueden realizar una función única o varias tratando el bus PCI cada función por separado, por ejemplo, un dispositivo puede ser un interfaz a LAN y otro dispositivo puede ser un MODEM, o bien el mismo dispositivo puede tener una función con interfaz LAN y otra con interfaz MODEM. Esta decisión la suelen adoptar los fabricantes de los dispositivos. Los dispositivos pueden estar integrados en la placa base o pueden ser placas de expansión insertadas en un slot PCI. La topología del bus PCI puede estar formada por un único segmento al que se conecten los dispositivos o puede estar formada por varios segmentos interconectados por puentes PCI (bridges) como muestra a modo de ejemplo la figura 2.

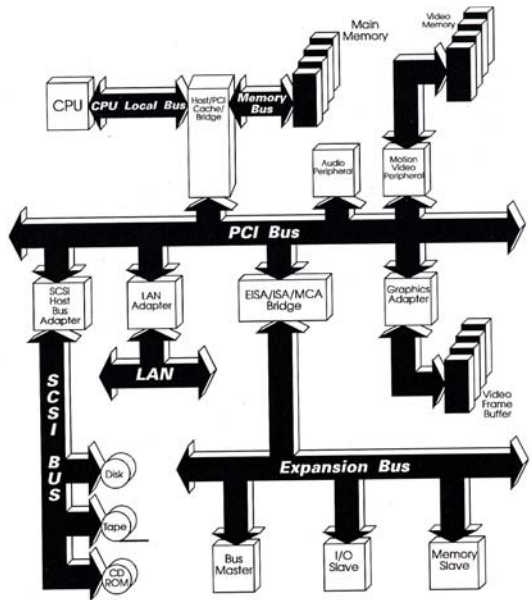

Figura 1. Estructura de buses dentro de un computador

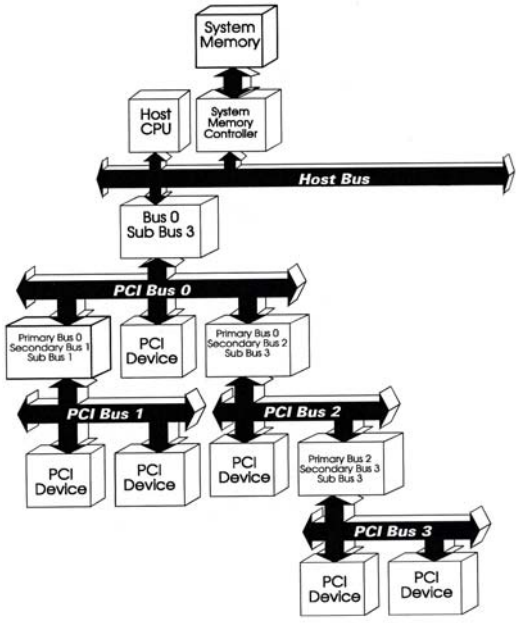

Figura 2. Ejemplo de bus PCI

3. EL BUS PCI DESDE EL PUNTO DE VISTA DEL PROGRAMADOR

Tras la inicialización del sistema, los dispositivos conectados al Bus PCI aparecen al programador como inactivos, y por tanto no se puede realizar transferencias de datos con ellos. Los dispositivos deben ser configurados para que puedan operar. Este concepto es el que supone una mayor dificultad al alumno, ya que los periféricos conectados a los buses que suelen mostrarse en Electrónica se encuentran listos para operar. Es decir en el diseño de los sistemas que habitualmente estudia el alumno, los dispositivos ya aparecen mapeados. Este no es el caso del Bus PCI donde el mapeo de los dispositivos es flexible y se realiza en función de los dispositivos conectados a él y las necesidades de cada uno de estos dispositivos. Esta flexibilidad es posible gracias a un espacio de direcciones, llamado espacio de configuración, que debe tener cada dispositivo separado del espacio habitual de trabajo en el que se mapea.

Además si el dispositivo es multifunción debe tener un espacio de configuración para cada función. Este espacio permitir mapear el dispositivo en tiempo de ejecución, asignar interrupciones y otras características del dispositivo.

A este espacio de configuración se accede a través de dos direcciones del espacio de entrada/salida del procesador (Tabla 1). Una dirección está reservada para escribir la dirección de la palabra o registro del espacio de configuración (CONFIG_ADDR) y la otra para el dato que se lea o escriba (CONFIG_DATA). Los controladores del bus PCI (PCI bridges) serán los encargados de realizar la transformación para acceder al espacio de configuración de cada dispositivo. El acceso a estas dos direcciones debe realizarse en 32 bits por lo que no podremos trabajar en un ordenador personal que no posea un microprocesador capaz de realizar tales transferencias. Por ejemplo, no podremos trabajar con el espacio de configuración en un ordenador personal con un microprocesador que trabaje en modo 8086.

Dirección	Dato
0CF8H	CONFIG_ADDRESS
0CFCH	CONFIG_DATA

Tabla 1. Puertos para acceder al espacio de configuración del bus PCI

Por tanto para acceder al espacio de configuración de un dispositivo deben realizarse dos accesos, uno para indicar la dirección del registro que queremos acceder y otro para leer o escribir el dato. Dichos accesos deberán ser atómicos para impedir que un proceso modifique datos del otro en el caso de que estemos tratando en un entorno multiproceso.

La dirección para acceder a un registro del espacio de configuración de un dispositivo o función de un dispositivo se construye como muestra la figura 3. El número de bus y el número de dispositivo se encuentra prefijado por hardware en la placa base al conectar los periféricos que integre la placa o en los slots de expansión PCI.

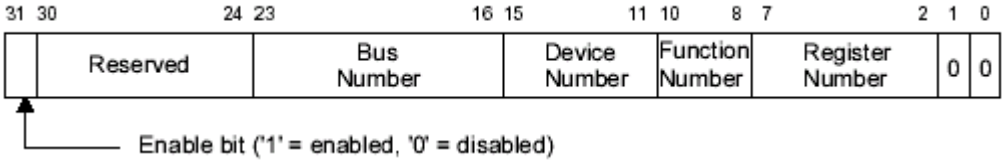

Figura 3. Dirección para acceder a un registro del espacio de configuración

El espacio de configuración se puede observar en la figura 4. Los registros sombreados son obligatorios. Un registro clave es “VendorID” que debe ser único para cada fabricante y se asigna por el grupo “PCI Special Interest Group”[4], siendo además el encargado de crear la normativa referente al bus PCI. El valor de “VendorId” *FFFFh* está reservado y no debe utilizarse bajo ningún concepto. Este valor desde el punto de vista hardware no tiene interés, pero desde el punto de vista software es crucial ya que podremos conocer si existe una placa conectada al bus, si el valor que leemos de “VendorId” es distinto de *FFFFh*. Lo que si tiene importancia desde el punto de vista hardware es el valor de los registros de “Base Address” y de “Interrupt Line”, ya que se le deben asignar el valor de la dirección base y de la interrupción en función de los valores de estos registros. El resto de cuestiones a tener en cuenta desde el punto de vista software las comentaremos junto con la descripción de la aplicación.

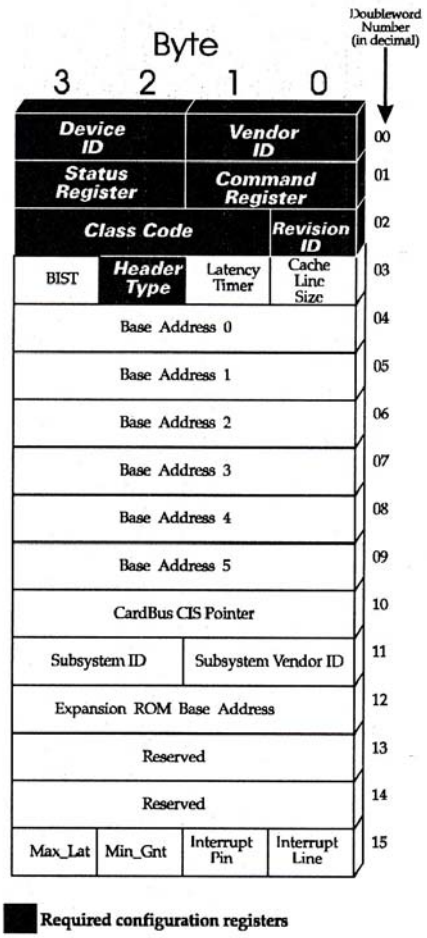

Figura 4. Registros del espacio de configuración de un dispositivo de tipo 00

4. APLICACIÓN: “PCI BUS HANDLER”

La aplicación está desarrollada en VC++ y para el sistema operativo Windows 98. Se encuentra disponible en la dirección http://www.uco.es/~el1orlom/investigacion. Se ha desarrollado para Windows 98 porque es un sistema operativo cómodo para trabajar con el hardware, ya que este sistema permite el acceso a memoria o entrada/salida directamente al no implementar ningún mecanismo de protección y es mucho más fácil trabajar con el hardware que en otros sistemas como Windows2000, Windows NT o Linux, donde no es posible el acceso sin la autorización del sistema. De todas formas el sistema operativo no tiene importancia en el caso que nos ocupa, ya que lo que realmente importa, desde el punto de vista docente, es poder acceder a los dispositivos y a su espacio de configuración de la manera más fácil posible, distinto es si, queremos trabajar con un dispositivo concreto que funcione correctamente donde lo que realizaríamos es un driver para ese dispositivo y para un sistema operativo concreto.

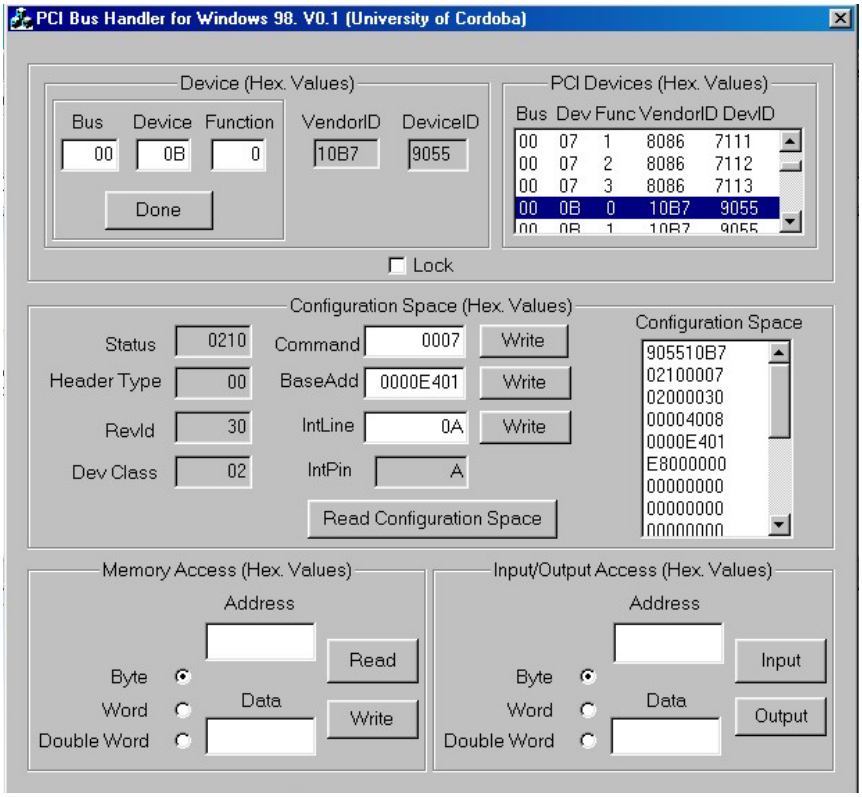

Figura 5. Aplicación "PCI Bus Handler"

La figura 5 muestra el interfaz de usuario de la aplicación donde se ha intentado unificar en una sola ventana todas las operaciones relacionadas con el bus PCI. Como ya se ha comentado en los puntos anteriores la finalidad de esta aplicación es el estudio de las operaciones más generales del bus como son el descubrimiento de todos los dispositivos conectados al bus, la configuración de dichos dispositivos y por último poder realizar transferencias con los dispositivos.

Desde el punto de vista del diseño de la aplicación se pueden distinguir dos partes: el interfaz al usuario, en este caso el alumno, y el acceso al bus PCI con un conjunto de funciones que acceden al espacio de configuración y que en este caso son las que resultan más interesantes por ser el objeto de estudio.

El acceso al espacio de configuración se realiza a través de un conjunto de funciones cuyos prototipos se muestran en la figura 6. En la figura 7 se puede ver con más detalle la función que accede al espacio de configuración con tamaño de datos de un byte.:

Lectura del espacio de configuración en tamaño byte, word y dobleword:

```
pciConfigInByte(int busNum,int deviceNum, int functionNum, int addrConfReg, UBYTE *pData);
pciConfigInWord(int busNum, int deviceNum, int functionNum ,int addrConfReg, UWORD *pData);
pciConfigInLong(int busNum, int deviceNum, int functionNum, int addrConfReg, UINT *pData);
```

Escritura en el espacio de configuración en tamaño byte, word y dobleword:

```
pciConfigOutByte(int busNum, int deviceNum, int functionNum, int addrConfReg, UBYTE *pData);
pciConfigOutWord(int busNum, int deviceNum, int functionNum, int addrConfReg, UWORD
*pData);
pciConfigOutLong(int busNum, int deviceNum, int functionNum, int addrConfReg, UINT *pData);
```

Figura 6. Prototipos de funciones para el acceso al espacio de configuración

```
    void pciConfigInByte
        (
        int busNum,          /* bus number */
        int deviceNum,       /* device number */
        int functionNum,     /* function number */
        int addrConfReg,     /* configuration space address*/
        UBYTE *pData         /* data read from the address */
         )
    {
            _outpd(ADDRESS_PORT_PCI, pciEmpaqueta(busNum, deviceNum, functionNum) |
                            (addrConfReg & 0xfc) | 0x80000000);
            *pData= _inp(DATA_PORT_PCI+(addrConfReg & 0x3));

        return;
}
```

Figura 7. Ejemplo de función para acceder a un registro del espacio de configuración

Como se puede observar en la figura 7, para acceder a un registro del espacio de configuración, se escribe primero la dirección del registro en la dirección CONFIG_ADDRESS (ADDRESS_PORT_PCI) y posteriormente se lee de la dirección CONFIG_DATA (DATA_PORT_PCI), que tendrá el valor del registro. La función "pciEmpaqueta" construye la dirección del dispositivo PCI y se debe hacer la OR lógica con *80000000h* para forzar el bit de más peso a 1, para que se considere un acceso válido tal como especifica la norma.

El interfaz de usuario de la aplicación es simple, los datos se introducen en las cajas de texto, y el código asociado a esta parte de la aplicación debe filtrar que no se introduzcan caracteres no hexadecimales o fuera de rango. Por otro lado, antes de realizar operación alguna sobre modificación de registros en el espacio de configuración, la aplicación debe asegurar que el dispositivo seleccionado en las cajas de texto correspondientes a "Bus", "Device " y "Function" existe.

A continuación vamos a describir el programa describiendo las actuaciones que se realizan en el bus PCI.

4.1. Descubriendo los dispositivos conectados al bus PCI

La primera etapa que se realiza en el bus PCI es el descubrimiento de los dispositivos conectados a él. En un computador real este trabajo lo realiza el firmware del equipo y/o el sistema operativo, por lo que el usuario no es consciente de esta búsqueda excepto en la fase de instalación o modificación del equipo. La primera sección de la ventana de la aplicación esta encaminada a las operaciones de descubrimiento de dispositivos (figura 8).

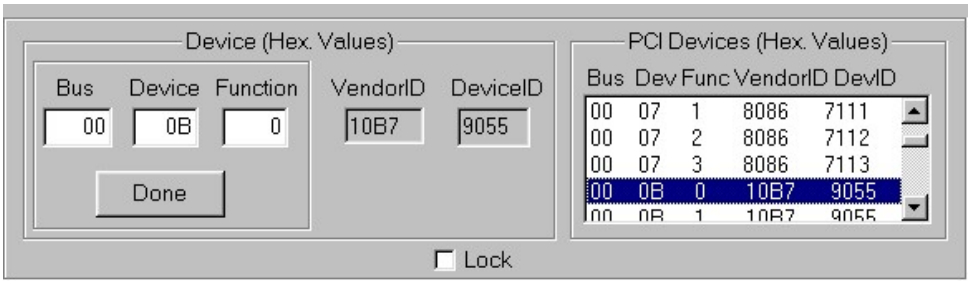

Figura 8. Descubrimiento de dispositivos del bus PCI

Existen dos formas de conocer si hay algún dispositivo conectado a una dirección de PCI. Una de ellas es realizando lo que en el bus se conoce como ciclos especiales y que por la

dificultad de comprensión no se utiliza en esta aplicación, la otra forma, más fácil de comprender, es acceder al registro "VendorId" y comprobando si se lee un valor distinto de *FFFFh*, siendo este último mecanismo el utilizado por nuestra aplicación. La función de búsqueda de dispositivos se puede expresar en pseudocódigo como muestra la figura 9.

```
For all_PCIbuses
    For all_PCIdevices
        Read VendorId          //function 0
        If VendorId!=FFFFh  //device exist
            For all function
                ReadConfigurationSpace() and show
            End for
        End if
    End for
End for
```

Figura 9. Pseudocodigo para el descubrimiento de dispositivos.

Los dispositivos localizados se muestran en la ventana derecha de la figura 8. En el caso de dispositivos de función única no se ha filtrado el acceso a los espacios de configuración correspondientes al resto de funciones porque la especificación del bus deja libre al diseñador de dispositivos que repita los espacios de configuración de la función única en todo el espacio de configuración del resto de funciones. Por este motivo, la ventana mostrará lo que haya decidido el fabricante. Una vez que se han descubierto todos los dispositivos, podemos acceder al espacio de configuración de uno concreto haciendo doble clic o rellenando los campos "Bus", "Device" y "Function" y haciendo clic en el control "Done". En este momento se mostrará el espacio de configuración del dispositivo seleccionado.

4.2. Visualizando y modificando el espacio de configuración

Para mantener seleccionado un dispositivo activamos el control "Lock", lo que provocará que se deshabiliten todos los controles que permiten el cambio de dispositivo. La aplicación muestra los campos del espacio de configuración obligatorios en el lado izquierdo y el espacio de configuración completo en la ventana del lado derecho (figura 10). Algunos campos son modificables por el alumno, por ejemplo, para mapear el dispositivo, activarlo o para asignarle una interrupción. Estas actuaciones deben realizarse con precaución ya que pueden provocar un malfunctionamiento del bus si solapamos el mapa de dos dispositivos o al fijar la interrupción.

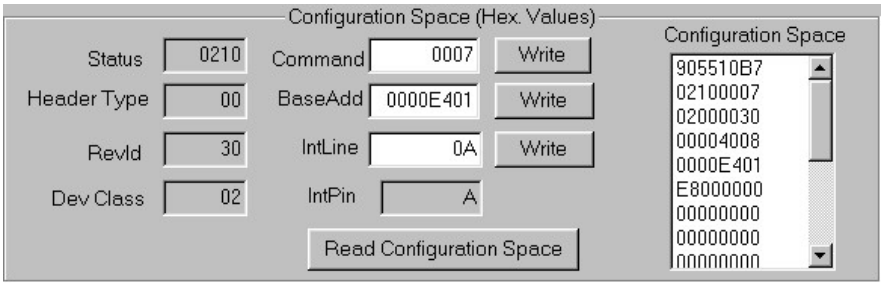

Figura 10. Espacio de configuración del dispositivo seleccionado

Cuando se escribe en un registro de configuración la aplicación realiza la escritura pero no se produce posteriormente una lectura automática para actualizar la caja de texto. El motivo es no provocar ciclos extras en el bus que dificulten el análisis del bus a nivel de cronograma

de señales cuando se utilice un analizador lógico. Si se desea leer el espacio tras una modificación, se debe hacer clic en el botón de lectura “Read Configuration Space”.

4.3. Accediendo al dispositivo

La parte inferior de la ventana permite realizar operaciones de entrada/salida o de lectura/escritura en memoria (figura 11). Lo habitual será que deseemos acceder al espacio de entrada/salida o de memoria que tendrá asignado el dispositivo con el que estamos trabajando pero aún así no se pone ninguna restricción y el alumno podrá realizar las transferencias que desee. Por ejemplo, si se quiere acceder al espacio de configuración se podrá realizar también desde la ventana de entrada/salida a través de los registros “CONFIG_ADDRESS” y “CONFIG_DATA”.

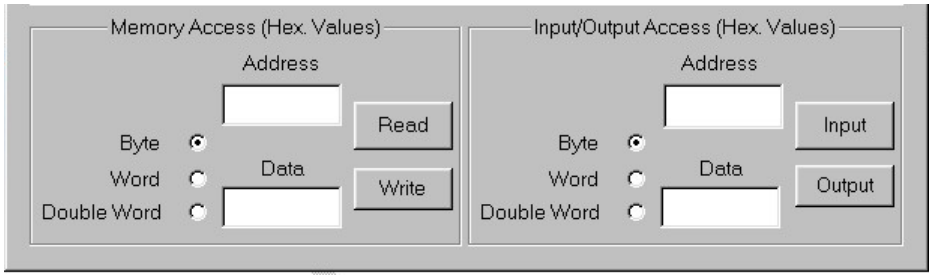

Figura 11. Acceso a memoria y entrada/salida

5. CONCLUSIONES

La aplicación mostrada en este artículo junto con una placa que se ha desarrollado para que el alumno pueda realizar cambios en su espacio de configuración y observar los ciclos de bus PCI utilizando un analizador lógico y los conectores que posee a tal efecto, son unas herramientas muy útiles para el estudio de este bus. Sin embargo, y a pesar de que sólo se estudian los ciclos más generales en el bus dejando de lado algunas funcionalidades más complejas, sigue siendo complicado para el alumno. No era este el caso del bus AT que, al ser un bus más simple y rígido, y ser la prolongación de un bus de un microprocesador que estudian en algunas asignaturas, les resulta más asequible. Este mismo efecto de mitificación o dificultad lo estamos observando con el bus USB frente a un puerto serie tradicional.

Lo que si supone una ayuda en el aprendizaje de este bus es que se puede disponer de una aplicación que visualiza y modifica el espacio de configuración. En cuanto a los Proyectos Fin de Carrera que estamos comenzando, esta aplicación junto con las placas que hemos desarrollado facilitan al alumno el desarrollo de placas con interfaz PCI, ya que pueden probar sus diseños VHDL en estas placas antes de que se fabrique la que está diseñando y detectar así posibles errores.

6. FUTUROS TRABAJOS

La línea de futuros trabajos está dirigida en la mejora y ampliación de esta aplicación ya que solo se puede utilizar con sistema operativo Windows 98 porque este sistema operativo no realiza protección en el acceso a memoria o recursos. Por tanto los primeros trabajos que deben ponerse en marcha es la migración a otros sistemas operativos como LINUX, Windows 2000 o NT.

Por otro lado, con esta aplicación sólo podemos conocer el mapa de memoria o entrada/salida de los dispositivos PCI y no el mapa completo. Sería muy útil crear una

aplicación que muestre los mapas de memoria y entrada salida completos, y disponer también de una aplicación que, en el caso de ordenadores personales, muestre el enrutamiento de las interrupciones.

7. BIBLIOGRAFÍA

[1] T. Shanley and D. Anderson, “PCI system architecture ”, Addison-Wesley, 1997.

[2] PCI Special Interest Group, ”PCI Local Bus Specification. Revision 2.2”, December 18,1998.

[3] PCI Special Interest Group, ”Bios PCI Local Bus Specification. Revision 2.2”, December 18,1998.

[4] PCI Special Interest Group, 2575 N. E. Kathryn #17, Hillsboro Oregon 97124, http://www.pcisig.com