

# Diseño de una Interfaz PCI para una Tarjeta Coprocesadora Basada en el DSP TMS320C40-40

*Ing. Israel Rivera Zárate.*

*Profesor e Investigador del CIDETEC-IPN*

*M. en C. Héctor Samuel García Salas.*

*Profesor e Investigador del CIDETEC-IPN*

*M. en C. Antulio Morgado Valle.*

*Profesor e Investigador del CIC-IPN*

**E**n este artículo se presenta el diseño de una interfaz orientada al bus PCI basada en el controlador S5933 de AMCC y lógica programable. La interfaz está integrada en una tarjeta de coprocesamiento basada en el DSP TMS 320C40-40 de Texas Instruments. La interfaz permite transferencias de datos en modos esclavo y maestro del bus PCI. La tarjeta coprocesadora está destinada al tratamiento de gráficos tridimensionales en tiempo real.

---

## INTRODUCCIÓN

---

El empleo de gráficos tridimensionales en tiempo real se explota ampliamente en aplicaciones de simulación de eventos y procesos, diseño de instrumentos y piezas, y en la Realidad Virtual. Gráficamente un escenario tridimensional se integra por objetos, un objeto es una construcción poligonal, esto es, un conjunto de cuerpos geométricos básicos (por ejemplo prismas, cubos, esferas, planos, etc.), que se definen a su vez por otros más elementales (polígonos), éstos por sus vértices y finalmente cada vértice por sus coordenadas espaciales. Esta información es representada en la computa-

dora como una serie de estructuras de datos referenciadas. Las rutinas principales de tratamiento de gráficos tridimensionales son: escalamiento, rotación, translación, eliminación de superficies ocultas, luminación, generación de color y despliegue

---

## DESCRIPCIÓN

---

La presentación gráfica en pantalla en tiempo real exige una frecuencia de exposición de 30 cuadros/seg., lo que conduce a un intervalo de tiempo de 33 mseg. Esto es, se dispone de 30 mseg para la ejecución de las rutinas de tratamiento gráfico y su presentación en pantalla. La resolución gráfica a emplear corresponde a los 320x200 pixeles y paleta de 256 colores (64 KB de datos). Conforme aumenta la complejidad de los escenarios, es decir, aumenta el número de objetos o bien de polígonos, la restricción temporal adquiere mayor importancia. Se sabe además, de la literatura relacionada [1], que el tiempo establecido para la transferencia de cada cuadro no debe exceder una décima parte del intervalo con el fin de no monopolizar el bus, esto es, no debe exceder 3.3 mseg. Lo anterior establece que la tarjeta debe ser capaz de transferir 64 KB en 3.3 mseg, esto es; la tarjeta debe proporcionar una tasa de transferencia de 19 MB/s.

Atendiendo a las necesidades de desempeño se seleccionó trabajar

con el bus PCI (Peripheral Component Interconnect - Interconexión de Componentes Periféricos) por ser un bus local de alto rendimiento, que además se está volviendo un estándar para los sistemas gráficos. La razón de su alto desempeño se debe a que conecta a los dispositivos periféricos directamente al bus del procesador del sistema, proporciona un ancho de datos de 32 bits y opera a una frecuencia de 33 Mhz que le permite desarrollar un ancho de banda de 132 MB/seg. Existe un bus local competitivo denominado VESA o VL-Bus (Video Electronics Standards Association - Asociación de Estándares Electrónicos de Video) que opera con una ruta de datos de 32 bits a una frecuencia de 50 Mhz pero que desarrolla un ancho de banda de tan sólo 107 MB/seg; debido a que requiere el empleo de varios ciclos de reloj por dato a transferir. Aunado a esto, los fabricantes de equipo de cómputo proporcionan de facto cada vez más, ranuras de expansión (slots) PCI en vez de VL-bus. Respecto a otros buses, se tiene por ejemplo el ISA (Industry Standard Architecture - Arquitectura Estándar de la Industria) cuyo ancho de banda no resulta adecuado a las necesidades ya que alcanza tan sólo los 8 MB/seg; debido a que opera con un ancho de datos de 8 o 16 bits a una frecuencia máxima de 8 Mhz, además que fue diseñado para la operación de periféricos lentos. Por su parte los buses EISA (Extended Industry Standard Architecture - Arqui-

ectura Extendida Estándar de la Industria) y MCA de IBM (Microchannel - Microcanal) aunque poseen un ancho de datos de 32 bits no desarrollan una alta tasa de transferencia por operar a frecuencias de 8.22 y 10 Mhz respectivamente.

La tarjeta coprocesadora se aloja en una computadora basada en el procesador Pentium de Intel @120 Mhz con un requerimiento mínimo de memoria de 16 MB y monitor Super VGA. El objetivo del diseño estriba en transferir cada cuadro gráfico confeccionado en la tarjeta hacia la memoria de cuadro (frame buffer) de la Adaptadora de Vídeo, mediante un acceso directo de memoria (DMA) en modo maestro del bus PCI sin intervención de la CPU anfitriona. Para llevar a cabo este propósito, se eligió el empleo del controlador de bus PCI S5933 de AMCC debido a sus capacidades de comunicación en modos esclavo y maestro del bus PCI, además de su gran flexibilidad para el diseño de interfaces con sistemas basados en microp procesadores.

#### INTERFAZ CON EL BUS PCI

El controlador de bus PCI S5933 de AMCC cuenta con tres interfaces de bus físicas: una hacia el bus PCI, otra hacia el bus Add-On, y una última hacia una memoria externa no volátil (donde se almacena el contenido del encabezado de configuración). Las transferencias de datos se realizan entre el bus PCI y el bus Add-On donde se interfaza el DSP C40. Las transferencias en el bus Add-On pueden realizarse a través de tres secciones funcionales integradas en el dispositivo: los registros tipo buzón (mailbox registers), las pilas tipo FIFO (first input first output - lo que entra primero sale primero), o bien, la trayectoria directa de datos (Pass-Thru data path) [2].

La comunicación hacia el bus PCI se realiza mediante lógica interconstruida en el controlador, lo que resta es determinar su forma de operar en el caso de la aplicación particular. La operación del controlador es definida en registros de control que pueden mapearse en el espacio de puertos o de memoria (definido en los registros del encabezado de configuración del dispositivo). Por lo tanto deben reservarse estos recursos del sistema.

La memoria ROM es la 24C02 de National Semiconductor, es del tipo serial, aunque puede utilizarse una de 8 bits; pero es condición necesaria para el manejo de otras terminales que permiten la operación en modo maestro del bus PCI. Esta memoria resguarda la información del encabezado de configuración y de otros parámetros de programación del controlador.

Los registros tipo buzón del S5933 ofrecen una ruta bidireccional de datos de 32 bits y son establecidos principalmente para la transferencia de datos de control de programa, comandos, e información del estado de la ejecución de aplicaciones. Además, se pueden configurar interrupciones dirigidas al bus PCI o al bus Add-On basadas en la ocurrencia de eventos específicos en los buzones.

La interfaz que establece una ruta directa entre el bus PCI y el bus de la aplicación Add-On permite mapear memoria integrada en la tarjeta, esto es: el procesador anfitrión accesa a la memoria de la tarjeta considerándola parte del sistema. Desde el punto de vista del procesador anfitrión la memoria está ubicada en direcciones de su mapa de memoria.

Por su parte, el DSP C40 cuenta con su propio espacio de direcciones y define la ubicación de la memoria donde le sea conveniente. La correspondencia entre la dirección genera-

da por el procesador anfitrión y el C40 recibe el nombre de mapeo, y en el caso particular, es realizado por el propio C40. La dirección de la memoria de la tarjeta en el mapa de memoria del anfitrión es asignada por el Bios o el sistema operativo durante la fase de inicialización de los dispositivos instalados en los buses de expansión, y que cumplen con el estándar plug and play [2],[3]. A través de esta interfaz el anfitrión puede enviar las estructuras de datos de nuevos escenarios a procesar.

El controlador S5933 cuenta con dos pilas separadas tipo FIFO. Una de ellas maneja el movimiento de datos del bus PCI hacia el bus Add-On y la otra lo hace en dirección opuesta. Ambas se integran por 8 Dwords (palabras de 32 bits). El bus PCI requiere que la transferencia de datos se haga en forma de ráfaga de datos (burst), es decir; una fase de direcciones y fases sucesivas de datos. El S5933 permite la comunicación de datos de manera síncrona o asíncrona a la señal de reloj del bus PCI de 33 Mhz. Un sistema logrará máxima transferencia sólo si es capaz de efectuar accesos en modo ráfaga síncronos.

En el caso particular, un DSP C40 desarrolla ciclos de bus de la mitad de la frecuencia de la señal de reloj, es decir; para el C40-40 son de 20Mhz. La diferencia de frecuencias entre el procesador y el controlador de bus impiden establecer de antemano un sistema síncrono a la frecuencia del bus PCI.

El análisis de diagramas de tiempo relacionados con la tasa máxima de transferencia posible a través de las interfaces Pass-Thru y FIFO, revelan que debido a su comportamiento asíncrono respecto al bus PCI y a la necesidad de un estado de espera debido a la lógica programable, ambas son capaces de desarro-

llar como máximo 20 MB por segundo. Por lo tanto en operación como maestro del bus, la interfaz diseñada satisface adecuadamente la tasa de transferencia requerida. Como esclavo, el controlador S5933 puede emplear cualquiera de las dos interfaces. Cabe mencionar que el controlador restringe la operación en modo maestro únicamente a la FIFO.

Por su parte, el DSP C40-40, cuenta con dos buses externos denominados como Local y Global respectivamente, cada uno consiste de un bus de datos de 32 bits, un bus de direcciones de 31 bits y de dos conjuntos de señales de control. El C40 ejecuta instrucciones en un solo ciclo de bus, centra su desempeño en una pipeline multietapa y en buses internos individuales para manejo de direcciones y datos para acceso a memoria de programa y memoria de datos, lo que le permite la ejecución de hasta 11 operaciones por ciclo. Ofrece un desempeño de 20 MIPS [4].

El C40 disminuye su rendimiento cuando en sus accesos a memoria opera con estados de espera, lo anterior repercute en la ejecución de la pipeline debido a que debe esperar antes de continuar en la siguiente

etapa. Para explotar su máximo desempeño se determinó el empleo de memorias de cero estados de espera (tiempo de acceso necesario=20 ns) tanto para almacenar el código, como para datos. Debido a que el C40 posee un bus de datos de 32 bits, su código está representado en formato de 32 bits, es decir; las instrucciones de programa tienen longitud de 32 bits. Por lo anterior se consideró el empleo de memorias con capacidad de 128Kx32.

Para lograr esto, para el almacenamiento del código se descartó el empleo de ROMs debido a su baja densidad y su alto tiempo de acceso. Para memorias de datos, se descartó el empleo de memorias dinámicas debido a la necesidad del empleo de controladores. También se descartaron las memorias cache por la necesidad de un controlador y su baja densidad.

El C40 cuenta con un programa almacenado en su ROM interna (Boot Loader Program – Programa Cargador de Arranque), que le permite copiar el contenido de una memoria ROM externa en una memoria RAM externa y pasarle el control. De esta forma arranca con una memoria lenta que requiere varios ciclos de espe-

ra pero ejecuta finalmente su código a máxima velocidad desde la RAM. Las memorias elegidas son SIMMs de Cypress con densidad de 128Kx32 y con un Tacc=20ns.

### INTERFAZ CON EL CONTROLADOR DE Bus PCI S5933

Atendiendo a las necesidades de comunicación asíncrona entre el controlador de bus PCI S5933 y el DSP C40, se realizó un estudio de los ciclos de bus de ambos dispositivos, llegando al establecimiento de una lógica programable basada en tres PLDs GAL22V10 (10ns). La programación de estos dispositivos permite la lectura y escritura a los registros del controlador del lado de la interfaz Add-On (donde se interfaza el DSP C40), el mapeo de memoria integrada en la tarjeta a través de la interfaz Pass-Thru, también la lectura y escritura a las FIFOs respectivamente, así mismo las acciones de control necesarias para volverse maestro del bus PCI y la lógica de generación de estados de espera del C40. Ver figura 1.

### DECODIFICADOR DE DIRECCIONES

En esta aplicación, se destinó una memoria de datos y una memoria de

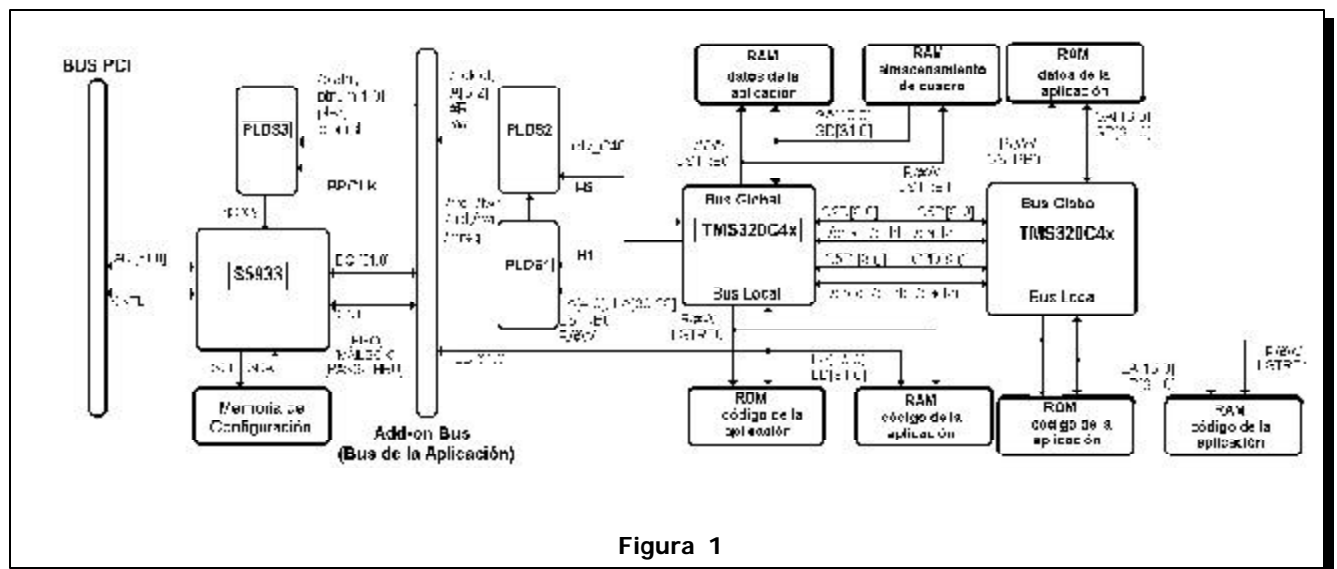


Figura 1

cuadro (para el almacenamiento del cuadro a transmitir vía DMA) del lado del bus global, en tanto que el controlador S5933 es accesado como un dispositivo mapeado dentro del espacio de direcciones del bus local. El espacio de direcciones del bus local se dividió en dos regiones principales mediante la escritura al campo STRB ACTIVE del registro de control de memoria local; una abarca la ROM de código de la aplicación y al controlador, y la otra a la RAM donde se respalda la ROM de código para ejecución a máxima velocidad. La GAL22V10 etiquetada como PLDS1 decodifica los ciclos de bus local del C40 empleando la fase de reloj H1 del C40. Además, mediante la señal de control STRB0 y las líneas de direcciones LA30-LA28 y LA4-LA0 (LA31 es implícitamente un 0 lógico), divide la región en 4 secciones correspondientes con los registros de control e interfaces del bus Add-On del controlador S5933. El mapa de memoria del C40 decodificado por el PLDS1 aparece en la tabla 1.

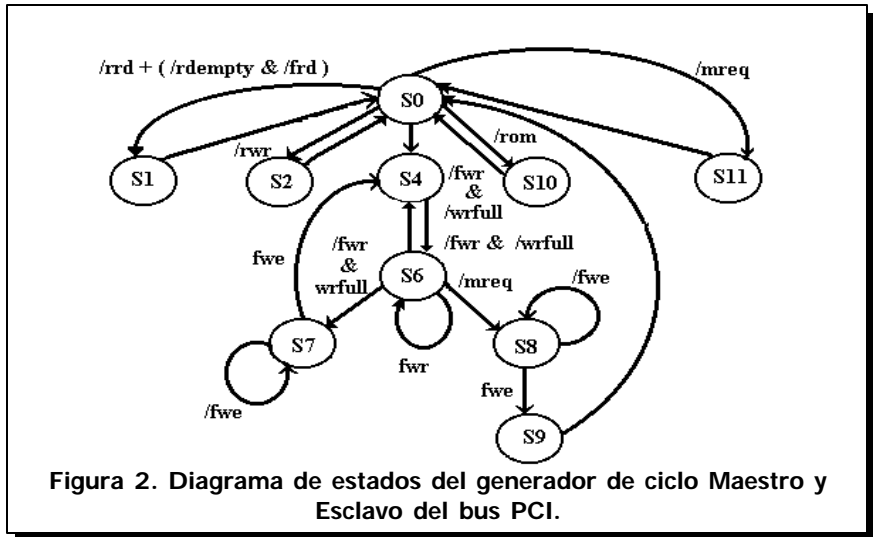


Figura 2. Diagrama de estados del generador de ciclo Maestro y Esclavo del bus PCI.

**GENERACIÓN DEL CICLO MAESTRO DEL BUS PCI**

El DSP C40 genera ciclos como maestro del bus mediante accesos de escritura a la interfaz FIFO.

Para identificar esta condición la salida del PLDS1 es conducida a un segundo PLD, etiquetado como PLDS2. Cuando la FIFO alcanza su máxima profundidad (8 Dwords), la lógica integrada en el controlador para el manejo de la FIFO genera en el bus Add-On la señal WRFULL. El PLDS2 es una máquina de estados sincronizada con la fase de reloj H3 del C40, y ante la presencia de la señal WRFULL genera la señal de control AMWEN requerida por el controlador para solicitar la posesión del bus. Para garantizar que el C40 no accese a la pila bajo esta condición, genera los estados de espera necesarios manipulando adecuadamente la señal LRDY0 del C40. La razón del empleo de la fase de reloj H3 se justifica en términos de un menor empleo de estados de espera atribuidos a los retardos de propagación de la lógica programable. El PLDS2 genera además las señales de control para el acceso a la ROM de código, el acceso asíncrono a los registros de control y a los registros

tipo buzón. El diagrama de estados de este PLD aparece en la figura 2.

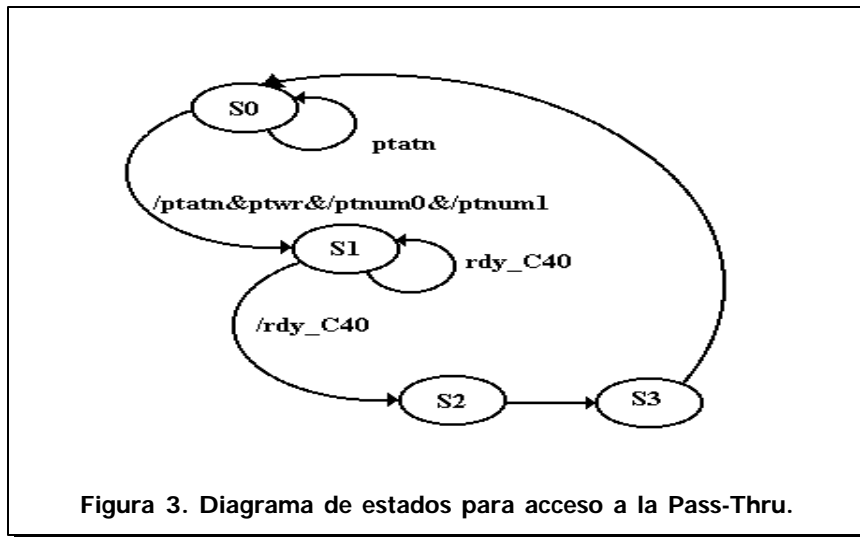
**GENERACIÓN DEL CICLO ESCLAVO DEL BUS PCI**

El C40 acepta accesos como dispositivo esclavo del bus PCI a través de la FIFO o la Pass-Thru. El PLDS2 controla los accesos basados en la FIFO. Para realizar accesos mediante la Pass-Thru se hace necesario guardar sincronía con el reloj del bus PCI. Por esto, el PLD etiquetado como PLDS3 constituye una máquina de estados sincronizada con la señal BPCLK generada por el controlador S5933. La señal BPCLK es la versión reforzada de la señal de reloj PCI.

La interfaz Pass-Thru cuenta con una terminal de entrada para la generación de estados de espera, el PLDS3 maneja esta terminal tomando en cuenta el estado de la señal LRDY0 que le es suministrada por el PLDS2. El diagrama de estados aparece en la figura 3.

Dirección	Unidad Direccionada
0000 0000h	ROM interna (Programa de Carga para Arranque)
3000 0000h	RAM código (Respaldo de la ROM de código)
4000 0000h	Solicitud Maestro PCI
5000 0000h	Mailboxes
5000 0008h	FIFO
5000 000Ah	Direcciones Pass-Thru
5000 000Bh	Datos Pass-Thru
5000 000Ch	Control
6000 0000h	ROM externa (Código de la aplicación)
8000 0000h	RAM datos (Datos de la aplicación)

**Tabla 1. Asignación del espacio de direcciones del DSP C40.**



### CONCLUSIONES

Actualmente la tarjeta se encuentra en fase de prueba de las rutinas de comunicación con la tarjeta y se basan en las bibliotecas escritas en lenguaje C proporcionadas por AMCC. El código de tratamiento gráfico se elaboró en lenguaje C y actualmente se está depurando en el programa de simulación del DSP C40 de Texas Inst. La integración del sistema está muy próxima y dará paso a pruebas y caracterización del prototipo.

### BIBLIOGRAFÍA

- [1] "Virtual Reality Systems". R. A. Earn Shaw. Academic Press, 1993.
- [2] AMCC S5933 PCI Controller Data Book, 1997.
- [3] "PCI System Architecture". Tom Shanley. Addison Wesley, 1994.
- [4] TMS320C40 User's Guide. 1993.

