Evolución tecnológica del hardware de vídeo y las GPU en los ordenadores personales

Francisco Charte, Antonio J. Rueda, Macarena Espinilla, Antonio J. Rivera

Departamento de Informática. E.P.S. Universidad de Jaén.
{fcharte,ajrueda,mestevez,arivera}@ujaen.es

Resumen. En este artículo se ofrece una revisión de los hitos más importantes en la evolución del hardware gráfico. La comunicación entre los ordenadores y las personas ha ido avanzando a lo largo del tiempo, alcanzando la interactividad con la aparición de los sistemas de tiempo compartido a principios de la década de los 60 del siglo pasado. Los ordenadores personales, cuya expansión se inició casi dos décadas después, adoptaron desde un inicio la visualización de información en una pantalla como medio principal de comunicación con el usuario. El hardware a cargo de esa tarea ha ido evolucionando paulatinamente hasta, en la actualidad, convertirse en parte indispensable de la arquitectura del computador, hasta tal punto que una gran parte de los ordenadores portátiles y de sobremesa incorporan el hardware gráfico en el mismo circuito integrado que aloja al microprocesador.

Palabras Clave: Vídeo, hardware gráfico, GPU, shaders.

Abstract. This article provides a review of the most important milestones in the evolution of graphics hardware. Communication between computers and people has been advancing over time, reaching interactivity with the emergence of time-sharing systems in the early 1960s. Personal computers, whose expansion began almost two decades later, used the visualization of information on a screen as the main means of communication with the user from the very beginning. The hardware in charge of this task has gradually evolved to become an indispensable part of the computer architecture, to such an extent that a large part of laptops and desktop computers incorporate the graphic hardware into the same integrated circuit that houses the microprocessor.

Keywords: Video, graphics hardware, GPU, shaders.

1 Introducción

Los primeros ordenadores disponibles comercialmente, y que comenzaron a emplearse en empresas como las aerolíneas norteamericanas [1], grandes bancos y algunas universidades, los conocidos como *mainframes*, eran sistemas extremadamente caros, por lo que aprovechar al máximo su tiempo de procesamiento era un objetivo esencial. Para

conseguirlo, las tareas ejecutadas por estas máquinas se llevaban codificadas en tarjetas perforadas a un centro de procesamiento de datos, donde se ponían en cola a fin de que el ordenador fuese ejecutándolas por lotes. No se contemplaba la interacción directa entre el ordenador y el usuario final, ya que esto conllevaba dedicar el valioso tiempo de uso de la máquina a una sola persona.

En 1961 el MIT (*Massachusetts Institute of Technology*) llevó a cabo una demostración de uno de los primeros sistemas de tiempo compartido [2]. Este tipo de sistemas permiten que múltiples usuarios interaccionen con el ordenador a través de una terminal que, originalmente, era un dispositivo similar a un teletipo: una impresora con un teclado. El usuario introducía los comandos a ejecutar a través del teclado, empleado como una máquina de escribir, y recibía la respuesta del sistema directamente en papel. El ordenador atendía simultáneamente a múltiples usuarios, dividiendo su tiempo entre todos ellos, pero debido a su velocidad cada usuario tenía la noción de estar trabajando con el sistema a su completa disposición. El principal obstáculo para esta vía de comunicación, aparte del coste de este tipo de terminales y el gasto de papel y tinta, era la velocidad de comunicación, limitada por la propia velocidad de impresión del dispositivo.

La aparición del primer terminal basado en un sistema de vídeo, una pantalla con un teclado, se retrasó casi una década, hasta que en 1969 DataPoint Corp. lanzó el DataPoint 3300 [3]. Este era un terminal que, conectándose al ordenador por la misma vía que se había usado hasta entonces para el teletipo, emulaba el funcionamiento de este, pero prescindiendo del papel. La mayor dificultad para el desarrollo de un terminal así, basado en un sistema de vídeo, estribaba en la cantidad de memoria que se necesitaba para almacenar una página (una pantalla completa) de información a fin de visualizarla en la pantalla. La capacidad podía, en algunos casos, exceder la memoria con la que contaban los ordenadores de la época. El abaratamiento de la memoria RAM (*Random Access Memory*), a partir de la introducción del Intel 1103 en 1970 [4], permitió superar ese obstáculo y popularizar el uso de los terminales basados en vídeo.

Cuando los ordenadores personales, también conocidos como microordenadores [5], comenzaron a estar al alcance del público en general, a partir de mediados de la década de los 70 y, sobre todo, en los 80, el uso de un sistema de vídeo como vía principal de comunicación con el usuario era algo estándar. Estos equipos incorporaban como parte del propio ordenador la memoria necesaria para almacenar la información a mostrar en la pantalla, así como el hardware necesario para convertir el contenido de esa memoria en una señal visible en pantalla. En este momento nace el hardware de vídeo integrado en el ordenador, cuya evolución es el objeto del presente artículo.

Al tratar el progreso en los adaptadores de vídeo, o adaptadores gráficos, destinados a ordenadores personales suele considerarse un agrupamiento en generaciones, atendiendo a la funcionalidad ofrecida por el hardware en cada momento. En la sección 2 de este artículo se ofrece una visión general sobre cómo el hardware de vídeo ha cambiado en las últimas décadas. La sección 3, centrada ya en los últimos 20 años, describe las características fundamentales de las cinco primeras generaciones de adaptadores de vídeo para ordenadores personales. En la sección 4 se detalla cómo ha cambiado el cauce interno o *pipeline* gráfico a lo largo de esas generaciones.

2 Visión general del hardware de vídeo en ordenadores personales

La evolución del hardware de vídeo utilizado en los ordenadores personales ha sido vertiginosa en las últimas décadas, llegando en la actualidad a ofrecer una potencia de cálculo muy superior a la de las costosas estaciones de trabajo[1] (*workstations*) que hasta hace veinte años copaban el segmento profesional, dedicado principalmente a tareas de CAD (*Computer Aided Design*), CAM (*Computer Aided Manufacturing*), CAE (*Computer Aided Engineering*), investigación y simulación científica, etc.

Del clásico adaptador de vídeo, dotado únicamente de la circuitería necesaria para convertir la información digital, almacenada en memoria[2] relativa a los atributos de los píxeles, en una señal analógica adecuada para ser enviada a un monitor, se ha pasado al campo de las GPU (*Graphics Processing Unit*) [6], procesadores especializados con un alto nivel de paralelismo y que eximen al microprocesador de todas las tareas gráficas. Entre ambos extremos han quedado las tarjetas de vídeo que, aparte del adaptador, ya contaban con una cierta cantidad de memoria propia, así como aquellas capaces de llevar a cabo una parte del trabajo que hasta ese momento recaía en la CPU (*Central Processing Unit*), como la aplicación de texturas, primero, y la ejecución de transformaciones, después, a través de operaciones implementadas en hardware y sobre las que el programador tenía muy poco control, más allá de establecer algunos parámetros limitados.

Las actuales GPU también cuentan con el hardware necesario para llevar a cabo todas esas tareas sin depender de la CPU pero, además, el tratamiento aplicado a cada vértice o fragmento es programable en mayor o menor medida, dependiendo del hardware concreto y la API (*Application Programming Interface*) que se utilice. Todo ello permite un control mucho mayor por parte del desarrollador a la hora de establecer transformaciones, definir algoritmos de sombreado e iluminación, aplicar texturas, etc.

2.1. Hardware de vídeo en microordenadores

Los microordenadores[3], sistemas surgidos desde mediados de la década de los 70 hasta principios de los 90 [5] con microprocesadores de 8 bits y en algunos casos de 16 bits, eran productos dirigidos al usuario doméstico, razón por la que el precio era un factor determinante. El hardware de vídeo de estas máquinas era, en la mayoría de los casos, muy simple.

---

[1] A diferencia de los *mainframes*, las *workstations* eran ordenadores diseñados su uso por una sola persona, aunque solían estar conectadas en red, contando con capacidad de procesamiento y para mostrar gráficos en alta resolución, a diferencia de los terminales conectados a los *mainframes* y los ordenadores personales de la época, que carecían en general de capacidades gráficas-

[2] Los primeros adaptadores de vídeo usados en los ordenadores personales ni siquiera contaban con una memoria propia, convirtiendo en señal de vídeo la información que se almacenaba en una cierta porción de la propia memoria principal del sistema. El único aspecto configurable era la dirección de memoria a partir de la que se leían los píxeles.

[3] En http://museopc.ujaen.es/ puede encontrarse información detallada sobre muchos de los microordenadores mencionados en esta sección.

En el caso del Commodore PET (véase la Figura 1), el primer ordenador comercializado por Commodore en el año 1977, el hardware de vídeo se diseñó a medida usando circuitos TTL discretos a fin de generar la imagen enviada al monitor integrado. Muchos otros ordenadores de la época, incluyendo los Apple I y Apple II, Jupiter ACE, Osborne 1, TRS-80 y Sinclair ZX-80, usaban este mismo enfoque para la producción de gráficos. El hardware de vídeo a medida continuó utilizándose en equipos de principios de los 80, aunque integrando toda la circuitería TTL (*Transistor-Transistor Logic*) en un único chip tipo ASIC (*Application Specific Integrated Circuit*). Este fue el caso de varios modelos de Sinclair, como los ZX-81, ZX-Spectrum y QL, con su ULA (*Uncommited Logic Array*), y de otros producidos por Thomson, Oric y Acorn.

Figura 1. Commodore PET 2001 (1977).

La circuitería de vídeo a medida tenía su contrapartida en el hardware de vídeo genérico basado en un CRTC (*Cathode Ray Tube Controller*), un circuito estándar encargado de generar las señales de sincronización y control para poder mostrar una imagen en pantalla. Uno de los CRTC más usados fue el Motorola 6845. El ordenador aportaba la memoria en la que se almacenaban los datos a partir de los cuales se generaba dicha imagen, encargándose además de todas las tareas gráficas. Algunos de los modelos

cuyo hardware de vídeo se basaba en un CRTC fueron los Commodore PET 4000/8000, Kaypro y microordenadores japoneses como el Sharp MZ-700 y Sharp X1.

El incremento en el número de fabricantes de microordenadores y de unidades vendidas provocó la aparición de circuitos integrados específicos que operaban como controladores de vídeo. Estos se encargaban no solamente de generar la imagen, como los CRTC, sino también de gestionar la configuración de los distintos modos de texto y gráficos, generar los juegos de caracteres, configurar las paletas de color, etc. Descargan, por tanto, de una cierta cantidad de trabajo al microprocesador, pero este seguía siendo el responsable de generar las imágenes a mostrar finalmente en pantalla. Uno de los controladores de vídeo más conocidos de la época era el VIC-II, al ser el que integraba el Commodore 64 que, durante mucho tiempo, fue el ordenador más vendido de la historia.

Desde mediados de los 80 en adelante la mayor parte de los microordenadores incorporan procesadores de vídeo más sofisticados, ya fuesen genéricos (fabricados por un tercero y disponibles en el mercado) o diseñados a medida. En el primer grupo estarían los VDP (*Video Display Processor*) de Yamaha, como el V9938/V9958 usado en los MSX2 y MSX2+. Del segundo forman parte integrados míticos como los "Agnus" y "Denise", diseñados para el Commodore Amiga, o el "Nick" de los Enterprise 64/128. Estos circuitos contaban normalmente con su memoria dedicada, conocida como VRAM (*Video RAM*), y operaban de forma paralela e independiente a la CPU, siendo capaces de resoluciones mucho más altas y usando miles de colores. Fueron, en su momento, lo más cercano a una GPU actual.

2.2. Hardware de vídeo en los orígenes del PC

La mayoría de los ordenadores usados actualmente tienen sus orígenes en la arquitectura del IBM PC, presentado en agosto de 1981. Incluso los Apple Mac actuales, desde el cambio introducido en el año 2006 en su fabricación, tienen la arquitectura de un PC. Aquel PC original era un sistema pensado para la empresa, razón por la que el sistema de vídeo con el que se vendía, conocido como MDA (*Monochrome Display Adapter*), fuese monocromo, al igual que el monitor. Se ponía el énfasis en la calidad del texto. No obstante, IBM ofrecía como opción cambiar el vídeo MDA por un adaptador CGA (*Color Graphics Adapter*).

CGA contemplaba dos modos gráficos: uno de baja resolución con 4 colores y otros de alta resolución con solo 2. En modo texto se disponía de una paleta de 16 colores. El adaptador de vídeo CGA podía conectarse a un monitor específico, como el MDA, pero también directamente a un TV, abriendo la puerta a usos más domésticos. IBM desarrolló CGA a partir del Motorola 6845 anteriormente mencionado, siendo por tanto un CRTC con algunas características más avanzadas, entre ellas la inclusión de 16Kbytes de memoria para almacenar los datos de la imagen.

Un año después del lanzamiento del PC, con su configuración inicial MDA/CGA, apareció el primer competidor de IBM en el campo del hardware gráfico: la empresa Hercules. Esta diseñó un adaptar de vídeo (véase la Figura 2) cuya principal característica era la de ofrecer un modo de texto compatible con MDA y un modo gráfico con mayor resolución que CGA. Dicho modo, no obstante, era monocromo, pensado más para la empresa que para su uso doméstico.

Hercules fue la primera, pero en los años siguientes muchas otras empresas se lanzaron al negocio de la fabricación de hardware de vídeo para PC. La competencia entre dichas empresas se concentró inicialmente en ofrecer cada vez mayor resolución, tanto en número de píxeles como en número de colores en pantalla. Estos avances se reflejaron en la comercialización de los conocidos adaptadores EGA (*Enhanced Graphics Adapter*), VGA (*Video Graphics Array*), SVGA (*Super VGA*), etc. La principal diferencia entre el hardware ofrecido por cada empresa estribaba en la cantidad de memoria que incorporaba el adaptador, un factor que restringía los modos gráficos que podía representar.

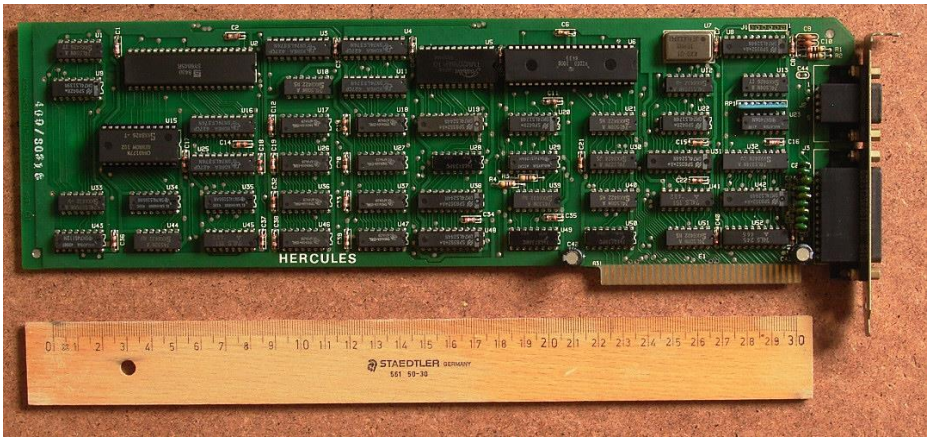

Figura 2. Adaptador de vídeo Hercules comercializado en 1984.
Fotografía de Rainer Knäpper reproducida bajo Licence Art Libre.

El concepto de adaptador de vídeo basado en el CRTC Motorola 6845, con una cantidad mayor de memoria, un generador de caracteres y gestión de paletas de color, era prácticamente todo lo que aportaba este hardware de vídeo, manteniendo la compatibilidad hacia atrás con CGA. Desde el momento en que los fabricantes incorporaron al adaptador de vídeo capacidades adicionales, en principio un procesador capaz de aplicar ciertas funciones automáticamente a los píxeles de una imagen, podemos comenzar a hablar de GPU. A partir de entonces el hardware gráfico no se limita a tomar el contenido de la memoria de vídeo y enviarlo a una pantalla, sino que interviene en el proceso de generación de la propia imagen.

3 Generaciones de los adaptadores de vídeo

Hasta hace pocos años todos los procesadores de tarjetas gráficas, lo que se conoce como GPU, contaban con un conjunto de funciones fijas para la generación de imágenes. Si imaginamos que la GPU es como una tubería en la que introducimos por un extremo, el de entrada, un flujo de datos, normalmente coordenadas de vértices, y obtenemos por el otro, el de salida, una imagen, esas funciones fijas permitían ajustar,

siempre dentro de unos límites preestablecidos, aspectos como la iluminación, las transformaciones y aplicación de texturas. Sería, por tanto, como tener a lo largo de la tubería una serie de interruptores que pueden tomar una de un conjunto limitado de posiciones.

El nacimiento de las GPU programables, a partir de 2001, supone un salto cualitativo crucial, ya que el programador puede cambiar porciones de esa hipotética tubería sustituyendo las funciones fijas por funciones a medida, eliminando cualquier límite en cuanto a los efectos gráficos que pueden generarse en tiempo real. Esas funciones se denominan coloquialmente sombreadores o *shaders* [7], aunque realmente su función no es únicamente la de aplicar sombreados. Una función de este tipo se envía como código ensamblador al controlador de la tarjeta gráfica, encargándose esta de generar el código ejecutable que, alojado en la propia GPU, procesará cada uno de los vértices o fragmentos de una escena.

Para comprender la importancia que tiene la existencia de GPU programables, y situar este hecho en su contexto actual, lo mejor es echar la vista atrás y analizar qué ofrecían las GPU del pasado. En realidad, se trata de un pasado muy reciente, ya que la historia de las GPU tiene apenas dos décadas. Hasta la aparición de los primeros adaptadores de vídeo con GPU, en 1998, todo el trabajo de generación de gráficos recaía en la CPU del ordenador, limitándose el mencionado adaptador [8] a generar la señal de vídeo, tal y como se apuntaba anteriormente. Todo el cálculo lo efectuaba la CPU y el hardware de vídeo era poco más que un DAC (*Digital Analogic Converter*).

Es a partir de la presentación de las Voodoo3 y TNT, denominaciones del hardware gráfico de las empresas 3dfx y nVidia[4], cuando se comienza a hablar de las GPU. A medida que estas fueron incorporando capacidades adicionales se empezaron a agrupar en familias y, posteriormente, en generaciones. Hoy en día las generaciones reconocidas históricamente son las mencionadas en las subsecciones siguientes.

3.1. La primera generación

Se trataba de GPU con capacidad para aplicar sobre los píxeles una serie de operaciones matemáticas simples, con el objetivo de calcular el color final facilitando así la aplicación de texturas. Además, *rasterizaban* [9] directamente los triángulos, de manera que se eximía a la CPU de la manipulación de píxeles individuales.

La aplicación de transformaciones a los vértices, y cualquier otra tarea que no fuese la rasterización y aplicación de una o dos texturas (dependiendo de los modelos) seguía quedando completamente en manos de la CPU. No obstante, el trabajo que realizaba la GPU mejoraba de manera notable el rendimiento, respecto a sistema sin GPU, llegando a alcanzarse tasas de entre 6 y 9 millones de polígonos por segundo.

Esta primera generación abarca temporalmente desde 1998 hasta mediados de 1999, formando parte de ella productos como las ATI Rage, nVidia RIVA TNT y la Voodoo3 (Figura 3) de la hoy desaparecida empresa 3dfx Interactive [10]. Hay que tener en cuenta que estas GPU no eran programables, sino "configurables", es decir, el programador podía establecer una serie de parámetros que determinaban cómo se efectuaría el rasterizado o la manera en que se aplicarían las texturas, pero nada más. El *pipeline*

[4] En http://www.tomshardware.com/reviews/nvidia,87.html podemos encontrar una comparativa de la época de estos dos productos.

de estas GPU, por tanto, tenía una estructura fija que llevaba a cabo siempre las mismas operaciones, con unos parámetros ligeramente variables.

Como curiosidad, en su época se denominaba a estos adaptadores "aceleradoras 3D", para distinguirlas de los adaptadores de vídeo corrientes. Algunos de ellos, como era el caso de la Voodoo3, incluían funciones de procesamiento en 3D pero, por el contrario, carecían de la capacidad para operar en 2D, por lo que este tipo de gráficos seguían siendo generados al completo por la CPU.

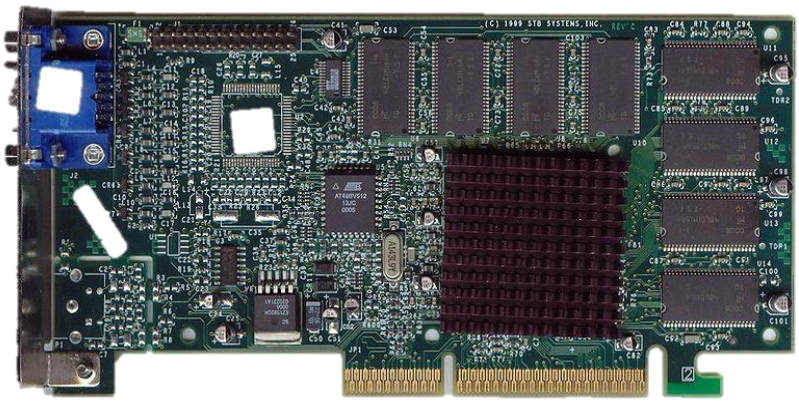

Figura 3. Voodoo3 con conector AGP[5].

3.2. La segunda generación

Esta generación se inicia con la aparición de las primeras GPU capaces de llevar a cabo operaciones de transformación e iluminación 3D, funciones que se agregan a la rasterización y aplicación de texturas. Abarca desde finales de 1999 hasta la aparición de la siguiente generación, ya en 2001.

Los representantes más destacables de esta generación son las ATI Radeon 7500, S3 Savage3D y nVidia GeForce 256 (anunciada por nVidia como la primera GPU del mundo, según se aprecia en la Figura 4) y, posteriormente, la GeForce2. Era un hardware capaz de procesar entre 15 y 25 millones de polígonos por segundo

El cauce de procesamiento de estos dispositivos seguía siendo configurable, no programable, si bien integraba funciones matemáticas más complejas y, además, tenía la

[5] Imagen de Wikimedia Commons.

posibilidad de utilizar aritmética con signo y de mayor precisión. Además de los cambios en el cauce gráfico, cada generación introdujo mayor integración y velocidad de la GPU, nuevas interfaces de conexión (PCI, AGP, PCIe, etc.) y avances en la velocidad de la memoria. Son aspectos, no obstante, que quedan fuera del ámbito de este trabajo.

Figura 4. Anuncio de la GeForce 256 por parte de nVidia.

3.3. La tercera generación

Es a principios de 2001, con la presentación del circuito integrado GeForce3, cuando aparece el primer dispositivo que podría calificarse realmente como GPU, al incorporar por primera vez la posibilidad de programar una parte del *pipeline*. También forman parte de esta generación, que se extiende hasta 2003, productos como la ATI Radeon 8500 (Figura 5), la GeForce4 y el hardware gráfico fabricado específicamente para la consola Xbox de Microsoft.

Los procesos de rasterización y aplicación de texturas siguen ejecutándose en unidades configurables de función fija, pero las transformaciones aplicadas a los vértices de la geometría pasan a ser programables. Aparecen los denominados *vertex shaders*, que abren las puertas a que, por primera vez, los programadores definan funciones a medida que serán ejecutadas no por la CPU, sino por la GPU.

A pesar de la posibilidad de definir *vertex shaders*, estos primeros diseños de cauce programable tenían muchas limitaciones y restringían, por ejemplo, el número de parámetros de entrada o el número de instrucciones que podía tener como máximo un *shader*. Además, estos programas se ejecutaban siempre de manera secuencial, lo que implicaba que no podían definirse bucles.

Figura 5. Circuito integrado ATI Radeon 8500.

Los productos de esta generación eran capaces de procesar entre 30 y 60 millones de polígonos por segundo, volviendo a multiplicar la potencia de la generación anterior que, a su vez, había multiplicado la de la primera generación.

La aparición del cauce parcialmente programable en el hardware gráfico marca también la aparición de los primeros lenguajes de alto nivel para la programación de *shaders*, incorporados en DirectX 8.0 y en OpenGL a través de una extensión. Previamente era preciso recurrir al lenguaje ensamblador propio de cada GPU.

3.4. La cuarta generación

Si en la tercera generación se añadió la capacidad para programar la unidad encargada de procesar los vértices de la geometría, en la cuarta se extiende esa capacidad a otro elemento del *pipeline*: la unidad encargada del *rastering*, incluyendo la texturización, sombreado y resolución de visibilidad. Nacen así los denominados *fragment shaders* o *pixel shaders*.

Las GPU de esta generación son casi completamente programables, pero tienen una estructura heterogénea en la que se dedican unos núcleos de procesamiento concretos a trabajar con vértices y otros a tratar fragmentos (*pre-píxeles*). En ambos casos el número de unidades es reducido, por ejemplo 4 núcleos de procesamiento de fragmentos en los primeros diseños, y así lo son también los programas que es posible escribir, con una longitud máxima de 12 instrucciones y no pudiendo operar con más de 4 texturas de manera simultánea.

Son miembros de esta generación las familias GeForce FX de nVidia y Radeon 9x00 de ATI, con capacidad para procesar hasta 200 millones de polígonos por segundo. Temporalmente hablando, esta generación se solapa parcialmente con la anterior, abarcando los años 2002 y 2003.

DirectX 9 [11] contemplaba la programación de los nuevos *pixel shaders*, generando un código que, con posterioridad, era compilado y optimizado por parte del controlador específico del fabricante, según el esquema de la Figura 6. Ese código, ya ejecutable, era el que se enviaba al hardware. OpenGL, por su parte, contaba con una extensión adicional, denominada `ARB_fragment_program`, que ofrecía una interfaz similar.

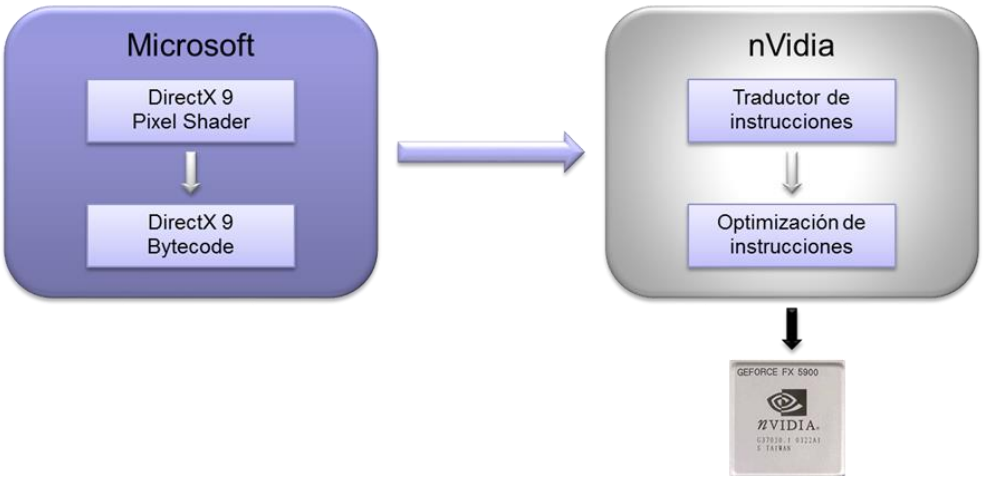

Figura 6. Pasos desde la escritura de un *pixel shader* en DirectX 9 hasta la ejecución.

3.5. Quinta generación y posteriores

Algunos autores sitúan la quinta generación de GPU aproximadamente a partir de 2009, mientras que otros hablan de una quinta generación entre los años 2004 y 2005, denominando sexta generación a la posterior, que abarca desde entonces hasta el presente. En cualquier caso, lo importante es conocer las características básicas de una generación y otra.

En los productos aparecidos entre 2004 y 2005 se introducen algunos avances en la tecnología ya existente, aunque sin que se produzca un salto sustancial en la misma. En estos productos los *vertex shaders* tienen, por primera vez, acceso a la memoria de la tarjeta de vídeo, no solamente a la información de los vértices que están procesándose. Por su parte, los *fragment shaders* pueden tener una mayor longitud y, también por primera vez, pueden ejecutar saltos y, en consecuencia, abren la puerta al uso de bucles. A esta generación corresponden la familia 6800 de nVidia y la ATI X1800.

Con 2006 llegan al mercado las GPU que componen la quinta generación, cuyos máximos representantes son los núcleos G80 de nVidia y R600 de ATI, usados en varias gamas de producto de ambos fabricantes. Las características más destacables de esta generación, que vienen acompañadas de una nueva versión de la especificación de

shaders y también una actualización de DirectX, son las siguientes:

- Unificación de los núcleos de la GPU [12], de forma que no están especializados en tratamiento de vértices o píxeles, como en las generaciones previas, sino que tienen una función más genérica y pueden de esta forma distribuir mejor el trabajo que requiera cada programa. Es el modelo *Unified Shading Architecture*.
- Aparición de un nuevo tipo de programa para GPU, denominado *geometry shader* [13], incrementando así las funciones que pueden ejecutarse en la GPU y que, hasta el momento, se llevaban a cabo en la CPU. Tras estos aparecieron los denominados *tessellation shaders* y más recientemente los *compute shaders*. Estos últimos permiten implementar computación de propósito general mediante *shaders*.
- Aunque el término GPGPU (*General Purpose-computation on Graphics Processing Units*) [14], que hace referencia a la ejecución en la GPU de algoritmos de propósito general, realmente nació con la tercera generación de GPU, para poder ejecutar algoritmos de propósito general mediante los *shaders* disponibles en ese momento había que recurrir a determinadas técnicas y trucos que no resultaban sencillos. La aparición de lenguajes como CUDA y OpenCL, conjuntamente con la quinta generación de GPU, simplificó enormemente esta tarea. La flexibilidad a la hora de programar la GPU, con la aparición de dichos lenguajes de más alto nivel, da lugar al nacimiento del GPGPU moderno o *GPU Computing*.

Desde la aparición de los citados núcleos G80 y R600 en adelante, la propuesta de los fabricantes ha ido básicamente encaminada a incrementar el número de núcleos de procesamiento con que cuentan las GPU, aumentar la cantidad de memoria integrada y acelerar todo el funcionamiento incrementando las frecuencias de reloj, gracias a la mayor escala de integración en la fabricación de circuitos integrados. Un exponente clásico de la potencia de esta quinta generación es la nVidia GTX 285. Esta dispone de 240 procesadores de *shaders* operando a 1.5 Ghz y una memoria GDDR3 de 1 GB a la que se accede a través de un bus de 512 bits. En la actualidad, el hardware de consumo más potente cuenta con varios miles de núcleos y 4 o más GB de memoria GDDR5.

4 Evolución del *pipeline* gráfico

A la vista de la historia esbozada en la sección previa, se deduce que las GPU, y más concretamente el cauce interno o *pipeline* [15] por el que va transcurriendo la información gráfica hasta terminar su procesamiento, ha ido cambiando de una manera paulatina pero constante.

4.1. Pipeline gráfico: etapas configurables

Partiendo de los datos sobre geometría de la escena que facilita la aplicación, para obtener una imagen es necesario efectuar una serie de transformaciones a los vértices, aplicar parámetros de iluminación, reconstruir las primitivas en 2D, rasterizar esas primitivas y asociar información sobre texturas, llevar a cabo distintas operaciones sobre estos fragmentos para, finalmente, obtener los píxeles que se almacenarán en el buffer y que formarán la imagen a enviar a la pantalla.

Antes de que apareciesen las primeras GPU los adaptadores de vídeo, tal y como se apuntaba anteriormente, se limitaban a convertir la información digital de los píxeles para enviarla a la pantalla, actuando en este sentido como un *frame buffer*. El esquema de la Figura 7 muestra el cauce gráfico con este tipo de hardware. Como puede apreciarse, todo el trabajo recaía en la CPU.

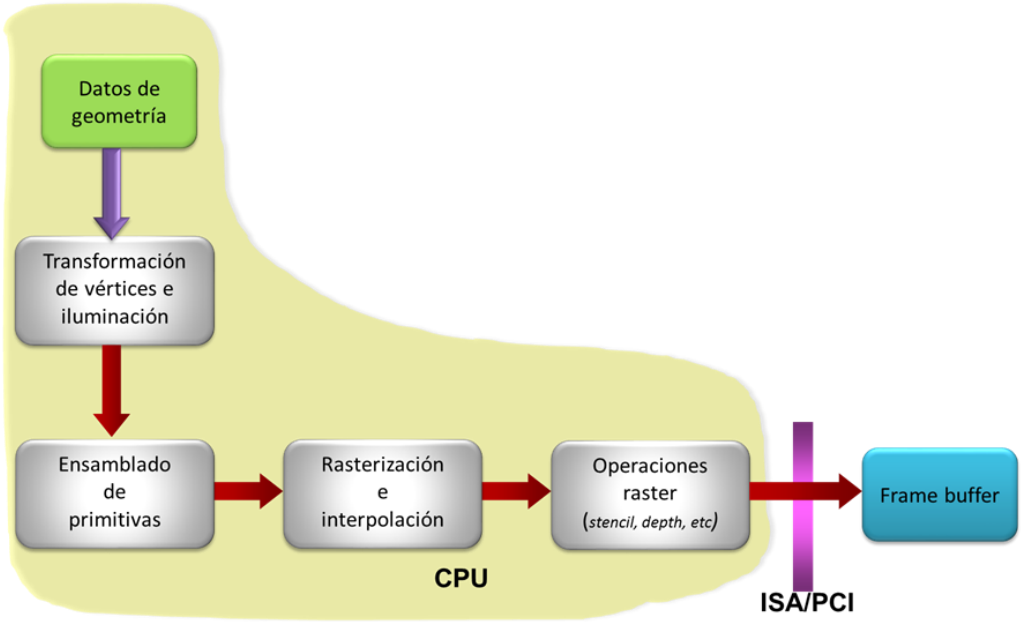

Figura 7. Cauce gráfico de los adaptadores de vídeo primitivos en ordenadores personales.

La primera generación de GPU eximió a la CPU de las operaciones relacionadas con el *rastering* y la aplicación de texturas. El cauce gráfico correspondería al representado en la Figura 8. Los bloques de la GPU encargados de la rasterización, interpolación y ejecución de operaciones *raster* tienen codificada una función fija parametrizable, de forma que la CPU configura hasta cierto punto esas etapas del *pipeline*.

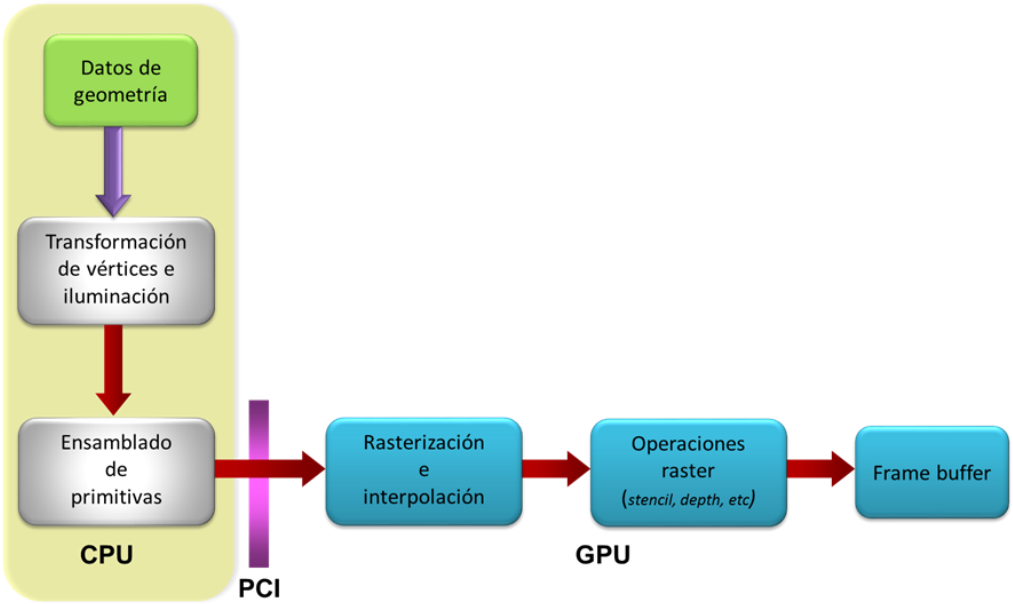

Figura 8. Cauce gráfico típico en la primera generación de GPU.

Es con la segunda generación de GPU cuando se pasa a tener un cauce que, como puede verse en el diagrama de la Figura 9, se ejecuta al completo en el adaptador de vídeo. La aplicación se limita a facilitar los datos de geometría. Cada una de las etapas cuenta con una serie de parámetros configurables, tal y como se indicó anteriormente.

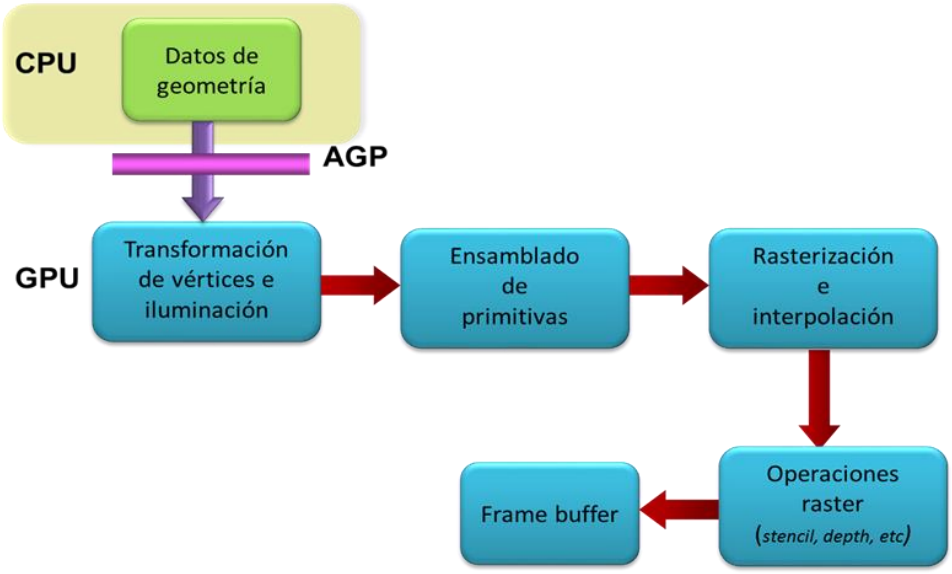

Figura 9. Estructura del cauce gráfico correspondiente a la segunda generación.

4.2. Pipeline gráfico: etapas programables

La tercera generación de GPU introduce en el cauce gráfico el primer elemento programable: los *vertex shaders*. Estos pequeños programas, que se ejecutan en el interior de la GPU y no en la CPU, pueden llevar a cabo operaciones sobre los vértices que no puedan efectuarse en la antigua etapa configurable. Un ejemplo sería la utilización de un algoritmo de iluminación que no esté implementado directamente en el hardware.

Como se aprecia en la Figura 10, la etapa del cauce en la que se ejecutan los *vertex shaders* está en una posición intermedia entre la etapa configurable de transformación y la etapa de ensamblado de primitivas. Al activar un *vertex shader* se está anulando por completo la etapa fija del *pipeline*, por lo que es necesario escribir código para sustituir todas las funciones que dicha etapa efectuaba: transformación de vértices y normales, normalización de normales, generación y transformación de coordenadas de texturas, cálculos de iluminación, material y color para cada vértice.

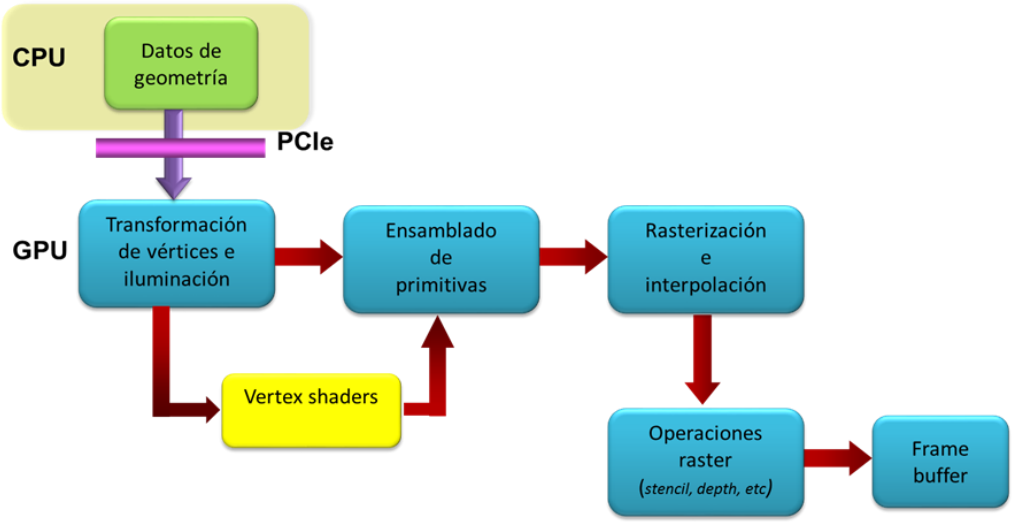

Figura 10. En la tercera generación apareció la primera etapa programable.

Aunque técnicamente el número de vértices que serán procesados en paralelo por un *vertex shader* dependerá del número de unidades de cálculo con que cuente la GPU, desde una perspectiva puramente lógica al implementar un programa de este tipo debe asumirse que va a ejecutarse de manera simultánea sobre todos los vértices de la escena. Esto implica que los cálculos de un vértice no pueden, en ningún caso, afectar a otros vértices. La ejecución es masivamente paralela en el hardware actual.

La relación de entrada-salida en un *vertex shader* es siempre de uno a uno, es decir, son programas que toman como entrada un vértice y generarán como resultado otro, nunca se elimina ni se genera nueva geometría.

La introducción de los *fragment shaders*, ya en la cuarta generación de GPU, extendió una vez más el cauce gráfico como queda reflejado en la Figura 11. Existe una nueva etapa programable, en este caso interpuesta entre las etapas fijas de rasterización

e interpolación y la encargada de ejecutar operaciones sobre el *raster* justo antes de llegar al *frame buffer*.

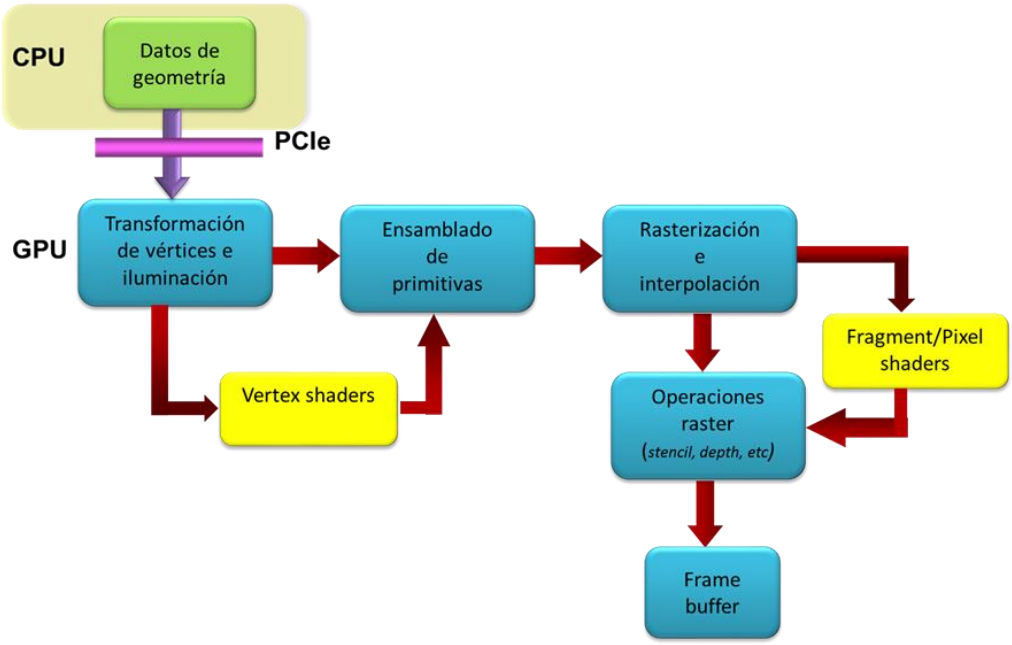

Figura 11. Las etapas programables de la cuarta generación de GPU.

Al igual que ocurre con los *vertex shaders*, al programar un *fragment shader* hay que asumir que este procesará en paralelo todos los pre-píxeles de la imagen, por lo que no pueden producirse efectos colaterales entre fragmentos. Es decir, el resultado de aplicar un cierto tratamiento a un fragmento no puede usarse como base para modificar otro fragmento distinto.

El *fragment shader* recibe los fragmentos una vez que se han llevado a cabo tanto la rasterización como la interpolación de los vértices, siendo tarea suya la aplicación de texturas y generación de otros efectos visuales, decidiendo qué color se asignará finalmente a cada uno de los píxeles que se enviarán al *frame buffer*.

En el *pipeline* de la Figura 11, que corresponde al de la cuarta generación, hay dos etapas programables pero también muchas limitaciones. Estas afectan tanto a la extensión máxima de los *shaders* como a las operaciones que pueden efectuar, sin estructuras de control ni acceso a la memoria del sistema de vídeo. No obstante, de forma global este cauce representa un avance muy significativo respecto al de las GPU de generaciones previas.

Dichas limitaciones desaparecen, en gran parte, en el cauce gráfico de las últimas generaciones de GPU, hasta llegar a las actuales, gracias a la aparición de una arquitectura unificada de núcleos de procesamiento. Dicha arquitectura, además, ha ido acompañada de nuevas especificaciones del modelo de programación de *shaders*, en las que se abren las puertas a la creación de programas más grandes y complejos, con capacidad de compartir información a través de la memoria integrada en la GPU.

En la Figura 12 se ha representado este nuevo cauce, en el que aparecen por primera vez los *geometry shaders*. Se trata de programas que interactúan con los *vertex shaders* y tienen la capacidad para generar nueva geometría a partir de los vértices originales.

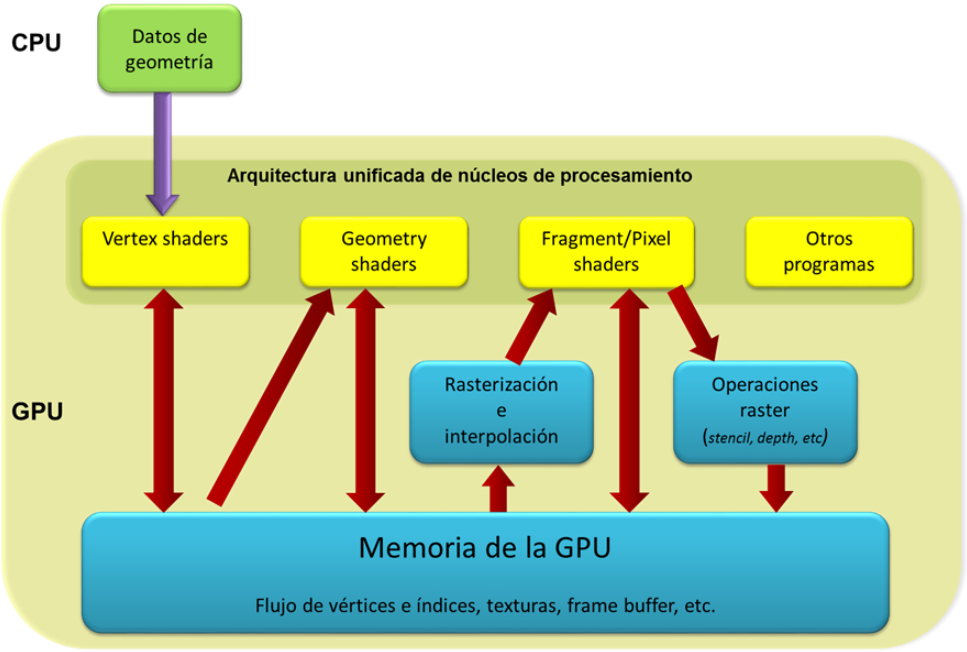

Figura 12. El *pipeline* de las actuales GPU es mucho más flexible.

A pesar de que el rendimiento gráfico se ha incrementado de manera constante, en parte debido a la evolución del *pipeline* de las GPU, para el programador esta nueva arquitectura representa casi una vuelta al pasado, cuando tenía que programar por sí mismo los algoritmos de transformación de vértices, sombreado, aplicación de texturas, etc. La diferencia estriba en que ahora esos algoritmos se ejecutan en la GPU y no en la CPU, aprovechando el gran paralelismo que otorga la disponibilidad de cientos de núcleos de procesamiento.

Obviamente el uso de *shaders* en el desarrollo de una aplicación gráfica es totalmente opcional. El programador puede seguir concentrándose únicamente en la generación de la geometría de la escena y dejar el resto del trabajo en las etapas configurables del *pipeline* clásico.

5 Conclusiones

El uso de terminales de vídeo para facilitar la comunicación interactiva con los sistemas de cómputo surgió hace medio siglo, siendo el medio preferente desde la aparición del ordenador personal, cuya expansión se inició a principios de la década de los ochenta. Desde entonces, el hardware a cargo de producir la imagen que los usuarios ven en sus

pantallas, tanto texto como gráficos, ha ido evolucionando y haciéndose cada vez más sofisticado. La consecuencia más destacable de este progreso, aparte de la patente mejora en la calidad de los gráficos y la velocidad con que pueden ser generados, es el hecho de que se descarga a la CPU de una tarea que consumía mucho tiempo de procesamiento. De hecho, en muchos microordenadores la generación de gráficos detenía por completo cualquier otra tarea, absorbiendo por completo al microprocesador.

En el presente artículo se ha hecho un recorrido por algunos de los hitos más destacables de la evolución del hardware gráfico, desde sus inicios, cuando ni siquiera incorporan memoria propia, hasta los adaptadores de vídeo modernos que no ofrece únicamente funciones gráficas, sino también posibilidades de cómputo de altas prestaciones mediante técnicas GPGPU. Incluso el hardware de vídeo integrado en los microprocesadores de última generación, como la familia Intel Core iX, ofrece un nivel de prestaciones que era prácticamente impensable hace apenas dos décadas.

Referencias

1. Plugge, W. R., & Perry, M. N. (1961, May). American Airlines' Sabre electronic reservations system. In Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference (pp. 593-602). ACM.
2. R. M., Fano and P. J. Corbato, “Time-Sharing on Computers” in Scientific American, Information (San Francisco, 1966), 76-95.
3. Walther, G. H. (1974, May). On-line user-computer interface: The effects of interface flexibility, terminal type, and experience on performance. In Proceedings of the May 6-10, 1974, national computer conference and exposition (pp. 379-384). ACM.
4. Sideris, G. (1973). Intel 1103-MOS memory that defied cores. Electronics, 46(9), 108-113.
5. Charte, F. (2011). El pasado de la computación personal: historia de la microinformática. ISBN: 978-84-8439-575-1.
6. Chen, J. Y. (2009, December). GPU technology trends and future requirements. In Electron Devices Meeting (IEDM), 2009 IEEE International (pp. 1-6). IEEE.
7. Guenter, B., Knoblock, T. B., & Ruf, E. (1995, September). Specializing shaders. In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (pp. 343-350). ACM.
8. Wilton, R. (1987). The programmer's guide to PC and PS/2 video systems: maximum performance from the EGA, CGA, HGC, and other graphics adapters. Microsoft Press.
9. Foley, J. D., & Van Dam, A. (1982). Fundamentals of interactive computer graphics (Vol. 2). Reading, MA: Addison-Wesley.
10. Eccles, A. The Diamond Monster 3Dfx Voodoo 1, Gamespy Hall of Fame, 2000.
11. Pipeline stages in Direct3D. https://msdn.microsoft.com/en-us/library/bb205123.aspx.
12. Lindholm, E., Nickolls, J., Oberman, S., & Montrym, J. (2008). NVIDIA Tesla: A unified graphics and computing architecture. IEEE micro, 28(2).
13. Kazakov, M. (2007, October). Catmull-Clark subdivision for geometry shaders. In Proceedings of the 5th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa (pp. 77-84). ACM.
14. Charte, F., Rivera, A.J., Pulgar, F. J., del Jesus, M.J. Explotación de la potencia de procesamiento mediante paralelismo: un recorrido histórico hasta la GPGPU. Enseñanza y Aprendizaje de Ingeniería de Computadores, 6: 19-33 (2016). http://hdl.handle.net/10481/41910.
15. Blinn, J. (1996). Jim Blinn's corner: a trip down the graphics pipeline. Morgan Kaufmann.