Graphics Cards: Recent and Future Trends

João Manuel Andrade

Dep. Informática, Universidade do Minho
4700 – 057 Braga, Portugal
joao.andrade@portugalmail.pt

Abstract: In the late few years, with virtual reality, multimedia and especially interactive entertainment, e.g. games, the need for massive, on time, 3D graphics had a tremendous increase. The roles of graphic cards became much more important and evolved into highly efficient processing engines that can now be viewed as highly specialized co-processors whit its own big processing and data feeding challenges. The balance between what is done by the processor or the graphic card, the use of "brute force" versus more efficient geometrical algorithms, the huge impact of memory bandwidth and the overall platform integration, all this in order to deliver the best frame rate with optimal quality, are the issues in today's and near future graphic processing. This places an issue also in graphical benchmarking, as the evaluation of graphic cards - both the image quality and their performance - becomes increasingly relevant to find where the real trends are, and to distinguish marketing from real cutting edge solutions.

1 Introduction

Video or graphics circuits, usually fitted to a card but sometimes found on the motherboard itself, are responsible for creating the picture displayed on the monitor. The advent of graphical operating systems and the 3D gamming world dramatically increased the amount of data to be displayed to levels where it was impractical for it to be handled by the main processor. The solution was to off-load the handling of all screen activity to a more intelligent generation of graphics card. They evolved into a highly efficient processing engine, which can really be viewed as a highly specialised co-processor. By the late 1990s the rate of development in the graphics chip arena had reached high levels [1].

Today the complexity of a graphics processor is outstanding: the recent nVidia V20 chip has 57 million transistors – more than twice the amount of the GeForce 2 and 20% more than Intel's Pentium 4. The nVidia Xbox GPU is capable of processing more than 1 trillion operations per second.

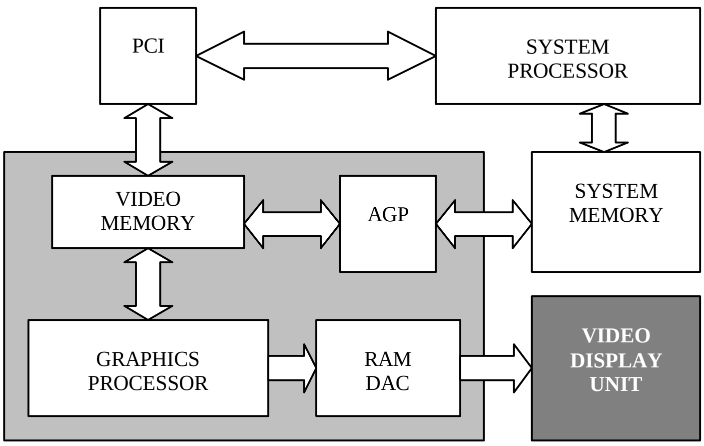

Fig. 1 Simple Graphics System Schema

This communication aims to present an overview of the main issues that affect graphical performances starting with a glance of a graphics card (Fig. 1). Then it briefly refers some issues that impact graphical performance: graphic libraries, drivers and benchmarking. Follows is a brief presentation of today mainstream chip makers and their top products.

The communications goes on presenting recent technological trends in graphic chips, with a closer look on effects programmability, and concludes by drawing some considerations about the future of graphical processing, evaluating integration versus specialization.

1.1. Graphics Processor Unit - From CPU to GPU

In the early VGA systems the CPU had a heavy workload processing the graphics data, and the quantity of data transferred across the bus to the graphics card placed excessive burdens on the system.The problem has been solved by the introduction of a dedicated graphics processing unit. Instead of sending a raw screen image across to the frame buffer, the CPU sends a smaller set of drawing instructions, which are interpreted by the graphics card proprietary driver and executed by the GPU - the card on-board processor.

Operations including bitmap transfers and painting, window resizing and repositioning, line drawing, font scaling and polygon drawing are handled by the GPU, which is designed to handle these tasks in hardware at far greater speeds. The graphics processor then writes the frame data to the frame buffer. As there is less data to transfer, there is less congestion on the system bus, and the CPU workload is greatly reduced.

The CPU, together wit the motherboard, chipset, memory and the PCI or AGP-slot compose the system responsible to provide the 3D-scene with all its players, objects, light sources for each frame and as well as any special kind of motion or artificial intelligence (in gamming). The geometry calculations, today called "transform and lighting" (T&L), have to be done either entirely or in parts by this system as well. The faster the system is, the more frame data it can send to the 3D card. If it is not fast enough it is stalling the 3D card and thus lowering the frame rate. With current 3D accelerators spewing out over a 100 million pixels per second, this is beyond the abilities of even the fastest CPUs. The 3D card literally has to wait for the CPU to finish its calculations [2].

There are two different ways of getting over this problem, and they involve where T&L is done. The 3D-hardware manufacturers advocate the use of a dedicated geometry processor. Such graphic processors would take over the geometry calculations from the main CPU. On the other side of the debate, this is the least acceptable solution for processor manufacturers - because once geometry processors become standard on graphics boards, it could only take a mediocre processor to perform other functions such as running the operating system and monitoring devices – and the technological lead could pass to the graphic chips manufacturers. Their response has been to boost the 3D performance of their CPUs by the provision of specialised instruction sets - Katmai New Instructions[1] (KNI) in the case of Intel and 3DNow! in the case of AMD [3].

Graphics operations are prime candidates to be implemented in hardware [4]. These are highly repetitive - with the same set of instructions performed millions of times per second. A dedicated engine can be optimised for the necessary mathematical functions, making it fairly simple to create an efficient, purpose-focused silicon design. This results of the observation that they (as well multimedia operations in general) take place as:

- Small, highly repetitive loops;
- Frequent multiplies and accumulates;
- Compute-intensive algorithms;
- Highly parallel operations.

[1] Katmai New Instructions: the 70 new Single Instruction Multiple Data (SIMD) instructions supported by the Pentium III - formerly codenamed Katmai - which came to market in the spring of 1999 designed to optimise the performance of multimedia and graphics applications.

1.2. Memory

The memory that holds the video image is also referred to as the frame buffer and is usually implemented on the graphics card itself. Display memory that temporarily stores (buffers) a full frame of picture data at one time. Frame buffers are composed of arrays of bit values that correspond to the display's pixels. The number of bits per pixel in the frame buffer determines the complexity of images that can be displayed. Early systems implemented video memory in standard DRAM. However, this required continual refreshing of the data to prevent it from being lost and could not be modified during the refresh process. The consequence, particularly at the very fast clock speeds demanded by the modernization of graphics cards, is that performance is badly degraded.

An advantage of implementing video memory on the graphics board itself is that it can be customised for its specific task and, indeed, this has resulted in a proliferation of new memory technologies, such as VRAM, EDO DRAM, SDRAM, SGRAM, Rambus...

Some designs integrate the graphics circuitry into the motherboard itself and use a portion of the system's RAM for the frame buffer. This is called Unified Memory Architecture (UMA). Since such implementations cannot take advantage of specialised video memory technologies they will, in principle, always result in inferior graphics performance [1].

1.3. Buses

Buses are the interface between the video processor and the system processor. The two main types used for graphics are PCI bus and Accelerated Graphics Port (AGP). Today almost all graphics cards use AGP.

Accelerated Graphics Port (AGP) is specifically intended for high-speed interfaces between video card and processor. It is a 32 bits point-to-point connection between the video card and the processor and is actually a port and not a bus. Initial AGP run at 66 MHz, but it has the capability of running up to four times the PCI bus speed with AGP texturing. In AGP texturing, the first called 2X because it used the so-called double-clocking technology to achieve twice the baseline bandwidth. By sending data on both the rising and falling edges of a 133 MHz clock this mode increases the bandwidth to 533 MB/sec. The 4X mode the transfer is twice as much as in 2X (1066 MB/sec. using a 266MHz clock.) [5].

2 Issues in Graphics Performance

2.1. Graphics Libraries

The evolution of graphics libraries has had a significant impact on the evolution of video cards, and the reverse also applies. When there is a gamming or multimedia need, generally it is first software implemented through emulation. Later, graphics manufacturer, in order to improve performance, implement it in hardware, and then those characteristics can be accessed in hardware through abstraction. A recent example of this are Vertex and Pixel Shaders, included in DirectX 8.0 [8][22] in 2000, and now available in the most recent chipsets as nVidia GeForce3 [6] or ATI Radeon 8500 [23].

The main graphics libraries present today are OpenGL and DirectX (DirectX is not only a graphics API, but a complete gaming and multimedia API) [7]. Each has a different role. OpenGL is a cross-platform standard for 3D rendering and 3D hardware acceleration. The software runtime library ships with all Windows, MacOS, Linux and Unix systems. It was behind the success of the first 3D computer games, such as Wolfstein and Quake.

DirectX emerged in sequence of the Talisman [28] project, as an attempt of Microsoft to create a standard for software developers to use when programming multimedia applications. Being a strong bet, it emerged to attempt to control an area till then dominated by Glide and OpenGL.

The DirectX philosophy is forward thinking realizing that the tech today will be implemented in the hardware of tomorrow. So if the hardware of the system supports the particular software function (e.g. fast rendering of objects, lighting effects, meshing, etc.) it passes the process down to the hardware, otherwise to the HEL module to simulate the hardware process. Despite Microsoft claims that DirectX is going to be the end of OpenGL there are going to be some major hurdles for Microsoft to overcome, essentially to do with the facts that: OpenGL is multi platform. DirectX is not, and it is extremely unlikely that Microsoft would port it to MacOS or Linux which are widely used in the Computer Graphics industry; Linux render farms are becoming more and more present in the 3D industry.

2.2. Benchmarking

There are two main types of benchmarking a video card [17]:

- Real world benchmarks: these are essentially routine applications, which have inbuilt test features, such as Quake 3 Arena, which measures the card frame rate.
- Synthetic benchmarks, such as 3dMark, are used to test specific features of a card such as bump mapping, anti aliasing, or the ability to deal with certain situations such as the complex lighting of a scene.

Neither real world benchmarks nor synthetic ones can give an outright report on the quality of a card, since they dependent on other system hardware [2] - CPU, Memory, Buses, or the presence of other devices - drives, sound cards - that may impede performance, by competing for the same resources than the graphics card.

Another issue to attend when evaluating graphic cards, particularly using real world benchmarks, e.g. games, where frame rate is an evaluating factor, is that there is a minimal of requirements a system must have to meet the demand of a modern graphics card. If the system is slow, or if the graphical engine of a particular game is slow (e.g. Unreal Tournment [2]) different graphic cards will perform the same, because all will be waiting on the CPU to finish the transform stage to feed them to the rendering stage [2].

At the time of this writing, the flagship is ATI Radeon 8500, closely followed by nVidia, with GeForce Ti500 at the head, and then by the other Geforce [14].

2.3. The Right Drivers

Drivers can play an important performance, notably visible in benchmarking. In some cases the drivers are written to give artificial results for benchmarking, but in many other cases a new driver will actually enhance performance greatly [15]. Also having the right drivers on hardware devices that directly interact with the sound card is of the utmost importance for overall system performance. It is easy to see that, for e.g., an improper AGP driver may greatly affect the graphics performance, leaving the blame to the card [15].

3 Main Chipset Manufacturers at a Glance

Current top chipset manufacturers are nVidia and ATI, with some interference of Kyro and Matrox. 3dFX is not in the market anymore. NVidia, unlike other companies mentioned, does not really produce graphics cards themselves. Instead they produce chipsets which 3rd party companies use. It is the current market leader in the gamming area [9], and it is expanding its interests [10][11][12]. Their most recent releases are the GeForce3 family graphical chipset, the nForce chipset and the involvement in Microsoft's XBox project.

ATi is nVidias's main competitor at this point in time, when it comes to video cards for the mainstream market. Built on separate technologies, ATi continues to meet and sometime beat nVidia at providing revolutionary 3D hardware. They were the first company to utilize DDR RAM on their cards and have always had an excellent reputation for video/DVD playback. Their most recent chipset is the Radeon 8500, released to beat nVidia GeForce3 [16].

Matrox and STMicroelectronic (Kyro manufacturers) are also holding firm, thanks in part to a dedicated band of users who would not dream of buying from anywhere else. Matrox new G550 card has made some promising steps forward, but has also disappointed [9] and the company is gambling that its Headcasting technology [18][19] will be a winner.

The low-cost Kyro II chipset [20][21], however, has at first failed to impress [9], although at the time of its release it appeared different [30]. Minor alterations were made to the original innovative Kyro chip [20] and the results were far from convincing. It is, although, an entry-level alternative that can be popular with system integrators.

4 Recent Trends

4.1. Effects Programmability – Vertex Shaders and Pixel Shaders

The new architectures introduced in DirectX 8 have substantially changed the way that programmers should work with the graphics hardware available to them. The main change was the introduction of effects programmability, trough Vertex and Pixel Shaders, allowing full freedom for the programmer to build a all new combination of personal effects, but bringing also all the new challenges of getting acquainted an taking the most of a new approach (very different of the one of previous DirectX, with fixed-function pipelines).

Pixel and Vertex Shaders emerged to full fill the need of a technology that combined the speed and optimizations of a dedicated graphics processor with the flexibility and programmability of a CPU, allowing a virtually infinite range of visual effects at interactive frame rates [23].

The two top chipset of the two main manufacturers now implement in hardware this new technology. The fist was nVidia's GeForce3 [6], and soon followed ATI Radeon 8500 [23], this one improving the concept initially drawn in DirectX 8.0 [22][23].

A graphics engine usually contains three main sequential processing blocks: primitive processing, vertex processing and a pixel rasterizer.

Primitive processing is a simple way to describe what the majority of a 3D game engine does. it takes program or game-specific data and data structures and produces vertices for the vertex processor to process. Currently, the game engine carries out the majority of the primitive processing, but in the future it can be expect at least some primitive processing to be carried into the API domain.

Vertex processing can be further subdivided into a few major blocks of processing: transformation, lighting, and to a limited extent, texture coordinate transformation.

The final part of the pipeline takes data from the vertex processing step and puts it into the frame buffer as pixel values. Before reaching the pixel rasterizer, the triangles are usually converted into scan lines, then the pixels in the scan line are passed through the rasterizer, one at a time. Actually, modern hardware often has a number of pixel rasterization blocks that will write more than one pixel at a time.

The pixel rasterizer will take all of the information passed through from the vertex processor and compute a final pixel colour, based on these values. A basic example of its usage might be to take the diffuse color and multiply it with the texture color (using the texture coordinates to retrieve a colour from the current texture).

In DirectX 8, the vertex pipeline has been made entirely programmable. Arbitrary vertex data is processed in an arbitrary way using the vertex shader, which then places the output values into the write-only output registers.

The programmable pixel pipeline introduces a brand new pixel rasterization concept along the lines of vertex shaders. There are a number of reasons that pixel shaders should be polemic: the first related to the overcomplicated compatibility checking necessary for the fixed-function pipeline. In pixel shaders, DirectX has introduced an all-or-nothing style of compatibility [22].

At first, there was not much success in the use of Vertex Shaders and Pixel Shaders, mainly due to the need to verify which steps are hardware implemented [22], compatibility reasons [22] and the limited number of instructions (128) and constants (96) in the current version of vertex shader [22]. If exceeded the vertex shader will just fail to run on the hardware. This drew nVidia to look for promotion of its use.

Looking at the limitations of Vertex an Pixel Shaders, ATI saw a opportunity of improvement, and unified and renamed the concept in its SmartShader technology [23] .It was developed, according with ATI [23], towards maximizing efficiency and minimizing common performance bottlenecks, especially memory bandwidth. The key improvements made were, according with ATI [23]: support for up to six textures in a single rendering pass, allowing more complex effects to be achieved without the heavy memory bandwidth requirements and severe performance impact of multi-pass rendering; This brought a new design of the 3D graphics pipeline, as shown in Fig. 4.

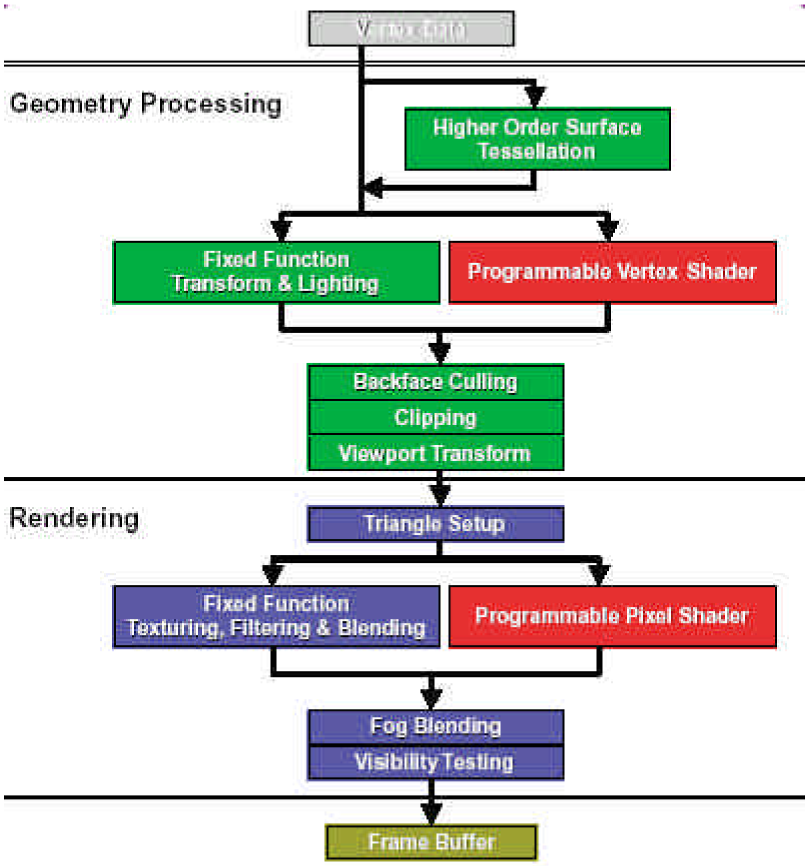

Fig. 4 The ATI 3D Graphics Pipeline (Courtesy of [23])

Vertex shaders are small programs or sets of instructions that are performed on vertex data as it passes through the geometry processing pipeline. Each vertex can consist of up to

16 distinct pieces of data, which are read by the vertex shader as individual streams.. A vertex shader program can have a maximum length of 128 instructions, and make use of up to 96 constant values and 12 temporary data registers (Fig 5). The actual instructions are very close to those found in assembly and allow for easy manipulation of vertex data [23].

Pixel shaders are small programs that are executed on individual pixels as they pass through the rendering pipeline. Up to six different textures can be sampled and manipulated in a single rendering pass to determine the colour of a pixel. Textures can be one-dimensional, two-dimensional, or three-dimensional arrays of values stored in memory. Each value in a texture is called a texel, and although they are most commonly used to store colour values, they can contain any kind of data desired including normal vectors (for bump maps), shadow values, or look-up tables. These specifications provide ample ability to perform a huge range of transformations on incoming vertex data.

Pixel shader programs can be divided into two parts. The first called the address shader, which performs up to eight mathematical operations (addition, multiplication, dot product, etc.) on texture co-ordinates or addresses. Up to six textures can be sampled either before or after the address shader is executed. The ability to sample a texture value, modify that value in the address shader, and use the modified value as an address to sample a different texture, allow pixel shaders to perform what are known as dependent texture reads. This technique is necessary to accomplish environment mapped bump mapping, anisotropic lighting, and many other important effects. The second part of a pixel shader is known as the colour shader, which consists of up to eight instructions that blend and modify the values (usually colours) of previously sampled textures to give the final pixel colour.

Are Pixel and Vertex Shaders here to stay? Existing programmable shader implementations place significant limits on what developers can actually do. These constraints can be narrowed down into the following categories: number of input variables; maximum program length; instruction set and performance.

Also a lot of the work that used to be done by the driver has been moved into the domain of the games programmer [22]. It is expected that DirectX 9 [22][8] will introduce flow control into the vertex shader architecture.

4.2. Moving on Integrated System Chipsets

Another market trend is the move of the main graphical chipset makers towards producing overall platforms. NVidia recently introduced its new base integrated system chipset, nForce [25][26]. According to nVidia, nForce ia a revolutionary architecture with a distributed platform processing design that frees up the CPU for other tasks; it includes system, memory, and networking technologies for the most efficient processing and performance, and integrating 3D graphics and 3D audio. It is not in the scope of this paper to discuss the conflicts between the DASP[2] with the pre-processing techniques of the Palomino core of the AMD Athlon, the advantages of the TwinBank UMA[3], the role of the IGP[4] in replacing Northbridge in its functions [25], or the particular advantages of the MCP[5]. What it is significant is that there was an attempt to technically control the conception of what is significant on a modern standard Personal Computer as all, a bit like when building a game console[6]. For instance in Virtual Reality sound and 3D walk together.

---

[2] nVidia Dynamic Adaptive Speculative Pre-processor.

[3] Unified Memory Architecture.

[4] Integrated Graphics Processor

[5] Media and Communications Processor.

[6] Interestingly, nVidia creates important parts of the Xbox console, its GPU and MPC.

In the very near future it is predictable broadband communications to join. Developing them together is a way to ensure their compatibility and that they will not compete irrationally for the same resources, for the sake of the image of their manufacturers. So it is believable that this may point a future trend in the hardware arena. One clear and immediate advantage of this is the unified drivers conception and release, easing a lot the life of users [25].

ATI is also on the move for similar projects, although not as ambitious as nVidia risky bet. It is expected to launch its Pentium 4 based integrated chipset in the first quarter of the current year [29].

5 Conclusion – Future Trends

5.1. Specialization versus Integration

Two main roads can be taken by graphics processing: the progressive specialization into a complete separate computation unit - with its own memory, CPU and communication processes - or the integration in a multi-purpose central processing unit. A third road can be a hybrid system in between, with shared memory, for example.

If in one hand market and the fast cycle of1 development of graphic chips favours the first, it can be argued that, in longer terms this situation may not sustain. This for two main reasons: first, when performing tasks where intense graphical computation is not needed, there is a huge waist of computational power in the GPU; second, and in relation to the first, the need of similar kinds of computation power that GPU is theoretically able to perform [4] is increasing with Multimedia and Communications contents.

To understand the complexity of potential future multimedia scenarios, let us consider one in which the user of a portable computer, sitting at a beach, is able to select one of hundreds of satellite channels for a live broadcast of a soccer game, and simultaneously hold a video conference with its remotely located friend to discuss the intricacies of the same game, without losing the ability to respond to any incoming fax or phone call [4]. At the same time it might want to be shopping in a VR mall. The same computer may also be used later to play a VR video game. Such a scenario would break down into the real-time encoding and decoding of multiple video and audio streams, including encryption/decryption and error correction. The video-stream processing may also imply, in future terms, real time 3D transformations. Audio-stream processing could include real-time spontaneous speech recognition to enable interesting searches (indexing) of the content. It can be seen several distinguishing characteristics of these multimedia-centric applications that can have profound implications for future processor design (adapted from [4]).

In many of these multimedia-centric applications, huge processing is needed. An encrypted/decrypted real-time video conversation will only make sense if it remains real-time. A closer look let us see that voice processing, video processing and encryption processing are of the same kind [4]. They often consist of small loops or kernels that dominate overall processing time. Within these loops and kernels, instruction references tend to be concentrated and hence exhibit good spatial and temporal locality. . Relative to non multimedia applications, this yields a much higher degree of correlation between overall application speedup and loop/kernel speedup [4]. The same single instructions are applied consecutively to multiple data (SIMD). So it doesn't make much sense having a lot of separate computational power just for the sake of 3D processing. Also it can be argued where to put the border between what (T&L) is done by the CPU or by the GPU, and asked if there is a border that can be moved, why should it exists at all?

If the PC moves towards a VR multimedia unit, caching will have to change [31]. The PC was designed with just such spreadsheets, word processors and other static processing applications in mind. That implied a static data environment with multiple instructions applied to it. Multimedia, in the other hand, is SIMD. In this kind of environment what is needed are broad means of communication with only small caching on the end [31]. It is interesting to see that that is what is being done in the system architecture of gaming consoles, naming the PlaysStation2 [31].

Despite all of this, it is still unclear what of these two trends will prevail in the near future. Mainly because there is something called Market, and because the graphical processing of VR will increasingly mean a lot more than visual processing itself. To understand this, lets thing of a sniper shooting at a far target. In the perfect VR simulation the conditions of a real situation should all be reproduced, including the exact bullet path to the specified located target, in order to determine if there was a hit. The computation power behind this is still beyond actual computation models. Reality is continuous; today's computation is discrete. In the end, the ultimate representation of reality is reality itself. This takes the strain of processing a lot further from 3D graphics and into VR as a whole. The current path is the path of progressively feasible illusion. In the end it is conceivable that complete integration will win and that what we see today spread and attached to the motherboard will be concentrated in a single silicon chip. Than, maybe, we will instead be talking about its internal parts, and discussing generalization versus specialization in the inside of it.

To end, maybe an interesting comparison: in times, when we had an Intel x86 processor, we had to buy a x87 processor to accomplish, with increased performance, certain tasks. Later, with the evolution of silicon chips manufacturing, they both merged. Could this be what will happen with dedicated graphic chips on Personal Computers?

References

[1] Pc Technology Guide – Multimedia/Graphics Cards, http: www.pctechguide.com/05graphics.htm, (2001)

[2] Pabst ,Thomas: 3D Benchmarking – understanding Frame Rate Scores, Tom's Hardware Guide, http: www4.tomshardware.com/graphic/00q3/000704/index.html, (2000)

[3] AMD-K6 MMX Enhanced Processor Multimedia Technology TM Whitepaper, Advanced Micro Devices, Inc., http: www.amd.com, (2000)

[4] Dubey, Pradeep K, Diefendorff, Keith: How Multimedia Workloads Will Change Processor Design, IEEE, www.ieee.org, (1997)

[5] Pires, Hugo: From a 3D point of View, Proceedings of ICCA 2000, Universidade do Minho, Braga, (2000)

[6] NVidia Corporation (ed.): The Infinite Effects GPUs - G-Force3Ti Product Overview, www.nvidia.com, (2001)

[7] DirectX, Microsoft Corp.,http:msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000410, (2001)

[8] Rahmat, Omid: Meltdown 2001 - DX9, XP gaming, and more 3D graphics, Tom's Hardware Guide, http: www4.tomshardware.com/business/01q3/010719/index.html, (2001)

[9] Thomson, Iain: Visual Treats, 3D Graphic Cards | Peripherals, ZDNet UK, PC Magazine, November, http: www.zdnet.co.uk/pcmag/labs/2001/11/3dgfx-cards/01.html, (2001)

[10] NVidia Corporation (ed.): Quadro DCC, Powering the Next Generation of Digital Content Creation, Product Overview, www.nvidia.com, (2001)

[11] NVidia Corporation (ed.): nForce, Product Overview, www.nvidia.com, (2001)

[12] Microsoft Corp. (ed.): Microsoft's Official Xbox Site, http: www.xbox.com/ (2001)

[13] Purriños, Andrés et al: Test: 8 tarjetas gráficas, Computer Hoy, N.º 76, Hobby Press, S.A. Madrid, (2002)

[14] MadOnion Homepage, http:www.madonion.com

[15] The Power Behind The Card, 3D Graphic Cards | Peripherals, ZDNet UK, PC Magazine, www.zdnet.co.uk/pcmag/labs/2001/11/3dgfx-cards/02i.html, (2001)

[16] Thomson, Iain: ATI's new 3D chipsets, 3D Graphic Cards | Peripherals, ZDNet UK, PC Magazine, November, http: www.zdnet.co.uk/pcmag/labs/2001/11/3dgfx-cards/02e.html, (2001)

[17] Price, Andrew et al: Video Cards and Their Evolution Over the Last 5 Years, Students Presentation on Computer Graphics, Semester 2, Department of Computer Science and Software Engineering, Faculty of Engineering, University of Newcastle, (2001)

[18] Rahmat, Omid: Matrox Launches The G550, Tom's Hardware Guide, http: www4.tomshardware.com/business/01q3/010719/index.html, (2001)

[19] Matrox Home Page, http: www.matrox.com/mga/home.htm

[20] PowerVR (ed.): 3D Graphical Processing. White Paper, http: kyro.st.com; www.st.com, (2000)

[21] STMicroelectronics (ed.): Tile Based Rendering - the future of 3D graphics (2000) http: kyro.st.com; www.st.com

[22] Penfold, Don: Vertex Shaders and Pixel Shaders, Tom's Hardware Guide, http: www.tomshardware.com/graphic/02q1/020116/index.html, (2002)

[23] ATI Technologis Inc. (Ed.): SmartShaderTM Technology White Paper, Ontario, http: www.ati.com, (2001)

[24] Weinand, Lars: Aquanox - Bringing 3D To Its Knees, Tom's Hard News weekly email Newsletter Vol II/No. 3, (2002)

[25] Amdmb.com - AMD Hardware Reviews and Information (ed.): The 21st Century PC, nVidias nForce Chipset Preview, http: www.amdmb.com/article-display.php?ArticleID=119&PageID=1, (2001)

[26] nVidia nForce Homepage, http:http://www.nvidia.com/view.asp?PAGE=nforce, (2001)

[27] nVidia Xbox Hoomepage, http:http://www.nvidia.com/view.asp?PAGE=xbox, (2001)

[28] Vitaliano, Franco: Intel MMX vs. Micrososft Talisman, Abbot and Costello Do Multimedia (1997)

[29] New Chipsets Crowd the Market, Weekly Platform Trends, Hardware Central Reports http. www.hardwarecentral.com, (2001)

[30] Scheffel, Uwe: A Speedy Tiler: Kyro II on Hercules' 3D Prophet 4500, Tom's Hardware Guide, http: www4.tomshardware.com/graphic/01q2/010425/index.html(2001)

[31] Stokes, Jon: The Playstation2 vs. the PC: A system-level Comparison of Two 3D Platforms, Ars Techica, http: www.arstechnica.com/reviews/1q00/playstation2/ee-1.html