International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 4 Issue 1, December 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Graphics Processing Unit: An Introduction

Matthew N. O. Sadiku, Adedamola A. Omotoso, Sarhan M. Musa

Roy G. Perry College of Engineering, Prairie View A&M University, Prairie View, Texas

ABSTRACT

Graphics Processing Unit (GPU) is a processor (or electronic chip) for graphics. GPUs are massively parallel processors used widely used for 3D graphic and many non-graphic applications. As the demand for graphics applications increases, GPU has become indispensable. The use of GPUs has now matured to a point where there are countless industrial applications. This paper provides a brief introduction on GPUs, their properties, and their applications.

KEYWORDS: *graphics processing unit, GPU computing, visual processing unit, heterogeneous computer system*

How to cite this paper: Matthew N. O. Sadiku | Adedamola A. Omotoso | Sarhan M. Musa "Graphics Processing Unit: An Introduction" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-4 | Issue-1, December 2019, pp.702-704, URL: www.ijtsrd.com/papers/ijtsrd29647.pdf

IJTSRD29647

INRODUCTION

Recently, computer performance has increased tremendously due to the outstanding growth of the number of transistors. This growth has impacted the world of scientific computing with the arrival of graphics processing unit (GPU), which can perform graphical and non-graphical computations [1]. The introduction of GPU in recent years has opened a way to perform faster calculations than central processing unit (CPU).

GPU is sometimes called visual processing unit (VPU). GPU is an ubiquitous, electronic chip which is mounted on a video card in every PC, laptop, desktop computer, and workstation. It is a programmable logic chip specialized for display functions. It is designed to rapidly manipulate and alter memory. Architecturally, the CPU consists of only few cores with lots of cache memory that can handle a few software threads at a time. A GPU is composed of hundreds of cores that can handle thousands of threads simultaneously. A CPU consists of four to eight CPU cores, while the GPU consists of hundreds of cores. This massive parallel architecture gives the GPU its high compute performance. GPU may be regarded as a coprocessor to the CPU which has its own DRAM and runs many threads in parallel. The difference between CPU and GPU is shown in Figure 1 [2].

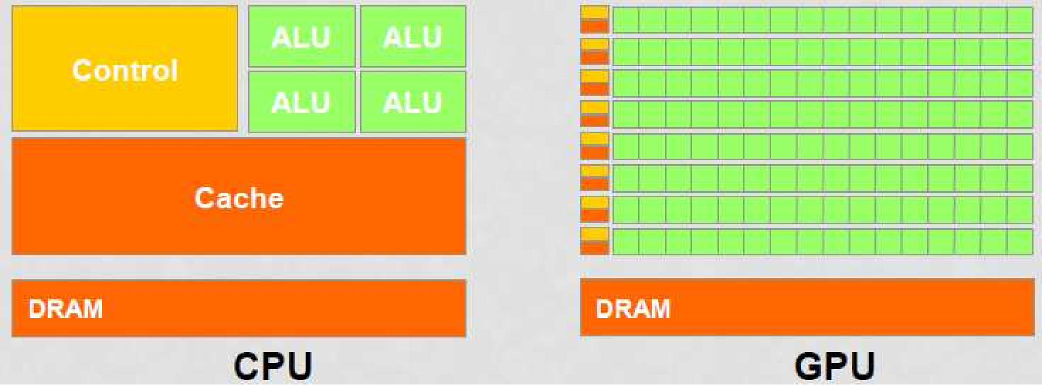

Figure 1 the difference between CPU and GPU [2].

The term GPU was popularized by NVIDIA Corporation in 1999 when the company introduced the first GPU. NVIDIA made GPU fully programmable for scientific applications and support higher-level languages such as FORTRAN, C and C++. NVIDIA's CUDA (Compute Unified Device Architecture) platform introduced in 2007 has become the dominant proprietary framework [3]. Besides NVIDIA, other GPU vendors include Intel, ATI, Sony, and IBM.

GPU BASICS

GPU is specialized for compute-intensive, highly data parallel computation, which what graphics rendering is all about. Although GPU can be used for 2D data, it is essential for rendering of 3D animations and video. GPU has a unique design of 'many-core' architecture, and each core is able to carry out thousands of threads simultaneously. The memory of GPU consists of a large number of cache blocks, and each block can be independently accessed [4].

Before GPU was invented, graphics on a personal computer were performed by a video graphics array (VGA) controller. GPUs offer parallel computing power that usually requires a computer or a supercomputer to accomplish.

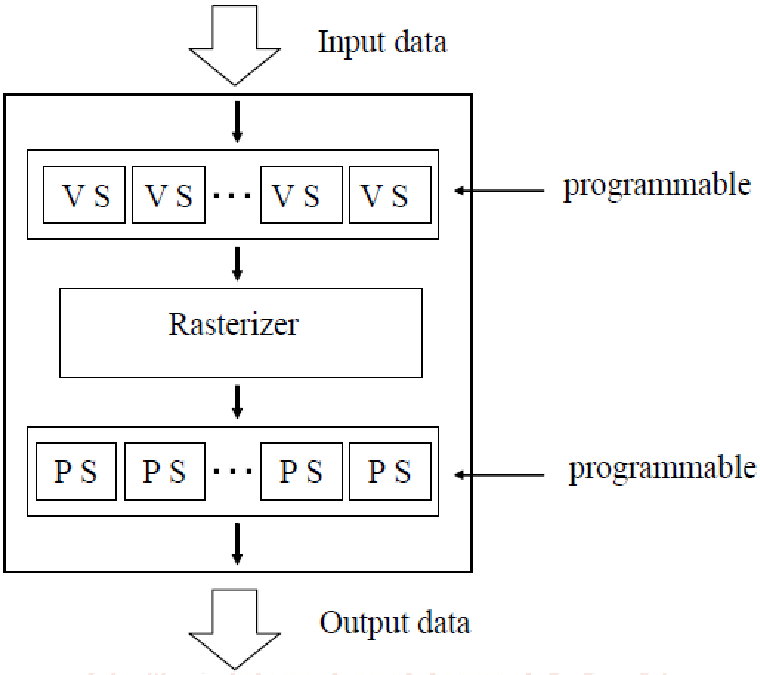

Figure 2 Block diagram of GPU: "VS" is a vertex shader and "PS" is a pixel shader [5].

As shown in Figure 2 [5], GPU has a vertex shader, rasterizer and pixel shader. The shader transforms geometry, a rasterizer rasterizes geometry, and a pixel shader draws pixels. Notice that both vertex shader and pixel shader are programmable. Basically, computation on a GPU is a three-step process: copy data to a GPU memory, operate GPU processing, and copy the results back from the GPU memory.

APPICATIONS

Graphics remains the leading application of GPUs since they were originally created for rendering graphics. Due to their cost and performance, they have become the new standard of image processing and for non-graphics applications. Some typical applications of GPU include GPU computing, scientific computation, and video decoding.

- **GPU computing:**

This is the application of a GPU to do general purpose scientific and engineering computing. In recent years, substantial efforts were made to adapt many algorithms for massively-parallel GPU-based systems since the GPU can perform many calculations simultaneously.

- **Scientific/Engineering Computation:**

Classic numerical methods, such as solution of linear equations and FFT, have been adapted for massively parallel GPU architectures. Computer scientists, along with researchers in fields such as medical imaging and electromagnetic have used GPUs to accelerate a range of scientific computations.

- **Video decoding:**

For a GPU implementation, a decoder uses innovations in the areas of inverse transforms, inverse quantization, loop filtering using waves, and performance-adaptive loop filtering. These techniques are directed at accelerating video encoding using a GPU as compared to encoding using just the CPU. The video frames may be collocated to enable parallel tasking of the GPU.

Other applications include general-purpose graphics processing units (GPGPU), mobile computers, membrane computing, scientific computing, electromagnetic modeling (using finite elements, finite difference, Monte Carlo, etc.), vectorization, game physics, computational biophysics, medical imaging, image processing and restoration, control systems (such as humanoid robots), de-noising, filtering, interpolation, gaming, and reconstruction,

BENEFTIS AND LIMITATIONS

GPU has the following benefits [6]:

- Performs same operations simultaneously on multiple pieces of data
- Organizes operations to be as independent as possible
- Arranges data in GPU memory to maximize rate of data access

The limitations of GPU include memory accesses, data movement between host and GPU, architectural dependencies, energy consumption and costs. Perhaps the most severe limitation is that each thread of execution can only write a single output value to a single memory location in a gathered fashion [7]. Most GPUs such as CUDA have no standard library, no parallel data structures, and no synchronization primitives. Although getting started with GPU programming can be simple, it can take months and years to fully master GPU hardware. Current GPU cache hierarchies are somehow inefficient in the face of streaming data.

CONCLUSION

GPUs are coprocessors that basically perform the rendering of 2D and 3D graphics. Today's GPU is not only a powerful graphics engine but also a highly parallel programmable processor. GPU has become an indispensable tool when it comes to large computing because it provides unprecedented computational power for scientific applications. Digital images and videos can be processed efficiently in GPU by exploiting its feature of parallel execution. More information about GPU can be found in [8].

REFERENCES

[1] A. C. Ahamed and F. Magoulès, "Conjugate gradient method with graphics processing unit acceleration: CUDA vs OpenCL," *Advances in Engineering Software*, vol. 111, 2017, pp. 32–42.

[2] A. Chauhan. "Graphics processing unit architectures," https://www.cs.indiana.edu/~achauhan/Teaching/B649/2011-Fall/StudentPresns/gpu-arch.pdf

[3] "Graphics processing unit," *Wikipedia,* the free encyclopedia https://en.wikipedia.org/wiki/Graphics_processing_unit

[4] H. Hsieh and C. Chu, "Particle swarm optimization (PSO)-based tool path planning for 5-axis flank milling accelerated by graphics processing unit (GPU)," *International Journal of Computer Integrated Manufacturing*, vol. 24, no. 7, 2011, pp. 676-687.

[5] N. Masuda et al., "Computer generated holography using a graphics processing unit," *Optics Express*, vol. 14, no. 2, Jan. 2006.

[6] P. Blakely, "Introduction to GPU hardware and to CUDA," http://people.ds.cam.ac.uk/pmb39/GPULectures/Lecture_1.pdf

[7] J. A. Anderson, C. D. Lorenz, and A. Travesset, "General purpose molecular dynamics simulations fully implemented on graphics processing units," *Journal of Computational Physics*, vol. 227, 2008, pp. 5342–5359.

[8] V. Kindratenko (ed.), *Numerical Computations with GPUs*. Spinger, 2014.