International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-3, Issue-1, January – 2015

GPU-Graphics Processing Unit

Ramani Shrusti K., Desai Vaishali J., Karia Kruti K.

**Department of Computer Science, Saurashtra University
Rajkot, Gujarat.**

Abstract— **In this paper we describe GPU and its computing. GPU (Graphics Processing Unit) is an extremely multi-threaded architecture and then is broadly used for graphical and now non-graphical computations. The main advantage of GPUs is their capability to perform significantly more floating point operations (FLOPs) per unit time than a CPU. GPU computing increases hardware capabilities and improves programmability. By giving a good price or performance benefit, core-GPU can be used as the best alternative and complementary solution to multi-core servers. In fact, to perform network coding simultaneously, multi core CPUs and many-core GPUs can be used. It is also used in media streaming servers where hundreds of peers are served concurrently. GPU computing is the use of a GPU (graphics processing unit) together with a CPU to accelerate general-purpose scientific and engineering applications. GPU was first manufactured by NVIDIA. CPUs have few cores which is used for serial processing and GPUs have thousands of smaller cores which are more efficient, designed for parallel processing. So, CPU + GPU is a powerful combination. Whenever the code is run on the machine, CPU runs serial portion and GPU runs parallel portion. GPU is used for general purpose applications like arithmetic and it is also used for gaming.**

Keywords— **CUDA, GPU, GPGPU, NVIDIA**

I. INTRODUCTION

Parallelism is the future of computing. Future microprocessor development efforts will continue to concentrate on adding cores rather than increasing single-thread performance. It can be possible through GPU.

Graphics Processing Unit (GPU) is a massively multi-threaded architecture and then is widely used for graphical and now non-graphical computations. Graphics Processing Units (GPUs), works with Central Processing Units (CPUs) in PCs, are special purpose processors designed to efficiently perform the calculations necessary to generate visual output from program data.

GPU has become into a powerful programmable processor, with both application programming interface (APIs) and hardware increasingly focusing on the programmable aspects of the GPU, so result is a processor with enormous arithmetic capability is GPU.

GPU is a processor generally used for display, video, 2D/3D graphics and visual computing. It is parallel, multithreaded multiprocessor used for visual computing. It also provide real-time visual interaction with computed objects via graphics images, and video.GPU serves as both a programmable graphics processor and a scalable parallel computing platform.

Graphics Processing Unit is also called Visual Processing Unit (VPU) is an electronic circuit designed to quickly operate and alter memory to increase speed the creation of images in frame buffer intended for output to display.

A. History :

Nvidia is The First vender of GPU and the term GPU was popularized by Nvidia in 1999 who marketed the GeForce 256 as "the world's first 'GPU', or Graphics Processing Unit, a single-chip processor with integrated transform, triangle setup/clipping, lighting and rendering engines that are capable of processing a minimum of 10 million polygons per second".

Another Rival vender ATI Technologies gave the term Visual Processing Unit or VPU with the release of the Radeon 9700 in 2002.

In following table we can see the evaluation of GPU from 1980 to Present.

Year	Detail
1980	No GPU. PC used VGA controller
1990	Add more function into VGA controller
1997	3D acceleration functions: Hardware, Texture, Shading
2000	A single chip graphics processor (beginning of GPU term)
2005	Massively parallel programmable processors
2007	Massively parallel programmable processors

TABLE-1--EVALUATION OF GPU

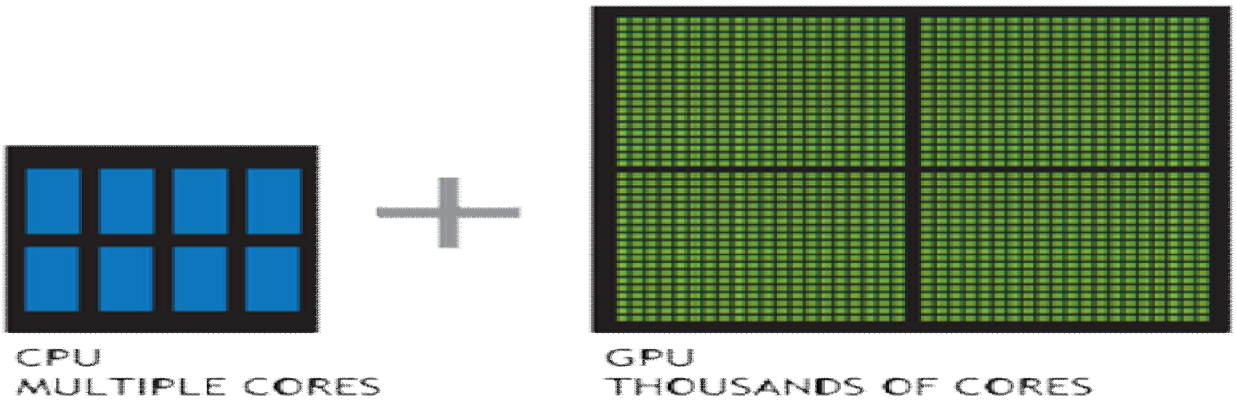


Fig. 1 Cores in CPU and GPU

II. WHAT IS GPU

GPUs have evolved to the point where many real-world applications are easily implemented on them and run significantly faster than on multi-core systems. In future hybrid systems will be seen with parallel core GPUs working simultaneously with multi core CPUs.

A. Characteristics of GPU :

The GPU is designed for a particular class of applications with the following characteristics. In the last few years, it has been identified that other applications also exists with similar characteristics and such applications can be mapped on to the GPU.Computational requirements are large. Real time performance needs millions of pixels per second and each pixel needs hundreds or more operations. To successfully complete the demand of complex real time applications GPUs must provide large amount of compute performance.

- Parallelism is substantial. Fortunately, the graphics pipeline is well suited for parallelism. Operations on vertices and fragments are well matched to fine-grained closely coupled programmable parallel compute units, which in turn are applicable to many other computational domains.
- Throughput is more important than latency. GPU implementations of the graphics pipeline prioritize throughput over latency. Generally in millisecond time scales the human visual system performs while operations within a modern processor take nanoseconds. This difference creates six-order-of-magnitude gap which means that latency of any operation is not important.

B. Why GPU?

Nowadays single processing is timing, so current market is up for parallel processing. The demands placed on GPUs from their native applications are, however, usually quite unique, and as such the GPU architecture is quite different from that of the CPU. Graphics processing is inherently extremely parallel so can be highly threaded and performed on the large numbers (typically hundreds) of processing cores found in the GPU chip.

So we can say that CPUs are great for Task Parallelism and GPUs are great for Data Parallelism.

1) Task Parallelism : Distribute the tasks across processors based on dependency. Coarse-grain parallelism.

2) Data Parallelism : Run a single kernel over many elements. Each element is independently updated and same operation is applied on each element. Lots of data on which the same computation is being executed.

C. GPU v/s CPU :

1) CPU : It has very fast caches which is greater for data reuse. It has lots of different processing/threads. It provide high performance on a single thread of execution.

2) GPU : It has lots of ALU units. It has fast access onboard memory. It run program on each fragment. Vertex. It provide high throughput on parallel tasks. The modern GPUs are very efficient in manipulating computer graphics and its highly parallel structure makes it more effective then general CPUs for processing large blocks of data that are done in parallel. As GPUs are done in parallel, so in manipulating computer graphics it performs more effectively. The GPUs can be found on silicon chip

or it can also be placed on motherboard which is closer to the CPU.

3) Test Matrices for measuring speed of GPU & CPU: In this testing we have Test the multiplication of two matrices by creating two matrices with random floating point values and we tested with matrices of various dimensions as following

TABLE 1: TEST MATRICES

DIM\TIME	CUDA	CPU
64x64	0.417465 ms	18.0876 ms
128x128	0.41691 ms	18.3007 ms
256x256	2.146367 ms	145.6302 ms
512x512	8.093004 ms	1494.7275 ms
768x768	25.97624 ms	4866.3246 ms
1024x1024	52.42811 ms	66097.1688 ms
2048x2048	407.648 ms	Didn't finish
4096x4096	3.1 seconds	Didn't finish

III.GPU + CPU

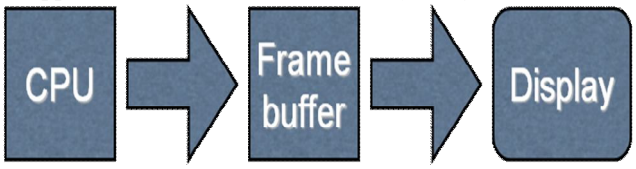
The Graphic Processing Unit (GPU) is an electronic circuit unit that is designed to rapidly manipulate and alter memory to increase the rate at which the system builds images in a frame. Its main purpose is to speed up the image building process that is intended for output to a display. The GPUs can be placed on the motherboard closer to the CPU or they can also be found on their own silicon chip.

Initially, the CPUs are responsible for handling all of the computing and instructions that it receives from the user and the system. Moreover, due to increase of technology, it demanded to take some pressure from CPU and transfer it to some other capable processor which gives you the best result. In this case it was compared that GPUs have more transistors than CPUs which can handle the work efficiently and provides better resolutions. Most of the GPUs transistors perform calculation related to 3D technologies. They were originally used to accelerate the memory-intensive work of texture mapping and rendering polygons. Many GPUs also support technologies for advanced gaming or digital playback, offering better and advanced systems.

When combining GPU with CPU, it is called Heterogeneous System.

IV. ARCHITECTURE OF GPU

The input to the GPU is in the form of geometric objects whether it is 2D or 3D. Most of input is given in typically triangles, in a 3-D world coordinate system. By performing many steps, those objects are shaded and mapped onto the screen, where they are joined to create a final picture. It is useful to first explain the particular steps in the canonical Graphics Pipeline before showing how the pipeline has become executable.



A. Early Graphics :

Earlier there is no any specialized Graphics Hardware. All processing execute in Software on CPU. Then generated result transmitted to the frame buffer. And Last result display on Monitor as define in below given figure.

Fig 2 CPU Architecture

In earlier days all processing perform in software on CPU but as the GPU evolved now graphics related processing done by GPU.

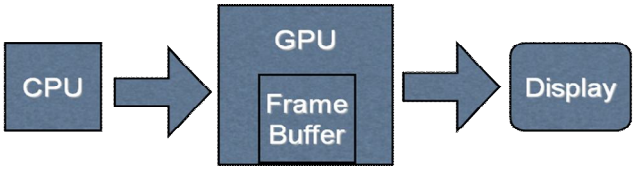


Fig 3 CPU including GPU Architecture

As shown in above figure Input from the CPU goes to GPU for processing and then after completion of processing generated result stored in frame buffer and at last result display on Monitor.

B. Graphics Pipeline :

There are some steps to complete the graphics process via fixed pipeline and that pipeline implemented on Graphics in GPU.

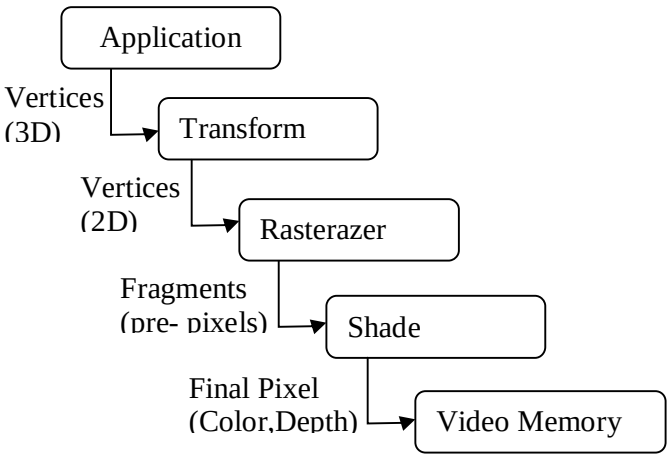


Fig 4 GPU Pipeline

1) Vertex Processor: The input primitives are formed from individual vertices. By processing the processor with the lights in the scene, each vertex must be moved in to screen space and should be shaded. For parallel hardware stage, each vertex should be computed independently and typical scenes have tens to hundreds of thousands of vertices. The vertices are assembled into triangles, the fundamental hardware-supported primitive in today's GPUs.

2) Rasterization: Rasterization is the process of determining which screen-space pixel locations are covered by each triangle. At each screen space pixel space triangle covers, it generates a primitive element called a fragment. Because many triangles may overlap at any pixel location, each pixel's colour value may be computed from several fragments.

3) Fragment Operations: Using colour information from the vertices and possibly fetching additional data from global memory in the form of textures (images that are mapped onto surfaces), each fragment is shaded to determine its final colour. Just as in the vertex stage, each fragment can be computed in parallel. This stage is typically the most computationally demanding stage in the graphics pipeline.

After completion of above stages the final Image is generated and goes for further process.

C. GPU Architecture Description :

Above we have seen Graphics pipeline for GPU that how graphics will created under GPU and now we will see the detailed Architecture of GPU and discuss about

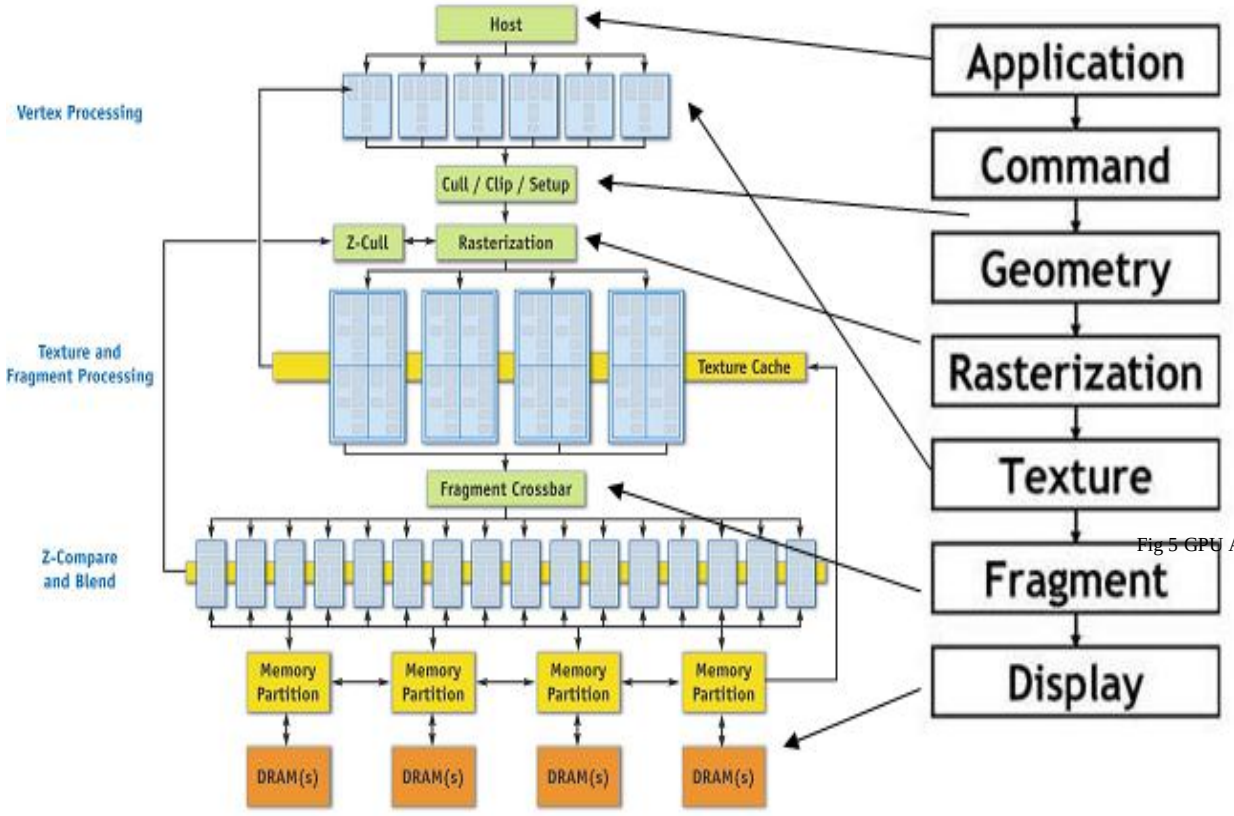
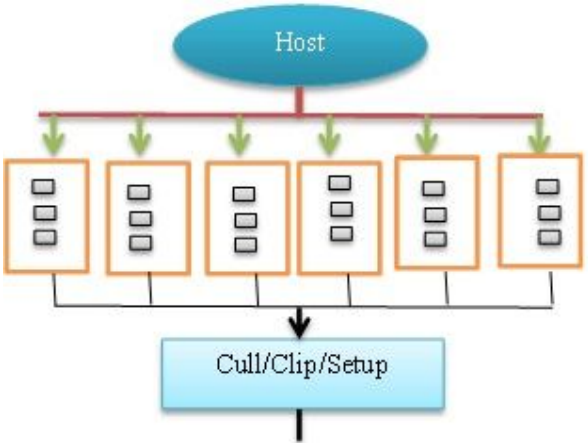


Fig 5 GPU Architecture in Detail

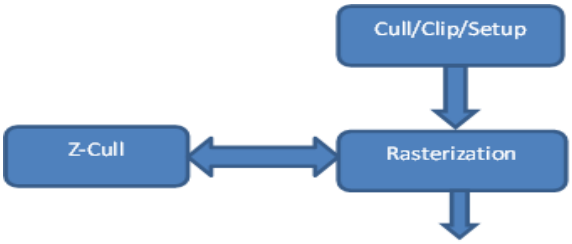
each and every components and its work[2][6].

As we seen in above figure different components are as follow in detail:

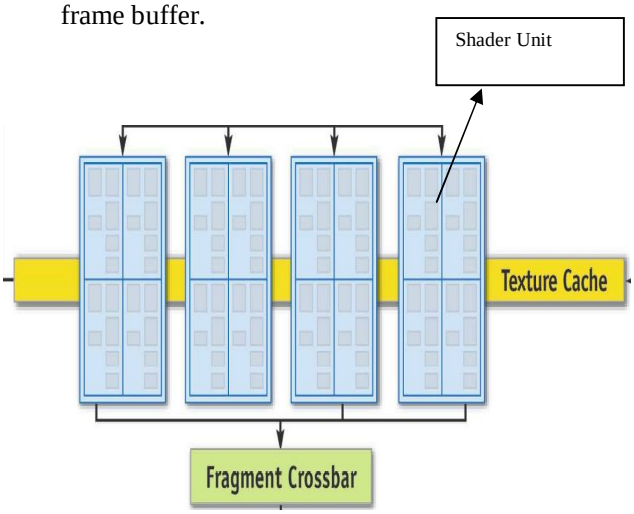
- Vertex Processor: It allows shader to be applied to each vertex. It also provides facility fetch data from texture by vertex shader.

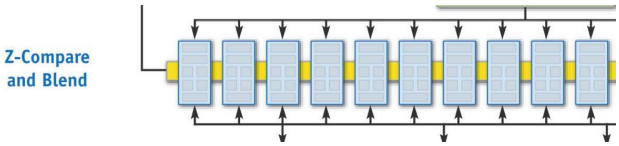


- Cull/Clip/Setup: It does per primitive operation and data preparation for Rasterization.
- Rasterization: Geometric object converted into pixel mapping.
- Z-Cull : It does quick pixel elimination



- A fragment is candidate Pixel and varying number of pixel pipelines.
- Texture Unit apply filters and shader unit can perform 8 math OPR [operation per second] with texture load.
- After above given process pixel almost ready for frame buffer.





- Z-Compare and Blend perform Depth Testing, Stencil Testing, Alpha operation and Load final colour for target buffer.

V. GPGPU

Graphics chips started as fixed-function graphics processors but became increasingly programmable and computationally powerful so, Computer scientists and domain scientists from various fields started using GPUs to accelerate a range of scientific applications. This was the advent of the movement called GPGPU, or General-Purpose computation on GPU.

GPUs does not execute a single thread very quickly but it emphasizes on executing many concurrent threads slowly which shows that GPUs have a parallel throughput architecture. This approach of solving general-purpose (i.e., not exclusively graphics) problems on GPUs is known as GPGPU.

While users achieved unprecedented performance (over 100x compared to CPUs in some cases), the challenge was that GPGPU required the use of graphics programming APIs like OpenGL and Cg to program the GPU. This limited accessibility to the tremendous capability of GPUs for science.

NVIDIA found that the potential of bringing this performance for the larger scientific community, invested in making the GPU fully programmable, and offered seamless experience for developers with familiar languages like C, C++, and FORTRAN. GPU computing momentum is growing faster than ever before. Today, some of the fastest supercomputers in the world rely on GPUs to advance scientific discoveries.

General-purpose computing on graphics processing units (GPGPU) is the utilization of a graphics processing unit (GPU), which typically handles computation only for computer graphics, to perform computation in applications traditionally handled by the central processing unit (CPU).

Large numbers of graphics chips or the use of multiple graphics cards in one computer further parallelizes the already parallel nature of graphics processing.

OpenCL is the currently dominant open general-purpose GPU computing language. The dominant proprietary framework is Nvidia's CUDA.

A. CUDA :

CUDA is Compute Unified Device Architecture which was the framework developed by Nvidia and this framework support OpenCL language.

CUDA is parallel computing and Programming Model implemented by GPU. CUDA has also been used to accelerate non-graphical applications in computational biology, cryptography and other fields by an order of magnitude or more. CUDA supports C, C++, Python, Fortran, Perl, Java, Ruby, Mat lab etc.

The initial CUDA SDK was made public on 15 February 2007, for Linux and Microsoft Windows. Mac OS X support was later added in version 2.0, which supersedes the beta released February 14, 2008. CUDA can easily work with almost all standard operating systems and it works with all all Nvidia GPUs from the G8x series onwards, including Quadro, the Tesla line and GeForce.

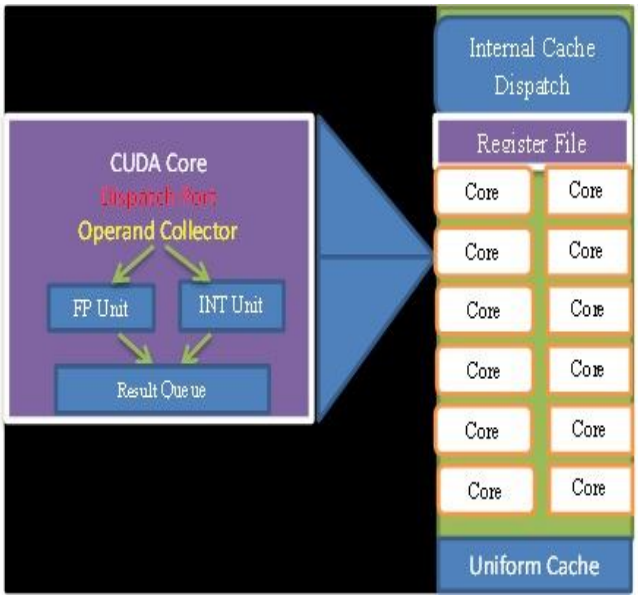


Fig 6 GPU with CUDA

B. OpenCL :

Open Computing Language (OpenCL) is a framework for writing programs that execute across heterogeneous platforms consisting of central processing units (CPUs), graphics processing units (GPUs). OpenCL can be used to give an application access to a graphics processing unit for non-graphical computing.

OpenCL was initially developed by Apple Inc., which holds trademark rights, and refined into an initial proposal in collaboration with technical teams at AMD, IBM, Qualcomm, Intel, and Nvidia.

OpenCL 1.1 adds significant functionality for enhanced parallel programming flexibility, performance and functionality and it was ratified by the Khronos Group on June 14, 2010.

VI. APPLICATIONS

Now we take a look at three applications that creatively use GPUs to make the overall computing experience as productive as aesthetically pleasing as it can be.

A. Gaming:

PC GPUs were originally invented for 3D gaming on PCs. One of the longest running gaming franchises on the PC is Sid Meier's Civilization. Civilization consists gamers everywhere chanting "just one more turn" for around a decade. The latest iteration, Civilization V, reinvents the game yet again, creating a gorgeous and compelling experience as you play the ruler of a kingdom that will "stand the test of time." Civilization V creatively uses Microsoft's DirectX 11 graphics API to immerse the player in the game. Rolling clouds hide unexplored areas of the map in a literal fog of war. Individual civilization leaders are artistically rendered in real time, rather than the previous canned animations, making use of graphical effects like heat shimmer and cloth animation to bring a little more realism to your virtual opponents. Using modern GPUs has also enabled Civilization V's developers to build animated characters that bring the maps to life. All this graphical goodness is wrapped in an exceptionally compelling game offering rich gameplay and robust replay ability.

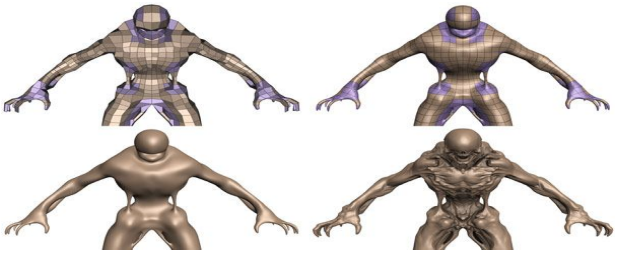


Fig 7 Gaming Character

B. Productivity:

Microsoft Office 2010 now offers GPU acceleration for some of its graphical elements, like WordArt and PowerPoint transitions. While Office 2010's use of the GPU won't overtask an AMD Radeon graphics card, AMD's Eye finity technology gives you the capability to run office on three to six displays using just one enabled Radeon HD 5000 or 6000 series card. The combination of GPU acceleration for key elements of Office 2010 plus three-monitor AMD Eye finity technology is a potent one. To integrate data across multiple applications easier and faster, have Microsoft excel, word and powerpoint in large windows, each on its own screen.

C. Video Editing:

Video editing demands heavy use of system resources even on high end PCs. Consumer applications, like Adobe Premiere Elements 9, are offering features previously available only for professionals. Transitions like page curl, sphere or card flip are all GPU-accelerated in Premiere Elements 9. Effects like refraction and ripple are also accelerated by a GPU. A graphics card with an AMD Radeon GPU will speed up preview and final rendering, making it faster and more fun to create your video.

VII. CONCLUSION

GPU is smaller less power consumption, easier to maintain, and inexpensive compared to a CPU cluster, so GPUs offer a convincing alternative. GPUs, originally designed to satisfy the rendering computational demands of video games, potentially offer performance benefits for more general purpose applications, including HPC simulations. We described the GPU architecture and detail about each component and its application like Gaming and other general purpose application where GPU mostly used.

GPU mostly used for computation for instructing that computation CPU is required where CPU instruct the GPU and GPU works according to that instruction, and give that result to the CPU.

VIII. REFERENCES

[1] Massively Parallel Computing CS 264 / CSCI E-292, Lecture #3: GPU Programming with CUDA | February 8th, 2011, Nicolas Pinto (MIT, Harvard)
[2] Wikipedia .com/GPU
[3] GPU Computing Graphics Processing UnitsVpowerful, programmable, and highly parallelVare increasingly targeting general-purpose computing applications. By John D. Owens, Mike Houston, David Luebke, Simon Green, John E. Stone, and James C. Phillips
[4] Software and Hardware Cooperative Computing, GPGPU, Prof. Hyesoon Kim School of Computer Science Georgia Institute of Technology
[5] Computers and Mathematics with Applications, journal homepage: www.elsevier.com/locate/camwa
[6] www.google.com