University of Klagenfurt
Digital Signal Processor (DSP)

GROUP MEMBERS :

MD SARWAR ZAHAN (MATRIX:1461419)

BASHIRU OTOKITI (MATRIX:1361474)

Topic: “GPU Processing” Proc. IEEE 96(5), 2008

AGENDA

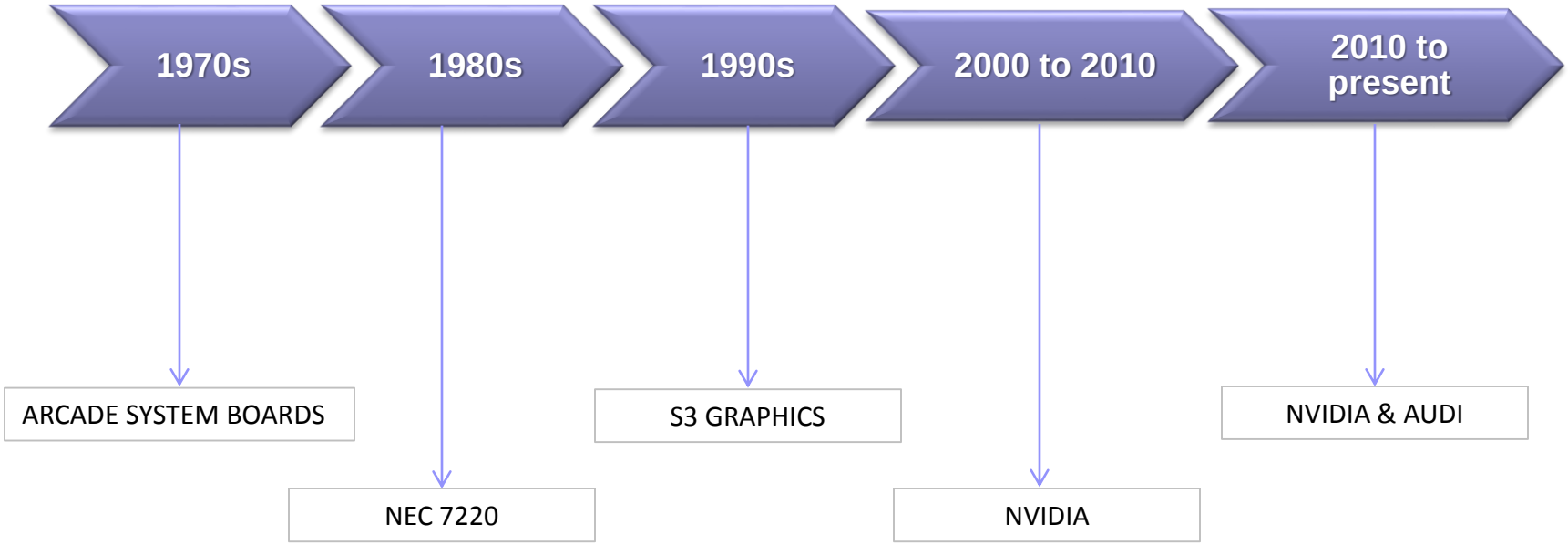
- Introduction
- About GPU
- Short History
- GPU Pipeline, Architecture
- GPU Algorithm
- CPU VS GPU
- Application of GPU
- Conclusion

WHAT IS GPU ?

A graphics processing unit (GPU) is a dedicated processor that performs rapid mathematical calculations for rendering high quality video and images .

The Abstract goal of a GPU is to enable a representation of a 3D world as realistically as possible.

SHORT HISTORY OF GPU



3D GRAPHICS RENDERING PIPELINE

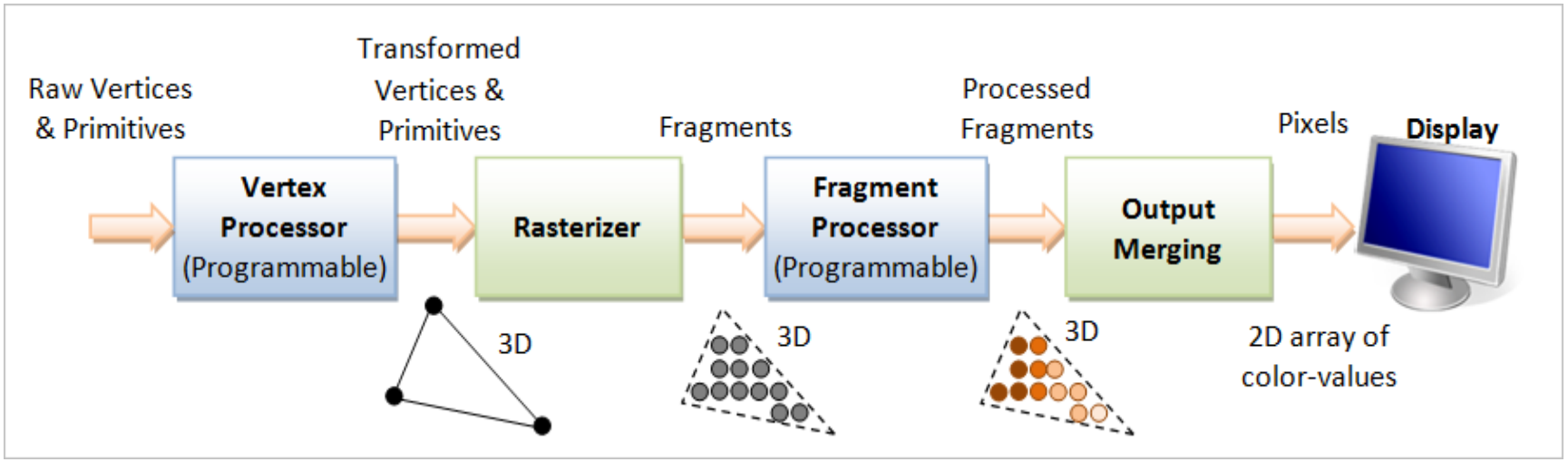


Image: 3D GRAPHICS RENDERING PIPELINE

Vertex Processing: Process and transform individual vertices.

Rasterization: Convert each primitive into a set of fragments.

Fragment Processing: Process individual fragments.

Output Merging: Combine the fragments of all primitives into color-pixel for the display.

GPU ARCHITECTURE

Image: NVidia GeForce 6800 Series GPU Board

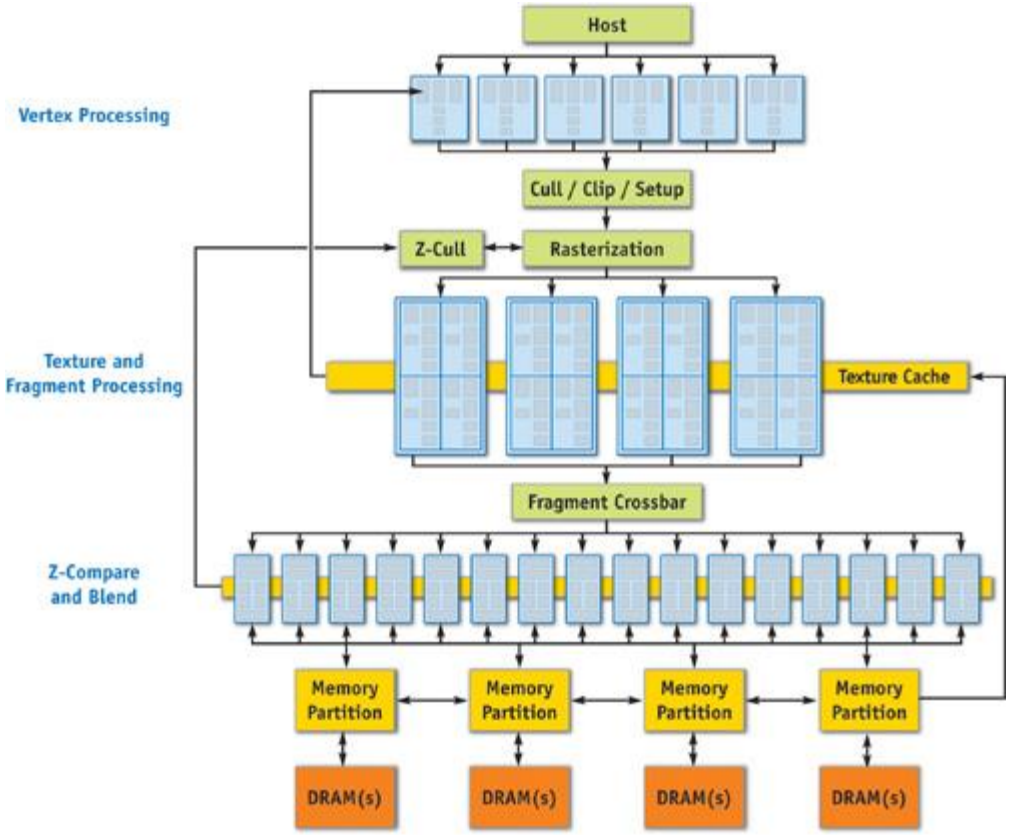


Image: NVidia GeForce 6800 GPU Architecture

- Host (CPU).
- 6 parallel vertex processors (receive data from the host).
- triangle setup stage (takes care of primitive assembly).
- rasterizer stage which produces the fragments.
- 16 processors (computes the output colors of each fragment).

GPU COMPUTING

- Parallelism is the future of computing.
- GPU has moved from a fixed-function into full-fledged parallel programmable processor.
- GPU follow a single program multiple-data (SPMD) programming model.

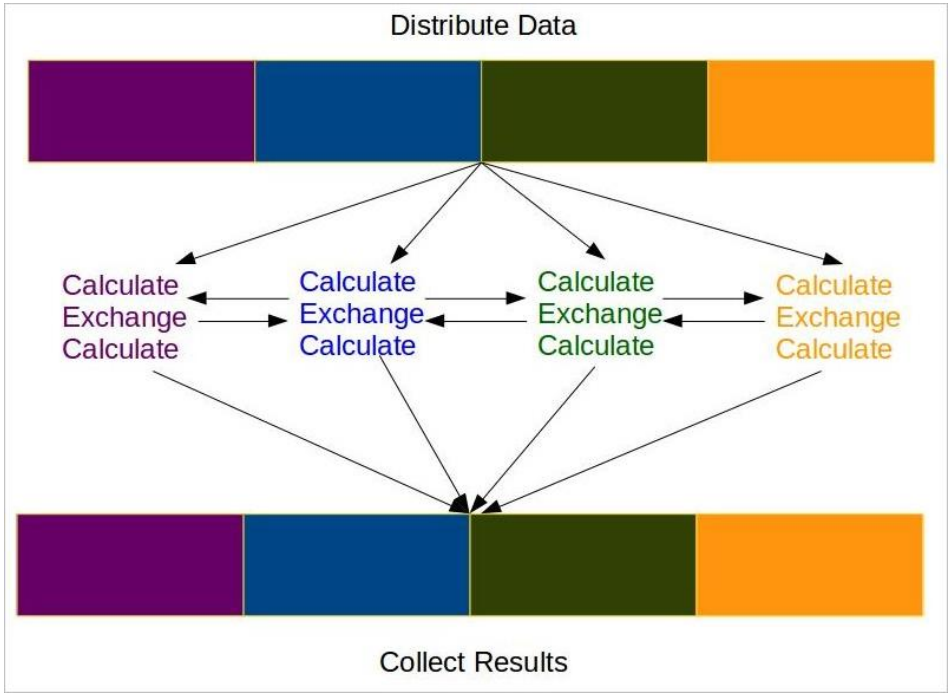


Image: SPMD Model

SPMD Tasks are split up and run simultaneously on multiple processors with different input for faster results.

GPU SOFTWARE ENVIRONMENTS

Famous languages for GPU programming:

- NVIDIA's (CUDA)
- OpenCL
- HLSL
- Cg

GPU PERFORMANCE EVALUATION

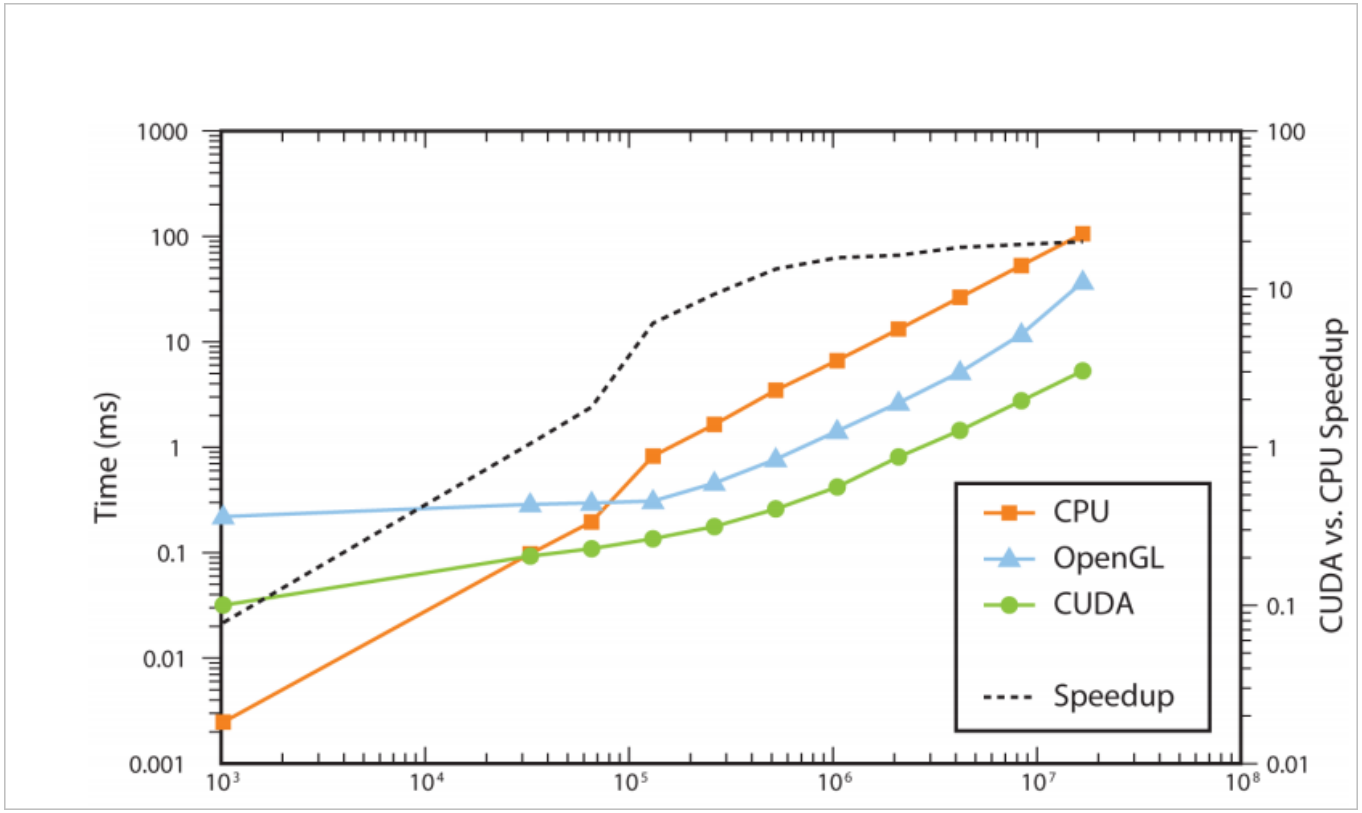


Image: GPU Performance

Scan performance on CPU, graphics-based GPU (using OpenGL), and direct-compute GPU (using CUDA). Results obtained on a GeForce 8800 GTX GPU and Intel Core2-Duo Extreme 2.93 GHz CPU

APPLICATIONS OF GPU

- Gaming.
- Digital image processing.
- Video processing.
- Audio signal processing.
- Physical based simulation.
- Scientific computing.

Image: 3D Game

GPU VS CPU

GPU

- Hundreds of simple cores.
- Highly optimize for parallel processing.
- Very high performance in multitasking.

Example:
NVIDIA GeForce GTX 280
1400 Mio.
933 GFLOPS .
113.3 GB/sec .

CPU

- Very few complex cores.
- Optimized for serial processing.
- Provide fast response times in a single task.

Example:
Intel Core i7-965 XE
731 Mio.
102 GFLOPS.
25.6 GB/sec.

FUTURE OF GPU COMPUTING

- Support double-precision floating-point.
- Higher bandwidth path between CPU and GPU.
- More tightly coupled CPU and GPU.
- Multiple GPU collaboration.
- Relationship to other parallel hardware and software.

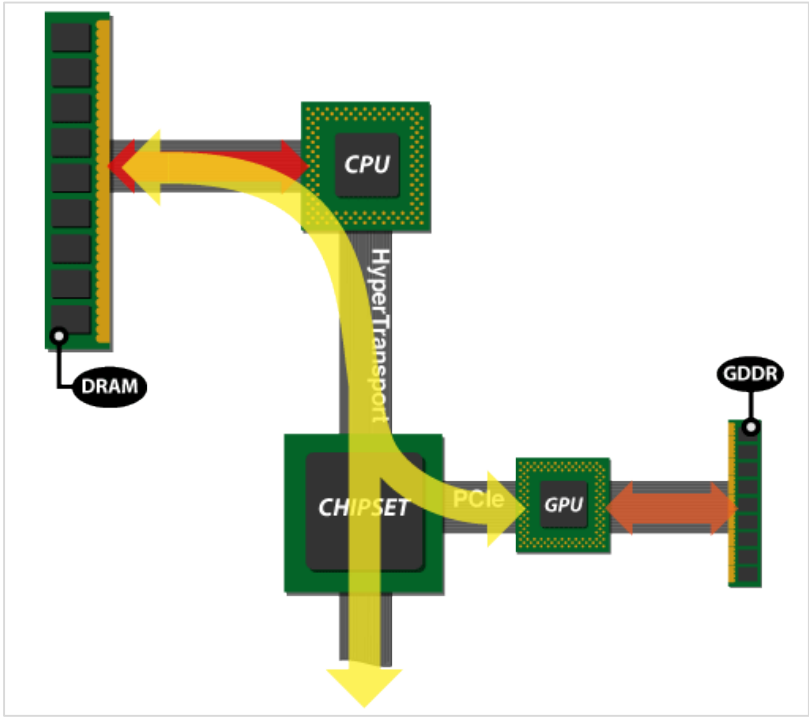


Image: HyperTransport path

GPU PROBLEMS

- Less arithmetic intensity.
- Poor programming models and tools.
- Design tradeoffs and their impact on the programming model.
- Managing rapid changes.
- Performance evaluation and headaches.

CONCLUSION

Future is towards to GPU computing. Now a days number of cores doubling every 2 to 4 years, and GPUs having their control logic applying on cores.

Remember when the first processor invented, it all about the networking.

Future computational power would be high collaboration between CPU and GPU.

Reference

[1] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, "GPU computing," Proceedings of the IEEE, vol. 96, no. 5, pp. 879–899, May 2008.

[2] http://www.nvidia.com/object/what-is-gpu-computing.html

[3] https://en.wikipedia.org/wiki/Graphics_processing_unit

[4] https://www.ntu.edu.sg/home/ehchua/programming/opengl/CG_BasicsTheory.html

[5] https://summerofhpc.prace-ri.eu/if-not-task-farming-then-what/

Question