Patrick Ogunbufunmi | Aditya Ramachandran | Tolulope Oshinowo

COMPUTER ARCHITECTURE PROJECT PROPOSAL

GPU ARCHITECTURE

Description

For our project, we plan on studying the origin, basic concepts, and producing a brief mini-lecture on the origin of GPU architecture. We plan to find out the circumstances around and at what point exactly the need arose for a separate computational architecture to handle intensive graphics processing. We plan to study the architectural similarities and different design constraints of a GPU and CPU, and make sense of common GPU technologies and jargon not easily understood by the layman. Our project will be a project of learning, where we try and relate computer architecture knowledge to the field of graphics architecture. We plan on analyzing the timeline of GPUs, observing at how they evolved from a gaming perspective, and noting how they're now used today in a variety of applications. We'll investigate what contributed to the trajectory and evolution of today's GPU's over the last 20-30 years, and will look into branching-offs in different markets (crypto, gaming, medical industry, etc).

Interesting concepts / technical jargon to investigate

- Threads -
  - Threads and thread management
    - Working on identical thread operations in parallel
    - Hyperthreading, multithreading
  - DISM
    - **D**eployment **I**mage **S**ervicing and **M**anagement),
- Processing cores in parallel vs serialized - Stream processors, aka CUDA cores
- Core speed vs Boost speed & Overclocking
- VRAM, Memory bus width, and other memory specifications
- Shaders and the Frame buffer

Deliverables:

Minimum Viable Product: We plan to gather content and information, and do research on the topic, and present our findings in lecture based format worth 2 weeks of information. We will talk about what we learnt, the origin of the GPU, the basic implementation and workflow of a standard GPU architecture, and reflect on our learnings throughout the process.

Planned: We plan on implementing any core subsystem of a GPU, i.e. a very simple vertex shader, if time permits. We plan to analyse such a subsystem and make it an understandable and workable project for the average Olin CompArch student. We probably won't be implementing any verilog in this deliverable, but we will show the data flow with a block diagram, and provide a basic set of Verilog implementation instructions to build a system.

Stretched: For our stretched deliverable, we are planning on implementing a key subsystem in a simple GPU architecture in Verilog, or any other applicable Hardware Description Language, which will perform a core function integral in GPUs. If time allows, we may be able to fully implement a simple but functional GPU so it follows a standard processing pipeline.

<u>Week by week Work plan</u>

Week 1:

- Research relevant documents and synthesize a common understanding of a standard GPU.

Week 2:

- Have a library of basic GPU architecture jargons readily explainable

Week 3:

- Start working on slides depicting the origin of the GPU and show how time tended toward the need for such an architecture

Week 4

- Have a deliverable ready for presenting based on previous findings

EXTRA

- Learn basic workflow of a GPU subsystem and add documentation / block diagram to slides
- Learn basic Verilog pipeline for GPU architecture and implement in code

References and Resources :

http://download.nvidia.com/developer/cuda/seminar/TDCI_Arch.pdf
https://eater.net/vga
http://people.cs.pitt.edu/~melhem/courses/3580p/gpu.pdf
https://vjordan.info/thesis/nvidia_gpu_archi/index.xhtml
https://medium.com/@smallfishbigsea/basic-concepts-in-gpu-computing-3388710e9239
https://courses.cs.washington.edu/courses/cse467/15wi/docs/prj3.pdf
3d fx