Introduction to Computer Architecture: Supervision 5

Lectures covered by the supervision: https://www.cl.cam.ac.uk/teaching/2526/IntComArch/

- Lecture 15: Graphics processing units (GPUs) – part II - Cuda, OpenCL.
- Lecture 16: Future directions. Energy efficiency. Performance. Reliability. Security.
- Summary: Computer design, SoC, and software engineering.

Past exam questions:
https://www.cl.cam.ac.uk/teaching/exams/pastpapers/t-ComputerDesign.html

Supervision questions:

1. 2019 Paper 5 Question 3 - parts a, b, c.
2. How is it possible that so many different devices, with completely different internal architecture are compatible with OpenCL and enable acceleration of execution? How does OpenCL enable this compatibility?
3. Compare and contrast the OpenCL programming model with CUDA and Vulkan using an example.
4. Describe a real-world problem you can solve with massive parallelism in GPUs. Write CUDA code to solve it. Describe how many threads you create, what is the size of thread blocks, how many thread blocks you have in your example, and how does the kernel grid look like. Elaborate on how this is deployed to a GPU.
5. Choose a real-world problem that can be efficiently solved by using many cores. Write a pseudo algorithm using OpenCL. Explain how your solution scales as the size of the problem increases. Explain how your solution scales as you move it to more powerful hardware (more cores).
6. Why is energy efficiency the "new fundamental limiter of processor performance"? In the context of energy efficiency, discuss heterogeneous computing. As examples, describe Arm's big.LITTLE system and Apple A16 Bionic chip.
7. Discuss a general multicore CPU vs ASIC vs DSP. When does it make sense to implement functionality in a specialised accelerator rather than within a general-purpose core?
    a. What are important aspects to compare?
    b. How would you decide which accelerator(s) to use if you were designing a system?
8. 2020 Paper 5 Question 3 - parts d, e.
9. Discuss Spectre and Meltdown bugs.
10. Create a diagram with all discussed elements of a computer system (e.g., cache, RAM). Discuss common units of time for operations that these elements perform (e.g., time to fetch memory from cache and RAM, time to execute an instruction).
11. Summarize the main message from "Lecture 15: Cuda, OpenCL" in 1-3 sentences?
12. Summarize the main message from Lecture "16: Future directions. Energy efficiency. Performance. Reliability. Security" in 1-3 sentences?

BONUS :

13. Create an example using SYCL (OpenCL based library).
14. Explain approximate computing.
15. Create an example of a SoC, on which some processes and threads might execute concurrently. Discuss sources of non-determinism starting from cache and pipelines, to RAM access, network, scheduling, pre-emption, synchronisation between threads… Consider there are also some cloud services. Discuss what range of delays non-determinism can cause.
16. Discuss Misra C in the context of security.
17. Create an example with CUDA and commit it to GitHub.

18. Create an example with OpenCL and push it to GitHub.

Save your answers into MS Teams or email them to me. Please use the following naming pattern:

ICA_Supervision_5_Answers_<last name>_<first name>_Michaelmas_2025

Send your answers as a pdf, doc, image, or any other format of a document for which there exists an easily available software to open.

Jasmin JAHIĆ
jj542@cam.ac.uk
https://www.cl.cam.ac.uk/~jj542/