IN5050: Programming heterogeneous multi-core processors

… because the OS-course is just too easy!

Home Exam 2: Video Encoding on nVIDIA Tegra Xavier Heterogeneous SoC

March 5th, 2025

Håkon Kvale Stensland

Precode

- Continue with your code from Home Exam 1
- You are not allowed to change the Motion Estimation, Motion Compensation, or DCT algorithms.
- You are not allowed to paste code from other projects/encoders.
- You only need to optimize the Codec63 encoder!
- You will not get points on further improvements to the Motion Estimation and Motion Compensation from Home Exam 1, you are allowed to adapt your code to the Volta architecture.

Your task

- Utilize the heterogeneous GPU and CPU architecture and tune your application with the Volta Tuning Guide and other tricks. On the Jetson Xavier, the CPU and GPU share the same memory.
- Start by profiling your solution from Home Exam 1 on the Tegra to see which parts of the encoder are the bottlenecks.
- Remember, more profiling might be needed to find new bottlenecks after optimizing one part of the code. Use the Nvidia profiler to profile the GPU code.
- Write a scientific report detailing which parts of the encoding process benefited from your optimizations. The report should also explain how your code works.
- Remember to also report on optimization that did not work as you expected!

How are you evaluated?

- (**20%**) An implementation that compiles, **runs**, and can **produce a correct video output** (remember c63pred).
- (**30%**) **Effective use of architecture:** Correct and efficient use of memory on GPU. Efficient data movement between CPU/GPU. Investigate the potential of half-precision. Multiple CPU threads for SIMD.
- (**20%**) **Parallelization potential:** DCT/iDCT with SIMD.
- (**20%**) **Report**. Profiling between different steps. What worked, what didn't work, and why? Quality of code.
- (**10%**) **Performance** of the solution.
- Bonus points can also be given for non-obvious optimizations.

Formal Information

- **Deadlines:**
  - Design: **Wednesday March 12th – 12:00**
  - Code: **Monday March 31st – 23:59**
  - Report: **Monday April 7th – 14:00**

- The assignment will be graded and count 33% of the final grade.

- Deliver your design and poster (as .pdf): in5050@ifi.uio.no
- Deliver your code and report : https://devilry.ifi.uio.no/

- Prepare a poster (PDF or PPT) and a short talk (5 minutes) for the class on (March 31st).
  - Best poster & presentation will be awarded!

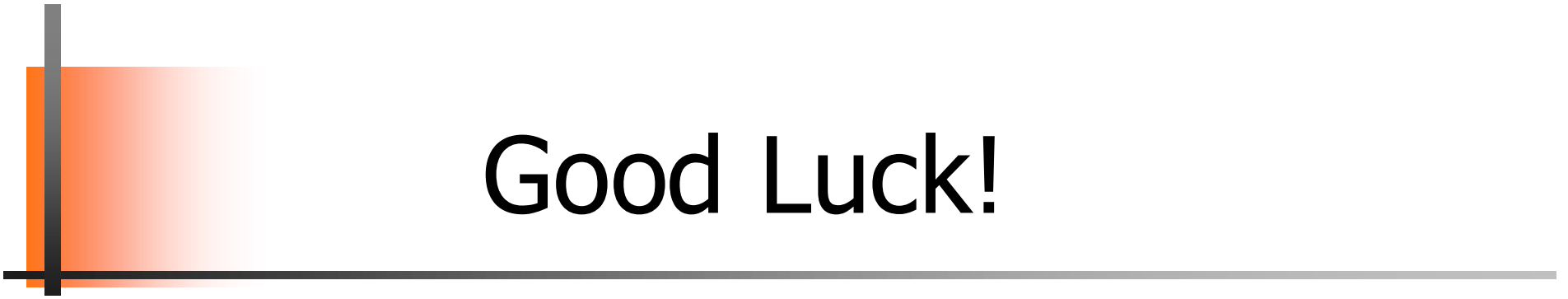
Last but not least!

- Codec63 precode is available for download in git. Clone the repository and work on your own local version.

  git clone https://github.com/griwodz/in5050-codec63.git

- Use the CUDA branch.
- Bugs can be reported in Mattermost or by e-mail.

- Use the Mattermost to discuss and share tips, tricks, and troubles!
  - Everything is allowed except the sharing of code! ☺


Good Luck!

Bonus: Nvidia GTC 2025

- Nvidia's GPU Technology Conference 2025 is this year a hybrid event.

  https://www.nvidia.com/gtc/

- Sign up for free, register for events, and participate.
- Several recommended talks for IN5050 on CUDA development, debugging, and performance analysis.
  - See Mattermost for more details