IN5050: Programming heterogeneous multi-core processors

… because the OS-course is just too easy!

Home Exam 2: Video Encoding on Tegra Xavier using nVIDIA CUDA

March 18th, 2022

Håkon Kvale Stensland

Video Encoding



- Pål still wants to encode some videos…
- Pål got a brand-new PowerPoint 2021 license, but animations are more resource demanding than he thought, and the email client uses the remaining CPU resources!
- However, Pål remember that his SoC has a GPU, so he needs you to help him use the GPU for video encoding (Use the CPU as little as possible, it is very occupied running Mail and PowerPoint animations).

Precode

- Same precode as Home Exam 1!
- You are not allowed to change out 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!
- Your implementation is supposed to only focus on the GPU, we will not give points for threading on the CPU side.

Your task

- Utilize the data parallel GPU architecture and use the *Volta Tuning Guide*, and other tricks to tune your application.
- Start by profiling the encoder to see which parts of the encoder that are the bottlenecks.
- Remember, after optimizing one part of the code, more profiling might be needed to find new bottlenecks. Use the Nvidia profiler to profile the GPU code.
- Write a scientific report with details on which parts of the encoding process that 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%**) Implementation that **compiles, runs**, and can **produce a correct video output** (remember c63pred).
- (**20%**) **Effective use of architecture**: Correct and efficient use of memory on GPU. Efficient data movement between CPU/GPU. Investigate potential of half-precision on GPU.
- (**30%**) **Parallelization potential:** Motion Estimation, DCT/iDCT and Motion Compensation.
- (**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: **Friday March 25th – 12:00**
  - Code: **Friday April 8th – 23:59**
  - Report: **Monday April 11th – 12:00**

- 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 (to show on screen) and a short talk (5 minutes) to pitch your poster for the class (April 8th).
  - Best poster & presentation will be awarded!

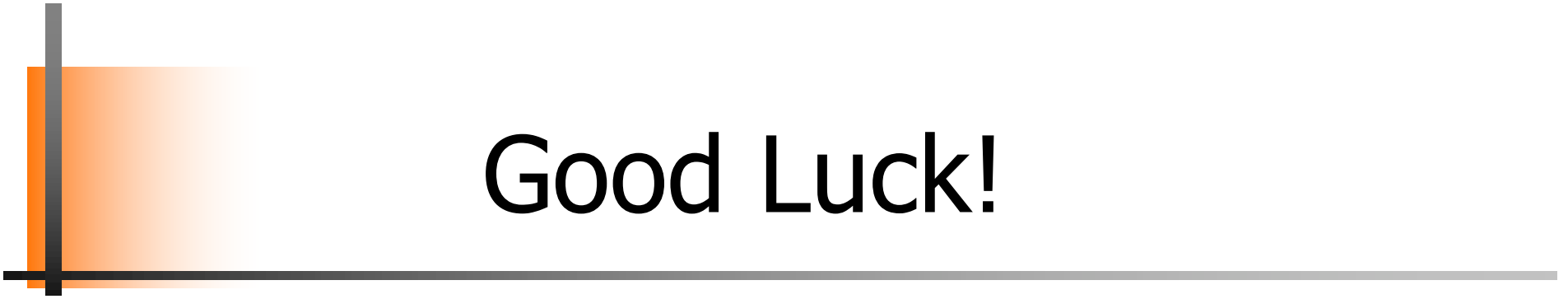
Last but not least!

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

git clone https://bitbucket.org/mpg_code/in5050-codec63.git

- Bugs in the code can be reported in Bitbucket's issue tracking system, Mattermost, or e-mail

- Use **Mattermost** for discussions and questions.


Good Luck!

Bonus: Nvidia GTC 2022

- Nvidia's GPU Technology Conference 2022 is because of the pandemic an online event this year

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

- Sign up for free, register for events and participate.
- Highly recommended talks for IN5050:
  - March 21 - 18:30-19:30 - Jetson Developer Tools [SE2597]
  - March 21 - 20:30-21:20 - Demystifying Unified Memory on Jetson [SE2600]
  - March 24 - 19:00-19:50 - How to Understand and Optimize Shared Memory Accesses using Nsight Compute [S41723]
  - … and many more