IN5050: Programming heterogeneous multi-core processors

… because the OS-course is just too easy!

Home Exam 1: Video Encoding on nVIDIA GPUs using CUDA

February 3rd, 2025

Håkon Kvale Stensland

Precode

- Codec63-precode will also be used on Home Exam 2
- 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.

** There is a CUDA-branch, and some fuctions are moved arround.*

Before you start…

- Four lines must be added to your group users *.bashrc* file

```
PATH=$PATH:/usr/local/cuda-12.8/bin
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-12.8/lib64:/lib

export PATH
export LD_LIBRARY_PATH
```

- The CUDA versions on the lab machines are:
  - CUDA 12.8

- CUDA Code samples are on GitHub:
  https://github.com/nvidia/cuda-samples
  - Copy and build in your user's home directory

Your task

- Utilize the data parallel GPU architecture, and other tricks to tune your application. Focus on the Motion Estimation / Motion Compensation part of the code.

- 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.
- (**30%**) **Parallelization potential:** Motion Estimation and Motion Compensation offloaded to the GPU.
- (**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: **Monday February 10th – 12:00**
  - Code: **Wednesday February 26th – 23:59**
  - Report: **Monday March 3rd – 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 (February 26th).
  - 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://github.com/griwodz/in5050-codec63.git
```

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

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

Good Luck!