IN5050: Programming heterogeneous multi-core processors

… because the OS-course is just too easy!

Home Exam 3: Distributed Video Encoding using Dolphin PCI Express Networks

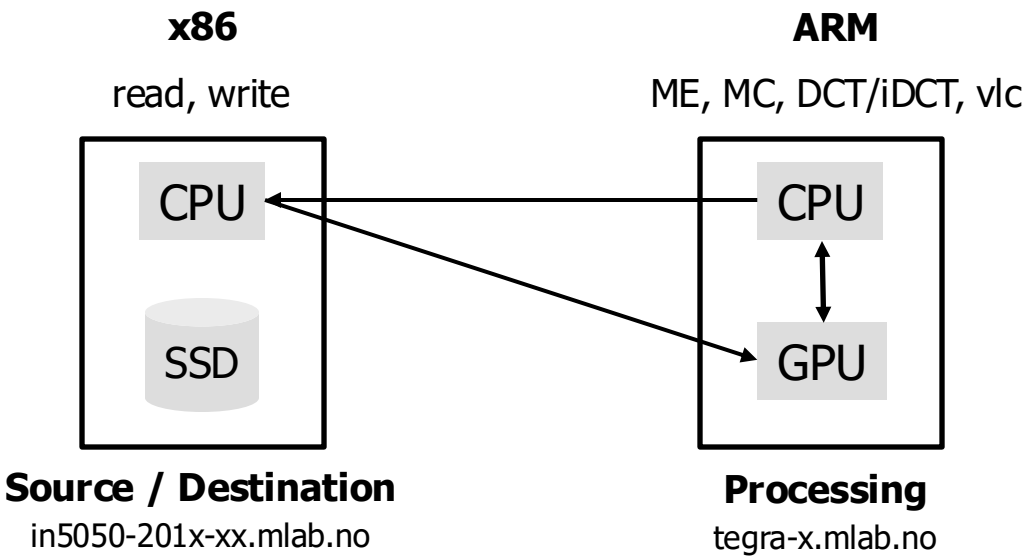
May 5th, 2025

Håkon Kvale Stensland

Precode

- For the encoder part, the same precode as Home Exam 1 and 2!
- Will be a dedicated Dolphin-branch of the precode with the run scripts presented last week.
- 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!
- The use of two machines and PCIe interconnect is an absolute requirement for passing this exam.

Architecture for Home Exam 3



- 1 source and destination machine (Intel x86).
  - in5050-201x-xx.mlab.no
- 1 processing machine with eight Carmel cores and a Volta GPU (GV11B).
  - tegra-x.mlab.no
- The machines are connected with a PCI Express switch.

Your task: I/O Machines

- You can have up to three frames “in flight” (not counting any double buffering).
- DMA is recommended for data transfers, use PIO for synchronization. Remember, you might have to try multiple approaches. There are also different techniques for using SISCI to communicate between machines.
- **Source & Destination (in5050-201x-xx.mlab.no):**
  - This machine has an older GPU, no need to use this GPU for the assignment.
  - Four cores are available on this machine, use threads if needed.
  - The CPU architecture is different compared to the processing machines (x86 vs. ARMv8.2)
- The I/O machine should not handle the video processing (ME, MC, DCT, iDCT).
- You can run VLC on the I/O machines if needed.

Your task: Processing Machine

- No extra points will be given for extending the SIMD/NEON or GPU/CUDA implementation from the first and second assignments on the processing machine.
- You may choose whether you use SIMD and/or GPU for the encoding steps on this assignment.
- The efficiently of your ME, MC, and DCT/iDCT will **not** be evaluated on this assignment; this has already been evaluated on H1 and H2.
- You can use all CPU cores on the processing machine as well. It has eight Nvidia Carmel ARM cores available.
- In the report, you should describe all the optimizations that have been done, as well as the ones that did not work.

How are you evaluated?

- (**20%**) Implementation that **compiles, runs**, and can **produce a correct video output** (we also check the motion prediction).
- (**25%**) **PCIe Interconnect:** DMA vs. PIO discussion, granularity of transfers, etc.
- (**35%**) **Distribution:** Correct setup, minimize blocking and stalling, efficient IO on x86, data movement on ARM.
- (**20%**) **Report**. Profiling between different steps. What worked and didn't work? Comparing different approaches and quality of code.
- Bonus points can also be given for non-obvious optimizations and good analysis of unexpected results.

Formal Information

- **Deadlines:**
  - Design: **Wednesday, May 14th – 10:00**
  - Code: **Wednesday, May 28th – 23:59**
  - Report: **Wednesday, June 4th – 14:00**

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

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

- Prepare a poster (to show on screen) and a short talk (5 minutes) to pitch your solution for the class (May 28th).

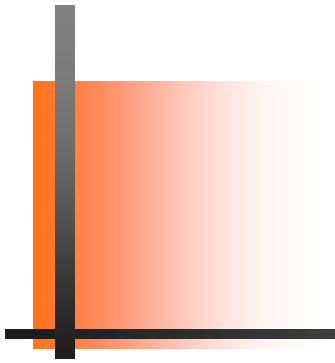
Last but not least!

- Codec63 precode is available for download in git. Clone the repository and work on your own local version.
- It will be a ”dolphin” branch.

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

git clone https://bitbucket.org/mpg_code/sisci-assignment.git

- Bugs in the code can be reported in Mattermost or by email.



Good Luck!

PS! Use the Q&A session!