IN5050:
Programming heterogeneous multi-core processors

Introduction

Håkon Kvale Stensland
January 22nd 2019
Overview

- Course topic and scope
- Background for the use of parallel processing with heterogeneous multi-core processors
- Examples of heterogeneous architectures
IN5050: The Course
People

- Håkon Kvale Stensland
  email: haakonks @ ifi

- Carsten Griwodz
  email: griff @ ifi

- Professor Pål Halvorsen
  email: paalh @ ifi

Course email:
in5050 @ ifi

- Guest lectures from FLIR Unmanned Aerial Systems
  Kristoffer Robin Stokke

- Guest lectures from Dolphin Interconnect Solutions
  Roy Nordstrøm & Lars Bjørlykke Kristiansen
Time and place

- **Lectures:**

  Tuesday **09:15 - 12:00** (sometimes **10:15** or **08:15**)

  *Pascal (OJD / IFI)*

- Parallel processing: Thinking parallel.
- The theory behind the programming models.
- Introduction to the architectures (SIMD, GPU, PCIe).
- Memory & Cache hierarchies.
- Interconnection Networks.
- Walk-through of simple programming examples on the new architecture.

- Check course webpage for latest details!
Time and place

- **Group exercises:**

  Friday **12:15 – 15:00**  
  *Fortress (OJD / IFI)*

- Introduction to video coding.
- Learn to program the architectures, and use the APIs needed for the solving the Home Exams.
- Poster session presenting the the Home Exam to the class.
- Walk-through and discuss an example solution to the simple video coding example.
- Questions and answers about using the new architecture.
- Presentation and walk-through of the next Home Exams.
About IN5050: Topic & Scope

**Content:** The course gives ...

− ... an overview of heterogeneous multi-core architectures in general and three architectures in particular.

− ... an *introduction* to programming heterogeneous multi-core processors
  
  • NEON SIMD for ARM processors
  
  • Nvidia’s family of GPUs and the CUDA programming framework
  
  • Multiple machines connected with Dolphin PCIe links

− ... some ideas of how to utilize heterogeneous multi-core processors for a multimedia workload.

− ... experience with working on architectures where the software infrastructure and documentation is not as streamlined as on x86.
About IN5050: Topic & Scope

**Tasks:**
The important part of the course is lab-assignments where you program each of the three examples of heterogeneous multi-core processors.

3 graded home exams (counting 33% each):

1. Home Exam 1: **ARM NEON**
   - Video encoding – Improve the performance of *video compression* by using NEON SIMD instructions a single ARM Cortex-A57 core.

2. Home Exam 2: **Nvidia graphics processing unit**
   - Video encoding – Improve the performance of *video compression using the Maxwell GPU* on the Nvidia Tegra X1 system on a chip.

3. Home Exam 3: **Distributed system** scenario
   - Video encoding – The same as above, but exploit the parallelism on multiple GPUs connected with Dolphin PCIe links.

You will be working together in groups of two. Try to find a partner before the group session next week!
Background and Motivation:

Moore’s Law

“The number of transistors in a dense integrated circuit will approximately double every two years”
Motivation: Transistors

Billion transistors integrated

1971:
- 2,300 - Intel 4004

2019:
- 21.1 billion - nVIDIA GV100 (Volta)
Motivation: Clock frequency?

- Before mid-2000s vision was that clock frequency would continue to increase linearly...
- **However**, clock frequency has not increased since 2012

![Graph showing clock frequency trends from 1970 to 2020](image)
Motivation: Power?

- As the number of transistors grows and the production process shrinks, the area for heat transfer also shrinks.
Putting it all together...

- First CPU with multiple cores on the same die released in 2005.
Multicores!
Symmetric Multi-Core Processors

AMD Ryzen ("Summit Ridge")
Symmetric Multi-Core Processors

- **Good**
  - Growing computational power

- **Problematic**
  - Growing die sizes
  - Unused resources
    - Some cores used much more than others
    - Many core parts frequently unused

- Why not spread the load better?

⇒ **Heterogeneous Architectures!**
nVIDIA Tegra X1 ARM SoC

- One of many multi-core processors for handheld devices

- 4 ARM Cortex-A57 processors
  - 4 ARM Cortex-A53 cores
  - Out-of-order design
  - 64-bit ARMv8 instruction set
  - Cache-coherent cores (not IO)
  - 128-bit ARM NEON SIMD

- Several “dedicated” co-processors:
  - 4K Video Decoder
  - 4K Video Encoder
  - Audio Processor
  - 2x Image Processor

- Fully programmable Maxwell-family GPU with 256 simple cores.
Jetson TX1 – The platform for IN5050

Embedded development kit from Nvidia with the Tegra X1 SoC, targeting deep learning and computer vision.

- Quad-core ARM Cortex-A57
- 4 GB LPDRAM4
- 16 GB eMMC
- USB3, USB2
- Gigabit Ethernet
- 4-lane PCI Express Gen2
- 256-core Maxwell GPU
- Ubuntu 16.04 LTS (Linux for Tegra)
- Up to 1 TFLOPS of FP16 performance
  - TPD: 10W
Co-Processors

- The original IBM PC included a socket for an Intel 8087 floating point co-processor (FPU)
  - 50-fold speed up of floating point operations

- Intel kept the co-processor up to i486
  - 486DX contained an optimized i487 block on-die.
  - Still separate pipeline (pipeline flush when starting and ending use)
  - Communication over an internal bus

- Commodore Amiga was one of the earlier machines that used multiple processors
  - Motorola 680x0 main processor
  - Blitter (block image transferrer - moving data, fill operations, line drawing, performing boolean operations)
  - Copper (Co-Processor - change address for video RAM on the fly)
General Purpose Computing on GPU

- The
  - high arithmetic precision
  - extreme parallel nature
  - optimized, special-purpose instructions
  - available resources
  - ...

... of the GPU allows for general, non-graphics related operations to be performed on the GPU

- Generic computing workload is off-loaded from CPU and to GPU

⇒ More generically: Heterogeneous multi-core processing
nVIDIA Volta GPU Compute Architecture

- 21.1 billion transistors
- 5120 “CUDA-cores”
- 640 “Specialized” cores for AI (tensor cores)

- 4096-bit memory bus (HBM2)
- 32 GB memory
- 900 GB/sec memory bandwidth

- 15 TFLOPS single precision performance

- PCI Express 3.0
- NVLink 2
Heterogeneous multi-core processors are already everywhere

**Challenge:** programming

- Need to know the capabilities of the system
- Different abilities in different cores
- Memory bandwidth
- Memory sharing efficiency
- Need new methods to program the different components