IN5050
Programming heterogeneous multi-core processors



Introduction

Håkon Kvale Stensland

January 17th, 2022

[simula . research laboratory]

UiO : University of Oslo

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
- Pål Halvorsen
- Guest lectures from:
  - Hudley
  - Muybridge
  - Dolphin Interconnect Solutions

Course email:
in5050 @ ifi

Time and place

- **Lectures**:

  Monday **09:15 - 12:00**

  Perl (OJD / IFI) & Zoom

- 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**

  Sed (OJD / IFI) & Zoom

- 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):
  - Deliver code and make a demonstration explaining your design and code to the class

  1. Home Exam 1: **ARM NEON**
     - Video encoding – Improve the performance of *video compression* by using NEON SIMD instructions a single custom ARMv8 core.
  2. Home Exam 2: **Nvidia graphics processing unit**
     - Video encoding – Improve the performance of *video compression using the Volta* GPU on the Nvidia Tegra Xavier system on a chip.
  3. Home Exam 3: **Distributed system** scenario
     - Video encoding – Use the complete Nvidia Tegra Xavier SoC to offload *video compression* from a standard x86 computer.

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

Background and Motivation:

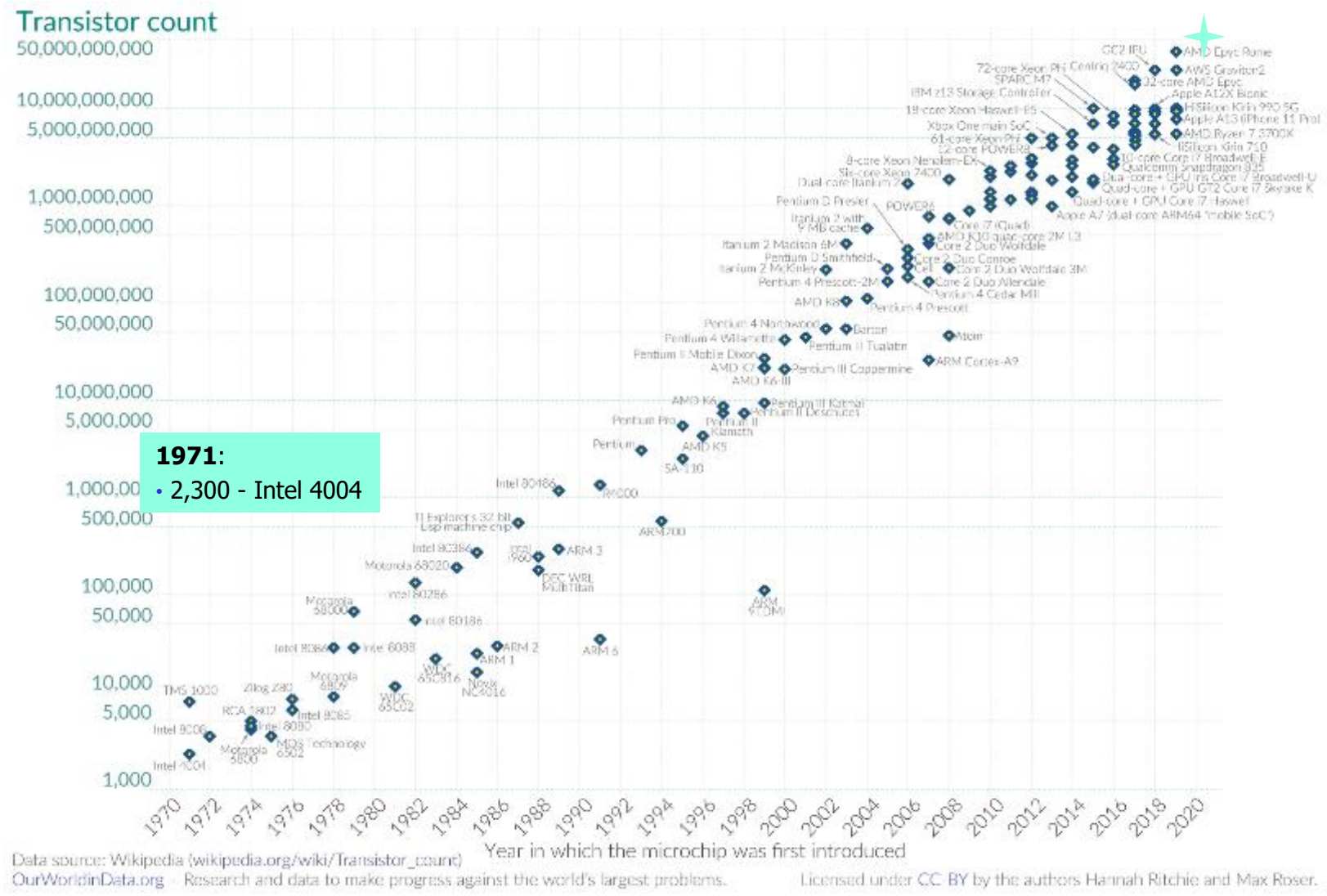
Moore's Law

"The number of transistors in a dense integrated circuit will approximately double every two years"

Motivation: Transistors

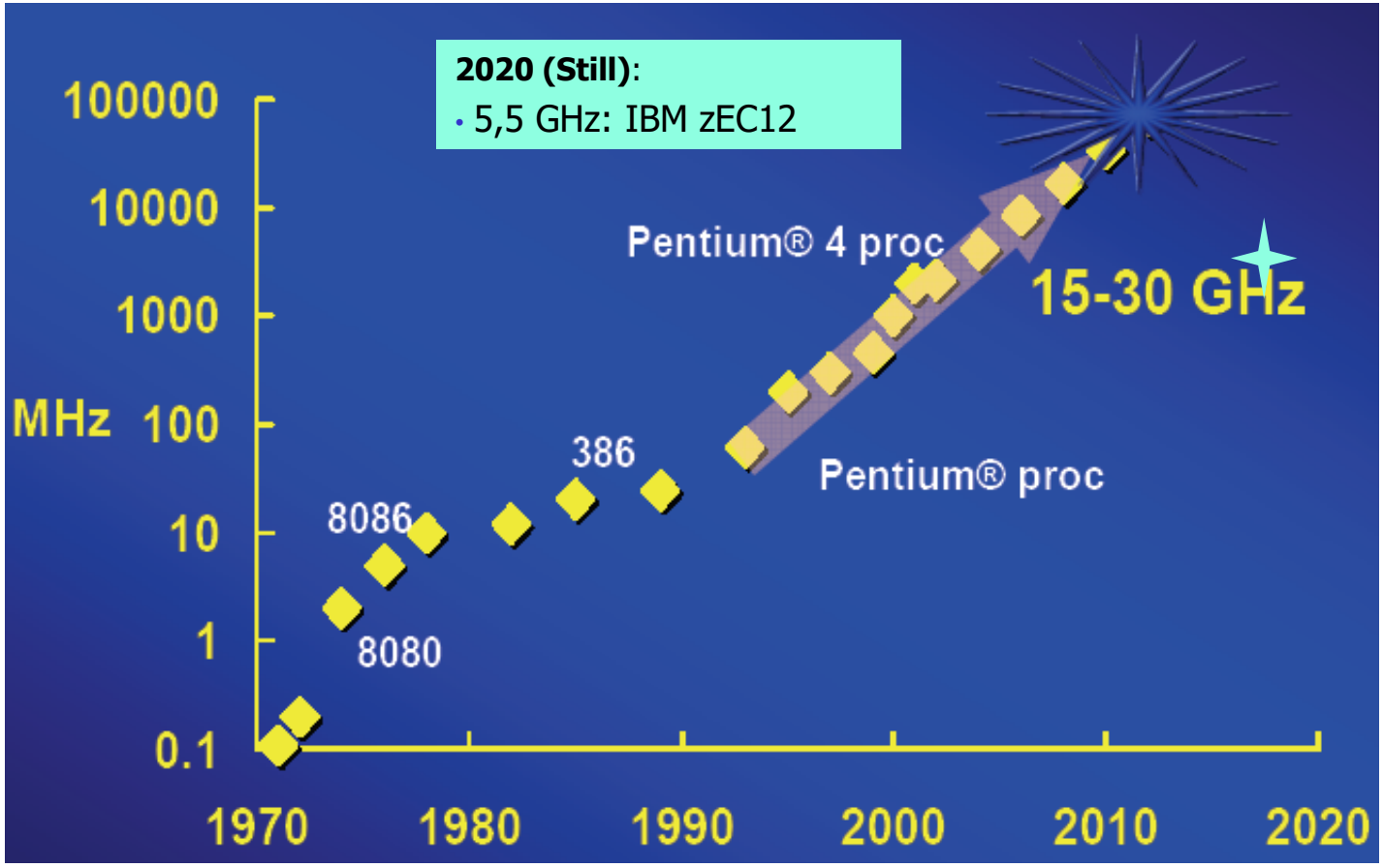
- Billion transistors integrated

2022:
- 59 billion – AMD MI250X (Alderbaran)



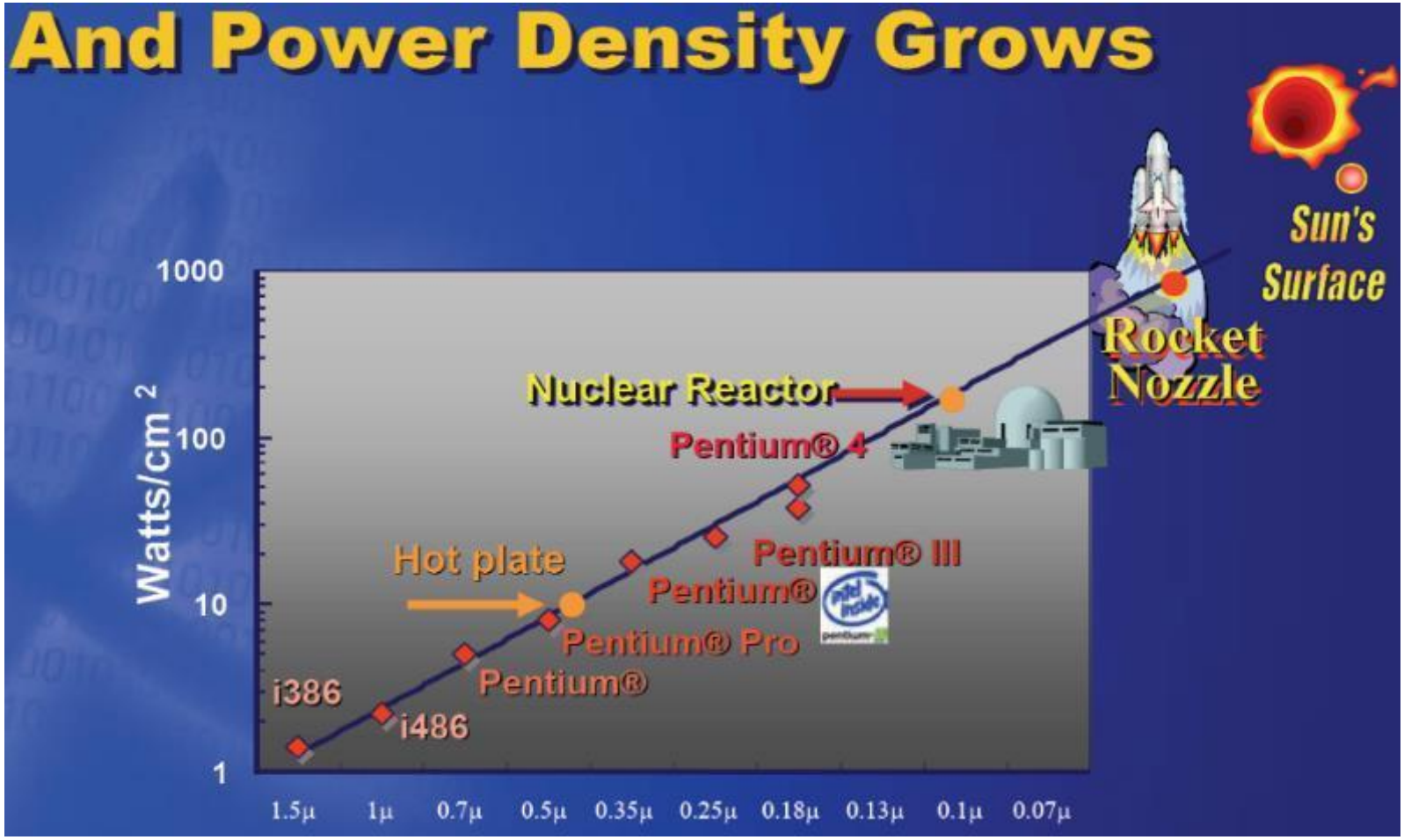
Motivation: Clock frequency?

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



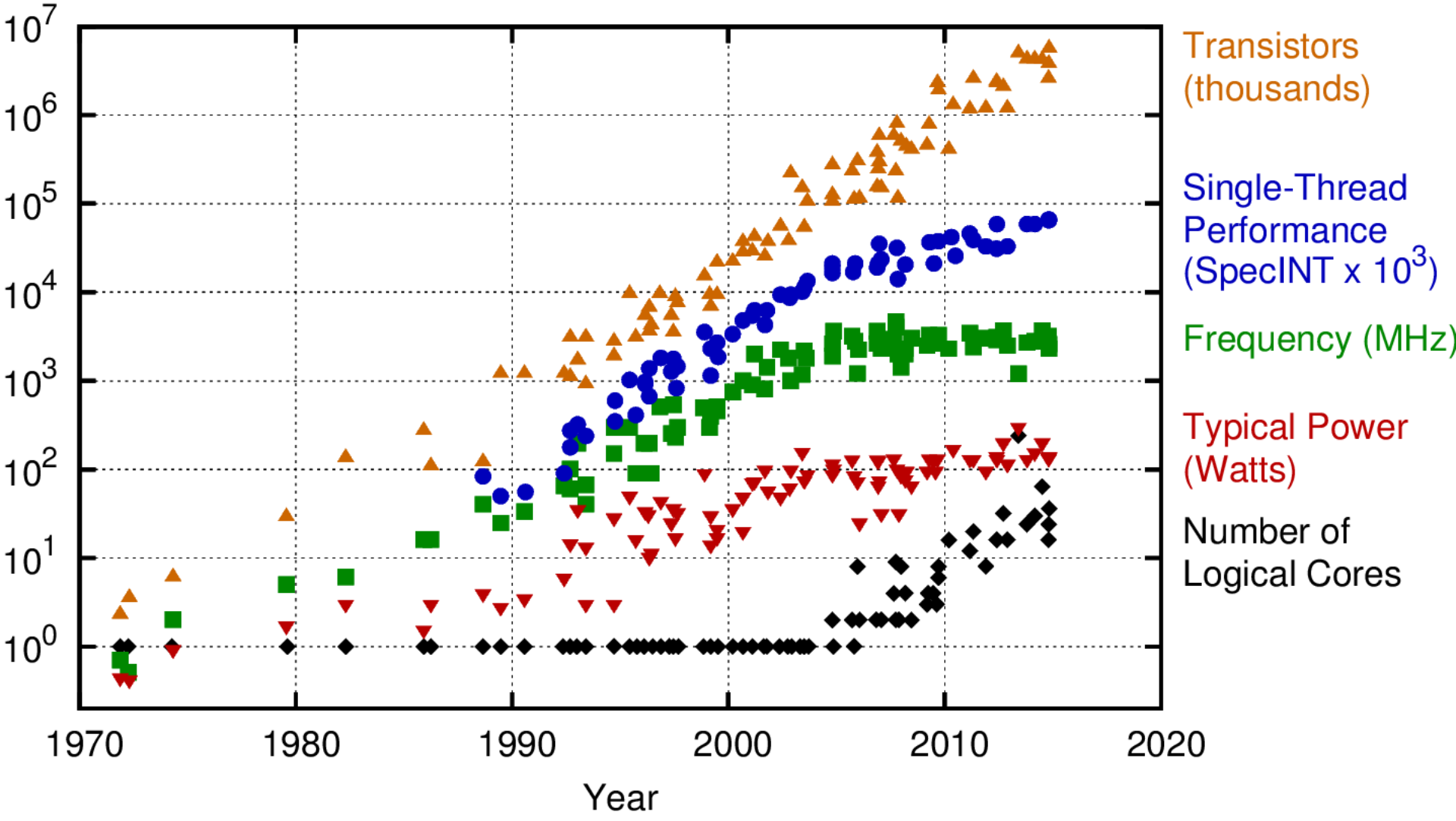
Motivation: Power?

- 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



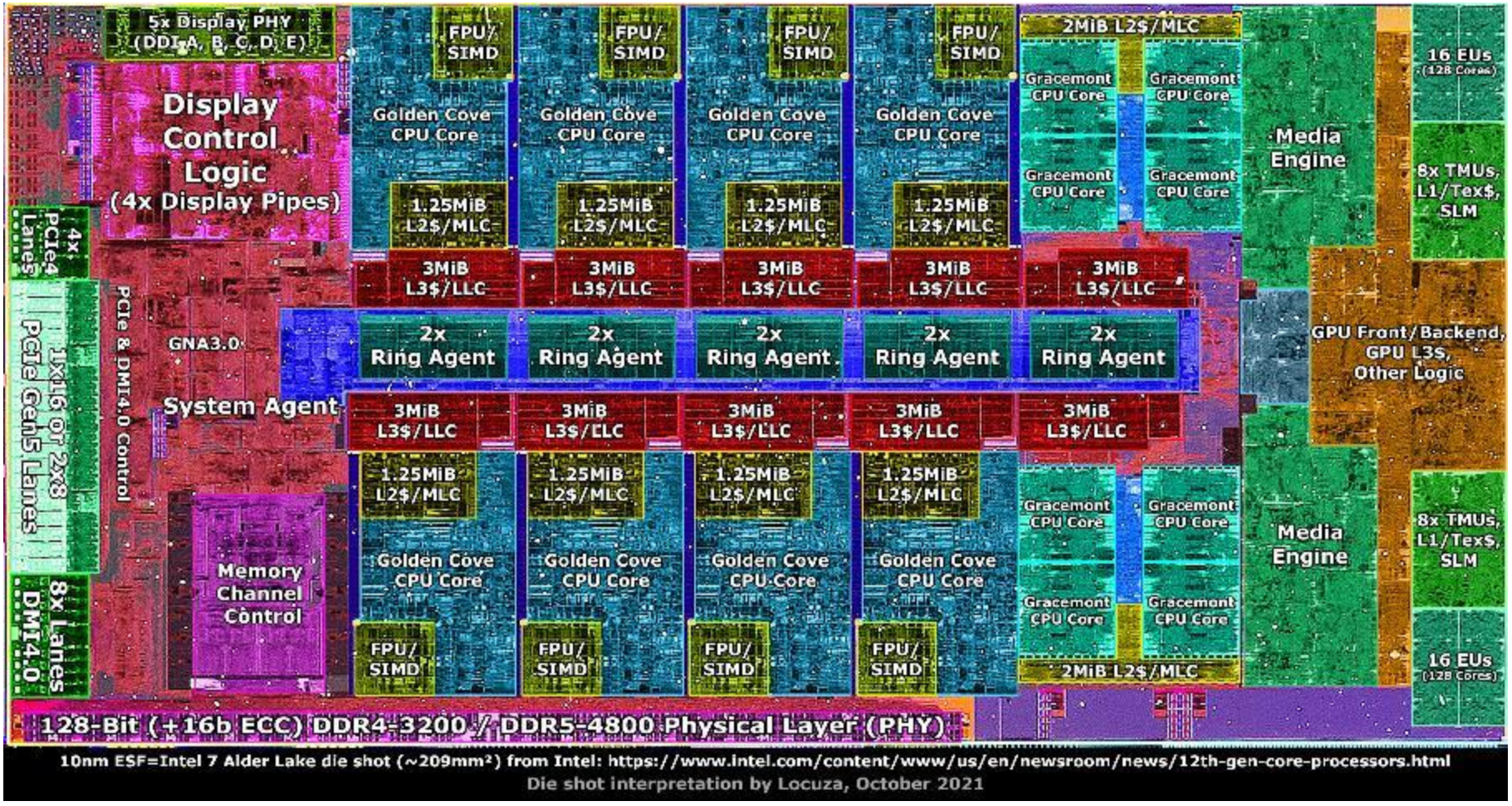
Intel "Rocket Lake"

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!**

Asymmetric Multi-Core Processors



Intel “Alder Lake”

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

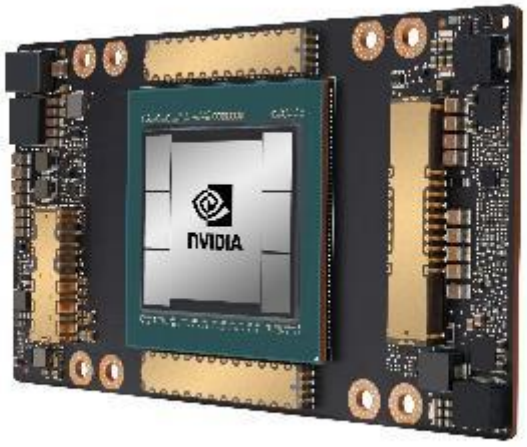
nVIDIA Ampere GPU Compute Architecture

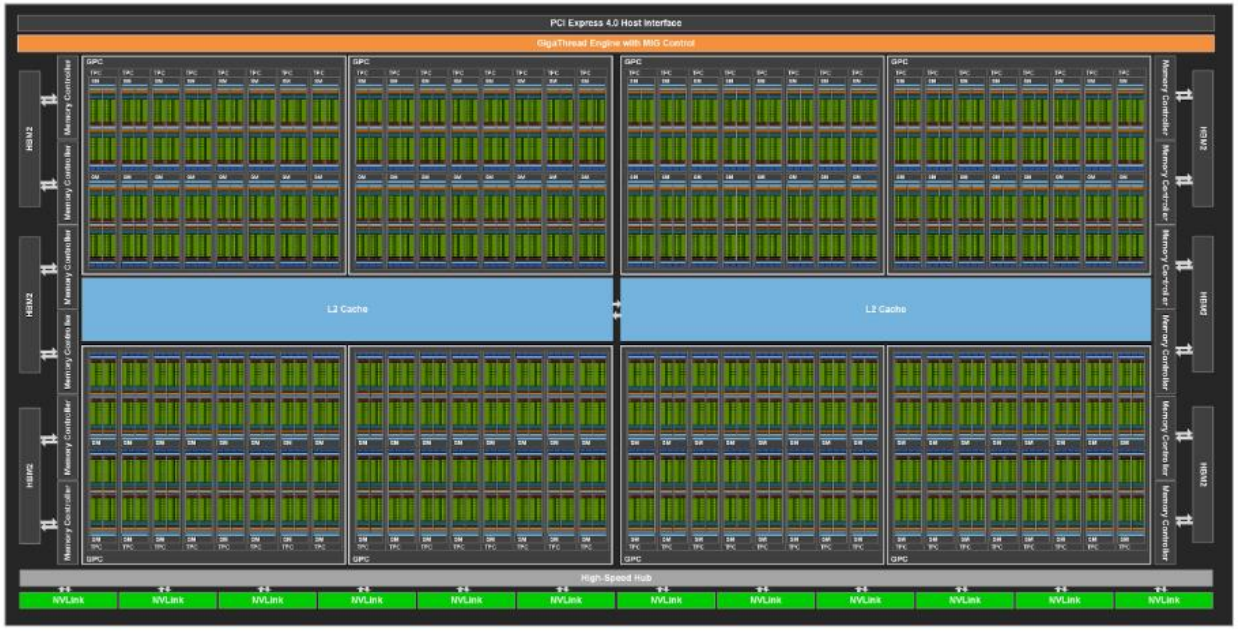
- 54,2 billion transistors
- 6912 “CUDA-cores”
- 432 “Specialized” cores for AI (tensor cores)

- 5120-bit memory bus (HBM2)
- 40 GB memory
- 1555 GB/sec memory bandwidth

- 19.5 TFLOPS single precision performance

- PCI Express 4.0
- NVLink 3





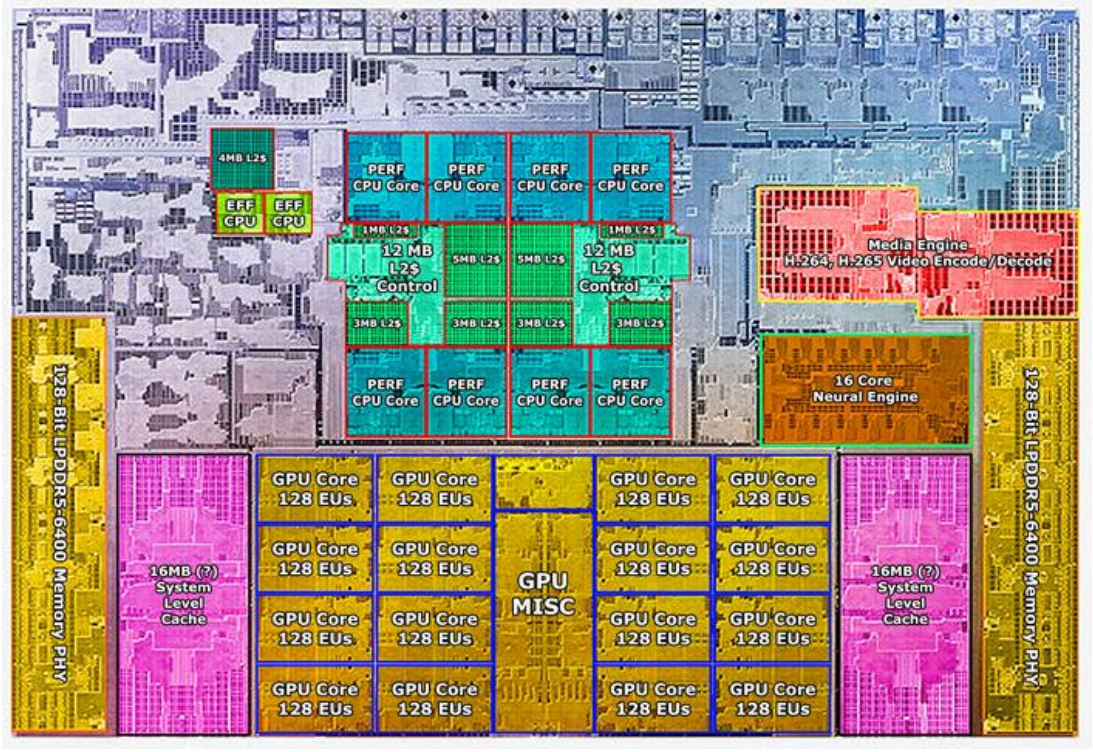


System on a Chip (SoC)

Making the system even more heterogeneous!

SoC – A collection of heterogeneous processors

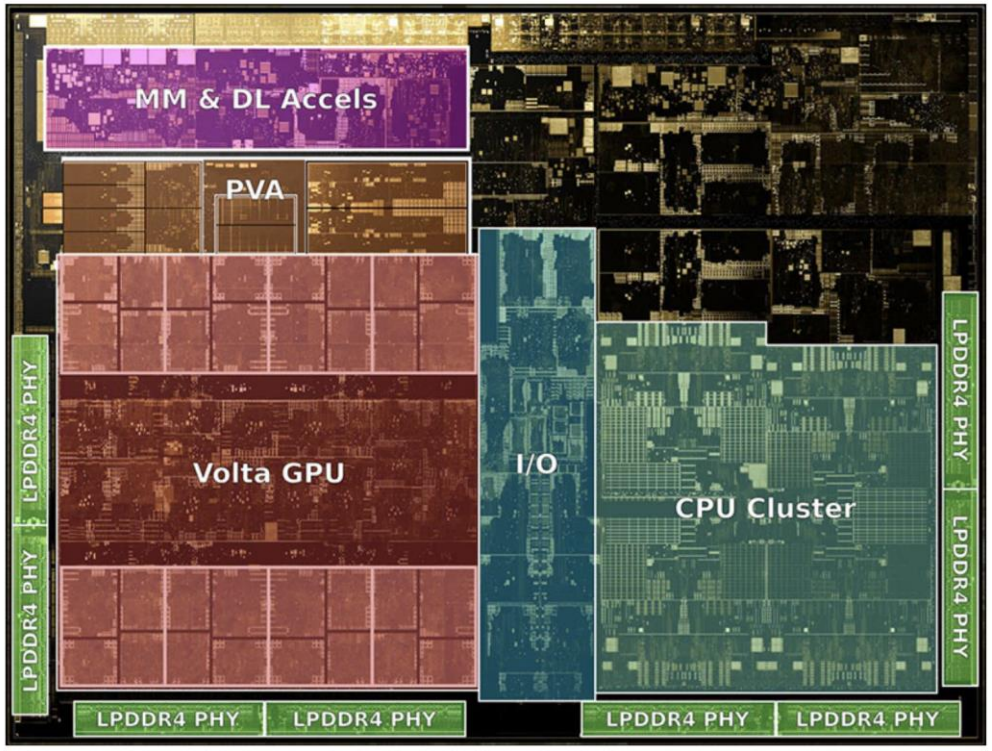
- Heterogeneous systems are more and more common in mainstream computers.
- Was already a trend in smart-phones…



- Apple M1 Pro SoC
  - Two CPU architectures: One for efficiency and one for performance
  - High-performance GPU
  - Cores for offloading ML inference.
  - Accelerators on GPU for ML training.
  - Image signal processor
  - Video encoders and decoders
  - Security processor for encryption
  - …

nVIDIA Tegra Xavier ARM SoC

- A heterogeneous architecture optimized for AI on autonomous devices.
- 8 Nvidia Carmel ARM CPUs
  - Custom architecture
  - 64-bit ARM ISA
  - 128-bit SIMD ISA
- Fully programmable Volta-family Graphics Processing unit
- Several “dedicated co-processors”:
  - 4x 4K Video Decoder
  - 4x 4K Video Encoder
  - Vision Accelerator (PVA)
  - Deep Learning Accelerator (NVDLA)



Jetson AGX Xavier – The platform for IN5050

Development kit from Nvidia with the Tegra Xavier SoC, targeting machine learning and computer vision.



- Octal-core Nvidia Carmel ARMv8.2
- 32 GB 256-bit LPDDR4x (137GB/s)
- 32 GB eMMC
- USB3.1, CSI-2
- Gigabit Ethernet
- 8-lane PCI Express Gen 4
- 512-core Volta GPU
- Ubuntu 18.04 LTS (Linux for Tegra)
- Up to 11 TFLOPS of FP16 performance using Tensor Cores
  - TPD: 30W

The End: Summary

- 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**