IN5050
Programming heterogeneous multi-core processors



Introduction

Håkon Kvale Stensland

January 20th, 2025

[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

Course email:

in5050 @ ifi

- Guest lectures from:
  - Kristoffer Robin Stokke
  - Dino Orucevic
  - Dolphin Interconnect Solutions

Time and place

- **Lectures**:

  Monday **14:15 - 16:00**

  Python (OJD / IFI)

  Wednesday **10:15 – 12:00**

  Shell (OJD / IFI)

- Lectures will give you a basic introduction to the architecture you are going to work with. Including APIs for programming.
- You are expected to learn more detailed use of the architectures on your own.

- Check course webpage for latest details!

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:
    - Computers with discrete Graphics Processing Units (GPUs)
    - Heterogeneous SoC architecture with GPU and ARM CPU cores
    - Multiple computers connected over a PCI Express network
  - … 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: **Discrete Nvidia graphics processing unit**
     - Video encoding – Improve the performance of *video compression using discrete Nvidia GPUs* on standard x86 computers.
  2. Home Exam 2: **Integrated Nvidia graphics processing unit on a heterogeneous SoC with ARM NEON SIMD**.
     - Video encoding – Improve the performance of *video compression* by using a integrated GPU on a heterogeneous SoC with NEON SIMD instructions.
  3. Home Exam 3: **Distributed system scenario**
     - TBD – Use a shared memory API to distribute a workload over computers connected with a PCI Express network.

- You will be working together in groups of two. Try to find a partner over the next couple of weeks

Background and Motivation:

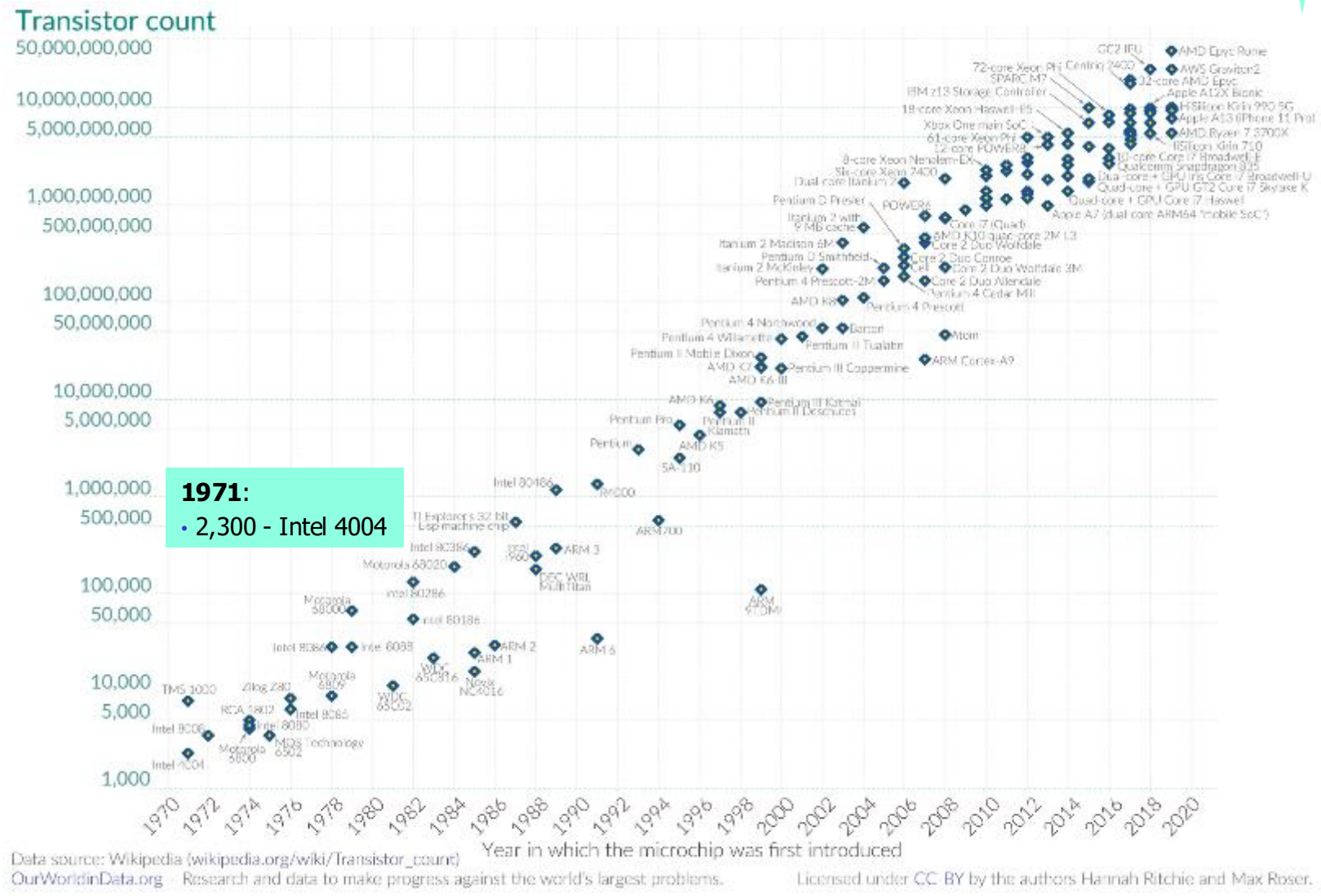
Moore's Law

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

Motivation: Transistors

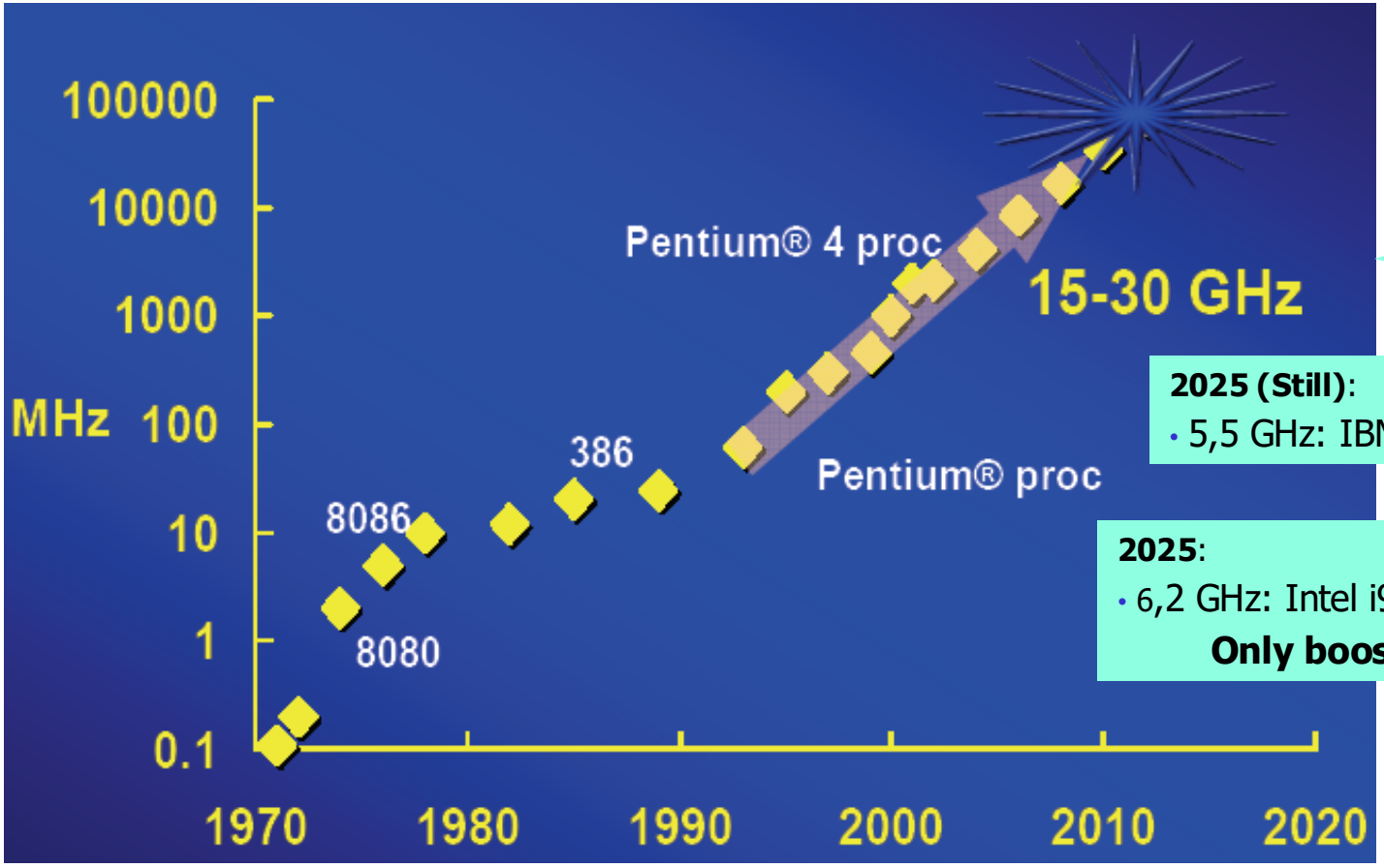
2024:
- 92 billion – Apple M3 Max SoC

- Billion transistors integrated



Motivation: Clock frequency?

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



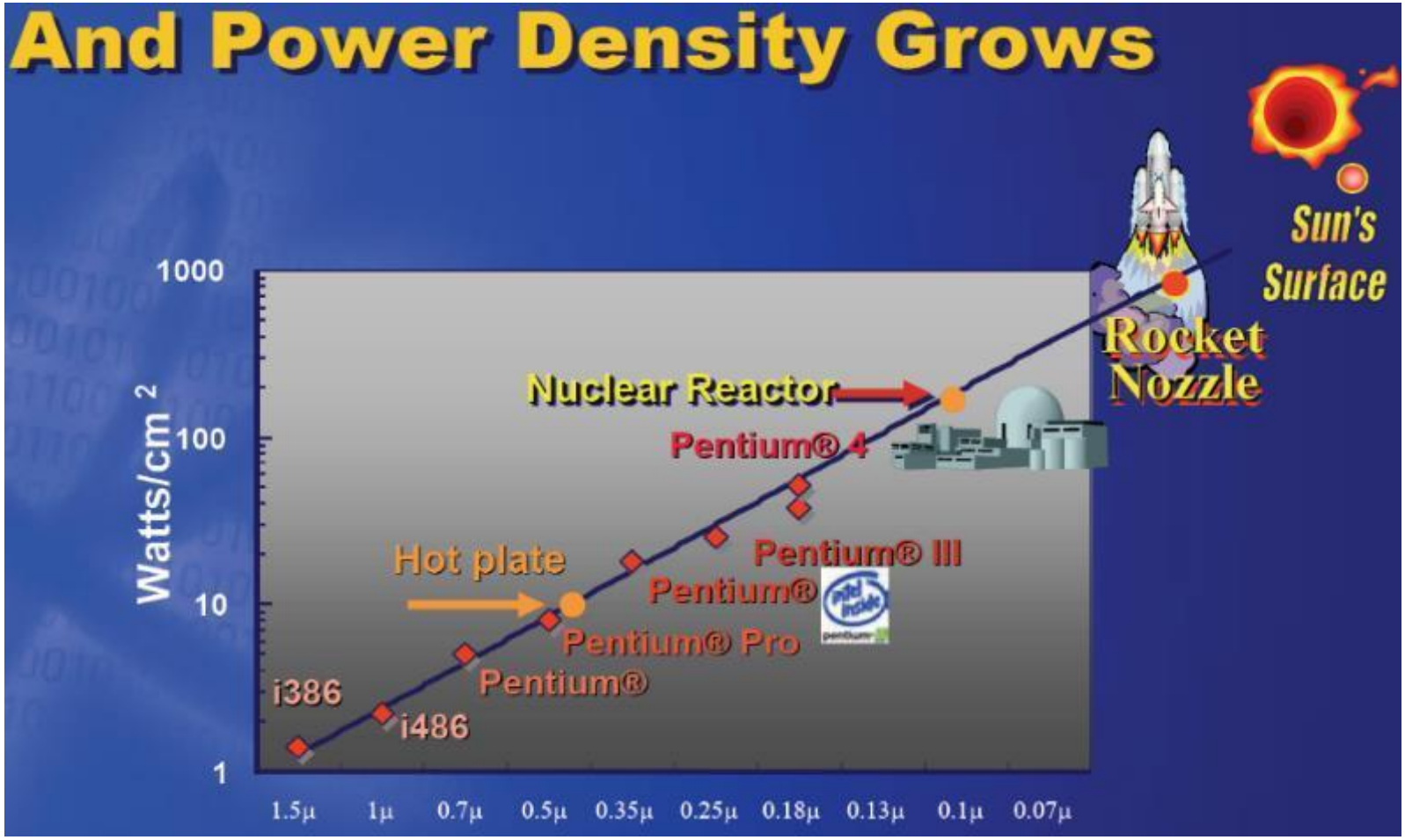
2025 (Still):
- 5,5 GHz: IBM zEC12

2025:
- 6,2 GHz: Intel i9-14900KS

Only boost frequency

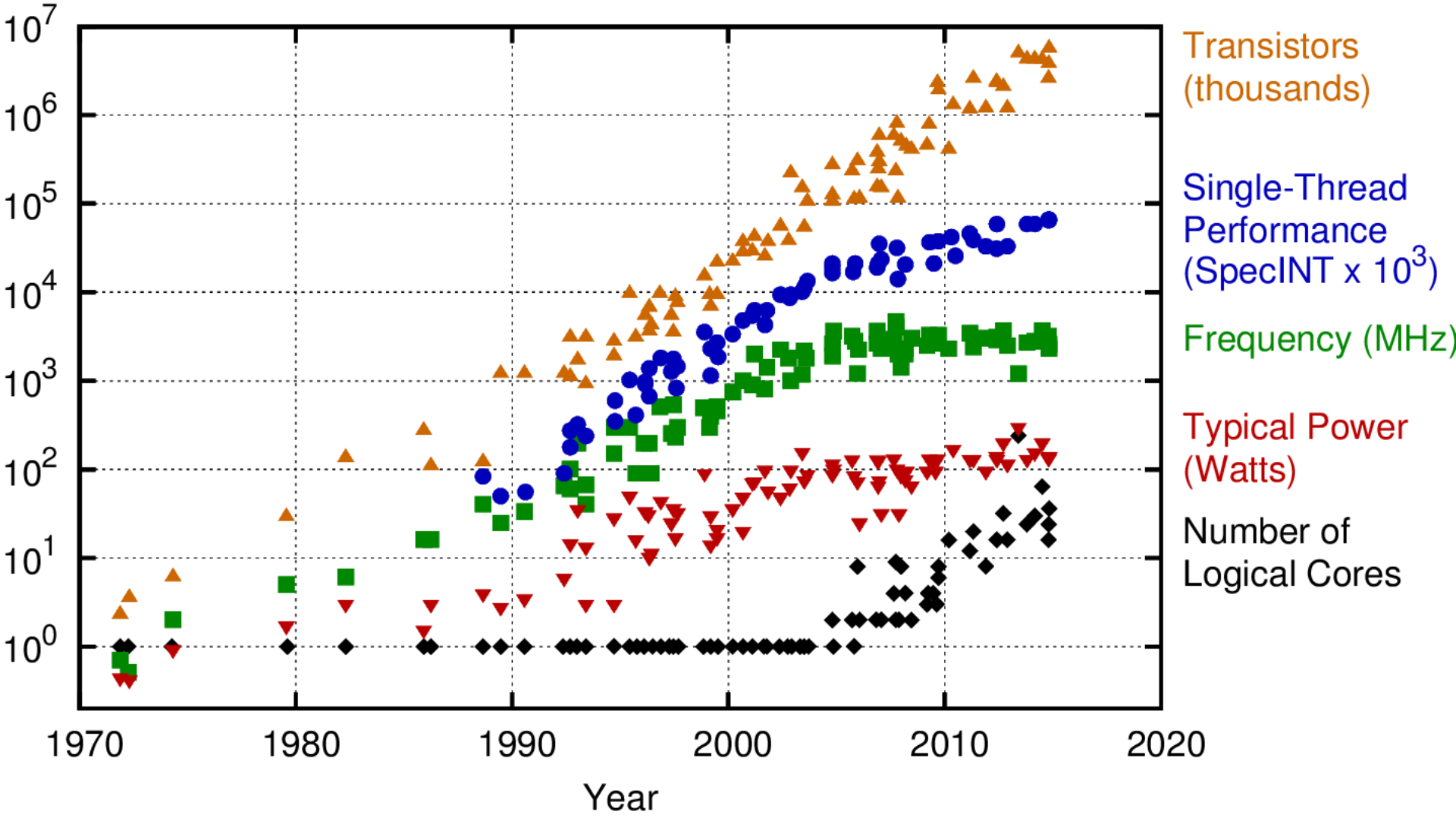
Motivation: Power?

- 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.



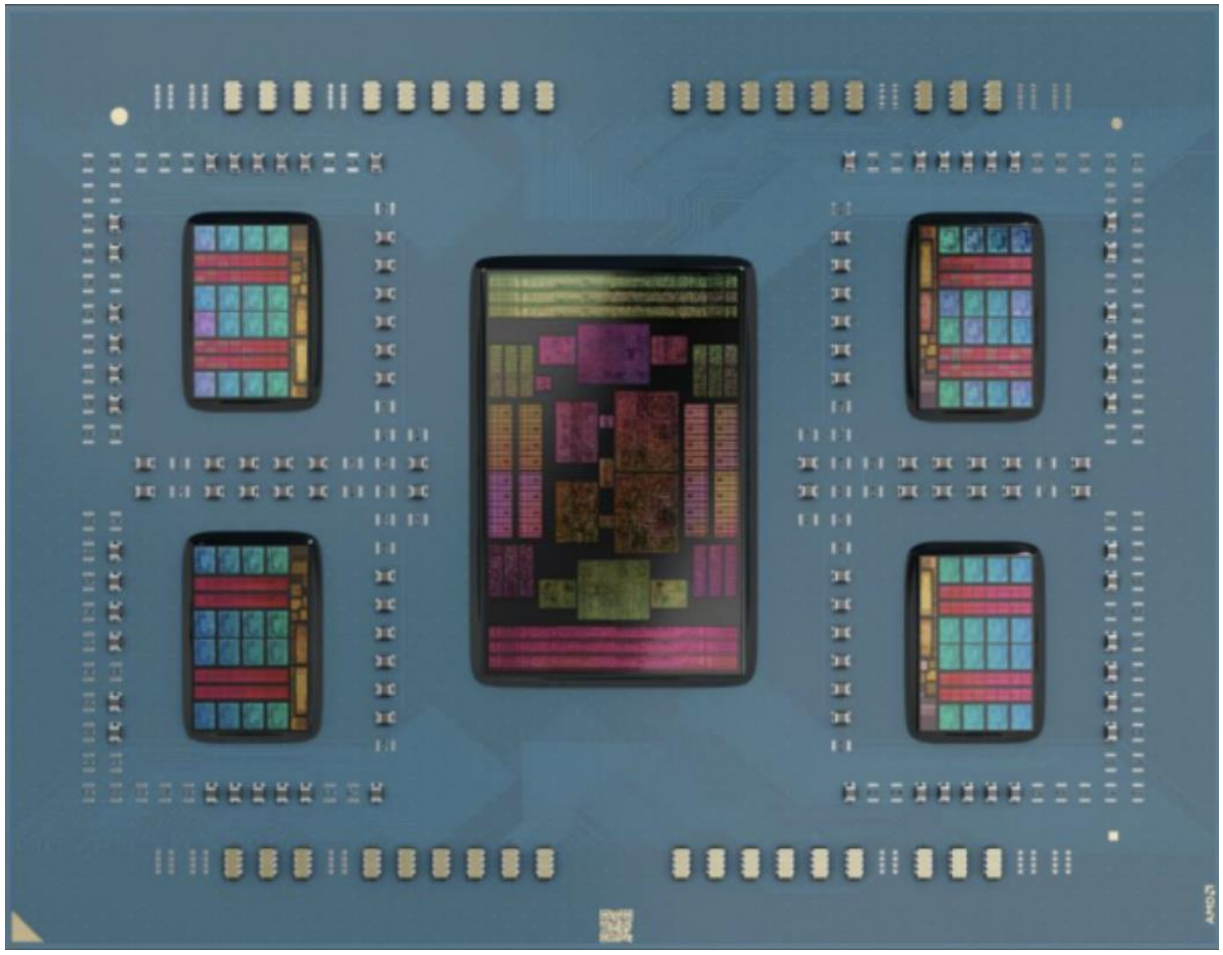

Multicores!

Symmetric Multi-Core Processors



Intel "Rocket Lake" (11th generation Core)

Symmetric Multi-Core Processors



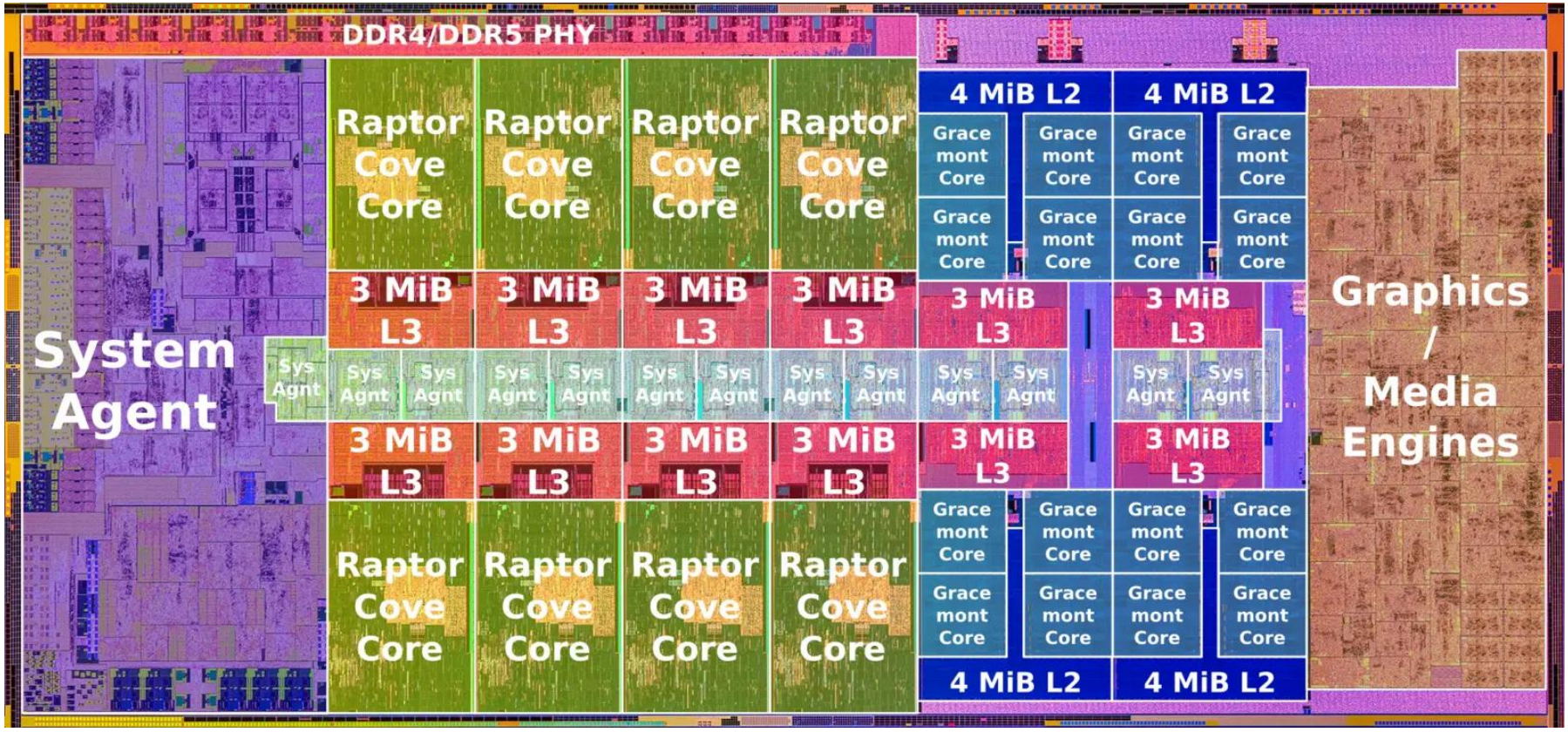
AMD Zen 4 "Genoa" (4th Gen EPYC)

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 "Raptor Lake" (14th generation Core)

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
  - …

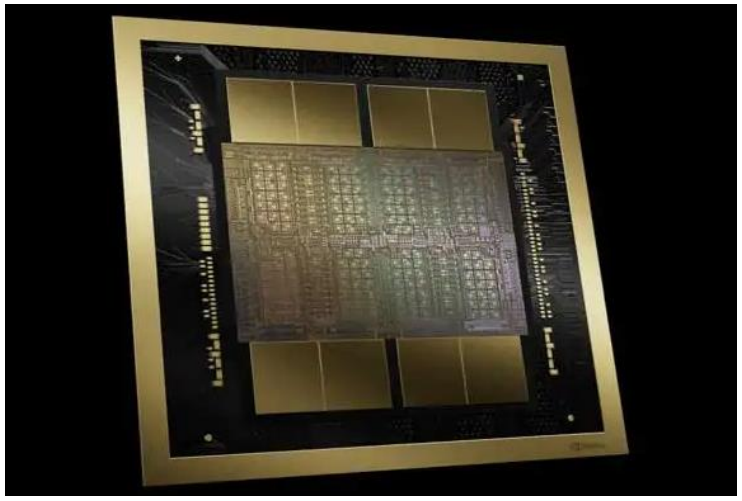
  … of the GPU allows for general, non-graphics related operations to be performed on the GPU

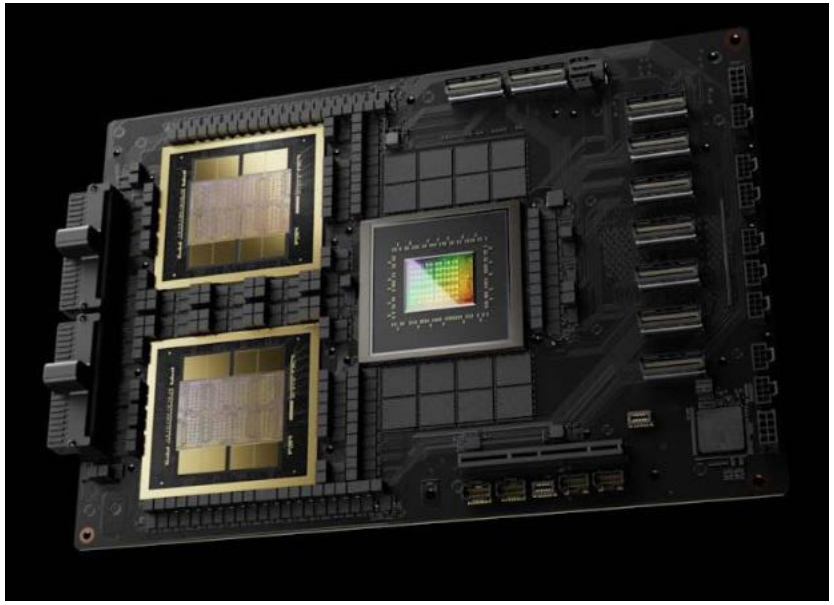
- Generic computing workload is off-loaded from CPU and to GPU

⇒ **More generically:**
Heterogeneous multi-core processing

nVIDIA Blackwell GPU Compute Architecture

- 2 x 104 billion transistors
- 2 x 16896 "CUDA-cores"
- 2 x 528 "Specialized" cores for AI (Tensor Cores)
- 2 x 4096-bit memory bus (HBM3)
- 2 x 50 MB L2 Cache
- 2 x 96 GB memory
- 8 TB/sec memory bandwidth
- 124 TFLOPS single precision performance
- PCI Express Gen 6
- NVLink Gen 5







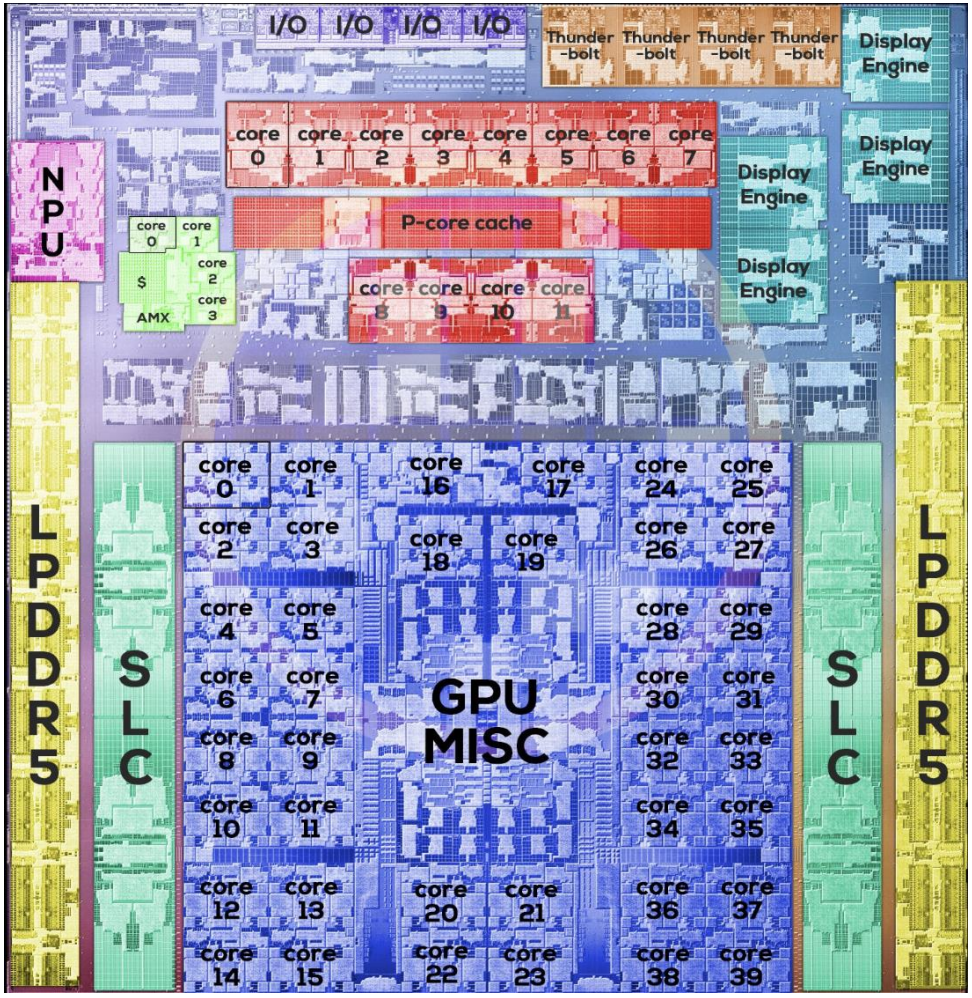
System on a Chip (SoC)

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…

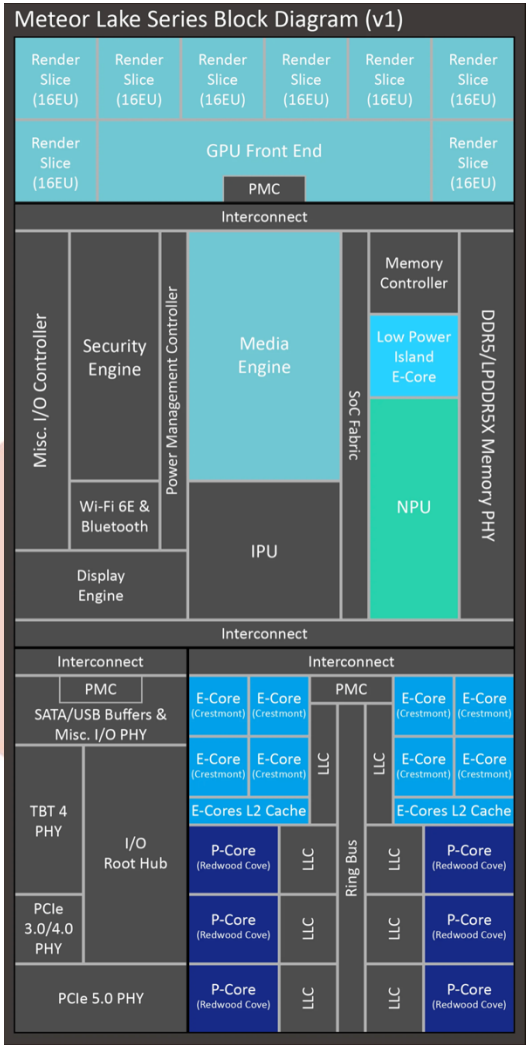
- Apple M3 Max SoC
  - 92 billion transistors on 3nm
  - Two CPU architectures:
    - 4 efficiency cores
    - 12 performance cores
  - High-performance GPU
  - Accelerator for ML (NPU).
  - Image signal processor
  - Video encoders and decoders
  - Security processor for encryption
  - High performance I/O



SoC's on mainstream x86 systems

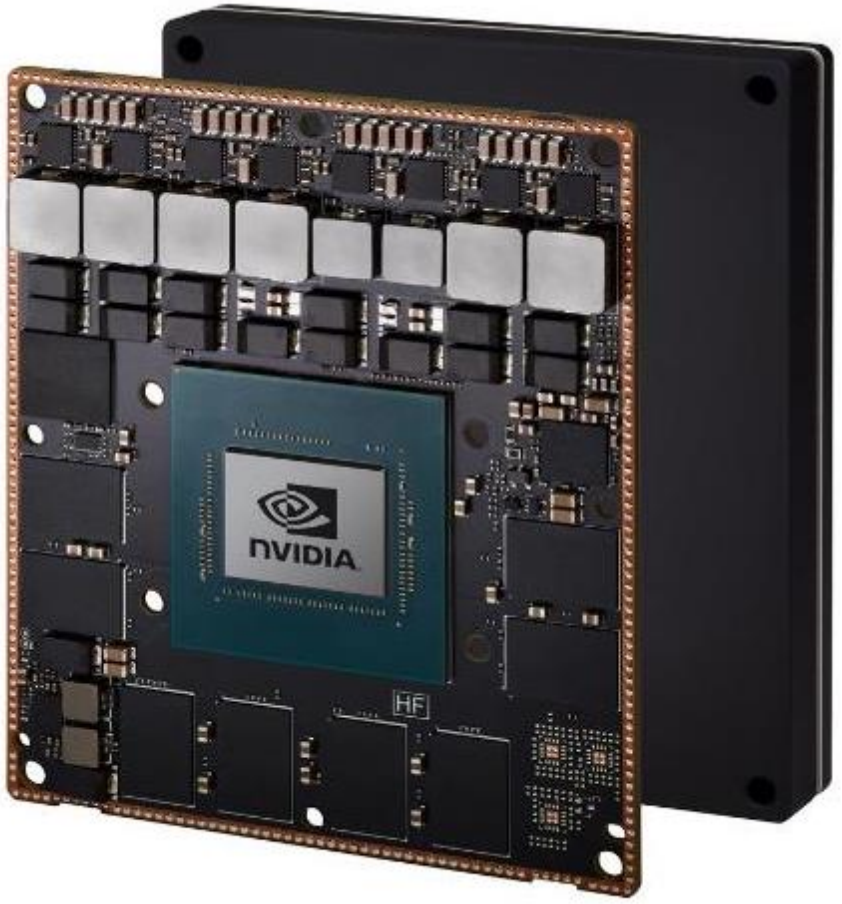
- SoC's are finally also showing up on x86 systems.
- First mainstream x86 SoC with fully modular chiplets.
- Intel Meteor Lake (1st Gen Core Ultra)
  - Independent chiplets for GPU, CPU, IO and "SoC"
  - *CPU:* Heterogenous with two architectures.
  - *IO:* PCIe, SATA, Tunderbolt
  - *SoC:* NPU, Media processors, CPU-cores, ISP, display, security and some IO
  - *GPU:* Processing part of the graphics





Jetson AGX Xavier – The platform for IN5050

Development kit from Nvidia with the Tegra Xavier SoC.



- 8-core Nvidia Carmel CPU (ARMv8.2)
  - Custom 64-bit architecture
  - 128-bit SIMD ISA
- 512-core Volta GPU (GV10B)
- 32 GB 256-bit LPDDR4x (137GB/s)
- Gigabit Ethernet
- 8-lane PCI Express Gen 4
- Jetson Linux 35.4.1 (Ubuntu 20.04 LTS)
  - Linux Kernel 5.10
- 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**