GPUs and Heterogeneous Systems
(programming models and architectures)

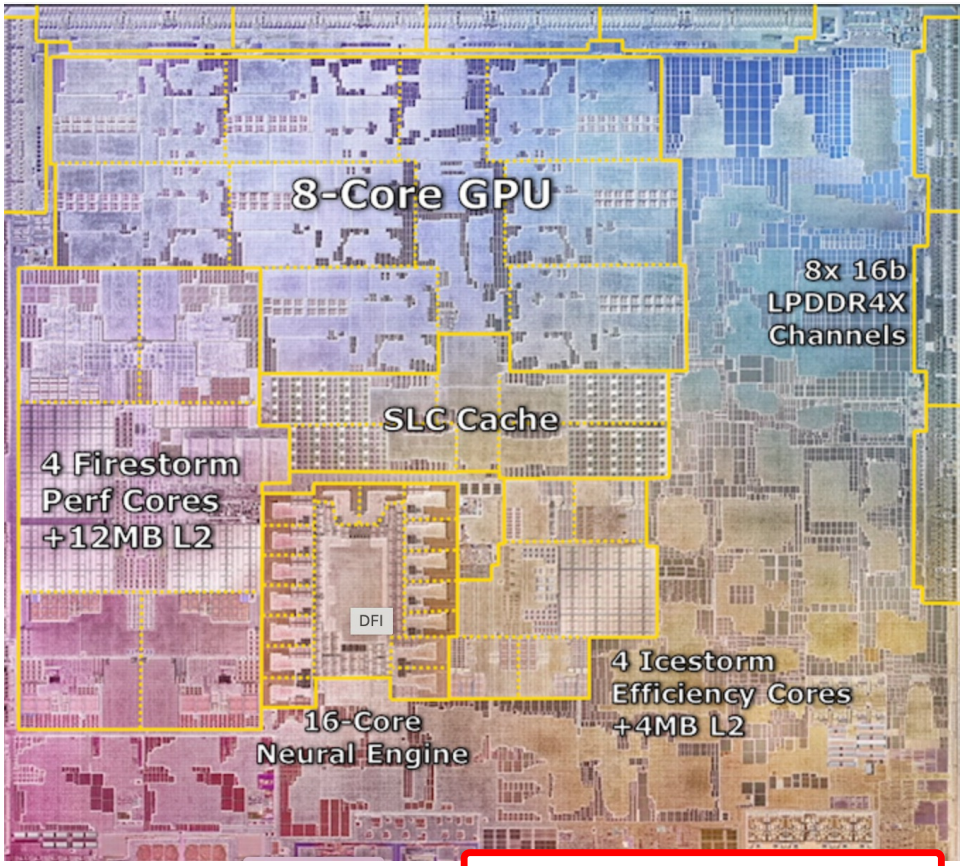
Heterogeneous systems:

motivations and challenges

Which are the heterogeneous systems in this room?

Mobile phones

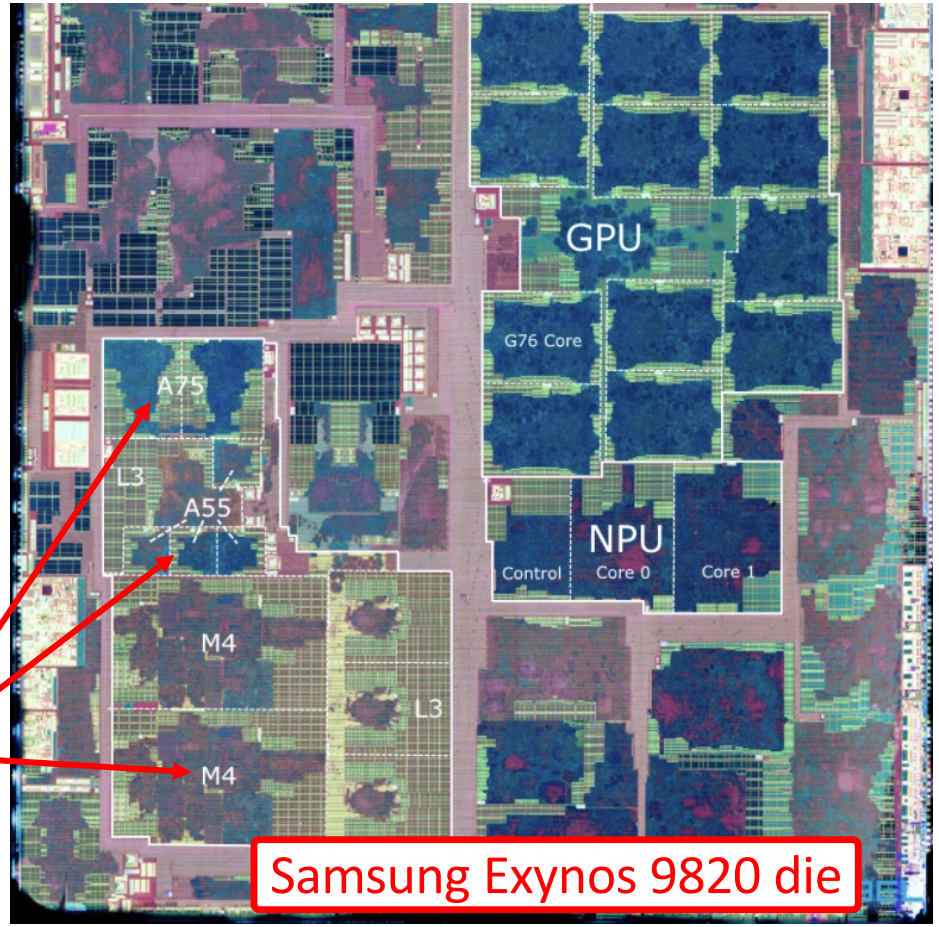
Iphone 12



Apple A14 die

Mobile phones

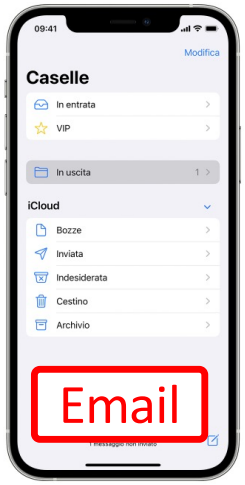
Samsung S10



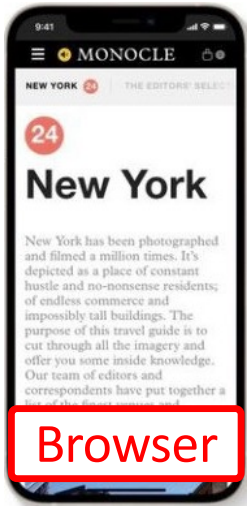
3 CPU clusters

Samsung Exynos 9820 die

Which kind of applications do we run on mobile phones?



Email

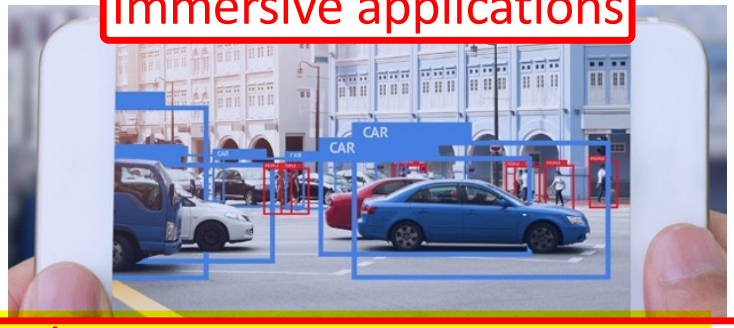


Browser

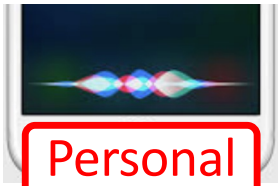
Image/video editing


Videogames

Social networks



Immersive applications


Personal assistant

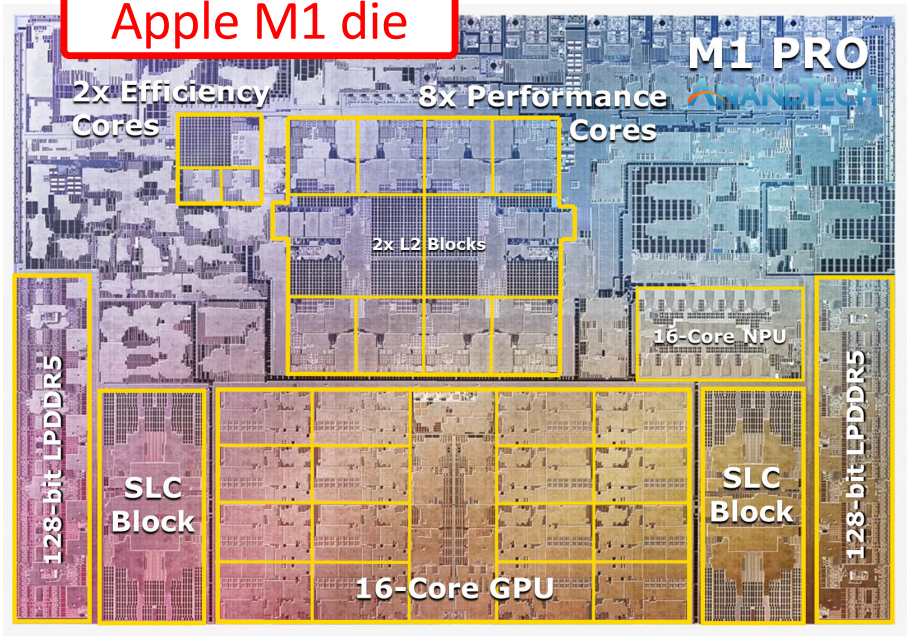
…and any others…

- Mixed high/low performance demanding workload
- A lot of video/image processing and artificial intelligence (AI)

Laptop computers

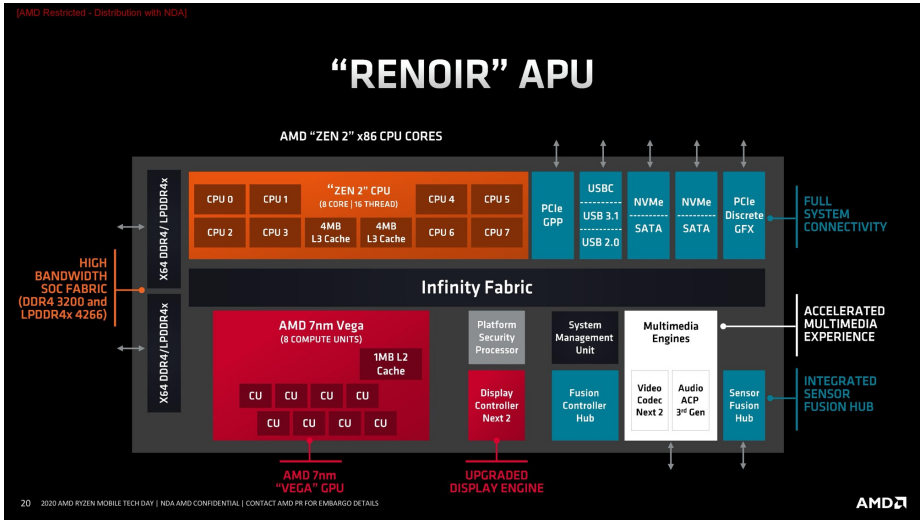
MacBook Pro

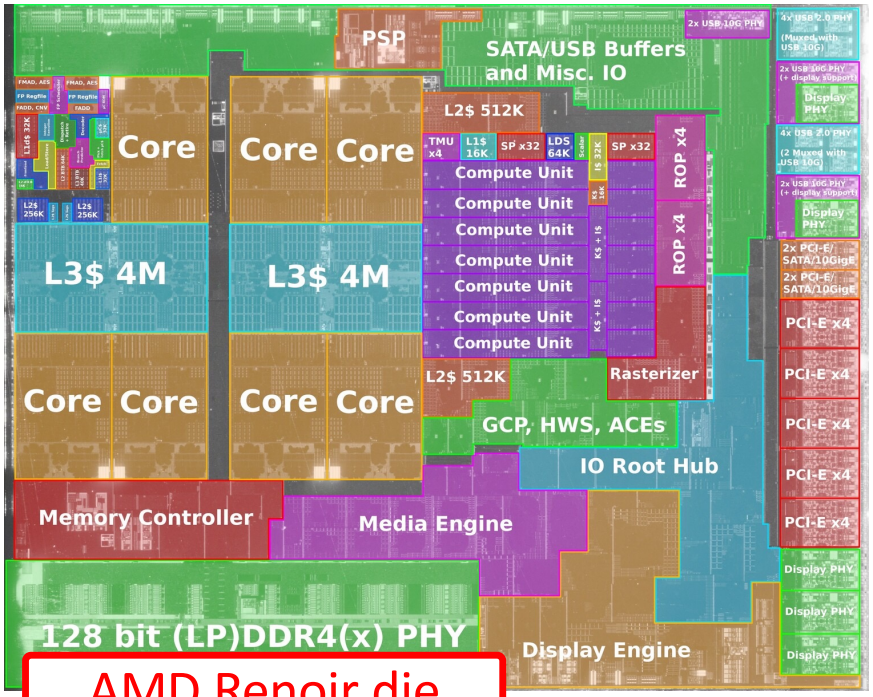
Apple M1 die



Laptop computers

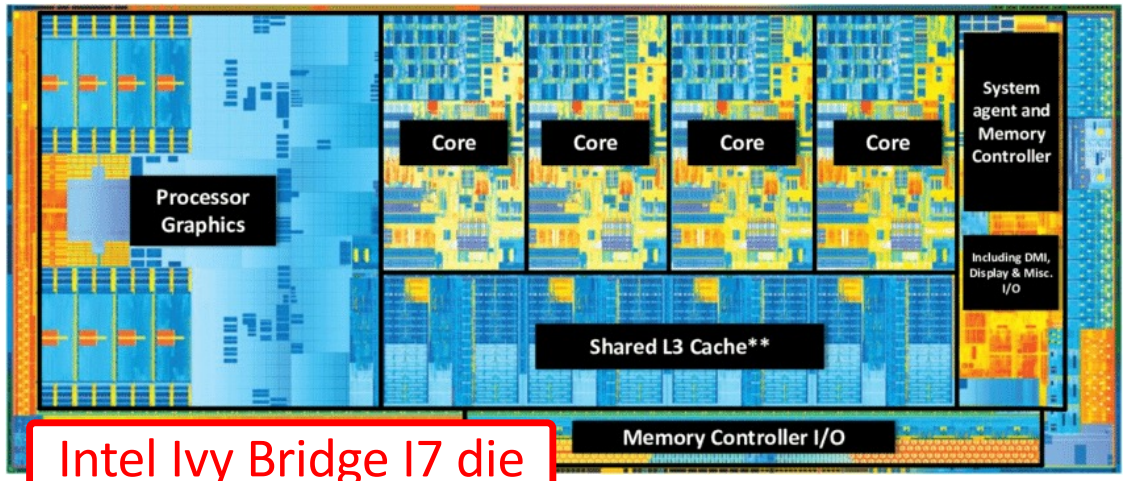
Lenovo laptop





AMD Renoir die

Desktop computers

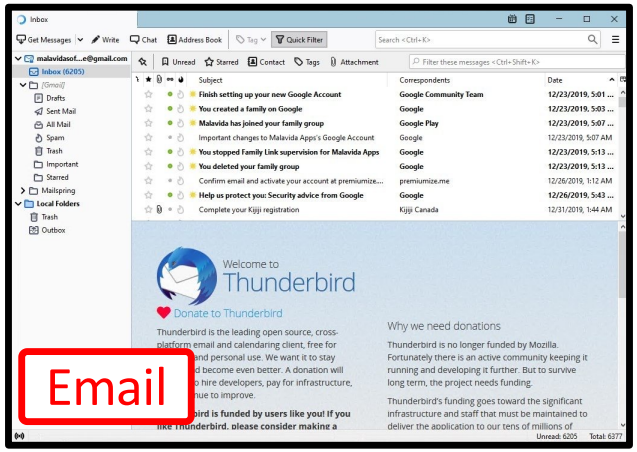


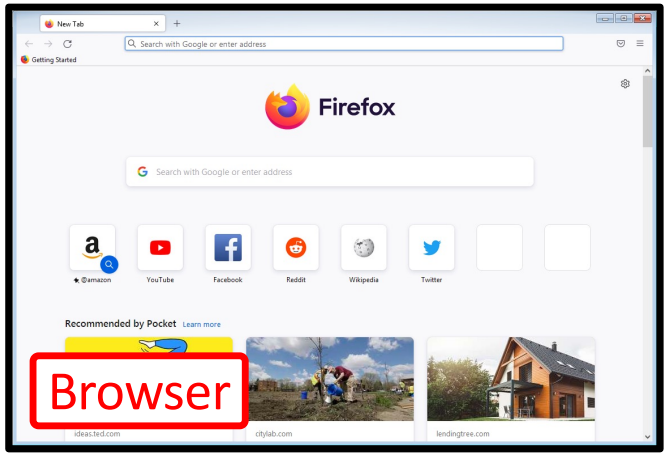
Intel Ivy Bridge I7 die

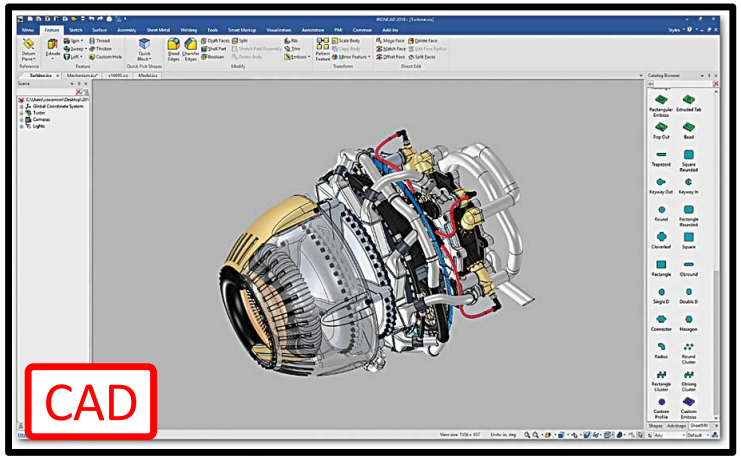
+

NVIDIA GPU

Which kind of applications do we run on laptops and desktop machines?



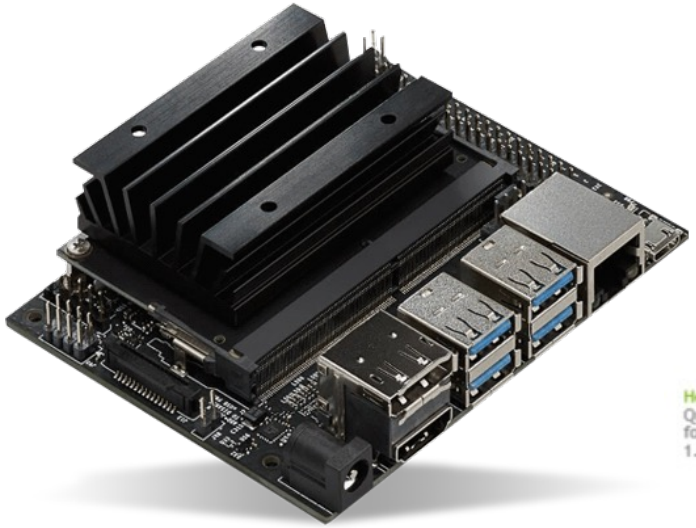




…and too many others…

- Mixed high/low performance demanding workload
- High data processing

Embedded and edge devices



NVIDIA Jetson Nano

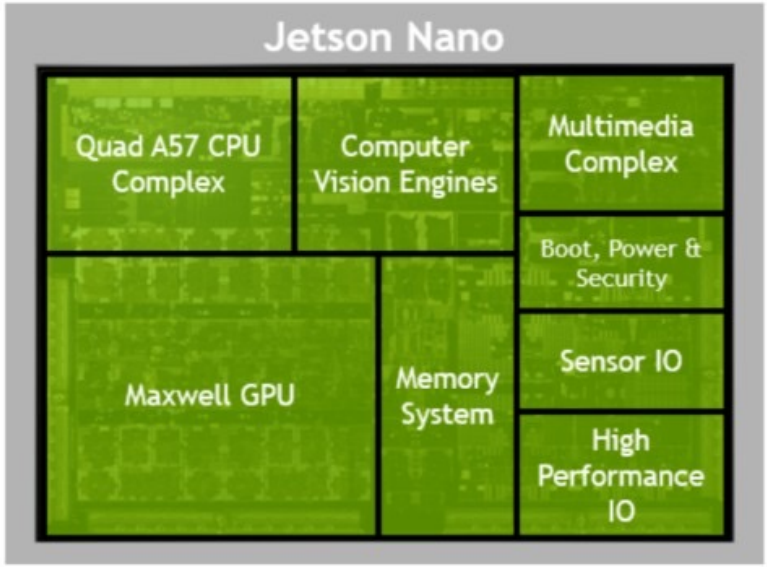
JETSON NANO

Low Cost AI Computer Module

Heterogeneous CPU Complex
Quad Cores A57 with 2MB L2
for multi-threaded operation
1.43Ghz

Maxwell Tensor Core GPU
128 CUDA Tensor Cores
512 CUDA GFLOPS (FP16)

Memory
4GB 64-Bit LPDDR4
Bandwidth 25.6 GB/s
16GB eMMC



Computer Vision Engines
ISP
Video Image Compositor

Multimedia Engines
Encode 4k, 4x1080p, 8x720p up to 120fps
Decode 4k60, 2x4k, 8x1080p, 16x720p up to 240fps
JPEG encode & decode 600 MP/s
H.264, H.265, VP9, VP8, VC-1, MPEG-2
HDMI, DP and eDP Display support

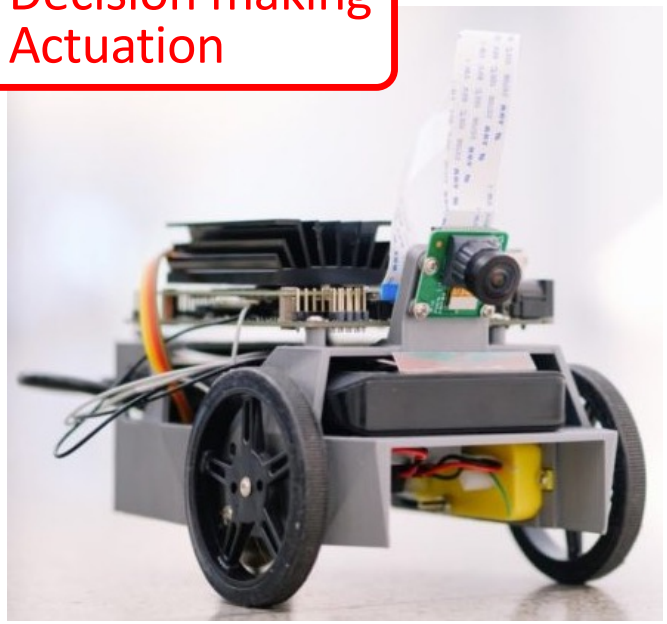
Boot, Power & Security
Boot and power management
ARM TrustZone Secure

Industry Standard IO
GPIO, I²C, I²S, SDIO, SPI, UART
Support up to 12 CSI @1.5Gbps

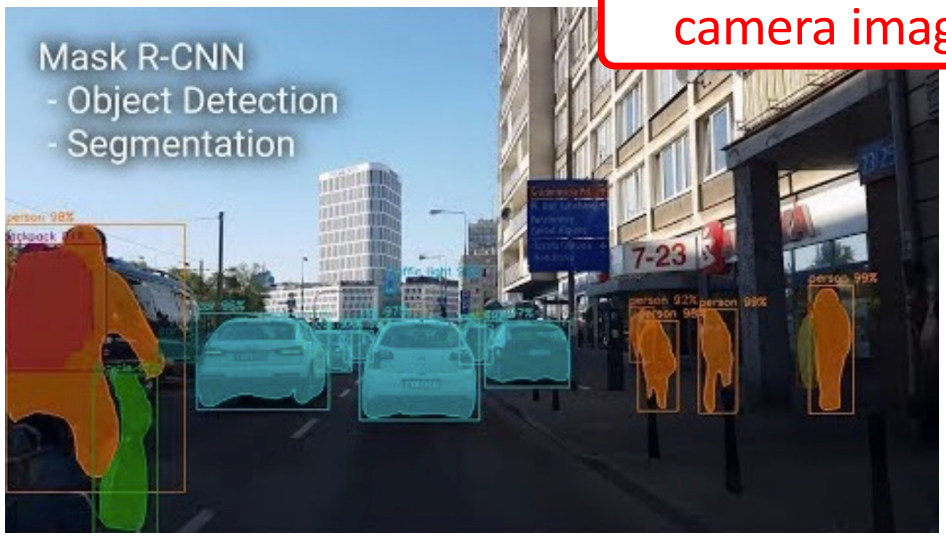
Industry Standard High-Speed IO
PCIe Gen2 rootport x1 | x2 | x4
RGMII Ethernet
USB 3.0 and 2.0
USB 3.0 Gen2 Host and Device

Which kind of applications do we run on edge devices?

Robot control:
- Perception
- Decision making
- Actuation



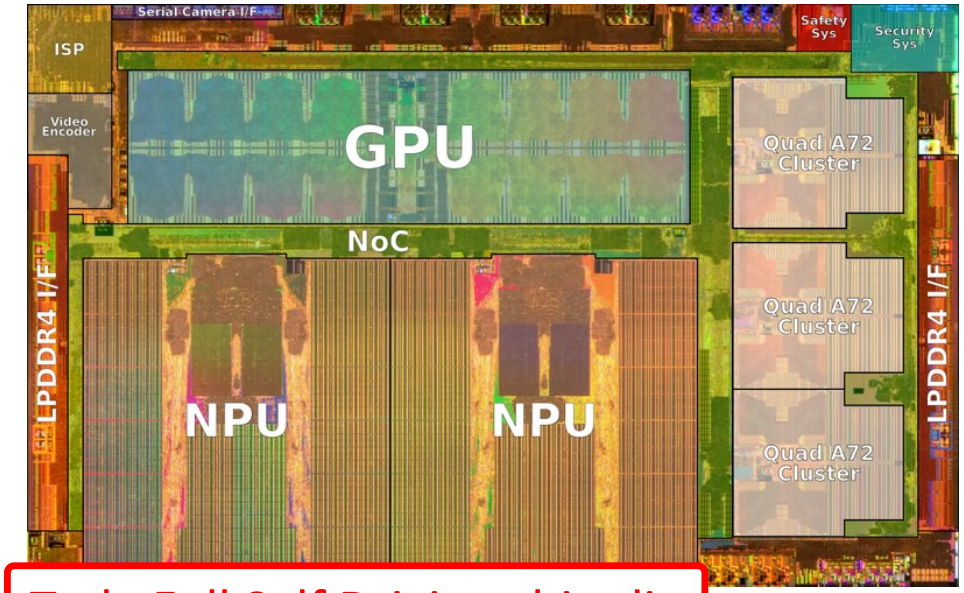
Perception requires realtime AI on camera images



…and many other embedded (AI) applications…

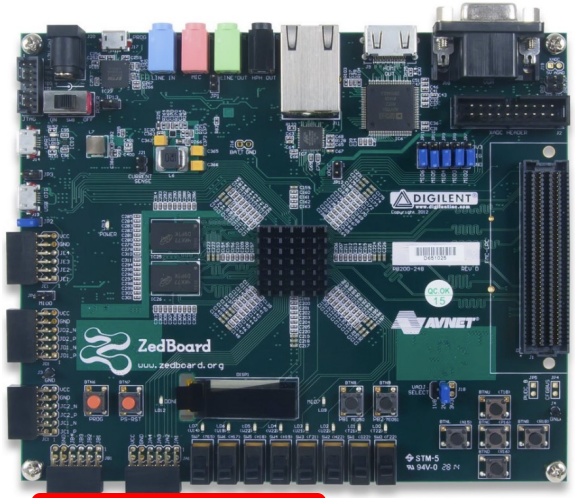
Further embedded and edge devices and applications

Tesla Full Self Driving



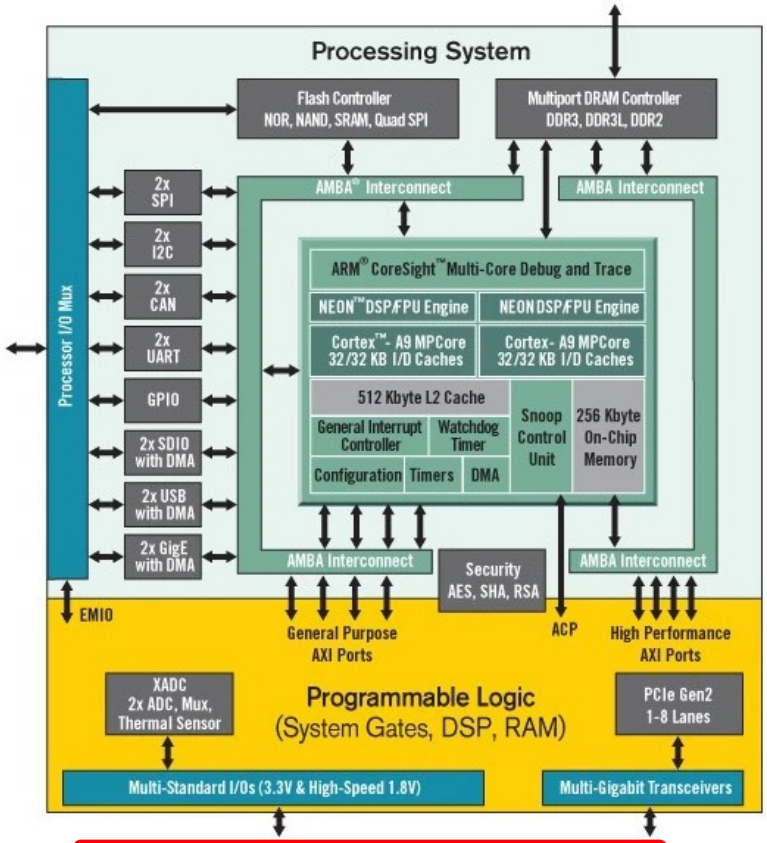
Tesla Full Self Driving chip die

Further embedded and edge devices and applications



Xilinx Zynq

Applications: automotive autonomous driving, drones, robots, industrial IoT…
Main tasks: motor control and embedded vision



Xilinx Zynq architecture

Top 500 supercomputers

www.top500.org

November 2021

1	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, **Fujitsu** RIKEN Center for Computational Science Japan
2	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, **IBM** DOE/SC/Oak Ridge National Laboratory United States
3	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, **IBM / NVIDIA / Mellanox** DOE/NNSA/LLNL United States
4	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, **NRCPC** National Supercomputing Center in Wuxi China
5	**Perlmutter** - HPE Cray EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, **HPE** DOE/SC/LBNL/NERSC United States
6	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, **Nvidia** NVIDIA Corporation United States
7	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, **NUDT** National Super Computer Center in Guangzhou China
8	**JUWELS Booster Module** - Bull Sequana XH2000 , AMD EPYC 7402 24C 2.8GHz, NVIDIA A100, Mellanox HDR InfiniBand/ParTec ParaStation ClusterSuite, **Atos** Forschungszentrum Juelich (FZJ) Germany
9	**HPC5** - PowerEdge C4140, Xeon Gold 6252 24C 2.1GHz, NVIDIA Tesla V100, Mellanox HDR Infiniband, **DELL EMC** Eni S.p.A. Italy
10	**Voyager-EUS2** - ND96amsr_A100_v4, AMD EPYC 7V12 48C 2.45GHz, NVIDIA A100 80GB, Mellanox HDR Infiniband, **Microsoft Azure** Azure East US 2 United States

Top 500 supercomputers

www.top500.org

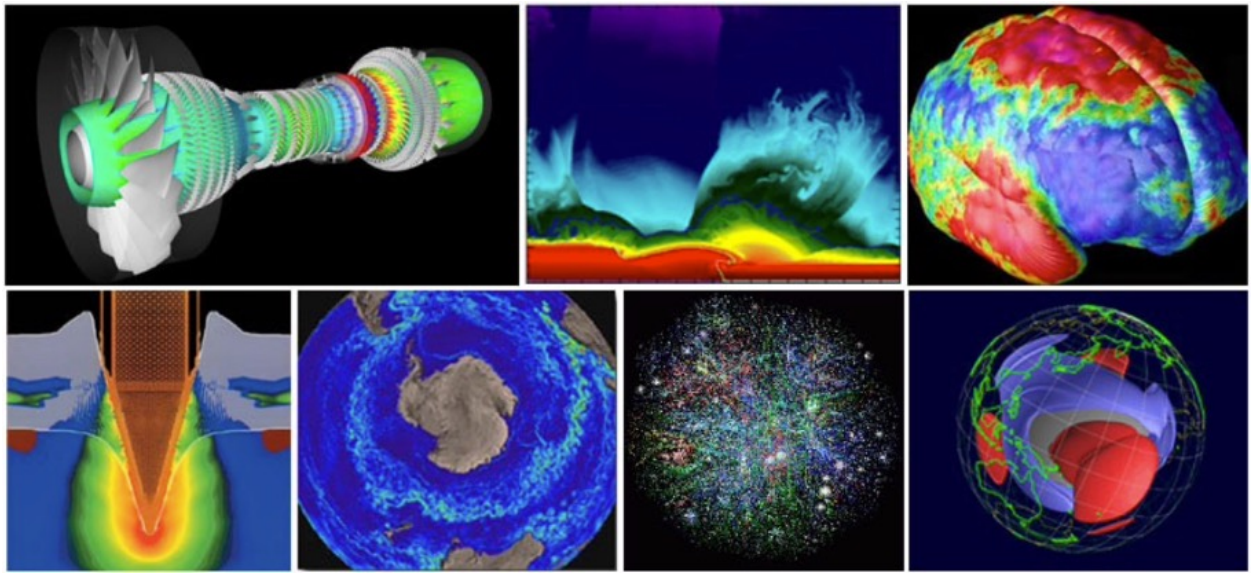
November 2023

1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States
2	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States
3	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Microsoft Azure United States
4	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan
5	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland
6	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, EVIDEN EuroHPC/CINECA Italy
7	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States
8	**MareNostrum 5 ACC** - BullSequana XH3000, Xeon Platinum 8460Y+ 40C 2.3GHz, NVIDIA H100 64GB, Infiniband NDR200, EVIDEN EuroHPC/BSC Spain
9	**Eos NVIDIA DGX SuperPOD** - NVIDIA DGX H100, Xeon Platinum 8480C 56C 3.8GHz, NVIDIA H100, Infiniband NDR400, Nvidia NVIDIA Corporation United States
10	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States

Which kind of applications do we run on supercomputers?

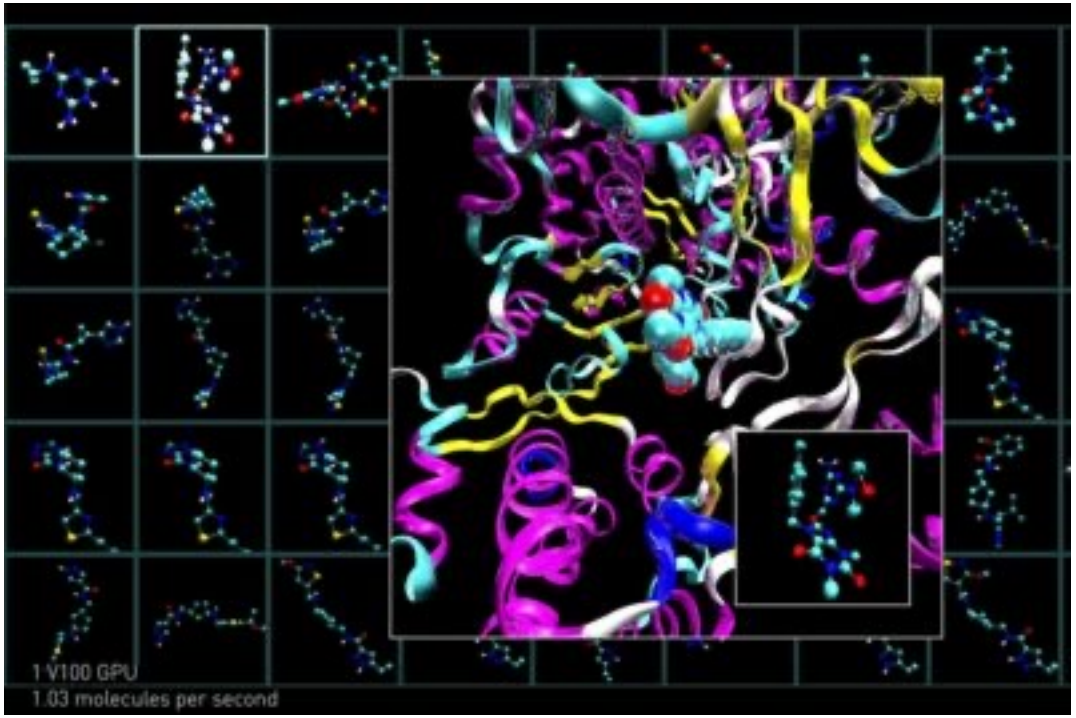
Computational science applications

- Atmosphere, Earth, Environment
- Physics - applied, nuclear, particle, condensed matter, high pressure, fusion, photonics
- Bioscience, Biotechnology, Genetics
- Chemistry, Molecular Sciences
- Geology, Seismology
- Mechanical Engineering - from prosthetics to spacecraft
- Electrical Engineering, Circuit Design, Microelectronics
- Computer Science, Mathematics
- Defense, Weapons



GPUs accelerating COVID-19 research

Molecular Docking for finding a cure for COVID-19

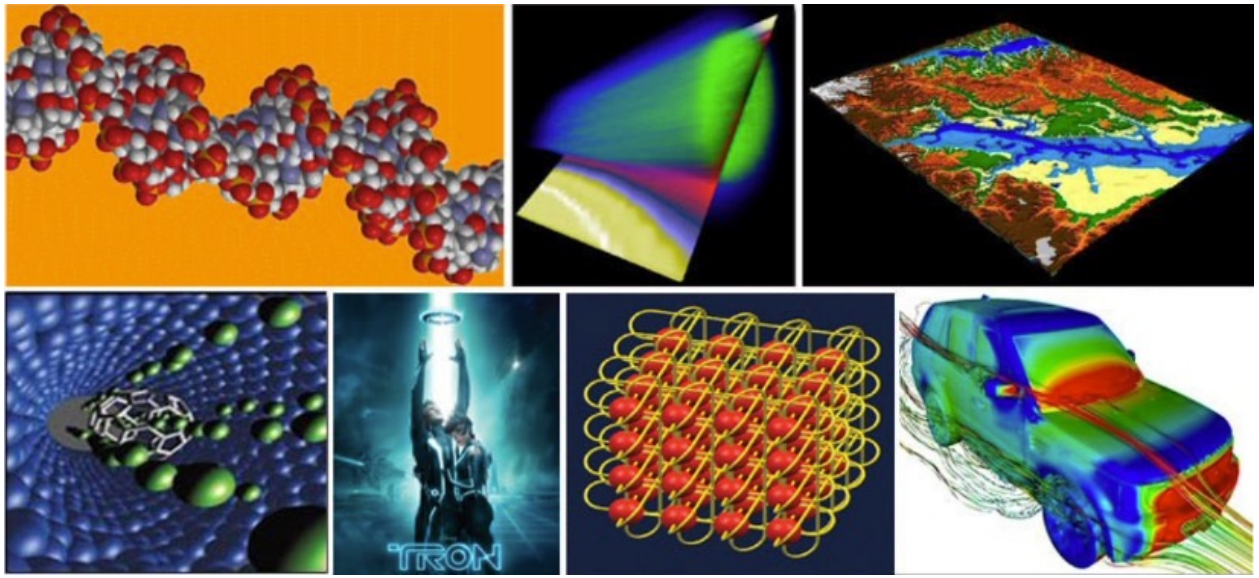


The search requires the screening of billions of drug candidates and identifying the right chemical structure that will most favorably bind and interfere with the virus.
However, calculating the binding potential is computationally intensive, taking years to screen a billion compounds on a CPU system.
The GPU accelerates this process.

https://blogs.nvidia.com/blog/2020/09/28/drug-discovery-covid-19/

Further industrial/commercial High-Performance Computing (HPC) applications

- Big data, databases and data mining
- Oil exploration
- Web search engines
- Medial imaging and diagnosis
- Pharmaceutical design
- Financial and economic models
- Advanced graphics and virtual reality (entertainment industry)
- Networked video and multimedia technologies



https://www.nvidia.com/en-us/gpu-accelerated-applications/

Evolution of computing system architectures

Need for performance!

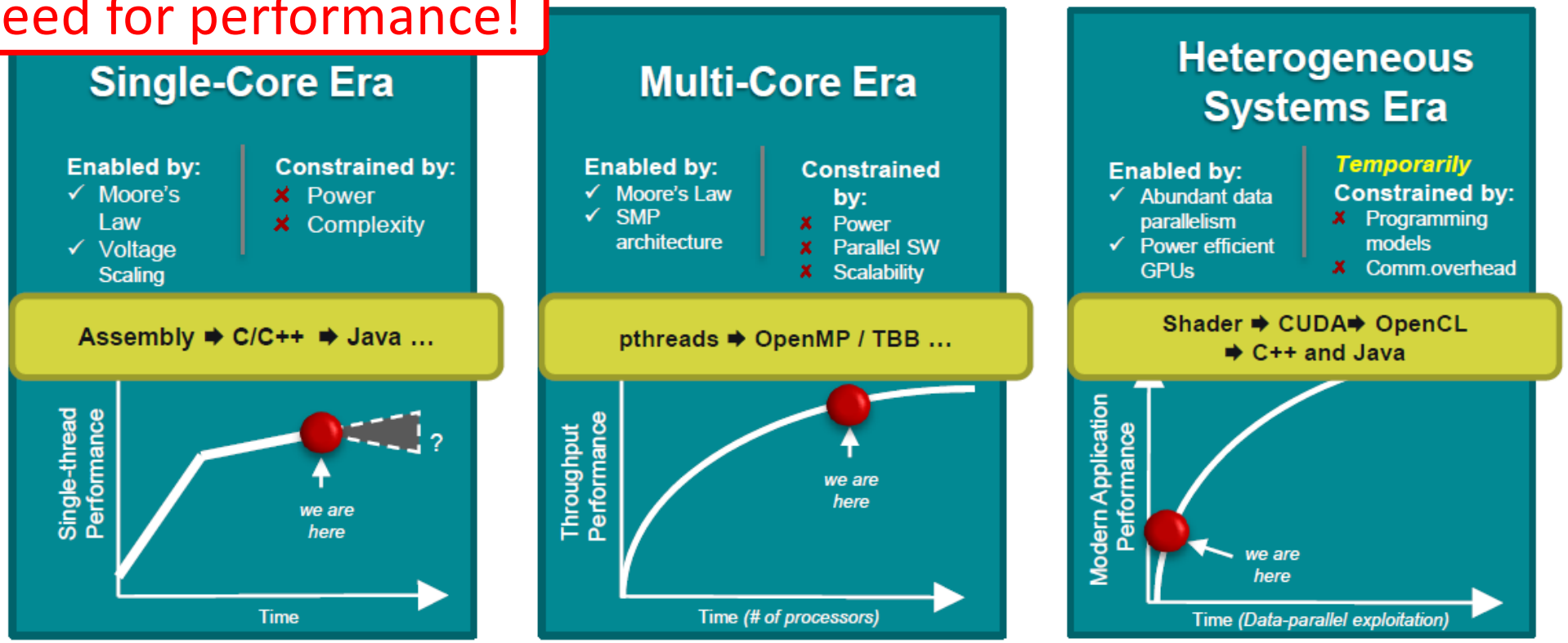
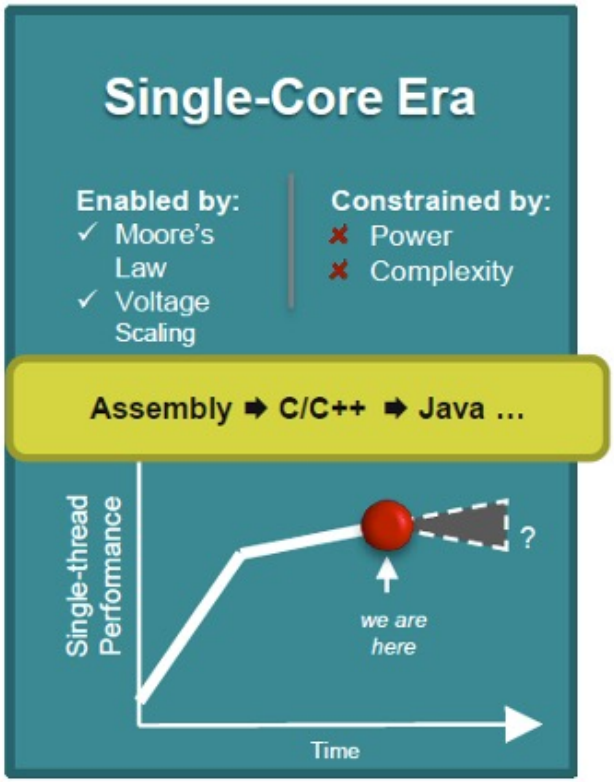
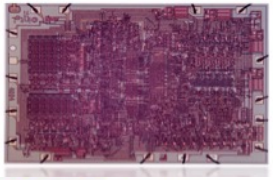


Image taken from http://hsafoundation.com

Single-Core Era

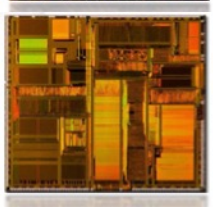


- Increasing voltage/frequency scaling

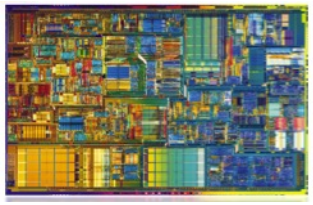


1971: 4004,
2300 trans. 740 KHz

1982: 80286,
134 thousand trans, 8 MHz

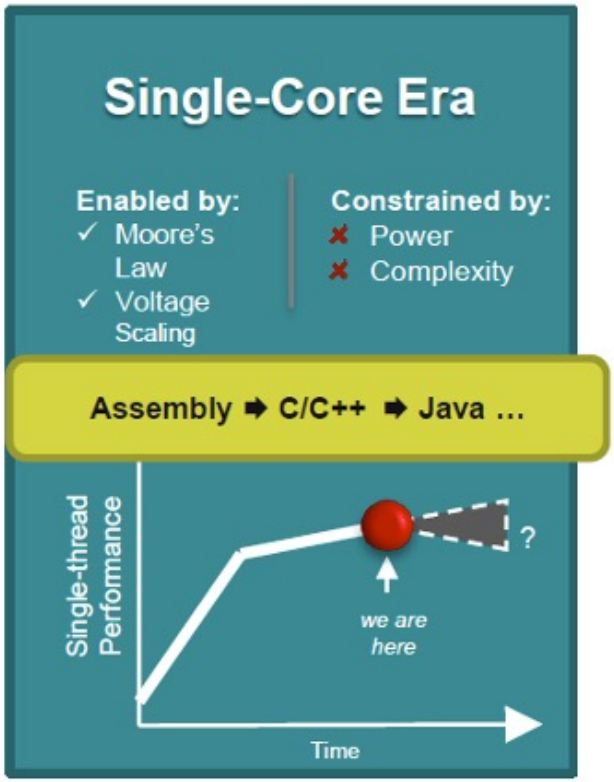


1993: Pentium P5,
1.18 mill. trans, 66 MHz

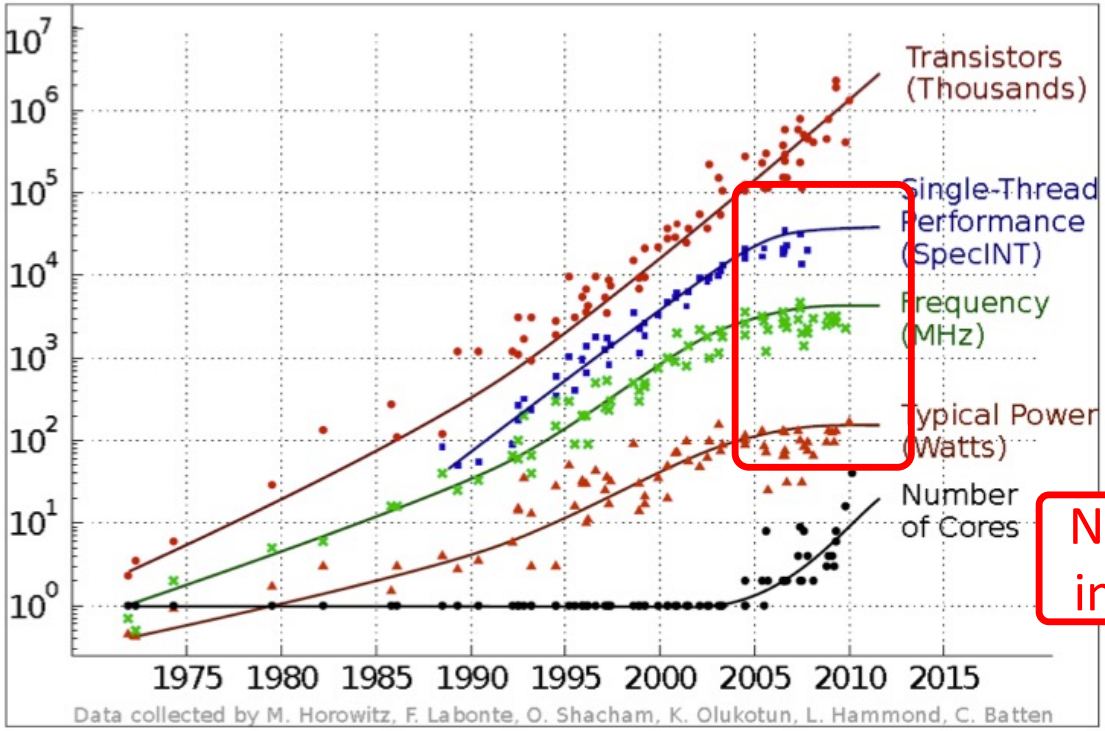


2000: Pentium 4,
42 mill. trans, 1.5 GHz

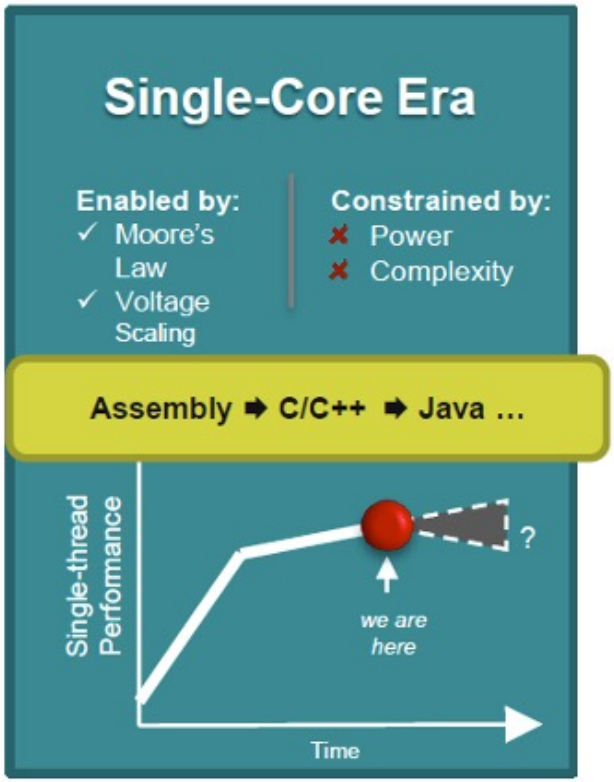
Single-Core Era



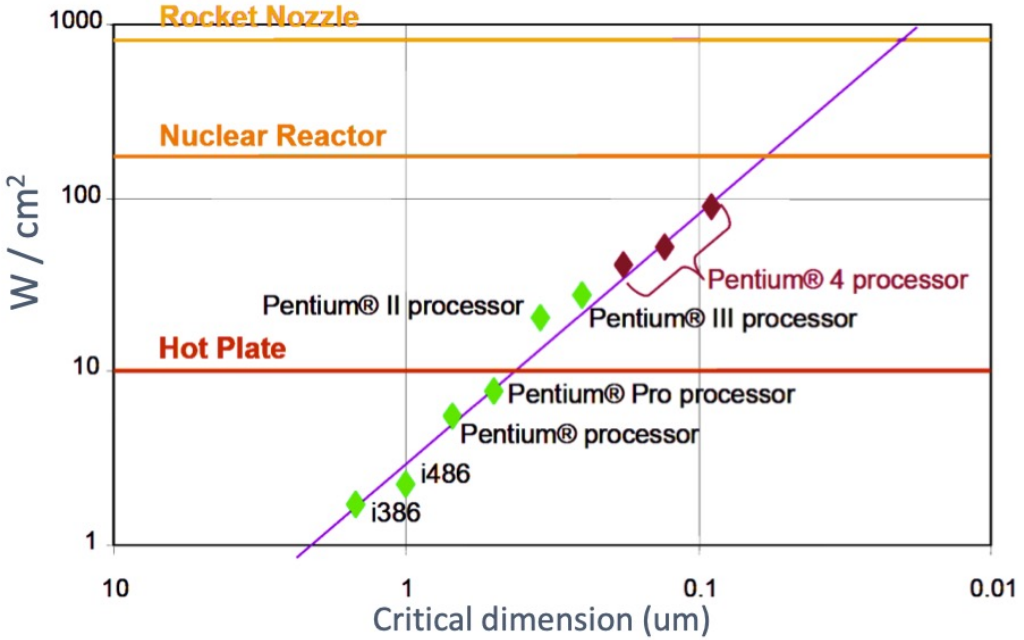
- Increasing voltage/frequency scaling



Single-Core Era

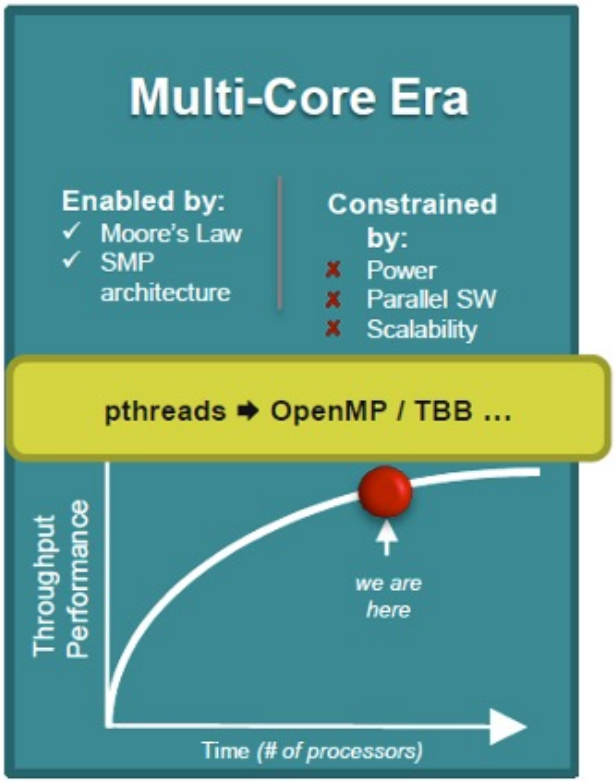


- Power wall has been reached in 2004
  - Too high heat density in Pentium 4!

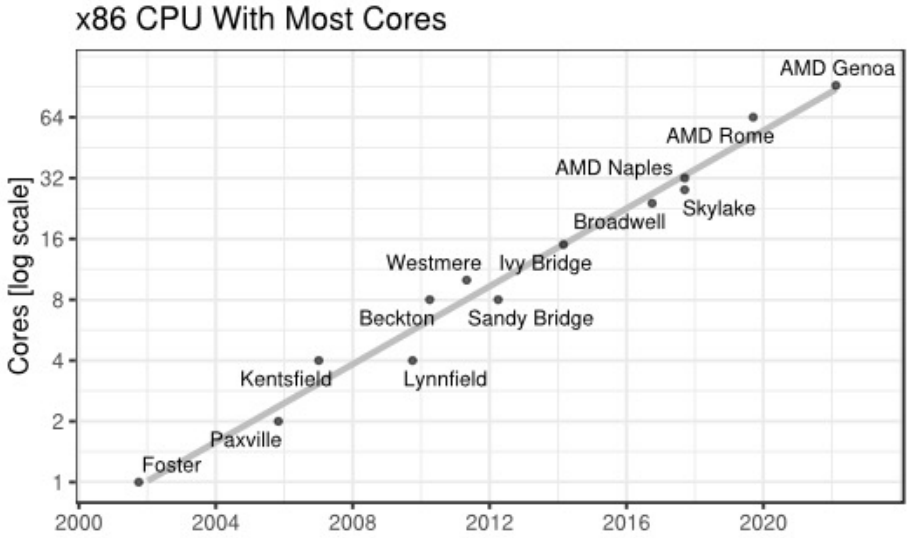


Original plot by G. Taylor, "Energy Efficient Circuit Design and the Future of Power Delivery" EPEPS'09

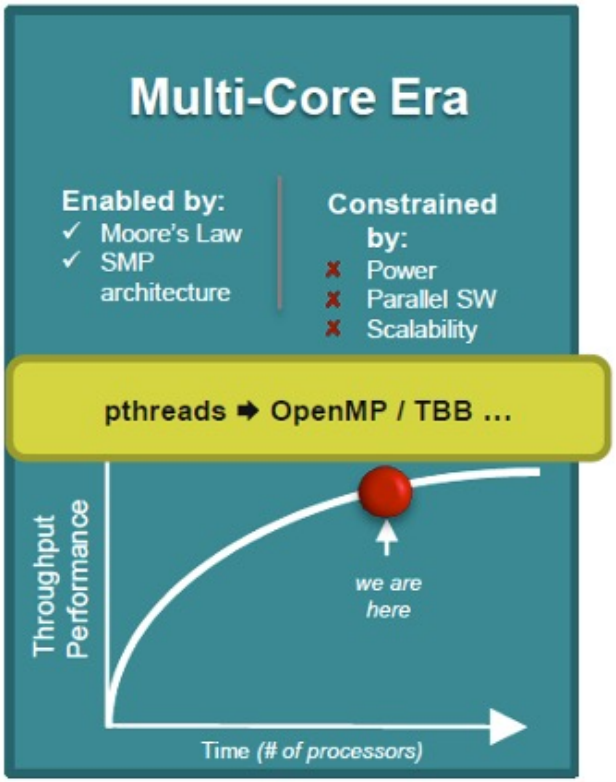
Multi-Core Era



- Integrate several cores in the same chip
  - We use to run many applications concurrently
  - Various applications can be **parallelized**!

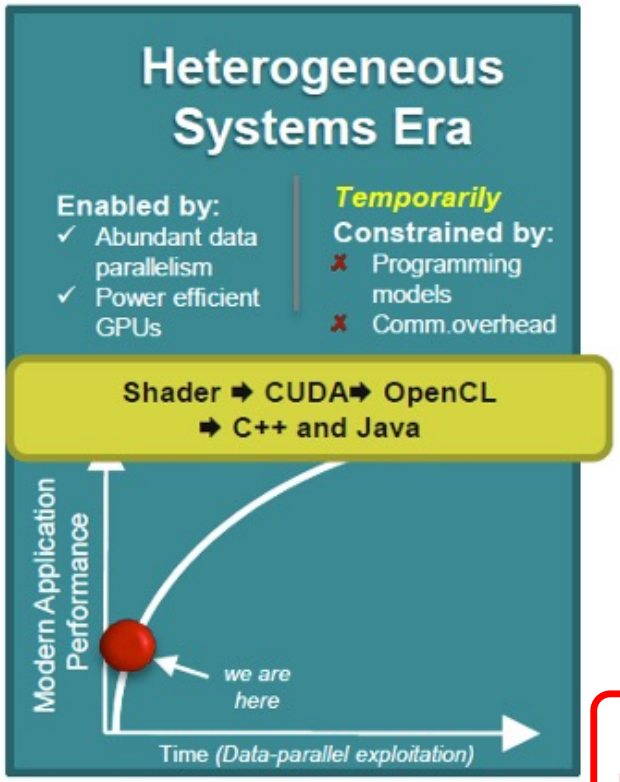


Multi-Core Era



- Integrate several cores in the same chip
  - We use to run many applications concurrently
  - Various applications can be **parallelized**!
- Limits in performance increase due to
  - Power consumption
  - Scalability issues

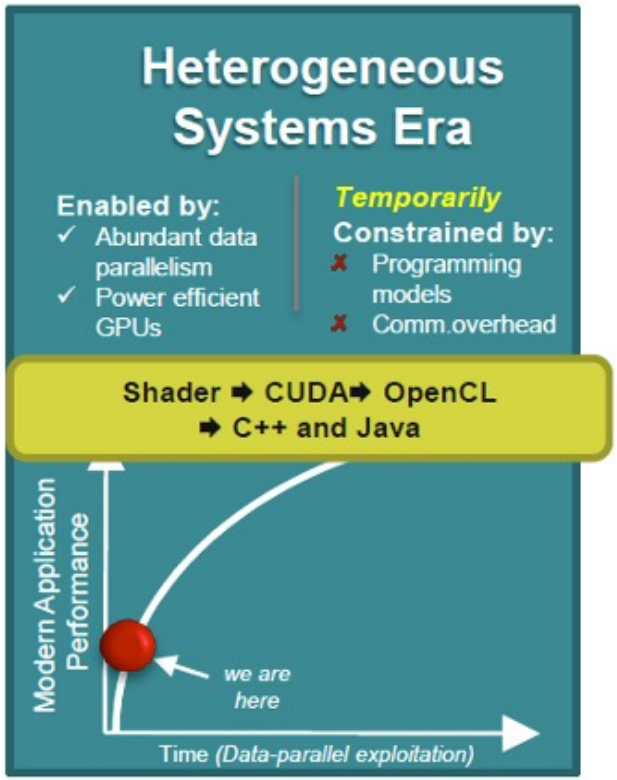
Heterogeneous Systems Era



- Integrate **heterogeneous units** in the same chip
  - Various applications can be **parallelized**!
  - Different parts of the applications may benefit from **specialized computing units**

FPGA and other types of devices used to be heterogeneous the early 2000s. Why did heterogeneous systems become popular more than 10 years later?

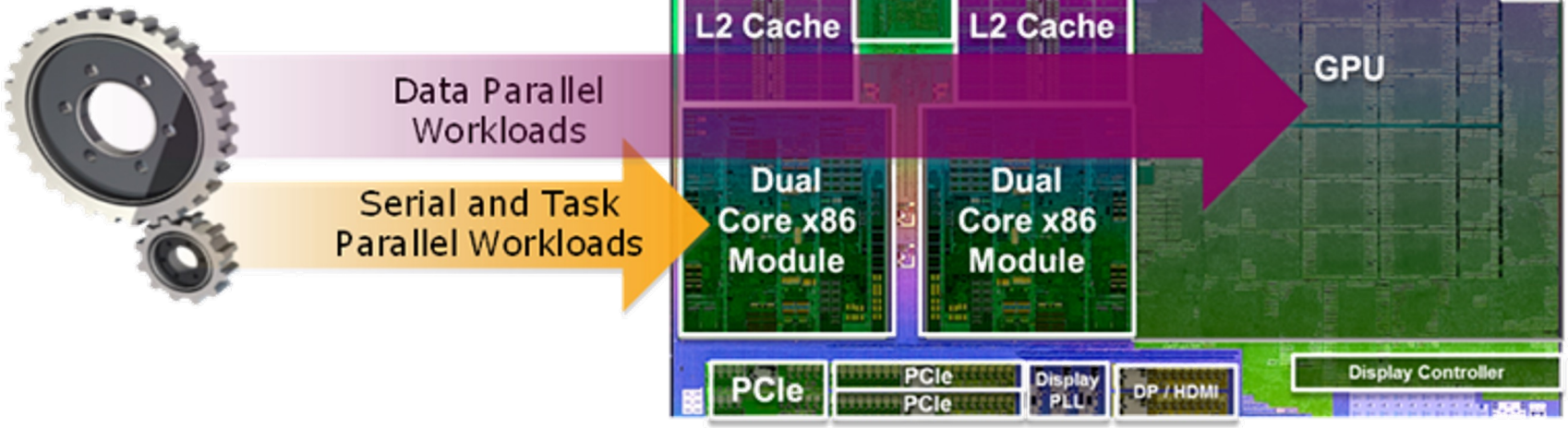
Heterogeneous Systems Era



- The large spread of GPUs in every type of computer has given heterogeneous system a so high popularity

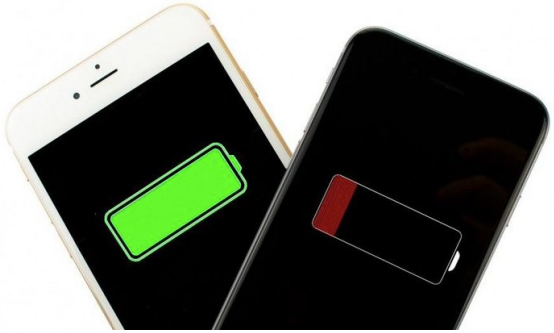
Heterogeneous Systems Era

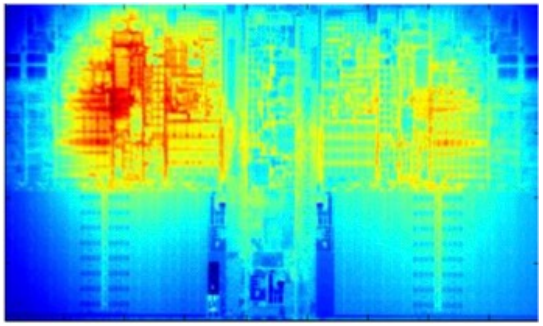
- Many compute-intensive applications can be partitioned in parts with different characteristics
- Each part can be efficiently accelerated by a different type of computing unit



Computing devices are power-constrained

- Modern devices are power constrained
  - Laptops, embedded and mobile devices
    - Battery duration
    - Thermal issues
  - HPC systems and supercomputers
    - High power supply and cooling costs





Computing units in heterogeneous systems are designed to offer high performance and energy efficiency!

Heterogeneous systems do not come for free!

- Issues in heterogeneous systems:
  - Integration of many different units and communication
  - Programmability
  - Resource management

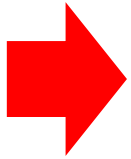
Integration of many different units and communication

- Architecture/HW engineers have to face with many integration issues:
  - Computing units' memories are separate; how to transmit in an efficient way data?
  - How to efficiently offload computing tasks from one unit to the other one?

HW aspects

Programmability

- Each type of unit has different programming languages and paradigms
- CPU
  - C, C++, Java, Python, …
- GPU
  - CUDA, OpenACC, …
- FPGA
  - VHDL, Verilog
- …



How many application engineers are expert in all these fields?

SW aspects

Resource management

- How to distribute a multi-programmed dynamic workload?
- How to configure SW/HW knobs?
  - E.g., dynamic voltage/frequency scaling
- How to achieve required performance at power-efficient level?

OS aspects

GPUs and Heterogeneous Systems
(programming models and architectures)

Some background on parallel architectures

Parallelism

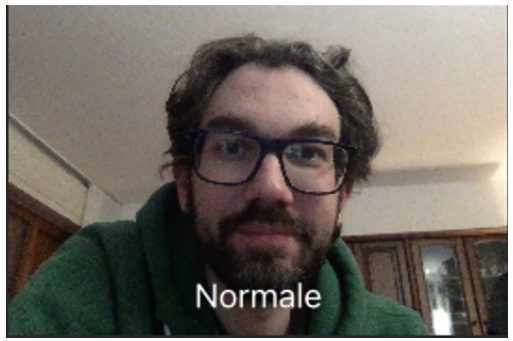
- Parallelism: independent operations which execution can be overlapped
- Operations: memory accesses or computations

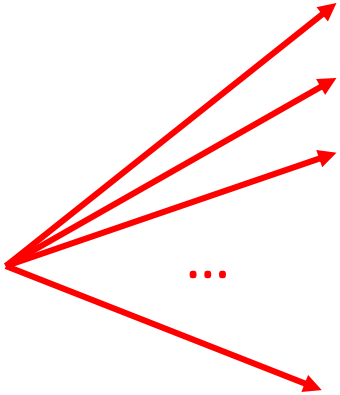
- Points of investigation:
  - Parallelism in applications
  - Parallelism in computing architectures

Parallelism in applications

- **Task parallelism**
  - The application contains many tasks/functions that can be executed independently in parallel

- E.g.: filters may be applied concurrently to the same image

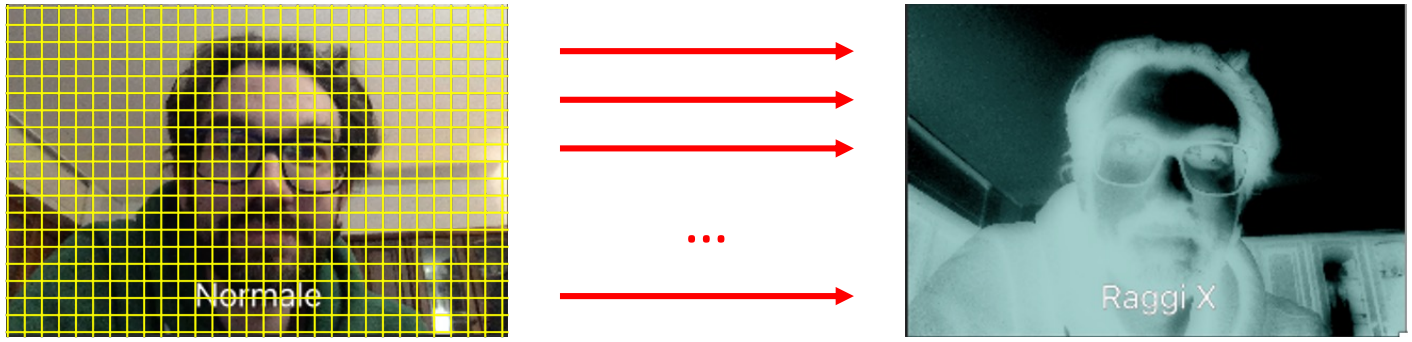




Parallelism in applications

- **Data parallelism**
  - The application operates on data composed of many items that can be operated on at the same time

- E.g.: many filters applies the same elaboration on each image pixel independently



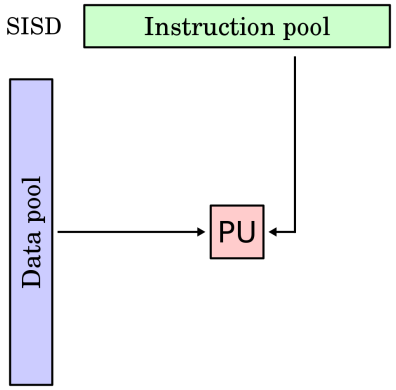
Flynn's taxonomy (1966)

- System architecture can be classified based on instructions and data:

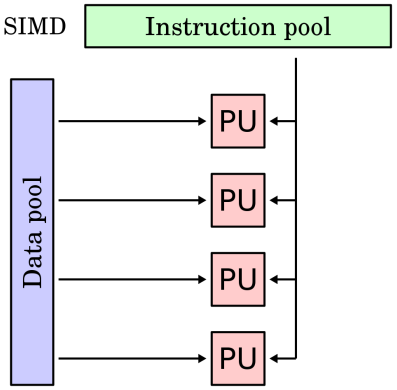
Instructions \ Data	Single Data (SISD)	Multiple Data (SIMD)
Single Instruction	Single Instruction Single Data (SISD)	Single Instruction Multiple Data (SIMD)
Multiple Instruction	Multiple Instruction Single Data (MISD)	Multiple Instruction Multiple Data (MIMD)

- Two more classes have been added to specialize MIMD
  - Single Program Multiple Data (SPMD) → A single program with multiple instructions, like MIMD, but confined to a program
  - Multiple Program Multiple Data (MPMD) → multiple programs, multiple SPMD

Flynn's taxonomy (1966)



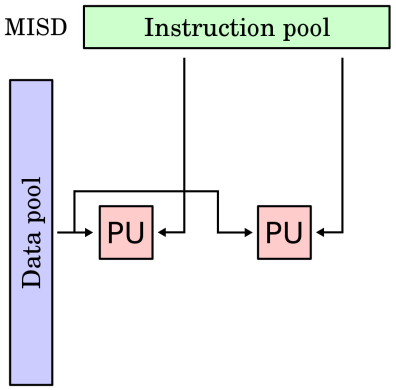
- SISD
  - Classical von Neumann architecture
  - E.g.: Traditional uniprocessor machines



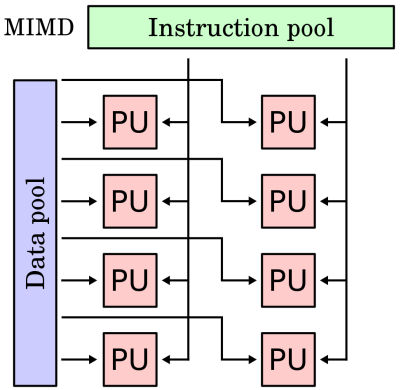
- SIMD
  - Multiple PUs perform the same operation simultaneously on different data items
  - E.g.: vector machines

PU: processing unit

Flynn's taxonomy (1966)



- MISD
  - Multiple instructions operate on one data stream
  - Uncommon architecture



- MIMD
  - Multiple autonomous processors simultaneously executing different instructions on different data
  - E.g.: multi-core machines

PU: processing unit

Flynn's taxonomy (1966) – further classes

- Single Program Multiple Data (SPMD)
  - Multiple autonomous processing units simultaneously executing a single program on different data items
  - Program execution can have an independent path for each data point

- Multiple Program Multiple Data (MPMD)
  - Multiple autonomous processing units simultaneously executing at least two independent programs

Architectural solutions for parallel computing

- In a computer architecture parallelism can be extracted at different levels:
  - Instruction-Level Parallelism (ILP)
  - Data-Level Parallelism (DLP)
  - Thread-Level Parallelism (TLP)

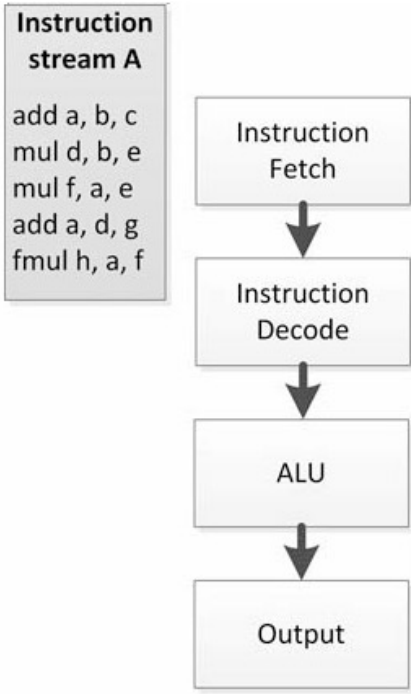
- At each level, various architectural solutions have been defined

Next examples taken from Kalei and others “Heterogeneous Computing with OpenCL 2.0”

Basic non-parallel architecture

- Instructions are executed in sequence
- A new instruction is issued when the previous one is completed
- Each instruction works on a single data item

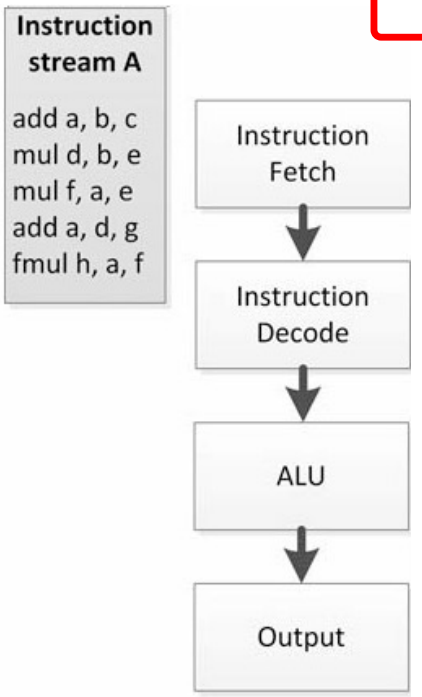
Traditional old scheme!



Pipelined architecture

- The architecture is organized in different stages
- Multiple instructions are overlapped in execution
  - Each instruction "occupies" a single stage
- Instructions are executed in the order they appear in the program code

ILP

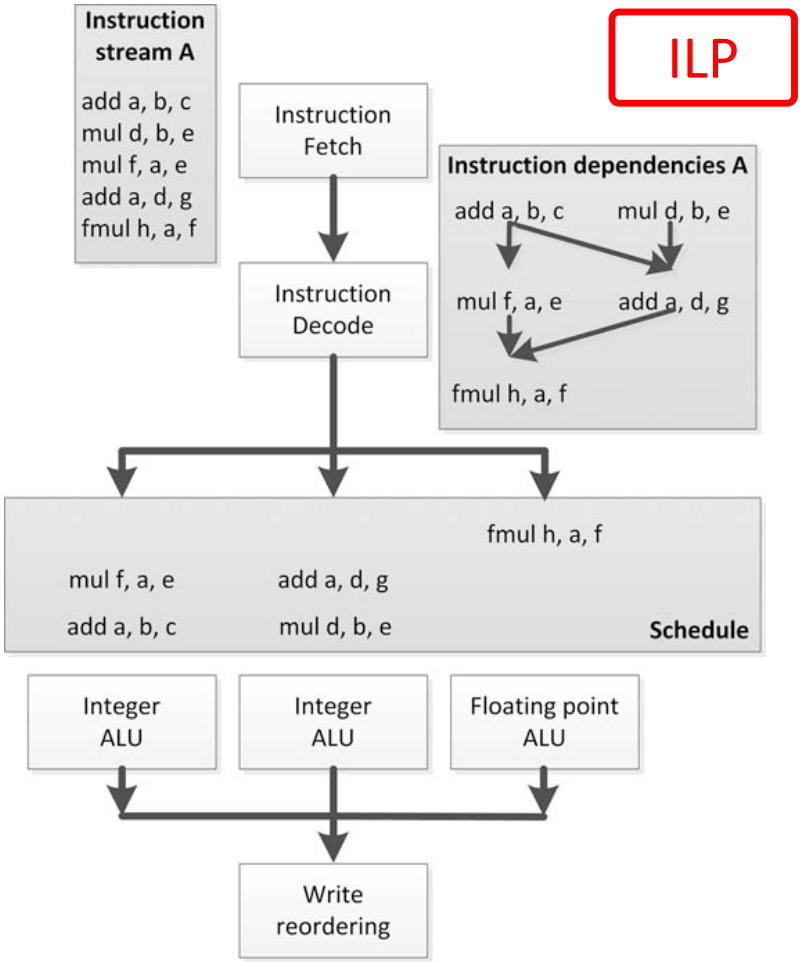


	Clock number								
Instruction number	1	2	3	4	5	6	7	8	9
Instruction i	IF	ID	EX	MEM	WB				
Instruction $i+1$		IF	ID	EX	MEM	WB			
Instruction $i+2$			IF	ID	EX	MEM	WB		
Instruction $i+3$				IF	ID	EX	MEM	WB	
Instruction $i+4$					IF	ID	EX	MEM	WB

<- in the timeline the output stage is partitioned in MEM and WB

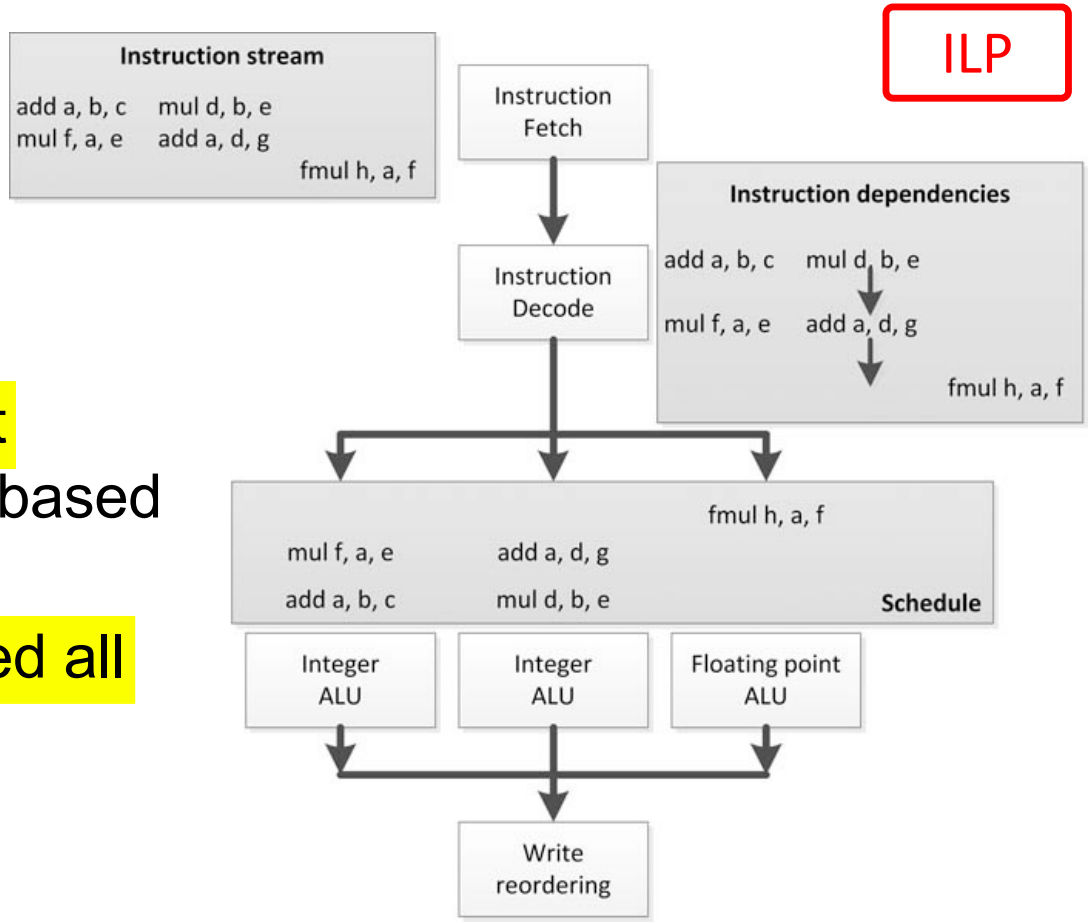
Superscalar architecture

- The architecture contains several ALUs
- The architecture keeps track of the data dependencies among instructions
- Instructions are scheduled out of order on the ALUs



Very Long Instruction Word (VLIW) architecture

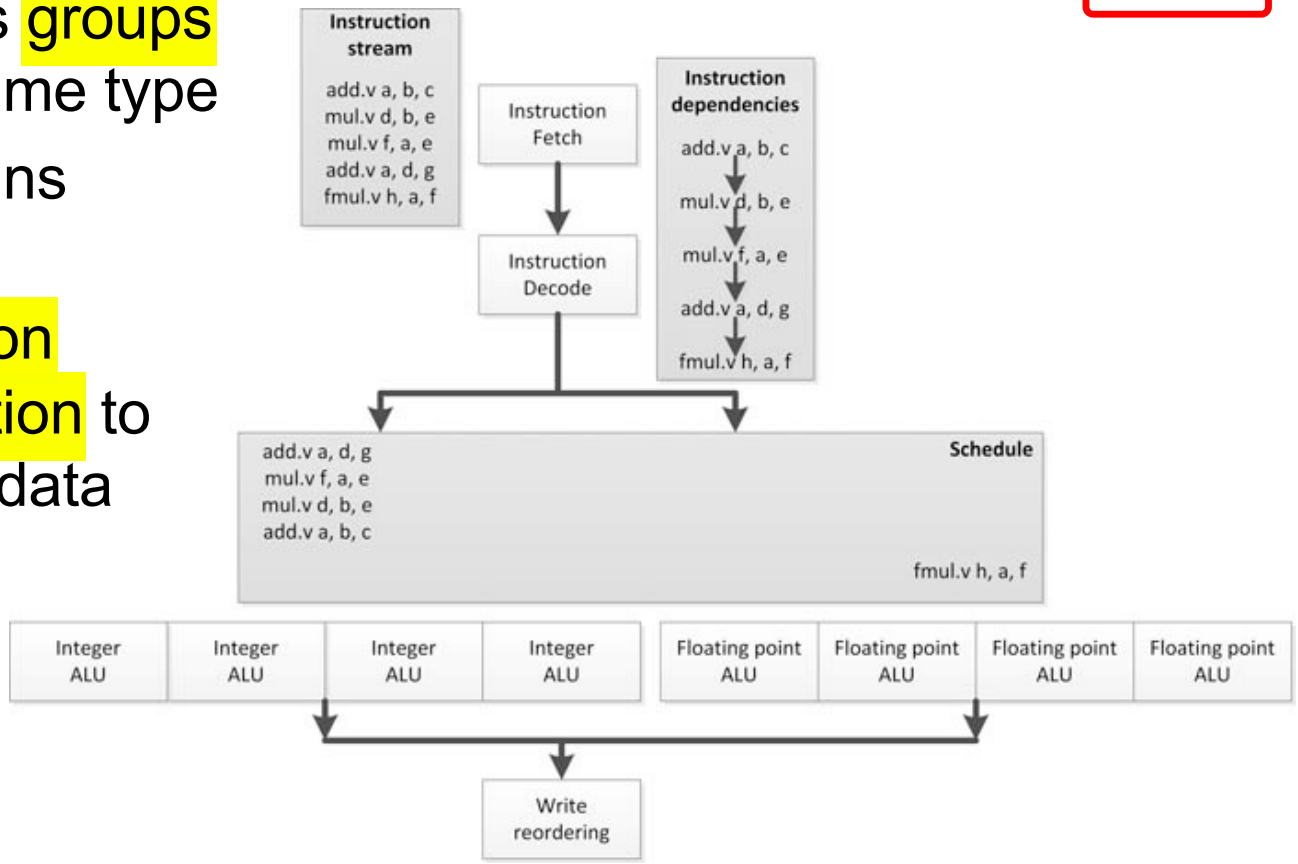
- The architecture contains several ALUs
- Dependency analysis is performed by the compiler
- The compiler packs independent scalar instructions in a long one based on the available ALUs
- Each long instruction is scheduled all together



SIMD architecture

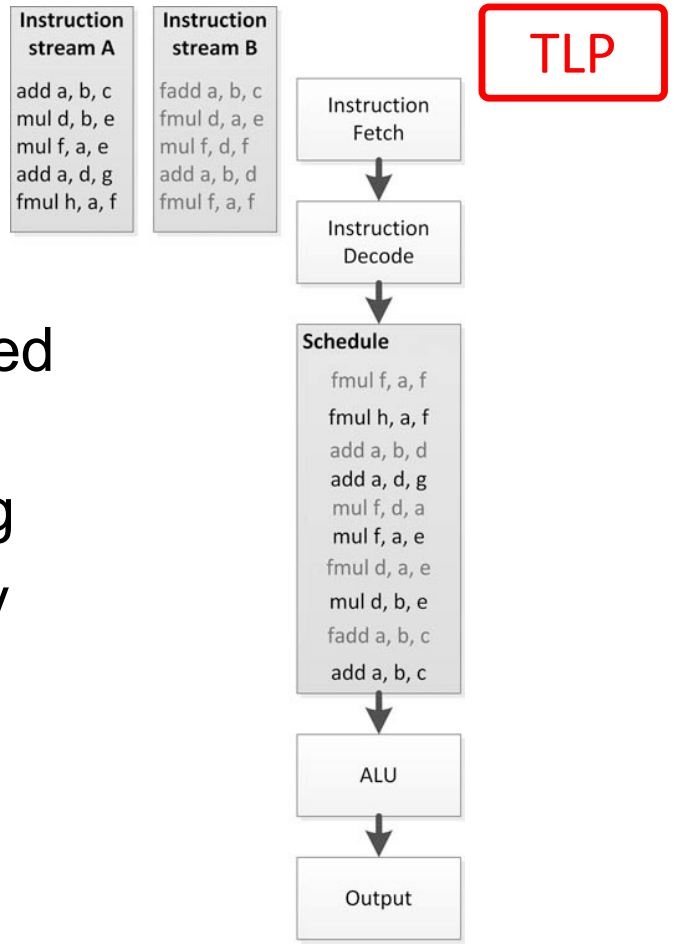
DLP

- The architecture contains groups of several ALUs of the same type
- The program code contains vectorized instructions
- Each vectorized instruction requires the same operation to be executed on different data items in parallel



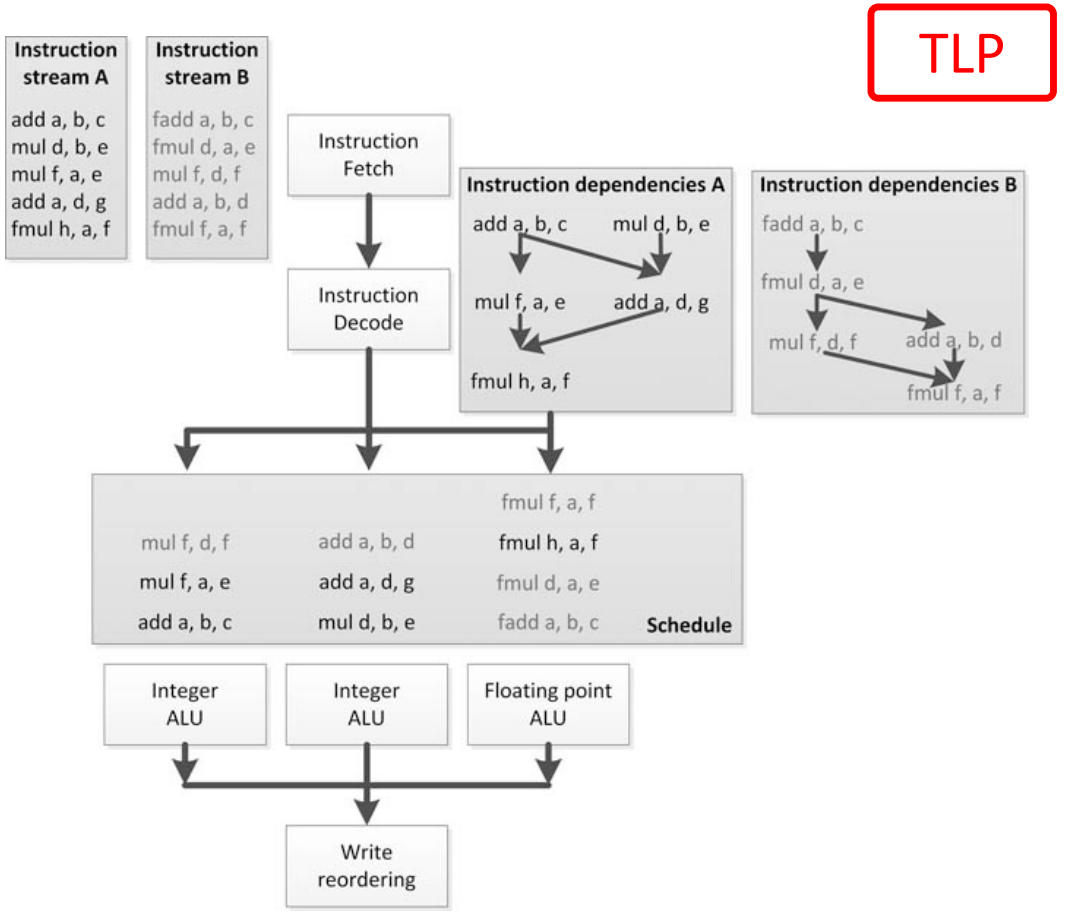
Interleaved Multi Threading architecture

- Multiple instruction streams (called **threads**) are executed together
  - Threads are independent from each other
- The architecture stores the execution data (called **context**) of all the threads
- The various threads are executed in time-slicing
  - Possibility to hide stalls of a single thread by executing the other threads



Simultaneous Multi Threading architecture

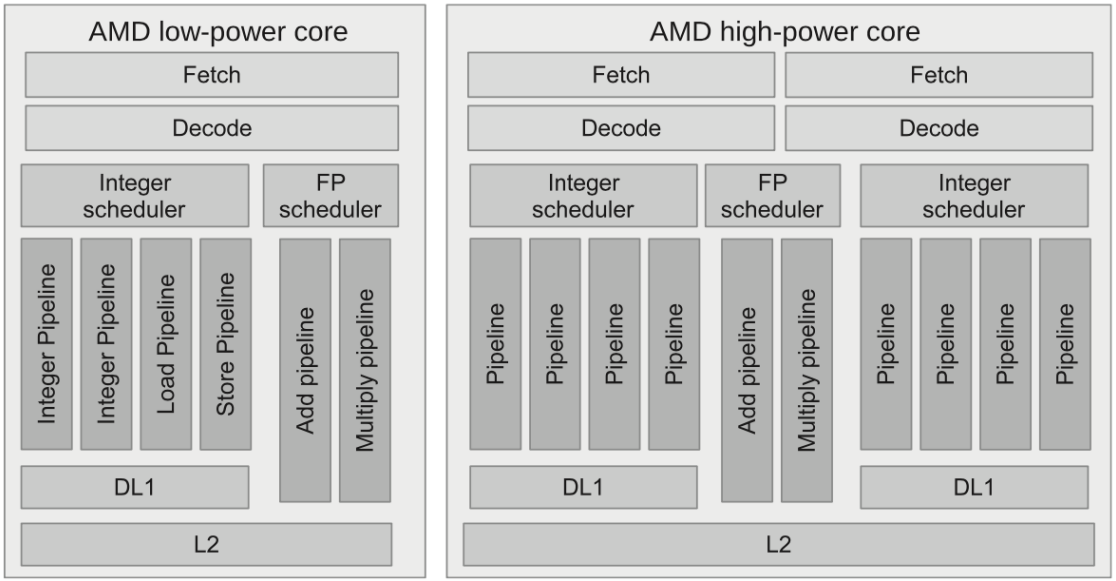
- Extension of the superscalar architecture executing multiple threads
- Threads have no dependency from each other!



Multicore architecture

TLP

- The architecture contains several cores
  - Cores may be of different types
- Each core executes a thread (at least)
- Need for a cache hierarchy to avoid memory accesses to be the bottleneck



Which kind of parallelism is exploited in GPUs?