Current Computer Architecture and Power Management (and the Intel® Simics® Simulator)

Jakob Engblom, PhD
Director of Simulation Technology Ecosystem, Simics team, Intel, Stockholm, Sweden

2023-05-04

intel®

Legal Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete http://www.intel.com/performance.

Intel, the Intel logo, Simics, Xeon, Xeon Phi, Atom, Quark, Core, Pentium, 3D Xpoint, Optane, Movidius, Iris, eASIC are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Jakob Engblom

D92

Currently:

- Director (of Simulation Technology Ecosystem), Simics Core team, at Intel in Stockholm, Sweden

Education:

- MSc, Computer Science, and PhD, Real-Time Systems, **Uppsala**

Experience: virtual platforms, simulation, embedded systems

- Product management, product marketing, technical sales, technical marketing, business development, training development, demos, … At IAR Systems, Virtutech, Wind River, and Intel

My own blog, since 2007:

- https://jakob.engbloms.se

Intel Community Blog

Where do we Fit into Intel?

Get our software for free at https://developer.intel.com/intel-isim

Laptop and desktop
- Intel® Core®
- Intel® Atom™
- Chipsets
- Thunderbolt*
- Graphics Processors (GPU)

Data Center
- Intel® Xeon®
- Chipsets
- Infrastructure processing units (smart network)
- GPU
- Bitcoin Miners (BZM)

AI and ML
- Movidius
- Habana
- Intel® Xeon®
- GPU

Connectivity
- Ethernet
- WiFi
- Bluetooth
- GNSS

FPGA
- SoC-FPGA
- FPGA
- eASIC hard-copy

Intel Labs
- Quantum computing
- Neuromorphic computing
- Software

Foundry
- Intel Foundry Services

Software
- OneAPI
- Development tools
- Compilers
- Simulation solutions
- Linux & Windows drivers
- UEFI & BIOS

What is in a Computer?

What’s in a ”Computer”?

(Main) Processor cores

- Run user-visible OS and applications

Main memory (“RAM”)

Graphics and display

Audio and media processing

- Camera, microphone, speakers, image processing, video playback, video compression, …

Storage (“Disk”)

- NVMe, M.2., SATA, PCIe, SSD, HDD, USB, Thunderbolt, …

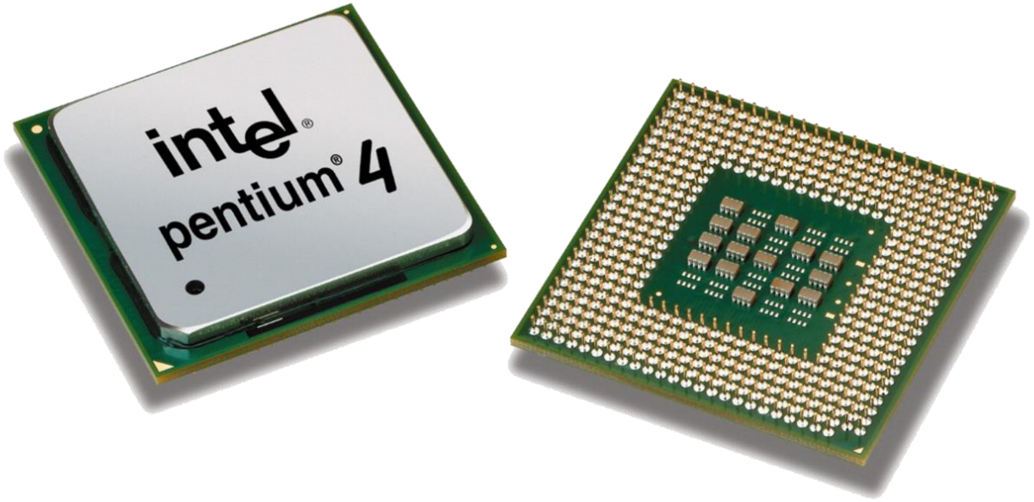
Networking

- Ethernet, WiFi, Bluetooth, …

Local peripherals -

- USB, Thunderbolt, Serial, Bluetooth, …

Once Upon a Time...



The "processor" was the essential part of a system

It measured the goodness of the machine:

- Megahertz
- Instructions per cycle
- Cache size (from the 1990s)

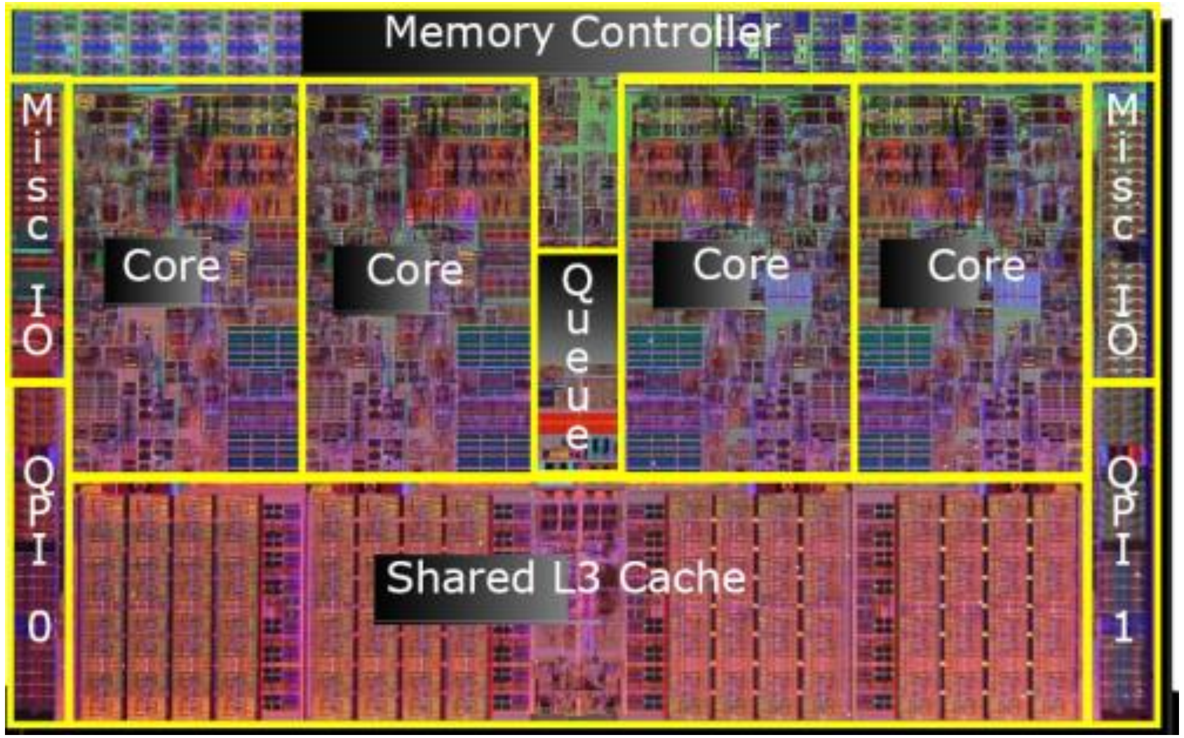
The supporting chipset was very basic

A better computer meant a better processor (mostly)

2009: Intel® Core™ i7 Processor: Still a Processor

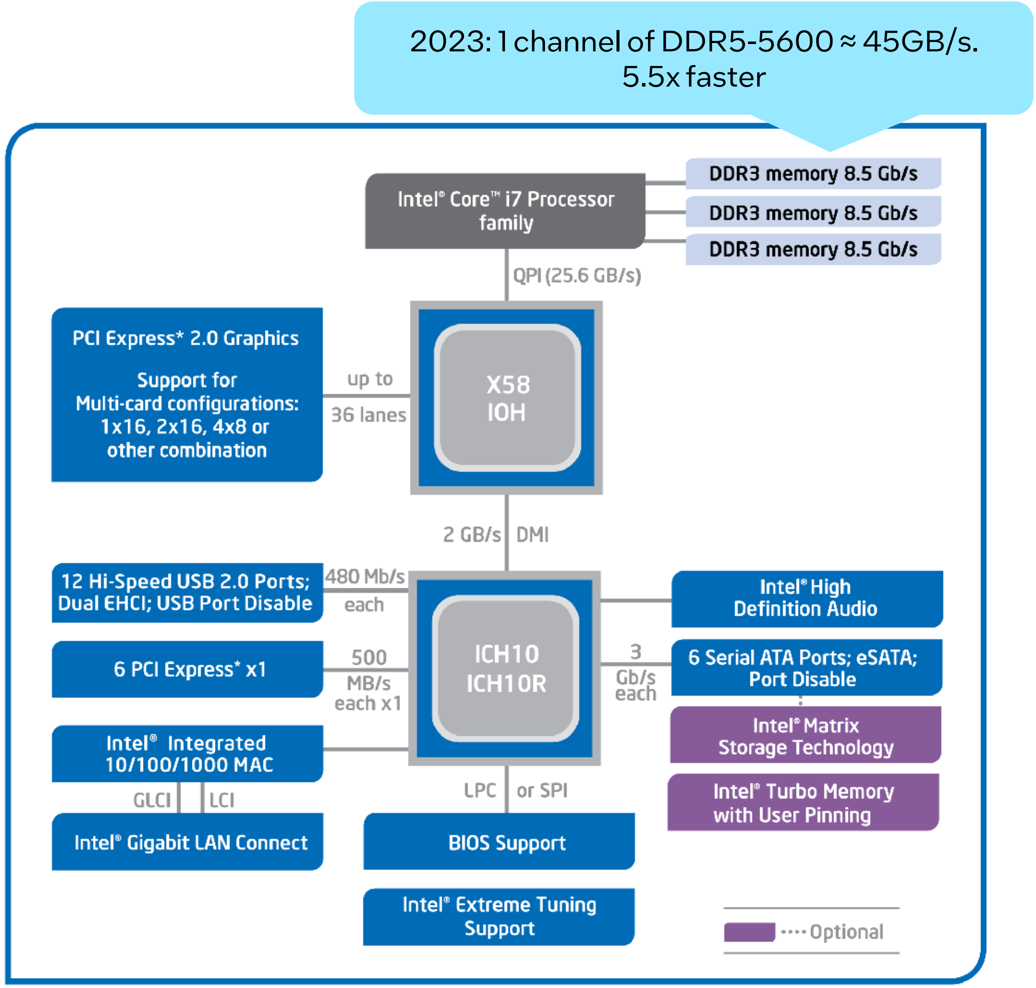
Intel® Core™ i7-960 Processor (2009)

- The processor chip is a processor with minimal other functionality
- Cores + cache
- Memory controller –moved on-chip in this generation
- Intel QuickPath Interconnect (QPI) – link to the rest of the system



- http://hexus.net/tech/reviews/cpu/16187-intel-core-i7-x58-chipset-systems-go-fsb-invited/?page=3

2009: Intel® X58 Express Chipset



Intel® X58 Express Chipset Block Diagram

Two chips + the processor

- Today, integrated as a single unit

I/O Hub (IOH)

- Fast link to the processor
- Graphics cards and other high-bandwidth PCIe devices

I/O Controller Hub (ICH10)

- Linked to the IOH over a slow link
- Main IO chip for slow IO
- SATA, Audio, USB, PCIe, Ethernet

Processors Today

2021: Laptop Processor (SoC)

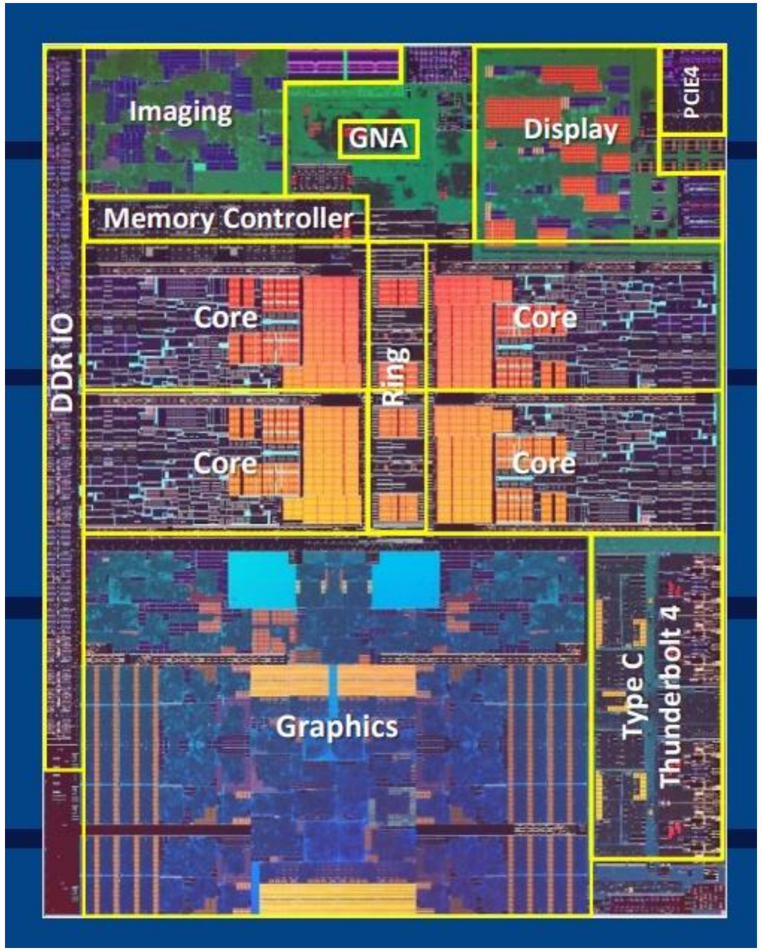
11th Gen Intel® Core™

- Quad-core laptop processor

Massive offload engines:

- Graphics block bigger than four processor cores
- Big imaging and display blocks
  - Can drive 4x4K displays, capture 4K video
  - Accelerates Artificial Intelligence algorithms
- USB Type C and Thunderbolt block as big a processor core

Small chipset in the same package adds legacy IO



Source: https://www.extremetech.com/computing/314565-intels-tiger-lake-is-spoiling-for-a-rematch-against-amds-zen-3

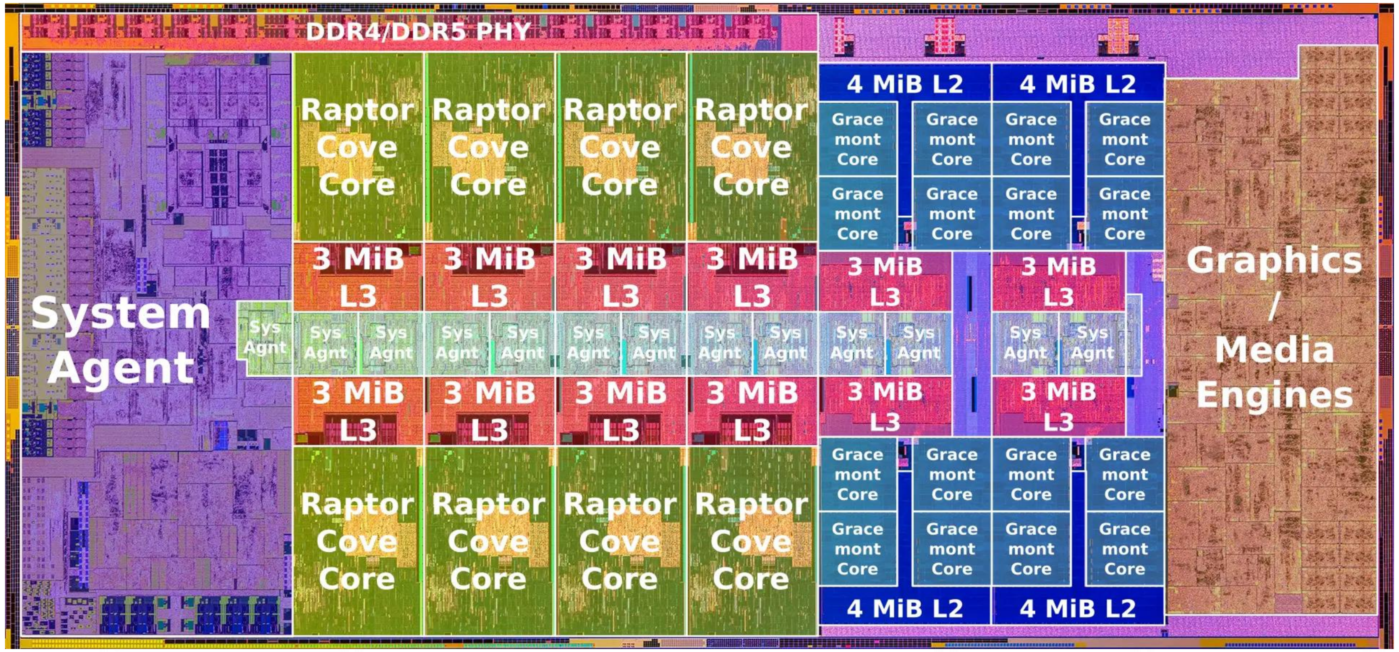
2022: Desktop Processor

13th Gen Intel® Core™

- 8+16 core desktop processor

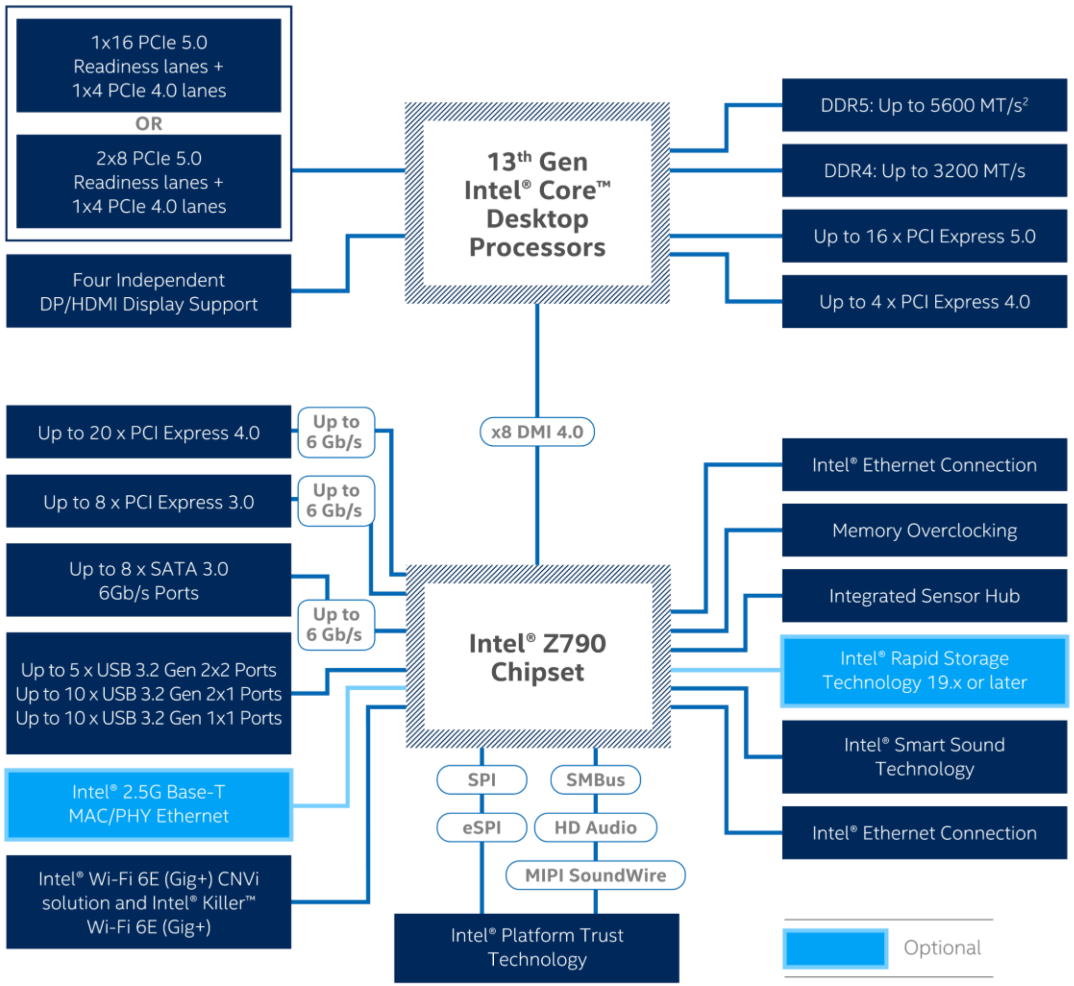
Different system balance

- Desktop=more area spent on processor cores
  - Smaller graphics block provides basic services,
- All other IO in the chipset



Source: Intel Raptor Lake annotated die shot from Wikichip
https://en.wikichip.org/wiki/File:intel_raptor_lake_die_%288%2B16%29_%28annotated%29.png

2022: Intel® Z790 Chipset



Processor

- Memory and fast PCIe

Chipset

- PCH, Platform Controller Hub

Massive IO:

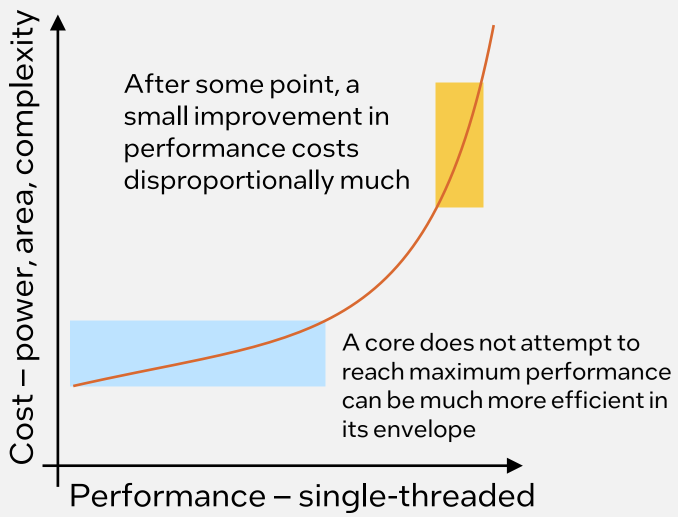
- 16 + 4 + 20 + 8 PCIe
- 15-20 USB (including Type-C)
- WiFi + wired Ethernet (require external PHYs)
- Displays, sound
- Thunderbolt – add as external chip

Different market

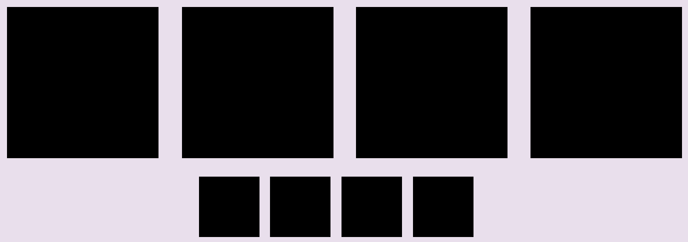
- Open for motherboards manufacturers to differentiate
- Desktop is not space-constrained = external chips OK

Big and Small (and Intermediate) Cores

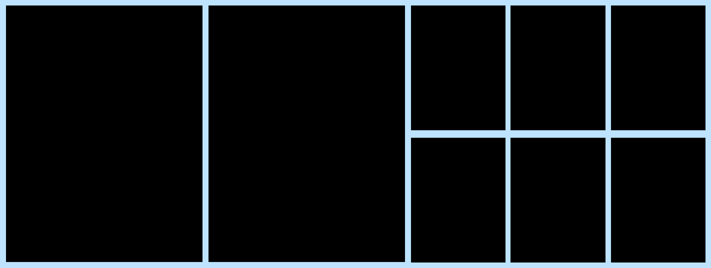
Graph just for illustration purposes – not based on any real numbers



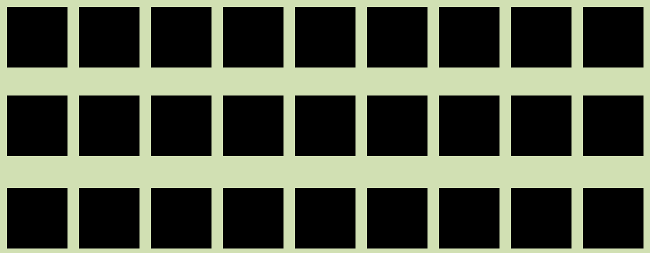
Processor core performance tradeoff



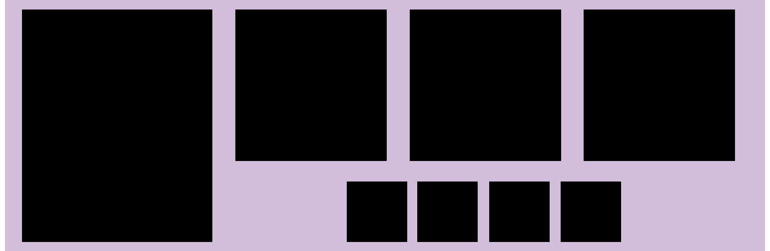
Mobile phone: Light background tasks on small cores, decent performance for foreground from big cores



Laptop: Maximum performance big cores, smaller cores can do significant processing at lower area and lower power – scale up multithreading



Throughput server or budget phone: Only small cores, trade single-threaded performance for multi-threaded (or just lower cost)



Mobile phone 2: One extra-big core for critical single-threaded tasks, plus “big”, plus “small”

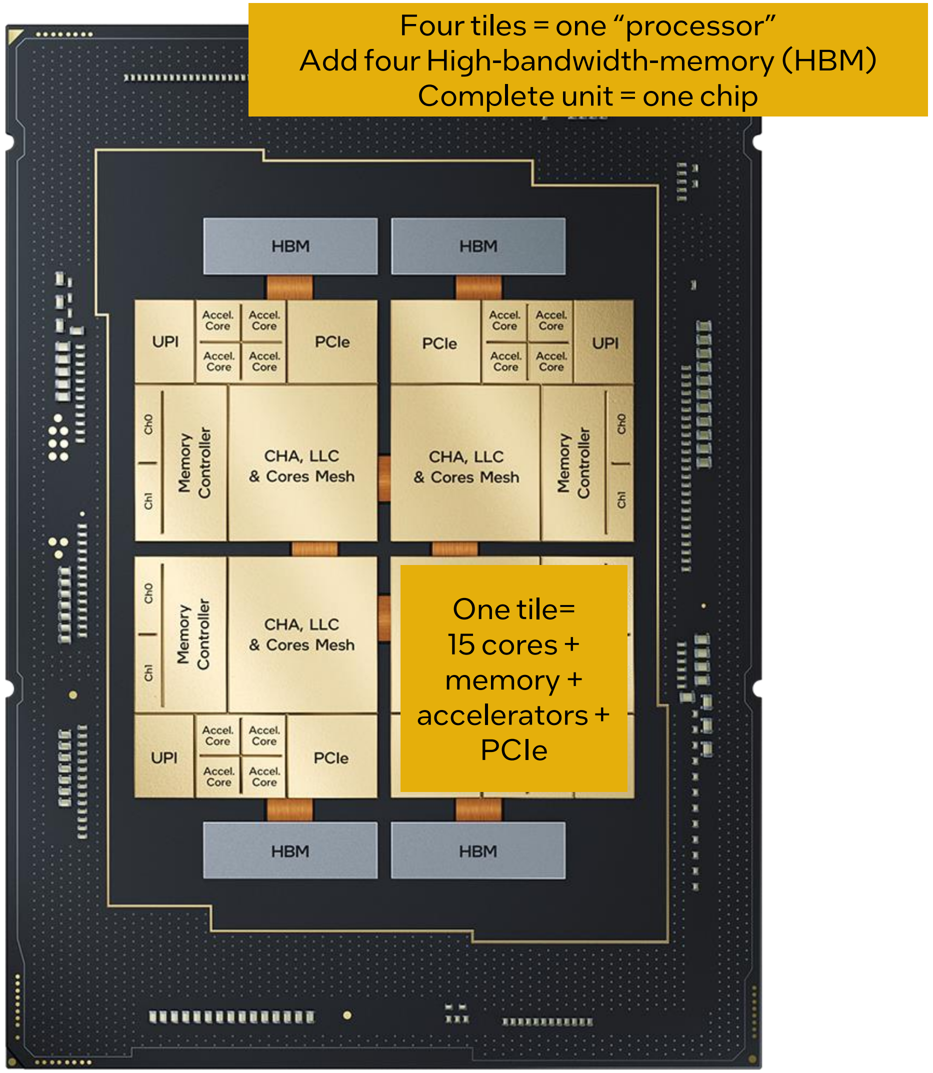
More reading:
https://chipsandcheese.com/2021/12/21/gracemont-revenge-of-the-atom-cores/

https://www.tomshardware.com/reviews/intel-core-i9-12900k-and-core-i5-12600k-review-retaking-the-gaming-crown/6

https://www.anandtech.com/show/17102/snapdragon-8-gen-1-performance-preview-sizing-up-cortex-x2

Disaggregated Architecture: Chiplets/Tiles

4th Gen Intel® Xeon® CPU Max Series



Increasingly hard to build single monolithic chips

Answer: use "**chiplets**" (a.k.a. "**tiles**")

- Pieces of silicon which are not stand-alone
- Each tile built on a suitable technology (speed, power, cost, density, …)

Each chiplet or tile:

- A specific functionality: processor cores, cache, IO, graphics, …
  - Notably used in FPGA designs for a few years
- A grouping of functionality to build a scalable balanced solution
  - Processors + cache + accelerators + connectivity

Benefits:

- Easier to build *truly big* chips
- Easier to build a scalable product line
- Easier to innovate in each area

2023: Server Processor built from Chiplets

4th Gen Intel® Xeon®

- 56 cores
- 4 on-chip accelerator blocks
- 8 memory controllers
- 8 PCIe and CXL (Compute Xpress Link) controllers
- 4 "UPI" links (other processors)

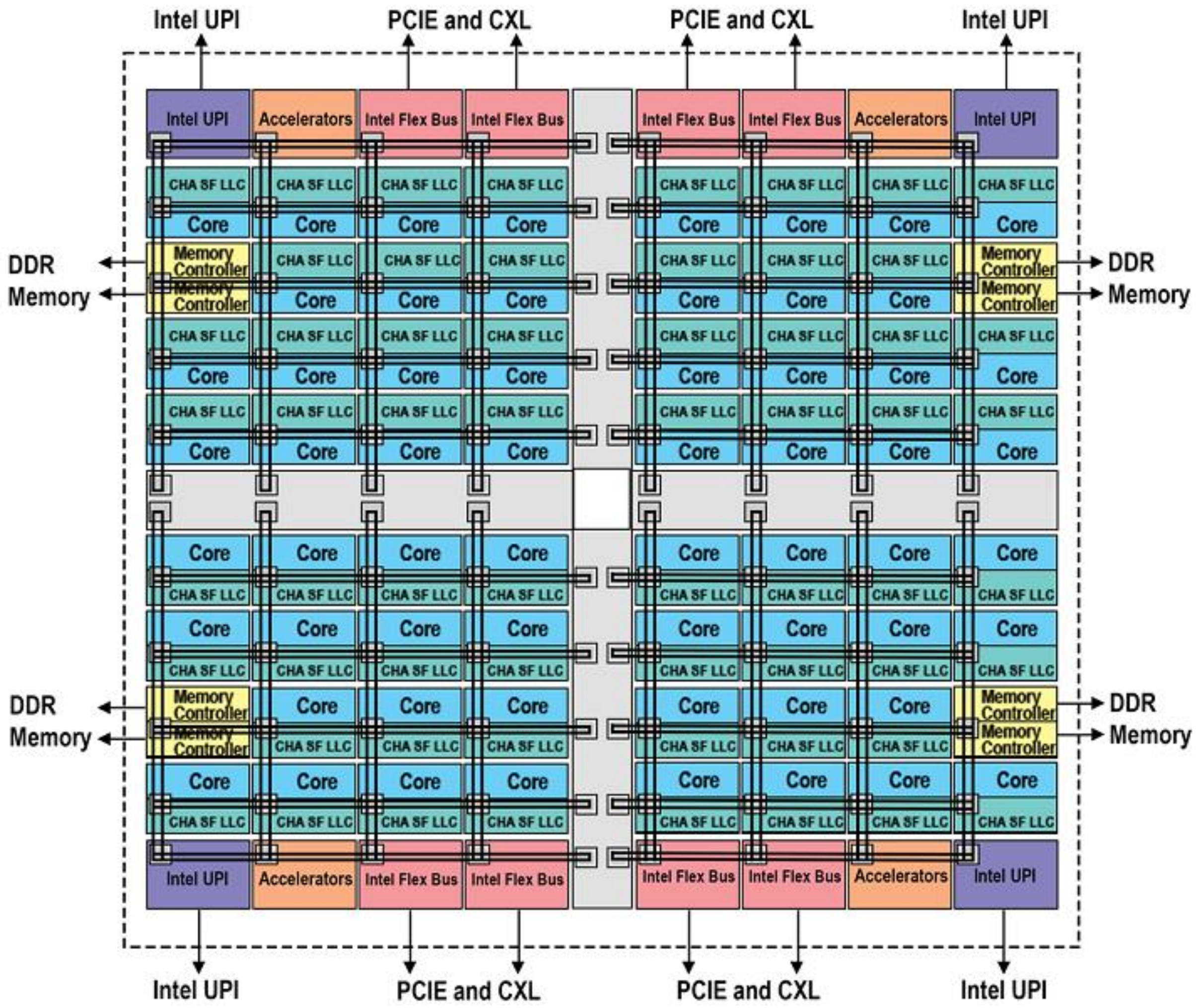
Built as 4 chiplets

- Single uniform-latency mesh between chiplets

Combine up to 8 chips in a single server

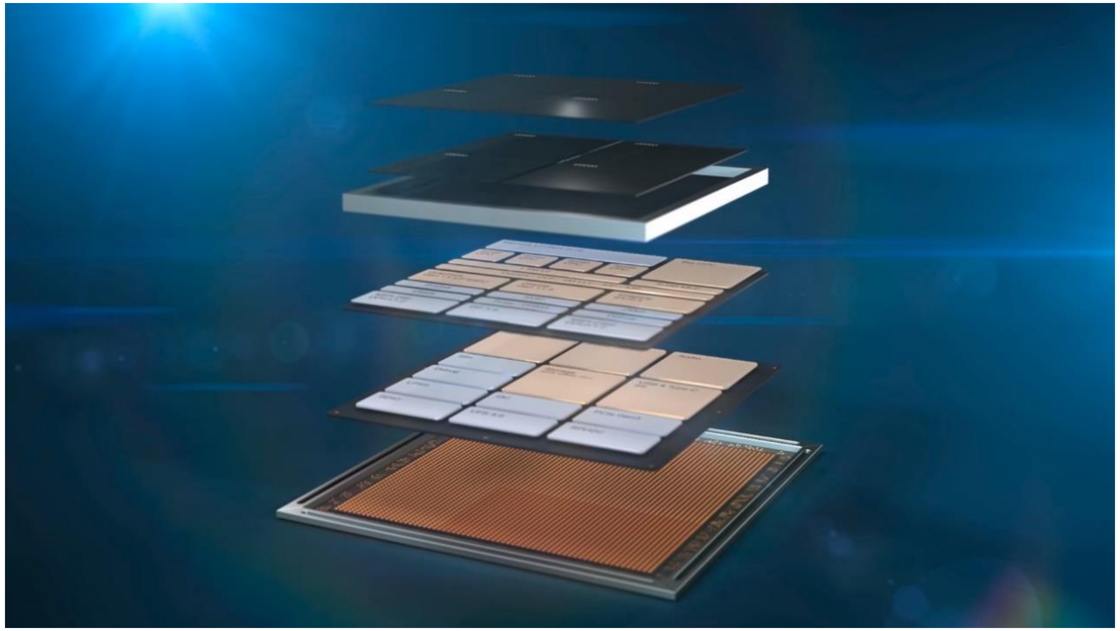
Other I/O on chipset:

- Network interfaces
- "Utility" – USB, serial, SATA, ...



Source: https://intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html

3D Stacking and Packaging



Good deeper reading:

https://community.cadence.com/cadence_blogs_8/b/breakfast-bytes/posts/3d-packaging-versus-3d-integration

Stacking dies in 3D in a single package (in addition to chiplets)

- Smaller footprint
- Higher bandwidth between chips
  - In particular, stack memory on top of processors
- Sometimes aggregation of separate products, sometimes co-designed chips

Constraints to consider

- Power – feeding the stack
- Thermal – cooling is harder
- IO – not as many external connectors as with separate chips

Accelerators: Fixed(ish)-Function Systems

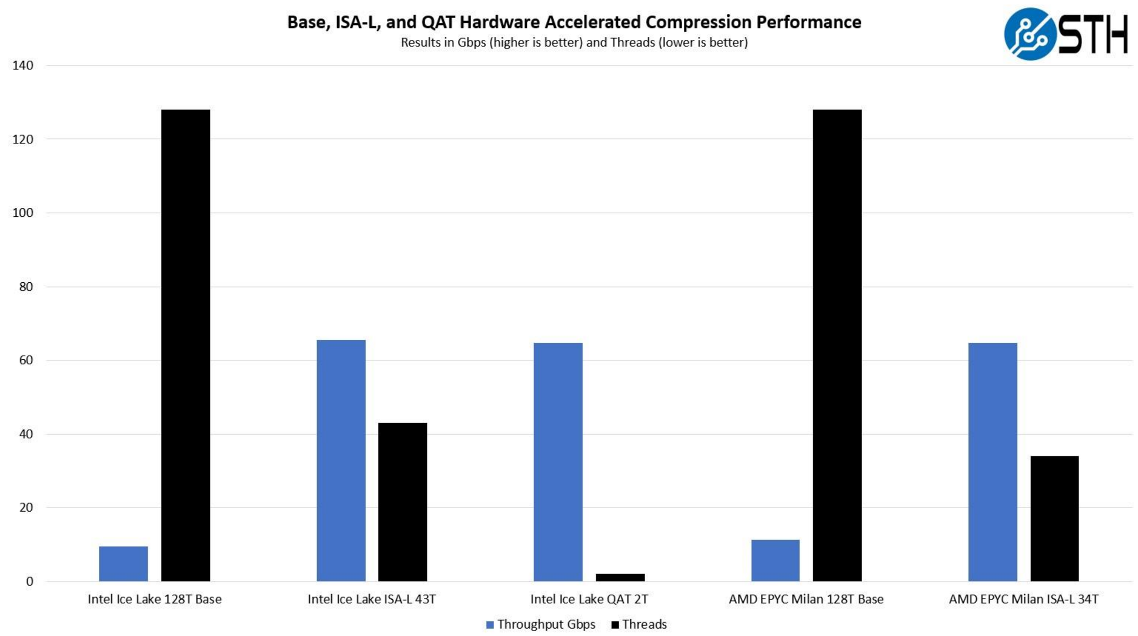
Graphics rendering	Audio processing	Camera image processing	Cryptography	Network packet processing
Data compression/ decompression	Neural network training	Neural network inference	Data movement	Video encoding/ decoding

Work is shifted to specialized fixed-function subsystems for higher performance and lower power – provided the workload happens often enough to warrant the investment

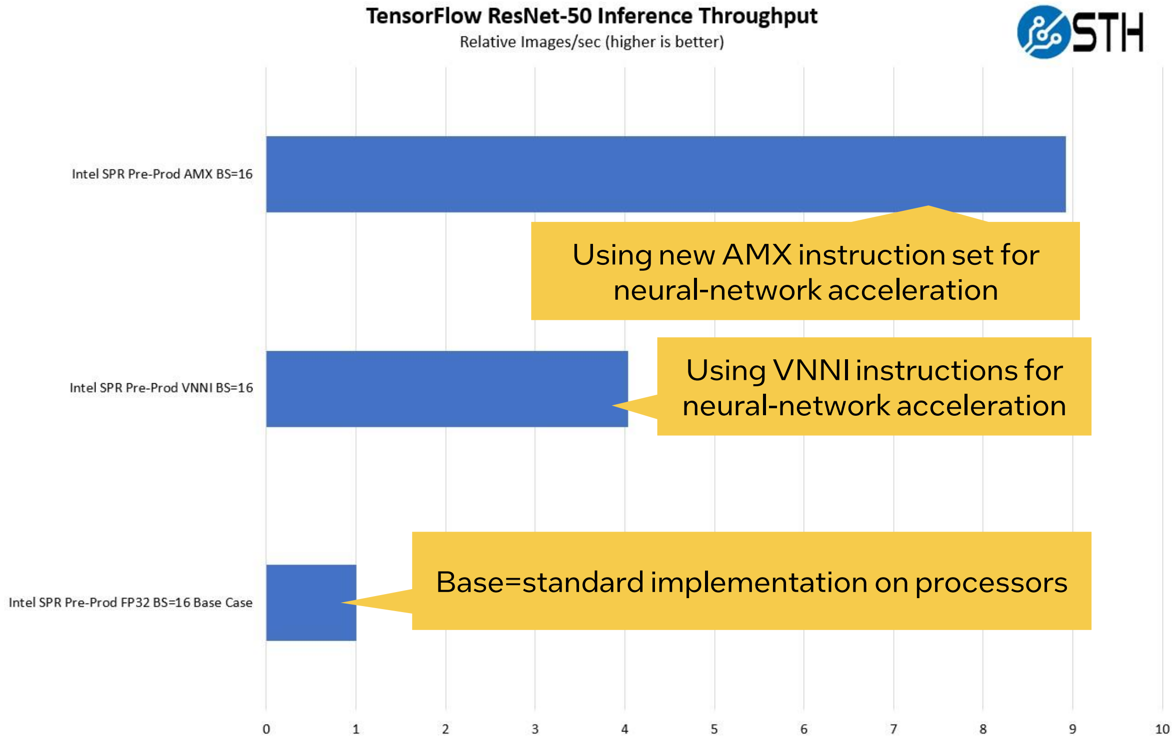
Benefits of Accelerators and Custom ISA

Simple instructions < specialized instructions < specialized accelerators

- Get the same work done using fewer threads
- Get higher performance per server



Source: https://www.servethehome.com/intel-quickassist-in-ice-lake-servers-what-you-need-to-know/3/

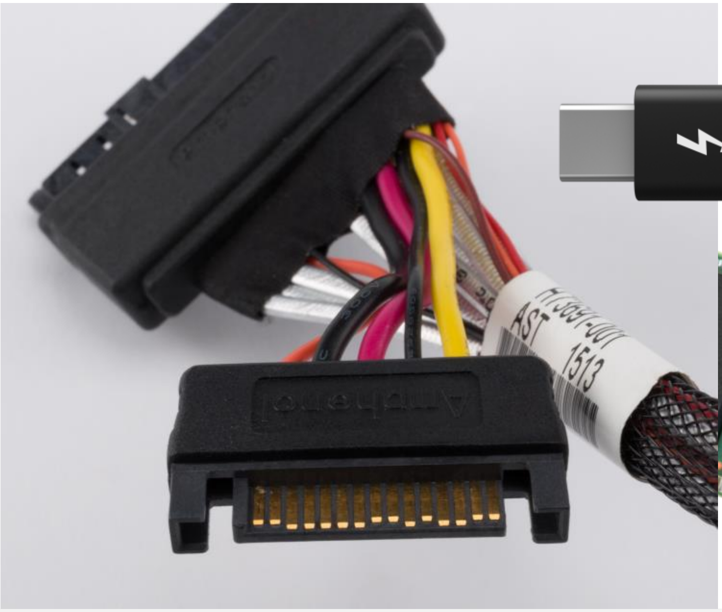


Source: https://www.servethehome.com/hands-on-with-intel-sapphire-rapids-xeon-accelerators-qct/3/

Input/Output Improvements

Once upon a time...

- Processor + memory
- Keyboard + display + disk (maybe)



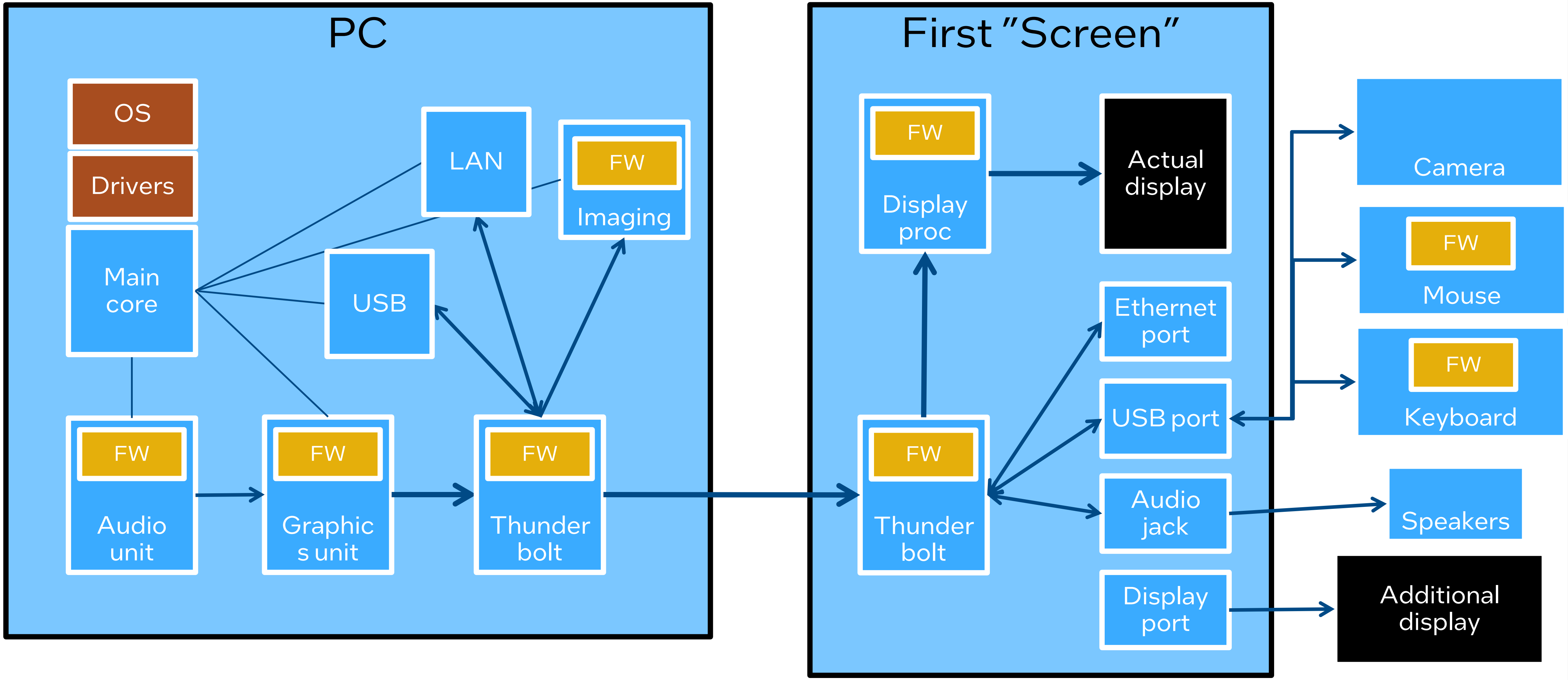
Today

- Keyboard + mouse + touch
- Cameras, microphones
- Ethernet and WiFi
- Disks – on NVMe, SATA, USB, ...
- Headsets on USB, Bluetooth
- External GPUs on Thunderbolt
- Display via HDMI, DV, Thunderbolt

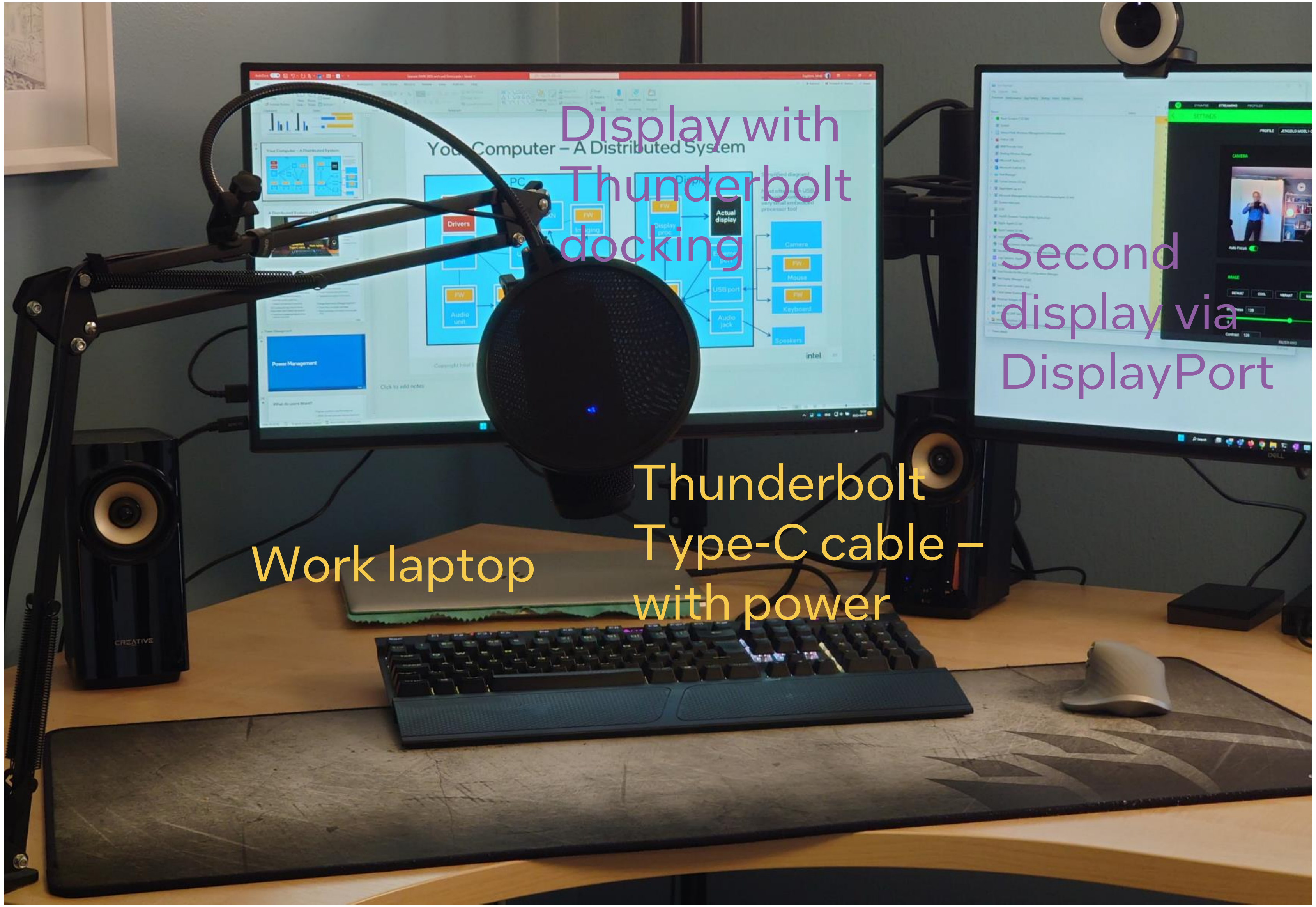
Trends:

- Flexible channels (USB-C)
- Smaller physical connectors

Computer Setup = Distributed Systems



A Distributed System at (My) Home (Office)



Attached to screen:

- Power – which is fed to laptop over USB Type-C
- Keyboard
- Mouse receiver
- Speakers
- Microphone
- Camera
- Headset receiver
- Second display

Summary: Long-Term Computing Trends

Processor cores are still important…

- … but other product aspects are just as important

Computers incorporate more and more diverse functionality

- Cameras, audio, graphics, …
- Always connected to networks

I/O is becoming much more important and takes up space

- Feed the processing engines from memory and disk

Processing performance improvements from specialization

- Graphics processors
- Fixed-function accelerators
- Tailored processor core sizes

Integration and disaggregation

- More fits in a single package
- Each package no longer just a single chip

Power Management

What do users Want?

Higher system performance

... With lower power consumption

... Giving off less heat = no fan

... With longer battery life

... Weighing less

NOT all that easy to do!

Power Efficiency come from Many Sources

- Manufacturing process
- Circuit design
- Computer architecture
- System optimization
- Power management

Where Does the Power Go?

$$P = P_{dynamic} + P_{leakage}$$

Total power:

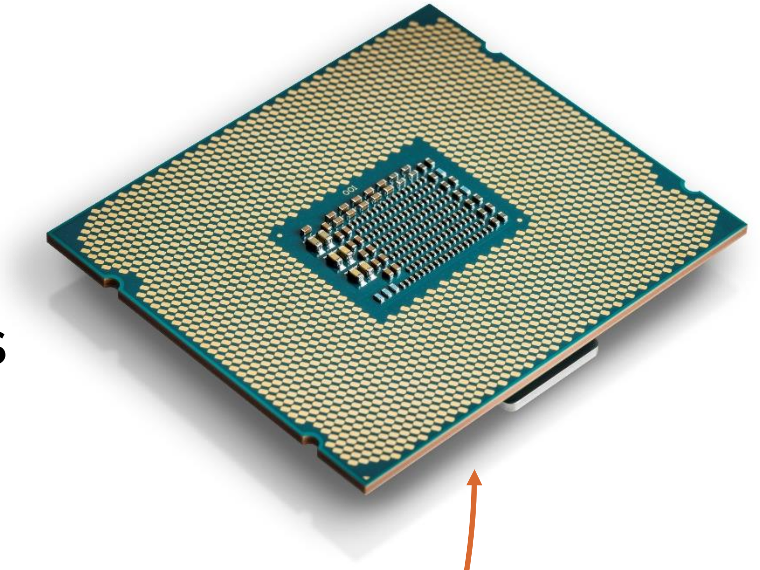
- Dynamic power during actual switching
- Leakage power from just being powered-on

$$P_{dynamic} = CV^2 f$$

Dynamic power:

- Basic capacitance
- × Voltage squared
- × Frequency
- Note that V affects f_{max}
  - Raising the frequency requires raising the voltage

Better Process and Circuits



Process technology and circuit design advances

- Transistors that use less power individually
- Lower drive voltages
  - Processors used to run on 5V, then 3.3V, now down to < 1V
    - Interesting side-effect: for a 100W power consumption, we have to feed 100A+
    - Approximately half of all “pins” on a chip package are for power distribution
- Lower leakage power
- Faster switching between different power modes and frequencies

All things equal, the same design on a better process
= lower power or higher frequency at the same power

Power-Efficient Computer Architecture

Reduce "wasted work" in the chip

- Clock gating – shut off clock
  - Removes dynamic power
- Power gating – shut off power
  - Removes static power (leakage)
- Over time, gating has applied to ever smaller parts

Power states:

- Settings for frequency, voltage, on/off, ...
- Subsystems are set to lowest possible state to save power
- Increasing the number of controllable units and the number of steps

Processor core and pipeline design – trade performance vs power

- Different core designs hit different trade-offs
- Many slow cores, a few fast cores, or a mix?
- *(As discussed earlier)*

Cache and memory system

- Cache hit = lower power than memory access
- Faster external RAM costs more power
- Bigger caches cost more processor power, but might reduce overall power consumption

Apply special-purpose accelerators

- Specialized compute is typically more power-efficient, if it gets used

System Optimization and End-Device Design

Overall system design choices

- Display size, technology, resolution, update frequency, brightness, …
- Battery size
- Choice of processor variant
  - For example, use a slower but lower-power variant or a faster but higher-power from the same family? What is right for the specific market being targeted?
- Memory choice
  - LPDDR (Low-Power DDR) vs regular DDR, speed rating
- Speed of wireless functions
- Cooling efficiency
- Available ports

Not easy for an end-user to grasp all the details

Certifications like Intel evo: provide consumers with an indicator to look for, and provide computer makers with advice to design better systems

Hardware Control Points & Sensors

Hardware continously adds more control points to reduce waste:

- *Per-core* voltage and clock-frequency adjustments (used to be per chip)
- More power states in more devices
- Faster changes to power states (off->on, clock & voltage scaling)
  - Note that going to low power state is not free - takes time to power or clock back up to full speed, operations take more time to complete

Sensors multiply across the chips and system

- Power levels
- Thermal levels – very important limiter for performance

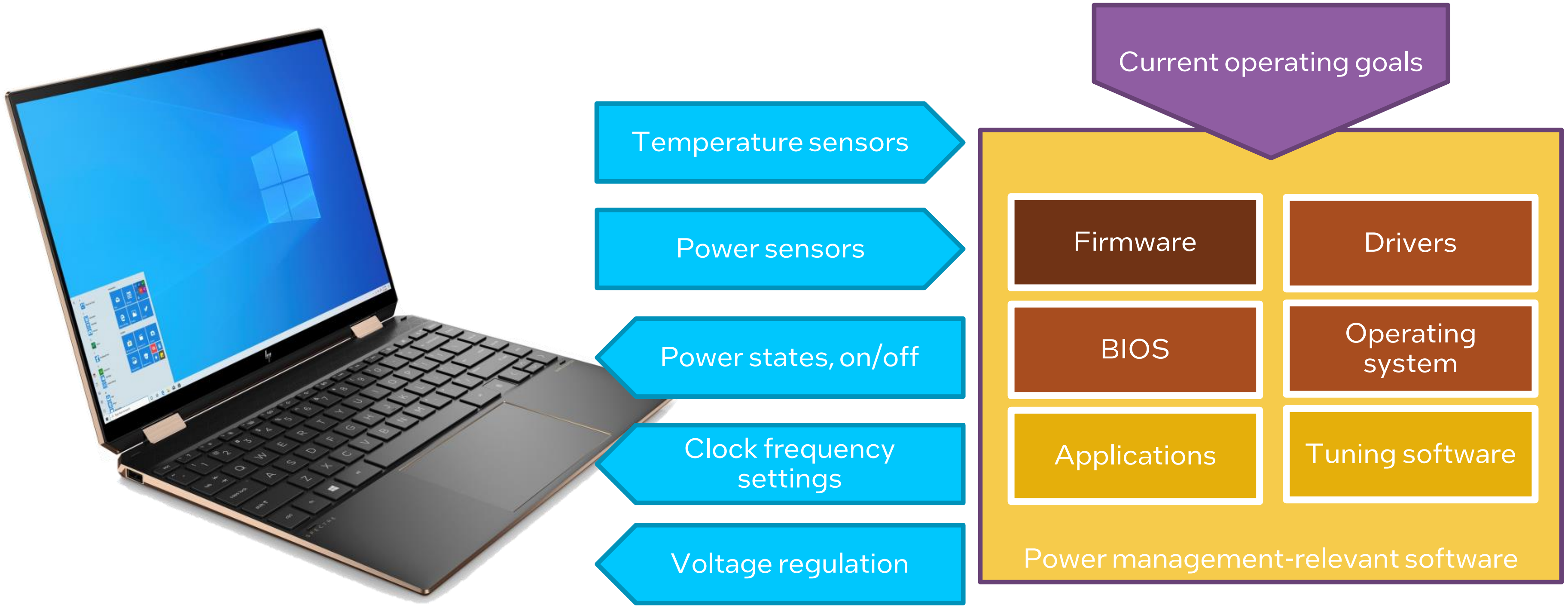
All of which come together in a power management unit (or units)

Power Management Software

Given that we have done our best in architecture & silicon...

The **biggest lever** we have today to improve power/performance is the power management software

- Control feedback loop implemented in hardware, firmware, and software – driving power states and gating



Power Management Firmware and Software

Optimize performance

- Profile current load
- Balance power draw vs user experience
- Allow higher performance if temperature and power availability allows it
- Select the right core to run a workload on for optimum results
- Set power/performance operating points

Sleep & wake-up

- Put system into deeper sleep
- Power off and on units in the correct order, wait until operation is stable

Avoid disaster

- Throttle to avoid drawing too much power from the platform
  - Each chip has a design limit the rest of the computer expects it to adhere to
- Throttle to avoid overheating the chip

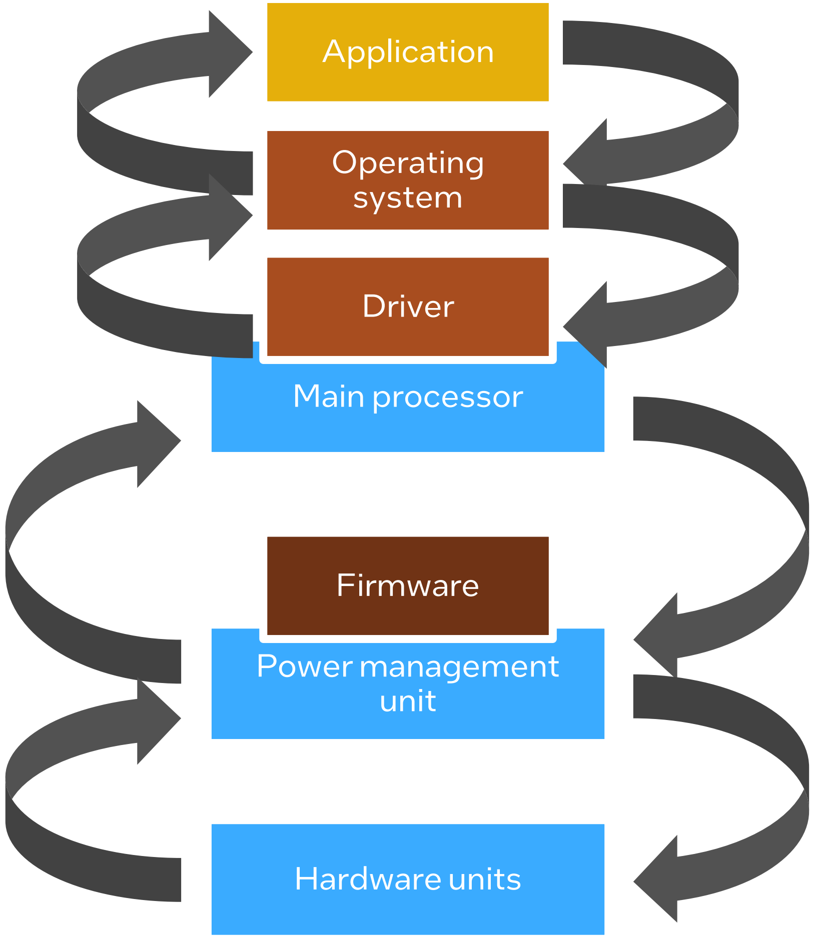
Layered Optimization and Goal Setting

Operating system (OS) will ask power management hardware to go to certain states based on its idea of the current load

- ACPI states: "active", "sleeping", etc., for processor, devices, and global
- Applications can give hints to the OS about what it wants from power control

Power management firmware

- Make quick adjustments based on the current measurements
  - Adjust clock frequencies and operation every ms or so
- Responsible for sequencing sleep, nap, hibernate states



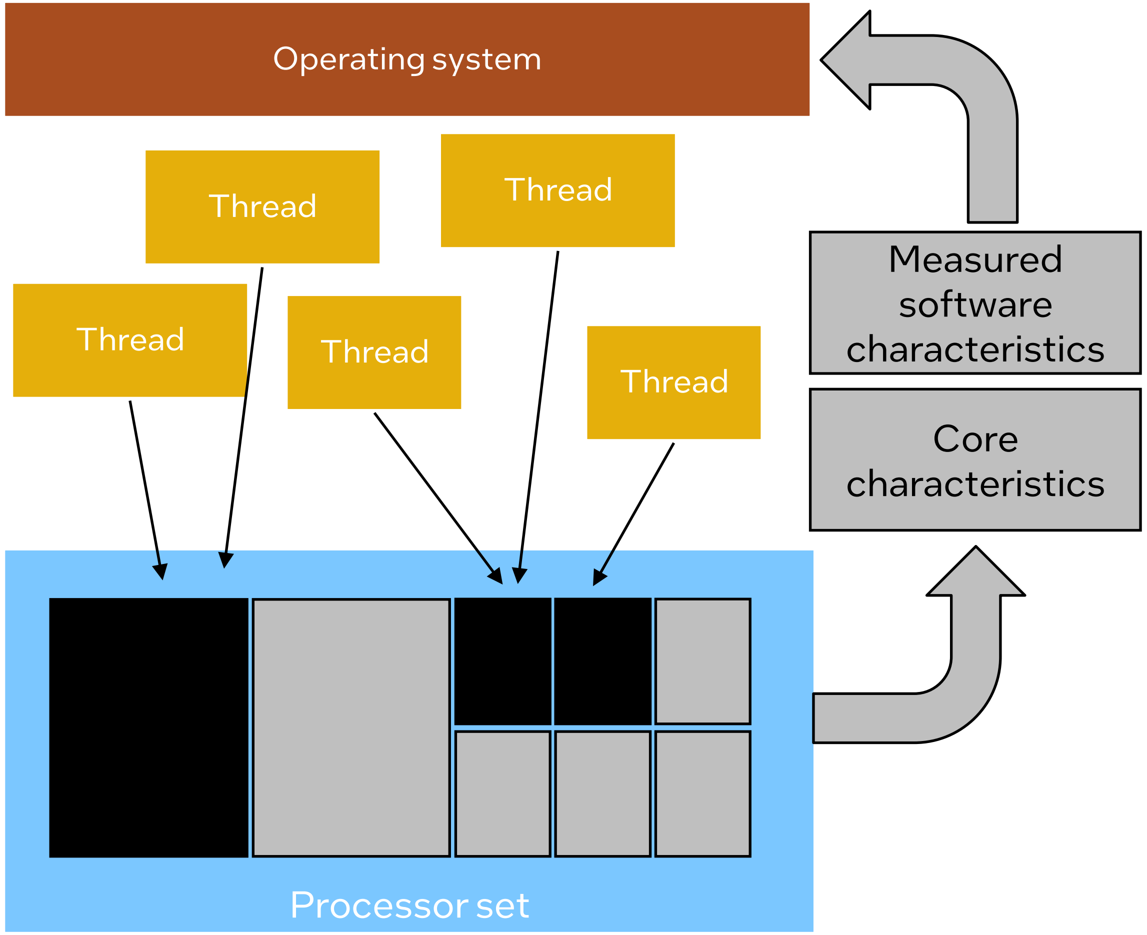
Scheduling for Heterogeneous Cores

Heterogeneous hardware complicates scheduling

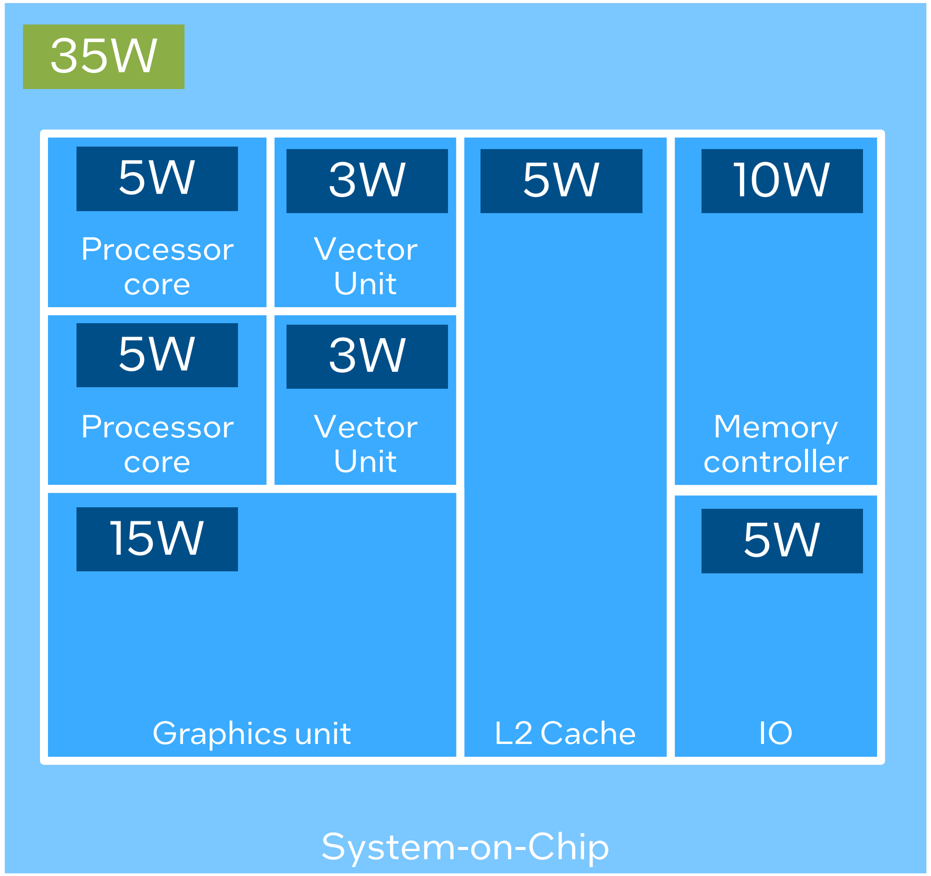
- Activate cores in a good order
  - Ex: High-performance or low-power first?
- Allocate threads to the most suitable cores
  - Ex: some software gains more than other from a fast core over a slower core
- Race-to-sleep or slow-but-steady?

Hardware help to OS necessary

- Specify core characteristics
- Report software characteristics
  - Instruction types used, memory bandwidth, ...



Power Management: Max is not Sum of all Max



Hypothetical chip, rather simplified

Fictional example for illustration

Total chip power allowed = 35W

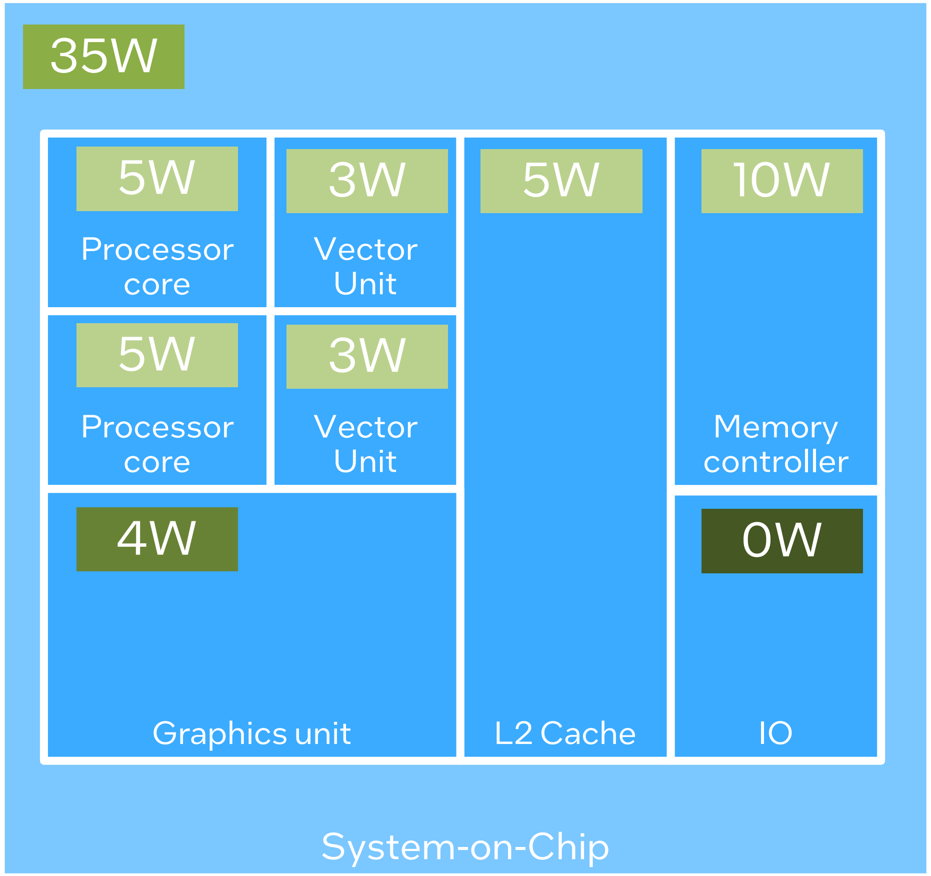
- Dictated by heat sink, power supply, and market segmentation

Total max power = 51W

- Throttle one part of the chip to allow others to run at full speed

Power management needs to keep the power inside allowed bounds

Power Management: Set According to Workload

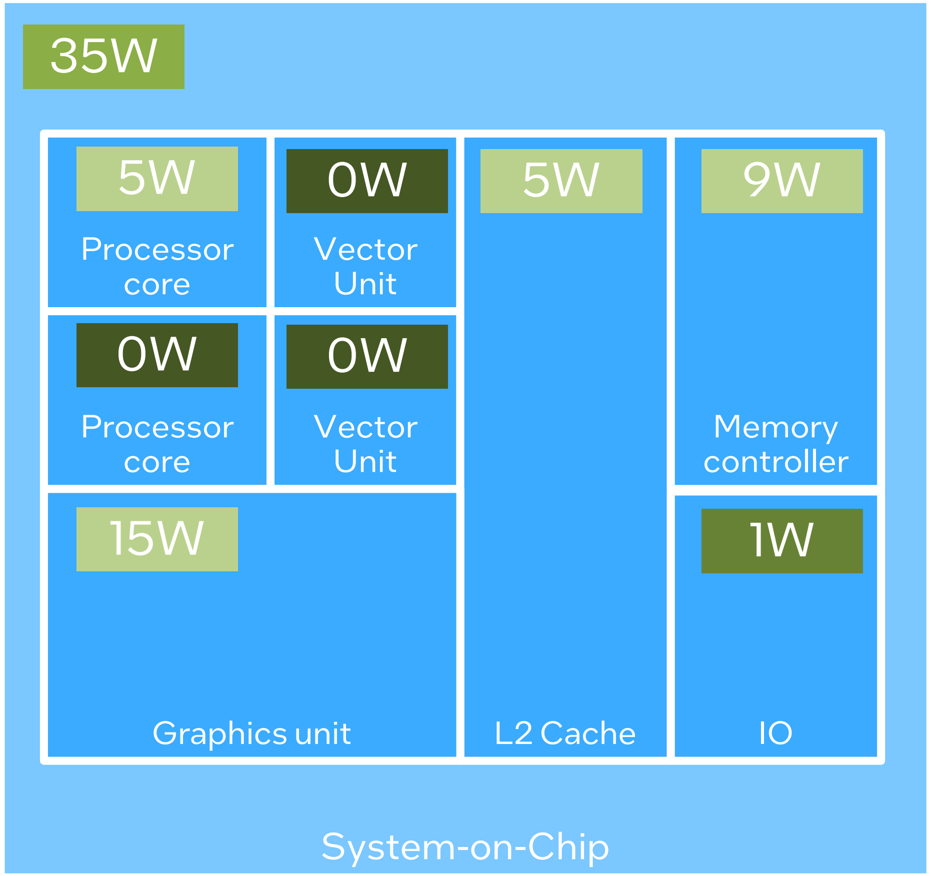


Hypothetical chip, rather simplified

Compute-focus:

- Power up cores, memory, and vector units
- Throttle graphics to make room
- Turn off IO, we assume we run from memory

Power Management: Set According to Workload



Hypothetical chip, rather simplified

Gaming:

- Graphic processing take priority
- Run one processor core at full speed – latency matters more than throughput
- Disable vector units – such work is now on the graphics unit
- A bit of IO needed for sound and chat
- Memory controller busy - but cannot be given full power since that would exceed the global limit

Setting the trade-offs right is tricky

Actual performance can be very different from theoretical peak performance

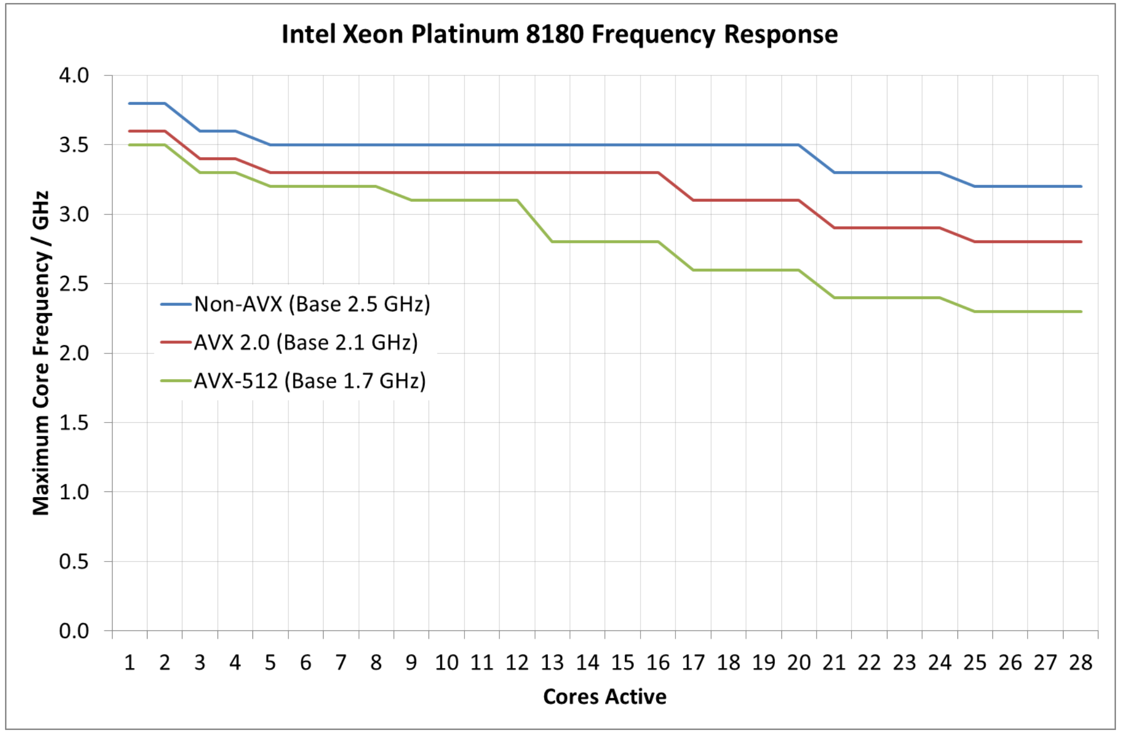
Note "Turbo" Processor Speed and Multicore

Processor speeds typically defined:

- Base frequency
- Max/turbo frequency
  - *Might be several levels of turbo*

Processor speeds vary all the time

- Use only a few cores = clock higher
- Use many cores = clock lower
- Using dense units like AVX = clock lower
- Both power & heat can limit the speed



https://www.anandtech.com/show/11544/intel-skylake-ep-vs-amd-epyc-7000-cpu-battle-of-the-decade/8

Note: Power, Heat, Performance

Goal = Maximum performance

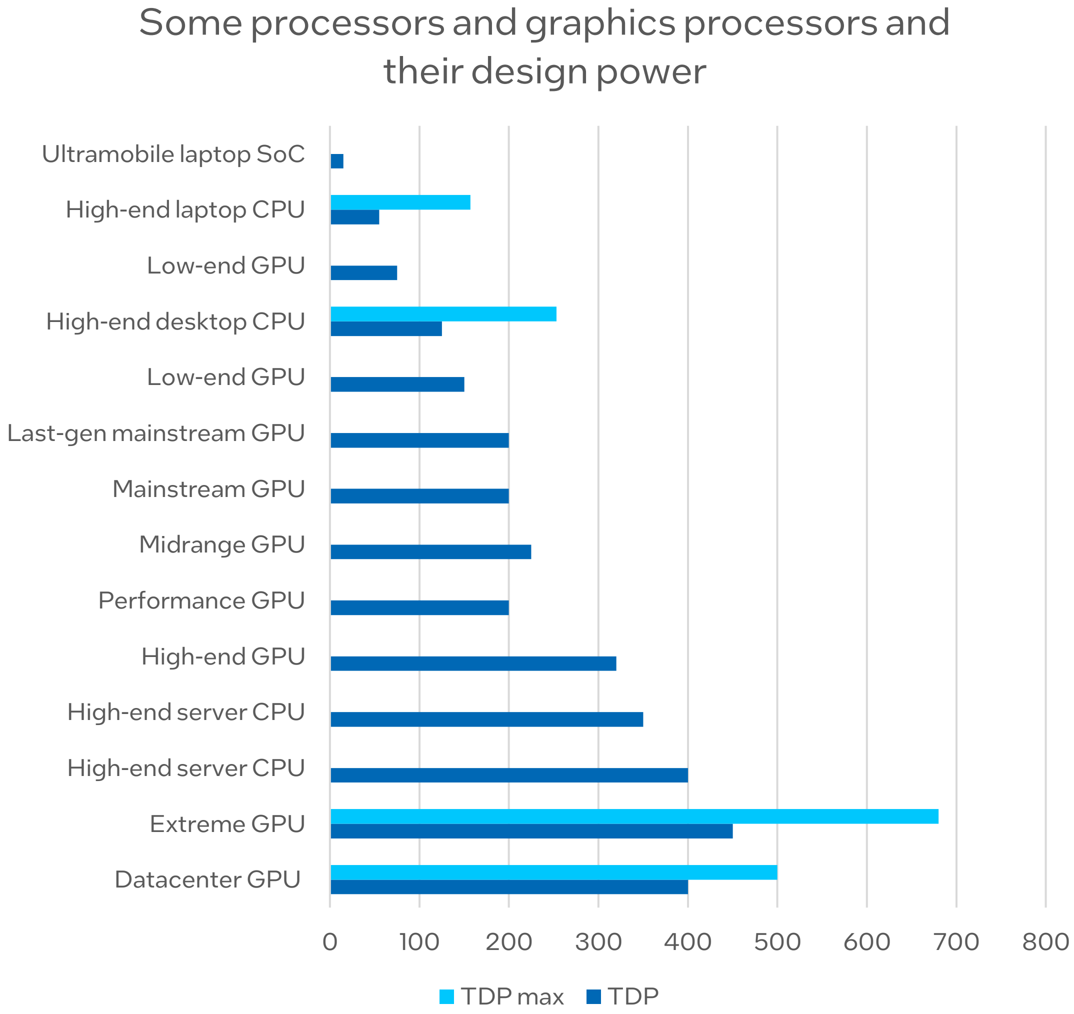
- More cores
- Higher clocks (easy cheat)
- More complex hardware
- = higher power & more heat

Sustained performance

- Datacenter, gaming
- Heat management key design factor

Burst performance

- Laptop, phone – benchmark on short runs



Importance of Heat ("Thermals")

Noctua* DH15S cooler, image from https://noctua.at/en/press-images/NH-D15S

Temperature is often a limiting factor

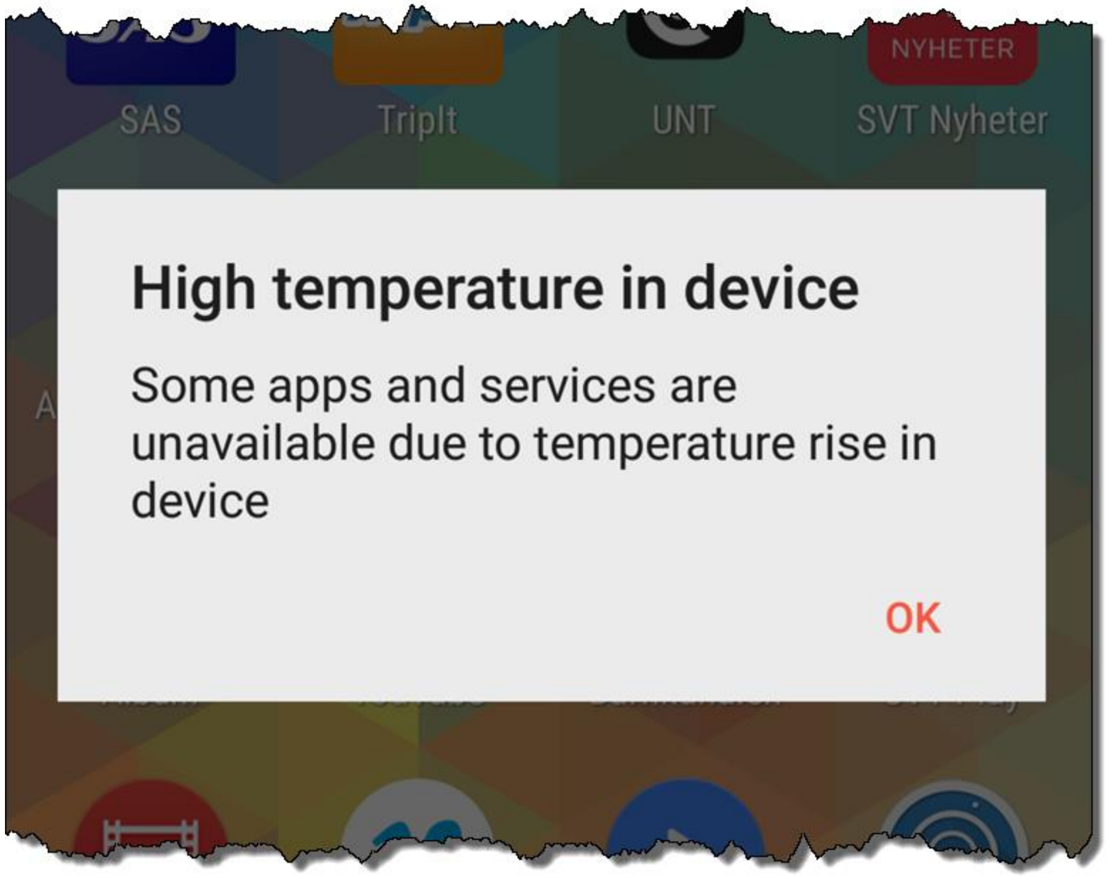
- Energy and power can be available…
- … but hardware throttles to avoid overheating
- Typical max temp for a chip is 95° to 100° C (!)
- Powerful cooling allows sustained high clocks
  - Search for "overclocking with liquid nitrogen"…

Cooling a crucial part of system design

- Heat sinks
- Fans and airflow (getting more important)
- Liquid coolers
- Affects memory and storage as well as the core processors and chipset chips

If a computer does not seem to run optimally: check the fans, blow out dust, etc.

Example: Avoiding Heat Disaster with Sensors



An old Sony* Android* mobile phone playing YouTube* videos

- Phone was noticeably warm

This happened when (guessing):

- Screen was on
- WiFi pulling in data
- Processor & accelerators decompressing video streams at high resolution
- *Was a bit more than the package was designed to handle...*

*Other names and brands may be claimed as the property of others

Example: Heat is a Critical Performance Problem

Example:

PCIe 5 M.2 NVMe SSDs

- In theory: 2x bandwidth of PCIe 4
- Heat dissipation the bottleneck
  - *"1GBps = 1W of power"*
- Small form factor = challenge to cool

Kingston* FURY Renegade PCIe 4.0 NVMe M.2 SSD with heatsink

Image from https://www.kingston.com/en/ssd/gaming/kingston-fury-renegade-nvme-m2-ssd

Gigabyte* Aorus* PCIe gen 5 SSD with cooler, image from https://www.gigabyte.com/SSD/AORUS-Gen5-10000-SSD-1TB#kf

*Other names and brands may be claimed as the property of others

Summary

Power efficiencies come from silicon improvement, architecture improvements, system optimization, and power management

Chips are full of sensors and actuators used by power management

Power management is a nested dynamic feedback loop

Broken power management can literally fry a chip

Heat dissipation and power consumption limits performance

The Intel® Simics® Simulator

Also known as my day job

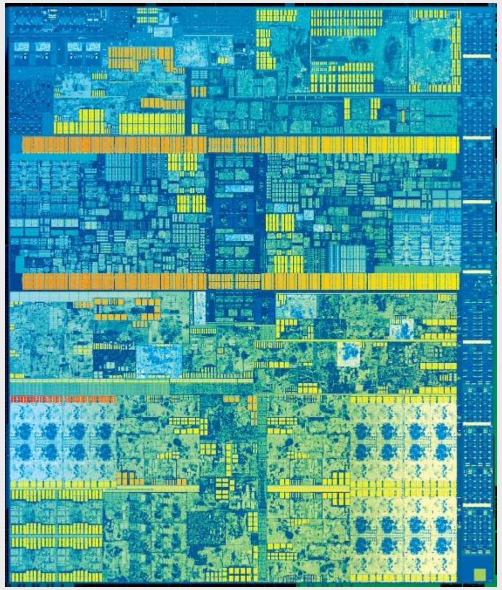
intel®

Why use simulation for software development?

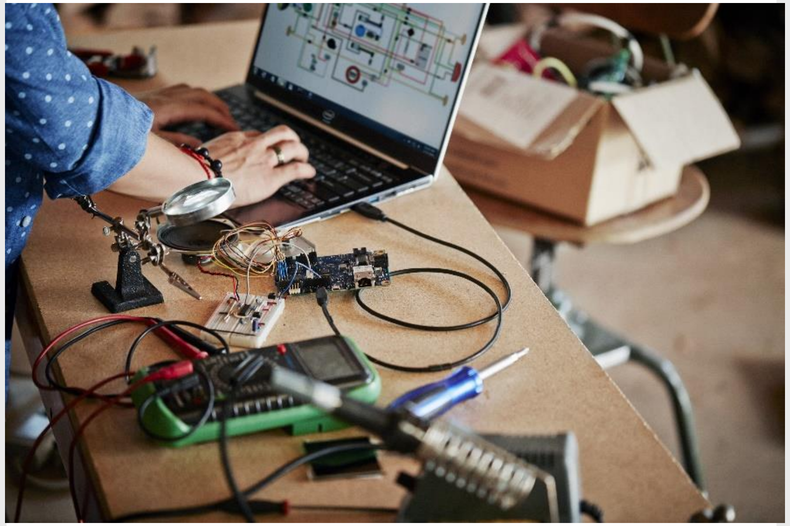
Hardware: A Hard Development Platform?

Hardware is Hard When it is in...

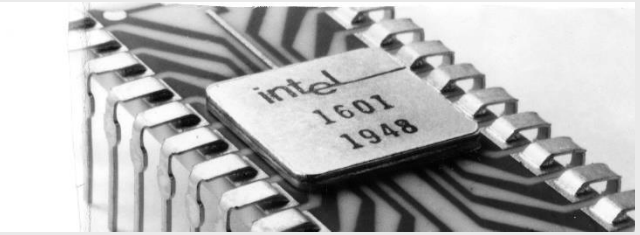
Not yet available



Flaky prototype stage



Not available anymore

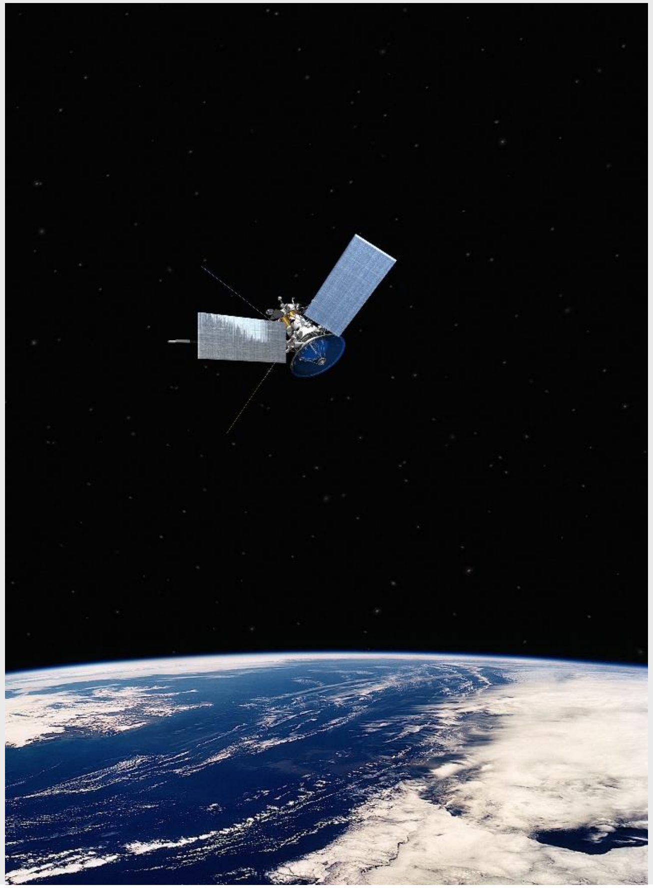


Hardware is Hard When it is...

Inconveniently large & complex

Dangerous to play with

Inaccessible & expensive



The Idea of a Virtual Platform

Run your software without the hardware – on a software model

About the Intel® Simics® Simulator

Simics® History

Development started in 1991

- Spin-off from research project
- Pre-silicon OS bring-up

Virtutech company founded in 1998

- Sun & Ericsson first customers

Acquired by Intel in 2010

- External sales via Wind River

Wide usage

- Intel-internal
- Intel ecosystem
- Embedded systems

Major milestones

- 2.0: Heterogeneous systems
- 3.0: Reverse execution & debug, 2005
- 3.2: Intel VT-X acceleration
- 4.0: Multi-threaded (coarse), 2008
- 4.2: Distribution, 2009
- 4.4: Eclipse GUI, 2010
- 4.6: TCF Debugger, 2012
- 4.8: Eclipse expanded, 2013
- 5: Multicore multithreading, 2015
- 6: More threading & integration, 2018

How it Works

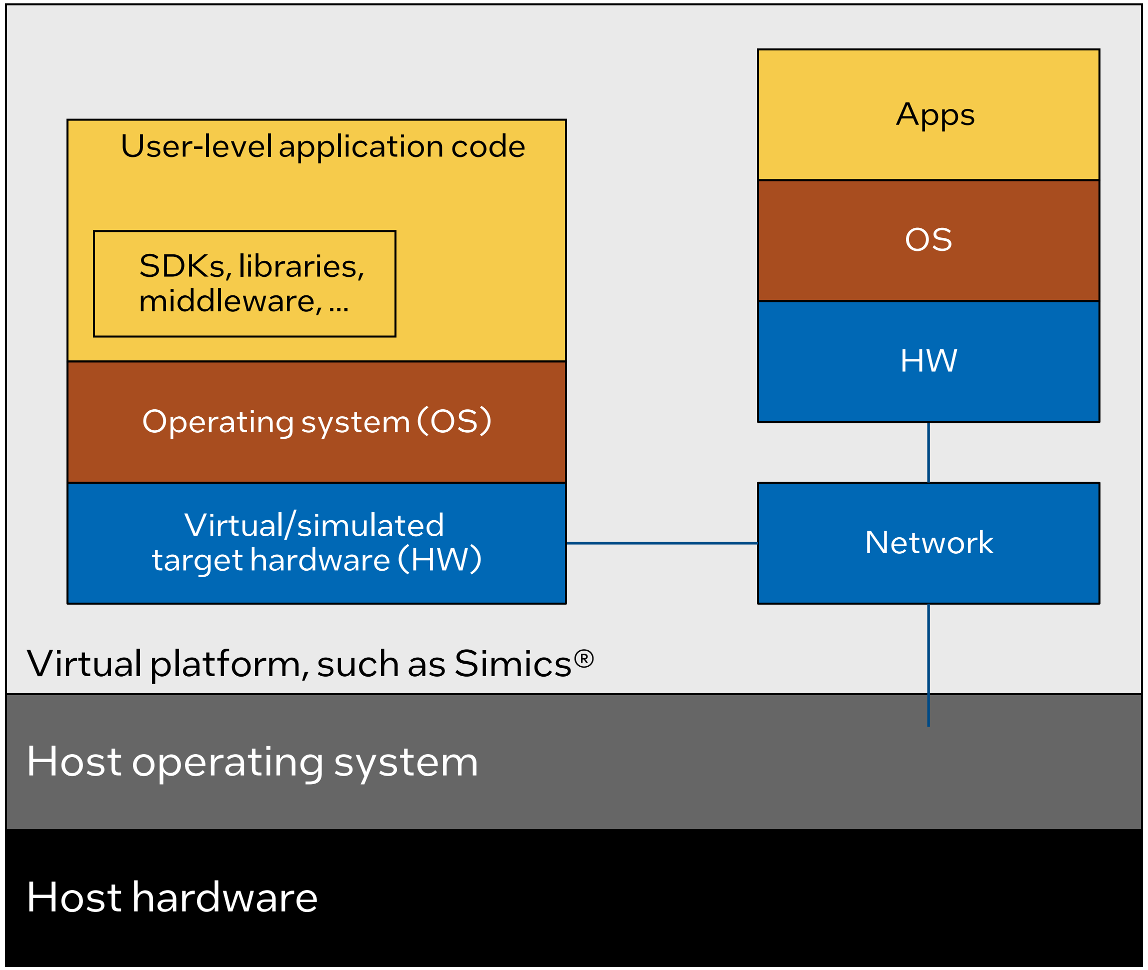
Full system virtual platforms

- Simulated (virtual) target hardware
- The same software as the physical system
- The software cannot tell the difference

Important properties:

- Fast enough to run real software workloads
- Simulate any computer system
- Single board, multiple boards, standard parts, custom chips, IO, networks, ...

Frees testing and development from the dependence on physical hardware



The Basic Idea

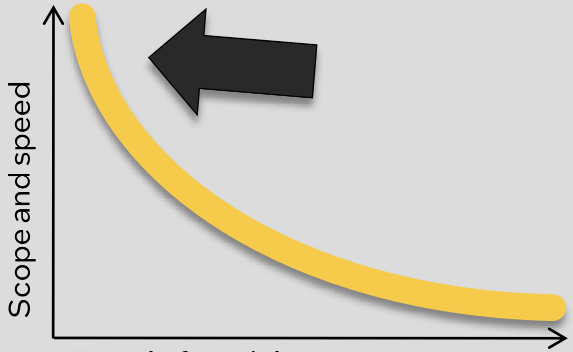
Fundamentally Simics is about **running real software on virtual hardware** in order to test & debug the software, the software-exposed aspects of the hardware, and the hardware design

“Software” can mean many things…

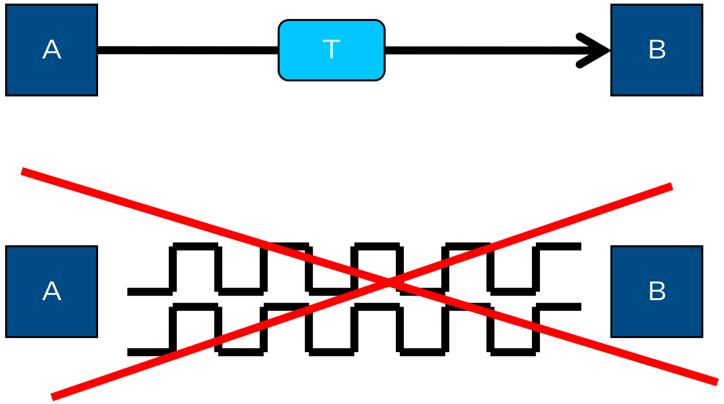
- **Firmware,** that is deeply hidden inside a chip
- **BIOS/Bootloader/UEFI,** that is used to boot the machine
- **Device drivers,** that manage hardware for an operating system
- **Operating systems**
- **Middleware**, providing services for other software
- **Applications,** that any programmer would write
- **Distributed systems**, software running across many separate machines
- From bytes to terabytes of code!

Simics® Simulation: Level of Abstraction

Goal: Fast & scalable simulation

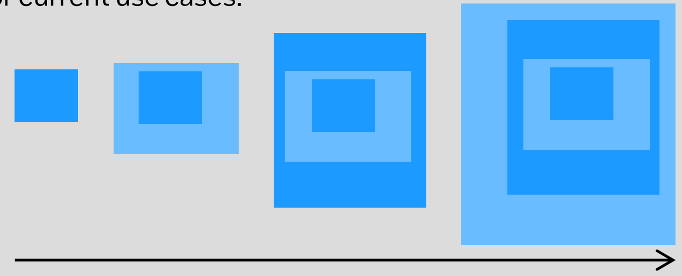


Transaction-level modeling (TLM)



Lazy and agile modeling

Build up the model piece by piece over time, as use cases materialize or become possible. Only model what is needed for current use cases.



Goal: run the real software

- User application code
  - Middleware and libraries
- Target operating system (s)
- Boot/BIOS/UEFI | Drivers

Target model includes all software-visible functional aspects of hardware, such as processor instructions, supervisor modes, device registers, interrupts, etc.

Model function & basic timing

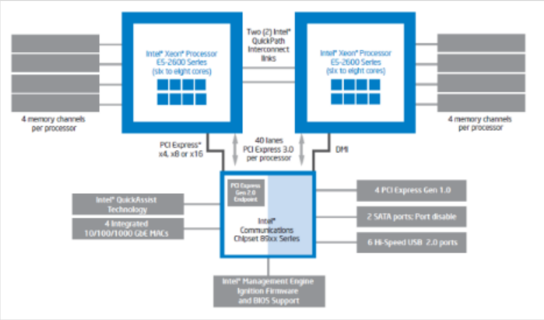
- Processor instruction set
- System memory map (not bus system)
- Device register interface
- Loose timing model
- Packet-level models of networks
- Event-driven simulation, not cycle-driven

Add timing and µarch when needed

- Processor simulators from designers
- Cycle-accurate hardware models
- Cache model (timing)
- Processor timing models
- Power models

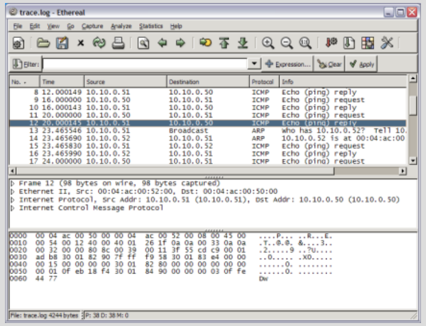
Virtual Platform Debug Features

Insight into all components

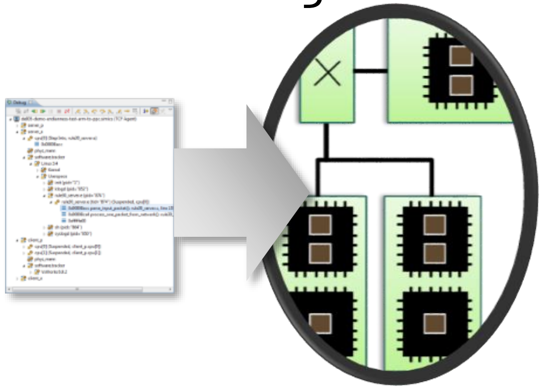


Synchronous entire-system stop

Trace anything



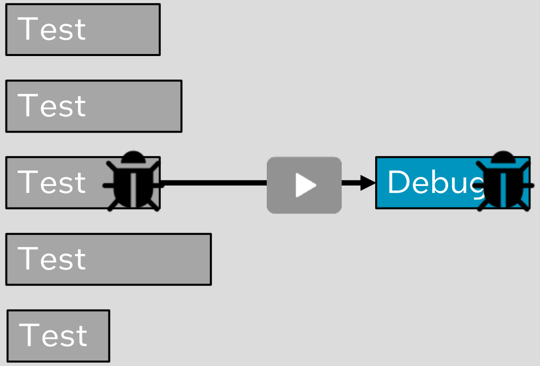
System-level symbolic debug



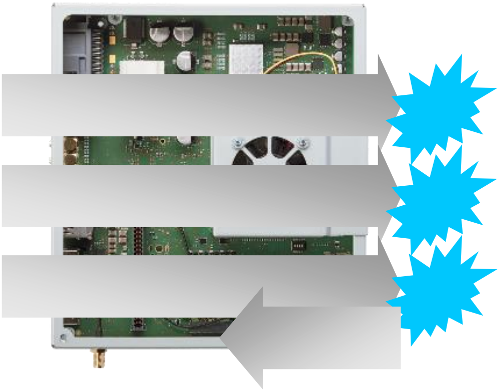
Unlimited powerful breakpoints

```
break -x 0x0000 length=0x1F00
break-io board.mb.sb.lan
break-exception int13
break-log "spec violation"
```

Record-replay debug



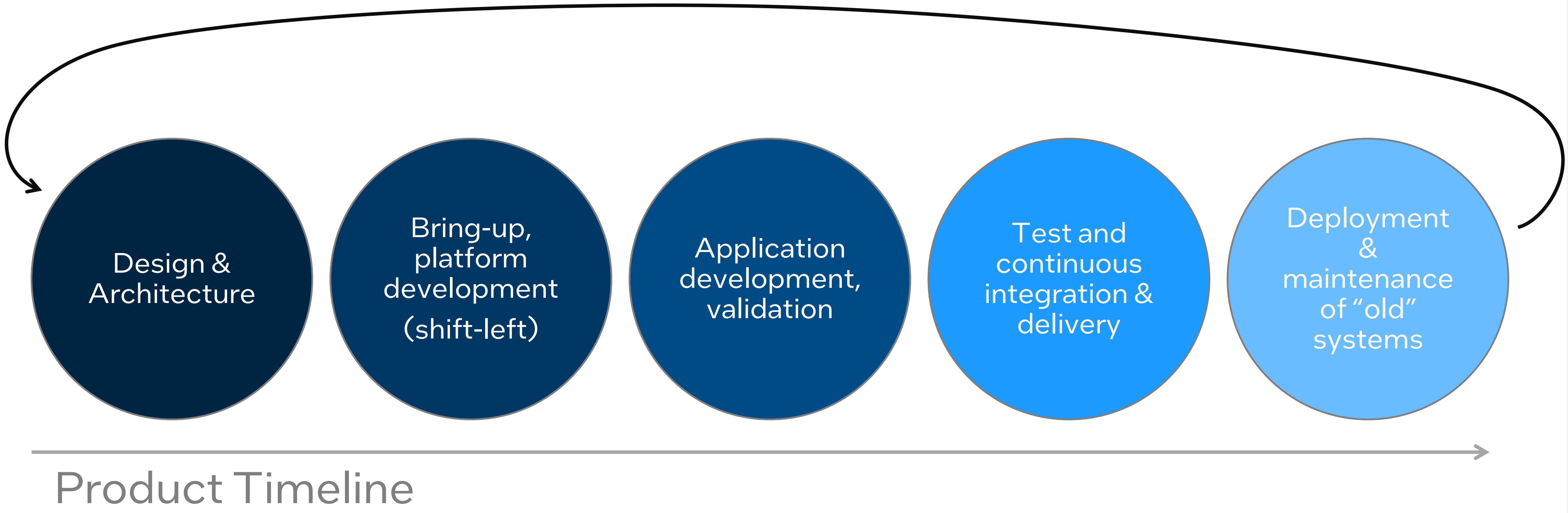
Repeatability & Reverse debug



Collaboration between developers

Simics® Simulator Use Cases

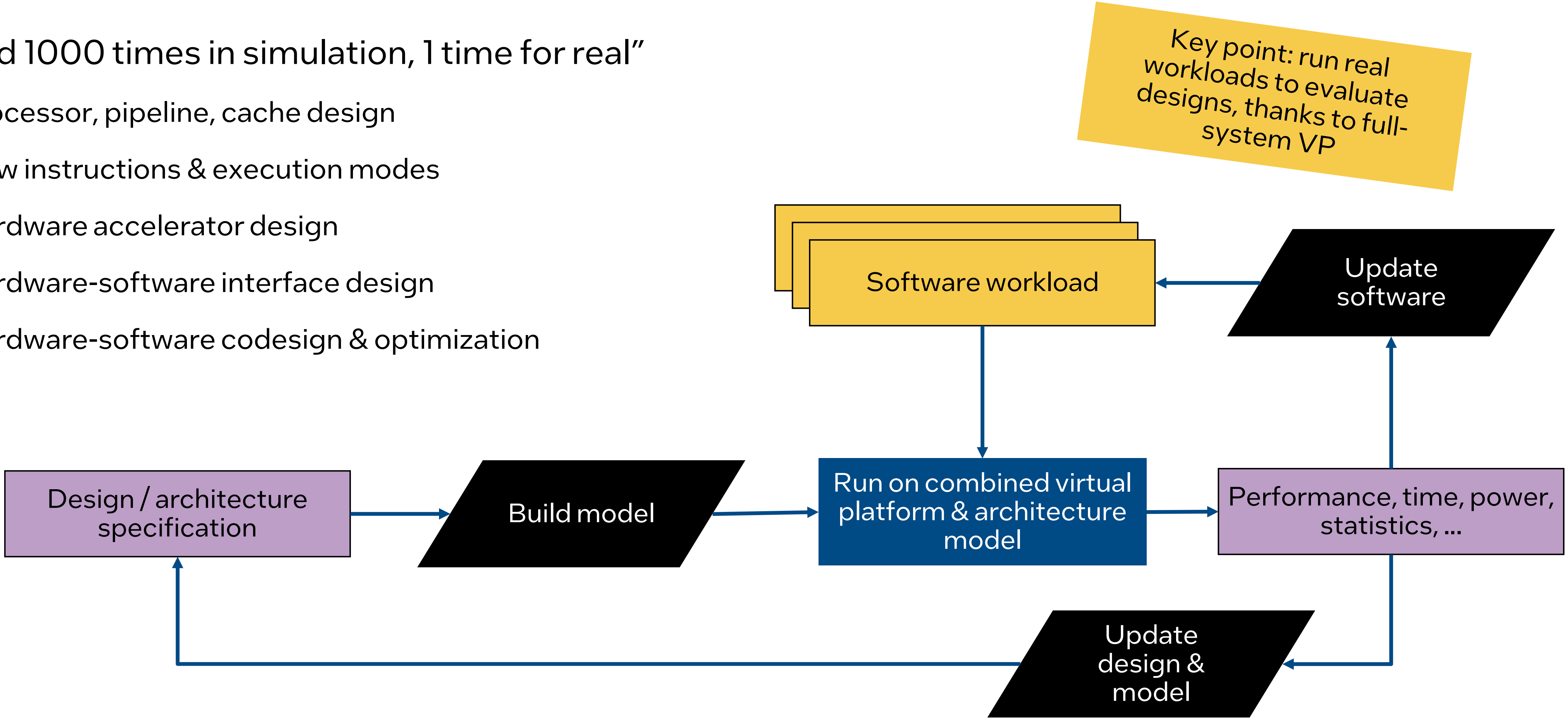
Virtual Platforms & the Product Lifecycle



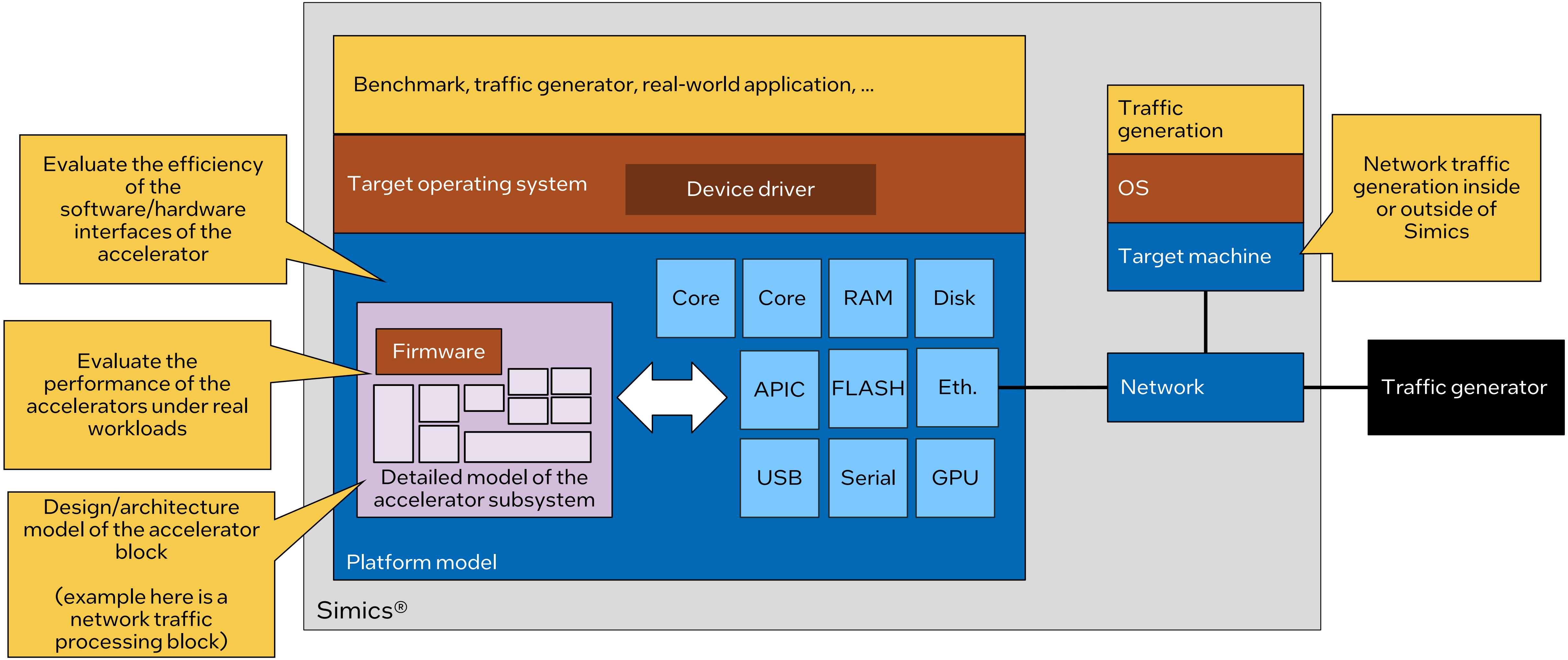
Computer Architecture (on Virtual Platform)

"Build 1000 times in simulation, 1 time for real"

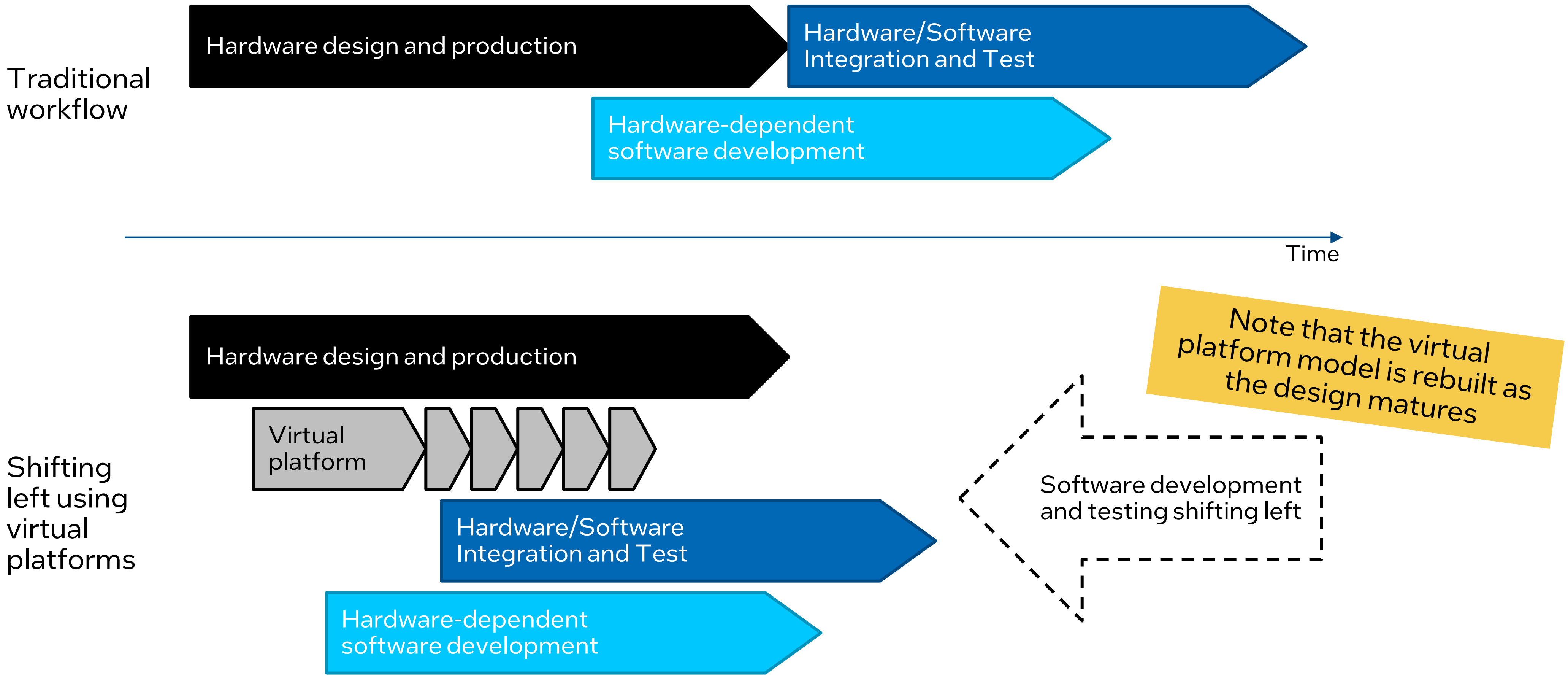
- Processor, pipeline, cache design
- New instructions & execution modes
- Hardware accelerator design
- Hardware-software interface design
- Hardware-software codesign & optimization



Computer Architecture: for Subsystem

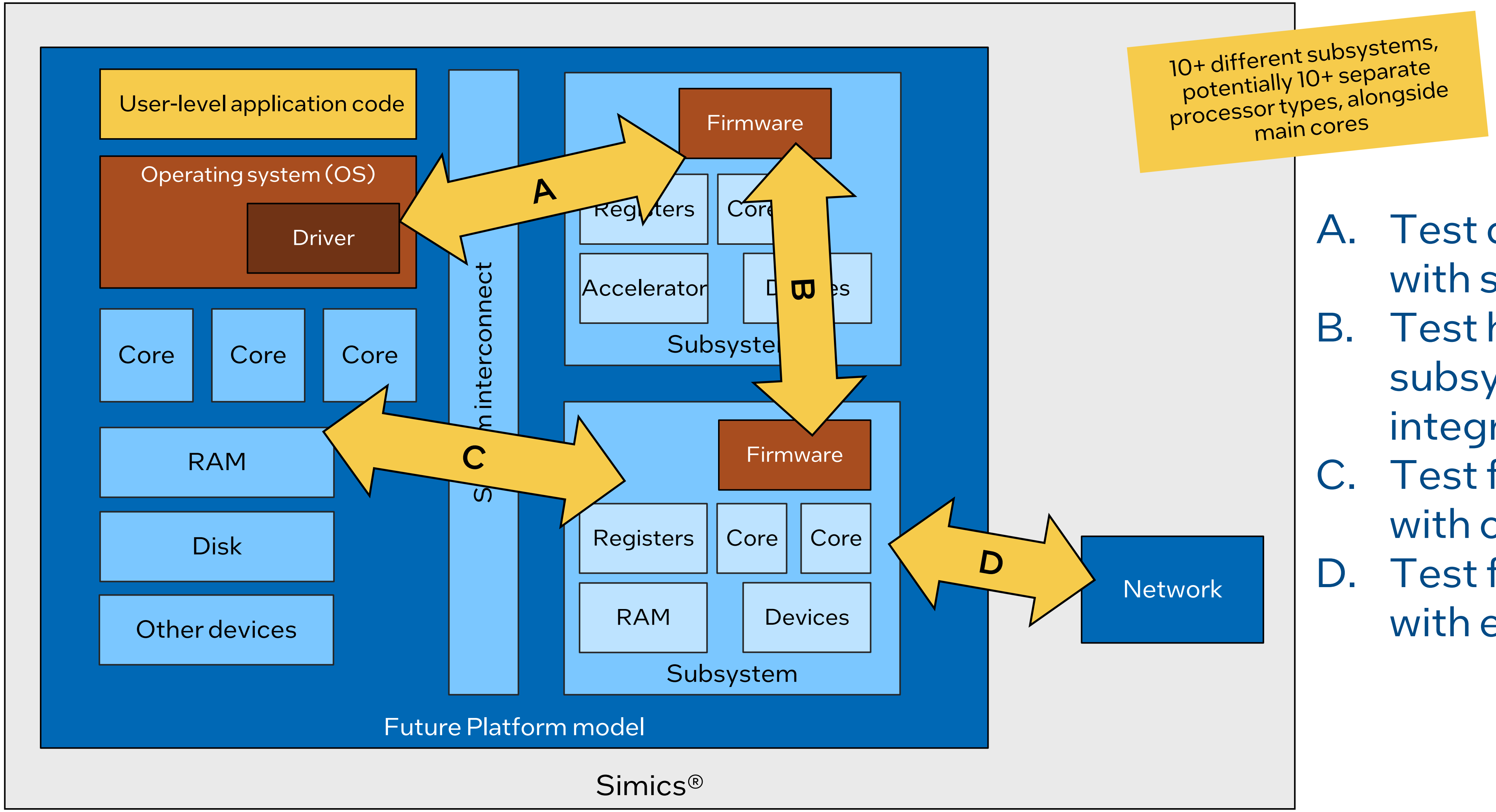


Shift-Left / Early Software Development



Classic case – Earliest examples from the 1950s

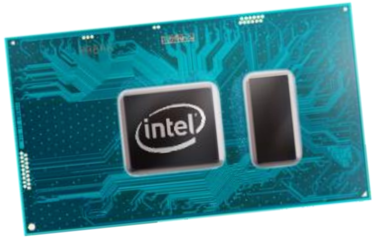
Shift-Left: Going into Details with Firmware



A. Test driver interaction with subsystem firmware
B. Test how different subsystems interact – integration is always "fun"
C. Test firmware interaction with other hardware
D. Test firmware interaction with external world

Shift-Left: With the Ecosystem

Silicon vendor builds basic code, makes sure the platform works



Chip/Platform

Board designer adds more components, ports more operating systems, validates additional functionality

(Custom) Board

Typically, this is a customer of the silicon vendor, a separate company

OEMs build on the boards to build complete products.

Digital twins

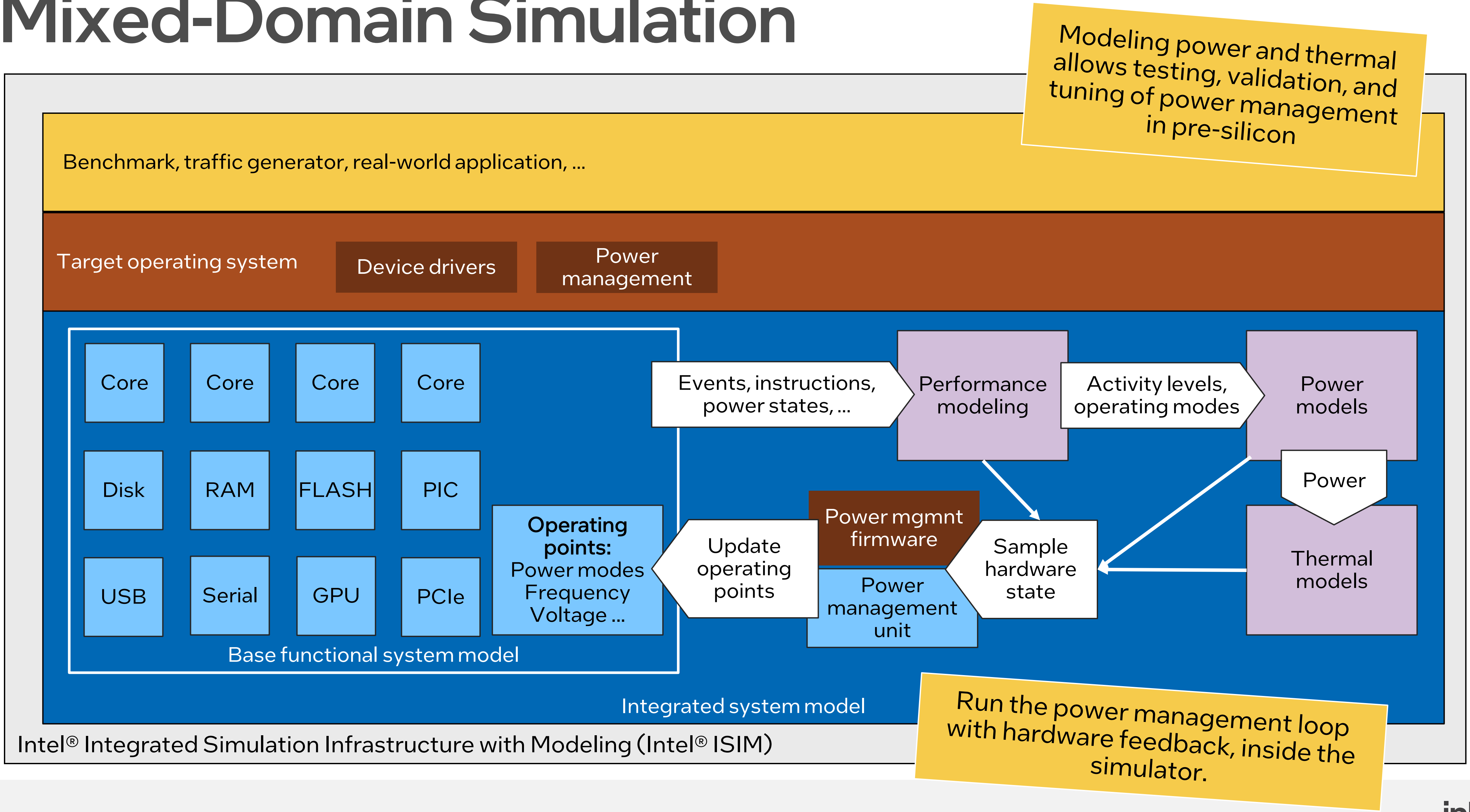
Virtual system integration

OEM Product

Could be the same company as the board designer, or yet another company

*Other names and brands may be claimed as the property of others

Mixed-Domain Simulation



Public Release of Intel® Simics® and Intel® Integrated Simulation Infrastructure with Modeling (Intel® ISIM)

Download and Learn More at

https://developer.intel.com/intel-isim