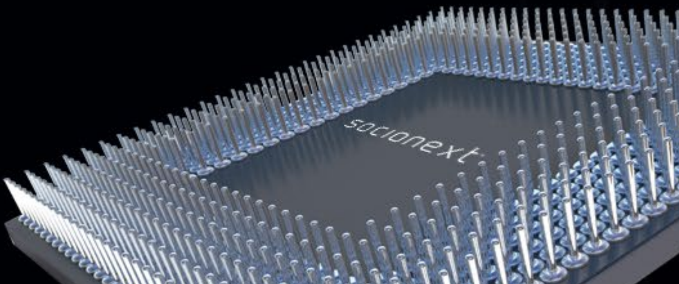
WHITE PAPER

Socionext's Solution SoC Approach to AI Acceleration in the Data Center



Overview

Most AI model training and a significant part of AI inference is now performed on general-purpose GPU computing devices. This is logical in an unconstrained environment of rapid evolution, where the flexibility of a general-purpose solution is more important than the efficiency of an optimized one. But there also are macroeconomic and microeconomic concerns that are causing the migration to more targeted, optimized solutions.

Rapid Evolution and Growth of AI Models

The last five years have been characterized by the use of large AI models within products and services, especially by the hyperscalers and 'new wave" automotive companies. During that time, the growth of AI model size, particularly for Large Language Models (LLMs), has been spectacular, as noted in the graph below.

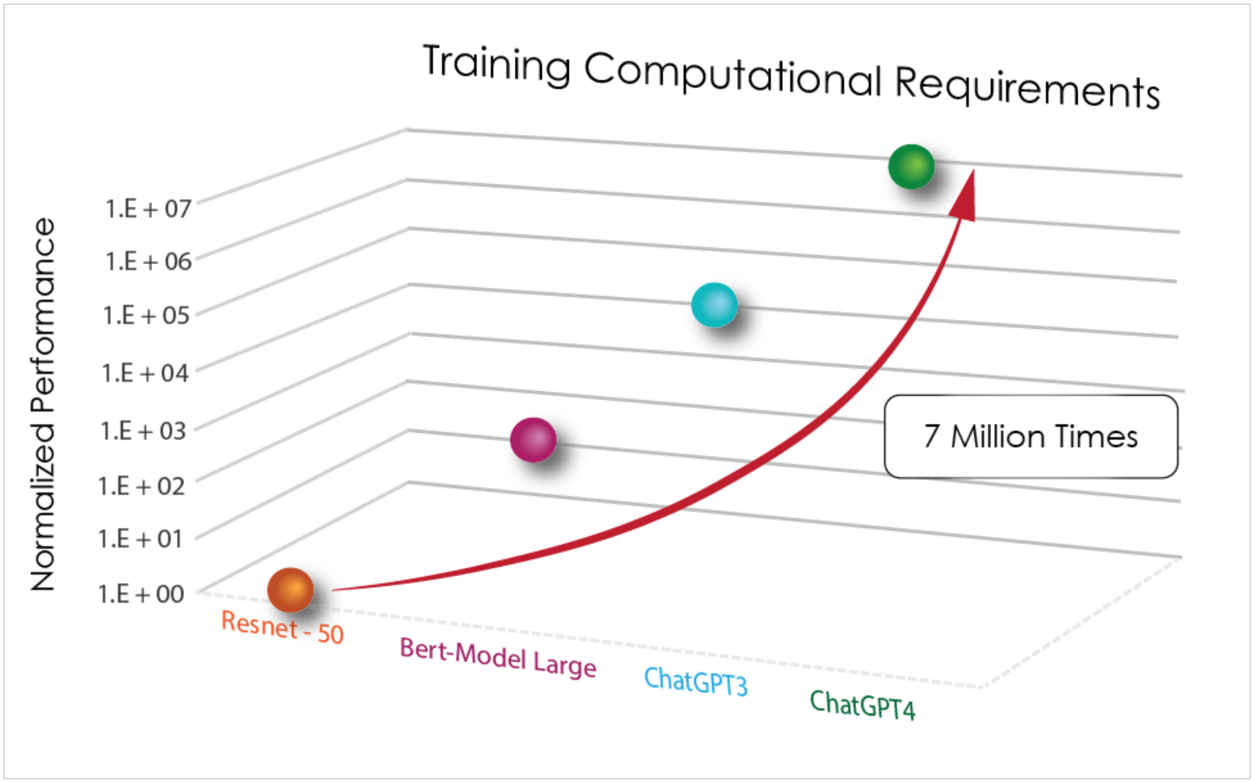


Figure 1: Trajectory of AI model training computation requirements [1]

[1] Created by SOCIONEXT based on:
Tom B. Brown et. Al. [2005.14165] Language Models are Few-Shot Learners (arxiv.org)
Shreyas Saxena et al. [2303.11525] Sparse-IFT: Sparse Iso-FLOP Transformations for Maximizing Training Efficiency (arxiv.org)
https://lambdalabs.com/blog/demystifying-gpt-3

AI Model Growth

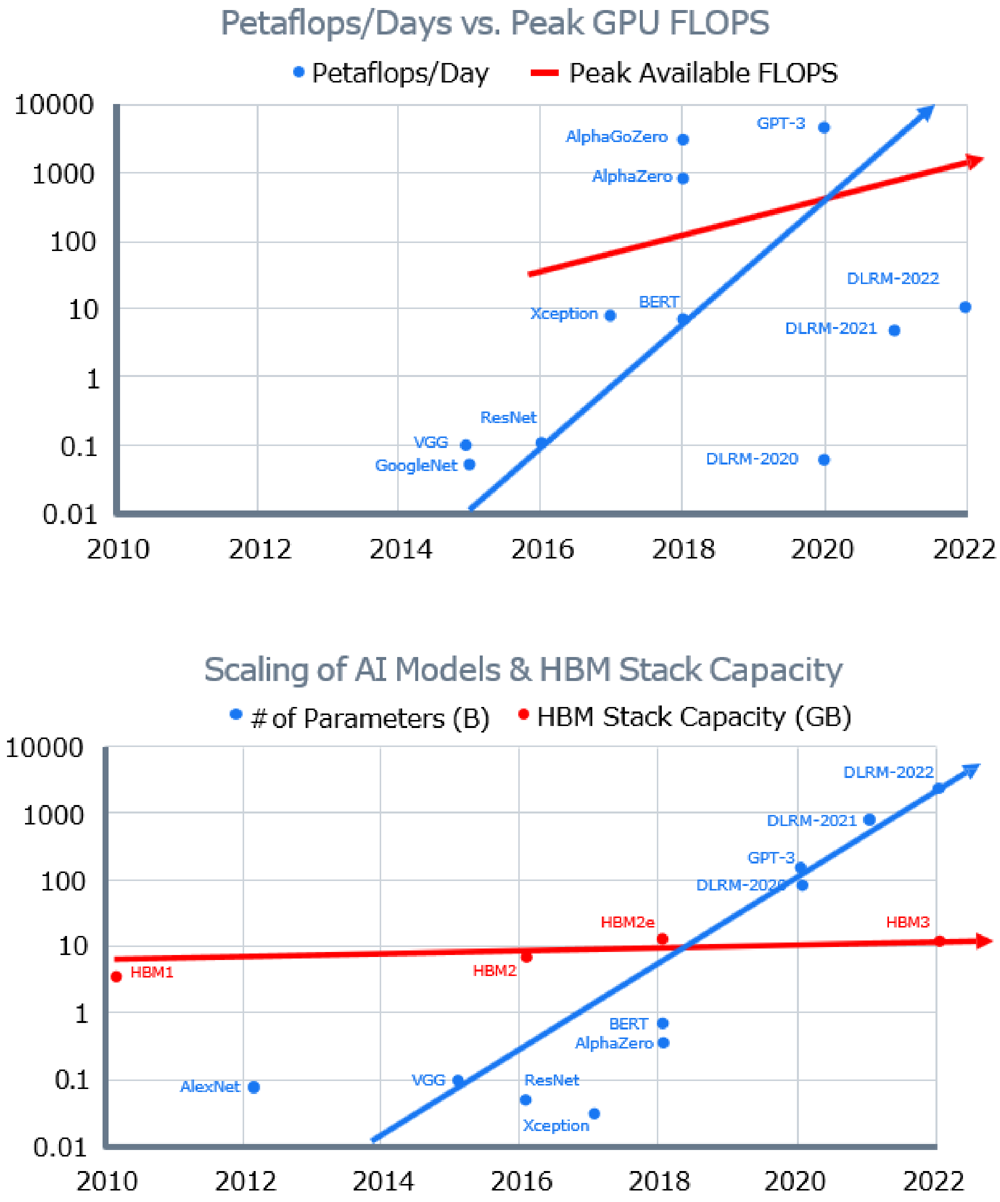


Figure 2: Evolution of AI model performance and scale compared to GP GPU [2]

The growth in model computation requirements is significantly greater than the improvements that the industry is making in GP GPU processing performance and memory capacity.

[2] Meta, AI Hardware and Edge AI Summit, October 2023

Interface specifications are particularly slow to evolve due to the number of parties involved in the drafting of standards and the need for interoperability, which results in a significant lag between technical capability and actual implementation.

This means that continued reliance upon existing approaches will exacerbate the trends towards higher power requirements, larger size and greater cost

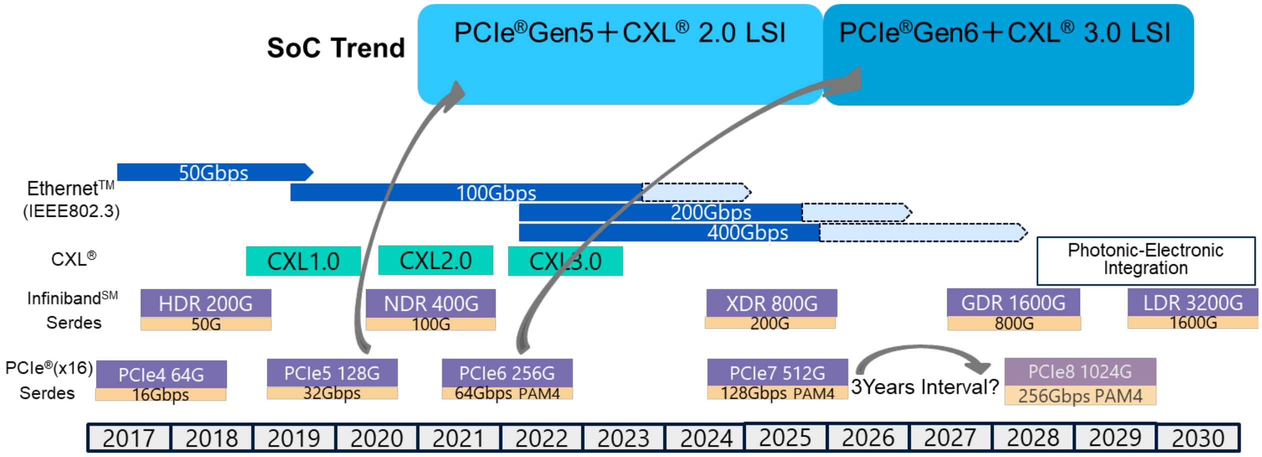


Figure 3: Inter-device interface evolution (Source: Socionext estimate)

Macroeconomic Challenges

Current implementations demand significant electrical infrastructure and large physical footprints. These impacts are intensifying rapidly, sparking public debate over priorities.

Electricity Consumption

For example, worldwide consumption of electricity by data centers is projected to grow from around 400 Terawatts (equivalent to the total power consumption of Germany) in 2022 to over 1000 Terawatts (equivalent to the total power consumption in Japan) in 2026. Contributing to this is the increased thermal power density associated with AI computing, which means that heat must be removed using sophisticated cooling systems that also require lots of power. The Electric Power Research Institute estimates that ChatGPT consumes 2.9 watt-hours per request, or 10x the electricity of traditional search queries. This rapid increase in power consumption puts pressure on electricity grids and challenges states' electricity supply decarbonization plans.

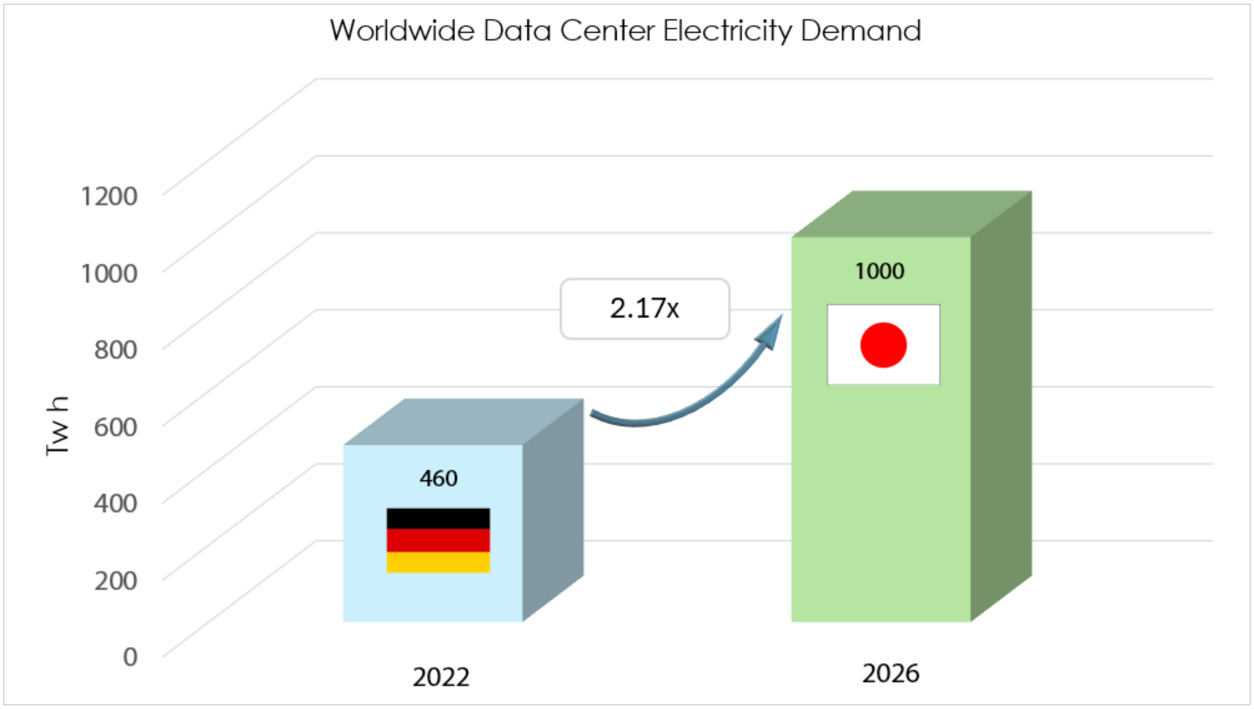


Figure 4: Worldwide data center electricity demand [3]

Less reported than the impact on electricity consumption, but equally challenging is the amount of space required for data centers, which often compete with residential property for land. AI data centers now cover more than one million square feet or 100,000 square meters in size. At an average apartment size of 1000 square feet or 100 square meters, this equates to one thousand apartments. In regions with housing deficits, this is becoming a source of conflict. Alternative sources of land can include rural areas less impacted by regional shelter requirements; but these areas often include limitations in infrastructure and a desire among residents to avoid extensive development.

Microeconomic Challenges

Beyond the macroeconomic issues, there are microeconomic challenges, including significant attention from the financial community regarding return on investment as the technology matures and is applied to commercial applications. The industry is confronting the reality that the power of AI must be harnessed at a cost that is economically practical for implementers and users. Initiatives such as the open-source Llama project from Meta accelerate the adoption of AI, and the commoditization of AI models creates pressure to reduce costs throughout the stack.

Macroeconomic and microeconomic challenges imply the need to significantly reduce the power, size and cost of AI data centers, and by extension their components. But the growth in AI model size and processing requirements seem to be driving the industry in just the opposite direction. Effecting a positive outcome requires architectural change.

[3] Created by SOCIONEXT based on:
IEA "Electricity 2024 Analysis and Forecast to 2026"

Solutions

These challenges are most efficiently addressed by optimizing the deployment of AI models through enhancements to the AI accelerators themselves, along with tighter integration in the compute environments that feed and orchestrate them.

While the General Purpose GPU (GP GPU) has been the AI acceleration engine of choice to this point, the diversity of system requirements and the maturation of AI models is pushing the industry towards optimized architectures. For example, the AI models used in ranking and recommendation systems employed by social media companies and content providers are very sensitive to model size and network bandwidth during training and memory capacity in inference. In contrast, the large language systems models employed in generative AI are very sensitive to model size and scale during training and compute, memory bandwidth and latency during inference[4].

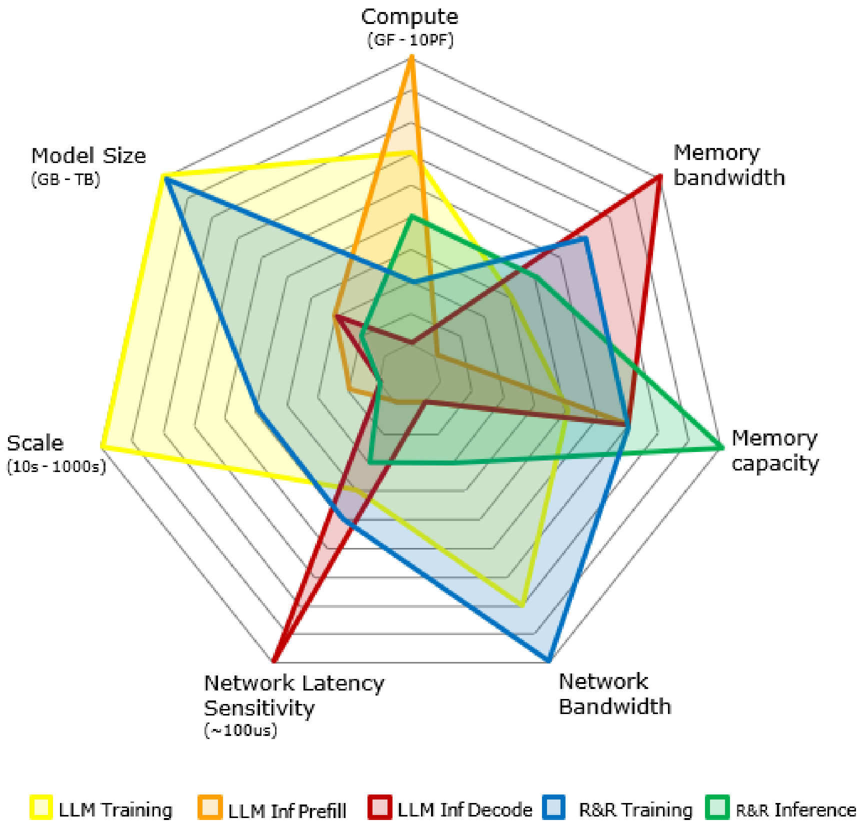


Figure 5: Diversity of AI [4]

As the underlying algorithms mature, there is an opportunity to optimize the architecture against the key metrics of performance, power, and area, according to the application. Examples of architectural optimizations include graph / dataflow processing, packet-based processing, near- or in-memory processing, and programmable logic. These architectural enhancements can be combined with application–specific optimizations such as quantization, compression and pruning.

[4] Meta, AI Hardware and Edge AI Summit, October 2023

SoC Integration – the “Solution SoC” Approach

AI accelerators are typically paired with general purpose CPUs that orchestrate data flow and operations across the AI fabric. A ‘complete’ AI processor might include a CPU complex, an AI complex, and high-bandwidth buses. Off-device, high-speed interfaces connect a tiered sub-system of HBM and DDR DRAM and other networked devices via PCIe and Ethernet. These challenges are substantial but necessary to develop a complete AI processor that delivers the advances in AI architecture to market.

Today, a leading-edge SoC might take hundreds of engineers to develop from scratch. In most cases, it is neither feasible nor desirable to take this approach. To realize their projects within an acceptable timeline and cost, system architects should specify their software and hardware requirements at a high level and focus their development efforts on the most differentiating aspects of their product. The other aspects of the project can then be delivered by an SoC partner such as Socionext. This is the “Solution SoC” approach.

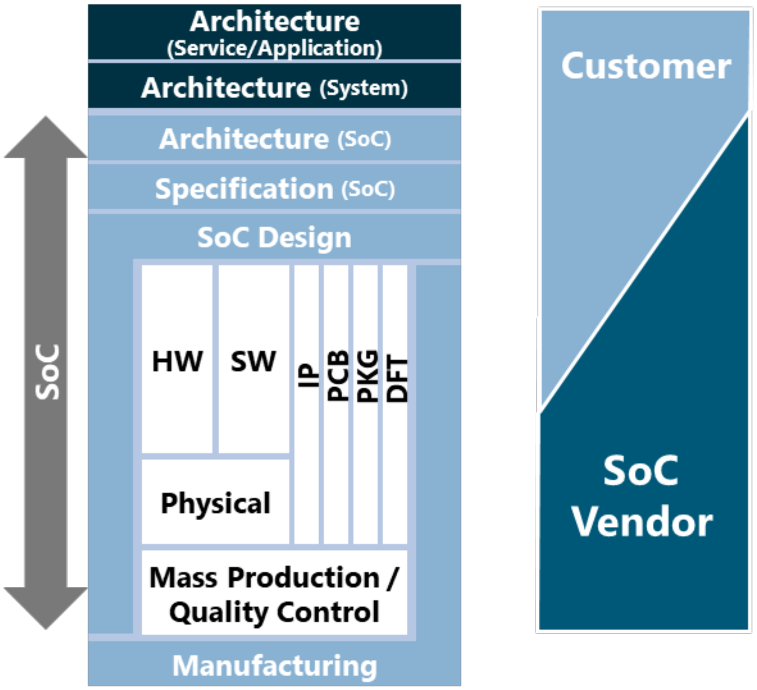


Figure 6: The Solution SoC approach

In the Solution SoC development model, the system architect interprets the requirements defined at the service or application level and translates them into system-level requirements. The system architect and the SoC vendor then collaborate to define the SoC requirements and partition the implementation work between them. The extent of the collaboration can range widely. In the turnkey case, the implementation work is handled entirely by the SoC vendor, ending with the completed hardware and software delivered to the customer for integration into their product. At the other extreme, the customer may develop large functional blocks and work closely with the SoC vendor for the logical and physical integration of the functional blocks into the full design. The customer may also be responsible for a large part of the software development, and a suitable development environment must be established as part of the project.

By working with an experienced SoC vendor, the customer can take advantage of the vendor's deep experience to reduce risks to the project (including the need for re-work) through correct specification of the device, timely staffing of a design team, selection of proven, well-qualified partners, and efficient development. The SoC vendor has access to a wide range of ecosystem partners such as design services and intellectual property providers, often on preferential terms, and can recommend the best implementation for the project. The SoC vendor also reduces operations and supply chain risk through long-term partnerships with semiconductor foundries, packaging manufacturers, and assembly and test houses.

Example Solution SoC Integration Challenges

Socionext has over 40 years of experience in the development and mass production of SoCs, with a strong focus on multi-core compute devices for hyperscale data center applications

The example below, a device now in mass production, illustrates the challenges of advanced SoC integration. The target application required a high core count and very high core and bus operating frequencies. The device contains 40 Arm CPU cores running at 3 GHz integrated into a 2 GHz Bus coherent mesh backplane. To achieve these high operating speeds across so many cores, the design uses a specialized place and route flow, including a structured clock tree implementation, bus wiring, and IP that is customized for better quality of routing. Creating the device required collaboration across many functions: front end, back end, design for test, and software.

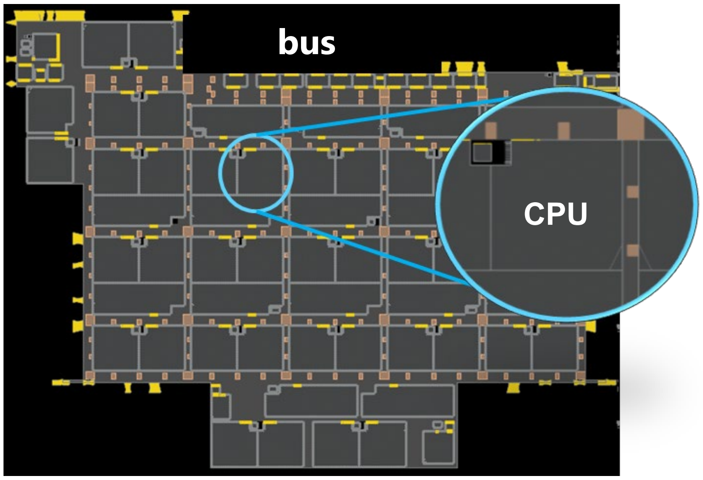


Figure 7: Multicore CPU

Clock Tree Management

Among the many challenges in SoC integration, clock latency management is critical to achieving the required performance. If the clock latency is too high, the clock skew across the chip lowers the maximum operating frequency. Clock latency is particularly challenging to manage in emerging heterogeneous compute architectures, where CPU and AI processor cores of different sizes reside on the same device. One effective solution is to use different clock structures, as illustrated in the diagram below. For a small sized block, the design uses a traditional structure generated by clock-tree synthesis. For medium size blocks, such as those found in a CPU or GPU, the design uses an H tree. For large blocks with regular structures, a fishbone tree is used. A hybrid clock architecture can reduce clock latencies and achieve the same clock latency across all blocks while minimizing the clock tree overhead.

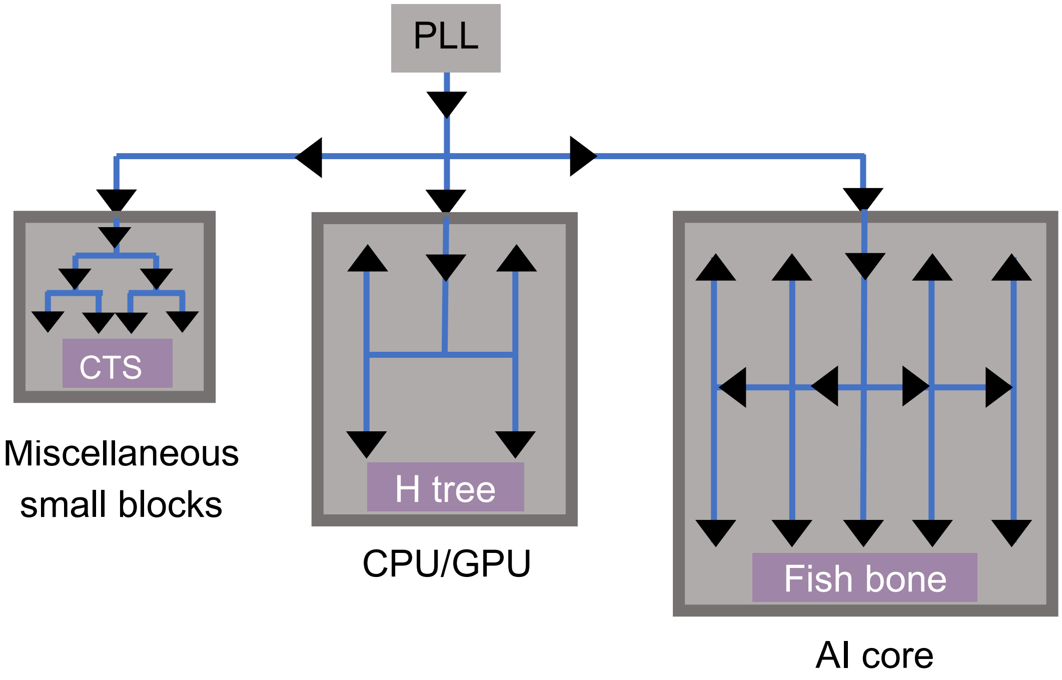


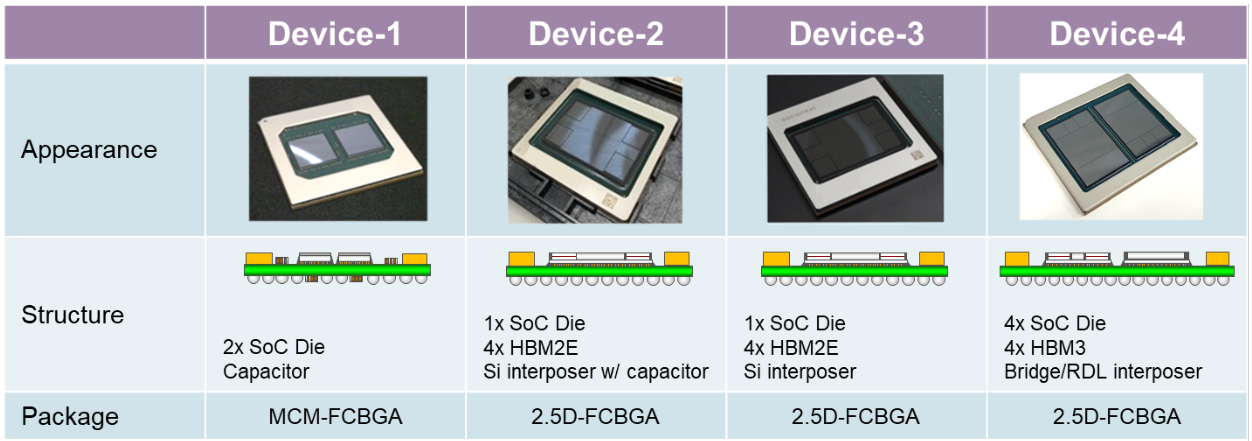
Figure 8: Hybrid clock architecture

High Performance Packaging

The compute density and high memory bandwidth requirements of AI systems place great demands on the package. It must implement:

- A heterogeneous system of CPUs, AI accelerators and processor and memory die
- Wide high-speed interconnect both between die and off-package.
- Management of high thermal power dissipation
- Mechanical robustness and reliability
- all while minimizing package area to reduce system size

Novel package architectures with silicon interposers or redistribution layers require careful co-design and simulation that includes the die, packaging, and system level to achieve the desired performance. This entails close collaboration between the customer and the SoC vendor. The customer benefits from Socionext's substantial experience in the packaging of high-performance compute devices.



	Device-1	Device-2	Device-3	Device-4
Appearance				
Structure	2x SoC Die Capacitor	1x SoC Die 4x HBM2E Si interposer w/ capacitor	1x SoC Die 4x HBM2E Si interposer	4x SoC Die 4x HBM3 Bridge/RDL interposer
Package	MCM-FCBGA	2.5D-FCBGA	2.5D-FCBGA	2.5D-FCBGA

Figure 9: Example packaging

Evolution to Chiplets

In principle, CPU cores can be added to a design up to the reticle size. In practice, such designs become burdened by several factors including the complexity of the CPU interconnect fabric, the need to dissipate heat effectively and the high die cost due to reduced yields at large die sizes. The solution is to disaggregate the die into chiplets, each implementing distinct functions of the original, monolithic device.

Chiplets allow the integration, at the package level, of compute sub-systems in the most advanced process nodes for optimal performance, along with application-specific accelerators for AI, 5G/6G communications, and other applications through implementation in custom logic or FPGA.

A chiplet approach to System-on-Chip integration conveys many benefits compared to traditional monolithic integration:

- Reduced chip cost
- Better heat dissipation through distributed thermal management across chiplets
- Application specific chiplets – for example, GPU, AI acceleration, interface – which can be updated asynchronously on their individual technology lifecycle
- Optimized process technology considering maturity, performance, and power for each chiplet

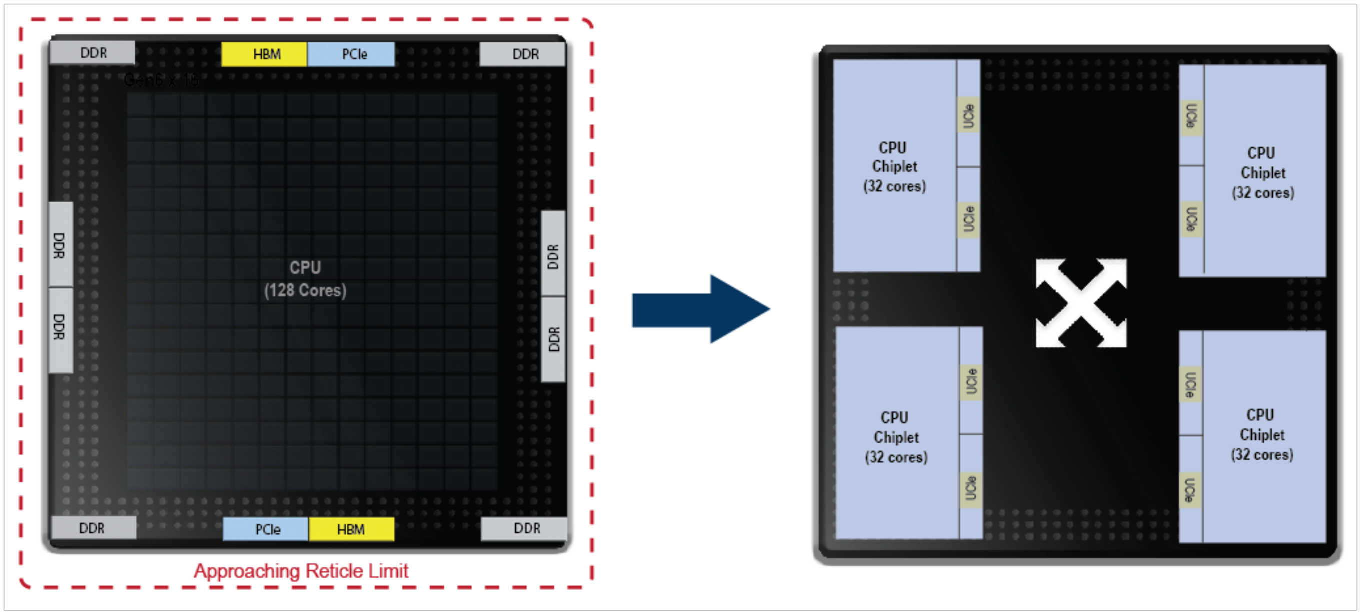


Figure 10: Disaggregation of monolithic processors into chiplets

The multi-core CPU chiplet, combined with the I/O hub chiplet, enables systems architects to create workload-optimized CPU/memory/IO subsystem solutions.

Below is an elaboration of a Hyperscale Compute Processing solution showing 4 CPU chiplets and an IO chiplet implementing HBM and DDR memory interface, all interconnected using UCIe-based low latency coherent mesh network for clustering.

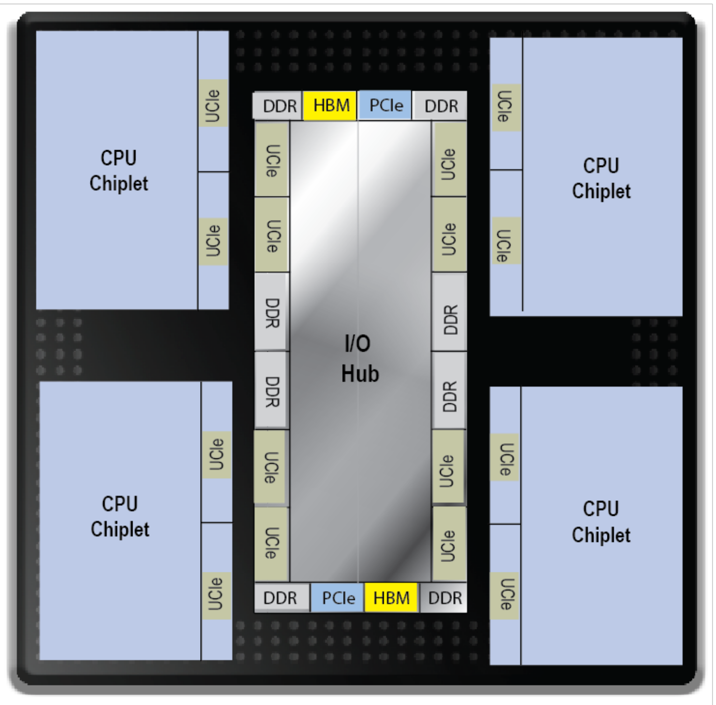


Figure 11: Example Hyperscale Compute Processing solution

Custom and configurable hybrid compute/AI accelerators can be created by leveraging the CPU and the I/O chiplets along with custom AI acceleration chiplet to create AI workload-specific systems delivering power-optimized solutions.

An example of a customized edge AI application includes 2 CPU chiplets connected through the IO chiplet to custom AI accelerator chiplets. These are packaged along with the HBM in a single FCBGA package.

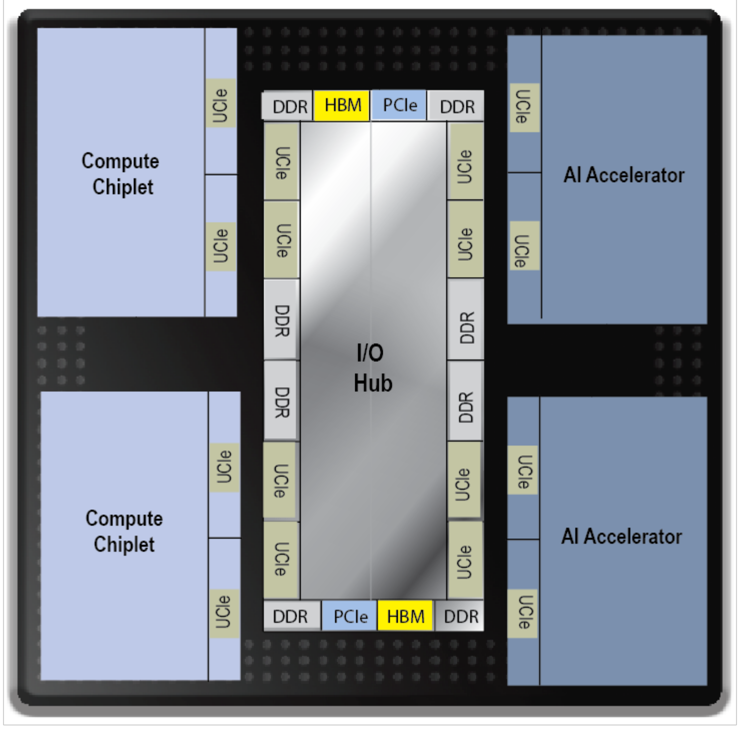


Figure 12: Example Hybrid Compute Processing / AI Acceleration solution

Socionext is developing chiplet-based SoCs to address the most demanding applications in the Automotive, Data Center and Industrial segments, and recently announced a collaboration with Arm and TSMC to develop 2nm chiplet SoCs aimed at hyperscale data center server, 5/6G infrastructure, DPU and edge-of-network markets.

About Socionext

Socionext was formed in 2015 from the fusion of Fujitsu LSI Division and Panasonic Semiconductor and has over 40 years of experience as a technology leader in the delivery of SoCs for high performance compute, networking and consumer applications. The challenge of developing and bringing to market novel AI architectures is substantial, but it can be managed by partnering with Socionext.

socionext™