Heterogeneous Integrated Computing: An Innovative Revolution Driven by Multi-Core Collaboration

Shouhe Zhu, Xiyan Zhu, Hao Li

June 20, 2025

Contents

1 Research Background

In recent years, applications such as autonomous driving, the metaverse, and artificial intelligence have continuously innovated and developed, leading to explosive growth in data scale, algorithm complexity, and computing power demand. However, the end of **Dennard Scaling** has caused single-core frequencies to stagnate at 3-5GHz; **Moore's Law**'s transistor density doubling cycle has extended to 3-5 years, reducing CPU performance annual growth to 3%; **Amdahl's Law** reveals that parallel acceleration is limited by the serial portion of tasks, leading to diminishing returns for kilocore-level CPU scaling. To enhance performance, there is an urgent need to explore heterogeneous integration technologies. Simultaneously, various accelerator processors have become crucial components of computing infrastructure, and CPU+xPU-based heterogeneous computing systems are increasingly becoming the mainstream architecture across various computing scenarios. Nevertheless, as the variety and quantity of heterogeneous computing systems grow, issues such as balancing xPU performance and flexibility, overcoming computation island problems between different xPUs, and increasing debugging and maintenance costs are becoming more prominent. Therefore, there is an urgent need to strengthen theoretical research and practical exploration in the direction of heterogeneous integrated computing. The evolution from CPU multi-core parallelism (e.g., AMD 96-core EPYC) to "CPU+XPU" heterogeneous architectures, and further to hyper-heterogeneous integration, is an inevitable technological progression aimed at breaking through the performance-flexibility trade-off limitations of single processors.

Taking the development of artificial intelligence as an example, an article in *Nature Electronics* in April 2022 showed that since 2018, with the emergence of large AI models, the demand for computing power has, on average, doubled every two months. Morgan Stanley estimated that "Google's 3.3 trillion searches in 2022 cost approximately 0.2 cents each," and John Hennessy stated that "large model-based search costs 10 times more than standard keyword search." Changes in demand and cost constraints, coupled with the enablement of new chip technologies such as NoC (Network-on-Chip) and SiP (System in Package), will inevitably drive a transformation in computing infrastructure. Computing architectures are gradually transitioning from the current fragmented, isolated heterogeneous computing towards heterogeneous integrated computing. Concurrently, a system-design-centric approach, where computing architectures are designed, defined, and planned according to application requirements, has become the optimal feasible solution for promoting multi-level technological integration.

2 Definition

Heterogeneous Computing refers to a system computation method composed of processors with different instruction sets and architectures. The main difference between a CPU and other processing engines is that a CPU is Self-Control (Turing-complete) and can operate independently, while other accelerator processors require the assistance of a CPU to run. Therefore, heterogeneous computing typically refers to the CPU+xPU heterogeneous computing architecture (xPU generally refers to various non-CPU accelerator processors).

Intel introduced the concept of "**Hyper-Heterogeneous Computing**" in 2019, emphasizing three aspects: system architecture, process and packaging, and unified heterogeneous computing software. However, at the most critical system architecture level, Intel only emphasized "multi-" without further elaborating on hyper-heterogeneous computing or providing detailed explanations for its design and implementation.

"**Heterogeneous Integrated Computing**" is a novel concept for which a unified industry definition has not yet been established. Conceptually, "heterogeneous integrated computing" falls within the scope of heterogeneous computing and can be defined as a higher-order form of heterogeneous computing.

Narrowly defined "heterogeneous integrated computing" is a new computing architecture and method achieved through fusion computing. In a **broad sense, "heterogeneous integrated computing"** involves the integration of different levels and types of technologies to efficiently utilize heterogeneous CPUs and various types and architectures of accelerator processors, thereby achieving larger scale, higher performance, and more efficient fusion of computing resources.

Broadly defined heterogeneous integrated computing primarily includes the following aspects:

1. **Hyper-Heterogeneity**: The number of heterogeneous processors in the system is three or more. "One

is called homogeneous, two are called heterogeneous, and three or more are called hyper-heterogeneous." Hyper-heterogeneity is a prerequisite for heterogeneous integrated computing.

2. **Hardware Integration**: Emphasizes deep collaboration (where a single task is processed by two or more processors) and deep fusion (where a specific task can run across different types of processors such as CPU, GPU, and DSA, or across different architectures within the same type of processor). Processors can communicate and transfer data via high-speed buses or high-performance networks, enabling collaborative computing through higher-level system partitioning and task scheduling.

3. **Software Integration**: Aims to reduce the complexity of heterogeneous integrated computing systems and enable cross-platform execution of computing tasks by integrating and optimizing technical resources such as operating systems, application software, programming models, programming languages, communication protocols, and data, thereby providing a unified software execution environment and compilation and development tools for heterogeneous (hardware) computing environments.

4. **System Integration**: Through reasonable task allocation and resource scheduling, heterogeneous integrated computing systems can achieve higher computing performance and better computing efficiency.

Traditional heterogeneous computing specifically refers to CPU+xPU computing architectures. The key difference between heterogeneous integrated computing and traditional heterogeneous computing is that traditional heterogeneous computing involves only one type of accelerator processor and focuses solely on the collaboration between the CPU and the accelerator processor. In contrast, heterogeneous integrated computing involves two or more types of accelerator processors and requires a strong focus on the collaboration and integration among all processors, as well as the integration between hardware and software, and within and between systems.

3 Principles of Heterogeneous Integrated Computing

The core principle of heterogeneous integrated computing lies in breaking the boundaries of traditional homogeneous computing or loosely coupled heterogeneous systems. Through efficient management and scheduling mechanisms, it tightly integrates and effectively coordinates different types, architectures, and advantageous computing units to form a single, powerful computing system, thereby achieving optimal utilization of computing resources. The more specific working principles of this technology can be summarized into six layers: **heterogeneous processor collaboration, task scheduling and partitioning, unified memory architecture, efficient interconnection communication, multi-architecture programming models, and software-hardware co-optimization**. The following sections will elaborate on these six core layers.

Figure 1: Advantages of Different Processors

Heterogeneous processor collaboration is the cornerstone of heterogeneous integrated computing, achieving overall performance improvement by leveraging the unique strengths of different processors. Table 1 presents a comparison of different processor characteristics. Taking a typical CPU+GPU+FPGA heterogeneous integrated computing system as an example, its collaborative mechanism is roughly as follows: the CPU preprocesses tasks and allocates them to the GPU and FPGA; each unit performs computations via OpenCL programming, and the results are transferred back to the CPU via PCIe for integration. Concurrently, the **Heterogeneous System Architecture (HSA)** further reduces communication latency by providing a unified virtual address space, enabling the CPU and GPU to share memory and tasks. This mechanism fully leverages the CPU's management advantages, the GPU's parallel processing capabilities, and the FPGA's energy efficiency and flexibility.

Table 1: Comparison of Heterogeneous Processor Characteristics

Processor Type	**Primary Function**	**Core Advantages**	**Typical Workloads/Application Scenarios**	**Main Limitations**
CPU	General-purpose computing, logical control, resource management	Flexible, adept at serial processing, branch logic	Operating systems, general applications, complex control	Limited parallel computing capability, relatively low energy efficiency
GPU	Large-scale parallel computing, graphics rendering	High throughput, SIMD/SIMT	Gaming, AI training/inference, scientific simulations, 3D modeling	Serial processing capability limited, not as versatile as CPU
FPGA	Reconfigurable hardware acceleration, custom logic	Extremely high flexibility, high energy efficiency, field reconfigurable	Real-time signal processing, hardware acceleration, network processing, prototyping	Complex design, long development cycle, relatively low frequency
ASIC	Specific function hardware acceleration	Ultimate performance, extremely low power consumption, small size	Cryptocurrency mining, dedicated AI inference chips, data center specific acceleration	Lack of flexibility, non-reconfigurable, high development cost, long cycle

Efficient task scheduling and partitioning are crucial in heterogeneous integrated computing, aiming to minimize execution time, balance load, and reduce communication overhead. Tasks are typically modeled as directed acyclic graphs, and their scheduling problems have been discussed in data structure courses, so they will not be elaborated here.

Unified Memory Architecture addresses data transfer bottlenecks and programming complexity in heterogeneous integrated computing. Traditional CPU and GPU memories are independent, requiring explicit copying, which often introduces latency and bandwidth bottlenecks. In contrast, **Unified Memory (UM)** provides a single, coherent memory address space accessible by any CPU or GPU. It eliminates the need for programmers to manually manage data transfers, significantly simplifying the programming model and improving memory usage efficiency. Taking **On-demand Paging Migration** as an example: when a processor (e.g., a GPU) attempts to access data that is not in its local memory but resides within the unified memory space, the system automatically triggers a page fault and migrates the required data page from its current location (e.g., CPU memory) to the processor accessing it. This effectively enables data exchange through shared pointers rather than data copying.

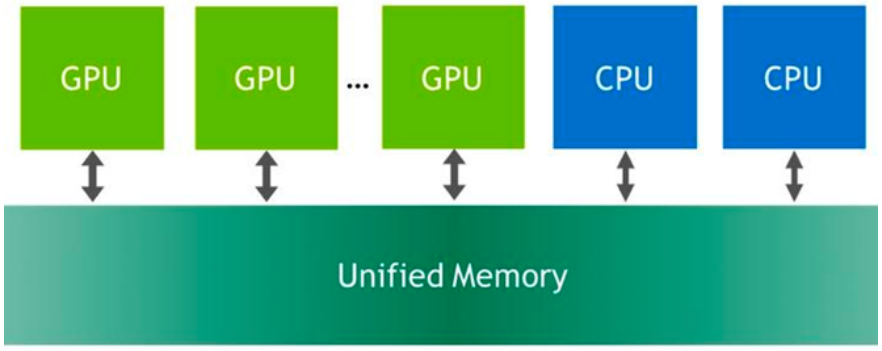

Figure 2: CUDA Unified Memory Architecture

Efficient interconnection communication is vital for heterogeneous systems to unleash their potential, ensuring rapid data exchange between processors. Existing technologies include: **PCIe (Peripheral Component Interconnect Express)**, a general-purpose high-speed bus used to connect high-bandwidth components such as CPUs, GPUs, and SSDs; **CXL (Compute Express Link)**, a high-speed serial protocol based on the PCIe physical layer that allows fast and reliable data transfer between different components within a computer system; **NVLink**, which provides high-bandwidth, low-latency communication between GPUs; and **InfiniBand** (InfiniBand Technology), which offers low-latency and high-throughput communication between servers.

Heterogeneous integrated computing products often involve different architectures, instruction sets, and programming models, posing significant challenges for software developers. To lower the development barrier, a unified programming abstraction is urgently needed. **Multi-architecture programming models** serve as a bridge connecting heterogeneous hardware with software applications. Their existence can significantly reduce software development complexity and enhance programmer productivity. For instance, **CUDA**, a well-known example, extends popular programming languages, enabling developers to harness the powerful parallel processing capabilities of GPUs. It also provides a robust ecosystem including GPU-accelerated libraries, compilers, development tools, and runtime libraries.

Software-hardware co-optimization is crucial for achieving extreme performance and energy efficiency in heterogeneous integrated computing. It demands tight integration between hardware design and software requirements, leveraging both ”soft” and ”hard” aspects to fully unleash computing potential. To realize greater benefits, both hardware architectures designed for heterogeneous integrated computing from the ground up and software stacks that facilitate heterogeneous integrated computing are necessary. It is precisely the combination of purpose-built hardware and a software stack that provides fine-grained control within a larger system abstraction framework that can fully achieve the deep optimizations offered by heterogeneous integrated computing.

4 Development and Applications of Heterogeneous Integrated Computing

4.1 Applications in Artificial Intelligence and Machine Learning (AI/ML)

With the rapid expansion of deep learning model scales, the demand for computing power in AI has surged, making heterogeneous integrated computing a core solution for accelerating training and inference. Its core idea is similarly to perform reasonable task allocation and collaborative processing based on the characteristics of different computing units and the computing requirements of AI/ML tasks.

NVIDIA’s core strategy in the AI/ML domain is to build a heterogeneous computing platform centered

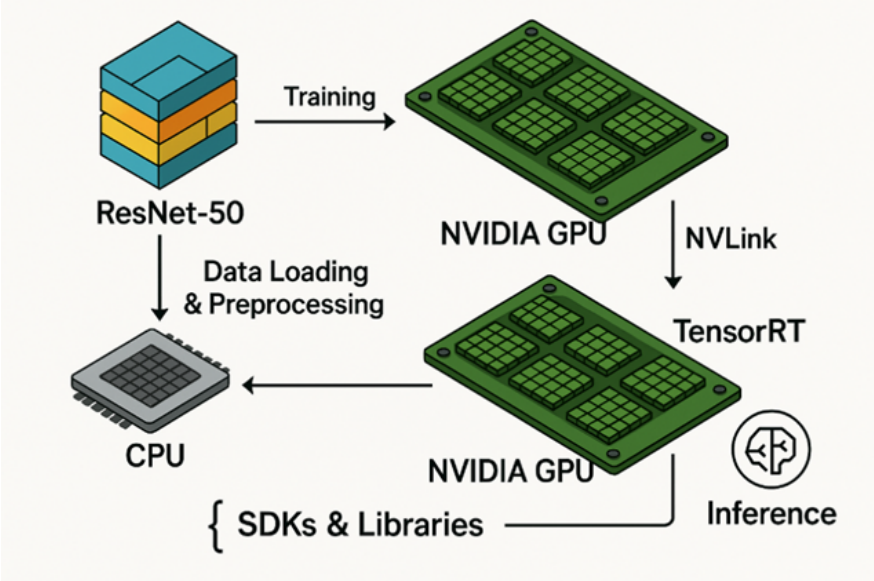

Figure 3: ResNet-50 Training Process based on NVIDIA Heterogeneous Integration Solution

around its powerful GPUs. Taking the training of large-scale image recognition models (such as ResNet-50, Figure 3) as an example, NVIDIA's heterogeneous computing integration technology plays a critical role in deep learning model training. **NVIDIA GPUs**, with their thousands of CUDA cores, parallelize and accelerate core matrix multiplication and convolution operations. Coupled with libraries like **cuDNN**, this reduces training time from weeks to hours. The CPU, meanwhile, is responsible for data loading, preprocessing, and task scheduling, effectively coordinating the entire training process. For multi-GPU systems, **NVLink** high-speed interconnect technology improves data exchange efficiency, further accelerating model parallel training. After training, models are optimized and quantized using **TensorRT** (a high-performance deep learning inference optimizer and runtime) to achieve high-speed, low-latency inference on NVIDIA GPUs, meeting real-time application requirements. Furthermore, NVIDIA provides SDKs and libraries specifically for AI/ML tasks, simplifying development and deployment. Together, these form an efficient GPU-centric heterogeneous computing platform. This software-hardware co-design strategy allows NVIDIA's heterogeneous integration solutions to demonstrate strong competitiveness in the AI/ML domain.

4.2 Applications in Image Processing and Computer Vision

Image processing tasks typically involve large amounts of pixel data, and the parallel computing capabilities of GPUs enable them to efficiently perform tasks such as denoising, object recognition, and edge detection. Through heterogeneous computing, the close collaboration between GPUs and CPUs improves the efficiency, real-time performance, and accuracy of image processing.

In medical image analysis, the **collaborative architecture design of quantum algorithms and edge computing** offers a new approach to addressing traditional computational bottlenecks and real-time demands. By embedding the efficient parallel processing capabilities of quantum computing into edge nodes, rapid feature extraction and pattern recognition can be performed on massive medical image data at the local device level, while also reducing latency risks associated with cloud transmission. For instance, in medical scenarios, quantum annealing algorithms can accelerate the localization of abnormal regions in high-resolution CT images, while edge computing nodes utilize distributed caching mechanisms to achieve real-time preprocessing of image data, significantly shortening diagnostic response times.

To ensure security for cross-domain applications, the architectural design integrates lightweight encryption protocols and quantum key distribution technologies, guaranteeing the privacy of medical data. At the computational flow level, **quantum variational algorithms** are employed to optimize the feature selection process at edge nodes. By constructing quantum-state-encoded weight matrices, they effectively enhance

the robustness against noise interference in image classification tasks. Furthermore, tailored to the heterogeneous characteristics of edge devices, the architecture adopts a **hybrid quantum-classical computing mode**, where parameter optimization tasks within convolutional neural networks are offloaded to quantum processors, while classical computing units are responsible for gradient updates and model fine-tuning, thereby achieving exponential increases in computational efficiency under limited hardware resources. This integrated architecture not only overcomes the performance bottlenecks of traditional algorithms in complex scenarios but also lays a scalable technical foundation for subsequent integration.

4.3 Applications in Communication

With the rapid advancement of communication technologies, especially the deployment of 5G and future 6G networks, communication systems face immense data processing and transmission pressures. The application of heterogeneous integrated computing in the communication domain can effectively enhance the performance and efficiency of communication systems, meeting the growing communication demands.

Efficient Signal Processing: In 5G/6G base stations, a large volume of wireless signals needs to be processed, including signal modulation/demodulation, encoding/decoding, and beamforming. These tasks demand extremely high computational power, particularly with the application of **Massive MIMO (Multiple-Input Multiple-Output)** technology, which requires real-time processing of massive amounts of signal data. Heterogeneous integrated computing, by combining CPUs, GPUs, and FPGAs, can significantly improve signal processing efficiency. For example, the CPU is responsible for control and management tasks, the GPU is used for parallel processing of large-scale matrix operations, and the FPGA is utilized for implementing specific signal processing algorithms, thus achieving efficient and low-latency signal processing.

Resource Management and Scheduling: Base stations need to dynamically allocate computing resources based on different user demands and network states. Heterogeneous integrated computing systems can dynamically adjust the load on CPUs, GPUs, and FPGAs according to real-time traffic and signal quality, ensuring efficient resource utilization. For instance, when a base station detects a sudden increase in user traffic in a certain area, it can allocate more GPU cores to the signal processing tasks for that area, while leveraging the flexibility of FPGAs to rapidly adjust signal processing algorithms.

5 Technical Challenges and Future Trends of Heterogeneous Integrated Computing

5.1 Technical Challenges

While heterogeneous integrated computing offers immense performance potential, its practical deployment and widespread application still face numerous technical challenges. These challenges are distinct from the intra-system collaboration issues discussed earlier, focusing more on the complexity of **cross-ecosystem, cross-platform, and the entire design-deployment lifecycle**:

- **Fragmented Heterogeneous Software and Hardware Ecosystems and Interoperability Challenges**: Currently, the heterogeneous computing landscape is highly diverse, with various processors (CPUs, GPUs, FPGAs, ASICs, etc.) possessing their own distinct instruction sets, memory models, and programming paradigms. This leads to a proliferation of proprietary programming languages (e.g., CUDA for NVIDIA GPUs), libraries (e.g., cuDNN), and development tools. This highly fragmented ecosystem necessitates significant developer effort to learn and adapt to different programming models, making it difficult for applications to achieve seamless migration and efficient execution across different heterogeneous platforms, thereby greatly increasing development, debugging, and maintenance complexity. The primary challenge for heterogeneous integrated computing is how to build a unified, open, and highly compatible software stack and toolchain to achieve true **cross-platform interoperability**.
- **Complexity of Full-Stack Optimization and Debugging**: A heterogeneous integrated computing system is a complex collaborative entity. Performance optimization is no longer limited to the single-processor level but requires **full-stack collaborative optimization** from the application layer, programming model, runtime, operating system, interconnection bus, down to the underlying hardware

architecture. For example, bottlenecks in memory access patterns, data transfer latency, task granularity partitioning, or processor load balancing can severely impact overall performance. Simultaneously, debugging across multiple processor types and abstraction layers is exceedingly complex, making it difficult to effectively trace and pinpoint performance bottlenecks or errors. Traditional performance analysis and debugging tools for single-processor environments are often inadequate for heterogeneous integration scenarios, necessitating the development of new diagnostic tools that can provide a **global view and fine-grained control**.

- **Severe Challenges of Power Consumption and Heat Dissipation**: As the number of heterogeneous processors increases and integration density rises, the overall power consumption of systems escalates sharply. This is particularly prominent in data centers and high-performance computing scenarios, where heat dissipation becomes a critical issue. High power consumption not only increases operating costs but also imposes higher demands on hardware design and packaging technologies. **Effectively controlling system power consumption and addressing heat dissipation challenges while pursuing ultimate performance** is a key constraining factor for the large-scale deployment of heterogeneous integrated computing. This requires innovation across multiple dimensions, including chip architecture, packaging technology, cooling solutions, and intelligent power management.

- **Ensuring Security and Reliability**: Due to its complexity and multi-component nature, heterogeneous integrated computing systems introduce new security vulnerabilities and reliability risks. For example, how are trust boundaries defined between different processors? How are data integrity and confidentiality ensured in shared memory regions? How does the failure of a single processor affect the stability of the entire system? In critical mission scenarios (e.g., autonomous driving), any computational error or system failure could lead to severe consequences. Therefore, **building end-to-end security protection mechanisms within heterogeneous integrated architectures and ensuring high system reliability under various operating conditions** are challenges that urgently need to be addressed.

5.2 Development Trends

Heterogeneous integrated computing is evolving along several directions to overcome existing challenges and fully unlock its potential:

- **Hardware Level: Moving Towards Higher Integration with "Chiplets" and "SoCs"**: Future heterogeneous integration will accelerate towards tighter hardware integration. **Chiplet technology** integrates computing units of different functions and processes (e.g., CPU cores, AI accelerators, memory controllers, I/O interfaces) as small chips within a single package, achieving on-chip level collaboration through high-speed interconnects (e.g., **UCIe**). This not only breaks through the area limitations and yield issues of traditional monolithic chips but also allows for flexible customization and combination according to specific application needs, optimizing performance, power consumption, and cost. Simultaneously, **System-on-Chip (SoC)** will further deepen integration, combining more heterogeneous processing units, storage, and communication modules onto a single chip, providing a more efficient physical foundation for heterogeneous integration.

- **Software Level: Building Unified, Intelligent Runtime and Programming Models**: To address ecosystem fragmentation, the future will see more **unified and intelligent runtime systems**. These systems will be able to automatically perceive underlying heterogeneous hardware resources and intelligently schedule and optimize tasks based on their characteristics, enabling automatic code parallelization, dynamic task migration, and elastic resource allocation. **Declarative programming models** (e.g., Domain-Specific Languages, DSLs) will become more prevalent, allowing developers to focus on describing computational logic rather than underlying hardware details, with compilers and runtimes responsible for efficient mapping to heterogeneous hardware. Furthermore, **AI-assisted compilation optimization and performance tuning tools** will become standard, utilizing machine learning techniques to predict optimal task allocation and resource configuration schemes.

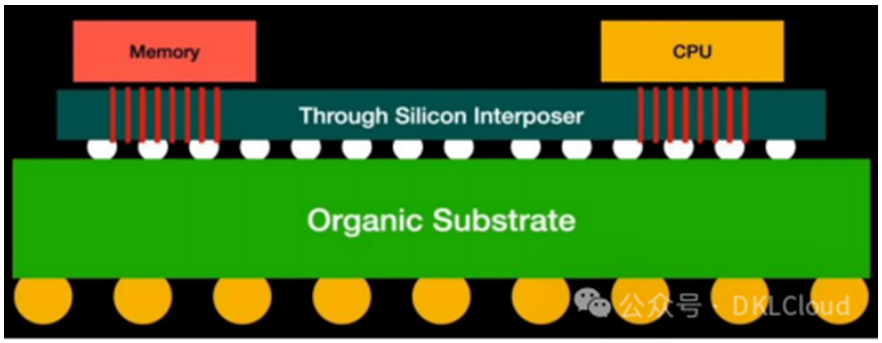

Figure 4: Chiplet Packaging Schematic

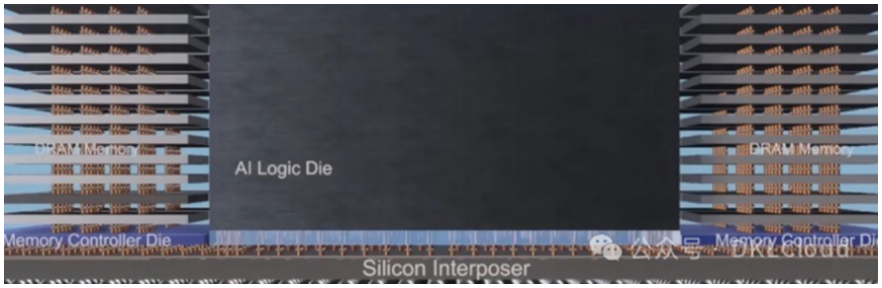

Figure 5: Actual Chiplet Package

- **Architecture Level: Memory-Centric and Dataflow-Driven**: The data movement bottleneck caused by the separation of computation and storage in traditional Von Neumann architectures is particularly pronounced in heterogeneous systems. Future architectures will place greater emphasis on **memory-centric design**, for example, by adopting **Processing-in-Memory (PIM)** technology, which offloads some computational logic directly into memory units, reducing the overhead of data movement between processors and memory. Concurrently, **dataflow-driven architectures** will become a trend, where the execution order of computational tasks no longer strictly depends on instruction sequences but is driven by data availability, thereby better adapting to parallel heterogeneous environments.
- **Application Level: Deep Integration and Edge-to-Cloud Collaboration**: Heterogeneous integrated computing will no longer be confined to single devices or data centers but will form a **full-link collaborative computing model from edge devices and terminals to the cloud**. For example, in autonomous driving, highly real-time perception and decision-making are performed on edge AI chips, while complex path planning and model training occur on high-performance heterogeneous clusters in the cloud, with both collaborating through efficient communication protocols. This deep integration will foster more innovative applications, such as real-time rendering and interaction in AR/VR, intelligent manufacturing and predictive maintenance in the industrial Internet of Things, and precision diagnostics in healthcare.

6 Conclusion

Heterogeneous integrated computing, through hardware architecture innovation (Chiplet/3D stacking), breakthroughs in software abstraction (unified memory/cross-platform programming), and system-level optimization (computing network/FaaS), has achieved "collaborative evolution of flexibility and efficiency." It is not only an engineering technical path to extend Moore's Law but also a fundamental infrastructure revolution that reconfigures the computing paradigm. With the advancement of standardization (UCIe/CXL) and the explosive demand from AI/scientific computing, heterogeneous integrated architectures will provide core

computing power support for scenarios such as autonomous driving, the metaverse, and quantum simulation.

While heterogeneous integrated computing holds immense promise, its development still faces a series of severe challenges, including fragmented software and hardware ecosystems, complex full-stack optimization and debugging, power consumption and heat dissipation challenges, and security and reliability assurance. Looking ahead, as trends such as chiplet technology, unified runtime systems, memory-centric architectures, and deep edge-to-cloud collaboration continue to evolve, heterogeneous integrated computing is expected to gradually overcome these difficulties and revolutionize how we tackle complex computational tasks, bringing unprecedented innovation opportunities across various industries.

7 Division of Labor

Shouhe Zhu: Contributed topic ideas; Defined team responsibilities; Coordinated and integrated teamwork; Created PPT sections 1-4; Authored report document sections 3-4; Polished, formatted, and translated the report document.

Xiyan Zhu: Contributed topic ideas; Authored report document sections 1, 4, 6; Designed PPT introduction and summary; Delivered class presentation.

Hao Li: Contributed topic ideas; Created PPT sections 4-5; Authored report document sections 4-5.

References

[1] Muthukumaran Vaithianathan. The Future of Heterogeneous Computing: Integrating CPUs, GPUs, and FPGAs for High-Performance Applications. *International Journal of Emerging Trends in Computer Science and Information Technology*, 6(1), 2025.

[2] George Kyriazis. *Heterogeneous System Architecture: A Technical Review*. AMD, 2012.

[3] Ministry of Industry and Information Technology, Fifth Electronic Research Institute. *White Paper on Unified Identification and Service of Heterogeneous Computing Power*. 2023.

[4] Jonathon Evans, Michael Andersch, Vikram Sethi, Gonzalo Brito, and Vishal Mehta. *NVIDIA Grace Hopper Superchip Architecture Whitepaper*. Nov 10, 2022.

[5] Jing Pei, Lei Deng, Sen Song. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature*.

[6] Muthukumaran Vaithianathan. The Future of Heterogeneous Computing: Integrating CPUs, GPUs, and FPGAs for High-Performance Applications. *International Journal of Emerging Trends in Computer Science and Information Technology*, 6(1), 2025.

[7] Quantum Algorithm Fusion and Edge Computing Optimization for Medical Imaging and Autonomous Driving Security Practices.