

DEPARTMENT: LEADERSHIP COMPUTING

Modular High-Performance Computing Using Chiplets

Bapi Vinnakota  and John M. Shalf , Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA

The performance growth rate of high-performance computing (HPC) systems has fallen from 1000× to just 10× every eleven years. The HPC world, like large cloud service provider data centers, has turned to heterogeneous acceleration to deliver continued performance growth through specialization. Chiplets offer a new, compelling approach to scaling performance through adding workload-specific processors and massive bandwidth to memory into computing systems. If design and manufacturing challenges are resolved, chiplets can offer a cost-effective path for combining die from multiple function-optimized process nodes, and even from multiple vendors, into a single application-specific integrated circuit (ASIC). This article explores opportunities for building and improving the performance of bespoke HPC architectures using open-modular “chiplet” building blocks. The hypothesis developed is to use chiplets to extend the functional and physical modularity of modern HPC systems to within the semiconductor package. This planning can reduce the complexity and cost of assembling chiplets into an ASIC product and make it easier to build multiple product variants.

THE PERFORMANCE GROWTH CRISIS

The information in Figure 1 shows that the performance growth of high-performance computing (HPC) systems, as measured by the LINPACK benchmark, used to be 1000× every 11 years but has recently moderated to merely 10× per 11 years. Many argue that the LINPACK benchmark is no longer relevant because of its emphasis on floating-point operations per second (flops) over memory performance, but it should be concerning that, even with a benchmark that is arguably too easy, growth rates are dropping off rapidly. The recently introduced HPC Gradients (HPCG) benchmark also shows a similar drop-off in performance improvement, and replacement rates for modern HPC systems have dropped to their lowest measured rate since the inception of the Top500 list. The bottom line is that the approach that the HPC community has depended upon to procure systems

that deliver exponential performance improvements to the scientific and engineering users over the past three decades is failing.

Heterogeneous Architectures and Challenges

General-purpose flops (or even memory operations in HPCG) cannot be the measure of success for future systems. Moore’s economic theory (usually called *Moore’s law*) is faltering. In response, HPC systems have pivoted to the use of accelerators such as FPGAs, GPUs, and others.^{1,2} We have already seen the start of this trend with the delivery of our first Exascale computing systems as heterogeneous systems of modular CPU, GPU, and accelerator racks.

Effectively exploiting accelerators in applications is a nontrivial task. The overhead of data movement and increased control-flow complexity between CPUs and accelerators has to be far less than the speedup from accelerators. Researchers have consistently demonstrated that the latency and the bandwidth of the network between the general-purpose CPUs, memory, and accelerators has a significant impact on application

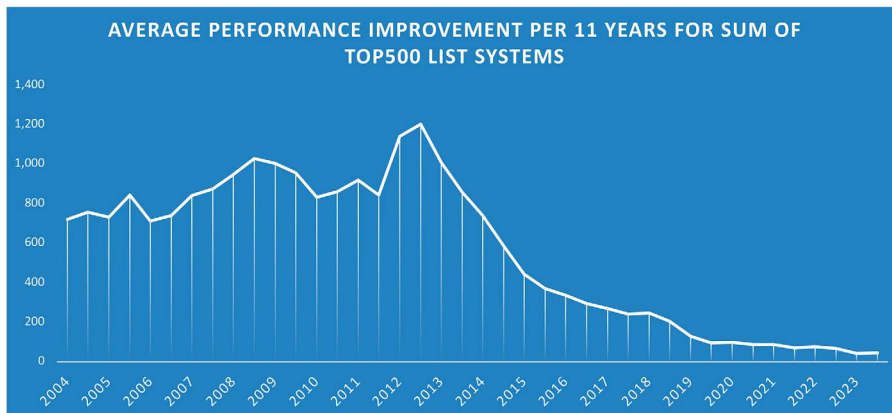


FIGURE 1. High-performance computing (HPC) performance development according to Linpack. The vertical axis is the factor of performance improvement for the sum of all systems on the Top500 list as measured by the LINPACK benchmark. Whereas performance improved by a factor of $1000 \times$ every 11 years for the first two decades of the Top500, this figure has dropped precipitously in the past decade.

performance.^{3,4} Moving data over these networks also consumes substantial energy. Today's systems aim to reduce the performance cost of data movement by overprovisioning rack-to-rack bandwidth, further increasing system cost and energy consumption. Future systems require a new approach to prevent bandwidth starvation of accelerators and other specialized hardware.

The "ideal" approach to incorporating accelerators is to design them on the same die as the general-purpose CPU. For example, even for applications less sensitive to latency such as general matrix multiplication, " ... when similar optimizations are employed in off-chip integration, on-chip integration presents up to 20% better performance, with 17% less total energy consumption."³ Modern (GPU/CPU) dies are already large, limiting the amount of additional functionality that can be added to them. Large dies increase product costs nonlinearly. Design and engineering costs grow disproportionately with die size. Large dies are also more expensive to manufacture and more likely to be defective and yield fewer good dies per wafer. To cost-effectively produce large application-specific integrated circuits (ASICs), industry has turned to approaches that realize an ASIC across multiple dies, referred to as chiplets, within one semiconductor package.⁵

Vision: Open/Multivendor HPC with Chiplets

HPC thought leaders recently suggested¹ that future HPC systems may (in a return to past practice) need to transition away from commodity processor-based systems (typically Intel or AMD CPU processors) to bespoke systems built with ASICs customized to HPC.

The Fugaku supercomputer is an example of a modern HPC system centered around a custom ASIC. As seen in the dramatic workload shift to large language models (LLMs) at large-scale cloud providers within a five-year span, it is likely that as the workloads on such a system will evolve; they will require new accelerators and, correspondingly, the custom ASICs in these systems will need to evolve. Traditional design methods are too expensive to use to support the evolution of the custom ASICs likely to be required by future bespoke HPC systems.

Chiplet technology has become more popular over the last decade as an approach to address the design and manufacturing cost challenges of complex high-performance ASICs. By reusing intellectual property as chiplets across designs, the portion of a new product that needs to be designed from scratch is much lower, reducing costs and time to market. Partitioning a large die into smaller dies also lowers manufacturing costs. For example, AMD's costs for a server die were lowered by more than $2\times$ by partitioning a 780-mm^2 die into four 200-mm^2 dies.⁵ These gains do not come for free. Partitioning an ASIC across multiple dies increases design and engineering complexity in product manufacturing. These challenges have largely limited the use of chiplets to large companies and very high-volume products. Industry is developing a set of open standards for chiplets that have the potential to create an "open-chiplet economy," serviced by both small and large companies.^a With this open ecosystem, custom ASICs can potentially be economically assembled by combining chiplets from multiple companies.

^a<https://www.opencompute.org/blog/building-an-open-chiplet-economy>.

Successful chiplet-based products in large companies (e.g., AMD CPUs and Intel's Ponte Vecchio) develop products by 1) developing a functional modular system architecture that partitions the target design into chiplets for functional components; 2) developing a physical modular architecture that provisions each chiplet with adequate die-to-die (D2D) interconnect bandwidth, power, and size budgets; and 3) choosing a cost-effective packaging technology and identifying physical locations in the package for each chiplet. This planning reduces the complexity and cost of assembling chiplets into an ASIC product and makes it easier to build multiple product variants.

The hypothesis developed in this document is to use chiplets to extend the functional and physical modularity of modern HPC systems to within the semiconductor package. We propose defining an open, baseline-configurable HPC-oriented system-on-chip (SoC) architecture composed of functional modules, that also obey physical modularity restrictions, to be realized as chiplets. We develop this reference architecture from the HPC ASIC used in the Fugaku supercomputer.⁶ We show how designers can insert new accelerators and input-output (I/O) devices, compatible with the system, by following the modularity rules of the architecture. That is, the baseline can be inexpensively extended to produce multiple custom ASICs of various types by reusing a majority of the baseline in each variant and replacing a small fraction with new logic. This approach can leverage the emerging open-chiplet economy by allowing an ASIC to be built using chiplets from multiple vendors, reducing the cost of designing and manufacturing bespoke ASICs for future HPC systems.

An Overview of Chiplets

Gordon Moore predicted a need for component-level modularity in his article, "Cramming more Components onto Integrated Circuits," which presented Moore's law, in which he stated: "It may prove to be more economical to build large systems out of smaller functions, which are separately packaged and interconnected. The availability of large functions, combined with functional design and construction, should allow the manufacturer of large systems to design and construct a considerable variety of equipment both rapidly and economically."

A chiplet is such a part, designed to implement a functional circuit block in an SoC, placed with other chiplets that implement other functions in the SoC into a single package, and to be closely logically coupled with them through a D2D interconnect optimized for short-reach connections, as depicted in Figure 2. The goal is to manufacture each chiplet economically for optimum cost and performance for its function. The functions implemented in chiplets include common functions such as CPUs, I/O [Ethernet, Peripheral Component Interconnect Express (PCIe), and memory controllers], accelerators, and so on.

What is promising with chiplets is that derivatives can be developed at a lower cost than with monolithic designs. For example, the variant of a monolithic ASIC incurs foundry-related design expenses comparable to that of producing the original ASIC as it may require new lithography masks for the whole device. With chiplets, a design can be evolved at lower cost in two ways: 1) by creating a new component chiplet and new

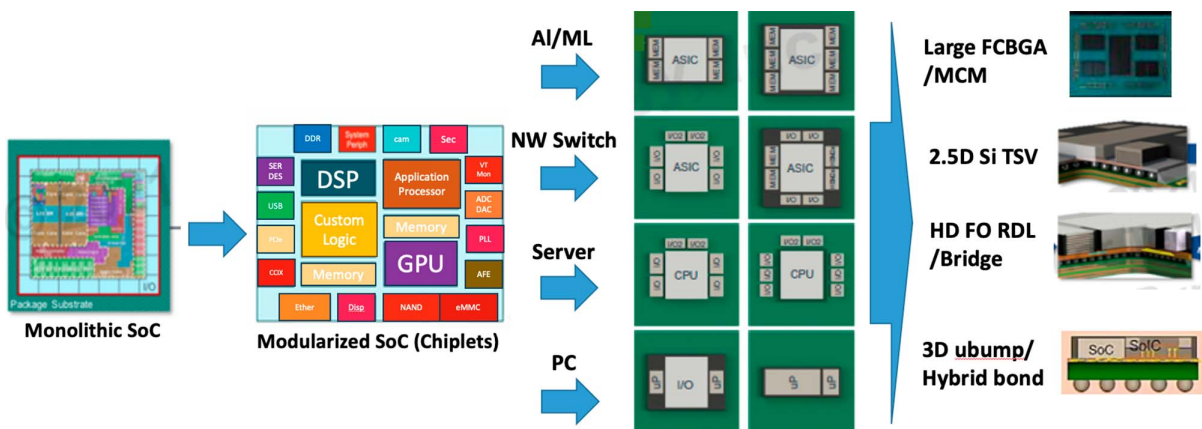


FIGURE 2. Chiplets enable modularization of monolithic SoCs into myriad specialized products using the same underlying building blocks. [Figure courtesy of Bill Chen (<https://eps.ieee.org/technology/heterogeneous-integration-roadmap.html>).] AI/ML: artificial intelligence/machine learning; MCM: multichip module; NW: network; FCBGA: flip chip ball grid array; Si TSV: silicon through-silicon via; HD FO RDL: high-density fan-out redistribution layer.

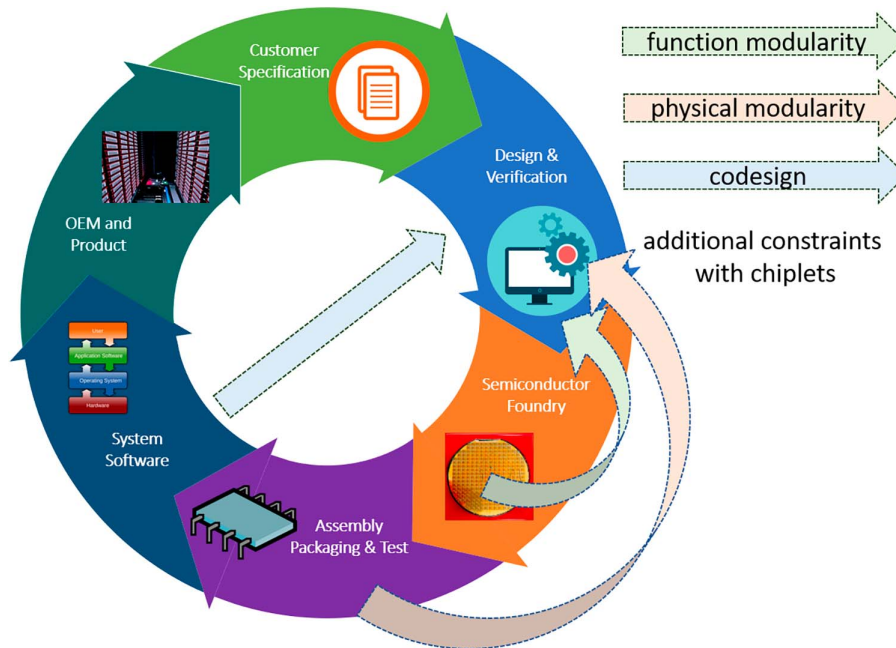


FIGURE 3. A chiplet-based design requires attributes of the design and manufacturing workflow to be brought forward. OEM: original equipment manufacturer.

package design, while reusing the rest of the chiplets as is (The foundry costs of a chiplet will be lower than for the monolithic version of the ASIC in which the chiplet is used.), and 2) by changing the relative composition of different types of chiplets. The only costs incurred here are for a new package design.

Relative to a monolithic ASIC design, a chiplet-based design incurs additional area, power, and costs. The D2D interconnect consumes power, area, and die edge space to provide the bandwidth necessary to service the logic in a chiplet. Chiplets require more complex packaging than monolithic designs. (Finished silicon dies require a protective metal, glass, or ceramic casing, referred to as a *semiconductor package*.) Multiple dies in one package make power delivery, cooling, mechanical, and signal analysis more complex. Many advanced packaging technologies have been developed for chiplet-based designs, and the specific choice is determined by application cost, performance, and power constraints.

Designing Chiplet-Based Products

To design chiplet-based products, relative to a conventional ASIC workflow, attributes usually considered to be “downstream” have to be brought forward, as shown in Figure 3:

- › *Functional modularity for manufacturing:* All the component chiplets in a product that together realize a target system function, need to be

manufactured together, perhaps across multiple semiconductor fabs. The inventory of all the chiplets needs to be managed together. Each chiplet’s function must complement that of others in the product. Every pair of physically connected chiplets have to be designed with both the same D2D interconnect protocol and provisioned with the same D2D bandwidth in that protocol.

- › *Physical modularity for package assembly and test:* All the chiplets in one product have to be designed for the same packaging technology and the package design must support the required line rate on every D2D link in it. Power, power delivery, thermal, mechanical stress, and other budgets must be allocated across all the chiplets and meet each chiplet’s needs. Chiplets need to be tested individually before insertion and then later tested as a system.
- › *Co-design:* The low-level firmware and perhaps even system and application software may need to be cognizant of the modularity. The software may impact the control and dataflow across chiplets. When new modules are inserted into the system, the software and the logic need to obey the functional and physical modularity.

To date, these challenges, typically met by developing families of chiplets simultaneously that follow internal engineering conventions, have limited the development

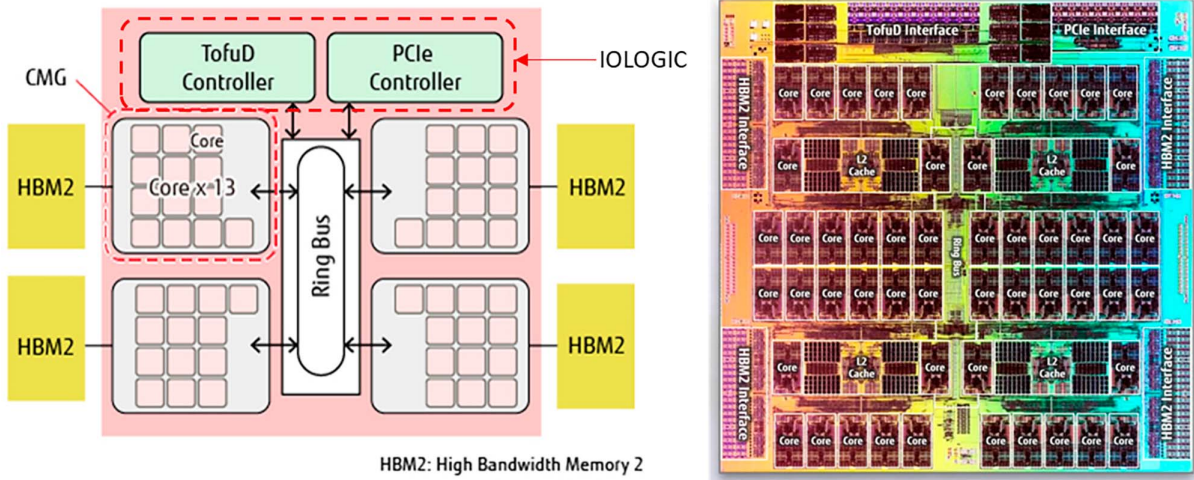


FIGURE 4. (a) Fugaku A64FX functional decomposition and (b) an ASIC. (Image courtesy of Fugaku publication.) HBM: high-bandwidth memory; CMG: core-memory group.

of chiplet-based products to large companies that develop high-volume products. The vision of the Universal Chiplet Interconnect Express consortium^b and the Open Compute Project's Open Domain Specific Architecture subproject^c is that the development of open standards for chiplets can enable a truly innovative inclusive ecosystem that enables a far larger number of vendors to produce useful chiplets.

Current open efforts focus on supporting functional modularity by developing open logical protocols for the D2D interconnect between chiplets. They largely do not constrain the physical modularity of chiplets in a system, (shown previously to be) essential for successful commercial products. For example, it could be that two chiplets that provide complementary functionality from different vendors and support the same logical D2D standard cannot be connected economically within a package because they are too hot or too large. We propose an open physical modular architecture that addresses these gaps in current open standards.

A FUGAKU THOUGHT EXPERIMENT FOR MODULAR HPC

Fugaku is a modular ARM-based supercomputer, known for its exceptional performance,⁶ that uses two custom ARM A64FX CPU ASICs per hardware blade server (the smallest modular computing element in a Fugaku system). We chose Fugaku because it is one of the few architectures explicitly co-designed for HPC

for which the power, area, interfaces, and subcomponents are publicly documented.^{6,7} As a thought experiment, we reimagine the A64FX using chiplets. We extend the thought experiment to demonstrate that chiplet partitioning opens up the SoC to myriad high-value specializations (variations on the baseline) that can be constructed at a much lower cost than building new monolithic ASICs for each variant.

Fugaku SoC Overview

As shown in Figure 4(a), the A64FX consists of three main types of logic: a compute complex, ring bus network on chip (NoC), and I/O logic. The compute complex consists of four core-memory groups (CMGs), each coupled with an off-die, in-package high-bandwidth memory (HBM). (The packaged device with HBMs is not pictured.) The A64FX is also equipped with high-speed interfaces that facilitate connectivity with other hardware blade servers in the Fugaku on a custom protocol (Tofu), PCIe, and interrupt I/O. Table 1 lists (inferred) significant physical attributes of an A64FX.

Fugaku A64FX Hypothetical Chiplet Decomposition

Consider a hypothetical functional partition of a Fugaku A64FX into a set of four compute and six I/O chiplets, as shown in Figure 5(b) (HBM modules are retained as is). The compute chiplet consists of a 13-core CMG and an NoC ring stop [shown in Figure 5(a)]. The I/O bandwidth to/from a compute chiplet needs to be equal to the bandwidth of three ring-stop I/O pairs (shown in Figure 5) plus one HBM interface (not shown

^b<https://www.uciexpress.org/>

^c<https://www.opencompute.org/wiki/Server/ODSA>

TABLE 1. Estimated specifications of Fugaku A64FX and the system.

Attribute	Detail
Die size/power/power density	20 mm × 20 mm (400 mm ²), 122 W, 0.3 W per mm ²
Network and memory I/O area power	Tofu 25 mm ² /9 W, 100 mm ² for 4 HBM + 1 PCIe
Ring bus extra area	35 mm ² , 10 mm ² /complex
Core complex	52 cores, 240 mm ² , 60 mm ² /complex
Core area/power	4.9 mm ² /2.2 W
Process node	TSMC 7-nm FinFET

TSMC: Taiwan Semiconductor Manufacturing Company; FinFET: fin field-effect transistor.

in Figure 5), a total of roughly 5 terabits per second (Tbps). Since this data transfer was previously on a single die and instead is now between die through the D2D interconnects, on a compute chiplet's four edges, we actually provision a total I/O bandwidth of 8-Tbps (2 Tbps/die edge) per compute chiplet. The D2D I/O is projected to consume approximately 6.4 W of power and offer roughly 0.6 Tbps per core in the compute chiplet. With the attribute values shown in Table 2 for a square compute chiplet, the performance and power of a package with four compute chiplets are consistent with the per-package characteristics of the A64FX CPU.

Architecture Variants

A modular chiplet-based design offers the promise of being able to economically develop many variants.

TABLE 2. Attributes of reference compute chiplet, and a map to A64FX cores.

Attribute	Value
Size	11 × 11 mm ²
D2D I/O bandwidth	8 Tbps (2 Tbps/side × 4)
Power density/power per die	0.3 W/mm ² /36.3 W
D2D power	6.4 W
Number of compute cores	13
Bandwidth per core	0.615 Tbps

Even replacing just a single chiplet in the baseline design with new functionality can change the system significantly. Examples of possible new compute chiplets include the following:

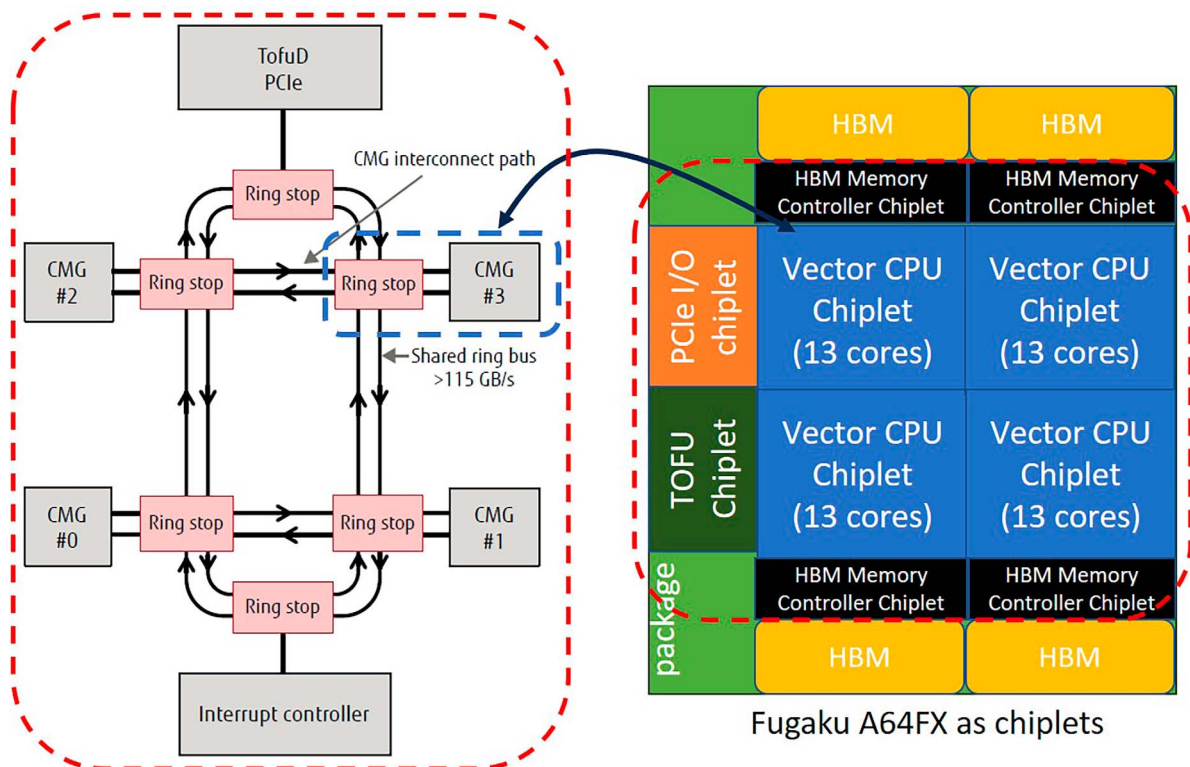
**FIGURE 5.** Fugaku (a) functional modularity and (b) proposed chiplet decomposition. [Panel (a) courtesy of Fugaku publication.]

TABLE 3. Potential logic performance for an $11 \times 11\text{-mm}^2$ compute chiplet module in 7-nm technology inferred from commercial products and prototypes.

Logic Type	Functionality in an $11 \times 11\text{-mm}^2$ Chiplet With 7-nm Technology
A64FX-like compute cores	13 cores
FPGA	350 K Achronix look-up tables 25 RISC-V cores 1.2 K Math engines 15 TOPS
Sea of cores	256 + SiFive U8 RISC-V cores
Domain-specific accelerator	MTIA from Meta, 100 8-bit TOPS ^d

FPGA: field-programmable gate array; MTIA: Meta Training and Inference Accelerator; TOPS: tera ops per second.

- › *Field-programmable gate array (FPGA)*: Commercial vendors, such as Achronix, have proposed the development of FPGA chiplets.
- › *Sea of cores*: Researchers and commercial vendors are developing accelerators that consist of tens or hundreds of small cores linked together through a high-speed network.
- › *Domain-specific logic*: Chiplets may be used to realize inference engines, LLM-specific logic, video processing logic, or other accelerators.
- › *GPU*: Chiplets that implement stand-alone GPU functionality for artificial intelligence/machine learning.

Any new chiplet needs to obey the size and power restrictions of the baseline architecture. Table 3 captures the projected functionality for $11 \times 11\text{-mm}^2$ compute chiplets of different types available commercially or prototyped today with 7-nm technology.

Beyond compute logic, package I/Os can also be changed more easily with chiplets to adapt to workload requirements:

- › The architecture can exchange memory capacity for memory bandwidth. System memory could be changed from an on-package HBM to an off-package double data rate (DDR) memory.
- › Copackaged optics chiplets could be added to increase network I/O bandwidth to other racks or bandwidth to off-package memory.

A variant may 1) change functionality without introducing new types of chiplets, for example, the relative proportion of GPUs to CPUs or the types of CPUs; 2) introduce new chiplet types, such as domain-specific

accelerators; 3) change the external I/O logic to networks or memory. Figure 6 shows examples of such variants. As any chiplets designed for the modular architecture must obey type size, heat, and power restrictions, chiplets from multiple vendors can be more easily combined to create multiple HPC ASICs. As variants are built off the same modular architecture, they will have the same physical attributes (size, power, thermals, and so on) easing system integration. Instead of extending Fugaku with separate racks of loosely coupled accelerators as is done today,⁸ hardware blade servers can be upgraded with packaged devices that include new accelerators as chiplets.

Scaling the Architecture Across Process Nodes

Semiconductor technology is constantly evolving to finer geometries, enabling more functionality to be squeezed into the same die area. The phrase “process node” is used to refer to a specific semiconductor manufacturing process, usually by the feature size of the smallest transistors that can be manufactured with the process. The Fugaku AF64 ASIC was developed in the 7-nm process node. To determine whether industry can use the modular architecture for multiple product generations (Gens), we assess its ability to scale across process nodes.^e Table 4 lists the expected scaling of power and thermal constraints for a fixed-size compute die across process nodes. To account for scaling inefficiencies, we assume that the area of a core scales linearly with feature size, not the square. We also assume that the power consumption per core only scales linearly with feature size because increases in core size are offset by improvements in power management.

As the fixed-size compute chiplet is scaled to more advanced process nodes, we expect an increase in the performance per compute chiplet, for example, in the number of compute cores per chiplet. The maximum power that can be dissipated per compute chiplet is limited by thermal constraints. The tradeoff with chiplets is that squeezing more logic into a die also requires the D2D interconnect bandwidth in the die to also grow. The power and area each have to be partitioned across compute cores and (enough) D2D circuitry to retain the D2D interconnect bandwidth per core in the design manufactured at the 7-nm^f process node. Table 5 shows that balanced growth, which preserves

^d<https://ai.meta.com/blog/meta-training-inference-accelerator-AI-MTIA/>

^e<https://www.tsmc.com/english/dedicatedFoundry/technology/logic>

^fD2D power/area consumption is largely defined by the packaging technology and not the process node.

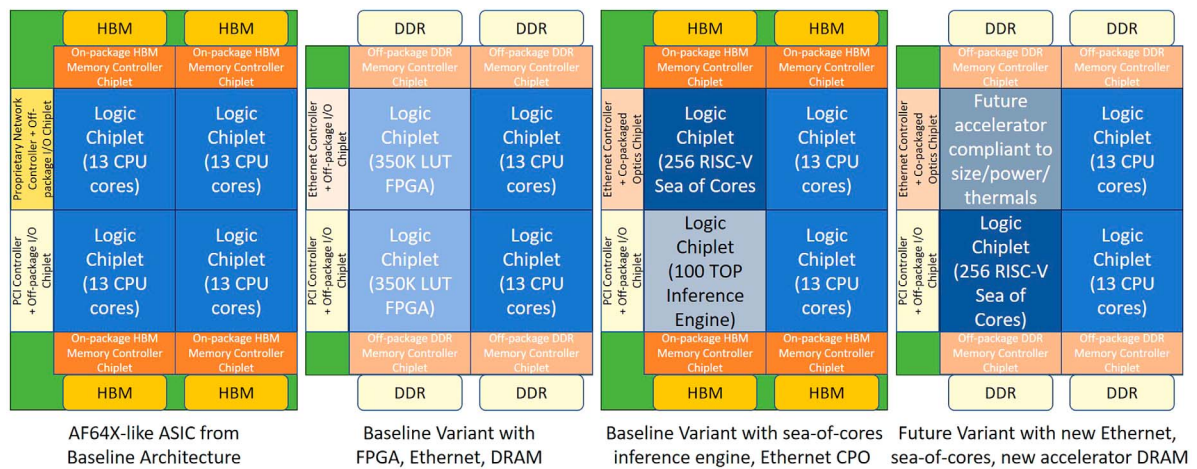


FIGURE 6. Architectural variants that are possible with a modular architecture of two chiplet types and sizes. Capabilities for $11 \times 11\text{-mm}^2$ chiplets inferred from public data. LUT: look-up table.

TABLE 4. Scaling constraints across process nodes for the compute chiplet.

Units		Architecture Generation				
		Gen 0	Gen 1	Gen 2	Gen 3	Node 4
nm	Process node	7	5	3	2	1
—	Relative power	1	0.8	0.7	0.75	0.75
W/ mm^2	Power density	0.3	0.36	0.432	0.5184	0.62208
mm^2	Die area	121	121	121	121	121
W	Die power	36.3	43.56	52.272	62.7264	75.27168
mm^2	Unit core area	4.9	3.5	2.1	1.4	0.7
W	Unit core power	2.2	1.76	1.232	0.924	0.693

Gen 0 architectural balance, that is, the bandwidth per core of the reference architecture in the 7-nm design, is possible. (At all the process nodes, core count is thermal constrained, leaving unused area that can potentially be allocated to power-efficient on-die caches.) Table 5 shows the projected performance achievable at advanced process nodes in a fixed-size $11 \times 11\text{-mm}^2$ compute chiplet.

STANDARDIZED FORM FACTORS FOR HPC MODULAR SoC

The Fugaku thought experiment leads to a potential modular HPC chiplet reference architecture. A widely accepted modular architecture will give future chiplet designers a specific design target with which their device is more likely to be used in a multichiplet HPC system in package. A similar idea has already been

TABLE 5. Core-count scaling for the compute chiplet.

Units		Architecture Generation				
		Gen 0	Gen 1	Gen 2	Gen 3	Gen 4
Tbps	Die I/O bandwidth	8	12	20	28	42
mm^2	D2D area	6.72	9.87	15.75	22.88	34.64
W	D2D power	6.4	9.58	15.18	22.22	33.68
mm^2	Logic area	114.28	111.13	105.25	98.12	86.36
W	Power for cores	29.9	33.98	37.09	40.51	41.59
—	Number of cores	13	19	30	43	60
Tbps	Bandwidth/core	0.61	0.63	0.63	0.65	0.7

TABLE 6. Mapping functional modules to discrete physical chiplet modules.

Chiplet Type	Size	Functions
Square for dense logic	Large Small	Accelerators, GPU, heavy cores Security, manageability, light cores
Tall/thin for sparse logic	Large Small	NIC, host controller, NoC + I/O, memory I/F Memory controller, NVRAM, optical I/O, bridge

NIC: negative impedance converter; NVRAM: non-volatile random-access memory.

shown to be commercially valuable to product designers. HBM is an example of a modular system that imposes significant restrictions on designers through a static form factor and interface. Yet, the value delivered makes these restrictions acceptable.

Inspecting the set of functional chiplets needed for the Fugaku thought experiment, one can observe that diverse functions can be mapped onto two physical chiplet types:

- 1) *Square*: Ideal for logic-dense functions.
- 2) *Tall/thin*: Ideal for functions with off-package I/Os, or more I/Os relative to logic.

Square chiplets can only communicate off-package through tall/thin chiplets. Table 6 shows how virtually all the functional logic common in SoCs can be mapped to these two physical types.

Modular chiplets must also be restricted in size. Too few sizes can lead to wasted area and/or limit target functionality, too many lead to interoperation challenges. We propose that the baseline design should start with support for two sizes: $11 \times 11 \text{ mm}^2$ (derived from the A64FX) and $20 \times 20 \text{ mm}^2$ (the largest used in high-volume devices). The manufacturing cost benefits⁶ from smaller die sizes fall off at approximately 100 mm^2 . As seen in Table 3, an $11 \times 11\text{-mm}^2$ chiplet appears to be adequate to meet general compute needs across a range of types of logic. So, we chose a target die size smaller than the AMD compute complex size of 200 mm^2 . Based on the parameters of an HBM interface, we propose a tall/thin chiplet size of $4 \times 11 \text{ mm}^2$.

Fully specifying the architecture will require further effort. We will need to explore the application performance and complexity tradeoffs for various parameters to define functional and physical modularity in great detail. The analysis will need to identify die size, power, thermal density, and I/O bandwidth limits; choose a packaging technology; specify limits on package size; constrain the location of D2D interconnect within a package; and specify protocols for device test, management, and operation. We aim to develop this architecture with broad industry collaboration.

⁶<https://community.arm.com/arm-research/b/articles/posts/three-dimensions-in-3dic-part-1>

CONCLUSION

The effective use of accelerators in heterogeneous HPC systems requires tight coupling between general-purpose CPUs and accelerators. Chiplet-based designs can potentially offer HPC system architects a tool to design bespoke systems that tightly couple accelerators at far lower costs than full-custom ASICs. Today, the complexities of chiplet-based designs raise their costs and limit their use to large companies. We proposed an open reference architecture, derived from the ASIC used in the Fugaku supercomputer, that can extend the hardware server-level modularity of current HPC systems to functional and physical chiplet modularity within the ASICs used in each hardware blade server. The architecture can solve several of the product development challenges experienced with chiplet-based designs by developing open standards to compose systems that ease the integration of chiplets from multiple vendors. Our analysis shows that the architecture offers a path to efficiently combine CPUs and diverse heterogeneous domain-specific accelerators into one product. This architecture can also scale effectively into the future, enabling new accelerators to be introduced without disruption, and provide performance growth with advances in logic technology. Our next steps will be to define the modular architecture in greater detail and solicit industry feedback.

ACKNOWLEDGMENTS

The authors acknowledge helpful discussions in Technical Working Group 3 in the Manufacturing Roadmap for Heterogeneous Integration and Electronic Packaging study commissioned by the National Institutes of Standards and Technology through the University of California, Los Angeles and SEMI. The authors thank Subramanian Iyer, Krutikesh Sahoo, Anu Ramamurthy, Naveed Hussain, Kemal Aygun, Gerald Pasdast, and Michel Koopmans.


REFERENCES

1. D. Reed, D. Gannon, and J. Dongarra, "HPC forecast: Cloudy and uncertain," *Commun. ACM*, vol. 66, no. 2, pp. 82–90, Feb. 2023, doi: [10.1145/3552309](https://doi.org/10.1145/3552309).

2. S. G. Cardwell et al., "Truly heterogeneous HPC: Co-design to achieve what science needs from HPC," in *Driving Scientific and Engineering Discoveries through the Convergence of HPC, Big Data and AI* (Communications in Computer and Information Science), vol. 1315, J. Nichols, B. Verastegui, A. Maccabe, O. Hernandez, S. Parete-Koon, and T. Ahearn, Eds., Cham, Switzerland: Springer-Verlag, 2020, pp. 349–365.
3. M. Asri, D. Malhotra, J. Wang, G. Biros, L. K. John, and A. Gerstlauer, "Hardware accelerator integration tradeoffs for high-performance computing: A case study of GEMM acceleration in N-Body methods," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 8, pp. 2035–2048, Aug. 1, 2021, doi: [10.1109/TPDS.2021.3056045](https://doi.org/10.1109/TPDS.2021.3056045).
4. A. M. Cabrera, A. R. Young, and J. S. Vetter, "Design and analysis of CXL performance models for tightly-coupled heterogeneous computing," in *Proc. 1st Int. Workshop Extreme Heterogeneity Solutions (ExHET)*, New York, NY, USA: Association for Computing Machinery, 2022, pp. 1–6, doi: [10.1145/3529336.3530817](https://doi.org/10.1145/3529336.3530817).
5. L. T. Su, S. Naffziger, and M. Papermaster, "Multi-chip technologies to unleash computing performance gains over the next decade," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, San Francisco, CA, USA, 2017, pp. 1.1.1–1.1.8, doi: [10.1109/IEDM.2017.8268306](https://doi.org/10.1109/IEDM.2017.8268306).
6. R. Okazaki et al., "Supercomputer Fugaku CPU A64fx realizing high performance, high-density packaging, and low power consumption," Fujitsu, Tokyo, Japan, 2020. [Online]. Available: <https://www.fujitsu.com/global/about/resources/publications/technicalreview/2020-03/article03.html>
7. E. Arima, Y. Kodama, T. Odajima, M. Tsuji, and M. Sato, "Power/performance/area evaluations for next-generation HPC processors using the A64FX chip," in *Proc. IEEE Symp. Low-Power High-Speed Chips (COOL CHIPS)*, Tokyo, Japan, 2021, pp. 1–6, doi: [10.1109/COOLCHIPS52128.2021.9410320](https://doi.org/10.1109/COOLCHIPS52128.2021.9410320).
8. K. Sano, A. Koshiba, T. Miyajima, and T. Ueno, "ESSPER: Elastic and scalable FPGA-cluster system for high-performance reconfigurable computing with supercomputer Fugaku," in *Proc. Int. Conf. High Perform. Comput. Asia-Pacific Region (HPC Asia)*, New York, NY, USA: Association for Computing Machinery, 2023, pp. 140–150, doi: [10.1145/3578178.3579341](https://doi.org/10.1145/3578178.3579341).

BABI VINNAKOTA is an engineer at Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA. He leads the Open Domain-Specific Architecture subproject within the Open Compute Project. Contact him at bvinnakota@lbl.gov.



JOHN M. SHALF is the department head for computer science at Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA. Contact him jshalf@lbl.gov.



IEEE COMPUTER SOCIETY
Call for Papers

Write for the IEEE Computer Society's authoritative computing publications and conferences.

GET PUBLISHED
www.computer.org/cfp

 IEEE COMPUTER SOCIETY  IEEE