Design Challenges of Intrachiplet and Interchiplet Interconnection

Chixiao Chen
Frontier Institute of Chips and Systems
Fudan University
Shanghai 200433, China

Jieming Yin
School of Computer Science
Nanjing University of Posts and Telecommunications
Nanjing 210023, China

Yarui Peng
Computer Science and Computer Engineering Department
University of Arkansas
Fayetteville, AR 72701 USA

Maurizio Palesi
Department of Electric Electronics and Computer Engineering
University of Catania
95125 Catania, Italy

Wenxu Cao and Letian Huang
School of Automation and School of Electronic Science and Engineering
University of Electronic Science and Technology of China
Chengdu 611731, China

Amit Kumar Singh
School of Computer Science and Electronic Engineering
University of Essex
Colchester CO4 3SQ, U.K.

Haocong Zhi
School of Software Engineering
South China University of Technology
Guangzhou 510006, China

Xiaohang Wang
School of Cyber Science and Technology
Zhejiang University
Zhejiang 310027, China

Editor's notes:
This article discusses the challenges in intrachiplet and interchiplet networks, including the need for faster simulation, better architectures, and performance requirements determined by emerging applications.

—Miquel Moreto, Universitat Politècnica de Catalunya
—Mahdi Nikdast, Colorado State University

To address the so-called area wall problem, multichiplet-based systems become a promising design paradigm in the post-Moore era. Companies like Intel, AMD, Apple, and so on design or fabricate state-of-the-art central processing unit (CPU) or graph processing unit (GPU) chips using the chiplet integration technology. Interchiplet and intrachiplet interconnection networks are key to chiplet-based many-core systems. There are a few design challenges in optimizing interchiplet and intrachiplet interconnection networks as follows.

- Chiplet-based many-core systems might integrate a large number of chiplets or cores. For example, the Celebras system-on-wafer chip system has 200,000 AI cores. Simulators are needed to accurately simulate large-scale chiplet-based many-core systems with fast speed and high accuracy.

Digital Object Identifier 10.1109/MDAT.2022.3203005
Date of publication: 29 August 2022; date of current version: 24 October 2022.

- Interchiplet interconnections have lower bandwidth and much higher latency compared to their intrachiplet counterparts, due to pin limit, the additional processing overhead of physical layer (PHY), and longer wires in interposer/redistribution layer (RDL). Therefore, interchiplet and intrachiplet interconnection networks should be carefully designed to provide highly efficient interchiplet communication.
- Chiplet-based many-core systems are designed to meet the ever-growing computation demand from various applications, like AI and high-performance computing.

Simulation methodology for chiplet-based many-core systems

Simulators, especially those cycle-accurate ones, are needed for early-stage design space exploration for chiplet-based systems. However, current multicore simulators cannot be used directly for multichiplet system simulation due to a lack of accurate interconnection modeling for interchiplet communication and the incapability of large-scale parallel simulation. Therefore, we propose a methodology for simulating multichiplet systems by integrating and modifying open-source simulators. This methodology supports parallel simulation for large-scale systems with accurate modeling of interchiplet and intrachiplet interconnection and has both distributed and shared memory models for multichiplet systems [8] (available for free download in https://github.com/FCAS-SCUT/).

The multichiplet simulation system consists of single-chiplet simulators and an intersimulator-process communication and synchronization protocol. The existing simulators (e.g., gem5, sniper, etc.) simulate individual chiplets and run in parallel, acting as the single-chiplet simulators of the simulation system. The intersimulator-process communication and synchronization protocol is proposed to simulate interchiplet communication. The multichiplet system has distributed, shared, or hybrid (e.g., globally distributed but a few chiplets share memory address) memory models.

Following the layers in the multichiplet system design as in Figure 1, the proposed simulator framework is comprised of the following layers. In the circuit and PHY, the model of latency and power are from [2]. In the microarchitectural and intrachiplet layers, open-source simulators are used to simulate the pipelines of the routers or the cores. Each individual chiplet is simulated by an existing open-source simulator. In the interchiplet network layer, a centralized network manager can configure different interchiplet network topologies according to configuration files.

In the system layer, both distributed and shared memory models are simulated with the timing model and functional model files. The functional model files carry data packets and the timing model files accumulate the latency of packets. The memory addresses are either private or shared among the chiplets, which are distinguished by address tables. In the application layer, an application programming interface (API) is provided for the programmer (benchmark developer) for remote communication. Timing and functional model files are generated by m5opt in full system (FS) mode simulators like gem5 or by system call handlers in syscall emulation (SE) mode simulators.

Path forward

With chiplet integration technology, more cores/memory units can be integrated. For the system-on-wafer chip, there can be millions of cores/memory units. Designing a fast and accurate simulator for a million-scale system becomes a must.

Digital die-to-die PHY design

As 2.5-D chiplet technology develops, interchiplet data communication was getting more concerning. Traditional SerDes high-speed links, which are normally adopted for interchip data transmission through printed circuit board (PCB) wirelines, can achieve up to 112 Gb/s [4] with only two differential pairs. However, they consume huge costs of power, area, and delay, thanks to the complex signal processing blocks, not necessary for chiplet scenarios. Moreover, such high-speed links' PHY contains analog equalizers, comparators, and even giga-hertz-sampling-rate analog-to-digital converters (ADCs), making it difficult to port between different fabrication technology. Tedious analog redesign efforts are also required. In this section, we present an all-digital PHY design method for die-to-die communication in chiplet technology. Compared with traditional Ser-Des, it features simple circuit topology, low power consumption, and good portability.

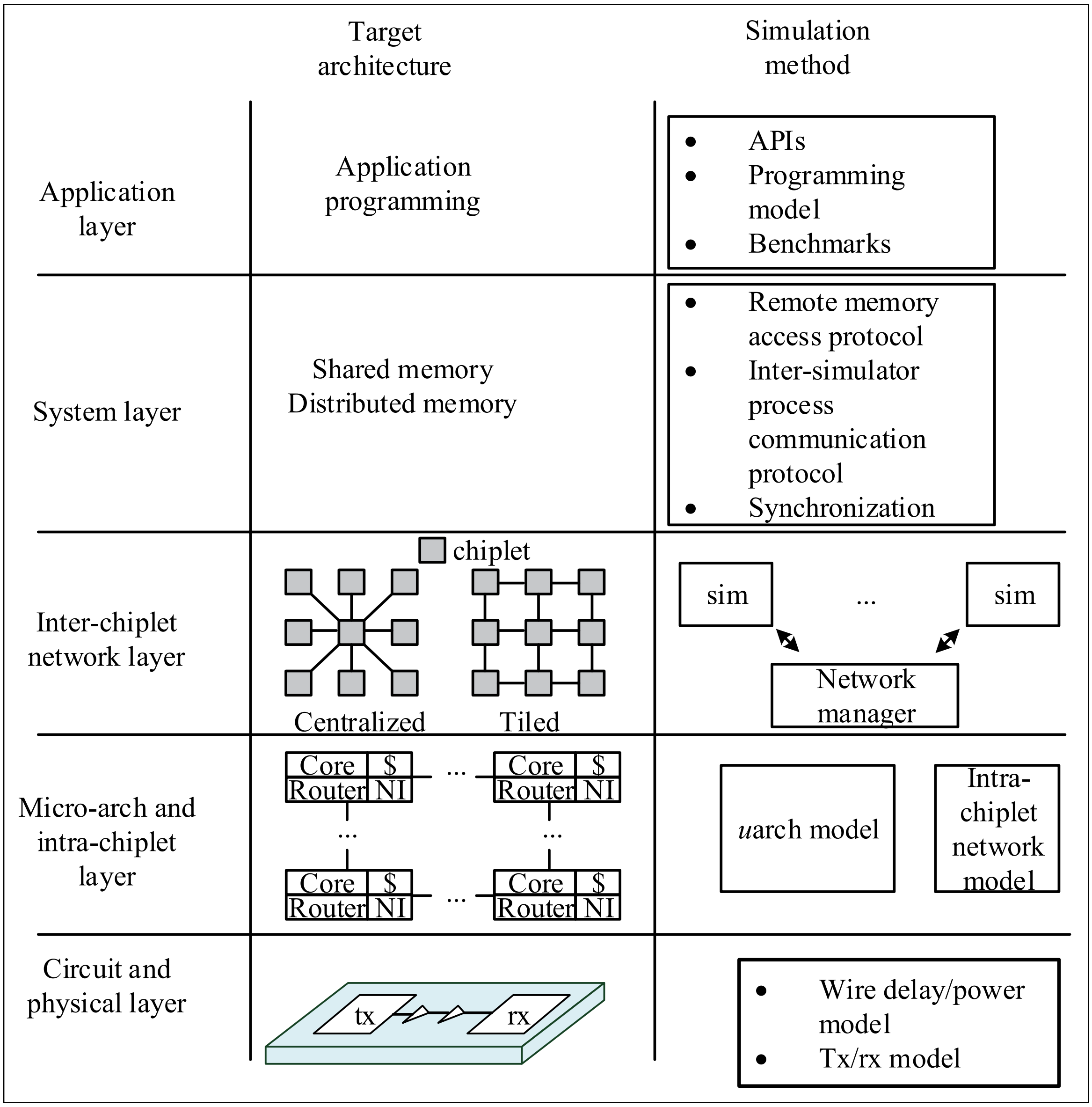


Figure 1. Overview of the simulation framework.

All-digital die-to-die PHY implementation

Figure 2 shows the overall system of a digital PHY, including a pair of a transmitter (TX) and a receiver (RX). TX converts the parallel data flow from the processor side core into a quadruple data rate serial data stream with a dedicated designed parallel-to-serial module. Rather than PLL or multiphase DLL, the PHY's clock is generated by a frequency doubler using digital-controlled delay lines (DCDLs). As a result, the data rate will be four times of input data signal due to the doubling clock and double data rate (DDR).

Multiple tri-state gates from the standard cell library are used for TX drivers. To configure various driving strengths, each TX driver contains 16 parallel tri-state gates. To verify the effectiveness, we extract S-parameters of three different channels. As the results show, 14 tri-state gates are required to drive a 4.70-mm channel if the eye-diagram width is up to 0.5 unit interval (UI), while only seven and eight tri-state gates are needed to achieve similar performance on the 1.33-mm and 2.34-mm channels.

The PHY's RX features a termination-resistor-less design. Thanks to the low-loss channel characteristic, we eliminate the termination resistor and use a standard inverter cell as the front-end comparator in RX.

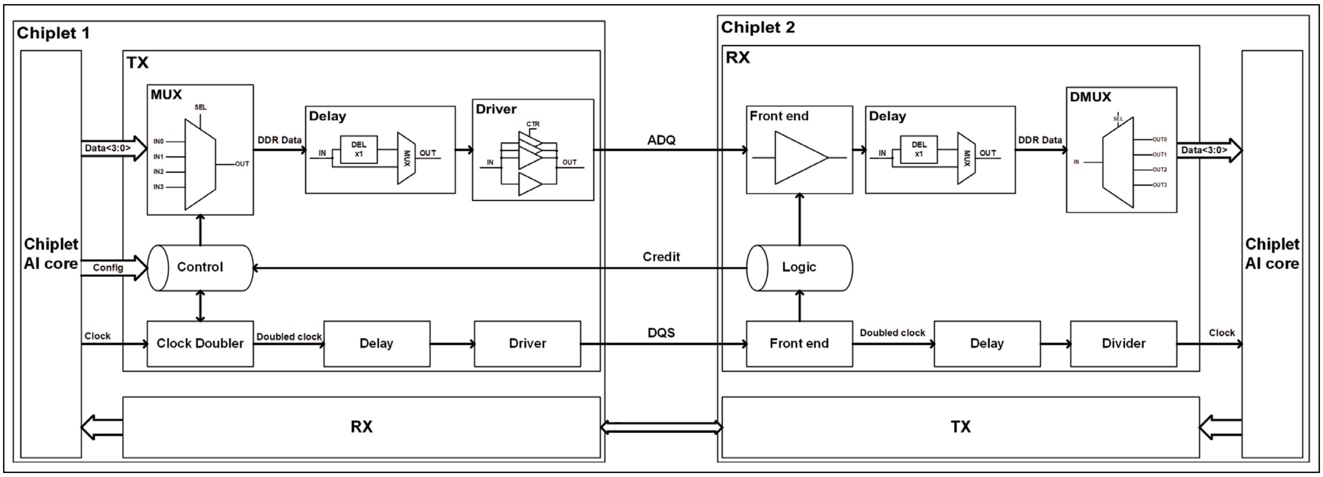


Figure 2. Overall die-to-die PHY architecture.

All components of the proposed PHY are from a standard cell library, indicating that it can be implemented by the standard digital placement and routing flow. In practice, the commercial EDA tools can accelerate the development process of this PHY and can be easily ported among different technology. Simulation results show that the entire PHY consumes 13.03 mW under a 6.4-Gb/p data rate, achieving the power efficiency of 0.41 pJ/bit.

Path forward

Though many designs inherit traditional high-speed analog Ser-Des paradigms, we still have a good vision for the future. The physical interconnect standard is a key knob enabling versatile multichiplet systems. Given that dies designed by different vendors are combined into an *integrated-chip-system,* all interfaces have to obey the same rule. Under the trend, recent standards such as BoW and UCIe are attracting more attention.

In-package network design

When designing chiplet-based systems, ensuring routing correctness can be challenging. Specifically, integrating individually designed chiplets into the same package might cause the final system to be deadlocked, even if each chiplet is deadlock-free. In this section, we present modular turn restriction (MTR), a composable routing methodology that enables modular design and integration of heterogeneous systems. Our methodology imposes turn restrictions applied only to traffic as it flows into or out of the chiplets from the interposer. Using MTR, each individual chiplet as well as the interposer is free to implement its own NoC topology and local routing algorithm.

Routing design challenge for chiplet-based systems

In multichip SoCs, chiplets can be independently designed by different vendors. As chiplets may be deployed in multiple products, including future products not even defined at chiplet design time, their global SoC routing information may not be available. Figure 3 (top) shows a multichiplet system, consisting of four GPU chiplets and a CPU chiplet. Each of the GPU and CPU chiplet contains a local NoC. These five chiplets are stacked on an active interposer that implements its own NoC to interconnect the chiplets and other common system functionality. Designing the in-package network for such a system is challenging, because while each individual chiplet's and interposer's NoC may be deadlock-free, they can still be connected together in a manner that introduces deadlocks in the final SoC (channel dependence loops that involve multiple chiplets can be formed easily). Most existing deadlock-free routing algorithms assume that complete system-level information is available, which does not necessarily hold in chiplet-based systems. Therefore, these approaches are not amenable to routing for modular, independently designed chiplets that may be reused in multiple SoC designs.

MTR methodology

MTR [7] leverages a simple-yet-powerful insight: from an individual chiplet's perspective, the rest of the system can be abstracted away into a single node. Turn restrictions are carefully applied to only the boundary routers that connect the chiplet to the abstract node, leading to tractable analysis and optimization of the granularity of individual chiplets. MTR consists of three important steps as follows.

Step 1: Select boundary routers for the target chiplet. A boundary router connects the chiplet to the interposer. Chiplet designers need to decide the number of boundary routers and their placement. The number of boundary routers determines the throughput a chiplet can sustain for sending/receiving off-chiplet traffic. Given an internal chiplet-level routing algorithm, the placement of boundary routers affects their inbound (from the interposer to the chiplet) and outbound (from the chiplet to the interposer) reachability and the on-chip traffic distribution.

Step 2: Apply turn restrictions on boundary routers. Once the boundary routers are determined, we can abstract away the rest of the system into a single node, as shown in Figure 3 (bottom). We use turn restrictions to break cycles containing the abstract node and a pair of boundary routers. The abstract node represents the rest of the system that designers of individual chiplets do not need to have knowledge of, hence turn restrictions do not apply to the abstract node. When choosing prohibited turns for boundary routers, connectivity must be preserved, so turn restrictions that cause a disconnected NoC are prohibited.

Step 3: Configure the interposer NoC. Packets are routed from one boundary router to another through the interposer. The system integrator needs to program the interposer's routing tables properly by taking into account the turn restrictions of all chiplets. To do that, certain chiplet-level information must be provided to the interposer. First, the system integrator needs to know the on-chip nodes (endpoints) that are reachable from each individual boundary router given the turn restrictions. Second, we optionally use the topological distances between each boundary router and its reachable on-chip nodes to optimize routing distances and load balancing.

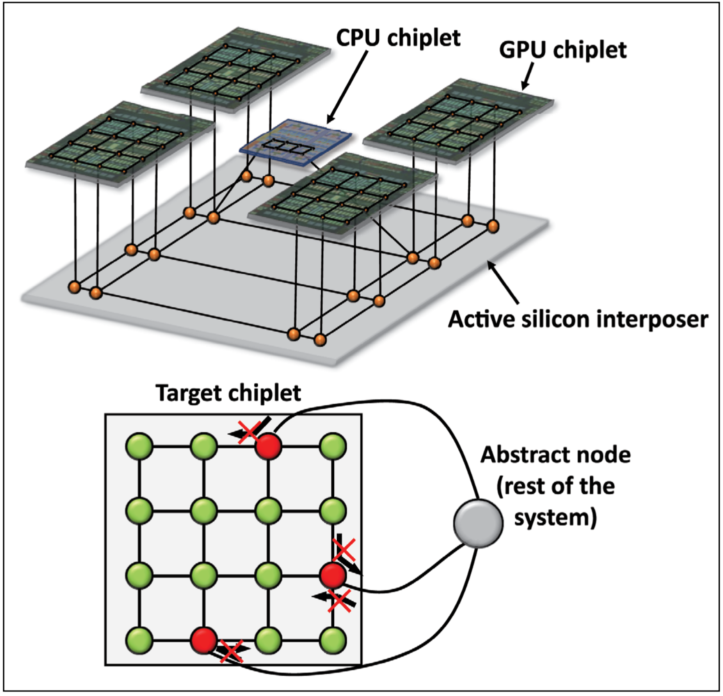


Figure 3. Baseline chiplet-based system (top) and the proposed MTR methodology (bottom).

Following the above steps, chiplet designers have the freedom to optimize their local NoC topology and routing algorithm, while the resulting system is guaranteed to be deadlock-free. In terms of microarchitectural design, each chiplet needs to implement two different routing tables. The first handles intrachiplet traffic that never goes to the interposer. The second routing table directs outbound traffic to the appropriate boundary router.

Path forward

Future chiplet-based systems can have a mix of 2.5-D and 3-D integration (some chiplets are integrated in a 2.5-D manner, while some are 3-D stacked). Finding an optimal placement and designing/optimizing in-package network topologies can be an important step during system integration.

Deadlock-free design: Model and algorithms

In this section, we propose to use the tree model to run the turn restriction algorithm (TRA) in the aforementioned MTR methodology and propose an improved method Presort-TRA to accelerate TRA. The Presort-TRA is proved to reduce the number of iterations of TRA by up to 50%.

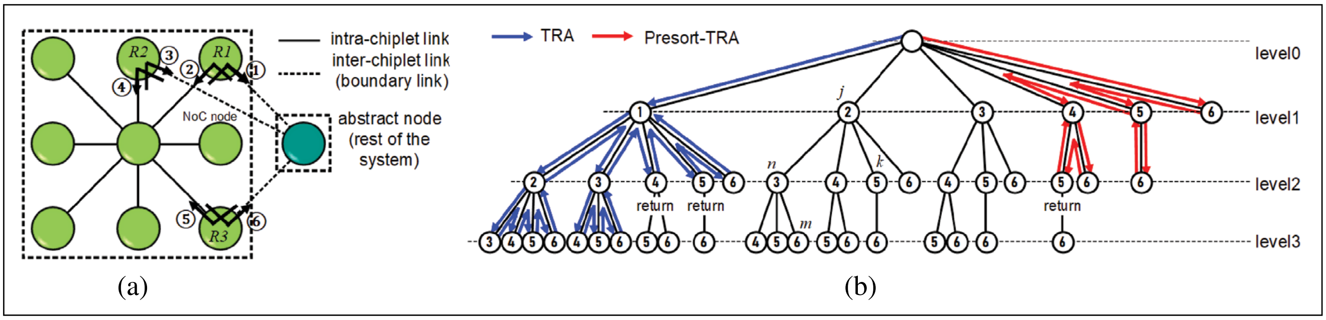


Figure 4. Example of the tree model and the interchiplet network. The boundary turns in the target chiplet are used to generate the RCT, and a search algorithm is applied to search for the optimal combination of restricted boundary turns. (a) Interchiplet network. (b) RCT and the searching procedure.

Tree model for TRA

Suppose the NoC on the target chiplet generates *N* different candidate boundary turns with restrictions. TRA can be depicted as a tree called recursive combinatorial tree (RCT), composed of all candidate boundary turns labeled 1–*N* in random order. Figure 4a shows an example of generating the boundary turns given an interchiplet network, where the target chiplet consists of three boundary routers labeled R1–R3, corresponding to six boundary turns labeled as ① to ⑥. The corresponding RCT of the system is shown in Figure 4b.

In the RCT, a node with label *k* is the boundary turn has $N - k$ child nodes labeled from $k + 1$ to *N*. When executing TRA, a depth-first search (DFS) algorithm is applied to the tree as shown in Figure 4b. The sibling nodes are visited in a random order in TRA. Figure 4b shows an example of TRA. Once a node in a higher level is visited, for example, from levels 1 to 2, the boundary turn with the current node's label is restricted. When the search returns from that higher level, the restricted node is released. Therefore, once a new node in the RCT is visited, a new turn restriction pattern is evaluated. Thus, each node except the root node in the RCT corresponds to a distinct combination of restricted boundary turns, which is represented by the node itself along with all of its nonroot parent nodes. For example, in Figure 4b, node *m* and its parent nodes *n* and *j* have the turn restriction combination of {2, 3, 6}, and node *k* and its parent node *j* have the combination of {2, 5}. In addition, MTR requires a limited number of restricted boundary turns. In Figure 4, the maximum number of restrictions is set to be 3. Thus, there are three nonzero levels in the RCT. The objective function is defined as ϕ = (AverageDistance/AverageReachability) [7] m the search algorithm. TRA searches through all boundary turns to minimize ϕ.

Presort-TRA

The efficiency of TRA can be improved by choosing the orders to label the boundary turns and to visit the sibling nodes of RCT. Therefore, the Presort-TRA algorithm is proposed to accelerate TRA by selecting the labeling order of boundary turns and the searching order of the sibling nodes. The Presort-TRA has two steps.

1. *Presorting*: All of the *N* candidate boundary turns are labeled from 1 to *N* in descending order by their ϕ values.
2. *Searching*: The RCT of each presorted boundary turns is formed. The DFS algorithm is performed on the RCT to search for the optimal combination of restricted boundary turns with minimal ϕ. Presort-TRA visits the sibling nodes in the RCT by following the descending order of their labels.

An example of how Presort-TRA works is shown in Figure 4b.

Cross-boundary chiplet package co-design

Co-design methodologies and benchmark design

2.5-D chiplet design is becoming increasingly popular as a low-cost scalable solution to further

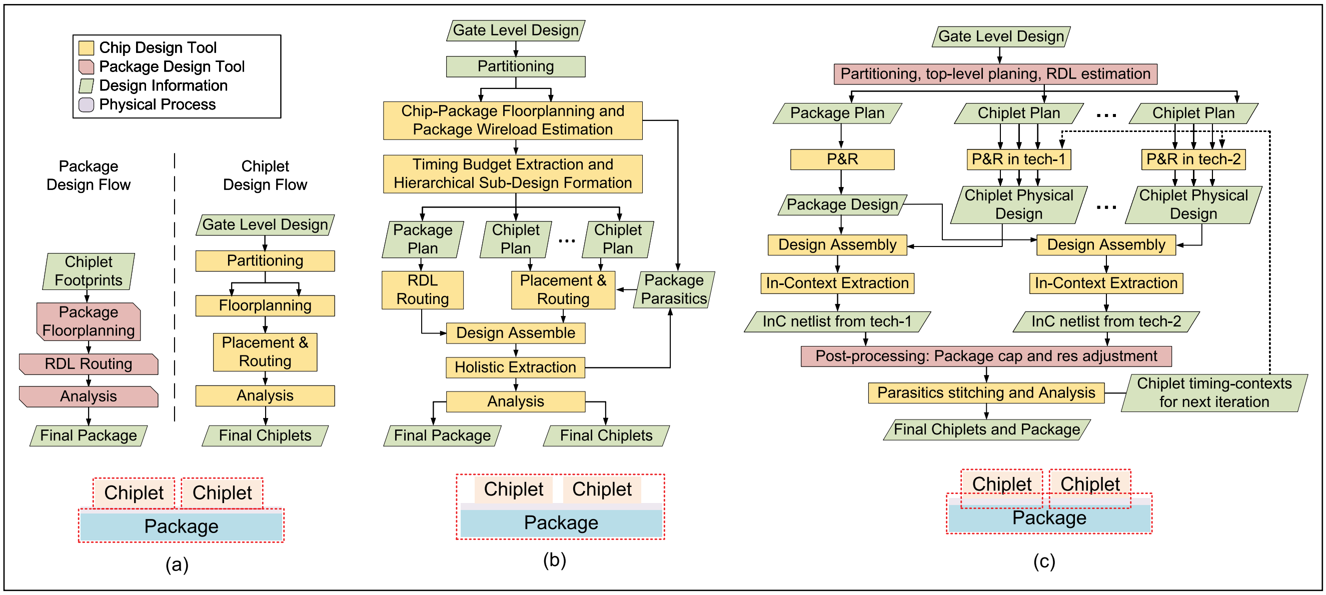


Figure 5. Comparison of three extraction flows. (a) Traditional die-by-die flow. (b) Our holistic flow for homogeneous chiplets. (c) Our in-context flow for heterogeneous chiplets.

push computational performance beyond traditional More-Moore scaling. The traditional die-by-die design flow separates engineers and computer aided design (CAD) tools into two distinct domains: very large scale integrated circuit (VLSI) and packaging. This decoupled strategy is effective for the industry to implement in their workflow by allowing design engineers to focus on a smaller knowledge domain while isolating design efforts and responsibilities. Especially for advanced 2.5-D/3-D packaging, it prevents the chiplets from reaching their full potential. A conservative interface will ensure compatibility, but also inevitably result in large design tolerances and reduce performance to achieve a broader reception.

One obvious solution is extending the 2-D design flow into a holistic approach by including every component in the design scope. The holistic system functions like a top-level giant chip design while each individual chiplet is like macros inside. It remains very compatible with the traditional physical design flow. However, this inevitability introduces other practical concerns: intellectual property (IP) protection, responsibility for integration, and fragmentation of heterogeneous integration.

To break the design boundary without imposing the need for detailed layout information from each chiplet, we designed a novel in-context design flow. Only a few top metal layers from each chiplet are exposed to the top level as the "interface layers." Similar to object-oriented programming, each chiplet only needs to share its public abstract view while holding the IP-sensitive private detailed implementations. This approach does not require complete design files from every component, while it can still capture most chiplet-package coupling for parasitic extraction, noise, timing, and power analysis. Revealing the noncritical properties, our in-context extraction remains heterogeneous-friendly and ensures IP protection. All three methods are compared in Figure 5.

To demonstrate our 2.5-D design methodologies, we design a microcontroller system based on ARM Cortex-M0 with seven metal layers for chiplet routing and three RDL layers. We then compare different partition methodologies and choose to utilize the knowledge of the system architecture to come up with an architecture-aware partition.

With our holistic flow [2], both package and chiplets are assembled into the same VLSI design environment. Therefore, we can extract the distributed parasitic netlist of the entire chip-package system and perform timing and power analysis. Then, we compare the results with the monolithic 2-D implementation. Using the traditional die-by-die flow, chiplets and packages are separately optimized. As a result, the highest system frequency drops to 245 MHz for the unoptimized 2.5-D system, which is much worse compared to the 2-D monolithic implementation (333 MHz). However, with an iterative timing optimization using holistic extraction, the timing degradation is almost eliminated, and the system performance is comparable to a single chiplet (300 MHz). Effective for homogeneous designs, holistic extraction is still computationally expensive to process the entire 2.5-D system layout.

Designed for heterogeneous integration, our in-context flow can be used to accelerate the extraction process [3]. It only includes essential interface layers from both the package and chiplet during extraction and then emerges the parasitic database with postprocessing. Also, multiple dies are extracted separately to allow the extraction of heterogeneous chiplets in parallel. We compare the extraction accuracy of holistic extraction to in-context extraction using our 2.5-D design. Our in-context extraction achieves less than 1% error compared to a holistic design. This allows the whole heterogeneous systems to achieve the same 300-MHz max frequency. Our in-context extraction remains heterogeneous-friendly and new rule decks can be calibrated incrementally by reusing existing rule decks. This approach does not require complete design files from every component, while it can still capture most chiplet-package coupling for parasitic extraction, noise, timing, and power analysis.

Paths forward

Our heterogeneous 2.5-D design flow and CAD tool PowerSynth [1] will further enable integrating both Si chips with SiC power electronics devices while ensuring performance, reliability, and low cost.

Multiobjective hardware mapping co-optimization for chiplet-based DNN accelerators

The quest toward computation efficiency together with the ever-increasing computation demand from emerging workloads is leading to the adoption of a scalable design paradigm that combines multiple subaccelerators (SAs) to build a large accelerator system. Such SAs can come in the form of chiplets that are connected by means of a network-on-package (NoP). In this context, hardware configuration (i.e., the number, placement, interconnection of the chiplets, and their configuration, i.e., number of processing elements, buffer sizes, etc.) and mapping strategy (i.e., how the workload is spatially and temporally scheduled) are the top two most important factors determining the overall accelerator performance.

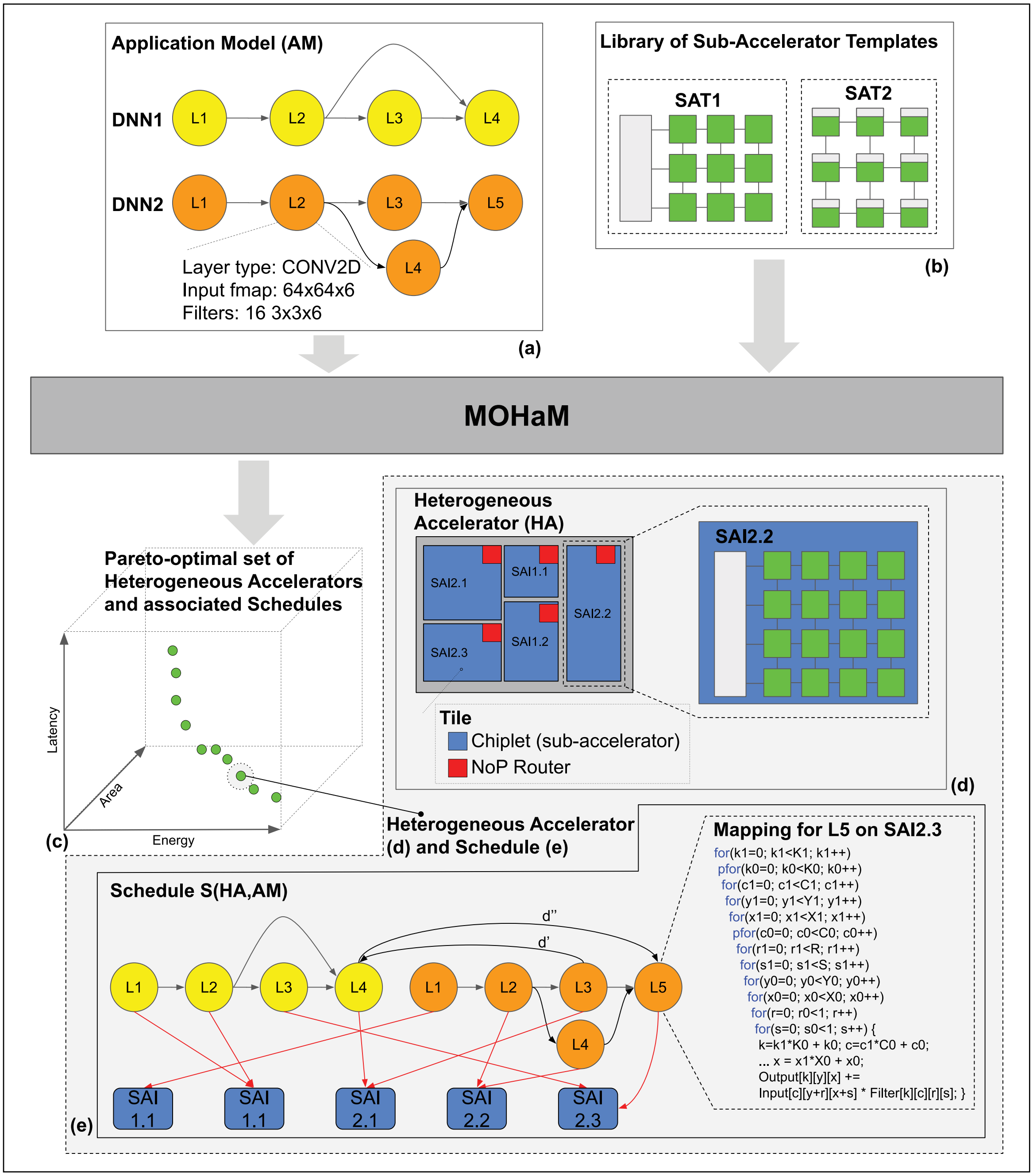
This section introduces a multiobjective hardware-mapping co-optimization framework (MOHaM) for multichip-module (MCM)-based multitenant deep neural network (DNN) accelerators. It is the first attempt at simultaneous exploration of hardware configuration and mapping strategy for multitenancy aimed at deriving Pareto-optimal system instances that optimize toward multiple conflicting design objectives.

MOHaM overview

The inputs and outputs of MOHaM are reported in Figure 6. It takes into input the application model (AM) and a library of parameterized SAs templates (SATs) and provides in the output the Pareto-optimal set of heterogeneous accelerators (HAs) with the corresponding *optimal schedules* that minimize energy, latency, and area.

An AM is a set of DNN models that generate the workload (Figure 6a). The DNN models in the AM are assumed to be independent of each other and thus can be executed in parallel. A parameterized SAT is a reconfigurable accelerator supporting different mappings by means of reconfiguration and parameterized in terms of the number of PEs and buffer sizes (Figure 6b). When each of the free parameters of an SAT is set, we obtain an SAT instance (SAI).

Each point of the Pareto-optimal set provided by MoHaM represents an HA and its specific schedule (Figure 6c). An HA is specified by the set of its SAIs, the NoP that allows chiplets to communicate with each other and with the external DRAM through the set of available memory interfaces (MIs), and a placement function that, for each SAI and MI, returns the tile where they are placed on. For instance, Figure 6d shows an HA formed by five SA chiplets interconnected by an NoP. The SAs are instances of two parameterized SATs, namely, SAT1 and SAT2, as shown in Figure 6b. Figure 6e shows an example schedule. Black edges denote the layer's dependencies, whereas red edges denote the mapping layer M into the SAs. Here, both L3 of DNN2 and L4 of DNN1 are mapped on the same SAI2.1 (i.e., instance 1 of SA2). Dependency d' defines their execution order, that is, L3 has to be executed before L4. Similarly, d'' defines the execution order



```
for(k1=0; k1<K1; k1++)
 pfor(k0=0; k0<K0; k0++)
  for(c1=0; c1<C1; c1++)
   for(y1=0; y1<Y1; y1++)
    for(x1=0; x1<X1; x1++)
     pfor(c0=0; c0<C0; c0++)
      for(r1=0; r1<R; r1++)
       for(s1=0; s1<S; s1++)
        for(y0=0; y0<Y0; y0++)
         for(x0=0; x0<X0; x0++)
          for(r=0; r0<1; r++)
           for(s=0; s0<1; s++) {
            k=k1*K0 + k0; c=c1*C0 + c0;
            ... x = x1*X0 + x0;
            Output[k][y][x] +=
            Input[c][y+r][x+s] * Filter[k][c][r][s]; }
```

Figure 6. Overall flow for MOHaM. (a) Application model. (b) Library of subaccelerator templates. (c) Pareto-optimal configurations. (d) Subaccelerator instances. (e) Applications schedule.

between L4 of DNN 1 and L5 of DNN2, that is, L4 has to be executed before L5.

MOHaM optimization engine

MOHaM optimization engine adopts a two-step approach. In both steps of the search, the Timeloop/Accelergy [5] framework is used as the cost model.

The first step is the mapping of each layer of the AM onto each SAT in the library. This step is built by leveraging multiobjective evolutionary approach to DNN hardware mapping (MEDEA) [6] that allows the search for a Pareto set of mappings of a layer on a specific architecture, using a genetic algorithm approach augmented with custom genetic operators.

In the second step, the Pareto mappings found for each layer are considered for the global scheduling search. The global scheduler is based on the NSGA-II multiobjective genetic algorithm. The selection and survival phases are those of the original algorithm. However, several custom genetic operators have been implemented to increase sampling efficiency, thus finding better individuals in less time, but also because only a small part of the genomes are valid. Searching with default random mutation and crossover operators is therefore not feasible. The result of a global scheduler run is a Pareto-optimal set of accelerators composed of heterogeneous SAs and, for each of them, the optimal schedule in such a way as to minimize energy, makespan, and area.

Path forward

Future research in this area should be devoted to the exploration of design space taking into account the architectural parameters of the communication subsystem and the different Silicon interposer technologies.

In this article, challenges in designing interchiplet and intrachiplet interconnection systems in chiplet-based systems were discussed. We expect the future lies in joint consideration of all possible aspects, *that is,* cross-level optimization and design. ■

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61971200; in part by the Zhejiang Laboratory under Grant 2021LE0AB01 and Grant 2021PC0AC01; in part by the Open Research Grant of the State Key Laboratory of Computer Architecture Institute of Computing Technology, Chinese Academy of Sciences, under Grant CARCH201916; in part by the Major Scientific Research Project of Zhejiang Laboratory under Grant 2021LE0AC01; in part by the Key Technologies Research and Development Program of Jiangsu (Prospective and Key Technologies for Industry) under Grant BE2021003; and in part by the National Key Research and Development Program of China under Grant 2019QY0705. This paper was presented at NOCS 2022 and appears as part of the *IEEE Design&Test* Special Issue.

References

[1] I. A. Razi et al., “PowerSynth design automation flow for hierarchical and heterogeneous 2.5-D multichip power modules,” *IEEE Trans. Power Electron.*, vol. 36, no. 8, pp. 8919–8933, Aug. 2021.

[2] M. A. Kabir and Y. Peng, “Chiplet-package co-design for 2.5D systems using standard ASIC CAD tools,” in *Proc. ASPDAC*, Jan. 2020, pp. 351–356.

[3] M. A. Kabir, D. Petranovic, and Y. Peng, “Coupling extraction and optimization for heterogeneous 2.5D chiplet-package co-design,” in *Proc. ICCAD*, Nov. 2020, pp. 1–8.

[4] J. Kim et al., “A 112 Gb/s PAM-4 transmitter with 3-tap FFE in 10 nm CMOS,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 102–104.

[5] A. Parashar et al., “Timeloop: A systematic approach to DNN accelerator evaluation,” in *Proc. ISPASS*, Mar. 2019, pp. 304–315.

[6] E. Russo et al., “MEDEA: A multi-objective evolutionary approach to DNN hardware mapping,” in *Proc. DATE*, Mar. 2022, pp. 226–231.

[7] J. Yin et al., “Modular routing design for chiplet-based systems,” in *Proc. ISCA*, Jun. 2018, pp. 726–738.

[8] H. Zhi et al., “A methodology for simulating multi-chiplet systems using open-source simulators,” in *Proc. NANOCOMM*, Sep. 2021, pp. 1–6.

Chixiao Chen is an associate professor with the National Key Laboratory of Integrated Chips and Systems, Fudan University, Shanghai, China. His research interests include mixed-signal integrated circuit design and custom intelligent software-hardware co-designs. Chen received a PhD in microelectronics from Fudan University. He is a Member of IEEE.

Jieming Yin is a professor with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests lie in computer architecture, with emphasis on SoC system integration, network on chips, and machine-learning-aided computer system design. Yin received a PhD from the University of Minnesota, Twin Cities, Minneapolis, MN, USA.

Yarui Peng is an assistant professor with the Computer Science and Computer Engineering Department, University of Arkansas, Fayetteville, AR, USA. His research interests are computer-aided design, analysis, and optimization for emerging technologies

and multichip packages, such as 2.5-D/3-D ICs and wide band-gap power electronics. He studies design methodologies and optimization algorithms for parasitic extraction, signal integrity, power integrity, and thermal reliability. He also develops design automation tools for power electronics to improve performance, reliability, and productivity. Peng received a PhD in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA. He is a Member of IEEE.

Maurizio Palesi is an associate professor in computer engineering with the University of Catania, Catania, Italy. His current research activity is focused in the area of domain-specific architectures. He is a Senior Member of IEEE.

Wenxu Cao is pursuing an MS with the School of Automation, University of Electronic Science and Technology of China, Chengdu, China, researching in domain-specific architecture and inter-/intrachip communication.

Letian Huang is an associate professor with the University of Electronic Science and Technology of China (UESTC), Chengdu, China. His research interests include multicore system-on-chips, network-on-chips, and heterogeneous integrated microsystems. Huang received a PhD in communication and information systems from UESTC. He is a Member of IEEE.

Amit Kumar Singh is a senior lecturer (associate professor) with the University of Essex, Colchester, U.K. His research interests are design/optimization of multicore-based computing systems for performance, energy, temperature, reliability, and security. Singh received a PhD from Nanyang Technological University, Singapore.

Haocong Zhi is pursuing a master's with the School of Software Engineering, South China University of Technology, Guangzhou, China. Her research interests include multichiplet systems.

Xiaohang Wang is a professor with Zhejiang University, Zhejiang, China. His research interests include many-core architecture, power-efficient architectures, optimal control, and NoC-based systems. Wang received a PhD in communication and electronic engineering from Zhejiang University.

■ Direct questions and comments about this article to Xiaohang Wang, School of Cyber Science and Technology, Zhejiang University, Zhejiang 310027, China; baikeina@163.com.