Clock Delivery Network Design and Analysis for Interposer-Based 2.5-D Heterogeneous Systems

Gauthaman Murali, Heechun Park, Eric Qin, *Graduate Student Member, IEEE*, Hakki Mert Torun, *Graduate Student Member, IEEE*, Majid Ahadi Dolatsara, Madhavan Swaminathan, Tushar Krishna, *Member, IEEE*, and Sung Kyu Lim, *Senior Member, IEEE*

***Abstract*—The 2-D CMOS process technology scaling may have reached its pinnacle, yet it is not feasible to manufacture all computing elements at lower technological nodes. This has opened a new branch of chip designing that allows chiplets on different technological nodes to be integrated into a single package using interposers, the passive interconnection mediums. However, establishing a high-frequency communication over an entirely passive layer is one of the significant design challenges of 2.5-D systems. In this article, we present a robust clocking architecture for a 2.5-D system consisting of 64 processor cores. This clocking scheme consists of two major components, namely, interposer clocking and on-chiplet clocking. The interposer clocking consists of clocks used to achieve global synchronicity and clocks for interchiplet communication established using the AIB protocol. We synthesized these clocking components using commercial EDA tools and analyzed them using standard tools, on-chip, and package models. We also compare these results against a 2-D design of the same benchmark and another 2.5-D clocking architecture. Our experiments show that the absolute clock power is up to 16% less, and the ratio of clock power to system power is up to 4% less in the 2.5-D design than its 2-D counterpart.**

***Index Terms*—2.5-D clocking, clock metrics, heterogeneous systems, hierarchical clocking, RISC-V architecture.**

I. Introduction

THOUGH the 2-D IC process technology has been scaling down continuously, few circuit modules, such as memory and analog modules, do not scale down as fast as CMOS technology to the lowest technology node. Sometimes, digital elements in a chip, which, when scaled down to a lower technology node, provide a minimal performance improvement. In such a scenario, scaling down the technology node may not be worth the cost incurred. The monolithic 2-D design does not support integrating heterogeneous technologies; we have to scale down the entire design to a lower technology node complicating the entire design process. However, this increases the chance of rendering multiple dies unusable in a wafer. In such scenarios, 2.5-D designs help by integrating heterogeneous modules (chiplets) on an interposer. This process helps improve the yield and makes the entire manufacturing process time-efficient by allowing reuse of past chiplet designs [1].

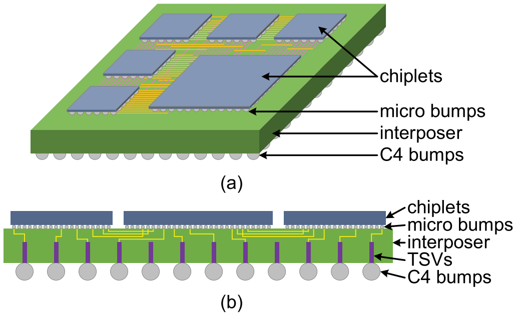

Fig. 1. Chiplet integration using an interposer-based 2.5-D system [2]. (a) Interposer-based 2.5-D IC. (b) Cross-sectional view of 2.5-D IC.

The 2.5-D technique enables the system designers to design any SoC by choosing off-the-shelf chiplets and heterogeneously integrating them into the target SoC, thereby drastically reducing the design time and design complexity by allowing reuse of predesigned chiplets as plug-and-play modules. Fig. 1 shows an example of an interposer-based 2.5-D design and its cross-sectional view illustrating the interchiplet connections and the package connections. Similar to a ball grid array (BGA) package, microbumps are created across the surface of chiplets to establish connections with the interposer. Furthermore, the interchiplet routing is done by connecting the corresponding microbumps using wires routed across the interposer over different metal layers. The external signals are routed across the metal layers through the through-silicon vias (TSVs) before they exit the package via C4 bumps.

2.5-D designs integrating CPU, GPU, and high bandwidth memory (HBM) have started hitting the commercial markets. As the commercialization of 2.5-D designs has begun, it is necessary to compare and analyze different aspects of 2.5-D designs against existing 2-D designs to have a clear idea of the new technology. So far, researchers have compared the performance of different interposer technologies and methodologies to improve the signal and power integrity of signals on the interposer. The other major component that affects a design's

Manuscript received June 17, 2020; revised October 11, 2020 and January 1, 2021; accepted January 28, 2021. Date of publication February 24, 2021; date of current version April 1, 2021. This work was supported by the DARPA CHIPS Project under Award N00014-17-1-2950. *(Corresponding author: Gauthaman Murali.)*

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: gauthaman@gatech.edu; limsk@ece.gatech.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TVLSI.2021.3058300.

Digital Object Identifier 10.1109/TVLSI.2021.3058300

performance is the clocking behavior, and it is mandatory to ensure that any new design methodology provides a better clock performance than the current methodologies, or at least that it does not degrade the existing behavior. In this article, we propose a clocking methodology for a 2.5-D system, built using vertically integrated electronic design automation (EDA) flow for chiplet creation and integration, and report its clock metric analysis results, such as clocking power, skew, and latency.

II. Motivation

Several works [3]–[7] have been performed related to improving different aspects of clock trees in 3-D ICs. However, very little work is found on 2.5-D clock networks. Huang and Zheng [8] propose a global 2.5-D clocking architecture with one chiplet acting as the clock source to all the dies. They try to achieve synchronicity by using a one-driver-per-relay architecture, which minimizes the clock skew in 2.5-D designs by dynamically tuning the clock driver's delay in the source chiplet to match the clock delays across the interposer to various chiplets. However, this clocking architecture has several disadvantages.

1) The source chiplet distributes high-frequency clock signals across the interposer. With the interposer being passive, reconstructing the clock signals at the destination chiplets becomes a herculean task. Fig. 3 shows how the clock degrades through the interposer as the frequency increases.
2) When the number of chiplets in the system is high, all clock signals originating from a single-source chiplet may lead to crosstalk issues on the interposer data signals. This, in turn, makes it difficult to reconstruct the data signals at the destination chiplets.
3) In the multiclock domain systems, all PLLs must be placed within the source chiplet, leading to heating issues in the source die.

To overcome these disadvantages, we propose a hierarchical clock network to improve the performance and reduce the clock power consumption of 2.5-D designs. A hierarchical clocking architecture tries to minimize the routing of high-frequency clocks on the passive interposer. Fig. 2 shows the degradation of a 1-GHz clock signal through transmission lines of different lengths on a passive interposer. It is observed that a 1-GHz frequency signal gets degraded when the interconnect length is longer than 7.5 mm. Thus, routing a 1-GHz signal for more than 7.5 mm on the interposer leads to clock integrity issues. As we increase the frequency of operation, this degradation occurs at lower interconnect lengths. Therefore, based on the frequency of operation, and if the interposer clock interconnect length exceeds the threshold within which the clock integrity can be maintained, hierarchical clocking architecture should be used to achieve better clocking performance. Also, the degradation of low-frequency signals on the interposer is less and can be easily reconstructed using double-inverter-based buffers. The degradation shown occurs in the absence of crosstalk. In a real circuit, the chances for crosstalk are higher, and the signal degradation caused by interposer may lead to an incorrect reconstruction of clock signals at the receivers within the destination chiplets. In addition to addressing these issues, a hierarchical clock network also works well for multiclock domain systems. Thus, this research focuses on proposing a scalable clocking architecture for homogeneous/heterogeneous 2.5-D systems. Also, we perform a comparison between 2-D and 2.5-D clock networks to estimate if 2.5-D designs can provide better clock performance than 2-D designs.

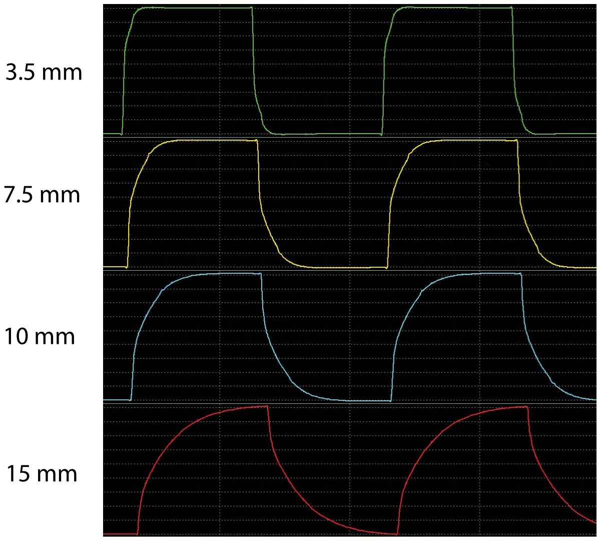

Fig. 2. HSPICE simulation of 1-GHz clock through transmission lines of different lengths on a passive interposer.

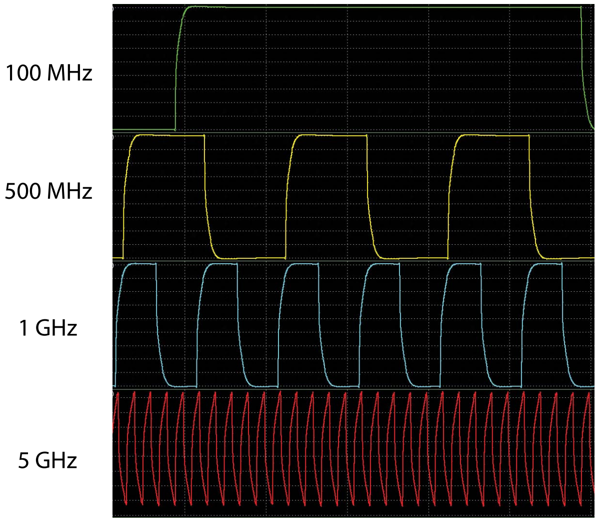

Fig. 3. HSPICE simulation of different clock frequencies through passive interposer.

III. Benchmark Architecture

A. 64 RISC-V Core Architecture

For our study, we adopt the Rocket-64 [2] architecture, a 64 core processor architecture based on RISC-V RocketCore [9] implemented in TSMC 28 nm. This architecture contains around eight million gates. The 2.5-D design of Rocket-64 architecture consists of eight Octa-Core RocketCore processor chiplets containing an L1 cache (0.25 MB) in each of them, eight L2 cache (1MB each) chiplets, a four-channel memory controller (MC) chiplet [10] (both logical

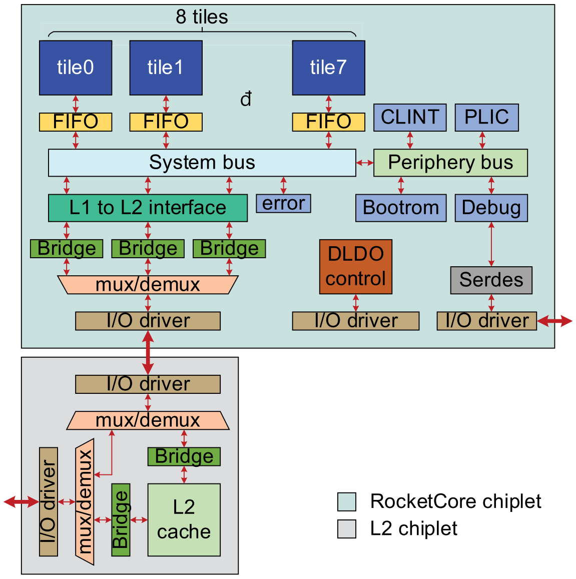

Fig. 4. Internal architecture of RocketCore and L2 chiplets [9].

and physical layers) for the 64 cores to interact with external DRAMs, one integrated voltage regulator (IVR) chiplet, and eight digital low-dropout (DLDO) voltage regulators to power up the entire 2.5-D system and a Network-on-Chip (NoC) chiplet (with eight routers) to arbitrate among the eight RocketCore-L2 cache chiplet pairs and the MC chiplet. The internal architecture of RocketCore and L2 cache chiplet pair is shown in Fig. 4. In this article, we use the homogeneous Rocket-64 described in [2] and a heterogeneous version of the same to understand the benefits of our proposed clocking architecture. In the heterogeneous version of Rocket-64, we reimplement RocketCore and NoC chiplets using TSMC 16 nm, MC chiplet in TSMC 40 nm, and retain the L2 Cache, DLDO voltage regulator chiplets at TSMC 28-nm node, and IVR chiplet at the GF 130-nm technology node. Fig. 5 shows the 2.5-D floorplan of our modified Rocket-64 architecture. One of the major advantages of 2.5-D systems is the ability to integrate chiplets at multiple technology nodes. With the current CMOS technology trend, 2.5-D designs allow building complete systems even if all the modules cannot be scaled down to the latest technology node. We have mimicked this scenario using the heterogeneous variant of Rocket-64 to observe the clock behavior of a system containing chiplets designed at multiple technology nodes.

B. Interposer Technology

We use TSMC CoWoS [11] 65-nm silicon interposer to integrate these heterogeneous chiplets. The interposer design rules used in this article are shown in Table I, and the cross section view of a TSMC CoWoS[1]-based interposer is shown in Fig. 6. The chiplets are connected to the interposer with a minimum spacing of 100 μm through μ-bumps. The delays introduced by these μ-bumps are significantly smaller than the current system delays of nanoseconds [12]. Based on TSMC CoWoS 65-nm silicon interposer design rules, the μ-bumps on the chiplets have a pitch of 40 μm. The μ-bump pitch plays a significant role in determining the size of the chiplets, especially as the chiplet technology scales down. Chiplets with many IO bumps cannot be scaled down beyond a certain limit due to the restriction imposed by μ-bump pitch constraint. This can lead to reduced chiplet area utilization. With current trends in interposer technology, the μ-bump pitch can be as small as 20 μm, as indicated in [13], thereby improving the area utilization of chiplets at lower technology nodes.

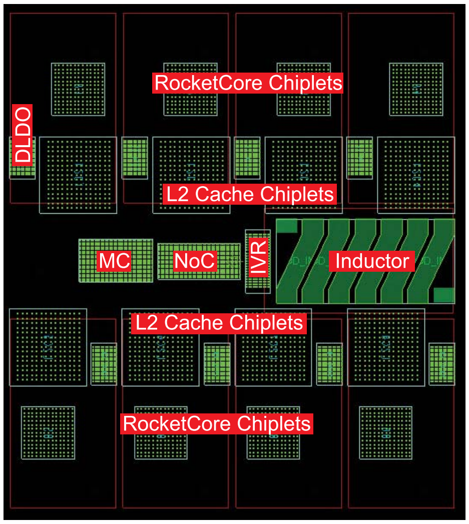

Fig. 5. Floorplan of our 2.5-D architecture that consists of 27 chiplets implemented in four different commercial technology nodes and one inductor. We use TSMC CoWoS 65-nm interposer technology. The chiplet layouts are shown in Fig. 22.

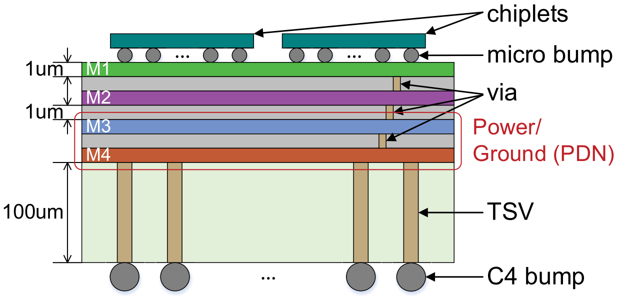

Fig. 6. Cross-sectional view of the interposer.

Our design's passive silicon interposer uses four metal layers, the top two for signal and clock routing, and the bottom two for PDN routing. The redistribution layers (RDLs) and the via connecting the layers have a thickness of 1 μm each. The width and pitch of the RDLs are 0.4 μm each. This plays a significant role in determining the maximum frequency of signals routed through the interposer with minimum crosstalk effects. The width of vias between each metal layer is 0.7 μm. For external communication, the signals are routed through 10-μm-wide and 100-μm-long TSVs to C4 bumps. The pitch of C4 bumps is 180 μm.

C. Architectural Differences Between 2-D and 2.5-D Rocket-64

The monolithic 2-D counterpart of the benchmark used contains the same components as that of the 2.5-D design

[1] Registered trademark.

TABLE I
DESIGN RULE OF TSMC CoWoS [11] 65-NM SILICON INTERPOSER TECHNOLOGY USED IN THIS WORK

Metal layer#	4
Metal thickness	1 μm
Dielectric thickness	1 μm
Min. line width/spacing	0.4 μm/0.4 μm
Via size	0.7 μm
Through Via size/depth	10 μm/100 μm
Die-to-die spacing	100 μm
MICRO-BUMP PITCH	**40** μm
C4 bump pitch	180 μm
PDN width/spacing	40 μm/90 μm

TABLE II
ARCHITECTURAL FEATURES: 2-D VERSUS HOMOGENEOUS 2.5-D ROCKET-64 DESIGN

Module	2D Design	2.5D Design	Technology Node
Rocket Core	8	8	28nm
L2 Cache	8	8	28nm
Memory Controller	4	4	28nm
Routers	12	8	28nm
IVR	0	4	28nm
DLDO	0	8	130nm
PLL	8	8	28nm

TABLE III
ARCHITECTURAL FEATURES: 2-D VERSUS HETEROGENEOUS 2.5-D ROCKET-64 DESIGN

Module	2D Design	2.5D Design	Technology Node
Rocket Core	8	8	16nm
L2 Cache	8	8	28nm
Memory Controller	4	4	40nm
Routers	12	8	16nm
IVR	0	4	28nm
DLDO	0	8	130nm
PLL	20	24	28nm

except for the NoC, IVR, DLDO, and AIB protocol logic. In the 2.5-D system, we use eight routers to arbitrate the transactions between the eight Rocket-8 chiplets and the MC chiplet, whereas, in the 2-D design, we use 12 routers. The difference in the router count between the two designs is due to the AIB protocol for interchiplet communication in the 2.5-D design. The AIB protocol restricts the number of data I/O signal bumps to 40 to reduce the number of signals routed on the interposer. Therefore, we streamline the entire I/O bus of each chiplet into a 40-bit bus using an appropriate FIFO synchronization mechanism. However, this adds latency in interchiplet communication. In the 2-D design, we do not restrict the number of connections between modules. Therefore, we use 12 routers for arbitrating signals between the Rocket-8 and the MC. Instead of increasing the router's interface width, we use the same 40-bit router in our 2-D design. Hence, we need more routers to route the increased interconnections between NoC and MC modules in a 2-D design. Fig. 23 shows our single-chip 2-D design of the Rocket-64 [2] architecture. Tables II and III tabulate the architectural features of 2-D and 2.5-D designs of Rocket-64.

IV. 2.5-D CLOCK NETWORK SYNTHESIS

Any clock network in a 2.5-D design consists of two components: interposer clocking and on-chiplet clocking. In our design, we use the Cadence SiP Layout [14] for interposer clock routing and clock tree synthesizer (CTS) in the Cadence Innovus Implementation System [15] for on-chiplet clock routing. To further motivate our research, we first discuss the disadvantages of using a flat-clocking architecture in 2.5-D designs.

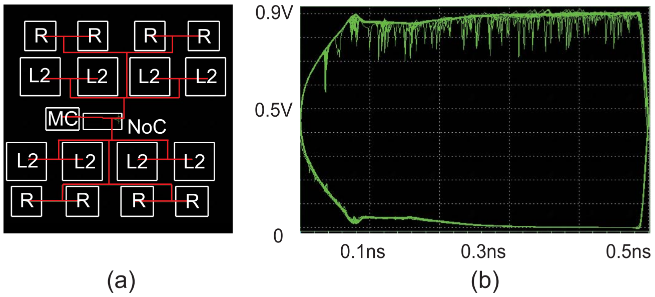

Fig. 7. (a) Flat 1-GHz interposer clock tree. (b) Eye diagram.

A. Flat Clocking Architecture

What would happen if we deliver high-frequency clock signals directly to all chiplets? To answer this question, we route a 1-GHz clock tree in the interposer for comparison. In this case, we assume that all the chiplets are operating at 1 GHz so that PLLs are not necessary. Fig. 7(a) shows the routing topology used for this 1-GHz clock and its eye diagram. First, we observe an excessive degradation of the flat 1-GHz clock signal from the eye diagram due to impedance mismatch at this frequency. Due to impedance mismatch, the signal reflected from the receiver superimposes the transmitted signal causing voltage-level fluctuations in the clock signal. This creates downward spikes in the high phase of the clock up to 0.58 V, as seen from the eye diagram in Fig. 7(b). However, the full swing range is from 0 to 0.9 V. Thus, even slight crosstalk can force the clock to a voltage less than 0.5 V, causing glitches in the reconstructed clock signal, thereby leading to functional issues. Thus, delivering a high-frequency clock signal through a long-distance passive interposer interconnect without using any clock buffers can be dangerous. On the other hand, if a 100-MHz signal is routed through such long interposer routes, the eye's degradation is almost zero, as shown in Fig. 14.

B. Hierarchical Clocking Architecture

To overcome the drawbacks of flat clocking architecture, we propose a hierarchical style clocking architecture for 2.5-D systems. In hierarchical clocking, the target functional clock frequency is achieved through an intermediate reference frequency signal. Fig. 3 shows that the lower the clock frequency on the interposer, the lesser the degradation. Therefore, routing a lower frequency clock on the interposer and then upgrading it to a higher frequency functional clock using phase-locked loop (PLL) circuits within chiplets helps avoid all signal integrity issues caused by the passive interposer layer in 2.5-D designs. As the final functional clock is generated through multiple frequency scaling stages, we call this technique a hierarchical clocking architecture.

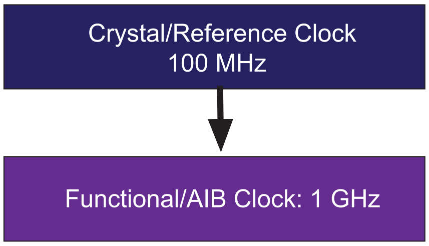

Fig. 8. Our hierarchical clocking architecture for homogeneous Rocket-64.

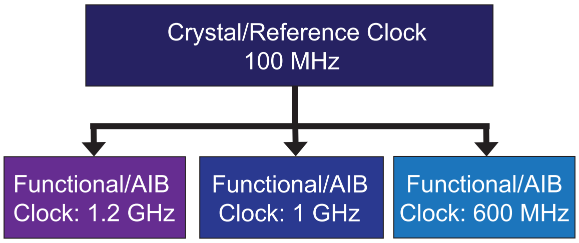
Crystal/Reference Clock
100 MHz
Functional/AIB Clock: 1.2 GHz
Functional/AIB Clock: 1 GHz
Functional/AIB Clock: 600 MHz

Fig. 9. Our hierarchical clocking architecture for heterogeneous Rocket-64.

The reference clock aligns all the functional clocks generated by the PLLs at each rising edge, thereby ensuring synchronicity across all the chiplets. For example, a 100-MHz reference clock signal used to generate 1-GHz clock signals in different chiplets aligns all the 1-GHz clocks once every 10 ns. The PLL drift on the 1-GHz clock during the 10-ns period is within the safe limits to ensure synchronous operation. Such reference clocks (global alignment signals) are typical in commercial 2-D ICs to ensure the synchronicity of different PLLs in the design and ensure synchronous communication across connected modules. The functional clock generated is then used either for the operation of the chiplet in which the clock is generated or for interchiplet communication. Synchronous communication between chiplets is established through special chiplet clocking architecture involving duty cycle correction (DCC) and clock edge skewing to ensure data and clock signals align appropriately.

The hierarchical clocking architecture and different clocking components in it are explained in detail in the following. We demonstrate the advantages of hierarchical clocking architecture considering two different variants of the Rocket-64 processor benchmark.

C. Hierarchical Clocking for Two Variants of Rocket-64

Passive interposers prohibit the optimization of high-frequency clocks using buffer insertion techniques. This makes it necessary to downgrade the clock frequency if we route the clock signal over a greater distance. Taking this limitation into account, we use the clocking architecture shown in Fig. 8 for our homogeneous 2.5-D Rocket-64 and the clocking architecture shown in Fig. 9 for our heterogeneous 2.5-D Rocket-64 design. The homogeneous Rocket-64 uses a single clock domain (1 GHz) for its functional operations, whereas the heterogeneous variant uses multiple clock domains (1.2 GHz,1 GHz, and 600 MHz).

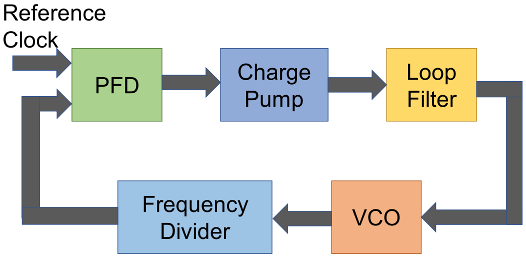

Fig. 10. PLL architecture.

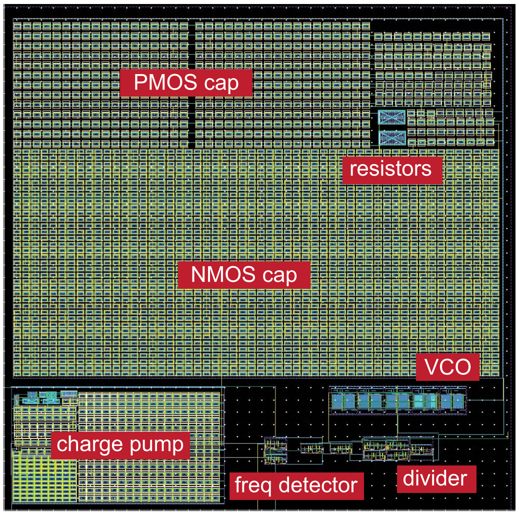

Fig. 11. Our PLL layout using TSMC 28 nm.

D. Our PLL Architecture and Design

The crystal/reference clock is scaled to high-frequency clocks using PLL circuits within different chiplets based on the variant of Rocket-64. We use a typical analog PLL [16] on TSMC 28-nm node that consists of a ring oscillator-based voltage-controlled oscillator (VCO), a digital phase-frequency detector (PFD), a digital frequency divider, a charge pump, and a loop filter. The loop filter of the PLL consists of pMOS and nMOS capacitors and polyresistors to generate the appropriate control signal from the charge pump output to control the frequency of oscillation of VCO. The architecture and layout of the PLL are shown in Figs. 10 and 11, respectively. The area of the PLL is 42 433 μm^2. The lock time of the PLL is approximately 110 ns.

E. Reference Clock Routing

The interposer clock network forms the base of clocking in any 2.5-D system. A crystal/reference clock from an external 100-MHz crystal oscillator is routed into the 2.5-D system through the C4 bumps, TSVs, and via stack and metal layer up to the clock microbumps of the chiplets consisting of PLLs using Allegro signal router in the Cadence SiP Layout tool. We have ensured that the clock C4 bump is placed as close as possible to the PLLs so that the reference clock does not undergo much degradation over the passive interposer layer. Due to the interposer's passive nature, we cannot have

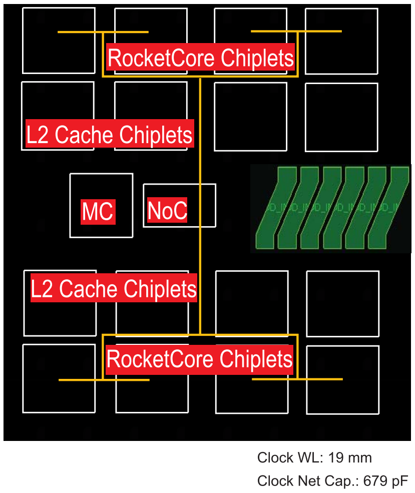

Fig. 12. 100-MHz crystal/reference interposer clock tree on interposer of homogeneous Rocket-64.

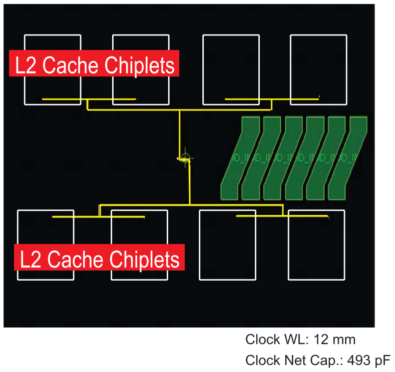

Fig. 13. 100-MHz crystal/reference clock tree on interposer of heterogeneous Rocket-64.

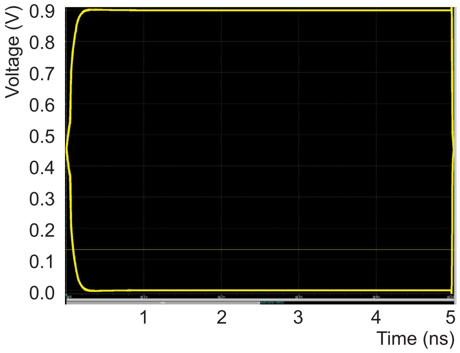

Fig. 14. Eye diagram of 100-MHz reference clock on the interposer.

equalizers on the interposer layer to reduce the effect of crosstalk. We reduce the crosstalk on clock signals by ensuring that the clock C4 and microbumps are surrounded by either power, ground, or semistatic signal (resets) C4 or microbumps. However, these measures do not entirely ensure a noise-free clock, and hence, we use AIB I/O drivers, which are capable of reconstructing cleaner clock signals from the degraded ones.

In a hierarchical clock routing architecture, the reference clock structure plays a vital role in achieving global synchronicity in the system. We experiment by placing the PLLs within different chiplets in homogeneous and heterogeneous Rocket-64 designs to check if the hierarchical clock routing performs well irrespective of the PLL's location. The reference clock tree structure in the two variants of Rocket-64 is explained in the following.

1) Homogeneous Rocket-64: In homogeneous Rocket-64, the reference clock is manually routed on the interposer layer in an H-Tree fashion to all the Rocket-8 and NoC chiplets. Within each of these chiplets, a PLL scales the reference clock to 1 GHz. The reference clock tree is shown in Fig. 12.

2) Heterogeneous Rocket-64: In heterogeneous Rocket-64, the reference clock is manually routed on the interposer layer in an H-Tree fashion to all the L2 cache chiplets. Within each L2 cache chiplets, three PLLs scale the reference clock to 600-MHz, 1-GHz, and 1.2-GHz clocks. The reference clock tree is shown in Fig. 13.

The manual routing required to build the H-Tree of the reference clock is very minimal. The number of manual routes required is equal to the number of chiplets containing PLLs, which is 8 for both the variants of the 64-core processor, in our case. Even for larger 2.5-D designs, with appropriate PLL placement and efficient chiplet technology node choice, the number of manual reference clock routing needed can be minimal. A wholly balanced reference clock H-Tree is necessary to make sure that every PLLs is synchronous. When the PLLs are synchronous, the high-frequency clock signals generated within Rocket-8, L2 cache, and NoC chiplets align on every reference clock edge to ensure global synchronicity. The reference clock's low frequency makes it less prone to interposer degradation, and this can be observed in the eye diagram of the reference clock shown in Fig. 14.

F. RLGC Models of Interposer Interconnects

We perform clock metric analysis of different types of clocks in our design using RLGC models of C4 bumps, TSVs [17], [18], vias, and interposer transmission line models [19]. We design high-speed interposer interconnect models using a Bayesian framework coupled with machine learning techniques and used them for our simulations. We also use multiconductor transmission line models, as shown in Fig. 15, to simulate the crosstalk behavior among coupled TSVs and wires. An example of the HSPICE model used to simulate the reference clock characteristics is shown in Fig. 16. We do not use additional receiver termination on the passive interposer layer, as adding termination on all microbumps occupies a lot of space. In the case of high-frequency signals that are routed over shorter wirelength, the reflective losses due to impedance matching are minimum over a lossy silicon transmission line. However, reference clock signals are critical in a design and run over long distances. The losses on high-frequency signals are intolerable. However, low-frequency signals do not face such issues even without the additional receiver termination, as can be seen from the eye diagram in Fig. 7.

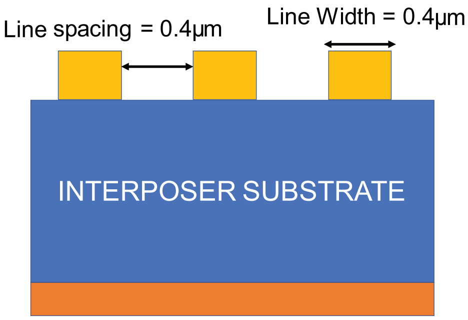

Fig. 15. Microstrip model of the multiconductor transmission line.

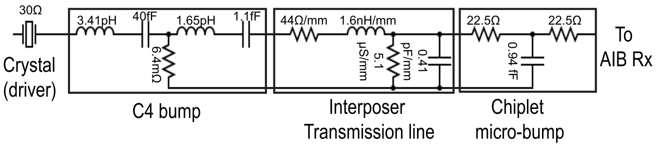

Fig. 16. HSPICE model used to simulate reference clock characteristics.

G. Functional Clock Generation

After the reference clock routing is completed, we calculate the clocks' propagation delay from the crystal clock C4 bump to various chiplets based on the RLGC models of C4 bumps, TSVs, and interposer wires, as described in Section III-C. We use this delay as the source latency to the PLLs within each NoC, Rocket-8, and L2 cache chiplets for on-chiplet clock tree synthesis. We model these clock delays using Synopsys design constraint (SDC) file [20], which is a Synopsys file format to model clock/reset related constraints. We input the SDC file to Cadence Innovus CTS to build and optimize the clock trees of 1 GHz within the Rocket-8 and NoC chiplets in the case of homogeneous Rocket-64 and the clock trees of 1.2 GHz, 1 GHz, and 600 MHz within the L2 cache chiplets in the case of heterogeneous Rocket-64. In heterogeneous Rocket-64, the clocks generated within the L2 cache chiplets are forwarded to the Rocket-8 and NoC chiplets for their operation. The 1.2-GHz clock forwarded from L2 cache chiplets to the corresponding Rocket-8, NoC, and MC chiplets in heterogeneous Rocket-64 is shown in Fig. 17.

H. AIB Clock Forwarding Technique

Adding a PLL within every chiplet, especially in cases where a chiplet operates synchronously with another chiplet, is not an efficient clocking technique. In a case where two chiplets communicate as a master–slave pair, the slave chiplet should derive its clock signal from the master's clock. We use Intel Advanced Interface Bus [21] (AIB) protocol. This chiplet standard uses special AIB drivers and clock forwarding architecture for interchiplet communication to establish interchiplet communication. These AIB drivers help in regenerating the degenerated interposer signals. Similar to most I/O buffers, the AIB buffer uses back to back inverters with great noise margin. When a signal/clock gets degenerated with noise on the passive interposer layer, these buffers help regenerate them into clean signals, provided the signal is not completely distorted on the interposer. Fig. 18 shows the AIB clock forwarding architecture.

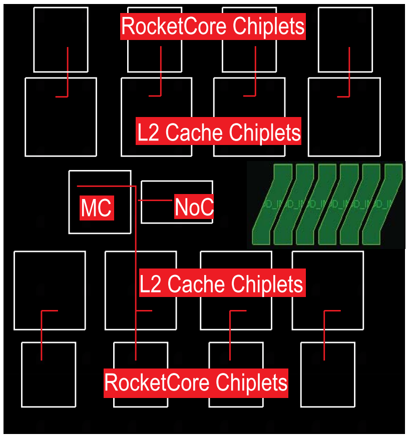

Fig. 17. 1.2-GHz functional clock tree on the interposer of heterogeneous Rocket-64.

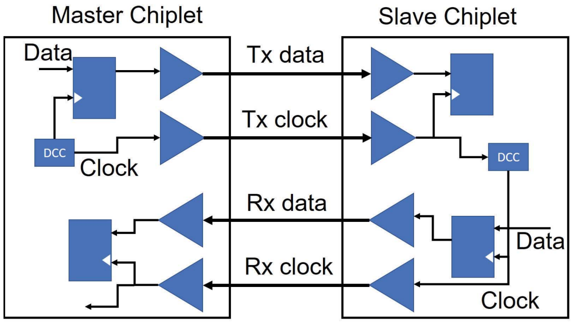

Fig. 18. Intel AIB [21] clock forwarding architecture.

We have implemented the following master–slave communications in our design using the AIB clock forwarding architecture: communication between Rocket-8 and L2 Cache, L2 Cache and NoC, and NoC and MC. The buffers shown in Fig. 18 are special AIB buffers, which can be configured to act as either clock buffer or data buffer and aid in the reconstruction of high-speed signals that get degraded while passing through shorter distances over the interposer. Fig. 19 shows the layout of our AIB driver. Fig. 20 shows a 1-GHz clock signal through a 3-mm wire on a passive silicon interposer and the corresponding clock signal reconstructed by the AIB buffer.

The clock forwarded from the master chiplet is routed back to the master by the slave chiplet when the slave responds to the master. Duty cycle corrector (DCC) circuits are used to correct duty cycle variations if the clock signal's duty cycle is affected by transmission over the interposer. The performance metrics, power, and area of AIB buffers for an operating frequency of 1 GHz are given in Table IV.

Using these AIB protocol features, we designed the following clocking style for master–slave chiplet pair communication in both variants of Rocket-64.

1) Homogeneous Rocket-64: The 1-GHz functional clock generated in each Rocket-8 chiplet is forwarded along with

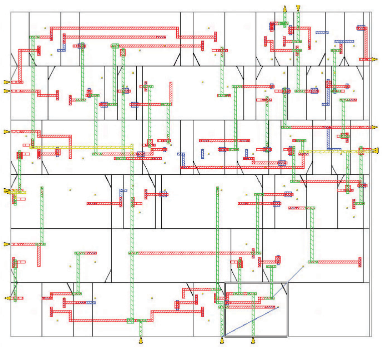

Fig. 19. Layout of our digitally synthesized AIB transceiver.

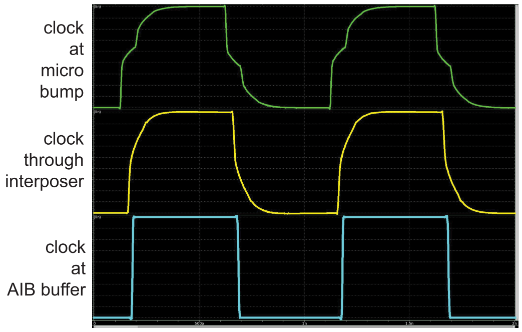

Fig. 20. HSPICE clock wave forms of the 1-GHz clock through the interposer and AIB transceiver.

TABLE IV
PROPERTIES OF THE AIB BUFFER

Metric	Value
Op. Frequency	1 GHz
Area	56 μm^2
Gate Count	69
Total Power	19 μW
Clock Power	6.1 μW
Clock Latency	4 ps

the data in a 40-bit AIB bus to the corresponding L2 Cache chiplet. When the L2 cache chiplet responds to the Rocket-8 chiplet, the same clock is rerouted internally within the L2 Cache chiplet and forwarded along with the L2 cache data back to the Rocket-8 chiplet, similar to the structure shown in Fig. 18. We also route the 1-GHz clock from the NoC chiplet with the data signals between the L2 Cache and NoC chiplet pairs and NoC and MC chiplet pairs in a similar fashion.

2) Heterogeneous Rocket-64: The 1.2-GHz functional clock forwarded to Rocket-8 chiplet from the L2 cache chiplet is forwarded back along with the data, in a 40-bit AIB bus, to the corresponding L2 Cache chiplet. When the L2 cache chiplet responds to the Rocket-8 chiplet, the same clock is rerouted internally within the L2 Cache chiplet and forwarded along with the L2 cache data back to the Rocket-8 chiplet, similar to the structure shown in Fig. 18. We also route 600-MHz clock and data signals between the L2 Cache and NoC chiplet pairs and NoC and MC chiplet pairs in a similar fashion.

It is necessary to ensure that, when the clock is forwarded along with the data, the signals are not skewed to break synchronicity in the communication. We use the Cadence SiP tool's Allegro signal router to perform the AIB clock routing by constraining the skew limits. The routing lengths are ensured to be within a safe limit of 3.5 mm. The clock and the data signals in a bus are routed such that the maximum skew between them is 3.96 ps. Unlike the reference clock, these AIB clocks pass through the interposer only for a shorter length. They are also regenerated within the chiplets as they pass from one chiplet to another, making them robust to degradation despite their high frequencies. Also, the clock signals are surrounded by semistatic signals in an AIB bus to reduce crosstalk.

TABLE V
CHIPLET CLOCK METRICS OF HOMOGENEOUS ROCKET-64

	L2 Cache	Rocket-8	NoC	Mem-Ctr
Target Clock Period (GHz)	1	1	1	1
Technology node (nm)	28	28	28	28
Clock Latency (ps)	264	566	216	273
Clock Skew (ps)	7	12	34	41
Clock Jitter (ps)	21	18	21	45
Clock Power (mW)	5	170	47	19
Clock Buffer Count	420	409	1,230	328
Clock Wire Length (mm)	35	428	110	39
Clock Net Sw. Cap. (pF)	24	487	145	31

I. Functional Clock Routing in Slave Chiplets

Once the AIB interposer clock routing is done in our homogeneous Rocket-64 design, we calculate the propagation delays of 1-GHz clocks on the interposer and clock skew between the clock and data signals in the AIB bus. We use these to generate SDC constraints for functional clock tree synthesis of L2 cache and MC chiplets.

In heterogeneous Rocket-64, we calculate the propagation delays of 1.2 GHz- and 600-MHz clocks on the interposer and clock skew between the clock and data signals in the AIB bus to generate SDC constraints for functional clock tree synthesis of Rocket-8, NoC, and MC chiplets.

The high-frequency clock signals are more degraded than the 100-MHz reference clock signal. Hence, it is necessary to reconstruct a clean clock signal from the degraded interposer clock. However, as mentioned earlier, the AIB buffer used as a part of clock forwarding architecture takes care of reconstructing the degraded clock signals.

Fig. 21 shows the chiplet layouts and their corresponding clock trees of homogeneous Rocket-64, and Fig. 22 shows that of heterogeneous Rocket-64. Tables V and VI provide the corresponding clock metrics of homogeneous and heterogeneous Rocket-64, respectively. We observe that Rocket-8 chiplet contains the most complex on-chip clock tree.

V. MONOLITHIC 2-D VERSUS 2.5-D COMPARISON

A. Experimental Setup

For the single-chip monolithic 2-D design of Rocket-64, we perform a hierarchical design using the TSMC 28-nm technology node. Unlike our 2.5-D design, the 2-D design cannot involve multiple technology nodes. The 2-D design

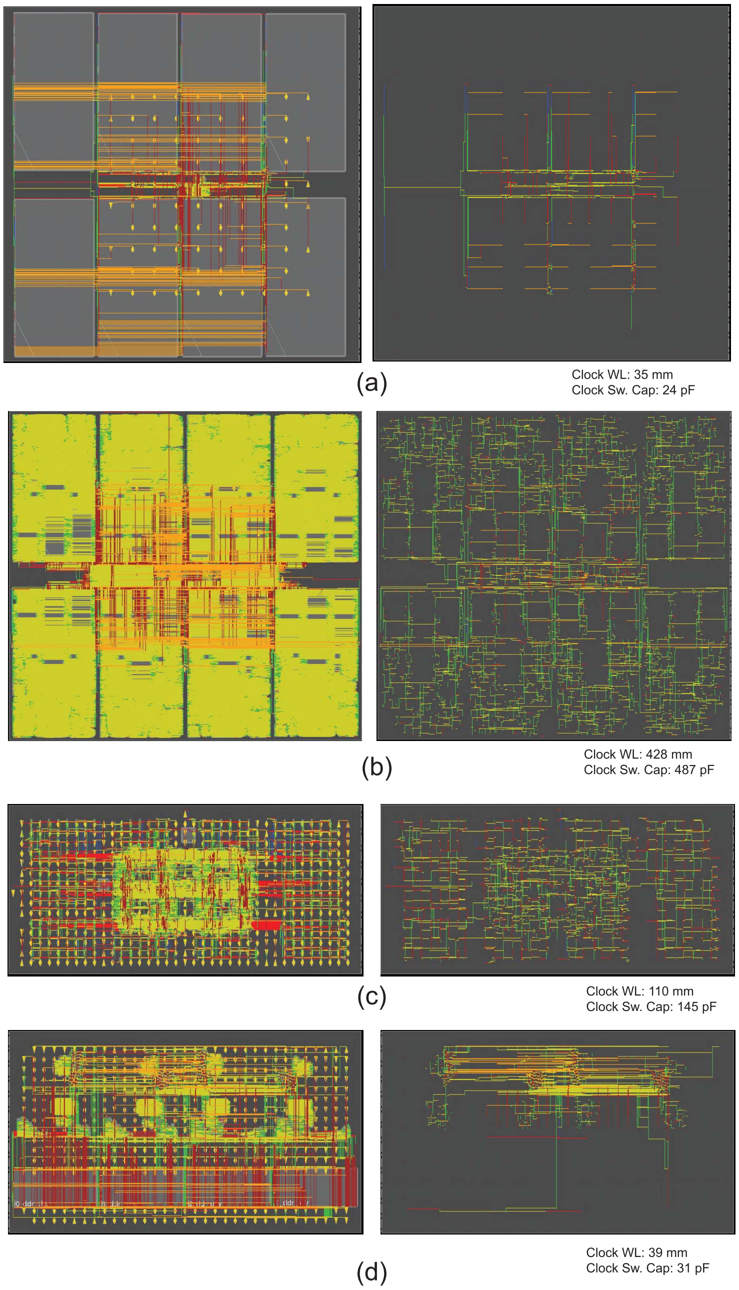

Fig. 21. Full-chip design and clock tree of our homogeneous Rocket-64 architecture. (a) L2 cache (TSMC 28 nm), (b) Rocket-8 (TSMC 28 nm), (c) NoC (TSMC 28 nm), and (d) MC (TSMC 28 nm) chiplets. Not drawn in scale. IVR (GF 130 nm) and DLDO (TSMC 28 nm) are shown in [2].

TABLE VI
CHIPLET CLOCK METRICS OF HETEROGENEOUS ROCKET-64

	L2 Cache	Rocket-8	NoC	Mem-Ctr
Target Clock Period (GHz)	1	1.2	1.2	0.6
Technology node (nm)	28	16	16	40
Clock Latency (ps)	152	239	309	526
Clock Skew (ps)	2	2	43	144
Clock Jitter (ps)	11	7	9	22
Clock Power (mW)	38	139	24	20
Clock Buffer Count	369	5,866	1,215	610
Clock Wire Length (mm)	41	313	97	55
Clock Net Sw. Cap. (pF)	28	357	79	42

does not require IVR and DLDO modules as the power delivery in a 2-D system is less stringent than that of a 2.5-D system. For a fair comparison, we use 100-MHz clock as the bus clock and use a PLL for each group of eight Rocket cores to scale it to their functional frequencies. Similar to the 2.5-D design, we design two variants of 2-D Rocket-64 with: 1) Rocket-8, NoC, and DDR-PHY modules operating at 1.2 GHz, L2 cache at 1 GHz, and MC at 600 MHz and 2) all modules operating at 1 GHz. Fig. 23 shows the overall design and the multidomain clock network of the 2-D design. Tables VII and VIII compare the clock power consumption of two variants of 2-D and 2.5-D designs.

B. 2-D Versus Homogeneous 2.5-D Rocket-64

We first compare our monolithic 2-D design against the homogeneous 2.5-D design. We make the following observations.

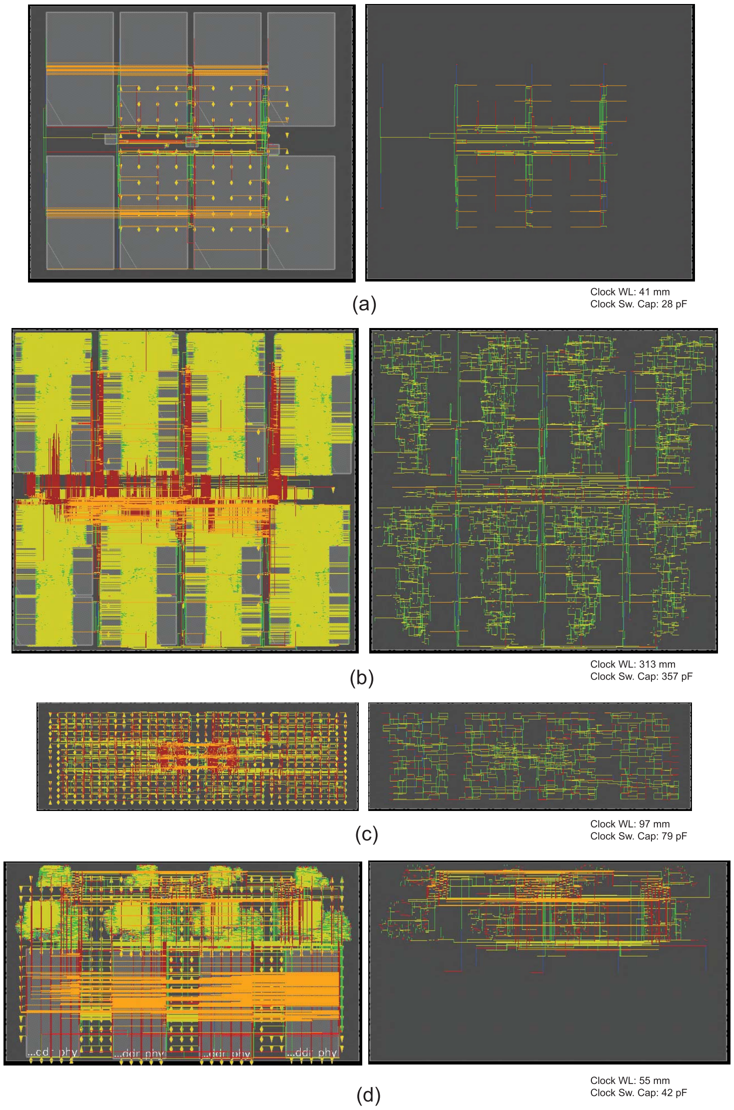

Fig. 22. Full-chip design and clock tree of our heterogeneous Rocket-64 architecture. (a) L2 cache (TSMC 28 nm), (b) Rocket-8 (TSMC 16 nm), (c) NoC (TSMC 16 nm), (d) MC (TSMC 40 nm) chiplets. Not drawn in scale. IVR (GF 130 nm) and DLDO (TSMC 28 nm) are shown in [2].

TABLE VII

CLOCK POWER COMPARISON: 2-D VERSUS 2.5-D HOMOGENEOUS ROCKET-64

Module (2D) or Chiplet (2.5D)	2D Design	2.5D Design
Eight Rocket-8	1,580 mW	1,260 mW
Eight L2 Cache	8.8 mW	40 mW
Memory Controller	10 mW	19 mW
NoC Router	81 mW	47 mW
PLL	110.3 mW	101.25 mW
Overall Power	**1.79** W	**1.57** W

TABLE VIII

CLOCK POWER COMPARISON: 2-D VERSUS 2.5-D HETEROGENEOUS ROCKET-64

Module (2D) or Chiplet (2.5D)	2D Design	2.5D Design
Eight Rocket-8	1,640 mW	1,110 mW (16nm)
Eight L2 Cache	8.8 mW	32 mW (28nm)
Memory Controller	13 mW	20 mW (40nm)
NoC Router	60 mW	23 mW (16nm)
PLL	241 mW	270 mW (28nm)
Total Clock Power	**1.98** W	**1.65** W

1) *RocketCore:* The capacitance of clock nets in each RocketCore module in 2-D design is 602 pF, whereas, in the 2.5-D design, it is 487 pF. The large capacitance contributed by long high-frequency nets and many buffers added on these nets cause the 2-D design to dissipate more clock power than the 2.5-D design.

2) *L2 Cache:* The 2.5-D design involves additional logic to support AIB protocol, which involves a significant amount of sequential circuits, causing an increase in the overall clock power. The capacitance of clock nets in each L2 cache module in 2-D design is 3.4 pF, whereas, in the 2.5-D design, it is 24 pF.

3) *MC:* The clock power of the 2.5-D four-channel MC is slightly higher than that of the 2-D design due to the presence of AIB logic. The capacitance of clock nets

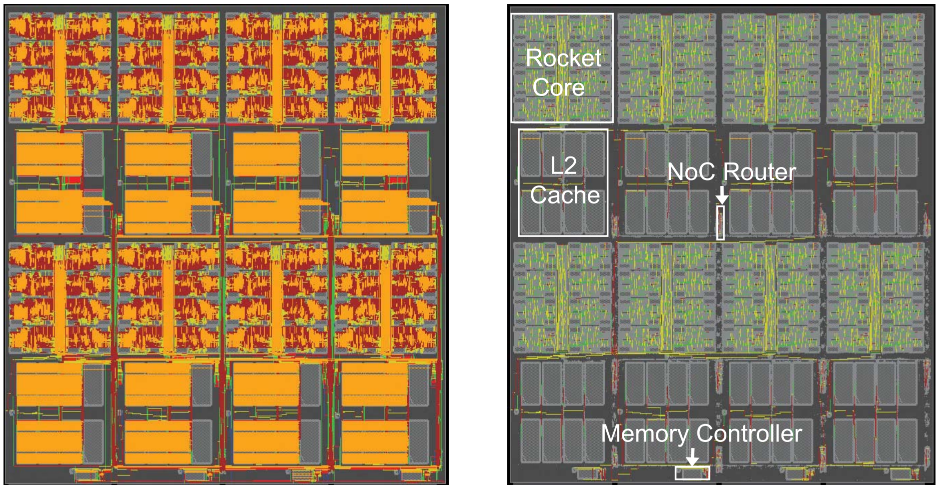

Fig. 23. Final layout and clock tree of 2-D monolithic SoC design of the modified Rocket-64 processor. The 2.5-D design is shown in Fig. 5.

in MC module in 2-D design is 21 pF, whereas, in the 2.5-D design, it is 31 pF.

4) *Router:* The 2-D design has 12 routers for arbitration, whereas the 2.5-D design has only eight routers, explaining the significant increase in clock power of the 2-D router design. The capacitance of clock nets in NoC module in 2-D design is 250 pF, whereas, in the 2.5-D design, it is 145 pF.
5) *PLL:* Both 2-D and 2.5-D designs have one PLL per each Rocket-8 and NoC modules. The additional power seen in the 2-D design is due to large capacitance contributed by long wires running from PLL to Rocket-8 and L2 cache modules. In the 2.5-D design, these long wires pass through the passive interposer layer, reducing the effective capacitance seen by the PLL.
6) *Overall:* The overall clock power of the 2.5-D design is 12% lower than that of the 2-D design. This power reduction is due to the long low-frequency interposer clock nets in the 2.5-D design instead of long high-frequency clock nets in the 2-D design. The individual chiplet clock power presented in Table VII does not include the power dissipated on interposer nets. The total clock power of all modules/chiplets in 2.5-D design accounts for 1.46 W. The interposer clock nets account for the additional 110 mW of clock power. The number of clock nets that pass through microbumps of each chiplet is around 3–4, a combination of 100-MHz and 1-GHz clocks, and, hence, the low interposer clock power. However, the overall power of the 2.5-D design is higher than the 2-D design, as shown in [2].

C. 2-D Versus Heterogeneous 2.5-D Rocket-64

Next, we compare our monolithic 2-D design with the heterogeneous 2.5-D design. We make the following observations.

1) *RocketCore:* The power of the 2-D Rocket-8 design is higher than the 2.5-D counterpart, as the 2-D Rocket-8 is at a higher technology node than the 2.5-D design. The capacitance of clock nets in each RocketCore module in the 2-D design is 527 pF, whereas, in the 2.5-D design, it is 357 pF.
2) *L2 Cache:* Similar to the homogeneous variant, the 2.5-D design involves additional logic to support AIB protocol, which involves a significant amount of sequential circuits, causing an increase in the overall clock power. The capacitance of clock nets in each L2 cache module in 2-D design is 3.4 pF, whereas, in the 2.5-D design, it is 28 pF.
3) *MC:* The clock power of the 2.5-D four-channel MC is slightly higher than that of the 2-D design due to the presence of AIB logic and higher technology node. The capacitance of clock nets in MC module in 2-D design is 29 pF, whereas, in the 2.5-D design, it is 42 pF.
4) *Router:* In addition to the lesser number of routers in the 2.5-D design, the routers are designed at the 16-nm node. This lowers the power consumption of 2.5-D routers further. The capacitance of clock nets in the NoC module in the 2-D design is 199 pF, whereas, in the 2.5-D design, it is 79 pF.
5) *PLL:* To reduce the number of high-frequency signals on the interposer, we placed all the PLLs within the L2 cache chiplets in the 2.5-D design. There are 24 PLLs (three in each L2 cache chiplet) in the 2.5-D design. The 2-D design has 20 PLLs (one PLL per partition), so the PLLs consume less power in the 2-D design.
6) *Overall:* The overall clock power of the 2.5-D design is 16.7% lower than that of the 2-D design. This further power reduction is due to the presence of lower technology node chiplets in the 2.5-D design. The individual chiplet clock power presented in Table VIII does not include the power dissipated on interposer nets. The total clock power of all modules/chiplets in the 2.5-D design accounts for 1.45 W. The interposer clock nets account for the additional 200 mW of clock power. The number of clock nets that pass through microbumps of each chiplet is around 3–4, which are a combination of 100-MHz, 600-MHz, 1-GHz, and 1.2-GHz clocks, and, hence, the low interposer clock power.

TABLE IX

HIERARCHICAL VERSUS FLAT INTERPOSER CLOCK ROUTING. THE LATENCY HERE DENOTES THE MAXIMUM DELAY FROM THE CLOCK C4 TO CLOCK MICROBUMP

Clock Metric	Chiplet	Hierarchical	Flat
Clock latency	L2 cache	71 ps	74 ps
	Rocket	50 ps	89 ps
	NoC	62 ps	48 ps
	Mem controller	66 ps	58 ps
Clock skew	L2 cache	0.25 ps	8 ps
	Rocket	0 ps	2 ps
Clock Jitter	-	1.2 ps	1.4 ps
Eye Height	-	439 mV	130 mV
Eye Width	-	415 ps	405 ps

To emphasize the scale of our benchmark, we present the runtime details in this paragraph. With the basic Rocket-64 RTL netlist readily available, it took us around a week to perform the RTL design of the interfaces required for 2.5-D designs and design the 28-nm PLL analog block. The entire RTL synthesis was done in around 36 h for the 2-D design and 48 h for the 2.5-D design. The physical design stage took around 60 h for the 2-D design and around 85 h for the 2.5-D design. The above results are based on the simulation of these designs.

Thus, we have demonstrated that clock delivery network optimization is manageable in 2.5-D designs and can even outperform 2-D counterparts, irrespective of homogeneous or heterogeneous chiplets, single-clock or multiclock domains, and PLL location. However, this requires rigorous co-optimization of chiplet and interposer portions.

D. Flat Versus Hierarchical Clocking Architecture

Table IX compares our heterogeneous hierarchical clock tree versus flat. We observe that the hierarchical clock tree performs better in almost all metrics except for NoC and MC latency. This is because the clock μ-bumps of NoC and MC chiplets are closer to the crystal clock C4 bump, and the clock frequency of these chiplets is lower in the flat clock network design than that with the hierarchical clock network.

VI. CONCLUSION

In this article, we have proposed a robust clock architecture for a many-core 2.5-D processor design. Our architecture relies on a hierarchical clock distribution network that utilizes a novel clock forwarding scheme and on-chip PLL for frequency conversion. Using this tool, we presented a 2-D versus 2.5-D clocking architecture comparison using GDS layouts of all chiplets and interposer and sign-off quality power, performance, and clock reliability metrics. Unlike the common belief that clock delivery is much more challenging in 2.5-D designs, we demonstrated that, with rigorous co-optimization of chiplet and interposer portions, clock delivery network optimization is manageable and can outperform the 2-D counterpart.

REFERENCES

[1] D. Stow, I. Akgun, R. Barnes, P. Gu, and Y. Xie, "Cost analysis and cost-driven IP reuse methodology for SoC design based on 2.5D/3D integration," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2016, pp. 1–6.

[2] J. Kim *et al.*, "Architecture, chip, and package co-design flow for 2.5 D IC design enabling heterogeneous IP reuse," in *Proc. ACM Design Autom. Conf.*, 2019, pp. 1–6.

[3] V. F. Pavlidis, I. Savidis, and E. G. Friedman, "Clock distribution networks in 3-D integrated systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 12, pp. 2256–2266, Dec. 2011.

[4] F.-W. Chen and T. Hwang, "Clock-tree synthesis with methodology of reuse in 3D-IC," *J. Emerg. Technol. Comput. Syst.*, vol. 10, no. 3, Apr. 2014, p. 22, Art. no. 22, doi: 10.1145/2567668.

[5] H. Xu, V. F. Pavlidis, and G. D. Micheli, "Effect of process variations in 3D global clock distribution networks," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 8, no. 3, pp. 1–25, Aug. 2012.

[6] T. Lu and A. Srivastava, "Gated low-power clock tree synthesis for 3D-ICs," in *Proc. Int. Symp. Low Power Electron. Design*, Aug. 2014, pp. 319–322.

[7] F.-W. Chen and T. Hwang, "Clock tree synthesis with methodology of re-use in 3D IC," in *Proc. 49th Annu. Design Autom. Conf. (DAC)*, 2012, pp. 1094–1099.

[8] S.-Y. Huang and C.-C. Zheng, "Die-to-die clock skew characterization and tuning for 2.5D ICs," in *Proc. IEEE 25th Asian Test Symp. (ATS)*, Nov. 2016, pp. 221–226.

[9] K. Asanovic, "The rocket chip generator," EECS Dept., Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2016-17, 2016. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-17.html

[10] *Double Data Rate (DDR3) DRAM Standard*, JEDEC, Arlington County, VA, USA, 2007.

[11] R. Chaware, K. Nagarajan, and S. Ramalingam, "Assembly and reliability challenges in 3D integration of 28 nm FPGA die on a large high density 65 nm passive interposer," in *Proc. IEEE 62nd Electron. Compon. Technol. Conf.*, May 2012, pp. 279–283.

[12] P. Ehrett *et al.*, "Analysis of microbump overheads for 2.5D disintegrated design," Univ. Michigan, Ann Arbor, MI, USA, Tech. Rep. CSE-TR-002-17, 2017.

[13] I. Akgun, J. Zhan, Y. Wang, and Y. Xie, "Scalable memory fabric for silicon interposer-based multi-core systems," in *Proc. IEEE 34th Int. Conf. Comput. Design (ICCD)*, Oct. 2016, pp. 33–40.

[14] Cadence. *Allegro Package Designer Plus SiP Layout Option*. Accessed: Oct. 10, 2020. [Online]. Available: https://www.cadence.com

[15] Cadence. *Innovus Implementation System*. Accessed: Oct. 10, 2020. [Online]. Available: https://www.cadence.com

[16] Y.-L. Hsueh *et al.*, "28.2 A 0.29 mm^2 frequency synthesizer in 40 nm CMOS with 0.19psrms jitter and <-100 dBc reference spur for 802.11ac," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 472–473.

[17] S. Choi, H. Kim, K. Kim, J. Park, D. H. Jung, and J. Kim, "Signal integrity analysis of silicon/glass/organic interposers for 2.5D/3D interconnects," in *Proc. IEEE 67th Electron. Compon. Technol. Conf. (ECTC)*, May 2017, pp. 2139–2144.

[18] A. E. Engin and S. R. Narasimhan, "Modeling of crosstalk in through silicon vias," *IEEE Trans. Electromagn. Compat.*, vol. 55, no. 1, pp. 149–158, Feb. 2013.

[19] H. M. Torun, M. Larbi, and M. Swaminathan, "A Bayesian framework for optimizing interconnects in high-speed channels," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Aug. 2018, pp. 1–4.

[20] Synopsys. *Synopsys Design Constraints*. Accessed: Oct. 10, 2020. [Online]. Available: https://www.synopsys.com

[21] D. Kehlet *et al. Accelerating Innovation Through a Standard Chiplet Interface: The Advanced Interface Bus (AIB)*. Accessed: Oct. 10, 2020. [Online]. Available: https://www.intel.com