Thermally-aware Multi-core Chiplet Stacking

Gaurav Kothari
State University of New York at Binghamton, USA
gkothar1@binghamton.edu

Kanad Ghose
State University of New York at Binghamton, USA
ghose@binghamton.edu

2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD) | 979-8-3503-2225-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCAD57390.2023.10323991

***Abstract*—Heterogeneous integration has enabled the interconnection of chiplets in 2.5D and 3D configurations within a package. Stacking a high-performance multi-core processor chiplet on top of another is challenging due to hot spot exacerbation in the stack. Temperature-induced DVFS throttling defeats any potential performance gain that results from the shorter vertical connections in-between the on-chip interconnection networks in each chiplet. We present and evaluate minimally-invasive floorplan transformation techniques that space out chiplet hot spots away from each other in the 3D stack using layout mirroring and offsetting. Chiplet redesign efforts are reduced, and cycle times are preserved. The resulting thermally-aware multi-core chiplet stacking techniques reduce the peak temperatures and temperature-induced performance throttling compared to naive chiplet stacking. Empty offset areas are then used to extend the on-chip cache capacity for further performance improvement. The multi-core stacking techniques are illustrated on a 14 nm Intel Skylake-SP-like (server) floorplan model using a cycle-level multi-core CPU performance simulator incorporating power and thermal modeling components.**

***Index Terms*—3D chiplet stacking, Heterogeneous integration**

I. Introduction

In recent years, at technology nodes beyond 14 nm, the cost per unit area of a functioning silicon microchip has increased exponentially due to lower yields, and high design and testing costs [2]. This makes it economically challenging to sustain the doubling of device counts with each technology node since Moore's Law implicitly assumes the same chip area to be maintained. Coupled with the commensurately reduced scaling of circuit block dimensions, these have led to the increasing use of heterogeneous integration (HI) techniques for connecting several smaller chips ("Chiplets") inside the package using low-latency, high bandwidth connections for realizing the equivalent of a larger, single chip, such as a multi-core processor or other complex systems within a common package. With 2.5D HI, chiplets inside the common package are located on an interposer (or wafer) and connected via highly parallel and short connections within the interposer/wafer. In 3D HI, chiplets are stacked vertically with the necessary and even shorter vertical connections. A combination of these two integration modes is also possible. CPU/GPU vendors, such as AMD, Intel, and NVIDIA, have heterogeneously integrated product offerings [31] [15] [16]. This paper focuses on heterogeneously-integrated 3D stacking of high-performance multi-core chiplets with identical cores and core counts and with identical chiplet dimensions. Such stacking reduces the physical reach of the vertical connections among the chiplets, reducing performance overhead that is otherwise introduced in a 2.5D configuration. The goal of such multi-core chiplet stacking is to realize the equivalent of a single die with twice as many cores as each chiplet, with a net (areal) footprint closer to that of each individual chiplet for higher yield and lower cost compared to an equivalent single chiplet realization. In doing this, design/validation cost and the time-to-market, redesign efforts must be minimized.

*Supported in part as Task No. 2878.007 through the SRC-CHIRP center.

A difficult challenge in stacking multi-core chiplets centers on heat dissipation. The heat generated in the chiplet further away from the heatsink is dissipated through the chiplet closer to the heatsink. Consequently, due to the high thermal resistance in the heat dissipation path, high-temperature induced clock throttling can occur and degrade performance. Intel's Lakefield processor [16] has mitigated this challenge by placing several low-power components such as USB, audio, Debug, and PCI blocks in the chiplet further away from the heatsink, while the chiplet containing high-performance components such as CPU cores and integrated GPU remains closer to the heatsink. Recent solutions from Intel [26] and AMD [1] have integrated multi-core chiplets in a 2.5D configuration. In fact, stacked high-performance multi-core processors with identical cores and core count have not appeared thus far.

This paper presents a detailed evaluation of techniques for stacking multi-core chiplets in terms of their thermal and performance implications and any necessary redesign and connectivity requirements. A methodology is introduced for stacking functionally identical multi-core chiplets containing high-performance cores that uses simple layout re-orientations within the chiplets without requiring design changes to the critical components of each chiplet and their relative positions. Critical paths are not affected and significant redesign/revalidation efforts (and associated expenditures) are thus avoided. The technique avoids placing high power dissipating functional blocks within each chiplet close to each other to reduce temperature-induced throttling. Additionally, vertical stacking reduces the number of hops in the interconnection network needed to maintain connectivity with the LLC (last-level cache) slices and speed up cache coherence activities, benefiting memory-intensive applications.

Specifically, this paper makes the following contributions:

1) Different multi-core chiplet stacking techniques for functionally identical multi-core chiplets are evaluated for their thermal and performance implications and any required redesign/revalidation effort.
2) A fine-grained core stacking technique that accounts for potential hot blocks within the functionally-identical cores to avoid vertically-adjacent hot spots is introduced, as existing solutions are not universally applicable to all multi-core chiplet layouts.

3) The overhead of the vertical connections that are needed for power and on-chip signal connections are analyzed.

II. Background

For realizing the vertical connections between two chiplets, hybrid face-to-face bonding has emerged as the leading solution for fine-pitched vertical connections needed for signals [23]. Small metal pads are created at the points where electrical connections are required between chiplets, using oxide masks to locate copper vias, whose exposed end makes up the pads. The chiplet surfaces, including the pads, are then planarized and polished. The chiplets to be connected are then aligned and subjected to an annealing process that bonds the copper pads, relying on metal diffusion activated by annealing. AMD's recent V-Cache uses hybrid face-to-face (F2F) bonding [31], as does Intel's Foveros technology [16]. At this time, hybrid face-to-face bonding techniques provide the highest connection density [22], with connection pitches of less than 10 µm compared to solder-joined microbumps that leave a void in-between dies that need to be filled with thermal interface materials (TIM), adding resistance in the heat conduction path. F2F bonding permits the realization of highly parallel connections that use clock forwarding, requiring no PLLs or DLLs for clock/data recovery at the receiving chiplet. We assume the use of F2F hybrid bonding in this work.

A single chiplet or a 3D stack of chiplets is housed in a package and mounted on a rigid PCB base with bump connections via a ball-grid array (BGA) to the socket. The PCB provides the required mechanical rigidity and also incorporates metal layers for routing signal, power, and clock connections to the BGA bumps. The chiplet stack typically seats on a Silicon interposer (SI) that provides connectivity to other chiplets in the package, as shown in Figure 1. The SI is placed on the substrate, with electrical connections made between the substrate and the interposer via smaller bumps ("C4 bumps") compared to the BGA bumps. The interposer has multiple metal layers (called ReDistribution Layers, RDLs) to re-route all connections to appropriate points at the base of the first chiplet, **C1**, in the stack via microbumps (and, in some cases, larger bumps for power). The second chiplet, **C2**, is mounted on C1 using face-to-face hybrid bonding. Power is supplied to C2 from C1 through F2F hybrid bonded pads, or, alternatively, C2 can protrude beyond C1, and power can be supplied directly to C2 from the interposer through power vias at the edge of C2, as in the Intel Foveros [16] prototype.

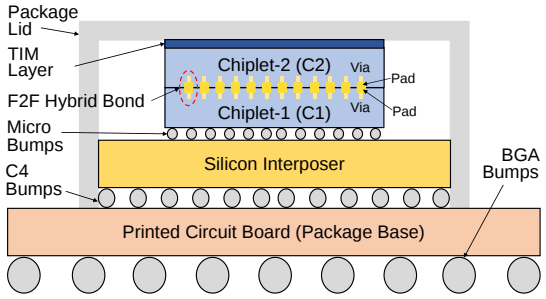


Fig. 1. Two-chiplet stack within a package with F2F hybrid bonding

Two types of 3D interconnections are required for stacking multi-core chiplets. The first are connections for power/ground and clock signals for the multi-core chiplet, the second type of connections are for network-on-chip (NoC) connectivity to drive data, address, and control signals to the shared last-level cache slices on the vertically adjacent multi-core chiplet. All of these can be realized using F2F hybrid-bonded pads, which are located adjacent to the router in each core tile. The number of connections required depends on the physical width of the links between routers. This requires the chiplet layout to be appropriately modified and thus increases the overall chiplet area in *any* multi-core stacking technique. The area impact is accounted for in our subsequent evaluations. Vertical connections for memory and IO are also needed in the sections of each chiplet that has IO and memory controllers and are not considered in this study.

III. Stacking Multi-Core Processor Chiplets

Multi-core chips with high core counts are typically implemented as a 2-D array of core tiles [6]. Core tiles are interconnected using a scalable network-on-chip (NoC) such as mesh or ring interconnect. Each core tile consists of **CORE** and **UNCORE** sections. The CORE section includes the processor's datapath and private L1 and L2 caches. The UNCORE section contains a local slice (bank) of the shared last-level cache (LLC) which is distributed across all cores, a local snoop filter to enforce data coherency, and an NoC router for communicating with other core tiles.

A large multi-core processor chip can be implemented as a 3D stack of smaller multi-core chips, each containing a small number of core tiles. For example, a 32-core planar processor can be implemented as a two-deep stack of chiplets with 16 cores each. The resulting benefits and drawbacks are:

- The die area (of each chiplet) is significantly reduced compared to a single chiplet implementation, enhancing the overall chiplet yield [34]. Further, this minimizes the interposer area and thus overall system cost.
- Cache coherence and LLC access delays are minimized due to reduced hops on the worst-case NoC path due to vertical connections between corresponding routers in each chiplet, leading to potential performance gains.
- Substantial micro-architectural changes are not needed. Redesign and validation costs are associated with adding interconnections and anything else needed to support stacking. Reducing the redesign and revalidation costs can lead to faster time-to-market and overall cost reductions resulting from higher yields for smaller chiplets.
- High-temperature induced (thermal) throttling due to stacking can reduce performance. This calls for stacking techniques that reduce such throttling.

Our evaluation of the stacking techniques is limited to two-deep stacks for high-end multi-core chiplets since thermal throttling, power routing needs, and integration costs quickly overshadow any benefits beyond a two-deep stack [34].

TABLE I
ARCHITECTURAL COMPONENTS MODELED BY MCPAT

Component	Block Descriptions
Fetch Unit (IFU)	Instruction Cache (I$), Instruction Decoder (ID), Branch Target Buffer (BTB), Branch Predictor (BP)
Load-Store Unit (LSU)	Load Queue (LQ), Store Queue (SQ), Data Cache (D$), Memory Management Unit (MMU)
FP Execution Unit (FP-EXE)	FP Register File (FPRF), FP ALU (FPU)
INT Execution Unit (INT-EX)	Issue Queue (IQ), Reorder Buffer (ROB), Rename Unit (RU), Integer Register File (IRF), Integer ALU (ALU), Complex ALU (CALU)
L2 Cache	Private L2 Cache (L2$)
Uncore	Snoop Filter (SF), NoC Router (RTR), L3/LLC Cache Slice (L3$)

A. Multi-core processor model used in this work

A processor model and floorplan based on an actual Intel Skylake-SP server [6] (at 14 nm) is used in this work to illustrate the three stacking techniques evaluated. An approximate simplified floorplan of core tiles mimicking the Intel Skylake-SP chip is constructed using area estimations derived from McPAT [24]. McPAT is an architectural-level power model that estimates the power, die area and timing of a multi-core processor. McPAT estimates the die area of all the significant architectural building blocks of a multi-core processor based on the architectural specifications. Table I lists all the blocks modeled by McPAT for this study. McPAT is configured to model Skylake-SP architecture as closely as possible using the publicly known specifications [6]. The resulting floorplan created using ArchFP floorplanning tool [14] with area inputs from McPAT, positions each block at approximately the same locations observed in real die photos [3].

This study includes only the micro-architectural blocks modeled by McPAT (for which we have the area and power models), listed in Table I. However, the actual CPU core still has many power-consuming components, which McPAT does not model [32]. Some additional ones not listed in [32] include voltage/power regulators (FIVR), clock delivery, trace cache, micro-op cache, micro-code ROM, and allocation queues (located between the fetch and decode stage). Since McPAT does not model these components, they are excluded from the study. Further, McPAT models a less complex Alpha-21264-based [21] branch predictor in contrast to more sophisticated and extensive branch predictors in Skylake-SP. Further, due to the limitation of the performance simulator used in this study which only supports SSE2 128-bit floating-point instructions, we assume the width of FPUs and floating-point registers as 128-bit compared to 512-bit FPUs in Skylake-SP. *Irrespective of these tool limitations, the stacking techniques presented and evaluated remain applicable to general classes of designs.*

Three multi-core stacking techniques are presented now.

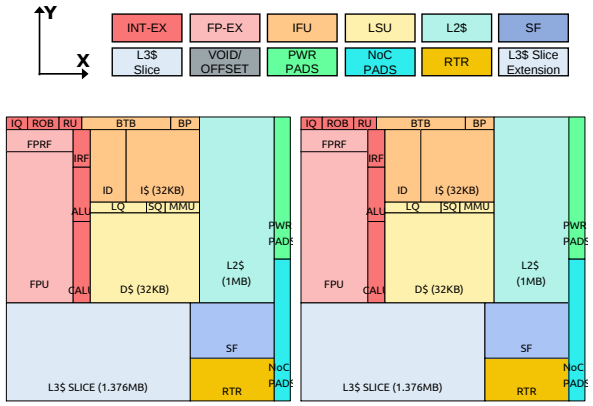
B. Naïve Stacking of Multi-core Chiplets

In naïve stacking, two identical multi-core chiplets are stacked on each other without any layout transformation other than adding room for vertical interconnections. Naïve stacking places corresponding blocks in each chiplet in vertically adjacent positions. Figure 2a shows the naïve stacking of two Skylake-SP core tiles. The main drawback of naïve stacking is the direct overlap of specific high (areal) power density blocks in each core tile within the two stacked chiplet. These blocks are IALU, IRF, IQ, ROB, and RU within the INT-EX unit, BTB and BP within the IFU unit, and LQ, SQ, and MMU within the LSU unit. These relatively small structures are heavily multi-ported and have high power dissipation as they are typically accessed in almost every cycle. Throughout this paper, we collectively refer to them as **hot blocks**. LQ, SQ, and MMU can become potential hot spots in memory-intensive (load-store) phases of the workload where D$ misses are negligible. When vertically adjacent cores run at higher frequencies with heavy utilization, thermal hot spots are exacerbated at these locations, especially within the chiplet further away from the heat sink. This scenario can arise relatively often. To reduce the high junction temperatures within the hot blocks, the clock frequency of the core has to be throttled down accompanied by voltage reduction via DVFS control. This thermally-induced throttling reduces power dissipation and lowers junction temperatures below the safe limit but results in performance degradation. This defeats any potential performance gains expected from 3D stacking.

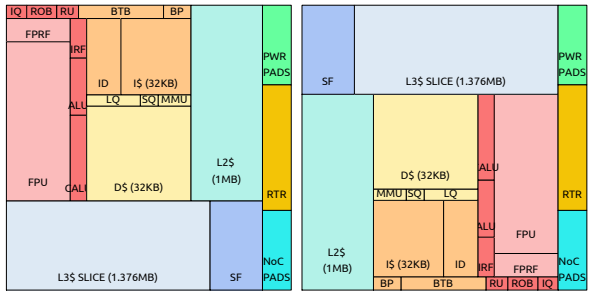
C. Methodologies for Thermally-aware Multi-core Chiplet Stacking (TMS)

To avoid the frequent DVFS throttling with naïve stacking of multi-core chiplets, minimally-invasive layout transformation techniques, which we call *Thermally-aware Multi-core Chiplet Stacking* (**TMS**) are possible. The phrase **minimally invasive** implies that the adjacency among critical logic blocks on the CPU's data path is preserved, which does not affect timings and avoids significant datapath redesign. In TMS, the floorplan of the vertically adjacent core tile in the stack is flipped (or mirrored) in the 2D plane, either along the X-axis, Y-axis, or along both axes, to eliminate the overlap of hot blocks. In some cases, flipping the floorplan cannot avoid the overlap, so precisely calculated empty offset areas are introduced along the edges of the core tiles to avoid frequent vertically-adjacent hot spots. We call this *offsetting*. Throughout this work, we assume that the area opened up by offsetting is just adequate for eliminating the overlap of hot blocks. Two variants of TMS and their stacking methodologies are as follows:

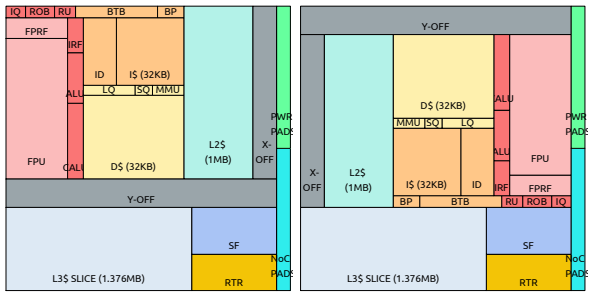
*1) Coarse-grained TMS (**TMS-CG**):* In TMS-CG, flipping and offsetting are applied to the vertically adjacent core tile as a whole, leaving the entire CORE and UNCORE layout intact in each layer. This variant is not new and has been examined in other work [11] [36] without using any knowledge of the likely hot core-internal blocks (See Section IV). Figure 2b illustrates the TMS-CG stacking configuration for the Skylake-SP core tile stack, where the layout of the entire vertically adjacent core tile (Top) is flipped along both the X-axis and Y-axis in the 2D plane. Flipping the vertically adjacent core tile (Top) along the X-axis and Y-axis eliminates the overlap of hot blocks of the two vertically adjacent core tiles without the need for offsetting. It positions the hot blocks in each core



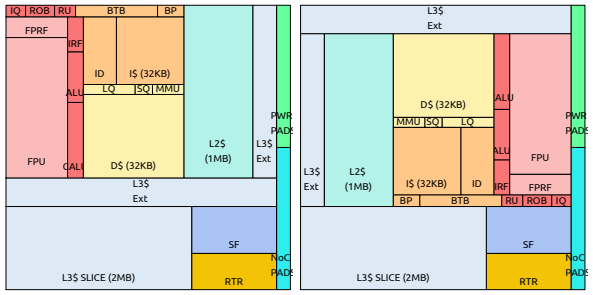
(a) 3D-NAIVE Stacking (Bottom core tile on LEFT, Top core tile on RIGHT stacked as is without any layout transformation)



(b) TMS-CG Stacking (Bottom core tile on LEFT, Top core tile on RIGHT flipped along both the X-axis and Y-axis, Router (RTR) is placed between power and NoC pad areas)



(c) TMS-FG+O Stacking (Bottom core tile on LEFT, Top core tile on RIGHT with its CORE section flipped and offsetted along both the X-axis and Y-axis)



(d) TMS-FG+L3 Stacking (Bottom core tile on LEFT, Top core tile on RIGHT with its CORE section flipped and offsetted along both the X-axis and Y-axis, offset areas used to increase L3$ slice capacity)

Fig. 2. NAIVE and TMS stacking configuration for Skylake-SP based core tile layout

tile diagonally away from each other in the vertical plane and partly covers them with a cooler UNCORE section (which contains the L3$ slice) of the vertically adjacent core tile. However, this configuration requires a modified position for the NoC router. The routers of each core tile are displaced from their original positions and placed between their respective power and NoC pad areas, resulting in two benefits. First, the distance (and the wire lengths) between the routers of the two vertically adjacent core tiles remains unchanged, which does not impact timings. Second, this preserves the adjacency of routers in each core tile with their corresponding UNCORE section, with which it frequently interacts. A higher redesign and re-validation effort is associated with the router repositioning in *both* of the chiplets requring wire rerouting, driver resizing, and other layout changes.

*2) Fine-grained TMS (**TMS-FG**):* Here flipping and offsetting are applied only to the CORE part of the vertically adjacent core tile based on the known location of hot blocks within each core. To the best of our knowledge, TMS-FG has not been proposed or explored earlier. As seen later, TMS-FG requires a lower redesign and revalidation effort than TMS-CG. TMS-FG itself has two sub-variants, as described next, Figure 2c shows the TMS subvariant configuration called **TMS-FG+O**, in which the CORE section layout of the vertically adjacent core tile (Top) is flipped and offsetted along both the X-axis (X-OFF) and Y-axis (Y-OFF) in the 2D plane. Balancing offsets are also added to the bottom tile on opposite ends to equalize the die area of both the core tiles. Flipping positions the hot blocks of each core tile diagonally away from each other in the Z-direction. The partial overlap seen after flipping is further removed by adding offsets. The area overhead of TMS-FG+O for Skylake-SP core tile is estimated to be 17.8% due to the addition of offset blocks. The offset areas can be left vacant or used to extend the L3$ slice capacity, leading to the other sub-variant **TMS-FG+L3**. Figure 2d shows the TMS-FG+L3 configuration, in which the vacant offset areas house an extra 512 KB L3$ slice bank, expanding the overall L3$ slice capacity per core from 1.375 MB to 2 MB. Note that TMS-FG configurations retain the original location of the UNCORE section eliminating the need to relocate the routers.

TMS, in general, can avoid frequent hot spot formation across the core tile stack, thereby reducing thermal throttling and allowing cores, even those on the chiplet farthest from the heat sink, to operate at higher frequencies for an extended duration, retaining all the benefits of 3D stacking, and resulting in performance improvement. TMS can also be coupled with existing dynamic thermal management techniques in software for further improvements. TMS does not require a significant redesign of the flipped core chiplet other than using a mirrored layout and adding offset areas. The silicon within the empty offset areas can be left unpopulated or used to extend the capacity of LLC/L3$.

D. Die Area Assessment

The first set of connections for our stacking scheme is between two vertically adjacent routers to maintain connectivity

TABLE II
AREA ESTIMATED BY MCPAT IN mm^2 FOR A SINGLE-CORE AND 16-CORE CPU FOR INTEL SKYLAKE-SP-LIKE LAYOUT AT 14 NM

Layout	Single Core Area	16 Cores CPU Area
2D	5.09293	81.48 (1 x 16-core chiplets)
3D-NAIVE, TMS-CG	5.39793 (+6%*)	43.18 (-47%*) (2 x 8-core chiplets)
TMS-FG+O, TMS-FG+L3	6.31793 (+24%*)	50.54 (-38%*) (2 x 8-core chiplets)

*% Increase in area relative to 2D layout

across the on-chip networks. We assume the width of the data link between two routers as 256 bits (+10 ECC bits) in each direction. In addition to data signals, 64 bits (+8 ECC bits) for address and 8 bits for control signals are assumed. Thus, 346 connections are required in each direction, resulting in 692 fine-pitched pads between two vertically adjacent routers. We conservatively assume pads [30] of $10\,\mu\mathrm{m}$ diameter and $20\,\mu\mathrm{m}$ pitch. With this specification, the total die area required per core tile for NoC pads is 0.1525 mm^2. The other set of connections needed for stacking is to supply power to the multi-core chiplet which is not adjacent to the interposer. The chiplet adjacent to the interposer receives power and ground connections through the interposer. The upper multi-core chiplet receives power and ground connections via fine-pitched pads through the power pad areas. For power, we conservatively assume 692 fine-pitch pads, identical to those used for the NoC; these translate to a die area of 0.1525 mm^2 per core tile. Hence, to place both NoC and power pads, the die area of each core tile increases by 0.305 mm^2. Table II shows the area estimated by McPAT for a single core and 16-core processor in 2D and all the 3D stacking configurations discussed in Section III. As shown in Table II, the area of a single core tile increases by 6% from the 2D to 3D-NAIVE/TMS-CG configurations. Similarly, the area of TMS-FG+O/TMS-FG+L3 increases by 24% compared to 2D, to accommodate NoC pads, power pads and offset areas.

Regardless of the increase in the core tile area from 2D to 3D configurations, the overall area occupied by multi-core processor significantly reduces as we transition from 2D to 3D configurations. As shown in Table II, the overall die area occupied by a 16-core count Skylake-SP processor in 2D layout estimated by McPAT is 81.48 mm^2, whereas 3D implementation by stacking two 8-core count multi-core chiplets reduces the overall die area by 47% for 3D-NAIVE/TMS-CG configurations and 38% for the TMS-FG+O/TMS-FG+L3 configurations. Smaller die sizes significantly improve the chiplet yield and reduce overall costs.

IV. RELATED WORK

Thermal herding as an approach to 3D out-of-order processor design [28] places hot blocks within the processors closer to the heatsink to reduce the thermal resistance in the heat dissipation path requiring functionally different chiplets. The 3D implementation reduces critical paths (as well as power density), enabling faster clocking and performance gains. Different flavors of thermal herding are possible but all require full-blown chiplet (re)design and validation. [28] . Hameed et al. [18] present a 3D processor implementation that essentially implements herding that exploits typical function block usage statistics, dynamically adapting CPU resource usage, and using DVFS, requiring substantial micro-architectural design changes.

Mathur et al. [25] perform a thermal analysis of 3D stacks of high-end processors that uses wafer-level F2F bonding and rely on the design-time partitioning of logic and memory components (caches). Their results indicate the obvious fact that placing the logic components over caches produces the least junction temperature increases. Energy budgets are used in [11] in accessing four major stacking configurations of processors with caches, including the use of mirroring (which is the same as the coarse-grained TMS (TMS-CG) of Section III). The other three techniques involve major redesign to place cores on top of one another and caches on top of one another or assume that core cache areas within a CPU chiplet are equal, an unrealistic assumption for many real CPUs.

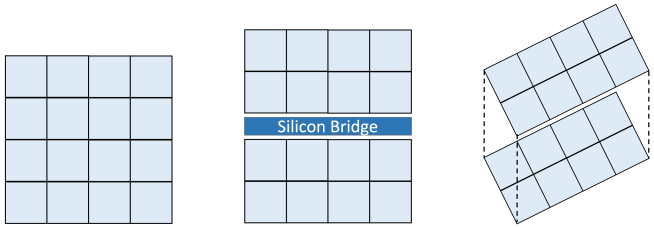
Zhang et al. [35] present a technique for sharing (pooling) caches across layers of multi-core chiplets, with DRAM layers at the bottom of the stack and rely on a job scheduler to exploit the larger effective cache capacity enabled by the short vertical connections. In principle, a thermally-aware job allocator can be added to improve the overall performance of TMS even further. Zhu et al. [36] propose using run-time monitoring and OS-controlled proactive techniques that exploit workload behavior to reduce junction temperatures in a monolithic 3D multi-core processor. The configuration studied assumes two planes of four cores, each with an L2 cache covering the four cores. This configuration, of course, requires a redesign of the physical layout used for a planar, 2D chiplet, rearrangements of critical connections, clock, and power redistribution. A rotation of the floorplan of one multi-core layer (after redesign) with respect to the other - a coarse-grained approach - is shown to reduce the junction temperatures further. Other work on stacking multi-core chiplets that require significant redesign [12] [13] [33] [20] includes various floorplanning algorithms that consider each functional block's power profile and generate an optimized floorplan in which the hot components are placed away from each other across the stack or placed on the layer closer to the heat sink. Monolithic 3D implementations of processors, which, in contrast to chiplet stacking, rely on a single monolithic silicon chip, have also been explored in [17]. Such implementations require significant redesign, process technology, and EDA innovations.

In contrast to all of these techniques, we focus on minimally-invasive layout adjustments that avoid significant IP redesign and without any impact on the critical path.

V. EXPERIMENTAL EVALUATION

A. Simulation Setup

All techniques are evaluated using HotSniper [27] toolchain, which tightly integrates the Sniper [9] multi-core performance simulator with HotSpot [19] (thermal model) and McPAT [24]



(a) 2D (b) 2D-BRIDGE (c) 3D

Fig. 3. Intel Skylake-SP-based 16-core processor configurations

(power model). Sniper was extended to model a 16-core Intel Skylake-SP-based server processor at 14 nm in 2D and 3D configurations. Two variants of 2D configuration are modeled. The first one is 2D, shown in Figure 3a, which consists of a single chiplet with 16 core tiles interconnected using a 4x4 mesh topology. The second one is 2D-BRIDGE, shown in Figure 3b, which consists of two 8-core chiplets, positioned side by side, connected using a Silicon bridge, with core tiles in each chiplet interconnected using a 4x2 mesh topology. We assume a latency of 5 clock cycles in each direction to cross the Silicon bridge. Figure 3c shows the 3D configuration in which two 8-core chiplets are stacked on each other. Based on the stacking methodologies discussed in section III, the 3D configuration is further categorized as 3D-NAIVE, TMS-CG, TMS-FG+O, and TMS-FG+L3. For HotSpot thermal model, we assume the thickness of all multi-core chiplets as $250\,\mu\text{m}$. A thermal interface material (TIM), heat spreader, and heat sink are modeled adjoining to the chiplet-1 (or C1). Table III lists all the parameters used for the HotSpot thermal model.

For performance simulation, we select Sniper's instruction-window centric (IW-centric, also known as ROB model) [10] out-of-order core model, which models a highly detailed super-scalar out-of-order SMT x86 core, with two hardware threads per physical core. For modeling Intel Skylake-SP, we configure the micro-architectural parameters of the IW-centric model as closely as possible to the publicly available specifications [6]. The default scheduling unit of the IW-centric model, based on Intel Nehalem [4] architecture, was extensively augmented to model Skylake-SP architecture. Further, we extended the NoC model to support a 3D mesh topology with ZXY routing. Table IV lists all the parameters used for performance simulations.

We extended Sniper to model temperature-aware per-core DVFS, which can adjust the clock frequencies of each core, from 1.0 GHz (F_{Min}) to 4.0 GHz (F_{Max}) in 500 MHz steps, with 2.5 GHz as the base frequency (F_{Base}). Uncore (L3$ and NoC) runs at a fixed frequency of 1.5 GHz. DVFS increments the frequencies of each core by one step at each sampling interval as long as the peak temperature of the core is below TJ_{Max}, or the maximum junction temperature. If the peak temperature of the core exceeds TJ_{Max}, the core switches to thermal throttle mode, and the frequency of the core is reduced to 1.0 GHz (F_{Min}) until the peak temperature of the core drops below T_{Safe}, after which, the thermal throttle mode is disabled, and the core frequency is again incremented in steps.

TABLE III
HOTSPOT THERMAL MODEL SPECIFICATIONS

Layer	**Thickness** (μm)	**Resistivity** ($\frac{mK}{W}$)	**Heat Capacity** ($\frac{J}{m^3K}$)
Multi-core Chiplet	250	0.01	1.75 X 10^6
TIM	20	0.25	4 X 10^6
Heat Spreader	1000	0.0025	3.55 X 10^6
Heat Sink	6900	0.0025	3.55 X 10^6

TABLE IV
PARAMETERS USED FOR SNIPER PERFORMANCE SIMULATION

Technology Node **Cores**	14 nm 2.5 GHz, 16 cores, 2-way SMT (32 threads)						
Issue / Commit **ROB / LQ / SQ** **ALU / MUL / FPU**	7 / 4 224 / 72 / 56 4 / 2 / 3						
ITLB / DTLB **L2-TLB**	128 entry, 8-way / 64 entry, 4-way 1536 entry, 12-way						
L1I$ / L1D$** **L2$	Private, 32KB, 8-way / Private, 32KB, 8-way Private Unified, 1024KB, 16-way						
L3$	Shared, 22MB, 16-way Shared, 32MB, 16-way (TMS-FG+L3)						
Uncore	L3$ + Mesh NoC, 1.5 GHz 2-cycles mesh hop latency 5-cycles bridge hop latency (2D-BRIDGE)						
DVFS Freq (GHz)	1	1.5	2	2.5	3	3.5	4
DVFS Voltage	0.66	0.7	0.75	0.8	0.85	0.9	0.95
Sampling Interval **Transition Time** **T_{Safe}/TJ_{Max}**	500 μs 10 μs 75 °C / 85 °C						

We assume T_{Safe} and TJ_{Max} as 75°C and 85°C, respectively. We choose a DVFS sampling interval of $500\,\mu\text{s}$ to measure temperatures and modify the frequencies, with $10\,\mu\text{s}$ transition time to switch to a new frequency-voltage level.

B. Workloads and Simulation Methodology

Representative results for 9 workloads are presented for brevity; each contains 32 threads assembled using stand-alone and a combination of multiple multi-threaded shared memory applications from SPEC-OMP 2012 [7], PARSEC-2.1 [8] (where concurrent threads are spawnned by a master thread), Rodinia [5], and XSBench [29] (Table V). Two workloads classes are used: compute-intensive, and memory-intensive. Compute-intensive workloads exhibit a high IPC with little to no uncore activity and their working sets fit into private L1/L2 caches. Memory-intensive workloads have relatively high LLC

TABLE V
WORKLOAD SUMMARY

Workload (Always consist of 32 threads)	**Category**
rodinia-heartwall	Compute
omp-botsalgn	
parsec-swaptions	
omp-smithwa	Memory
parsec-comb* (fluidanimate, x264, streamcluster, facesim)	
parsec-x264	
rodinia-comb* (cfd, srad, hotspot3D, kmeans)	
omp-comb* (mgrid331, smithwa, md, ilbdc)	
xsbench	

* Combinations use 8 threads for each of the 4 benchmarks

TABLE VI
RESULTS FOR COMPUTE-INTENSIVE WORKLOADS

Workload	Configuration	Peak Temp (°C)	Perf. Gain relative to 2D (%)	% Thermal Throttle
rodinia-heartwall	2D	73.4	-	-
	2D-BRIDGE	73.3	-0.01	-
	3D-NAIVE	88.6*	-6.61	4.4
	TMS-CG	79.5	0.68	-
	TMS-FG+O	77.5	0.67	-
	TMS-FG+L3	77.8	0.03	-
omp-botsalgn	2D	79.1	-	-
	2D-BRIDGE	79.3	-0.64	-
	3D-NAIVE	92*	-23.86	10.1
	TMS-CG	85.8*	-2.8	1.8
	TMS-FG+O	83.8	0.21	-
	TMS-FG+L3	84.3	0.08	-
parsec-swaptions	2D	74.7	-	-
	2D-BRIDGE	74.7	-0.07	-
	3D-NAIVE	91.4*	-18.97	10.6
	TMS-CG	82.6	0.11	-
	TMS-FG+O	82.6	0.39	-
	TMS-FG+L3	82.8	0.81	-

* Peak Temperature exceeds TJ_{Max} (85°C)

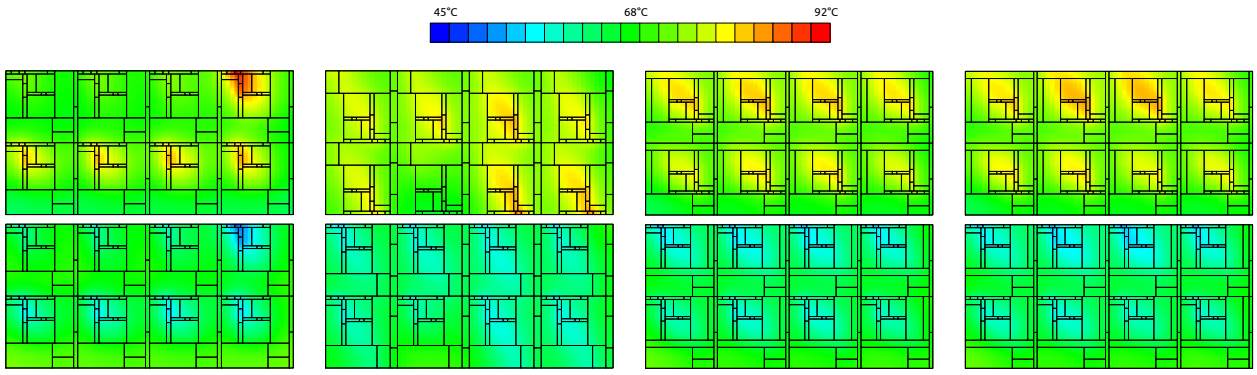
and DRAM accesses with cross-core data sharing that generate cache-coherence traffic. We fast-forward all the applications to the start of region-of-interest (ROI) and warm up the simulator caches for following 500 million instructions. We then switch to detailed cycle-level simulation mode for the interval comprising of the next 5 billion committed instructions, aggregated over all threads. For each workload-configuration pair, we record the peak temperature, the percentage of total execution time spent in the thermally-throttled mode, and performance gains within the interval. The performance gains are calculated as a percentage reduction in workload execution time relative to the baseline 2D design.

C. Results

Table VI outlines the evaluation results of all the compute-intensive workloads, showing the peak temperature, performance gain against 2D configuration, and percentage of total execution time spent in thermal throttle mode. For all the workloads listed in Table VI, the peak temperature in the case of 3D-NAIVE consistently exceeds TJ_{Max}, causing thermal throttling. Highly compute-intensive workloads such as *omp-botsalgn* and *parsec-swaptions*, which exhibit a per thread IPC in the range of 3-5 in the peak intervals, reported a significant amount of thermal throttling in 3D-NAIVE configuration. For *omp-botsalgn*, 10.1% of the execution time was spent in the thermal throttle mode, resulting in a 23.8% performance drop, whereas *parsec-swaptions*, spent 10.6% of the execution time in the thermal throttle mode, causing an 18.9% drop in performance. Due to the direct overlap of hot blocks of vertically adjacent core tiles in 3D-NAIVE configuration, thermal hotspots are formed specifically on the core tiles belonging to the multi-core chiplet which is not adjacent to the heat sink. This is demonstrated in Figure 4a, which shows the temperature distribution of both the stacked multi-core chiplets in 3D-NAIVE configuration for *omp-botsalgn*, for the hottest interval which reported a peak temperature of 92°C. In 3D-NAIVE, such thermal hotspots are expected to form often causing thermal throttling. When the core operates in the thermal throttle mode, it runs at the lowest frequency setting (F_{Min}) until the core's peak temperature falls below T_{Safe}. Prompt recovery from thermal throttle mode is challenging as peak temperatures across successive DVFS sampling intervals are still relatively higher than T_{Safe}, especially during compute-intensive phases. Due to this, the core remains in the lowest frequency setting for an extended duration until the temperature drops below T_{Safe}, leading to a substantial performance loss. This situation is illustrated in Figure 5a, which shows the peak temperature and frequency variation for the entire simulation duration, for the hottest core in 3D-NAIVE configuration for *omp-botsalgn*.

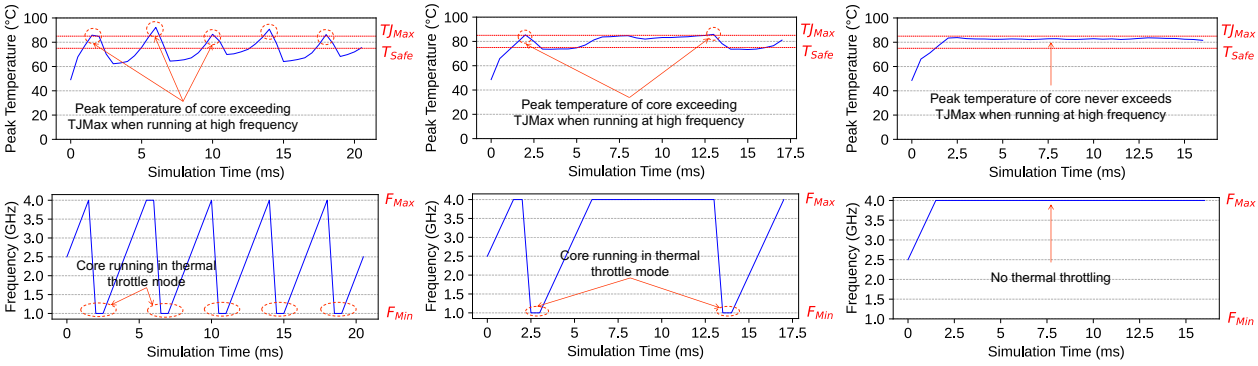
As shown in Table VI, all TMS configurations reported noticeably lower peak temperatures compared to their respective 3D-NAÏVE configuration. Excluding TMS-CG in *omp-botsalgn*, the peak temperatures for all other TMS configurations never exceeded TJ_{Max}, eliminating thermal throttling. TMS-CG has entire vertically adjacent core tile flipped along both the X-axis and Y-axis. In TMS-FG+O/TMS-FG+L3, the CORE section of the vertically adjacent core tile is flipped and offsetted along both the X-axis and Y-axis. Hence, in all TMS configurations, hot blocks belonging to each core tile are placed away from each other horizontally and vertically, in the core tile stack. This avoids the frequent formation of thermal hotspots during compute-intensive phases and prevents temperatures from surpassing TJ_{Max}, especially on the multi-core chiplet which is not adjacent to the heat sink. This can be noticed in the temperature distribution maps for all three TMS configurations in Figure 4b, Figure 4c, and Figure 4d. TMS-CG for *omp-botsalgn* reported a slight thermal throttling because of thermal hotspot formation due to the close proximity of ALU and MMU of vertically adjacent cores. In spite of this, as seen in Figure 5b, peak temperature for the hottest core surpassed TJ_{Max} merely twice during the entire simulated duration, allowing the core to operate at the highest frequency for prolonged intervals compared to 3D-NAIVE. In TMS-FG+O/TMS-FG+L3, the offsets used to remove the overlap of hot blocks also ensure they are adequately away from each other to prevent the formation of the thermal hotspot, hence it never undergoes thermal throttling for the entire duration, as shown in Figure 5c. Since compute-intensive workloads don't have significant NoC activity, notable performance improvements are not seen for 3D/TMS configurations against 2D.

Figure 6 shows the performance gain relative to 2D configuration for memory-intensive workloads. These workloads didn't report thermal throttling for any 2D and 3D configurations, therefore due to space constraints peak temperatures are not shown. Noticeable performance improvement is seen for all the memory-intensive workloads as we migrate from 2D to 3D configuration, ranging from 11% (*xsbench*) to 32% (*parsec-comb*), with 17% on average across all workloads. This speedup can be attributed to reduced coherence and



(a) 3D-NAIVE (b) TMS-CG (c) TMS-FG+O (d) TMS-FG+L3

Fig. 4. Temperature distribution for the hottest interval in each 3D configuration for 16 core Skylake-SP-based processor model for compute-intensive *omp-botsalgn* workload. Figure on Top shows Chiplet-2 (away from the heat sink), Figure on the Bottom shows Chiplet-1 (adjacent to the heat sink)



(a) 3D-NAIVE (b) TMS-CG (c) TMS-FG+O / TMS-FG+L3

Fig. 5. Peak temperature and frequency variation for the entire simulation duration, for the hottest core in compute-intensive *omp-botsalgn* workload. TMS-FG+O and TMS-FG+L3 report similar behavior for *omp-botsalgn* hence only the TMS-FG+O plot is shown due to space constraints

LLC access delays due to the presence of 3D mesh interconnect resulting in reduced mesh hops on the worst-case NoC path. Additional performance improvement is seen for DRAM bound workloads such as *rodinia-comb* (16%), *omp-comb* (20%) and *xsbench* (15%) in TMS-FG+L3 configuration, whose empty areas vacated in the offsetting step are used to extend the L3$ capacity. Noticeable performance drop is reported for 2D-BRIDGE configuration for all workloads, ranging from 3% (*omp-smithwa*) to 21% (*parsec-comb*), with 8% on average. This slowdown is due to the additional delay required to communicate across the Silicon bridge.

VI. Conclusion

We consider the heterogeneous integration of functionally identical N-core chiplets in a 3D stack to realize the equivalent of a 2N core chiplet. The goal is to do this without incurring significant redesign and revalidation. Simultaneously, DVFS throttling inherent to stacking multi-core chiplets is reduced using relatively simple layout changes. Two techniques that contrast with existing techniques in using the knowledge of potential hot blocks within the cores are presented and evaluated for Intel Skylake-SP (server) based chiplet at 14 nm. Clock throttling is reduced against a naïve stacking technique in both solutions. Implications of signal and power connections

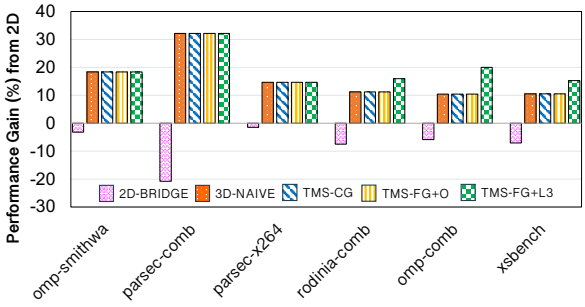


Fig. 6. Performance gain (% reduction in workload execution time) reported for memory-intensive workloads against baseline 2D configuration

within the stack are also assessed. Of the two thermally-aware multi-core stacking techniques, TMS-FG avoids router repositioning in both chiplets. Our results also demonstrate the advantages of lowering the hop count of the interconnection/coherence network with stacking, on memory-intensive workloads. When empty areas vacated in the offsetting step are used to extend the L3$ capacity, as in TMS-FG+L3, memory-intensive applications actually show a noticeable speedup. This speedup will improve with the increase in total core count.

REFERENCES

[1] "AMD EPYC 7003 Series Processors." [Online]. Available: https://www.amd.com/en/processors/epyc-7003-series

[2] "Chapter 2: High performance computing and data centers," in *Heterogeneous Integration Roadmap 2021 Edition*. IEEE EPS. [Online]. Available: https://eps.ieee.org/images/files/HIR_2021/ch02_hpc.pdf

[3] "Chip Annotation Viewer," https://misdake.github.io/ChipAnnotationViewer/?map=Skylake_Core&commentId=633022524, [Accessed 29-Jul-2022].

[4] "Nehalem - microarchitectures - intel." [Online]. Available: https://en.wikichip.org/wiki/intel/microarchitectures/nehalem_(client)

[5] "Rodinia benchmark suite — cs.virginia.edu," https://www.cs.virginia.edu/rodinia/doku.php, [Accessed 15-Jul-2022].

[6] "Skylake (server) - microarchitectures - intel." [Online]. Available: https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(server)

[7] "SPEC OMP2012 — spec.org," https://www.spec.org/omp2012/, [Accessed 15-Jul-2022].

[8] C. Bienia and K. Li, "Parsec 2.0: A new benchmark suite for chip-multiprocessors," in *Proceedings of the 5th Annual Workshop on Modeling, Benchmarking and Simulation*, June 2009.

[9] T. E. Carlson, W. Heirman, S. Eyerman, I. Hur, and L. Eeckhout, "An evaluation of high-level mechanistic core models," *ACM Transactions on Architecture and Code Optimization (TACO)*, 2014.

[10] ——, "An evaluation of high-level mechanistic core models," *ACM Transactions on Architecture and Code Optimization (TACO)*, 2014.

[11] J. M. Cebrián, J. L. Aragón, and S. Kaxiras, "Token3d: Reducing temperature in 3d die-stacked cmps through cycle-level power control mechanisms," in *Euro-Par 2011 Parallel Processing*, E. Jeannot, R. Namyst, and J. Roman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 295–309.

[12] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3d ics," in *IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004.* IEEE, 2004, pp. 306–313.

[13] D. Cuesta, J. Ayala, J. Hidalgo, M. Poncino, A. Acquaviva, and E. Macii, "Thermal-aware floorplanning exploration for 3d multi-core architectures," in *Proceedings of the 20th symposium on Great lakes symposium on VLSI*, 2010, pp. 99–102.

[14] G. G. Faust, R. Zhang, K. Skadron, M. R. Stan, and B. H. Meyer, "Archfp: Rapid prototyping of pre-rtl floorplans," in *2012 IEEE/IFIP 20th International Conference on VLSI and System-on-Chip (VLSI-SoC)*. IEEE, 2012, pp. 183–188.

[15] D. Foley and J. Danskin, "Ultra-performance pascal gpu and nvlink interconnect," *IEEE Micro*, vol. 37, no. 2, pp. 7–17, 2017.

[16] W. Gomes, S. Khushu, D. B. Ingerly, P. N. Stover, N. I. Chowdhury, F. O'Mahony, A. Balankutty, N. Dolev, M. G. Dixon, L. Jiang *et al.*, "8.1 lakefield and mobility compute: A 3d stacked 10nm and 22ffl hybrid processor system in 12× 12mm 2, 1mm package-on-package," in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2020, pp. 144–146.

[17] B. Gopireddy and J. Torrellas, "Designing vertical processors in monolithic 3d," in *Proceedings of the 46th International Symposium on Computer Architecture, ISCA 2019, Phoenix, AZ, USA, June 22-26, 2019*, S. B. Manne, H. C. Hunter, and E. R. Altman, Eds. ACM, 2019, pp. 643–656. [Online]. Available: https://doi.org/10.1145/3307650.3322233

[18] F. Hameed, M. A. Al Faruque, and J. Henkel, "Dynamic thermal management in 3d multi-core architecture through run-time adaptation," in *2011 Design, Automation & Test in Europe*. IEEE, 2011, pp. 1–6.

[19] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "Hotspot: A compact thermal modeling methodology for early-stage vlsi design," *IEEE Transactions on very large scale integration (VLSI) systems*, vol. 14, no. 5, pp. 501–513, 2006.

[20] Y. Huang, Q. Zhou, Y. Cai, and H. Yan, "A thermal-driven force-directed floorplanning algorithm for 3d ics," in *2009 11th IEEE International Conference on Computer-Aided Design and Computer Graphics*. IEEE, 2009, pp. 497–502.

[21] A. Jain, W. Anderson, T. Benninghoff, D. Berucci, M. Braganza, J. Burnetie, T. Chang, J. Eble, R. Faber, O. Gowda, J. Grodstein, G. Hess, J. Kowaleski, A. Kumar, B. Miller, R. Mueller, P. Paul, J. Pickholtz, S. Russell, M. Shen, T. Truex, A. Vardharajan, D. Xanthopoulos, and T. Zou, "A 1.2 ghz alpha microprocessor with 44.8 gb/s chip pin bandwidth," in *2001 IEEE International Solid-State Circuits Conference. Digest of Technical Papers. ISSCC (Cat. No.01CH37177)*, 2001, pp. 240–241.

[22] S.-W. Kim, M. Detalle, L. Peng, P. Nolmans, N. Heylen, D. Velenis, A. Miller, G. Beyer, and E. Beyne, "Ultra-fine pitch 3d integration using face-to-face hybrid wafer bonding combined with a via-middle through-silicon-via process," 05 2016, pp. 1179–1185.

[23] M. LaPedus, "Bumps Vs. Hybrid Bonding For Advanced Packaging — semiengineering.com," https://semiengineering.com/bumps-vs-hybrid-bonding-for-advanced-packaging/, [Accessed 22-May-2023].

[24] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "The mcpat framework for multicore and manycore architectures: Simultaneously modeling power, area, and timing," *ACM Trans. Archit. Code Optim.*, vol. 10, no. 1, apr 2013. [Online]. Available: https://doi.org/10.1145/2445572.2445577

[25] R. Mathur, C.-J. Chao, R. Liu, N. Tadepalli, P. Chandupatla, S. Hung, X. Xu, S. Sinha, and J. Kulkarni, "Thermal analysis of a 3d stacked high-performance commercial microprocessor using face-to-face wafer bonding technology," in *2020 IEEE 70th Electronic Components and Technology Conference (ECTC)*, 2020, pp. 541–547.

[26] N. Nassif, A. O. Munch, C. L. Molnar, G. Pasdast, S. V. Lyer, Z. Yang, O. Mendoza, M. Huddart, S. Venkataraman, S. Kandula, R. Marom, A. M. Kern, B. Bowhill, D. R. Mulvihill, S. Nimmagadda, V. Kalidindi, J. Krause, M. M. Haq, R. Sharma, and K. Duda, "Sapphire rapids: The next-generation intel xeon scalable processor," in *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, vol. 65, 2022, pp. 44–46.

[27] A. Pathania and J. Henkel, "Hotsniper: Sniper-based toolchain for many-core thermal simulations in open systems," *IEEE Embedded Systems Letters*, vol. 11, no. 2, pp. 54–57, 2018.

[28] K. Puttaswamy and G. H. Loh, "Thermal herding: Microarchitecture techniques for controlling hotspots in high-performance 3d-integrated processors," in *2007 IEEE 13th International Symposium on High Performance Computer Architecture*, 2007, pp. 193–204.

[29] J. R. Tramm, A. R. Siegel, T. Islam, and M. Schulz, "XSBench - the development and verification of a performance abstraction for Monte Carlo reactor analysis," in *PHYSOR 2014 - The Role of Reactor Physics toward a Sustainable Future*, Kyoto, 2014. [Online]. Available: https://www.mcs.anl.gov/papers/P5064-0114.pdf

[30] P. Vivet, E. Guthmuller, Y. Thonnart, G. Pillonnet, C. Fuguet, I. Miro-Panades, G. Moritz, J. Durupt, C. Bernard, D. Varreau *et al.*, "Intact: A 96-core processor with six chiplets 3d-stacked on an active interposer with distributed interconnects and integrated power management," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 1, pp. 79–97, 2020.

[31] J. Wuu, R. Agarwal, M. Ciraula, C. Dietz, B. Johnson, D. Johnson, R. Schreiber, R. Swaminathan, W. Walker, and S. Naffziger, "3d v-cache: the implementation of a hybrid-bonded 64mb stacked cache for a 7nm x86-64 cpu," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65. IEEE, 2022, pp. 428–429.

[32] S. L. Xi, H. Jacobson, P. Bose, G.-Y. Wei, and D. Brooks, "Quantifying sources of error in mcpat and potential impacts on architectural studies," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, 2015, pp. 577–589.

[33] L. Xiao, S. Sinha, J. Xu, and E. F. Young, "Fixed-outline thermal-aware 3d floorplanning," in *2010 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2010, pp. 561–567.

[34] Y. Xie and J. Zhao, "Die-stacking architecture," *Synthesis Lectures on Computer Architecture*, vol. 10, pp. 1–127, 06 2015.

[35] T. Zhang, J. Meng, and A. K. Coskun, "Dynamic cache pooling in 3d multicore processors," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 12, no. 2, pp. 1–21, 2015.

[36] C. Zhu, Z. Gu, L. Shang, R. P. Dick, and R. Joseph, "Three-dimensional chip-multiprocessor run-time thermal management," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 8, pp. 1479–1492, 2008.