Hier-3D: A Hierarchical Physical Design Methodology for Face-to-Face-Bonded 3D ICs

Anthony Agnesina[*], Moritz Brunion[†], Alberto García-Ortiz[†], Francky Catthoor[‡], Dragomir Milojevic[‡], Manu Komalan[‡], Matheus Cavalcante[§], Samuel Riedel[§], Luca Benini[§¶], Sung Kyu Lim[*]
[*]Georgia Institute of Technology, Atlanta, USA [†]Universität Bremen, Bremen, Germany,
[‡]IMEC, Leuven, Belgium [§]ETH Zürich, Zürich, Switzerland [¶]Università di Bologna, Bologna, Italy
agnesina@gatech.edu

ABSTRACT

Hierarchical very-large-scale integration (VLSI) flows are an understudied yet critical approach to achieving design closure at giga-scale complexity and gigahertz frequency targets. This paper proposes a novel hierarchical physical design flow enabling the building of high-density and commercial-quality two-tier face-to-face-bonded hierarchical 3D ICs. We significantly reduce the associated manufacturing cost compared to existing 3D implementation flows and, for the first time, achieve cost competitiveness against the 2D reference in large modern designs. Experimental results on complex industrial and open manycore processors demonstrate in two advanced nodes that the proposed flow provides major power, performance, and area/cost (PPAC) improvements of 1.2-2.2× compared with 2D, where all metrics are improved simultaneously, including up to 20 % power savings.

CCS CONCEPTS

• **Hardware → 3D integrated circuits**; **Physical design (EDA)**.

KEYWORDS

Hier-3D; Physical Design Methodology; Wafer-level Bonding; Face-to-Face (F2F) Bonded 3D ICs

ACM Reference Format:
Anthony Agnesina[*], Moritz Brunion[†], Alberto García-Ortiz[†], Francky Catthoor[‡], Dragomir Milojevic[‡], Manu Komalan[‡], Matheus Cavalcante[§], Samuel Riedel[§], Luca Benini[§¶], Sung Kyu Lim[*]. 2022. Hier-3D: A Hierarchical Physical Design Methodology for Face-to-Face-Bonded 3D ICs. In *ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED '22), August 1–3, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3531437.3539702

1 INTRODUCTION

To deliver the perceived benefits of 3D ICs outside the purview of research and academia [6], a hierarchical 3D design flow must subdivide complex, manycore, large-memory giga-scale designs into sub-blocks, which are independently synthesized and physically placed-and-routed (P&R) as separate design units. Then, the resultant mapped sub-blocks are recombined into subsequent runs of higher-level blocks—a process repeated as the hierarchy is traversed up to the top level.

This hierarchical approach offers the following benefits: (1) Large designs can be implemented with acceptable runtime and memory usage, where typically significant reuse is made of (nearly) identical sub-blocks. It is impossible with today's machines and tools to implement industrial-size SoC designs flat, (2) Concurrent implementation of design tasks can be split across multiple development teams, and (3) Third-party intellectual property (IP) blocks can be elegantly integrated as a natural part of the process flow. While a hierarchical flow mitigates many issues of a flat approach, numerous tedious and error-prone tasks are still required to close timing and optimize PPA at the top-level netlist. These include budgeting for block-level timing constraints and setting appropriate hierarchical physical constraints.

EDA vendors' tools partially manage these remaining issues in their proposed hierarchical flows that can simplify the implementation of large 2D designs. However, none of these flows are currently optimized for 3D hierarchical implementations. Instead, academic work [5] focuses on sequential die-by-die approaches, where hierarchy levels are artificially created to use standard block-level flows where blocks are placed on different tiers and routed in 3D. Furthermore, in these hierarchical flows, the blockage of macros makes placement and routing much harder on higher hierarchy levels than in flat implementations.

In this work, we propose *Hier-3D*, a physical design methodology for 3D bottom-up hierarchical implementations to co-optimize power, performance, and area/cost combined design metrics, as modern-day 3D flows do not satisfactorily address the latter. The key contributions are as follows.

- We propose a first-of-its-kind hierarchical physical design flow for 3D designs, which significantly improves the placement and routing utilization across the 3D stack by reusing the unassigned silicon area of preceding hierarchy levels.
- Our Hier-3D flow exploits the inherent logical hierarchy to enable hierarchical multi-tier standard cell placement, greatly expanding the design space of 3D ICs.
- We demonstrate 1.2-2.2× PPAC improvements and 1.2-1.5× runtime speedup on three highly diverse open-source and industrial low-power benchmarks and, for the first time, cost improvement compared to the 2D reference. These results are incommensurate with 2D and standard commercial and academic 3D flows.

2 RELATED WORK

In the following, we present two existing state-of-the-art approaches to implementing face-to-face (F2F) wafer-to-wafer (W2W) bonded 3D ICs: a die-by-die-like Sequential-2D [8] and a Macro-3D flow [2]. We believe other available 3D flows unmatch the 3D awareness enabled by Macro-3D, making it most advanced towards a future, native 3D flow. The Sequential-2D is a realization of the well-known die-by-die integration scheme, which already has extensive tool support in the EDA industry. Both flows can be used for 3D hierarchical designs in a traditional black box, bottom-up approach and will serve as baselines for evaluating our proposed Hier-3D flow.

This work is licensed under a Creative Commons Attribution International 4.0 License.

ISLPED '22, August 1–3, 2022, Boston, MA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9354-6/22/08.
https://doi.org/10.1145/3531437.3539702

2.1 Sequential-2D Flow

The Sequential-2D flow follows the EDA industry's standard approach to implementing 3D designs.

Key Idea. It enables 3D integration through the separate implementations of a die stack's top and bottom die with a standard 2D flow. 3D physical awareness is established by defining pins inside the core area at valid copper pad locations.

Strengths. This approach does not restrict the partitioning scheme, as logic and memory cells can be placed in both dies. Moreover, it reuses the standard commercial 2D tools capabilities, modeling the F2F bonding pads as IO pin shapes.

Limitations. Partitioning a design into a top and bottom die inherently introduces an additional level into the implementation hierarchy. If it does not follow the natural partitioning provided by the logical hierarchy, it requires a challenging additional constraint modeling step that can introduce PPA degradation. Moreover, sharing both dies' back-end-of-line (BEOL) resources would necessitate the insertion of feed-throughs in the netlist of each die for each shared intra-die net, an approach highly inflexible and therefore impractical without additional automation efforts.

Hierarchical Design. The Sequential-2D flow can be applied to a hierarchical design by introducing an additional hierarchy level into each block and forming a top and bottom sub-blocks. Implementations of subsequent hierarchy levels must respect the separated child block implementations in both dies.

2.2 Macro-3D Flow

Macro-3D provides state-of-the-art PPA optimization capabilities for 3D ICs in the memory-on-logic partitioning scheme.

Key Idea. The commercial 2D P&R tool is made aware of the complete die stack. The pins and routing obstructions of the memory macros are projected to the corresponding top layer to yield a holistic memory-on-logic flow. The copper pads are modeled as regular vias, allowing their automated insertion and inherently incorporating the impact of their parasitics on timing and power.

Strengths. Standard P&R engines can optimize the complete design for timing closure because of the complete design view across both dies. Further, the holistic stack view enables a unified routing step of both dies, allowing metal layer sharing, i.e., nets with a start- and endpoint in the same die can borrow metal resources from the other die, resulting in a more uniform metal layer utilization.

Limitations. The silicon area of the memory and the logic die are usually very different. Therefore, as W2W-based 3D integration requires matching die sizes, the Macro-3D flow increases the resulting manufacturing cost relative to the 2D die if the lost space cannot be reclaimed.

Hierarchical Design. Macro-3D can implement designs hierarchically by abstracting sub-blocks as full-block obstructions. However, by obstructing all metal layers between the front-end-of-lines (FEOLs) of both dies, routing through the abstraction is prohibited, and routing over it is impossible.

3 HIER-3D DESIGN METHODOLOGY

We propose our Hier-3D flow to mitigate the issues presented above. Targeted explicitly for silicon area minimization, it allows the building of high-density hierarchical commercial-quality F2F-bonded 3D ICs in a bottom-up fashion. It combines the previously presented advantages of the Sequential-2D and Macro-3D flows and introduces new key features to address their shortcomings, as summarized in Table 1.

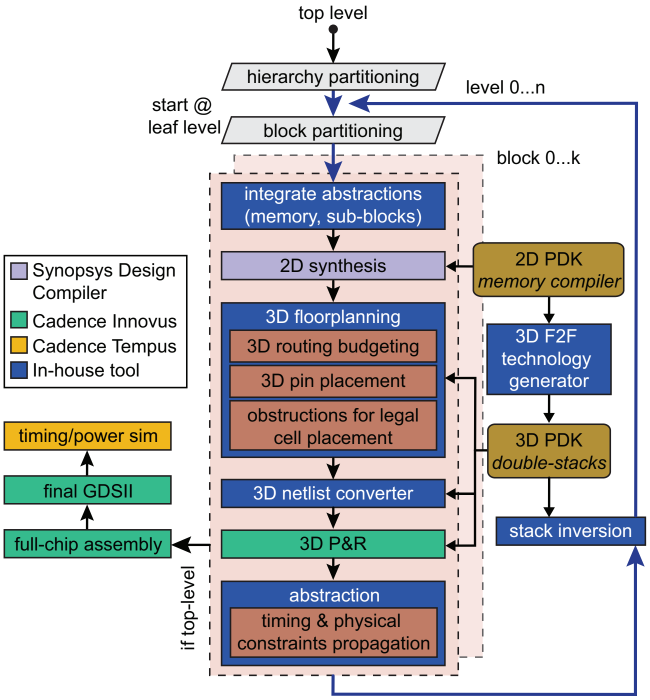

Figure 1: The key steps in our Hier-3D flow. The hierarchy depth defines the number of outer cycles/iterations. Per iteration, synthesized block-level designs are prepared with 3D floorplanning, 3D placed-and-routed, and abstracted through our physical and timing constraints propagation. Finally, the stack and abstractions can be inverted to enable standard cell placement on the opposite die at the upper hierarchy level.

The high integration density targeted by Hier-3D has vast implications on the PPAC characteristics. Indeed, dense block packing can increase the number of dies per wafer for cost, eliminate long timing-critical wires and reduce interconnecting energy by reducing distances. In addition, maintaining a holistic view across the die stack avoids overconstraining the block interfaces and enables efficient power optimization capabilities.

3.1 Flow Overview

Our overall 3D hierarchical flow is represented in Figure 1. The flow starts with an RTL whose hierarchy is predefined or created. Then, we synthesize each block within a given hierarchy level using the timing abstractions of the lower-level sub-blocks. Next, each block undergoes a 3D floorplanning step that places pins in the 3D stack and preserves routing resources for easier access and routing in the next step, respectively. The netlist is subsequently modified to instantiate the cells of the 3D process design kit (PDK) corresponding to the tier assignments.

This PDK includes shrunk cover memory macros and is updated at every level with the newly generated sub-blocks abstractions. The current block is then implemented using the timing and physical abstractions of the lower hierarchy levels. Please note that a given block implementation can span one die (=2D) or two dies (=3D), depending on the designer's choice. Next, the resulting implemented block is abstracted with our physical/timing constraints propagation method. Finally, the stack and the view of the abstracted block can be inverted for the next step to place standard cells on the opposite die. This loop continues until all hierarchy levels have been implemented.

Table 1: Qualitative comparison among state-of-the-art physical design tools for 3D ICs and this work.

	Sequential-2D	Macro-3D [2]	Hier-3D
key idea	implement two 2D dies separately	holistic memory-on-logic	hierarchy scheme for 3D
3D stack	two separate dies	double metal stack	double metal stack
main strengths	reuse of 2D environment, macros and standard cells in both tiers	full utilization of metal resources	full utilization of metal resources, maximize silicon area utilization
main weakness	dies are designed separately: limited optimization	limited to memory-on-logic: best suited for equal die area	inherently hierarchical: best suited for large designs
restricted partitioning	no	yes	no
holistic routing	no	yes	yes
utilize unused space	no	no	yes

The rest of this section focuses on the detailed presentation of the critical steps of the Hier-3D flow.

3.2 3D Pin Placement in Double Metal Stack

Our proposed methodology can exploit the tool's capability to freely assign a layer (z dimension) and all the block area (x, y dimensions) when placing the pins. By appropriately selecting the layers of the pins, the routing of the subsequent hierarchy level can be guided to utilize a particular die, offering additional options to plan the routing resource allocation.

Our in-house script automates the pin placement by creating a staggered pin grid with routing keep-out-zones (KoZ). These zones force the router to legalize in advance the F2F via placement on the pin in the following step. The pin grid also allows a denser signal routing for very wide IO busses, which would otherwise allocate many routing and placement resources for fan-out and fan-in only. The KoZ dimensions must be superior to the F2F pitch and are empirically set to 5× the F2F pitch in our experiments to allow a design rule check (DRC)-clean routing solution in advanced nodes.

3.3 Holistic Routing Resource Budgeting

The routing resources may need to be budgeted individually by planning and reserving resources at the block level to ease the routing in the upper hierarchy level. For example, if the first level utilizes all metal layers in the doubled stack, very inefficient detours of critical nets through the die stack might occur in the following hierarchy level. We circumvent this by constraining the routing of the nets that do not require both BEOLs' traversal. In our experiments, our budgeting balances the routing resources between sequentially implemented sub-blocks by restricting nets to the die where standard cells are being placed, through the intermediary of the *Cadence Innovus* command set_route_attributes.

3.4 Physical/Timing Constraint Propagation

To enable the utilization of unused placement and routing resources by the P&R engines, we extend the Library Exchange Format (LEF) abstractions of implemented sub-blocks to enumerate all objects and structures in the placement and routing database instead of wholly occupying all resources in the sub-block area. The physical abstractions are represented as detailed LEFs with an abstract BLOCK class type. In particular, the top memory macros projected as site-sized virtual cells during the current block implementation are exported as obstructions on the OVERLAP layer and as detailed routing obstructions (OBS statements) for the next hierarchical level, as shown in Figure 2. Because single standard cells cannot be propagated individually due to the shape complexity, which would cause high memory usage and file size, their shapes can be first clustered into much larger 2D polygons and saved as rectilinear-shaped overlaps similar to the memory macros. This approach enables a full context view of the current level and implemented sub-blocks with reduced memory requirements. Besides, because the router is now free to route through partitions, we improve the routing availability without the large runtime downsides of assembling sub-blocks as partitions. In addition, accurate timing representations are modeled by timing arcs from post-route extracted Liberty (LIB) files.

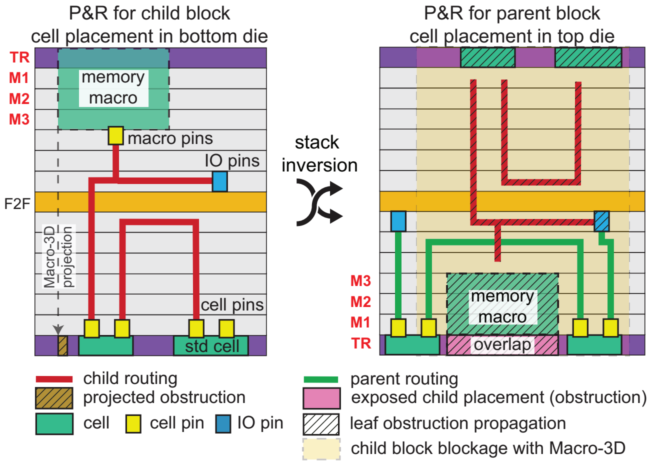

Figure 2: Hier-3D's physical constraint propagation, stack inversion, and multi-tier cell placement. Top macros are projected as site-sized cells not to obstruct standard cell placement in the bottom. IO pins can be placed anywhere inside the stack to facilitate upper-level connections. The physical routing/placement information is propagated to the next level to allow P&R of the top die with the inverted stack.

3.5 Sequential Multi-Tier Cell Placement

Figure 2 illustrates the standard cell placement on the second hierarchy level in Hier-3D. Standard cells can utilize the unused silicon area of the upper die, while in Macro-3D, an additional silicon area around the block is required, increasing the resulting floorplan.

To better exploit the capabilities/assumptions of the P&R tool meant for standard 2D environments during clock tree synthesis (CTS) and routing, we invert the 3D stack, including the abstractions, while traversing the implementation hierarchy. The inversion of a block/stack consists in swapping the top and bottom layer names inside the LEF/technology LEF (TECHLEF) files, as $Mj\_bot \leftrightarrow Mj\_top$. As a result, standard cells are always placed on the "bottom" die from the tool's perspective. During CTS, tree segment definitions assume that the top segment is defined to a higher metal layer than the trunk. However, with a holistic 3D stack, the clock tree should ideally branch

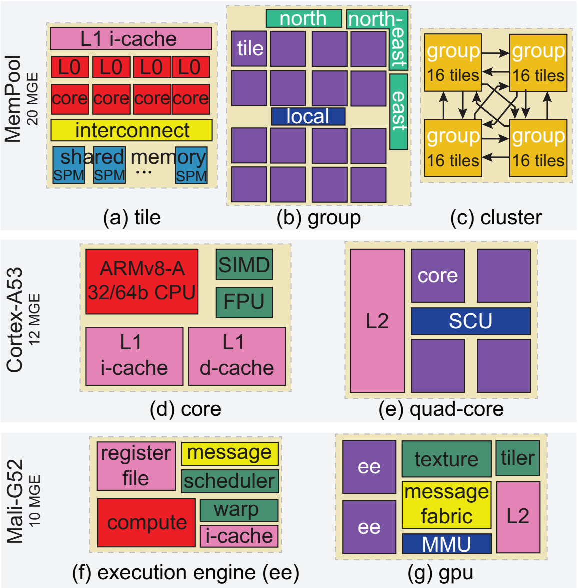

Figure 3: Our three hierarchical benchmarks, MemPool [4], ARM Cortex-A53, and Mali-G52, implemented in TSMC's 28 and 16 nm processes.

into two trunk and leaf definitions for the two FEOLs. This assumption remains valid with the inverted stack during the standard cell placement step. The clock balancing across tiers is simplified by the bottom-up hierarchical approach and by placing all standard cells on one tier per step. In addition, the opposite-tier macros—blocks or memories—have a balancing requirement automatically integrated inside their LIB through the max_clock_tree_path attribute. Moreover, during routing, the assumption of reducing electrical resistance with higher metal layers holds for the FEOL of the standard cells currently being placed, leading to a more standard metal layer configuration and, therefore, more effective use of the router's heuristics.

By default, the blocks' LEF obstructions do not prevent the P&R engine from placing cells at illegal positions due to the presence of routed wires from the lower-level hierarchy. Therefore, we purposely replicate the LEF obstructions by creating special wire shapes on metal layers where standard cells have their pin shapes in our target TSMC technologies (M1 for signal pins and M2 for power and ground rails). Finally, we force the tool to check for pin DRC violations during cell placement which then enforces a valid cell placement.

4 INDUSTRY BENCHMARKS

4.1 Architecture Description

To evaluate our proposed flow, we implement MemPool [4] as a representative example for tiled manycore architectures, the ARM Cortex-A53 representing ultra-high efficiency commercial multiprocessors, and the ARM Mali-G52 exemplifying mid-range graphics and multimedia processors. All three benchmarks are implemented using two advanced commercial process technologies. Figure 3 details their very different hierarchical architectures. The diversity and complexity exhibited by these experiments illustrate the general applicability of our novel 3D integration flow for large complex modern multi-core SoCs.

First, MemPool integrates 256 RISC-V cores with 1 MiB of shared scratchpad L1 data memory (SPMs). The *group* connects sixteen *tiles* through a full crossbar interconnect locally and to tiles in the top-level *cluster* through a hierarchical crossbar. Second, the Cortex-A53 single-core includes a 32 KiB L1 d-cache, an advanced SIMD extension, and an FPU. The four cores share 1 MiB of L2 cache with a snoop control unit (SCU). Third, the Mali-G52 has one double-pixel shader core, with two execution engines (EE) with 48 warps each and 128 KiB of L2 memory.

The three designs have 20M, 12M, and 10M gate equivalent (GE) complexity, respectively, thus requiring a hierarchical implementation flow to keep tool execution time reasonable. The complexity per core of the Cortex-A53 is significantly higher than that of MemPool's tile and includes a multi-level cache hierarchy, while MemPool features a software-managed shared L1 SPM. In contrast, the Mali-G52 is dominated by a high standard cell logic, while MemPool's architecture is interconnect-centric with a large amount of memory.

We normalize our data to avoid revealing proprietary information for these commercial processors.

4.2 Hierarchy Management

We follow the flow shown in Figure 1 for hierarchy handling. The hierarchy is defined for MemPool following the original publication [4], while for the ARM designs, we use the recommended hierarchical cuts from the reference documentation, as depicted in Figure 3. Moreover, we apply the provided timing constraints for each level for the synthesis and P&Rs.

4.3 Macro Floorplanning

The benchmarks exhibit heavy memory/logic imbalance. The single-core Cortex-A53 is dominated by standard cells, while memory macros dominate its upper level. In contrast, standard cells dominate all hierarchy levels of the Mali-G52 and MemPool.

The 2D floorplans of the MemPool sub designs are obtained from the original paper [4]. The 2D floorplans of the ARM Cortex-A53 and Mali-G52 are industrial-strength based on the official documentation. Instead, the Sequential-2D and Macro-3D floorplans are identical and obtained by projecting the 2D macro placement to the top tier and shrinking the floorplan to reach iso-logic density with the 2D counterpart. Note that the dimensions of the instantiated sub-blocks significantly constrain these highly area-optimized floorplans.

In Hier-3D, we implement the leaf level on the bottom die and invert the stack in each subsequent level. Note, however, that our flow further allows the study of diverse and complex floorplan constellations with different rules to decide which blocks are 3D or 2D and when and for which sub-blocks the stack inversion is applied. This enables the exploration of 3D logic-on-logic partitioning scenarios without compromising the theoretical partitioning design space.

5 EXPERIMENTAL RESULTS

5.1 Technology Settings

Our proposed approach can be applied to face-to-back (F2B) or F2F die stacking. However, we focus on F2F W2W bonding for our case studies, where two prefabricated wafers are stacked and bonded together on their top metal layer. While integration techniques such as die-to-wafer bonding exist to process differently sized dies [7], W2W bonding offers a vertical connection pitch of less than 1 µm [3] but requires two dies with matching dimensions.

We use a commercial TSMC 28 nm, high-κ metal gate, planar technology to implement the MemPool design, and TSMC 16 nm, FinFET Compact technology for the Cortex-A53 and Mali-G52 implementations. In the 3D implementations, the F2F via size, pitch, resistance,

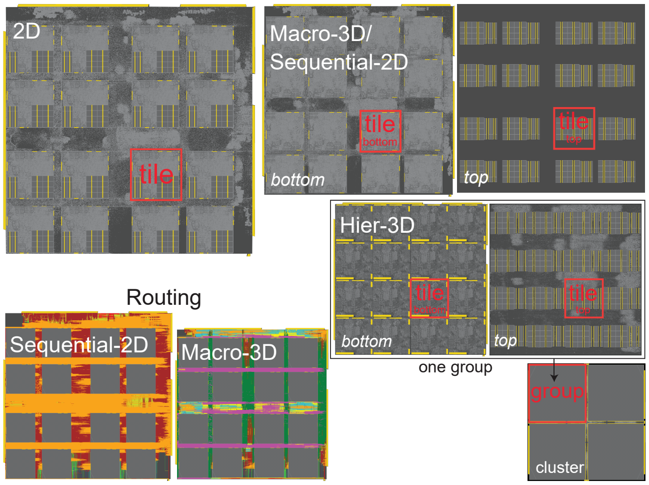

Figure 4: Placement of the MemPool 16-tile group designs using TSMC 28 nm process: 2D vs. Macro-3D/Sequential-2D vs. Hier-3D and Sequential-2D, Macro-3D routing.

Table 2: 64-tile MemPool Cluster 2D vs. Sequential-2D vs. Macro-3D vs. Hier-3D. 7.3M cells & 13.1M nets & 1536 memory macros. Values are normalized w.r.t. 2D implementation.

MemPool	2D	Seq-2D	Macro-3D	Hier-3D
metals used	6	6 (bot) 6 (top)	6 (bot) 6 (top)	6 (bot) 6 (top)
silicon area	1	1.49	1.14	0.75
total WL	1	0.87	0.80	0.71
density (%) bot/top	56.2	50.8/22.6	62.4/29.6	84.4/53.4
buffer count	1	0.91	0.67	0.61
# F2F bumps	-	130K	813K	482K
effective freq	1	1.00	1.05	1.42
total power	1	0.98	0.89	0.80
power × delay	1	0.98	0.85	0.57
die cost	1	1.57	1.19	0.79
power perf cost	1	0.65	0.99	2.22
runtime	1	1.09	0.91	0.68

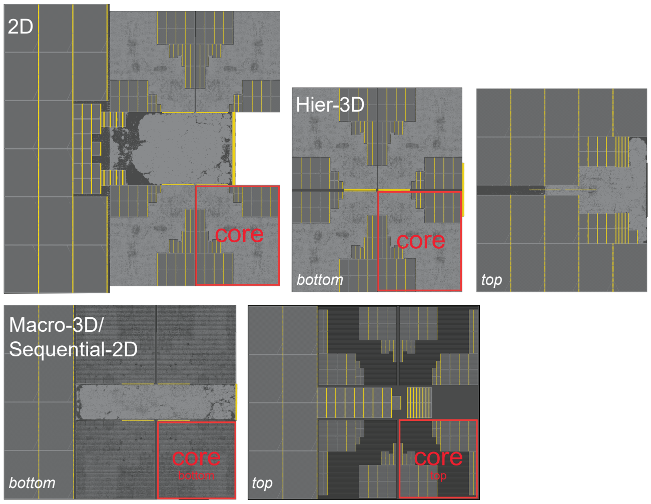

Figure 5: Placement (to scale) of the Cortex-A53 4-core designs using TSMC 16 nm process: 2D vs. Macro-3D/Sequential-2D vs. Hier-3D.

Table 3: 4-Core Cortex-A53 2D vs. Sequential-2D vs. Macro-3D vs. Hier-3D. 2.4M cells & 2.5M nets & 165 memory macros. Values are normalized w.r.t. 2D implementation. 2D silicon area does not include cutouts.

Cortex-A53	2D	Seq-2D	Macro-3D	Hier-3D
metals used	8	7 (bot) 6 (top)	7 (bot) 6 (top)	6 (bot) 7 (top)
silicon area	1	1.21	1.21	0.95
total WL	1	1.02	0.97	0.94
density (%) bot/top	77.5	73.7/62.6	72.1/62.5	81.0/90.7
buffer count	1	1.15	1.05	0.98
# F2F bumps	-	22K	81K	74K
effective freq	1	0.93	0.95	1.33
total power	1	1.14	0.97	0.97
power × delay	1	1.22	1.02	0.73
die cost	1	1.13	1.13	0.91
power perf cost	1	0.78	0.87	1.51
runtime	1	1.12	0.89	0.82

and capacitance are 0.5 µm×0.5 µm, 1.0 µm, 0.5 Ω, and 1 fF, respectively [3]. The 3D BEOL is defined in a custom 3D TECHLEF file where metal layers are replicated and mirrored, separated by a F2F via layer of 0.175 µm thickness. In addition, the custom TECHLEF includes design rules for the double metal stack. Based on a custom interconnect technology (ICT) file, we simulate the metal layers' resistance/capacitance (RC) and copper pads.

We use the in-house tools of Macro-3D [2] and implement the Sequential-2D flow as the construction of two sequential 2D implementations [8]. For a fair comparison, the 3D implementations include a (close to) balanced mirrored stack of the 2D configuration. Furthermore, we implement the designs with a *max-performance* target at the typical corner in all our experiments. Finally, our Hier-3D flow is automatized with *Tcl* and *Bash* scripts inside the *Cadence Innovus* environment.

5.2 MemPool Design Results

While the tile implementation PPA metrics are very similar across all integration flows, the group level is critical in the implementation of MemPool. The group is highly connected in the center, where the engine places most of the local interconnect logic. This creates heavy congestion, degrading timing, and increasing routing DRCs if the tiles are not spaced sufficiently. Thus, the floorplan size for Sequential-2D must be increased to obtain a DRC-clean design due to the reduced stack awareness compared to the Macro-3D implementation, as depicted in Figure 4. With our flow, the block-to-block spacing can instead be reduced to only 5 µm thanks to the shared BEOL and the use of both FEOLs for standard cells, providing substantial area and cost reductions. Moreover, this reduces the net lengths, resulting in significant power reduction and performance increase.

Table 2 highlights the huge PPA savings of Hier-3D and the resulting Power Performance Cost (PPC) metric computed using the methodology presented in [1] as PPC = Frequency / (Die Cost × Power). We see an impressive 2.2× PPC improvement, where all individual metrics are noticeably improved, which is quite unique. This result reflects the benefits of our flow in terms of higher die stack utilization.

5.3 Cortex-A53 Design Results

The PPA results of the single-core implementations are similar across all key metrics. The Macro-3D/Sequential-2D quad-core floorplan stacks two L2 data macros on top of each other, reducing the design footprint. However, a significant amount of silicon area in the upper die remains unused, as shown in Figure 5. The Hier-3D floorplan

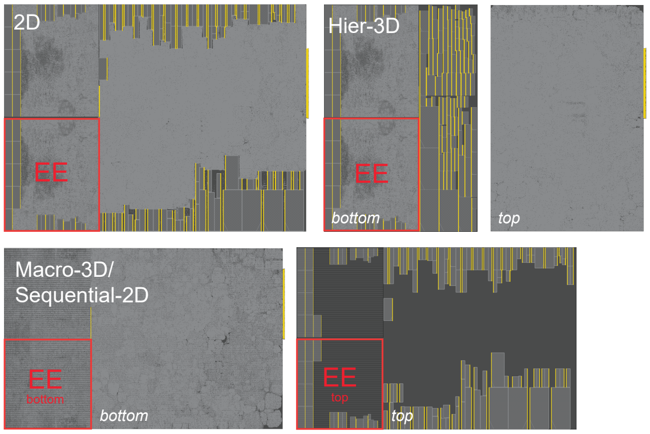

Figure 6: Placement (to scale) of the Mali-G52 2-EE designs using TSMC 16 nm process: 2D vs. Macro-3D/Sequential-2D vs. Hier-3D.

Table 4: 2-Execution Engine Mali-G52 2D vs. Sequential-2D vs. Macro-3D vs. Hier-3D. 4.4 M cells & 4.8M nets & 141 memory macros. Values are normalized w.r.t. 2D implementation.

Mali-G52	2D	Seq-2D	Macro-3D	Hier-3D
metals used	8	7 (bot) 6 (top)	7 (bot) 6 (top)	6 (bot) 7 (top)
silicon area	1	1.45	1.45	0.99
total WL	1	0.88	0.86	0.98
density (%) bot/top	77.7	74.5/31.3	74.3/31.3	78.1/70.2
buffer count	1	0.93	0.93	0.97
# F2F bumps	-	56K	325K	149K
effective freq	1	0.98	0.95	1.15
total power	1	1.14	0.96	0.98
power × delay	1	1.16	1.01	0.85
die cost	1	1.35	1.35	0.95
power perf cost	1	0.64	0.73	1.24
runtime	1	0.99	1.08	0.81

instantiates the 2D single-core abstraction, leaving the four single-core footprints unutilized in the upper die. Therefore, the top-level memory macros and SCU standard cells can be placed on top of the single cores, further reducing the silicon area.

In the quad-core implementation, Hier-3D optimizes all the limiting competing paths between the SCU standard cells to the L2 data macros, the single-cores, and the IOs, thanks to the denser floorplan and increased routability, yielding a significant frequency increase. Table 3 shows that the Hier-3D flow surpasses the Macro-3D flow in frequency and power, with drastic improvement in silicon area. Again, all composing metrics are improved simultaneously, providing a total PPC bump of 51 % over the 2D reference.

5.4 Mali-G52 Design Results

We floorplan the Hier-3D GPU using a 2D bin-packing method, assigning the upper and lower edge macros of the 2D floorplan into two separate bins. The packed macros are placed next to the 2D EE abstractions on the bottom die, allowing the top-level standard cells to fully utilize the upper die, as shown in Figure 6.

Table 4 shows a 15 % increase in frequency while reducing the total silicon area compared to 2D and a substantial die cost reduction compared with the two other 3D flows. Modifications of the macro placement would yield further wire length reduction and PPC improvements at the expense of the silicon area gains.

5.5 Summary

Despite the excellent reference of the timing-optimized industry-recommended 2D IC floorplans, our implementations consistently achieve a smaller silicon area while delivering substantial PPC improvements for the three benchmarks, even though the three designs are pretty different. Typically, designs display a constant power-delay product, where a frequency gain increases power. However, compared to the competing flows, Hier-3D remarkably overcomes this power versus performance trade-off, improving both metrics simultaneously.

The experiments run on a Linux server with a 24-core Intel Xeon E5-2640 @ 2.5 GHz with 15 MiB L3 cache. The substantial runtime improvements of Hier-3D observed on all benchmarks—the Mali-G52 P&R runtime is about three to four days—make it scalable to sizeable modern multi-core SoCs.

6 CONCLUSION

We propose a full-chip RTL-to-GDSII physical design solution that offers a commercial-quality F2F-bonded 3D IC physical layout for large hierarchical designs. Our flow includes new critical ideas, such as the routing and placement constraint propagation in the double metal stack view and stack inversion, enabling multi-tier cell placement. This design flow steppingstone vastly expands the design space exploration options and can help explore physical hierarchy more efficiently on a multi-level for 3D ICs. Our extensive experiments on large complex hierarchical designs of an open manycore processor and industrial ARM application and graphics processors show our flow offers 15 to 43 % power × delay reduction and more than 1.2× combined power, performance, and area/cost improvements compared with the 2D counterparts.

ACKNOWLEDGMENTS

This research is partially funded by the DARPA ERI 3DSOC Program under Award HR001118C0096, the Semiconductor Research Corporation under Task 2929, and the National Research Foundation of Korea under NRF-2020M3F3A2A02082445.

REFERENCES

[1] Anthony Agnesina et al. 2021. Power, Performance, Area and Cost Analysis of Memory-on-Logic Face-to-Face Bonded 3D Processor Designs. In *2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*.

[2] Lennart Bamberg et al. 2020. Macro-3D: A Physical Design Methodology for Face-to-Face-Stacked Heterogeneous 3D ICs. In *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*.

[3] E. Beyne et al. 2017. Scalable, sub 2um pitch, Cu/SiCN to Cu/SiCN hybrid wafer-to-wafer bonding technology. *2017 IEEE International Electron Devices Meeting (IEDM)* (2017).

[4] Matheus Cavalcante et al. 2021. MemPool: A Shared-L1 Memory Many-Core Cluster with a Low-Latency Interconnect. In *2021 Design, Automation Test in Europe Conference Exhibition (DATE)*.

[5] Johann Knechtel, Igor L. Markov, and Jens Lienig. 2012. Assembling 2-D Blocks Into 3-D Chips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2012).

[6] C. Niessen. 1983. Hierarchical design methodologies and tools for VLSI chips. *Proc. IEEE* (1983).

[7] Alain Phommahaxay et al. 2019. Enabling Ultra-Thin Die to Wafer Hybrid Bonding for Future Heterogeneous Integrated Systems. *2019 IEEE 69th Electronic Components and Technology Conference (ECTC)* (2019).

[8] Giuliano Sisto et al. 2019. Design Enablement of Fine Pitch Face-to-Face 3D System Integration using Die-by-Die Place Route. In *2019 International 3D Systems Integration Conference (3DIC)*.