IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

Received 28 April 2021; revised 5 July 2021 and 6 August 2021; accepted 23 August 2021.
Date of publication 30 August 2021; date of current version 28 September 2021.

Digital Object Identifier 10.1109/JXCDC.2021.3108898

An Effective Block Pin Assignment Approach for Block-Level Monolithic 3-D ICs

JINWOO KIM[1] (Graduate Student Member, IEEE), BON WOONG KU[2], JUNSIK YOON[1], and SUNG KYU LIM[1] (Senior Member, IEEE)

[1]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA
[2]Synopsys Inc., Mountain View, CA 94043 USA

CORRESPONDING AUTHOR: J. KIM (jinwookim@gatech.edu)

This work was supported by the Defense Advanced Research Projects Agency (DARPA) Electronics Resurgence Initiative (ERI) 3DSOC program under Award HR001118C0096.

ABSTRACT In a 2-D design, the block pins are located at the periphery of a block optimally since blocks are placed side-by-side horizontally in a single placement layer. However, monolithic 3-D (M3D) integration relieves this boundary constraint by allowing vertical block communication between different tiers based on an nm-scale pitch of 3-D interconnection. In this article, we present a design methodology named pin-in-the-area that assigns block pins at any position inside the boundary of a block using commercial 2-D place-and-route (P&R) tools and enables an efficient block implementation and integration for a block-level M3D integrated chips (ICs). Our pin-in-the-area starts from the netlist restructuring and connectivity-aware tier-by-tier chip planning, which defines blocks and decides their sizes and (X, Y, Z) locations for a two-tier M3D design. Next, we perform wirelength-driven 3-D placement to minimize 3-D half-perimeter wirelength (HPWL) and find optimal pin locations inside the boundary of a block. Once block designs are done, we apply the unique macro handling scheme to the top-level timing closure. Based on a 28-nm two-tier M3D hierarchical design result, we show that our solution offers 13.6% and 24.7% energy-delay-product reduction compared to the M3D design with pins assigned at the block boundaries and its 2-D counterpart, respectively.

INDEX TERMS 3-D integrated circuits, block pin assignment, block-level design, physical design (EDA), pin-in-the-area.

I. INTRODUCTION

The functional density of integrated circuits has been increasing thanks to the device scaling as of today. However, moving toward the 3-nm technology era is not predicted to be the same as before because, on top of the geometric scaling, the new device architectures, such as gate-all-around field-effect transistor (FET) and complementary FET, increase the process complexity and require the reduction in new additional defect mechanisms. Therefore, the traditional geometric scaling continues in combination with various vertical stacking approaches at the integration level.

Stacking multiple dies in 3-D fashion has evolved in many different ways, including stacking of either packaged dies [package-on-package (PoP)] or bare dies [stacked-integrated-circuits (SiC)], to realize smaller 3-D interconnection pitch. In the packaging-level 3-D integration, 3-D interconnects have been made by ball-grid-array and wire-bonding technologies, which is 100-μm-scale pitch.

In the die-level 3-D integration, microbump array and through-silicon-via (TSV) technologies have been used for 3-D interconnects. While a 6-μm [1] pitch has been demonstrated in the advanced TSV technology, a 20-μm [2] pitch of microbump array has been the main bottleneck. However, the wafer-level 3-D integration opens up a new era of 3-D integration by enabling bumpless sub-μm 3-D interconnects [3].

The sequential face-to-back wafer-level 3-D integration, which is also known as M3D, is an emerging 3-D integration technology that enables the monolithic fabrication by stacking multiple active layers vertically [4]. In block-level M3D integrated chips (ICs), blocks are placed on different tiers and routed using M3D technology. Existing works on block-level M3D ICs [5], [6] have developed simulated annealing-based 3-D floorplanning engines and presented promising power, performance, and area (PPA) savings. However, these works have assumed that the block pins are assigned along the periphery of each block.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

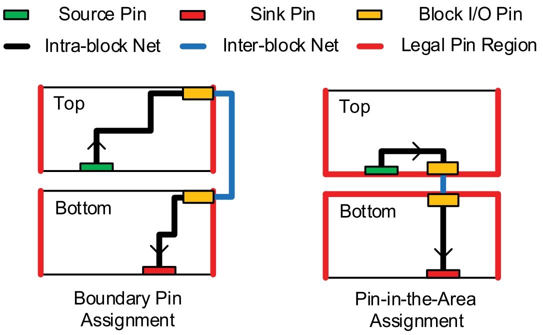

FIGURE 1. **Suppose two blocks are placed on the top and bottom tiers each, and one completely overlaps another. With block pins along the periphery of each block, wirelength is unnecessarily lengthened to touch the boundary pins. Pin-in-the-area enables pin assignment inside the boundary of a block, resulting in efficient interblock/intrablock net connections.**

As shown in Fig. 1, it is evident that the periphery of a block is not always an optimal location for the pin assignment when functionally partitioned blocks are vertically stacked. This is because interblock connections are unnecessarily lengthened to touch the boundary pins. Therefore, we develop an efficient block-level M3D IC design methodology named pin-in-the-area that tackles this type of pin assignment problem. The tight 3-D interconnection pitch achieved by M3D technology allows accommodating optimal pin locations inside the boundary of blocks.

In this article, we present an M3D hierarchical design approach named pin-in-the-area based on [7] giving freedom to the block pin locations assuming soft IP blocks. In the traditional 2-D hierarchical designs, block pins are assigned at the boundary of each block because connections among blocks are always made at their boundaries. However, the tight 3-D interconnection pitch in M3D technology allows direct vertical connections between the module blocks. By realizing block pin assignment at any location inside the boundary of each block, we show that the redundant routing detour for the interblock connections has been minimized and improve the top-level timing closure.

II. RELATED WORK

The pin assignment is one of the most important problems to reduce wirelength, meanwhile optimizing both critical delay and power consumption in M3D integration [5], [6]. Previously, the pin assignment in M3D has been optimized by simultaneous optimization of pin assignment, TSV placement [8], and genetic and simulated annealing algorithm to co-optimize area, wirelength, and temperature [9].

Zhong *et al.* [8] have proposed a heuristic algorithm based on Lagrangian relaxation to solve the pin assignment and TSV planning problem. By formulating the problem as a min-cost multicommodity flow model, they have improved the total wirelength of the target design by 38%. Hu *et al.* [9] proposed genetic and simulated annealing (GA-SA) algorithm to find the optimal 3-D IC floorplanning and assignments of on-chip and input/output (I/O) package pins considering the area, interconnection length, maximum temperature, and inductance of pins of the design.

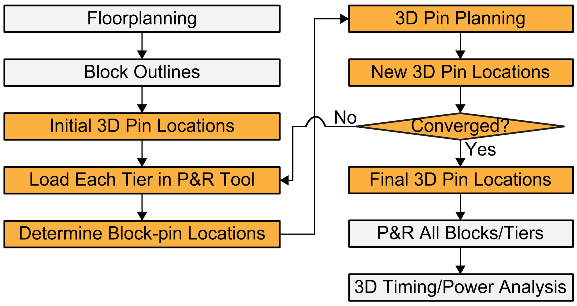

FIGURE 2. **Block-level design flow [5] that places block pins at the boundaries. Pin assignment steps are highlighted in yellow.**

However, these previous studies do not deviate from wire lengthening and wire congestion problems since the pins are placed at the boundary of blocks. In this work, pin-in-the-area design is the first work to improve the overall wirelength of block-level M3D ICs by providing more degrees of freedom for pin placements by placing block pins at any location inside the boundary.

III. PIN-IN-THE-AREA FLOW

A. PREVIOUS BLOCK-LEVEL DESIGN FLOW

Fig. 2 shows the previous block-level design flow [5], which places the block pins at the boundaries of blocks. In this flow, the outlines of all blocks are given in the 3-D space. Then, we determine the locations of 3-D pins and block pins. In this step, we give the initial 3-D pin locations and perform block pin assignments. Next, the new 3-D pin locations are updated based on the determined block pin locations. As it is a chicken and an egg problem, the pin planning flow is iterated until 3-D pin locations become stable.

Even though the final 3-D pin locations are optimized, 3-D pins are placed outside the block boundaries. With these results, 3-D nets should have detour paths, which can be further optimized. Therefore, we present pin-in-the-area, which provides the optimal block pin locations inside the boundary of a block using commercial 2-D place-and-route (P&R) engines to build high-quality two-tier block-level M3D ICs.

B. OVERVIEW OF PROPOSED FLOW

First, pin-in-the-area flow begins with netlist restructuring to divide the overall design into optimal functional blocks for a two-tier M3D design considering the hierarchy. Next, we perform tier-by-tier chip planning, which follows where we decide the shape and the 3-D location of a block. Once the floorplanning is done, a wirelength-driven 3-D placement is performed to co-optimize the block-level placement and pin locations that are at any location inside the boundary of a block iteratively to improve the wirelength of interblock and intrablock nets.

We then proceed with timing budgeting where the wirelength saving turns into the additional block-level timing

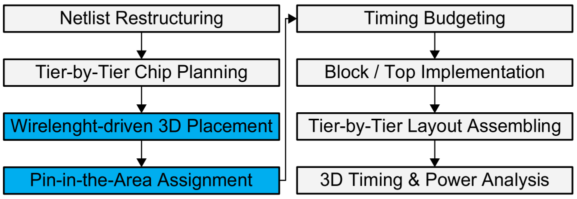
Netlist Restructuring
Tier-by-Tier Chip Planning
Wirelenght-driven 3D Placement
Pin-in-the-Area Assignment
Timing Budgeting
Block / Top Implementation
Tier-by-Tier Layout Assembling
3D Timing & Power Analysis

FIGURE 3. **Overview of pin-in-the-area flow. Pin assignment steps are highlighted in blue.**

margin. This step allows P&R tools to remove or downsize the redundant buffers during block implementation, which leads to reduced total power consumption. After block implementation and top-level timing closure, we finally perform sign-off 3-D timing and power analysis using assembled tier-by-tier layouts. The overview of pin-in-the-area flow is shown in Fig. 3.

C. BENCHMARK: RISC-V DUAL-CORE ROCKET PROCESSOR

RISC-V [10] is an open-source general-purpose instruction set architecture (ISA) based on reduced instruction set computing (RISC) principles. In this work, we implement a 28-nm dual-core rocket processor [11], [12], an open-source micro-architecture that executes scalar RISC-V ISA with a six-stage single-issue in-order pipeline. Rocket processor implements a memory management unit (MMU) that supports page-based virtual memory and is able to boot modern operating systems, including Linux.

Both caches are virtually indexed physically tagged with parallel translation lookaside buffer (TLB) look-ups. The data cache is nonblocking, allowing the core to exploit memory-level parallelism. A 64-entry branch target buffer, 256-entry two-level branch predictor, and return address stack together mitigate the performance impact of control hazards. Rocket has an optional IEEE 754-2008-compliant FPU, which can execute single- and double-precision floating-point operations, including fused multiply–add (FMA), with hardware support for denormals and other exceptional values. The fully pipelined double-precision FMA unit has a latency of three clock cycles.

D. NETLIST RESTRUCTURING

In this work, we use a commercial-grade 28-nm process design kit (PDK) and build a dual-core rocket processor whose block diagram is shown in Fig. 4. The original register-transfer level (RTL) netlists of rocket processors contain a multidepth block hierarchy. However, a block-level M3D IC makes 3-D connections only at the top level, while individual blocks remain as 2-D. Therefore, we transform the netlist into a two-level hierarchy, which includes top level and blocks.

We first flatten the netlist below the fourth hierarchy and ungroup tiny blocks whose standard cell area is under 10 μm^2. We then create a module by merging blocks if the sum of the macro area from those modules is over 10 μm^2. This netlist restructuring process is iterated until the second hierarchy, resulting in an area-balanced module definition. The netlist restructuring has been done semimanually by considering the functionality of each module.

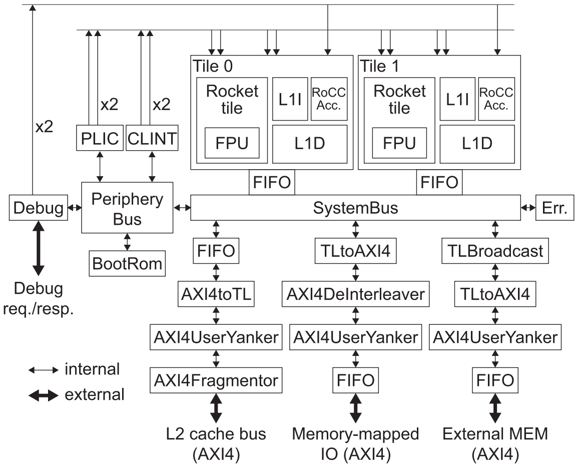

FIGURE 4. **Block diagram of dual-core rocket processor.**

TABLE 1. **Seventeen blocks of 28-nm RISC-V dual-core rocket processor [10]–[12] defined by the netlist restructuring process.**

Unit Group	Block Definition	Merged Functionality	Area (mm^2)
Core Unit (Core0)	DCache_0	D-Cache	0.102
	Frontend_0	I-Cache	0.151
	Rocket_0	Pipeline Logic	0.134
	FPU_0	Floating Point Unit	0.190
	Tile_0	Core Interface	0.026
Core Unit (Core1)	DCache_1	D-Cache	0.102
	Frontend_1	I-Cache	0.151
	Rocket_1	Pipeline Logic	0.134
	FPU_1	Floating Point Unit	0.190
	Tile_1	Core Interface	0.026
Interface Unit	PeripheryBus_1	Periphery Bus	0.017
	SLModule_6	Slave-AXI4	0.015
	SLModule_7	MemCon-AXI4	0.051
	SLModule_2	MMIO-AXI4	0.078
	TLBroadcast	System Bus	0.054
	TLDebugModule	Debuging	0.014
	TopSubModule	Mixed	0.035

Note that the area threshold for ungrouping depends on the technology node and the design benchmark. According to the technology and benchmark design, the overall design area varies. Therefore, the area threshold should be set based on those variables. Table 1 shows the definitions of blocks as the result of the netlist restructuring process. Each core is divided into pipeline logic, floating-point unit, instruction/data (I/D)-caches, and interface buffer blocks. Including bus units and memory controller units, totally 17 blocks have been defined in the end.

E. TIER-BY-TIER CHIP PLANNING

When blocks are defined by netlist restructuring, we synthesize the netlists and load them to a floorplanning engine. Tier-by-tier chip planning includes the decision of tier location of blocks and the floorplan of separate tiers. This block-level tier partitioning stage is critical to the final design quality in which it decides the number of 3-D block interconnections.

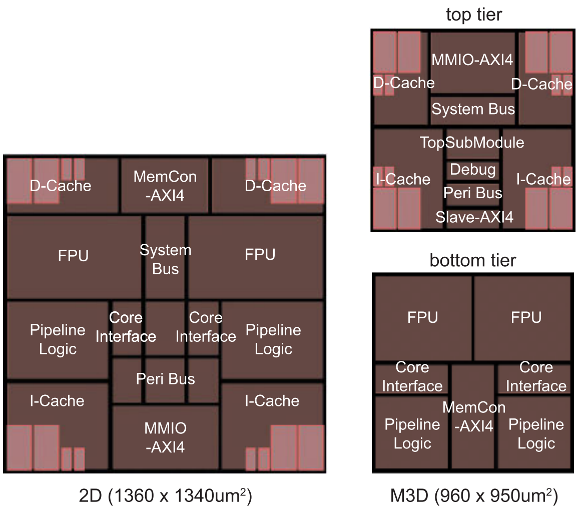

FIGURE 5. **Chip planning results for 2-D and M3D RISC-V dual-core rocket processor designs, respectively. The footprint of M3D design is set to be the half of 2-D design footprint assuming no silicon area overhead.**

Given that we build a two-tier M3D IC, the form factor (footprint) of our M3D IC is as 50% small as its 2-D IC counterpart assuming no silicon area overhead. When we decide the size and (X, Y, Z) location of each block, any two blocks with dense connections need to be stacked on each other as much as possible to maximize the benefits of direct vertical connections. In our M3D designs, we place I/D-cache memories on the top tier and the core blocks on the bottom tier to realize the memory-on-logic stacking principle, which targets efficient 3-D logic-memory interconnects. Next, we decide the tier locations of other logic blocks to balance the area skew on both tiers while optimizing the block interconnection.

After that, we insert an additional hierarchy level, which represents a tier between the top level and blocks in the synthesized netlist to organize it into the three-level hierarchy (top level, tiers, and blocks). Then, we generate two netlists at the second hierarchy for the top and bottom tiers. Each tier netlist is loaded on the floorplan engine again, and we perform tier-by-tier chip planning. It is obvious that this netlist modification defines a 3-D interblock net as a top-level net connected to the I/O port of each tier. Fig. 5 shows floorplans of 2-D and M3D RISC-V dual-core rocket processor design, respectively. Out of 6903 interblock nets, we achieve 3692 (53%) 3-D interblock nets as a result. Here, die-to-core and block-to-block spacings are 20 μm, and hard macro placement halo is 10 μm. The initial utilization of each block is assumed to be 65% ± 5% in both 2-D and M3D designs.

F. WIRELENGTH-DRIVEN 3-D PLACEMENT

The 3-D-aware placement solution for the individual blocks in the early design stage is necessary for extracting the best timing budget in M3D hierarchical designs. However, 2-D placement engines fail to produce the placement solution when block fences on separate tiers are flattened on the single placement layer. Therefore, we present an iterative tier-by-tier placement approach to produce an optimal 3-D-aware placement solution.

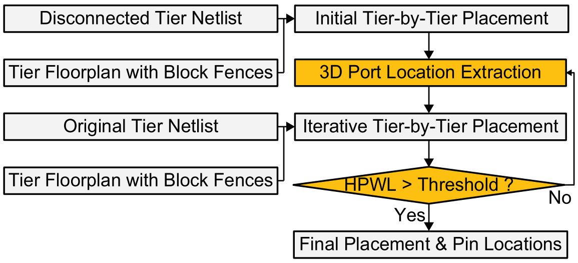
Disconnected Tier Netlist
Tier Floorplan with Block Fences
Original Tier Netlist
Tier Floorplan with Block Fences
Initial Tier-by-Tier Placement
3D Port Location Extraction
Iterative Tier-by-Tier Placement
HPWL > Threshold ?
No
Yes
Final Placement & Pin Locations

FIGURE 6. **Wirelength-driven 3-D placement method. This iterative approach extracts the output pins of driver cells from each tier and uses them as anchors for 3-D-aware tier-by-tier placement.**

Fig. 6 shows our iterative wirelength-driven tier-by-tier global placement method to produce an optimal 3-D-aware placement solution. To make tier-by-tier global placement 3-D-aware, both tiers need to keep synchronizing the target location of 3-D ports inside the boundary of a tier dies. Therefore, the basic idea of our wirelength-driven 3-D placement solution is to iterate the exchange of 3-D port information and tier-by-tier global placement.

In the beginning, we first modify the tier netlist by eliminating all the top-level primary I/O ports and interblock connections (disconnected tier netlist) and run the global placement of each tier. In this way, the initial tier placement does not have any bias against an external block or I/O connections. After the global placement, we extract the output pin location of the driver cell of a 3-D port (defined at the original tier netlist) and decide the location of the 3-D port. Next, both tiers share the information of 3-D port locations. Then, we perform the tier-by-tier placement from scratch with the original tier netlist and extracted 3-D port locations. Those 3-D ports serve as anchors for cells connected to 3-D interblock nets and restrain the placement engine from overoptimizing 2-D interblock nets. Therefore, this anchoring process optimizes both 3-D/2-D interblock connections.

With the RISC-V dual-core rocket processor design, Fig. 7 shows that the total half perimeter wirelength (HPWL) for all interblock nets at the first iteration is reduced by 30.4% compared to the initial tier-by-tier placement solution. Moreover, it monotonically decreases as the iteration proceeds. On the other hand, we observe 5% HPWL overhead with small disturbance for intrablock nets. At the end of each iteration, we compare the total HPWL of nets with the threshold value to meet the exit condition. The threshold value is defined by

$$\text{Threshold} = \text{Min. HPWL} \times (1 + e^{-(\#\text{iteration})}) \qquad (1)$$

where Min.HPWL is the minimum HPWL value during the whole iterations.

As the iteration proceeds, we update the 3-D port locations based on the latest tier-by-tier placement solution. When

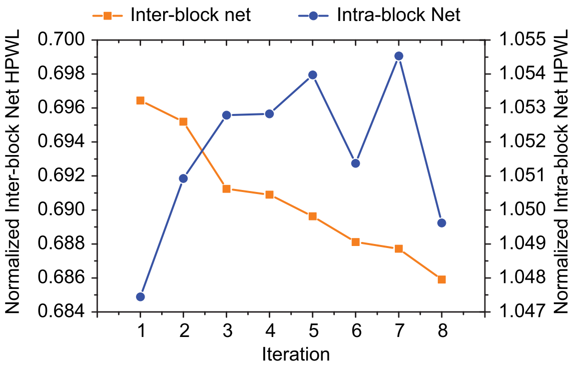

FIGURE 7. HPWL changes as the wirelength-driven 3-D placement proceeds. We observe 30.4% of HPWL reduction for interblock nets at the very first iteration, and it decreases monotonically as iteration proceeds. For intrablock nets, a small disturbance around 5% overhead is observed.

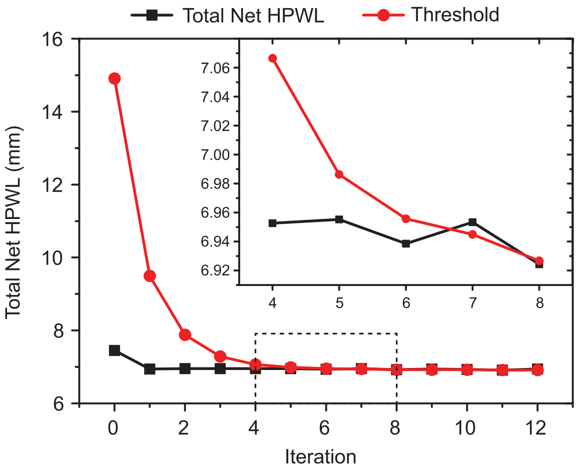

FIGURE 8. Changes in the total net HPWL and exponentially decreasing threshold as iteration proceeds. For the rocket processor design, we meet the exit condition at the seventh iteration and end up with 6.9% total HPWL savings.

the exit condition is met, we use the resulting tier-by-tier placement solution for 3-D timing budgeting. In the Rocket processor design, we meet the exit condition at the seventh iteration, as shown in Fig. 8, and find the optimal tier-by-tier placement solution at the sixth iteration. Fig. 9 shows the movement of cells directly connected to the interblock nets. The standard cells in the bottom tier move toward the optimal 3-D connection points from iterations 0 to 6.

G. PIN ASSIGNMENT AND TIMING BUDGETING

Based on the optimal tier-by-tier placement solution and the result of wirelength-driven 3-D placement, we assign the block pins at the final location of the 3-D port inside the boundary of each block. For these pin assignments, a special 3-D P&R environment using full 3-D metal stack information is required. 3-D technology library exchange format (LEF) contains the full layer definition used for both top and bottom tiers. 3-D macro LEF defines the pin locations of standard cells based on their tier. The *RC* database for the 3-D metal stack (3-D TCH) is also needed for the timing budgeting later.

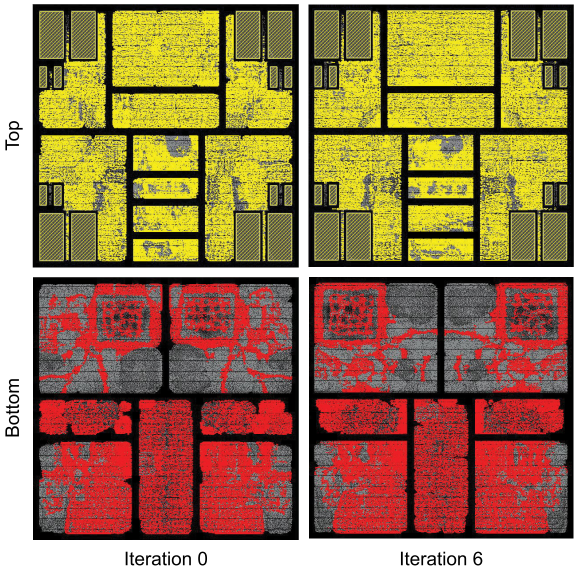

FIGURE 9. Placement changes after wirelength-driven 3-D placement. Cells connected to the interblock nets are in color.

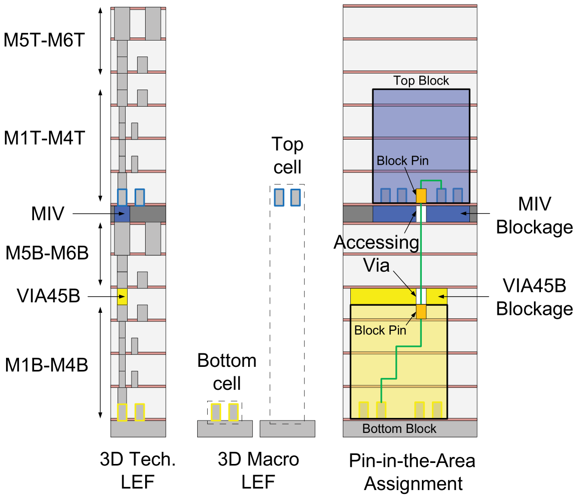

FIGURE 10. We reserve up to M4 layer for intrablock nets and use M5 and M6 layers for interblock connections on both tiers in an M3D design.

From these input files, we instantiate standard cells in a top-tier block with the top-tier macro LEF and cells in a bottom-tier block with the bottom-tier macro LEF. Since we know the tier location of cells, we update the macro of the cell in the original netlist based on its tier location. Finally, we load both tier-by-tier placement solutions on a single placement layer to accommodate all the synthesized cells of the design in the P&R tool. Although there are lots of overlaps between the top-tier and bottom-tier cells in this synthetic placement, the cell pin locations of them are still apart thanks to 3-D macro LEFs, as shown in Fig. 10. As there are unplaced top-level cells, additional top-level placement can proceed at this point to implement a full 3-D placement solution.

In our designs, we utilize six metal layers for the 2-D IC and for both top and bottom tiers in the M3D IC. We reserve

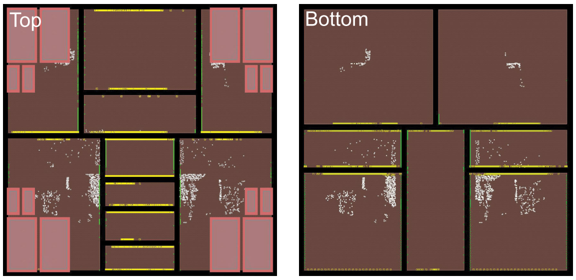

FIGURE 11. Result of pin assignment. Three-dimensional pins inside the boundaries of blocks are colored in white. 2-D pins are still assigned at the block boundaries if needed. The size of each pin is magnified by 50x for the visualization.

M1–M4 layers for intrablock nets and use M5 and M6 layers for interblock connections. Based on the full 3-D placement, we first complete the assignment for a block pin connected to a 3-D interblock net (3-D pin). 3-D pins for a bottom-tier block are fixed at the M4B layer inside the boundary of the block and the M1T layer for a top-tier block. Next, we create routing blockages at the VIA45B layer on top of a bottom-tier block and at the MIV layer underneath a top-tier block.

In addition, we create an accessing via at a 3-D pin location inside the boundary of a blockage layer. This prevents the routing engine from utilizing metal layers reserved for blocks and accessing the blocks through the block regions unless the block pins are assigned as 3-D ports. Once detail routing is done honoring the routing blockage, remaining block pins (2-D pin) are assigned at the periphery of a block automatically based on the interblock routing information. Fig. 11 shows the pin assignment result. Finally, based on the extracted parasitics, the timing constraints for individual blocks and top-level design are generated, and we use them for the block-/top-level implementations.

H. TOP-LEVEL TIMING CLOSURE

For top-level timing closure, we apply the state-of-the-art gate-level M3D flow [13] to our block-level M3D IC environment. As shown in Fig. 12, we first treat the individual blocks as hard macros and flatten both tiers on the single placement layer. Then, we expand the flattened top-level floorplan up to the 2-D IC footprint and replace hard macros with placement blockages. When macros are fully overlapped, full placement blockages are created. Otherwise, partial placement blockages with a 50% allowance are created to reflect the empty spaces in the nonoverlapped region. Note that the placement blockage regions also should be expanded by the same expansion factor. Based on those blockage regions, 2-D P&R engines identify the legal buffer placement locations.

Next, we perform the conventional 2-D P&R steps with block timing models and scaled *RC* parasitics to close the top-level timing. Timing buffers are inserted at either white spaces or partial placement blockages during top-level timing closure. Note that parasitic scaling is required to reflect the 3-D wirelength savings in advance. Then, we linearly map the placement result onto the original M3D footprint to determine the final (X, Y) location of top-level buffers. Assuming two-tier M3D ICs without silicon area overhead, $1/(2)^{1/2} = 0.707$ is used for the parasitic scaling and placement contraction factor.

To determine Z location of top-level timing buffers, we use bin-based placement-driven Fiduccia–Mettheyses (FM)-mincut partitioning algorithm [14]. For the top-level 3-D connection, 3-D routing should proceed first, and MIV locations are decided based on the routing result of 3-D interblock nets. Then, our in-house tool generates the new RTL netlist for each tier that contains the MIVs as primary in/out ports. The *RC* model of MIVs is also generated as a standard parasitic exchange format (SPEF) file at this stage. Assembling block layouts at the top-level finalizes the full-chip block-level M3D IC implementation. Fig. 13 shows assembled design layouts of 28-nm RISC-V dual-core rocket processors for the 2-D IC, for the M3D IC with block pins at the boundary of blocks (M3D boundary), and for the M3D IC with block pins inside the boundaries of blocks (M3D area), respectively. The footprint of 2-D IC is 1.82 and 0.91 mm^2 (-50%) for that of M3D ICs; therefore, there is no silicon area overhead in two-tier M3D designs.

TABLE 2. Wirelength of nets based on their connection types. Percentage values for both M3D designs are based on the comparison with 2-D design.

	Wirelength (m)		
Net Type	2D	M3D Boundary	M3D Area
3D Inter-block Nets		0.659	0.646
2D Inter-block Nets	2.781	2.102	2.042
Total Inter-block Nets	2.781	2.761 (-0.7%)	2.688 (-3.4%)
Total Intra-block Nets	8.481	8.185 (-3.5%)	7.984 (-5.9%)
Total Nets	11.262	10.946 (-2.8%)	10.672 (-5.2%)

IV. EXPERIMENTAL RESULTS

A. WIRELENGTH SAVING

Table 2 summarizes the wirelength of nets based on their connection types. As 50% of footprint reductions in both M3D designs lead the wirelength savings at the timing budgeting step, the total wirelength savings of M3D boundary and M3D area designs are 2.8% and 5.2%, respectively. It is worth noting that the wirelength saving on intrablock nets is the major source of total wirelength saving. Given that the wirelength of intrablock nets contributes to 75.3% of the total wirelength in the 2-D baseline, this implies that the benefit of additional timing margin offered by M3D integration affects the design quality more critically than reducing the wirelength of interblock nets directly. Table 2 also shows that the pin-in-the-area flow enables further optimized block implementation based on the better timing margin than that of a block with boundary pins.

B. CELL COUNT AND AREA SAVINGS

As the area of memory modules is kept the same as 0.129 mm^2 in all designs, we tabulate the number and area of standard cells for 2-D, M3D boundary, and M3D area

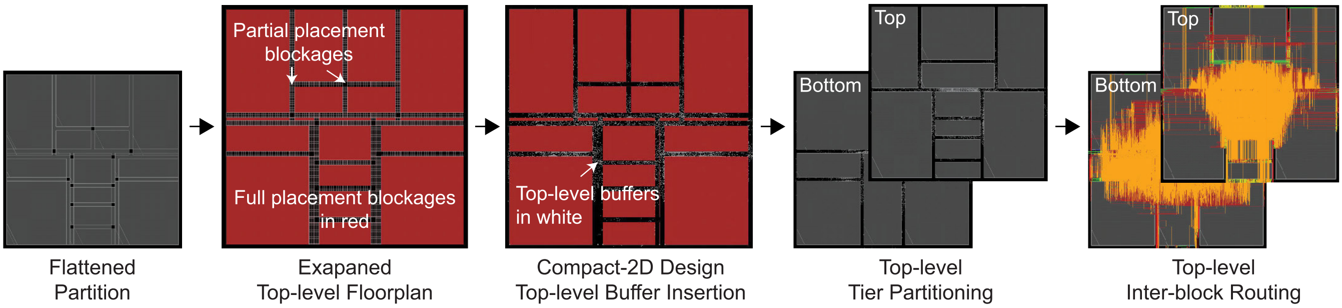
Partial placement blockages
Full placement blockages in red
Top-level buffers in white
Top
Bottom
Top
Bottom
Flattened Partition
Exapaned Top-level Floorplan
Compact-2D Design Top-level Buffer Insertion
Top-level Tier Partitioning
Top-level Inter-block Routing

FIGURE 12. **Top-level timing closure in pin-in-the-area flow. The individual blocks are treated as hard macros during top-level implementation.**

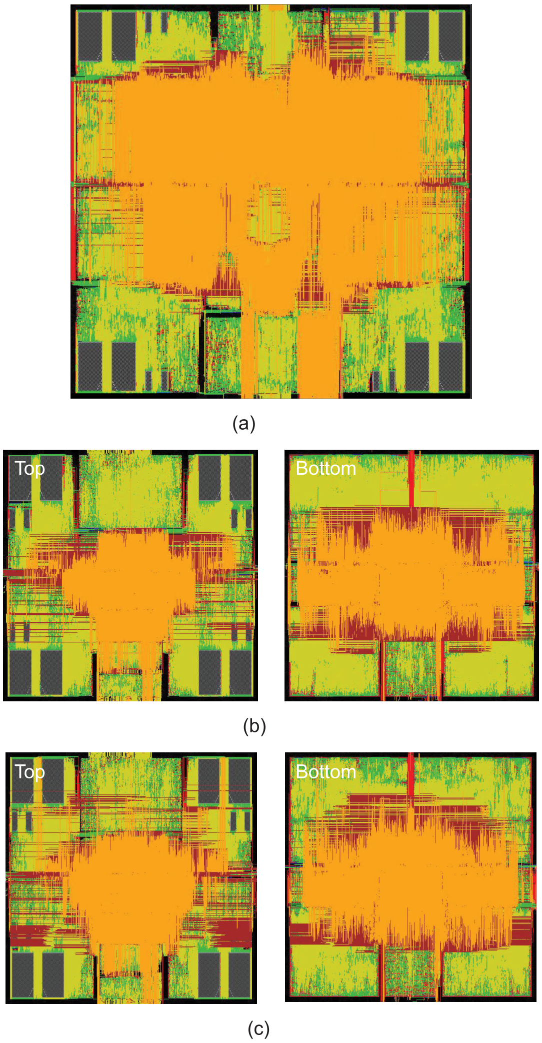

FIGURE 13. **Assembled design layouts of 28-nm RISC-V dual-core rocket processors. The footprint of the 2-D IC is 1.82 and 0.91 mm^2 for the footprint of two-tier M3D ICs. (a) 2D IC. (b) M3D boundary: block pins at the boundary. (c) M3D area: block pins inside the boundaries of blocks.**

designs, respectively, in Table 3. Top-level standard cells are top-level timing buffers, and block-level standard cells are the total sum of cells from individual blocks. Compared to 2-D, we observe that the M3D boundary and M3D area design reduces the total cell count by 4.2% and 9.6%, respectively.

TABLE 3. **Number and the area of standard cells for 2-D, M3D boundary, and M3D area designs.**

Cell Type	2D	M3D Boundary	M3D Area
	Std. Cell Count (#)		
Block-level Std. Cell	372,867	367,880 (-1.3%)	346,258 (-7.1%)
Top-level Std. Cell	22,263	10,674 (-52.1%)	10,940 (-50.9%)
Total Std. Cell	395,146	378,554 (-4.2%)	357,198 (-9.6%)
	Std. Cell Area (mm^2)		
Block-level Std. Cell	1.102	1.096 (-0.6%)	1.054 (-4.4%)
Top-level Std. Cell	0.051	0.021 (-59.4%)	0.021 (-58.0%)
Total Std. Cell	1.153	1.116 (-3.2%)	1.076 (-6.7%)
Place. utilization (%)	70.4	68.3 (66.9/69.7)	66.1 (65.4/66.7)

The 2-D baseline design contains 5.6% of the total standard cells as top-level timing buffers. While both M3D designs reduce the top-level buffers by around 50% due to the reduced top-level interblock interconnections, we observe that M3D area design further decreases the buffer count at an individual block implementation due to the increased timing margin at block boundaries. Area saving is a good metric that helps us to understand the combinational impact of cell count and drive strength reduction. This proves that buffers are reduced, but replaced by the larger buffer for the block-level cells, while both the number and drive strength of top-level cells are reduced.

C. ROUTING CONGESTION

Before block/top-level timing closure, we observe that M3D boundary and M3D area designs, indeed, achieve 10.4% and 12.1% wirelength savings compared to 2-D, respectively. Because individual blocks are not implemented yet, the impact of intrablock net wirelength can be excluded, and it allows us to capture the clear benefits of M3D integration to the interblock nets. However, all these wirelength savings become small after we actually implement blocks and perform top-level timing closure. To analyze the loss of wirelength savings, we check the routing congestions added by block and top implementation.

Table 4 shows the change of wirelength per metal layer caused by the block implementation and the top-level timing closure. For 2-D IC, a huge wirelength increase is observed at the M3 layer. Since intrablock nets reserve up to the M4 layer inside each block, this increase is mainly originated from individual block optimization. However, in M3D

TABLE 4. Wirelength changes per metal layer caused by block implementation and top-level timing closure.

Total WL (m)	2D	M3D Boundary	M3D Area
Before Timing Closure	10.62	9.52 (-10.4%)	9.34 (-12.1%)
After Timing Closure	11.26	10.95 (-2.8%)	10.67 (-5.2%)
ΔWL (m)	0.64	1.43	1.33
ΔWL Per Metal Layer (m)			
M1B	-0.02	-0.09	-0.10
M2B	0.13	-0.15	-0.27
M3B	**0.92**	-0.26	-0.16
M4B	0.08	-0.27	-0.32
M5B	-0.49	**0.79**	**0.68**
M6B	0.02	**0.47**	**0.42**
M1T		-0.03	-0.03
M2T		**0.32**	**0.37**
M3T		**0.33**	**0.35**
M4T		0.09	0.04
M5T		0.13	0.20
M6T		0.10	0.15

TABLE 5. Capacitance analysis for 2-D, M3D boundary, and M3D area designs.

Cap Type	2D	M3D Boundary	M3D Area
Total Wire Cap. (nF)	0.91	0.88 (-3.1%)	0.85 (-6.3%)
Total Pin Cap. (nF)	0.96	0.90 (-6.7%)	0.86 (-10.2%)
Total Cap. (nF)	1.87	1.78 (-5.0%)	1.71 (-8.3%)

TABLE 6. Iso-performance static power comparison at the maximum frequency of 2-D IC (538 MHz).

Static Power (mW)			
Power Type	2D	M3D Boundary	M3D Area
Sequential	47.47	46.72 (-1.6%)	46.56 (-1.9%)
Memory	26.60	25.75 (-3.2%)	25.77 (-3.1%)
Combinational	104.40	97.29 (-6.8%)	90.34 (-13.5%)
Clock	68.19	64.52 (-5.4%)	63.50 (-6.9%)
Internal	79.40	76.34 (-3.9%)	76.25 (-4.0%)
Switching	125.20	124.60 (-0.5%)	118.20 (-5.6%)
Leakage	42.04	33.37 (-20.6%)	31.73 (-24.5%)
Total Power	246.64	234.31 (-5.0%)	226.18 (-8.3%)

TABLE 7. Max-performance cross comparison. M3D area improves the energy–delay–product by 13.6% and 24.7% compared to 2-D and M3D boundary designs, respectively.

	2D	M3D Boundary	M3D Area
Clock Frequency (MHz)	538.0	585.0 (+8.7%)	640.0 (+19.0%)
Total Power (mW)	246.6	251.9 (+2.1%)	263.1 (+6.7%)
Total Energy ($pJ/cycle$)	458.3	430.5 (-6.1%)	410.9 (-10.3%)
Normalized EDP	1.00	0.86 (-13.6%)	0.75 (-24.7%)

IC designs, the top-level timing buffer insertion is a major attribute causing routing congestion. This is because timing buffers that are placed on the top tier actually occupy the empty spaces between top tier blocks. As a result, they make the 3-D interblock nets longer to detour them, which results in significant wirelength increases from M5B to M3T layers. In the case of M3D boundary design, this congestion becomes worse at M4B and M5B layers since every 3-D interblock nets have to detour the top-level cells. On the other hand, the M3D area design shows better 3-D routing overhead by enabling direct pin access to the top tier blocks at the M1T layer.

D. POWER SAVING AT ISO-PERFORMANCE

As shown in Table 5, a major factor to decrease the total capacitance turns out to be the pin capacitance reduction by cell count and drive strength savings at the individual blocks. Although we observe the wire capacitance reduction, the top-level routing congestion is found a bottleneck to preserve the wirelength savings in M3D ICs. Table 6 shows the static power metrics of 2-D, M3D boundary, and M3D area designs at 538 MHz, which is the maximum frequency of 2-D design; 0.1 of switching activity is used for primary inputs and sequential elements, and 2.0 for clock ports. Note that cell count reduction leads to a great combinational power decrease. M3D boundary design shows 5.0% of the total power saving with 3.9% and 20.6% savings in terms of internal and leakage powers, respectively. In the case of M3D area design, 10.2% and 6.3% pin and wire capacitance reductions give us 8.3% of total power saving compared to 2-D counterpart based on the great switching and leakage power decreases.

E. ENERGY-DELAY-PRODUCT SAVING AT MAXIMUM PERFORMANCE

For the cross comparison of 2-D and M3D designs at their own maximum frequencies, we tabulate the energy–delay–product metric in Table 7. We first observe that the M3D boundary and M3D area design achieves 8.7% and 19.0% faster clock frequency compared to 2-D. Total power and energy values are calculated at these maximum frequencies, and finally, we observe that the M3D area improves the energy–delay–product by 13.6% and 24.7% compared to 2-D and M3D boundary designs, respectively, and these improvements represent the benefit of pin-in-the-area flow.

V. CONCLUSION

In this article, we presented a physical design solution named pin-in-the-area for efficient block implementation and integration in M3D hierarchical designs. Our pin-in-the-area flow enables block pin assignment at any location inside the boundary of block regions for the direct vertical block communications in M3D ICs. We also proposed iterative wirelength-driven 3-D placement to co-optimize the block-level placement and pin locations. With 28-nm RISC-V dual-core rocket processor designs, we observed that the direct vertical block communication offers a larger timing margin for individual blocks and allows efficient block implementation resulting in a 9.6% total cell count and 10.2% pin capacitance reduction. Finally, we achieved 19.7% faster maximum clock frequency and 24.7% better energy-delay efficiency in block-level M3D design using pin-in-the-area flow than those of the conventional 2-D hierarchical design.

REFERENCES

[1] F. X. Che, L. Xie, Z. H. Chen, and S. Wickramanayaka, "Study on warpage and stress of TSV wafer with ultra-fine pitch vias for high density chip stacking," in *Proc. IEEE 19th Electron. Packag. Technol. Conf. (EPTC)*, Dec. 2017, pp. 1–7.

[2] A. Podpod *et al.*, "A novel fan-out concept for ultra-high chip-to-chip interconnect density with 20-μm pitch," in *Proc. IEEE 68th Electron. Compon. Technol. Conf. (ECTC)*, May 2018, pp. 370–378.

[3] E. Beyne, "The 3-D interconnect technology landscape," *IEEE Des. Test*, vol. 33, no. 3, pp. 8–20, Jun. 2016.

[4] H.-S.-P. Wong, "The end of the road for 2D scaling of silicon CMOS and the future of device technology," in *Proc. 76th Device Res. Conf. (DRC)*, Jun. 2018, pp. 1–2.

[5] S. Panth, K. Samadi, Y. Du, and S. K. Lim, "Power-performance study of block-level monolithic 3D-ICs considering inter-tier performance variations," in *Proc. 51st ACM/EDAC/IEEE Design Automat. Conf. (DAC)*, Jun. 2014, pp. 1–6.

[6] S. Panth, K. Samadi, Y. Du, and S. K. Lim, "High-density integration of functional modules using monolithic 3D-IC technology," in *Proc. 18th Asia South Pacific Design Automat. Conf. (ASP-DAC)*, Jan. 2013, pp. 681–686.

[7] B. W. Ku and S. K. Lim, "Pin-in-the-middle: An efficient block pin assignment methodology for block-level monolithic 3D ICs," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design*, New York, NY, USA, Aug. 2020, pp. 85–90, doi: 10.1145/3370748.3406580.

[8] W. Zhong, S. Chen, Y. Geng, and T. Yoshimura, "Lagrangian relaxation based pin assignment and through-silicon via planning for 3-D SoCs," in *Proc. IEEE 10th Int. Conf. ASIC*, Oct. 2013, pp. 1–4.

[9] Q. Hu and M.-S. Zhang, "A collaborative optimization for floorplanning and pin assignment of 3D ICs based on GA-SA algorithm," in *Proc. IEEE Int. Symp. Electromagn. Compat. Signal/Power Integrity (EMCSI)*, Jul. 2020, pp. 434–438.

[10] A. Waterman, Y. Lee, D. A. Patterson, and K. Asanovi, "The RISC-V instruction set manual. Volume 1: User-level ISA, version 2.0," Dept. Elect. Eng. Comput. Sci., California Univ. Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2014-54, 2014.

[11] Y. Lee *et al.*, "A 45 nm 1.3 GHz 16.7 double-precision GFLOPS/W RISC-V processor with vector accelerators," in *Proc. 40th Eur. Solid State Circuits Conf. (ESSCIRC)*, Sep. 2014, pp. 199–202.

[12] K. Asanovic *et al.*, "The rocket chip generator," EECS Dept., Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2016-17, 2016.

[13] B. W. Ku, K. Chang, and S. K. Lim, "Compact-2D: A physical design methodology to build commercial-quality face-to-face-bonded 3D ICs," in *Proc. Int. Symp. Phys. Design*, Mar. 2018, pp. 90–97.

[14] S. A. Panth, K. Samadi, Y. Du, and S. K. Lim, "Design and CAD methodologies for low power gate-level monolithic 3D ICs," in *Proc. Int. Symp. Low Power Electron. Design*, Aug. 2014, pp. 171–176.