Pin-in-the-Middle: An Efficient Block Pin Assignment Methodology for Block-level Monolithic 3D ICs

Bon Woong Ku[†] and Sung Kyu Lim[§]
[†]Synopsys Inc., Mountain View, CA
[§]School of ECE, Georgia Institute of Technology, Atlanta, GA
bon.ku@synopsys.com,limsk@ece.gatech.edu

ABSTRACT

In a 2D design, the periphery of a block serves as the optimal pin location since blocks are placed aside horizontally in a single placement layer. However, Monolithic 3D (M3D) integration relieves this boundary constraint by allowing vertical block communication between different tiers based on a nm-scale 3D interconnection pitch. In this paper, we present a design methodology named Pin-in-the-Middle that assigns block pins in the middle of a block using commercial 2D P&R tools to enable efficient block implementation and integration for two-tier block-level M3D ICs. Based on a 28nm 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 2D counterpart, respectively.

CCS CONCEPTS

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

ACM Reference Format:
Bon Woong Ku[†] and Sung Kyu Lim[§]. 2020. Pin-in-the-Middle: An Efficient Block Pin Assignment Methodology for Block-level Monolithic 3D ICs. In *ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED '20), August 10–12, 2020, Boston, MA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3370748.3406580

1 INTRODUCTION

The sequential face-to-back wafer-level 3D integration - also known as Monolithic 3D (M3D) - is an emerging 3D integration technology that enables the monolithic fabrication of vertically stacked multiple active layers [8]. In block-level M3D 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 3D floorplanning engines and presented promising power-performance-area savings. However, these works have assumed that the block pins are assigned along the periphery of each block. As shown in Figure 1, it is obvious that the periphery of a block is not always an optimal location for the pin assignment when blocks are vertically stacked. This is because inter-block connections are unnecessarily lengthened to touch the boundary pins. This motivates us to develop an efficient block-level M3D IC design methodology named Pin-in-the-Middle that tackles this new type of pin assignment problem. The tight 3D interconnection pitch achieved by M3D technology allows to accommodate optimal pin locations in the middle of a block. Taking advantage of it, we show that redundant detouring for the inter-block connections is minimized and this helps improve the design quality with better energy-delay-product.

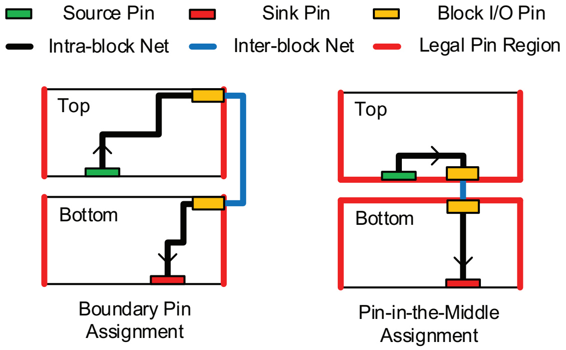

Figure 1: Suppose two blocks are placed on the top and the bottom tier each, and one completely overlaps another. With block pins along the periphery of each block, wirelength are unnecessarily lengthened to touch the boundary pins. Pin-in-the-Middle enables pin assignment in the middle of a block, resulting in efficient inter-/intra-block net connections.

2 PIN-IN-THE-MIDDLE FLOW

2.1 Overview

Pin-in-the-Middle assigns the optimal block pin locations in the middle of a block using commercial 2D P&R engines to build high-quality two-tier block-level M3D ICs. The flow begins with netlist restructuring to define optimal blocks for a two-tier M3D design. Next, tier-by-tier chip planning follows where we decide the shape and the 3D location of a block. Once the floorplanning is done, we perform wirelength-driven 3D placement which co-optimizes the block-level placement and pin locations (in the middle of a block) iteratively to improve the wirelength of inter-/intra-block nets. Next, we proceed timing budgeting where the wirelength saving turns into the additional block-level timing margin. This allows P&R tools to remove or downsize the redundant buffers during block

ISLPED '20, August 10–12, 2020, Boston, MA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7053-0/20/08…$15.00
https://doi.org/10.1145/3370748.3406580

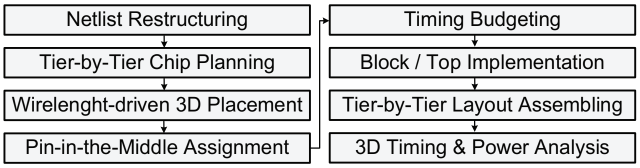
Netlist Restructuring
Tier-by-Tier Chip Planning
Wirelenght-driven 3D Placement
Pin-in-the-Middle Assignment
Timing Budgeting
Block / Top Implementation
Tier-by-Tier Layout Assembling
3D Timing & Power Analysis

Figure 2: An overview of our Pin-in-the-Middle flow.

Table 1: 17 Blocks of 28nm RISC-V dual-core Rocket processor [1, 3, 7] 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 Interface	0.015
	SLModule_7	MemCon-AXI4 Interface	0.051
	SLModule_2	MMIO-AXI4 Interface	0.078
	TLBroadcast	System Bus	0.054
	TLDebugModule	Debuging	0.014
	TopSubModule	Mixed	0.035

implementation, leading to the reduced total power consumption. After block implementation and top-level timing closure, we finally perform sign-off 3D timing & power analysis using assembled tier-by-tier layouts. An overview of Pin-in-the-Middle flow is depicted in Figure 2.

2.2 Netlist Restructuring

In this work, we use commercial-grade 28nm process design kit (PDK) and build dual-core Rocket processor [1, 3], an open-source microprocessor that executes scalar RISC-V [7] instruction set architecture. The original register-transfer level (RTL) files of Rocket processor contain multi-depth block hierarchy. However, a block-level M3D IC makes 3D connections only at the top-level while an individual block remains as 2D. Therefore, we transform the netlist into two-level hierarchy (top-level & blocks) to make it suitable for the purpose of this work by iteratively ungrouping and merging tiny blocks whose standard cell area is under $10\mu m^2$. Note that the area threshold for ungrouping depends on the technology node and the design benchmark. Table 1 tabulates the block definitions as a result of the netlist restructuring process. Each core is divided into pipeline logic, floating point unit, I/D-Caches, and interface buffer blocks. Including bus units and memory controller units, total 17 blocks survive.

2.3 Tier-by-Tier Chip Planning

After blocks are defined by netlist restructuring, we synthesize the netlists and load them to a floorplanning engine. Given that we build a two-tier M3D IC, the form factor (footprint) of our M3D IC is as 50% small as its 2D IC counterpart assuming no silicon area overhead. When we decide the size and (X,Y,Z) location of a 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 this work, we place I/D-caches on the top tier and core logics on the bottom tier realizing memory-on-logic stacking principle, which targets efficient 3D logic-memory interconnects. Next, we decide the tier location of other logic blocks to balance the area skew on both tiers while optimizing the block interconnection.

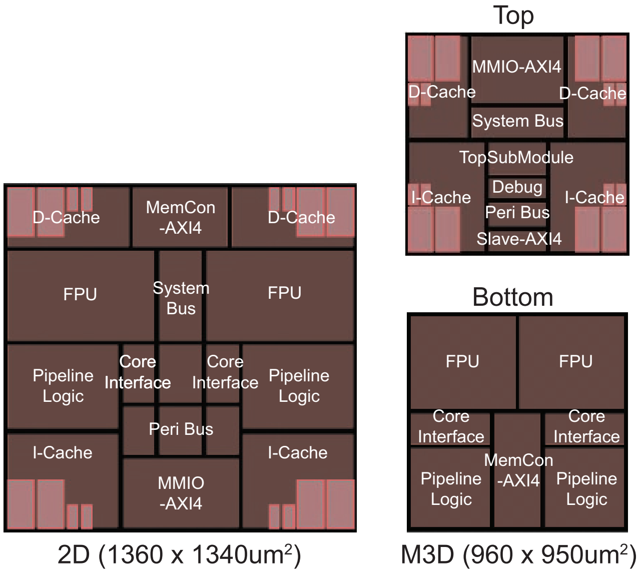

Figure 3: Chip planning results for 2D and M3D RISC-V dual-core Rocket processor designs, respectively. The footprint of M3D design is set to be the half of 2D design footprint assuming no silicon area overhead.

After that, we insert an additional hierarchy level representing a tier between the top-level and blocks in the synthesized netlist to organize it into the three-level hierarchy (top-level, tiers & blocks). Then we generate two netlists (tier netlist) 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 noteworthy that this netlist modification defines a 3D inter-block net as a top-level net connected to the I/O port of each tier. Figure 3 shows 2D and M3D RISC-V dual-core Rocket processor design floorplans, respectively. Out of 6903 inter-block nets, we achieve 3692 (53%) 3D inter-block nets as a result. Here, die-to-core and block-to-block spacing are $20\mu m$, and hard macro placement halo is $10\mu m$. The initial utilization of each block is assumed to be 65% ± 5% in both 2D and M3D designs.

2.4 Wirelength-driven 3D Placement

As shown in Figure 4, wirelength-driven 3D placement is an iterative tier-by-tier global placement method to produce an optimal 3D-aware placement solution. To make tier-by-tier global placement 3D-aware, both tiers need to keep synchronizing the target

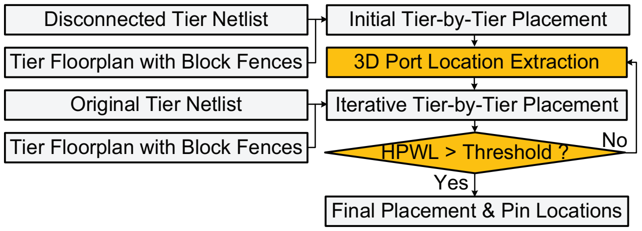
Disconnected Tier Netlist
Tier Floorplan with Block Fences
Original Tier Netlist
Tier Floorplan with Block Fences
Initial Tier-by-Tier Placement
3D Port Location Extraction
Iterative Tier-by-Tier Placement
HPWL > Threshold ?
No
Yes
Final Placement & Pin Locations

Figure 4: Proposed wirelength-driven 3D placement method. This iterative approach extracts the output pins of driver cells from each tier and uses them as anchors for 3D-aware tier-by-tier placement.

location of 3D ports in the middle of a tier die. Therefore, the basic idea of our wirelength-driven 3D placement solution is to iterate the exchange of 3D port information and tier-by-tier global placement. At the beginning, we modify the tier netlist by eliminating all the top-level primary I/O ports and inter-block connections (disconnected tier netlist) beforehand and run the global placement at each tier. This way, the initial tier placement does not have any bias against external block or I/O connections. Then we extract the output pin location of the driver cell of a 3D port (defined at the original tier netlist) to decide the location of the 3D port. Next, both tiers share the information of 3D port locations. Then we perform the tier-by-tier placement from scratch, but with the original tier netlist and extracted 3D port locations. Those 3D ports serve as anchors for cells connected to 3D inter-block nets and prevent the placement engine from over-optimizing 2D inter-block nets. Therefore, this anchoring process optimizes both 3D/2D inter-block connections.

With the RISC-V dual-core Rocket processor design, Figure 5 shows that the sum of half perimeter wirelength (HPWL) for all inter-block nets at the first iteration is reduced by 30.4% compared to the initial tier-by-tier placement solution, and it monotonically decreases as the iteration proceeds. On the other hand, we observe 5% HPWL overhead with small disturbance for intra-block 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

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

where $Min.HPWL$ is the minimum HPWL value out of the whole iterations. As the iteration proceeds, we update the 3D port locations based on the latest tier-by-tier placement solution. When the exit condition is met, we use the resulting tier-by-tier placement solution for 3D timing budgeting. In the Rocket processor design, we meet the exit condition at the 7th iteration as shown in Figure 6, and find the optimal tier-by-tier placement solution at the 6th iteration.

2.5 Pin Assignment and Timing Budgeting

After wirelength-driven 3D placement, we assign the block pin at the final location of the 3D port in the middle of a block. For this pin assignment, a special 3D P&R environment using full 3D metal stack information is required. 3D technology library exchange format

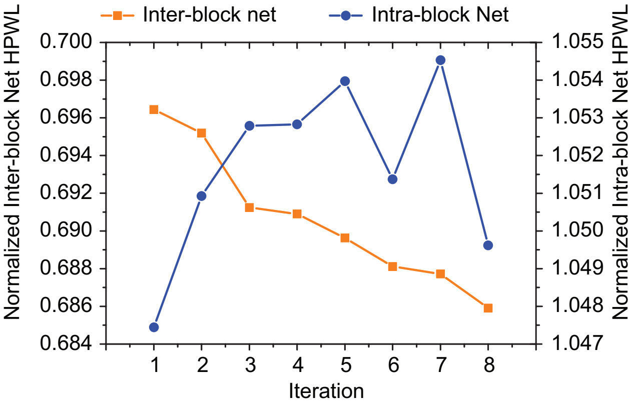

Figure 5: The half perimeter wirelength (HPWL) changes as the wirelength-driven 3D placement proceeds. We observe 30.4% HPWL reduction for inter-block nets at the very first iteration, and it decreases monotonically as iteration proceeds. For intra-block nets, small disturbance around 5% overhead is observed.

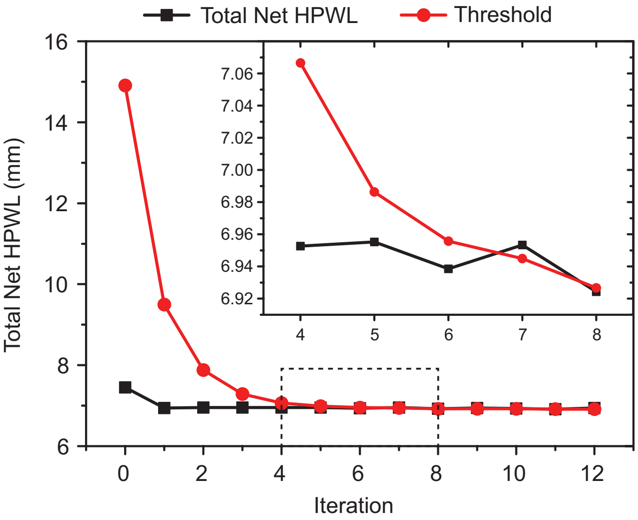

Figure 6: Changes in the total net HPWL and exponentially decreasing threshold as iteration proceeds. For the Rocket processor design, we meet the exit condition at the 7th iteration, and end up with 6.9% total HPWL savings.

(LEF) contains the layer definition used for the top and bottom tiers. 3D macro LEF defines the pin location of a standard cell based on its tier. The RC database for the 3D metal stack (3D TCH) is also needed for the timing budgeting later. These input files allow us to instantiate standard cells in a top-tier block with the top-tier macro LEF and cells in a bottom-tier block with bottom-tier macro LEF. Since we know the tier location of cells already, we update the macro of a cell in the original netlist based on its tier location. Then we load both tier-by-tier placement solutions on a single placement layer to accommodate all the synthesized cells

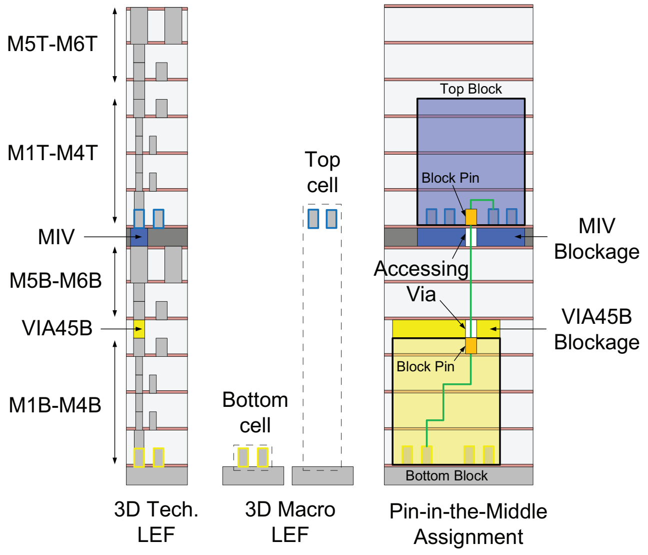

Figure 7: In this work, we reserve up to M4 layer for intra-block nets and use M5 and M6 layers for inter-block connections on both tiers in an M3D design.

of the design in the P&R tool. Note that 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 3D macro LEFs as shown in Figure 7. In case there are unplaced top-level cells, additional top-level placement can proceed at this point to implement a full 3D placement solution.

In this work, we utilize six metal layers for the 2D IC, and for both top and bottom tiers in the M3D IC. We reserve up to M4 layer for intra-block nets and use M5 and M6 layers for inter-block connections. Based on the full 3D placement, we complete the assignment for a block pin connected to a 3D inter-block net (3D pin) first. 3D pins for a bottom-tier block are fixed at M4B layer in the middle of the block, and 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 3D pin location in the middle of a blockage layer. This prevents the routing engine from accessing the blocks through the middle of block regions unless there are 3D pins. Once detail routing is done honoring the routing blockage, remaining block pins (2D pin) are assigned at the periphery of a block automatically based on the inter-block routing information. Figure 8 shows the pin assignment result. Lastly, 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.

2.6 Top-level Timing Closure

For top-level timing closure, we apply the state-of-the-art gate-level M3D flow [2] to our block-level M3D IC environment. First, the individual blocks are treated as hard macros and we flatten both tiers on the single placement layer. Then we expand the flattened top-level floorplan upto the 2D IC footprint and replace hard macros with placement blockages. When macros are fully overlapped, full placement blockages are created. Otherwise, partial placement blockages (50% allowance) are created to reflect the empty spaces in non-overlapped region. Note that placement blockage regions also should be expanded by the same expansion factor. Based on those blockage regions, 2D P&R engines identify the legal buffer placement locations. Next we perform the conventional 2D 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 3D 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, $1/\sqrt{2} = 0.707$ is used for the parasitic scaling & placement contraction factor.

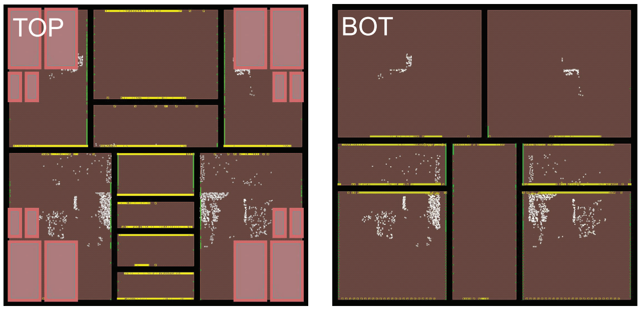

Figure 8: The pin assignment result. 3D pins in the middle of blocks are colored in white. 2D pins are still assigned at the block boundaries if needed. The size of pins are magnified by 50x for visualization.

To determine Z location of top-level timing buffers, we use bin-based placement-driven Fiduccia-Mettheyses (FM)-mincut partitioning algorithm [4]. For the top-level 3D connection, 3D routing should proceed first, and MIV locations are decided based on the routing result of 3D inter-block nets. Then, in-house tool generates the new RTLs 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. Figure 9 shows assembled design layouts of 28nm RISC-V dual-core Rocket processors for the 2D IC, for the M3D IC with block pins at the boundary of blocks (M3D Boundary), and for the M3D IC with block pins in the middle of blocks (M3D Middle), respectively. The footprint of 2D IC is $1.82mm^2$ and $0.91mm^2$ (-50%) for that of M3D ICs, so there is no silicon area overhead to build two-tier M3D designs.

3 EXPERIMENTAL RESULTS

3.1 Wirelength Saving

Table 2 summarizes the wirelength of nets based on their connection types. The total wirelength savings of M3D Boundary and M3D Middle designs are 2.8% and 5.2%, respectively. It is worth noting that the wirelength saving on intra-block nets are the major source

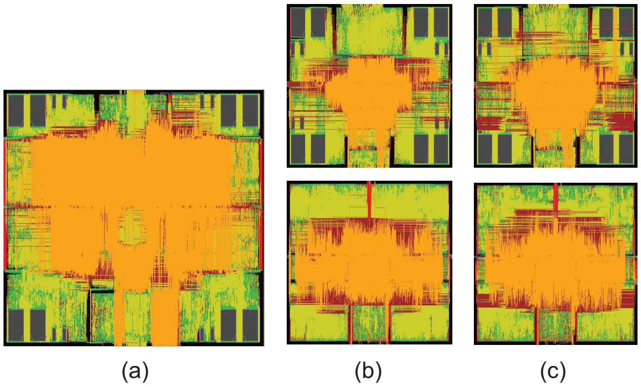

Figure 9: Assembled design layouts of 28nm RISC-V dual-core Rocket processors for (a) 2D IC (b) M3D IC with block pins at the boundary (M3D Boundary) (c) M3D IC with block pins in the middle of blocks (M3D Middle). The footprint of the 2D IC is 1.82mm^2 and 0.91mm^2 for the footprint of two-tier M3D ICs.

Table 2: The wirelength of nets based on their connection types. % values for both M3D designs are based on the comparison with the 2D design.

Net Type	2D	M3D Boundary	M3D Middle
	Wirelength (m)		
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%)

of total wirelength saving. Given that the wirelength of intra-block nets contributes to 75.3% of the total wirelength in the 2D 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 inter-block nets directly. Table 2 also shows that our Pin-in-the-Middle flow enables further optimized block implementation based on the better timing margin than that of a block with boundary pins.

3.2 Cell Count & Area Savings

Since the area of memory modules (0.129mm^2) is kept the same in all designs, we tabulate the number and area of standard cells for 2D, M3D Boundary, and M3D Middle designs, respectively. In Table 3, Top-level Std. Cells are top-level timing buffers and Block-level Std. Cells are the total sum of cells from individual blocks. Compared to 2D, we observe that M3D Boundary and M3D Middle design reduces the total cell count by 4.2% and 9.6%, respectively. The 2D 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 inter-block interconnections, we observe that M3D Middle design further decreases the buffer count at an individual block implementation thanks 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 larger buffer for the block-level cells, while both number and drive strength of top-level cells are reduced.

Table 3: The number and the area of standard cells for 2D, M3D Boundary, and M3D Middle designs.

Cell Type	2D	M3D Boundary	M3D Middle
	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%)

3.3 Routing Congestion

Before block/top-level timing closure, we observe that M3D Boundary and M3D Middle designs indeed achieve 10.4% and 12.1% wirelength savings compared to 2D, respectively. Because individual blocks are not implemented yet, the impact of intra-block net wirelength can be excluded and it allows us to capture the clear benefits of M3D integration to the inter-block 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 block implementation and top-level timing closure. For 2D IC, huge wirelength increase is observed at M3 layer. Since intra-block nets reserve up to M4 layer inside each block, this increase is mainly attributed from individual block optimization. However, for M3D ICs, the top-level timing buffer insertion is found as a major attribute causing routing congestion. This is because timing buffers placed on the top tier actually occupy the empty spaces between top tier blocks. As a result, they makes the 3D inter-block nets longer to detour them, which results in the significant wirelength increases from M4B to M3T layers. In case of M3D Boundary design, this congestion becomes worse at M3B and M4B layer since every 3D inter-block nets have to detour the top-level cells. On the other hand, M3D Middle design shows better 3D routing overhead by enabling direct pin access to the top tier blocks at M1T layer.

3.4 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 compares the static power metrics of 2D, M3D Boundary, and M3D Middle designs at the maximum frequency of 2D IC (538MHz). 0.1 switching activity is used for primary inputs and sequential elements, and 2.0 for clock ports. Note that cell count reduction leads to great combinational power decrease,

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

Total WL (m)	2D	M3D Boundary	M3D Middle
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 2D, M3D Boundary, and M3D Middle designs.

Cap Type	2D	M3D Boundary	M3D Middle
Capacitance (nF)			
Total Wire Cap	0.909	0.881 (-3.1%)	0.852 (-6.3%)
Total Pin Cap	0.959	0.895 (-6.7%)	0.861 (-10.2%)
Total Cap	1.868	1.776 (-5.0%)	1.713 (-8.3%)

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

Power Type	2D	M3D Boundary	M3D Middle
Static Power (mW)			
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%)

and our Pin-in-the-Middle flow results in 8.3% total power saving compared to 2D design based on the great switching and leakage power decreases.

3.5 Energy-Delay-Product Saving at Maximum Performance

For the cross comparison of 2D 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 Middle design achieves 8.75% and 19.02% faster clock frequency compared to 2D. Total power and energy values are calculated at these maximum frequencies, and finally, we observe that M3D Middle improves the energy-delay-product by 13.6% and 24.7% compared to 2D and M3D Boundary designs, respectively.

Table 7: Max-performance cross comparison. M3D Middle improves the energy-delay-product by 13.6% and 24.7% compared to 2D and M3D Boundary designs, respectively.

	2D	M3D Boundary	M3D Middle
Clock Frequency (MHz)	538.0	585.0 (+8.75%)	640.0 (+19.72%)
Total Power (mW)	246.6	251.9 (+2.15%)	263.1 (+6.69%)
Total Energy ($pJ/cycle$)	458.3	430.5 (-6.07%)	410.9 (-10.36%)
Normalized EDP	1.00	0.86 (-13.6%)	0.75 (-24.7%)

4 CONCLUSION

In this paper, we presented a physical design solution named Pin-in-the-Middle for efficient block implementation and integration in Monolithic 3D (M3D) hierarchical designs. Our Pin-in-the-Middle flow enables block pin assignment in the middle of block regions for the direct vertical block communications in M3D ICs. We also proposed iterative wirelength-driven 3D placement to co-optimize the block-level placement and pin locations. With 28nm RISC-V dual-core Rocket processor designs, we observed that the direct vertical block communication offers larger timing margin for individual blocks, and allows efficient block implementation resulting in 9.6% total cell count and 10.2% pin capacitance reduction. Finally, we achieved 19.7% faster maximum clock frequency, and 24.68% better energy-delay efficiency in block-level M3D design using Pin-in-the-Middle flow than those of conventional 2D hierarchical design.

ACKNOWLEDGEMENTS

This research is funded by the DARPA ERI 3DSOC Program under Award HR001118C0096.

REFERENCES

[1] K. Asanovic et al. The rocket chip generator. *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2016-17*, 2016.

[2] B. W. Ku et al. Compact-2D: A Physical Design Methodology to Build Commercial-Quality Face-to-Face-Bonded 3D ICs. In *Proceedings of the 2018 International Symposium on Physical Design*, ISPD '18, pages 90–97, 2018.

[3] Y. Lee et al. A 45nm 1.3 GHz 16.7 double-precision GFLOPS/W RISC-V processor with vector accelerators. In *European Solid State Circuits Conference (ESSCIRC), ESSCIRC 2014-40th*, pages 199–202. IEEE, 2014.

[4] S. Panth et al. Design and CAD methodologies for low power gate-level monolithic 3D ICs. In *Proc. Int. Symp. on Low Power Electronics and Design*, pages 171–176, 2014.

[5] S. Panth, K. Samadi, Y. Du, and S. K. Lim. High-density integration of functional modules using monolithic 3D-IC technology. In *2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 681–686. IEEE, 2013.

[6] 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 *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2014.

[7] A. Waterman, Y. Lee, D. A. Patterson, and K. Asanovi. The RISC-V Instruction Set Manual. Volume 1: User-Level ISA, Version 2.0. Technical report, CALIFORNIA UNIV BERKELEY DEPT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCES, 2014.

[8] H. . P. Wong. The End of the Road for 2D Scaling of Silicon CMOS and the Future of Device Technology. In *2018 76th Device Research Conference (DRC)*, pages 1–2, June 2018.