Cascade2D: A Design-Aware Partitioning Approach to Monolithic 3D IC with 2D Commercial Tools

Kyungwook Chang[1], Saurabh Sinha[2], Brian Cline[2], Raney Southerland[2],
Michael Doherty[2], Greg Yeric[2] and Sung Kyu Lim[1]
[1]School of ECE, Georgia Institute of Technology, Atlanta, GA
[2]ARM Inc., Austin, TX
k.chang@gatech.edu, limsk@ece.gatech.edu

ABSTRACT

Monolithic 3D IC (M3D) can continue to improve power, performance, area and cost beyond traditional Moore's law scaling limitations by leveraging the third-dimension and fine-grained monolithic inter-tier vias (MIVs). Several recent studies present methodologies to implement M3D designs, but most, if not all of these studies implement top and bottom tier separately after partitioning, which results in inaccurate buffer insertion. In this paper, we present a new methodology called 'Cascade2D' that utilizes design and micro-architecture insight to partition and implement an M3D design using 2D commercial tools. By modeling MIVs with sets of anchor cells and dummy wires, we implement and optimize both top and bottom tier simultaneously in a single 2D design. M3D designs of a commercial, in-order, 32-bit application processor at the foundry 28nm, 14/16nm and predictive 7nm technology nodes are implemented using this new methodology and we investigate the power, performance and area improvements over 2D designs. Our new methodology consistently outperforms the state-of-the-art M3D design flow with up to 4X better power savings. In the best case scenario, M3D designs from the Cascade2D flow show 25% better performance at iso-power and 20% lower power at iso-performance.

1. INTRODUCTION

As 2D scaling faces limitations due to the physical limits of channel length scaling, lithography limitation and increased parasitics and costs, monolithic 3D IC (M3D) has emerged as a promising solution to extend Moore's Law. Unlike through-silicon via (TSV)-based 3D ICs which bond fabricated dies using TSVs, in M3D ICs, fabrication is processed sequentially across two tiers. Compared to TSV-based 3D ICs, the sequential fabrication allows two tiers to have very fine grained connections using fine-pitched monolithic inter-tier vias (MIVs), which connect the last metal layer on bottom tier and the first metal layer on top tier. Owing to the small size and parasitics of MIVs, and recent research on manufacturing technology involving higher alignment precision and the ability to process thinner dies, We can harness true benefit of M3D ICs with fine grained vertical integration.

ICCAD '16, November 07-10, 2016, Austin, TX, USA
 ISBN 978-1-4503-4466-1/16/11…$15.00
DOI: http://dx.doi.org/10.1145/2966986.2967013

In M3D ICs, standard cells and hard macros are partitioned into two tiers, and MIVs are used for inter-cell connections. Using MIVs, we reduce wire-length by utilizing short vertical connections instead of using long wires in 2D space. M3D ICs also save standard cell area because lower number of buffers and lower drive-strength cells are needed to drive the reduced wire load. Power saving in M3D ICs are attributed to the reduced wire-length and buffer area.

Currently EDA tools do not support M3D designs and hence, previous studies have explored implementation approaches of M3D ICs using 2D commercial tools. In [1], in order to estimate cell placement and wire-length of a M3D design, the dimensions of cells and wires are shrunk, and a 'Shrunk2D' design is implemented in half area of the 2D design. However, using Shrunk2D design is prone to inaccurate buffer insertion because of inaccurate wire-load estimation. Moreover, the flow is completely design-agnostic, utilizes very large number of MIVs and hence partitions local cells into separate tiers resulting in a non-optimal 3D partition. Another M3D design methodology is proposed in [2], which folds 2D placement at the center of the die into two separate tiers. However, using their flow shows marginal wire-length savings, no power savings and does not take into account design details to guide partitioning resulting in a non-optimal solution.

In order to relieve worsening electrostatics associated with scaling planar transistors, the industry transitioned to 3D FinFETs. However, FinFETs have higher parasitic capacitance owing to their 3D structure and the introduction of local interconnects to contact the transistors. Therefore, to reduce power consumption in FinFET based nodes, it is crucial to reduce standard cell area effectively in addition to wire-length savings. Figure 1 shows the 'Cut-and-Slide' methodology of the Cascade2D flow with sets of anchor cells and dummy wires. As can be clearly seen, the anchor cells and dummy wires model the monolithic inter-tier vias (MIVs) and the Cascade2D implementation Figure 1 (a) is functionally equivalent to the M3D design in Figure 1 (b).

The main contributions of this work are as follows: 1) we present a novel M3D implementation methodology that incorporates design and micro-architecture insight to guide the partitioning scheme; 2) our methodology is partition-scheme agnostic and hence, making it an ideal platform to evaluate different partitioning schemes; 3) it effectively reduces standard cell area as well as wire-length compared to 2D designs, resulting in significant power saving; and 4) the proposed Cascade2D flow shows better power saving compared to state-of-art M3D implementation methodology.

2. IMPLEMENTATION METHODOLOGY

This section presents our RTL-to-GDSII design methodology, Cascade2D flow, to implement sign-off quality M3D ICs. Inputs

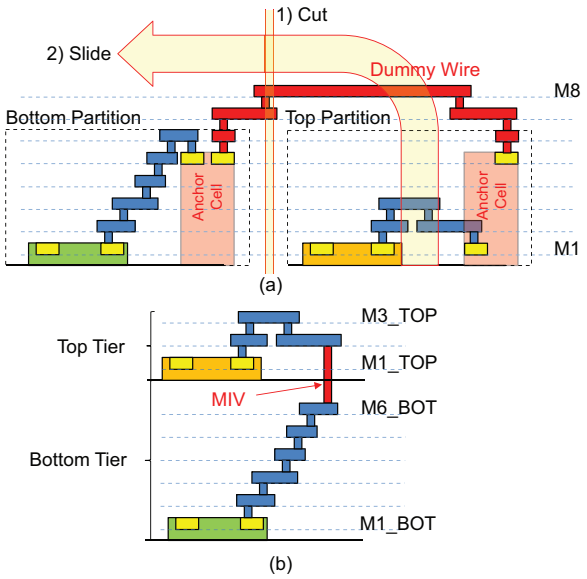

Figure 1: Monolithic 3D IC implementation scheme of Cascade2D flow. a) Cascade2D implementation with a set of anchor cells and dummy wires, which models MIVs b) equivalent M3D IC

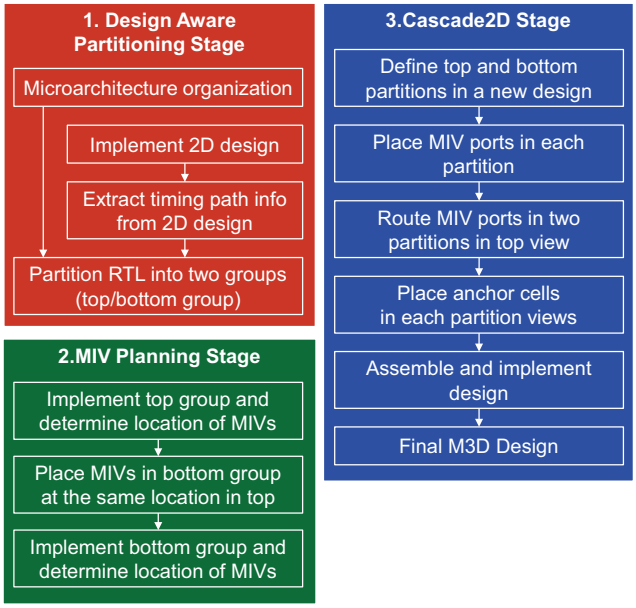

Figure 2: Flow diagram of the proposed methodology, Cascade2D flow

Table 1: Qualitative comparison of Cascade2D flow and state-of-the-art Shrunk2D flow

Cascade2D Flow	Shrunk2D Flow
• Can implement block and gate-level M3D • Capable of handling RTL-level constraints • Highly flexible; can implement any partitioning algorithm • Designer has complete control over tier-assignment of cells/blocks • Implements top and bottom tier in a single design • Buffer insertion based on actual technology parameters	• Can implement gate-level M3D only • Cannot handle RTL-level constraints • Implements min-cut algorithm for partitioning cells • Designer controls bin-size but not actual tier-assignment of gates • Implements top and bottom tier separately • Buffer insertion based on shrunk technology parameters

and outputs of the proposed method are as follows:

- Input: RTL of a design, design libraries, design constraints
- Output: GDSII layouts, timing/power analysis results

Table 1 presents a qualitative comparison of the Cascade2D flow with the state-of-the-art Shrunk2D flow for implementing M3D designs. Figure 2 shows the flow diagram of this methodology. First, functional blocks are partitioned into two groups, top and bottom group, creating signals crossing two groups, which become MIVs in M3D designs. Then, the location of MIVs are determined, and lastly, Cascade2D designs are implemented with sets of anchor cells and dummy wires in 2D space which is equivalent to the final M3D design.

2.1 Design-Aware Partitioning Stage

In this step, we partition RTL into two groups, top and bottom group, which represent top and bottom tier of the M3D design, respectively. The partition can be performed in two ways: 1) based on the organization of the design micro-architecture and 2) by extracting design information from 2D implementations.

Because M3D ICs offer vertical integration of cells, we can achieve power and performance improvement by placing inter-communicating functional modules separated by a large distance in the xy-plane in a 2D design, on separate tiers and reducing the distance in the z-plane in an M3D design. With a detailed understanding of the micro-architecture organization, functional modules can be pre-partitioned into separate tiers. For example, consider two functional modules whose connecting signals have a tight-timing budget (i.e. a data path unit and its register bank). Placing these modules into separate tiers and connecting them with MIVs can help reduce wire-length.

In case it is non-trivial to partition based on the understanding of micro-architectural organization, we can utilize design information from 2D implementation to help guide the partitioning process. By extracting timing paths from a 2D design, we can quantify the number of timing paths crossing each pair of functional modules. We call this number 'degree of connectivity' between functional modules. We also extract standard cell area of each functional module from the 2D design for cell area balancing between the tiers.

After obtaining the degree of connectivity of functional modules and their cell area, the design is partitioned into two groups based on the following criteria:

- Balance cell area of top and bottom group
- Maximize the number of timing paths crossing two groups

These criteria helps 1) the functional blocks, which have a very high degree of connectivity, to be placed into separate tiers and to minimize the distance between them and 2) to balance the standard cell area of the two tiers. Figure 3 shows an example of design-aware partitioning. Module A and B are fixed on two different groups based on organization of the design micro-architecture, module C, D, E, and F are partitioned maximizing the number of timing paths crossing two groups and balancing cell area of two groups. It should be emphasized, however, that the Cascade2D

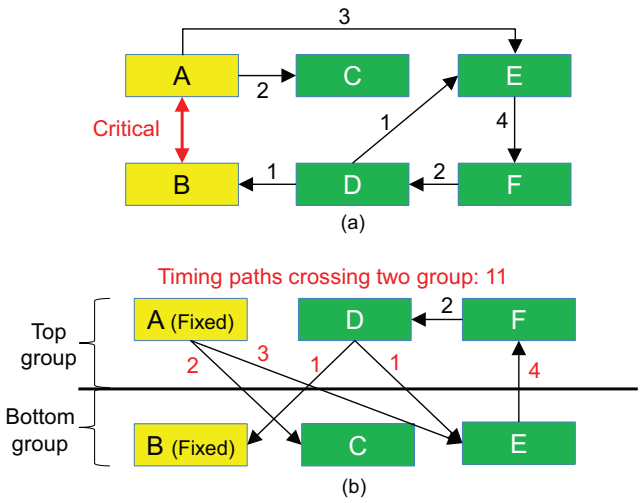

Figure 3: Example of our design-aware partitioning scheme a) Pre-partitioned modules (yellow box), and degree of connectivity (numbers on the arrows) of rest of modules (green box) b) Result of the design-aware partitioning

flow is extremely flexible, and can incorporate any number of constraints for partitioning cells or modules into separate tiers. Depending on the type of design, the designer may wish to employ different partitioning criteria than presented here and the subsequent steps (MIV Planning Stage and Cascade2D Stage) would remain the same. Hence, this flow is an ideal platform to evaluate different partitioning schemes for M3D designs.

At this stage, it is important to understand that there are two types of IO ports in our design. There are a set of IO ports that were created because of the 'design-aware partitioning' step. These IO ports connect the top and bottom groups of the design and they are referred as MIV ports in rest of the paper since they eventually become MIVs in M3D design. Additionally, we have a set of IO ports for the top-level pre-partitioned design. These are same as the conventional IO ports of the 2D design.

2.2 MIV Planning Stage

After partitioning the RTL into top and bottom groups, the location of the MIVs are determined. We first implement the top group, and place MIV ports above their driving or receiving cells on the top routing metal layer, so that wire-length between MIV ports and relevant cells are minimized. The MIV ports are placed over the standard cells, instead of the edge of the die, as would be done in a conventional 2D design. As explained in the previous sub-section, MIV ports are actually IO ports that connect the top and bottom groups. We leverage the fact that all cell placement algorithms in commercial EDA tools tend to place cells close to the IO ports to minimize timing. Hence, we implement the bottom group using the location of MIVs determined from the top group implementation. In this way, the cell placement of the top group guides the cell placement of the bottom group using the pre-fixed MIV ports.

We assume that the IO ports of the top-level design are connected only to the top tier in M3D designs. Hence it is possible that some IO signals need to be directly connected to functional modules in the bottom group. These 'feed-through' signals will not have any driving or receiving cells on the top group. Hence, the MIV ports for those signals cannot be placed with top group implementation and are determined during the bottom group implementation.

Figure 4 shows the location of MIVs after implementing the bottom group. After obtaining the location of complete set of MIVs, standard cell placement in top and bottom group implementation is discarded, and only MIV locations are retained.

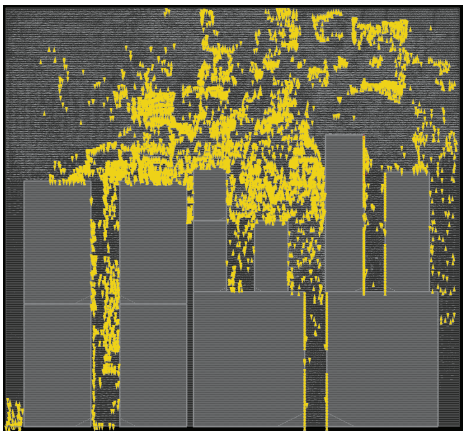

Figure 4: Location of MIVs (yellow dots) after completing MIV planning stage

2.3 Cascade2D Stage

In this step, we implement Cascade2D design, which models M3D design in a single 2D design with sets of anchor cells and dummy wires, using partitioning technique supported in Cadence® Innovus™.

We first create a new die with both tiers placed side-by-side, with the same total area as the original 2D design. We define top and bottom partitions in the die, and set a hard fence for placement, so that cells in the top partition are placed only on the top half of the die, and cells in the bottom partition only on the bottom half of the die. Then two hierarchies of the design are created as follows:

- 1st Level of Hierarchy: Top view, which contains only two cells, top-partition cell and bottom-partition cell. These two cells contain pins which represent MIVs for the top and bottom tier, respectively.
- 2nd Level of Hierarchy: Top partition-cell, which contains the top partition view where standard cells from the top group are placed and routed.
- 2nd Level of Hierarchy: Bottom partition-cell, which contains the bottom partition view where standard cells from the bottom group are placed and routed.

In the top view, we place pins, representing MIVs, in the top-partition cell and bottom-partition cell on the top routing metal layer (i.e. M6 in Figure 1). The pin locations are the same as the MIV location derived in Section 2.2. Figure 5 (a) shows placed pins for MIVs in the top view.

Then, using 3-4 additional metal layers above the top routing metal layer used in actual design, (i.e. M7-M8 in Figure 1), we route to connect the pins on the top-partition cell and bottom-partition cell. As the location of the pins are identical in the X-axis in top and bottom-partition cells, the routing tool creates long vertical wires crossing two partition cells. These additional 3-4 metal layers used to connect the pins of the top and bottom partitioning cells are called 'dummy wires' because their only function is to get logical connection between the two tiers in the physical design. The delay and parasitics associated with these wires will not be considered in the final M3D design.

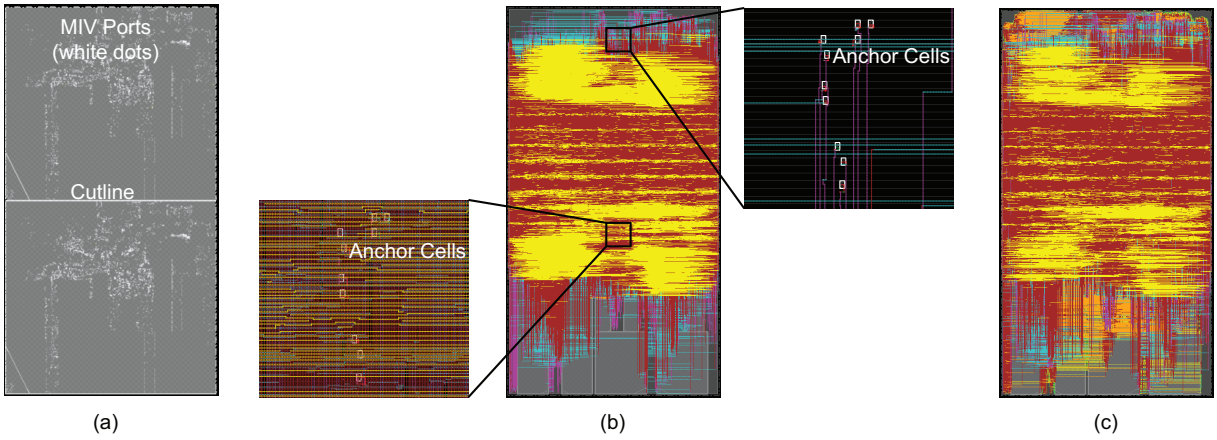

Figure 5: Die images in different steps in M3D implementation stage described in Section 2.3. a) top view after placing pins for MIVs, b) after assembling top view and top and bottom partition view, c) after implementing Cascade2D design

In an M3D design the last metal layer of the bottom tier is connected to the first metal layer of the top tier using an MIV. We wish to emulate this connectivity in a 2D design where the top and bottom tier are placed adjacent to each other. Hence, we need a mechanism to connect M1 in the top partition view with M6 in the bottom partition view. This is achieved through, what we call as 'anchor cells'. An anchor cell is a dummy cell which implements buffer logic. Anchor cells model zero-delay virtual connection between a dummy wire and one of the metal layers. After connecting the two partition cells with dummy wires, anchor cells are placed below the pins in each partition view. In this step, only anchor cells are placed but not logic cells.

Depending on the partition using anchor cells and metal layer to which a dummy wire needs to be virtually connected, three flavors of anchor cells exist: 1) top-tier-driving anchor cells (Figure 6 (a)), which are placed in the top partition, receiving signals from M1 of top partition, and driving a dummy wires, 2) top-tier-receiving anchor cells (Figure 6 (b)), which sends signal in the reverse direction, and 3) bottom-tier anchor cells (Figure 6 (c)), which are placed in the bottom partition, connecting a dummy wire to top metal layer of the bottom partition. After placement, anchor cells and the corresponding MIV ports are connected.

Next all hierarchies are flattened, i.e., top view and both partition views are assembled projecting all anchor cells in two partition views and dummy wires in top view into a single design. Figure 5 (b) shows the assembled design.

With the assembled design, we set the delay of dummy wires to zero, and anchor cells and dummy wires are set to be fixed, so that their location cannot be modified. These sets of anchor cells and dummy wires effectively act as 'wormholes' which connect M1 of the top partition and top routing metal layer of the bottom partition without delay emulating the behavior of MIVs (the MIV parasitics are added in the final timing stage).

Then we run regular P&R flow, which involves placement of logic cells in the design, CTS, post-CTS-hold, route, post-route, and post-route-hold. Owing to 1) 'wormholes', which provide virtual connection between M1 of the top partition and top routing metal layer of the bottom partition, and 2) the hard fence, which sets the boundary for top and bottom partition, the tool places each tier in its separate 2D partitioned space with virtual connections between them. At this stage, we call the resulting design 'Cascade2D'.

Clock tree synthesis (CTS) in Cascade2D flow is performed as

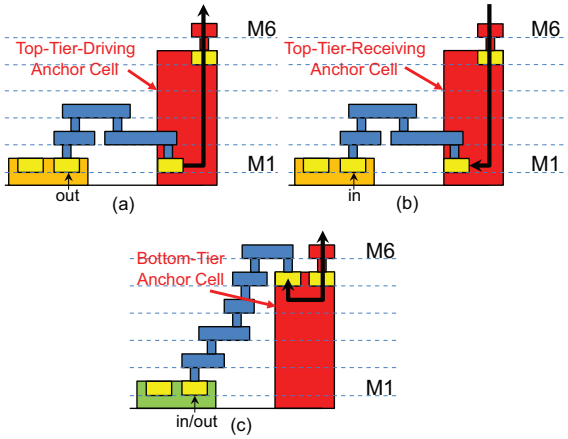

Figure 6: Three types of anchor cells a) a top-tier-driving anchor cell b) a top-tier-receiving anchor cell, c) a bottom-tier anchor cell

regular 2D implementation flow. A clock signal is first divided into two branches in the top partition. One of branches is used for generating clock tree in the top partition, and the other branch is connected to the bottom partition through a set of anchor cells and a dummy wire, and used for generating clock tree in the bottom partition.

Figure 5 (c) shows the Cascade2D design. Although we set the delay of dummy wires to zero their RC parasitics still exist in this stage of the design. Therefore, the Cascade2D design is again partitioned into top and bottom partitions, pushing all cells and wires to the corresponding partitions except dummy wires. Then, RC parasitics for each partition are extracted. The final M3D design is created by connecting these two extracted designs with MIV parasitics. Timing and power analysis is done on the final M3D design.

3. EXPERIMENTAL SETUP

3.1 Process Nodes and Design Libraries

The experimental set-up is same as that described in [8] and is reproduced here for the sake of clarity and completion. Table 2 shows the representative metrics for each process technology used

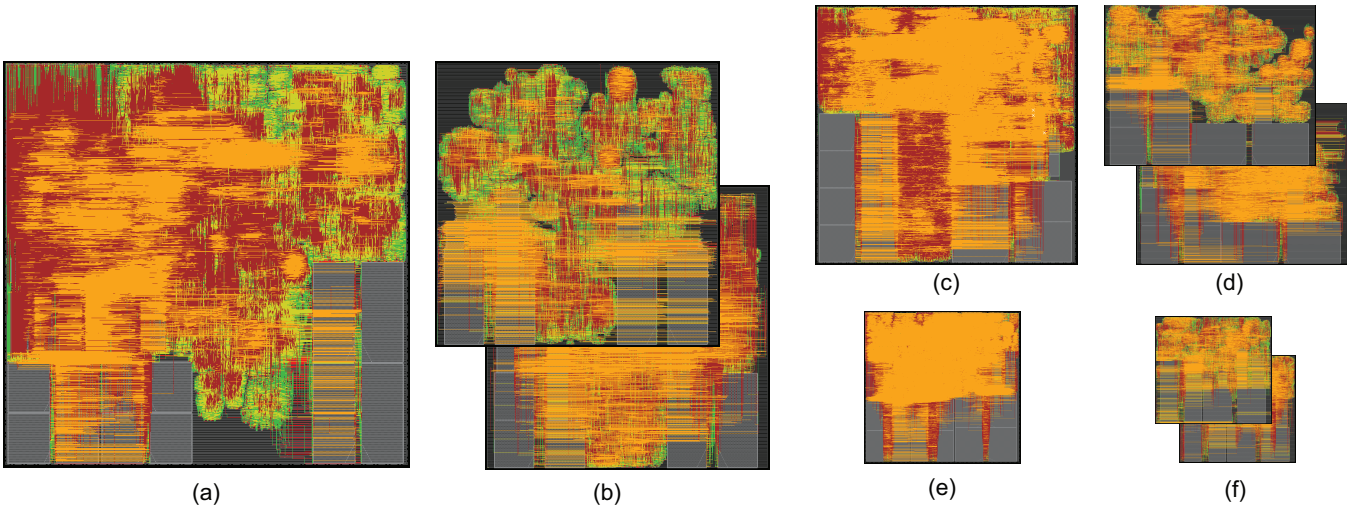

Figure 7: GDS layouts of a) 28nm 2D, b) 28nm Cascade2D M3D, c) 14/16nm 2D, d) 14/16nm Cascade2D M3D, e) 7nm 2D and f) 7nm Cascade2D M3D of the application processor at 1.0GHz

Table 2: Key metrics for foundry 28nm, 14/16nm and the predictive 7nm technology node used in this study. MIV stands more monolithic inter-tier via.

Parameters	28nm [3, 4]	14/16nm [5, 6]	7nm [7]
Transistor type	Planar	FinFET	FinFET
Supply Voltage	0.9V	0.8V	0.7V
Contacted Poly-pitch	110-120nm	78-90nm	50nm
Metal1 Pitch	90nm	64nm	36nm
MIV cross-section	80x80nm	40x40nm	32x32nm
MIV height	140nm	170nm	170nm

in our study, based on previous publications [3, 4, 5, 6, 7]. The 28nm process is planar transistor based while 14/16nm is the first generation foundry FinFET process. For these nodes, we have used production level standard cell libraries containing over 1,000 cells and memory macros that were designed, verified and characterized using foundry process design kits (PDK).

Since the 7nm technology node parameters are still under development by foundries, we utilized a predictive PDK to generate the required views for this study. We have developed the predictive 7nm PDK containing electrical models (BSIM-CMG), DRC, LVS, extraction and technology library exchange format (LEF) files. The transistor models incorporate scaled channel lengths and fin-pitches and increased fin-heights compared to previous technology nodes in order to improve performance at lower supply voltages. Multiple threshold voltages (V_T) and variation corners are supported in the predictive 7nm PDK. Process metrics such as gate pitch and metal pitches are linearly scaled from previous technology nodes [7] and design rules are created considering lithography challenges associated with printing these pitches. The interconnect stack is modeled based on similar scaling assumptions. A 7nm standard cell library and memory macros are designed and characterized using this PDK.

The M3D design requires six metal layers on both top and bottom tiers. The MIVs connect M6 of the bottom tier with M1 of the top tier. We limit the size of the MIVs to be 2x the minimum via size allowed in the technology node to reduce MIV resistance. The MIV heights take into account the fact that the MIVs need to traverse through inter-tier dielectrics and transistor substrates to contact to M1 on the top tier. The MIV height increases from 28nm to 14/16nm and 7nm technology nodes because of the introduction of local interconnect middle-of-line (MOL) layer in the sub-20nm nodes. MIV resistance is estimated based on the dimension of the vias and we used previously published values for MIV capacitance from [1].

Since M3D fabrication is done sequentially, high temperature front-end device processing of the top tier can adversely affect the interconnects in the bottom tier while low temperature processing will result in inferior top tier transistors. Recent work reporting low temperature processes that achieve similar device behavior across both tiers have been presented [9] and hence, all our implementation studies are done with the assumption of similar device characteristics in both the tiers.

3.2 Implementation Setup

The standard cell libraries and memory macros for the 28nm, 14/16nm and 7nm technology nodes are used to synthesize, place and route the full-chip design. 2D and M3D designs of the application processor are implemented sweeping the target frequency from 500MHz to 1.2GHz in 100MHz increments across the three technology nodes. Full-chip timing is met at the appropriate corners, i.e., slow corner for setup and fast corner for hold. Power is reported at the typical corner. The floorplan of the design is customized for each technology node to meet timing but kept constant during frequency sweeps. The chip area is fixed such that the final cell utilization is similar across technology nodes. In the next section we present the results from the Cascade2D flow and compare with the state-of-the-art M3D partitioning and implementation flow called Shrunk2D design [8].

4. RESULTS AND ANALYSIS

4.1 Power and Performance Benefit

Figure 7 shows the die images of 2D and Cascade2D M3D implementations of the commercial, in-order, 32-bit application processor on target frequency of 1.0GHz in 28nm, 14/16nm as well as 7nm technology nodes. Since 28nm and 14/16nm designs are unable to meet timing at 1.2GHz, designs of target frequency up to 1.1GHz are presented as results. For 7nm, we report its results up

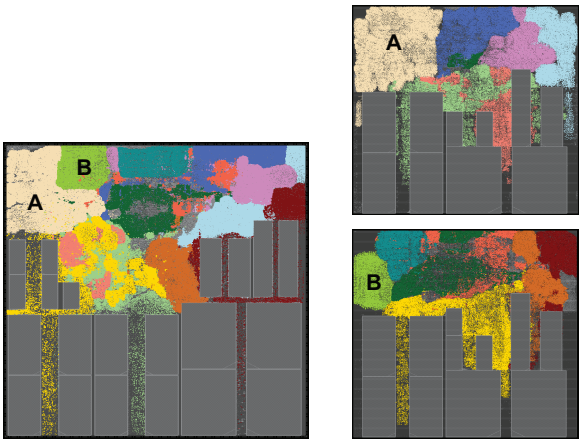

Figure 8: Color map of functional modules between 7nm a) 2D design and b) Cascade2D M3D design of the commercial processor at 1.0GHz

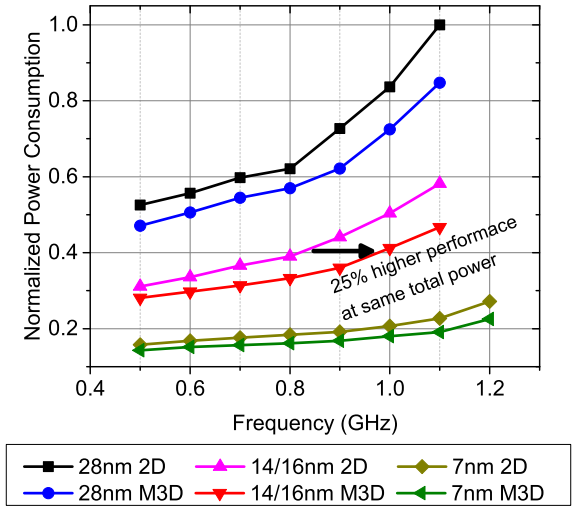

Figure 9: Normalized power consumption of 2D and Cascade2D M3D designs across technology nodes

to 1.2GHz.

From timing analysis of the 2D design, we found that functional module A and B in Figure 8 (a) have large number of timing paths crossing them. In the Cascade2D M3D design, those modules are floorplanned on top of each other minimizing the distance between them using MIVs, whereas those functional modules are floorplanned side-by-side in the 2D design. This vertical integration reduces wire-length of signals crossing the modules as well as standard cell area of the modules because of reduced wire parasitics.

The normalized total power consumption of the 2D and Cascade2D M3D designs across technologies are shown in Figure 9. We observe that Cascade2D M3D designs consume less power in all cases. Hence, at iso-power, M3D designs run at higher frequencies compared to the 2D designs. For example, considering the 14/16nm technology node and we see that M3D designs can have 25% higher performance at the same total power compared to the 2D designs. Figure 10 shows power saving comparison between

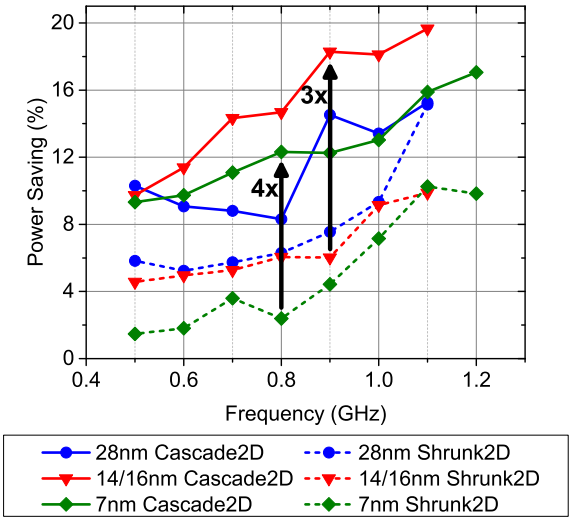

Figure 10: Power saving of Cascade2D M3D (solid lines) and Shrunk2D M3D (dotted lines) designs over 2D designs

Cascade2D M3D and Shrunk2D M3D designs from their 2D counterparts. Cascade2D M3D designs show up to 3-4X better power saving than Shrunk2D M3D designs depending on the technology node and design frequency. In the best case scenario, M3D design shows 20% power reduction than the 2D design (14/16nm technology node at 1.1Ghz frequency) at the same performance point.

4.2 Comparison to State-of-the-Art

To analyze the difference in power saving between Cascade2D M3D and Shrunk2D M3D designs, we use the following equation for dynamic power.

$$P_{dyn} = P_{INT} + \alpha \cdot (C_{pin} + C_{wire}) \cdot {V_{DD}}^2 \cdot f_{clk} \quad (1)$$

The first term P_{INT}, is internal power of the gates, and the second term describes switching power where C_{pin} is the pin capacitance of the gates, C_{wire} is the wire capacitance in the design, α is the activity factor, f_{clk} is the design clock frequency. Since internal power and pin capacitance depends on standard cell area, and wire capacitance is correlated to wire-length, we can extend Equation 1 to Equation 2, to describe the factors affect power saving of M3D designs.

$$\begin{aligned}\Delta P_{dyn} =& \Delta_{cell} \cdot (P_{INT} + \alpha \cdot C_{pin} \cdot {V_{DD}}^2 \cdot f_{clk}) \\ &+ \Delta_{wire} \cdot \alpha \cdot C_{wire} \cdot {V_{DD}}^2 \cdot f_{clk}\end{aligned} \quad (2)$$

where Δ_{cell} and Δ_{wire} are the difference in standard cell area and wire-length between 2D and M3D designs, respectively.

The primary advantage of Shrunk2D M3D designs comes from reduced wire-length, which results in reduced wire-switching power dissipation [8]. As shown in Figure 11, Shrunk2D M3D designs reduce wire-length by 20-25% consistently across technology nodes and frequencies. Wire-length reduction is mainly attributed to vertical integration between cells through MIVs. Table 3 compares the number of MIVs Shrunk2D M3D and Cascade2D M3D designs. Since Shrunk2D flow partitions 'cells' into two tiers whereas Cascade2D flow partitions 'functional blocks', the number of MIVs in Shrunk2D M3D designs is an order of magnitude higher than that in Cascade2D M3D designs. Better wire-length savings using Shrunk2D flow can be attributed to the large number of MIVs.

Table 4: Normalized iso-performance comparison of 2D implementations and their M3D counterparts of the application processor across technology nodes at 1.0GHz. All values are normalized to corresponding 28nm 2D parameters. Capacitance and power values are normalized to 28nm 2D total capacitance and 28nm 2D total power, respectively.

	Normalized 2D			Shrunk2D percentage change from 2D			Cascade2D percentage change from 2D		
Parameters	28nm	14/16nm	7nm	28nm	14/16nm	7nm	28nm	14/16nm	7nm
Std. cell area	1	0.331	0.077	-7.6 %	-6.8 %	-7.5 %	-9.5 %	-11.9 %	-8.8 %
Wire-length	1	0.728	0.404	-19.3 %	-24.1 %	-24.6 %	-11.9 %	-22.6 %	-12.2 %
Wire cap	0.531	0.375	0.205	-18.1 %	-14.2 %	-13.7 %	-9.5 %	-19.7 %	-19.2 %
Pin cap	0.469	0.422	0.203	-12.1 %	-6.3 %	-9.7 %	-11.1 %	-13.2 %	-7.9 %
Total cap	**1**	**0.797**	**0.408**	**-15.5** %	**-10.1** %	**-11.7** %	**-9.6** %	**-15.2** %	**-12.9** %
Internal power	0.428	0.282	0.128	-4.8 %	-7.6 %	-4.7 %	-14.5 %	-15.2 %	-11.1 %
Switching power	0.505	0.318	0.119	-13.4 %	-10.6 %	-10.1 %	-13.0 %	-20.8 %	-15.1 %
Leakage power	0.066	0.002	0.000	-7.7 %	-4.0 %	-2.0 %	-9.5 %	-7.7 %	-2.8 %
Total power	**1**	**0.602**	**0.247**	**-9.3** %	**-9.1** %	**-7.2** %	**-13.4** %	**-18.1** %	**-13** %

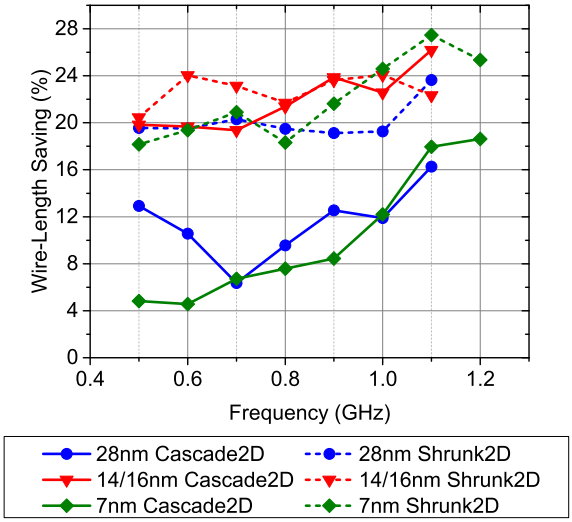

Figure 11: Wire-length reduction comparison between Cascade2D (solid lines) and Shrunk2D (dotted lines) M3D designs

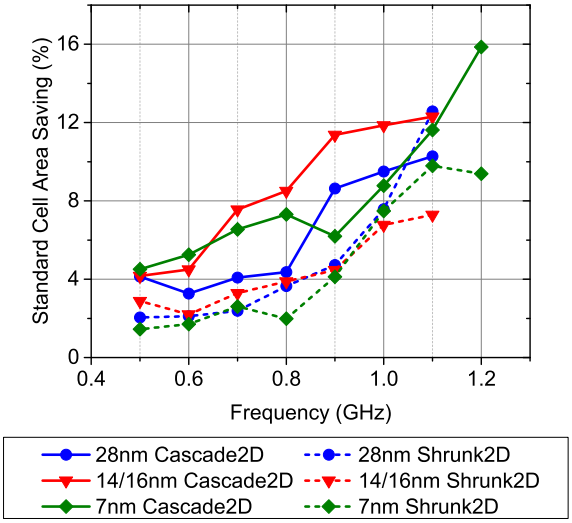

Figure 12: Standard cell area saving in Cascade2D (solid lines) and Shrunk2D (dotted lines) M3D designs

Table 3: Number of MIVs in 28nm, 14/16nm and 7nm M3D of the application processor at 1.0GHz

MIV count	28nm	14/16nm	7nm
Cascade2D	7,545	7,545	7,545
Shrunk2D	164,553	120,770	99,587

The large number of MIVs in Shrunk2D M3D designs helps to reduce wire-length, but it also increases the total capacitance of MIVs, limiting the wire capacitance reduction. As shown in Table 4, although Shrunk2D M3D designs reduce more wire-length than Cascade2D M3D designs in 14/16nm and 7nm designs, wire capacitance reduction of Cascade2D M3D designs higher than Shrunk2D M3D designs. Additionally, there is a negative impact of large number of MIVs on wire capacitance mainly because of the bin-based partitioning scheme of the Shrunk2D flow [1]. While bin-based partitioning helps to distribute cells evenly on both tiers, it has a tendency to partition cells connected using local wires into two tiers, increasing wire capacitance.

On the other hand, Cascade2D M3D designs save their power mainly by reducing standard cell area. Shrunk2D flow uses a 'shrunk 2D' design to estimate wire-length and wire parasitics of the resulting M3D design. However, while shrinking technology geometries, minimum width of each metal layer is also scaled, and extrapolation is performed by tools during RC extraction of wires. This extrapolation tends to overestimate wire parasitics, especially in scaled technology nodes, which results in large number of buffers inserted in the design to meet timing. In Cascade2D flow, buffers are inserted while implementing/optimizing top and bottom partition simultaneously with actual technology geometries, Cascade2D flow achieves more standard cell area than Shrunk2D flow as shown in Figure 12.

With a reduction in standard cell area, the cell density of the M3D design reduces as well. Hence, we leverage this feature of M3D designs to increase cell density and reduce die-area. We implement two separate M3D designs using the Cascade2D flow, one with the same total die-area as the 2D design and another with 10% reduced area. Table 5 shows that we can maintain similar power savings with a reduced die-area M3D design. The ability to get reduced die area makes M3D technology extremely attractive for main-stream adoption because less area directly translates to reduced costs.

As shown in Equation 2, standard cell area reduction affects both internal power, pin cap switching power reduction, whereas wire-length reduction reduces only wire cap switching power. Figure 13 shows power breakdown of 2D, Cascade2D M3D, and Shrunk2D M3D. As shown in the figure, internal power and pin capacitance

Table 5: Normalized iso-performance comparison of 2D design, Cascade2D M3D designs with same die area and 10% reduced die area at 1.1GHz in predictive 7nm technology node

Parameters	2D	Cascade2D	
Die-area	1	1	0.9
Density	69.7%	63.2%	71.1%
Total power	1	0.841	0.871

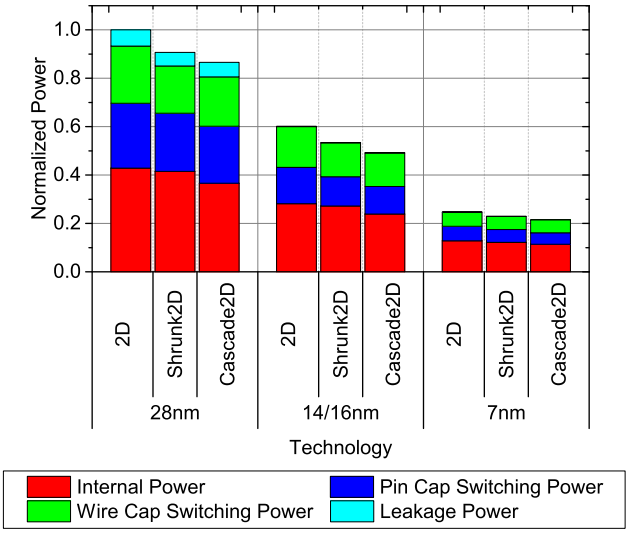

Figure 13: Power breakdown into internal power, pin cap switching power, wire cap switching power and leakage power for 2D, Shrunk2D M3D, and Cascade2D M3D designs at 1.0GHz in foundry 28nm, 14/16nm, and predictive 7nm technology nodes

switching power, which depends on standard cell area, account for over 70% of total power, and they contribute even more in 14/16nm and 7nm designs. Cascade2D M3D designs reduce more standard cell area compared to Shrunk2D M3D designs by attacking 70% of the total power; they achieve better power savings consistently, even though wire-length reduction of Cascade2D M3D designs is less than Shrunk2D M3D designs.

Table 6 shows the comparison of run-time between the Cascade2D flow and the Shrunk2D flow. For the Shrunk2D flow, we assume that the design library with shrunk geometry is available. The total run-time for each flow is comparable. It is important to note that both flows need a reference 2D design. The 2D design is needed in the Shrunk2D flow to evaluate the quality of the final M3D design, while it is useful in the Cascade2D flow to extract timing and standard cell area information for the design-aware partitioning step.

5. CONCLUSIONS

In this paper, we present a new methodology called 'Cascade2D' to implement M3D designs using 2D commercial tools. The Cascade2D flow utilizes a design-aware partitioning scheme where functional modules with very large number of connections are partitioned into separate tiers. One of the main advantages of this flow is that it is extremely flexible and is partition-scheme agnostic, making it an ideal methodology to evaluate different M3D partitioning algorithms. The MIVs are modeled as sets of anchor cells and dummy wires, which enable us to implement and optimize both top and bottom tiers simultaneously in a 2D design. The Cascade2D flow reduces standard cell area effectively, resulting in significantly better power savings than state-of-the-art M3D flows developed previously. Experimental results with a commercial, in-order, 32-bit application processor in foundry 28nm, 14/16nm, and predictive 7nm technology nodes shows that Cascade2D M3D designs can achieve up to 4X better power savings compared to the state-of-the-art M3D designs from Shrunk2D flow, while using an order of magnitude less MIVs. In the best case scenario, M3D designs created using this new methodology result in 25% higher performance at iso-power and up to 20% power reduction at iso-performance compared to 2D designs. Additionally, by leveraging smaller standard cells we demonstrate that M3D designs can save up to 10% die-area which directly translates to reduced costs. These results highlight the fact that monolithic 3D possesses the potential to enable power, performance and area scaling equivalent to a full Moore's Law node and we hope that this work paves the way for more research to combat manufacturing, thermal, process variation and EDA tool challenges associated with this novel technology.

Table 6: Run-time comparison between Shrunk 2D flow and Cascade2D flow with the application processor at 1.0GHz in 7nm technology node

Shrunk2D flow		Cascade2D flow	
Step	Run-time	Step	Run-time
1. Shrunk2D impl.	5hr	1. Design-aware part.	0.5hr
2. Gate-level part.	0.5hr	2. MIV plan	2hr
3. MIV plan	0.5hr	3. Cascade2D impl.	4.5hr
4. Top/bottom tier impl.	1.5hr	-	-
Total	7.5hr	Total	7hr

6. REFERENCES

[1] 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. on Low Power Electronics and Design*, 2014.

[2] O. Billoint *et al.*, "A Comprehensive Study of Monolithic 3D Cell on Cell Design Using Commercial 2D Tool," in *Proc. Design, Automation and Test in Europe*, 2015.

[3] Inside the iPhone 5s, "https://www.chipworks.com/about-chipworks/overview/blog/inside-the-iphone-5s".

[4] S. Yang *et al.*, "28nm Metal-gate High-K CMOS SoC Technology for High-Performance Mobile Applications," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2011.

[5] S.-Y. Wu *et al.*, "A 16nm FinFET CMOS Technology for Mobile SoC and Computing Applications," in *Proc. IEEE Int. Electron Devices Meeting*, 2013.

[6] T. Song *et al.*, "A 14nm FinFET 128Mb 6T SRAM with VMIN-Enhancement Techniques for Low-Power Applications," in *IEEE Int. Solid-State Circuits Conference Digest of Technical Papers*, 2014.

[7] K.-I. Seo *et al.*, "A 10nm platform technology for low power and high performance application featuring FINFET devices with multi workfunction gate stack on bulk and SOI," in *Symposium on VLSI Technology Digest of Technical Papers*, 2014.

[8] K. Chang *et al.*, "Match-making for Monolithic 3D IC: Finding the Right Technology Node," in *Proc. ACM Design Automation Conf.*, 2016.

[9] P. Batude *et al.*, "3DVLSI with CoolCube process: An alternative path to scaling," in *Symposium on VLSI Technology Digest of Technical Papers*, 2015.