3DIC Benefit Estimation and Implementation Guidance From 2DIC Implementation

Wei-Ting J. Chan[†], Yang Du[§], Andrew B. Kahng[†+], Siddhartha Nath[+] and Kambiz Samadi[§]
[†]ECE and [+]CSE Departments, University of California at San Diego
[§]Qualcomm Research, San Diego, CA
{wechan, abk, sinath}@ucsd.edu, {ydu, ksamadi}@qti.qualcomm.com

ABSTRACT

Quantification of three-dimensional integrated circuit (3DIC) benefits over corresponding 2DIC implementation for arbitrary designs remains a critical open problem, largely due to nonexistence of any "golden" 3DIC flow. Actual design and implementation parameters and constraints affect 2DIC and 3DIC final metrics (power, slack, etc.) in highly non-monotonic ways that are difficult for engineers to comprehend and predict. We propose a novel machine learning-based methodology to estimate 3DIC power benefit (i.e., percentage power reduction) based on corresponding golden 2DIC implementation parameters. The resulting 3D Power Estimation (3DPE) models achieve small prediction errors that are bounded by construction. We are the first to perform a novel *stress test* of our predictive models across a wide range of implementation and design-space parameters. Further, we explore *model-guided implementation* of designs in 3D to achieve minimum power: that is, our models recommend a most-promising set of implementation parameters and constraints, and also provide *a priori* estimates of 3D power benefits, based on a given design's post-synthesis and 2D implementation parameters. We achieve $\leq$10% error in power benefit prediction across various 3DIC designs.

Categories and Subject Descriptors

B.7.2 [**Hardware**]: INTEGRATED CIRCUITS—*Design Aids*

General Terms

Algorithms, Design, Performance

1. INTRODUCTION

As the semiconductor industry nears the end of the CMOS roadmap, product-level benefits from successive technology nodes have decreased due to reliability, variability, power and thermal constraints. Three-dimensional integrated circuits (3DICs) have emerged as a promising solution to extend both the use of today's device and process technologies, as well as the historical Moore's-Law trajectory of value scaling. Eventual cost benefits of 3DIC have yet to be quantified in a mature supply chain and high-volume production context. However, a consensus value proposition for 3DIC has emerged across both industry and academia, namely, *power reduction* benefits (with implied reliability, cost, and user experience benefits) due to shorter connections that are simply unachievable with 2D integration.

Current 3DICs are based on through-silicon vias (TSVs), but integration density is limited by the pitch of TSVs, with mass production focusing on memory-on-logic designs with relatively few vertical connections [14]. Two emerging alternatives to TSV-based 3D integration are (i) sequential face-to-back (F2B) and (ii) fine-grain face-to-face (F2F) integration technologies. They enable orders of magnitude higher integration density compared to that of TSV-based technology, due to the extremely small size of inter-tier (vertical) vias. For example, CEA-LETI [2] [3] has pursued a sequential 3D integration using a low-temperature bonding process. Recent 3DIC design literature [16] [8] has explored implications of fine-grain F2F integration process. Figure 1 illustrates sequential F2B and fine-grain F2F 3DIC integration.

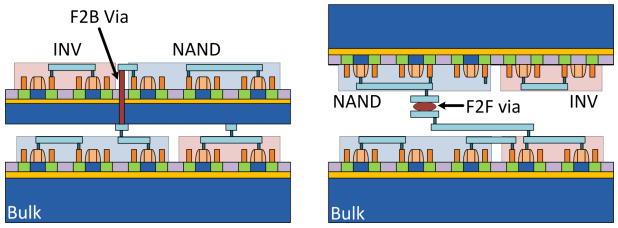


Figure 1: 3D integration: gate-level (a) F2B, and (b) F2F [16].

To perform fast and accurate *implementation-space exploration* (ISE), sometimes referred to as *pathfinding* [13], chip architects and designers require accurate 3D power estimation tools. This is especially critical with power-centric 3DIC value propositions. Unfortunately, power estimation of 3D implementations is challenging because (i) 3D benefit varies with netlist topologies, constraints and implementation styles, and (ii) there are no "golden" 3D implementation flows. To our knowledge, no tool today can accurately predict the power benefit of 3D implementation based on netlist, constraints, and whatever information might be available from 2D implementation. The lack of such an estimation tool results in a large number of iterations (often, not much better than "throwing darts") to identify the best set of implementation parameters and/or constraints for 3D implementation. Only after making many attempts in this manner can the designer gain an inkling of potential 3D implementation benefits for a given block.

In this work, we overcome the above challenge by developing an efficient 3D power estimation methodology, along with an accurate *3D Power Estimation* (3DPE) prediction tool. 3DPE predicts *benefit*, i.e., the "delta" in power (= reduction from 2D implementation) that will be achieved by a given 3D flow. We experimentally confirm that 3DPE can estimate 3DIC power *reduction* with error of $\leq$10% across a set of testcases implemented in foundry 28nm FDSOI technology.

Our 3DPE model development includes a novel exploitation of sensitivities of post-synthesis and post-place-and-route (SP&R) power to wireload model (WLM) and capacitance scaling; this yields new parameters that increase modeling accuracy.[1] We also perform a novel *stress test* of 3DPE by verifying that the model cannot produce unreasonable values of estimated 3D power benefit. While practitioners have struggled with a gap between theoretical limits of 3DIC benefit and observed benefits, our model stress test provides some encouragement in the form of model parameter combinations that suggest potential large 3DIC power benefits. Additional experimental studies confirm the usability of 3DPE in *model-guided implementation* (MGI), e.g., for a given design and set of constraints, 3DPE can identify wireload model scaling, floorplan aspect ratio, target utilization, etc.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
DAC'15, June 07 - 11 2015, San Francisco, CA, USA.
Copyright 2015 ACM 978-1-4503-3520-1/15/06...$15.00.
http://dx.doi.org/10.1145/2744769.2744771.

[1] Our models are based on the sensitivities of power to constraints and design parameters (e.g., mix of threshold voltage types and wirelength) between 2D and 3D implementations of the same designs, and are specific to the flow from [16]. The models must be rederived if the tool flow or technology changes.

settings in commercial SP&R flows to obtain minimum power in the final 3D implementation. We believe that the resulting modeling capability can be used for both ISE and MGI across architectural and physical implementation levels of design. We summarize our main contributions as follows.

- To our knowledge, we are the first to develop an estimation tool that focuses on the 3DIC value proposition of 3D power benefit. Our 3DPE model is achieved with *bounded error* machine learning techniques; it predicts delta power benefit of 3DIC with average (resp. maximum) error of $\leq$0.1% ($\leq$10%) based on netlist, design constraints and 2D implementation metrics.
- We develop novel estimation model parameters based on the sensitivity of synthesis and P&R outcomes to wireload model and capacitance model scaling. This is a heretofore unexplored approach to assessing how RTL and gate-level netlists will react to 3D vs. 2D implementation contexts.
- We propose novel validations of our 3DPE model, including (i) a "stress test" approach to verify that no unreasonable values of predicted power benefit can occur, and (ii) application of 3DPE in *model-guided implementation*.
- Of independent interest is a tight bound on the wirelength benefit of 3D integration vs. 2D integration (Section 2.3), which informs our use of WLM scaling in our modeling flow. Also of independent interest is our observation (Section 2.2) of qualitatively different 3DIC benefit results for the well-studied OpenSPARC T2 design [24], after improving the enablement of experiments starting from the 3D implementation flow of [15].

2. BACKGROUND DISCUSSION

The following subsections summarize (i) related literature; (ii) our baseline 2D and 3D implementation flows, which replicate the flows of [15] [16]; and (iii) a new, tight upper bound on potential wirelength reduction in 3DIC that can inform design-space exploration.

2.1 Related Work

Previous works have addressed (i) 3DIC design and implementation flow development, and (ii) prediction of 3DIC metrics (e.g., area, power, wirelength). There is an interdependency between (i) and (ii) in that a high-quality, reliable 3DIC implementation flow is needed to obtain reliable ground truth data for modeling.

For (i), while there is no golden EDA flow for 3DIC implementation, a number of researchers have implemented 3DICs using 2D EDA tools and flows in conjunction with in-house 3D design tools. Franzon et al. [17] propose a 3D design flow based on a 2D flow to implement an FFT processor. Kim et al. [11] implement a multi-core processor based on commercial 2D EDA tools and use in-house tools to place the vertical interconnects (VIs). The authors verify the result through fabrication in Tezzaron 3D technology at the 130*nm* node. Another 3DIC implementation flow addresses design requirements for sequential 3D [2] technology that permits cell-level 3D integration. Panth et al. [15] [16] propose a design flow for sequential 3D based on commercial EDA and in-house tools, and validate the flow on OpenSPARC T2 and other IPs [23]. This latter flow is, we believe, the most sophisticated and full-featured in the research literature; we have transplanted and used this flow in our present work.

For (ii), previous works have mainly performed analytical modeling of 3D wirelength and power. Mak and Chu [5] present a loose theoretical upper bound on the potential wirelength improvement possible with a 3DIC implementation of any design as compared to its 2DIC implementation. They report that for realistic sizes of VIs, the benefits will always be negative (-2% on average). Kim et al. [10] [12] use Rent's rule to predict wirelength distributions in 3DICs with two or more dies as well as by varying the number of VIs.The authors also derive analytical models to estimate 3D power when heights and widths of VIs, and the number of buffers inserted into the netlist, are varied. However, these models cannot predict the power benefit with 3D implementation when a 2D implementation already exists, since the models do not account for IC implementation details such as floorplan context, technology libraries, signoff corners and constraints. Toufexis et al. [18] propose an in-built statistical prediction engine to estimate area, performance and power, thereby enabling a fast implementation-space exploration flow for 3DICs. The authors use an interpolation scheme to predict power (details are not specified), with reported maximum modeling error of $\sim$58%.

2.2 2D, Shrunk2D and 3D Flows

As mentioned above, the 3DIC implementation flows of [15] [16] are currently published in the research literature. To develop our 3D power estimation model, we use the Shrunk2D (S2D) and 3D flows from [16] as proxies for golden 3DIC implementation. Through extensive interactions with the flow developer [19], we have transplanted the entire flow enablement (including EDA tool versions and PDK versions) and successfully replicated published results. We have subsequently made several automation-centered flow enhancements: automated floorplan adjustment to handle multiple block aspect ratios (ARs); AR- and perimeter-aware pitch selection and placement for pins; instantiation of memories specifically generated from foundry 28*nm* FDSOI technology enablement, with relative placements that scale with block AR; and unified flow and configuration files to enable automation across multiple small and large testcases. Furthermore, we automate parameter sweeps: clock period, capacitance scaling factor, V_t types, transparent use of F2F/F2B configuration, aspect ratio, target utilizations, and design rule constraints (maximum cap load, maximum transition time, etc.)

Of independent interest are the qualitative differences that we observe between our 3DIC benefit results (see Section 4, below) and those reported in [16]. For instance, [16] reports 16.08% power reduction (from 2DIC to 3DIC) for the OpenSPARC T2 (OST2) [24] design using a non-foundry 32/28*nm* PDK and memories scaled down from a 130*nm* technology. However, our experiments with 28*nm* FDSOI foundry design enablement show 3DIC power reductions ranging from -4.1% to 12.7% across a wide range of testcases and implementations; power reductions for *OST2* in particular range from -0.6% to 2.8%. We surmise that the discrepancies stem from such factors as (i) differences between the open-source SAED PDK [21] and real 28*nm* technologies; (ii) scaling of memory models that does not properly comprehend scaling of memory peripheral circuits or geometric considerations such as pin-out locations and memory aspect ratios;[2] and (iii) higher 2DIC QOR from a stronger baseline 2D flow.

2.3 Upper Bound on 3D Wirelength Benefit

As noted above, an upper bound on 3D wirelength (WL) benefit has been shown by [5]. We now derive a new, tight upper bound for 3D WL benefit, assuming a regular lattice of placement sites. Consider an *optimal* 3D placement that is embedded in a 3D grid graph, as shown in the left part of Figure 2. (The two tiers of the 3D placement respectively have Z-coordinate equal to 0 and 1.) We obtain a 2D embedding of the 3D placement (in other words, a 2D placement) as shown in the right part of Figure 2. Specifically, we make the following changes such that each edge of the 3D-placed netlist becomes $\leq 2\times$ longer in 2D (assuming that gate-pitch and gate-width remain the same across 2D and 3D).

- Without loss of generality, the 2D embedding of the 3D grid graph doubles each hop in the X-direction. Therefore, a vertex located at coordinate (i, j, z) in the 3D graph of Figure 2 is mapped to ($2(i-1)+1+z$, j) in the 2D embedding.

- In Figure 2, the length of $n1$ is $1 \cdot \delta_y = 1$, whereas the length of $n1'$ is $2 \cdot \delta_z + 1 \cdot \delta_y = 3$. As a result, we have an upper bound on 3D WL benefit of $(3-1)/3 \times 100 = 66.7\%$. Note that weight is assumed to be zero in the Z-direction because heights are assumed to be very small in 3DIC VIs as compared to pin-to-pin wirelength in a 2DIC. A detailed picture of this mapping is shown in Figure 3. Since unit wire on the X-Y plane can be connected to at most two VIs, 66.7% is a tight upper bound on the wirelength reduction.

It is difficult to speculate on power implications of our 3D WL benefit upper bound. E.g., even if vertical interconnections and gate input pins have zero capacitance, our result shows that net switching power can reduce by at most 66.7% when moving from 2D to 3D. However, implications for other power components (leakage and internal power) are much less obvious.

[2]The previous work shrinks the memory from 130nm technology. The shrinking does not consider pin location constraints in advanced nodes. For example, non-uniform distribution of pins and rectilinear footprints are seen in memories in advanced nodes.

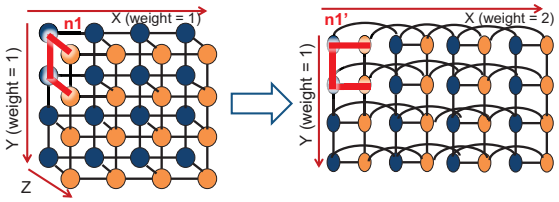


Figure 2: 2D embedding of a 3D graph (wirelength dilation in the X-direction is $\leq 2\times$).

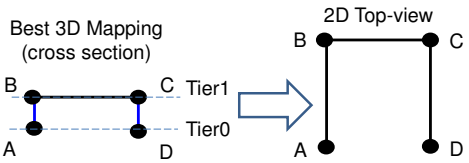


Figure 3: Further details of the three-hop net in Figure 2. In the best 3D mapping, weights of AB and $CD = 0$ (VIs between dies), and weight of $BC = 1$. In the 2D top-view, weights of AB, BC, and $CD = 1$. Tight upper bound of 3D wirelength benefit= $(3-1)/3 \times 100\% = 66.7\%$.

Table 1: Testcases and their post-synthesis results.

Testcase Type	Testcase Name	Number of Instances	Min Clock Period (ns)	% #Buffers/ #cells	% #FFs/ #cells
GPU	THEIA	212K	1.6	20	8
CPU	OST2 (1-core)	347K	1.6	16	22
Modem	Viterbi	98K	1.0	26	27
Multimedia	DCT	12K	1.0	33	6
Peripheral Engine (PE)	AES (crypto)	10K	0.9	22	5

3. MODELING OF 3DIC POWER BENEFIT

The open-source designs used in our experiments are described in Section 3.1. We describe our approaches to identify parameters with greatest influence on 3D power benefit in Section 3.2. In Section 3.3, we describe our machine learning-based methodology to develop 3DPE models.

3.1 Floorplan and Implementation of Testcases

We use a wide range of IPs as our testcases that include building blocks for a modern mobile SoC. The building blocks could be classified into CPU, GPU, modem, multimedia, and peripheral engines. For each class among , we use IPs from OpenCores [23] in which the number of instances in these testcases range from 10K to 347K.[3] Table 1 summarizes the synthesis results for these testcases. The percentage of buffers from all the cells ranges from 15% to 33% and the percentage of flip-flops ranges from 5% to 27%.

As CPUs and GPUs are two key components in mobile SoCs, we use CPU- and GPU-like designs with various memory sizes, shapes, we implement the *OpenSPARC T2* (OST2) core [24] and *THEIA* GPU [23] testcases in foundry 28*nm* FDSOI technology. We overcome the lack of customized memory sizes and number of read/write ports by choosing memories with closest word sizes and word numbers, from foundry 28*nm* memory libraries that cover the required word sizes and word numbers. To emulate the effects of lower capacitance and wirelength in 3D, we use engineered WLMs along with other design parameters to assess the sensitivity of 3DPE models to WLMs and these parameters.

To assess the sensitivity of 3DPE models to floorplan aspect ratios (AR), we implement testcases with AR ranging from 0.8 to 1.2. Given a fixed die area, our memory placement methodology is able to automatically place memory blocks with any floorplan AR in this range as described in Algorithm 1. Initially, we generate a floorplan with AR = 1.0 and cluster memories into four groups.[4] We then place these groups at four corners of the die area (Line 1). When the AR changes, we calculate the coordinates of the four corners of the modified die area, and adjust the placement of each memory group accordingly (Lines 6-10). When there are overlaps between groups due to AR being too large (or too small), we re-cluster the memories so as to remove the overlaps (Lines 11-13).

[3]The P&R runtime of each 2D or 3D run is 16 hours for *OST2*, five hours for *THEIA* and *Viterbi*, and two hours for *AES* and *DCT* when using two threads on a Xeon E5-2640 server with 128*GB* memory.

[4]The memory clusters $Cluster_{BL}$, $Cluster_{BR}$, $Cluster_{TL}$ and $Cluster_{TR}$ are respectively {memories in *IFU*, *FGU*}, {memories in *MMU*}, {non-array part of memories in *LSU*}, and {array part of memories in *LSU*, memories in *EXU0*, *EXU1* and *TLU*} for *OST2*; and are respectively {memories in *CORE0*}, {memories in *CORE3*}, {memories in *CORE1*}, and {memories in *CORE2*} for *THEIA*.

The clustering honors certain basic constraints, e.g., memories are placed face-to-face with respect to each other and each pair has routing channels in between them. We insert at least 5μm routing channels in between memories, and apply placement halos to these channels to avoid routing congestion. In 3D, we use the flow in [16] to place memories based on the corresponding 2D floorplan. Figures 5(a) and (b) respectively show the floorplans of *OST2* and *THEIA* in both 2D and 3D.

Algorithm 1 Floorplan scaling with memories

Procedure *genFloorPlan*
Inputs : AR_list, $area_{sram}$, $area_{postsyn}$, $util$
Outputs : Coordinates of memories for different placement AR

1: Place memories with $AR = 1.0$, such that four memory clusters ($Cluster_{BL}$, $Cluster_{BR}$, $Cluster_{TL}$, $Cluster_{TR}$) are at four corners (BL, BR, TL, TR) of the die
2: $area = (area_{postsyn}/util + area_{sram})$
3: $x_{orig} = \sqrt{area}$
4: $y_{orig} = \sqrt{area}$
5: **for each** $AR \in AR\_list$ **do**
6: $x = \sqrt{area \times AR}$
7: $y = \sqrt{area/AR}$
8: Move $Cluster_{BR}$ by $(x - x_{orig}, 0)$
9: Move $Cluster_{TL}$ by $(0, y - y_{orig})$
10: Move $Cluster_{TR}$ by $(x - x_{orig}, y - y_{orig})$
11: **if** There are overlapped memories **then**
12: Re-cluster memories to remove overlaps
13: **end if**
14: **end for**

3.2 Parameter Identification

We use the S2D flow [16] to sweep parameter values shown in Table 2 and generate the training and testing datasets. The difference in power between S2D and 3D is shown to be <5% in [16]. We confirm that this observation is still true after our modifications described in Section 2.2. Figure 6, which shows eight implementations of the Viterbi decoder across four categories I–IV in Table 3, confirms that we can use S2D as a proxy for 3D implementations in our studies. We execute our design of experiments (DoE) for each testcase using the parameter values shown in Table 2. We run both 2D and S2D implementations using these parameter values for each testcase and extract outcomes of various metrics such as the number of buffers, power, wirelength, cell area, etc. to generate training data points.

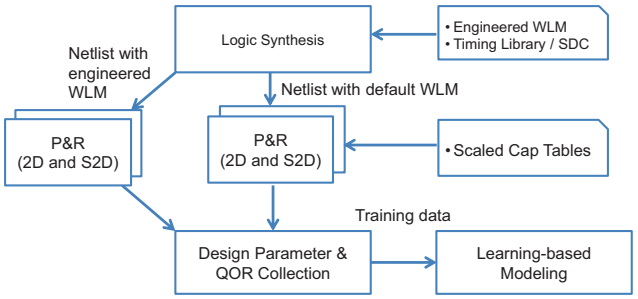


Figure 4: Our overall synthesis and implementation flow is based on the S2D flow of [16]. We generate multiple "engineered" WLMs by scaling capacitance. Our learning-based models can identify 3D benefits by comprehending the change in WL with the change in capacitance between 2D and 3D implementations.

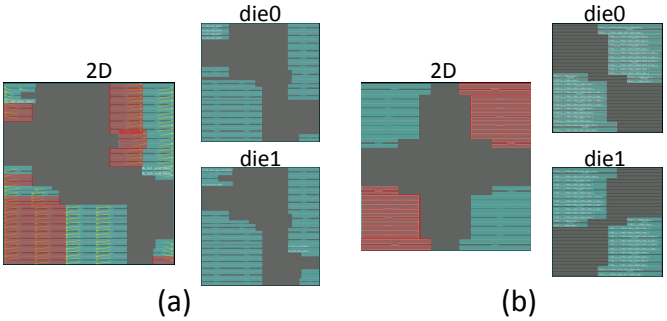


Figure 5: Floorplans of (a) *OST2 core* and (b) *THEIA* (GPU) with AR = 1.0. The red-shadowed memories are partitioned to die0.

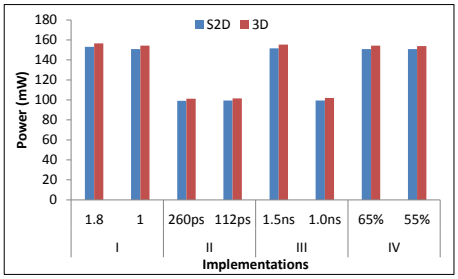


Figure 6: Comparison of power between S2D and 3D across eight implementations.

Table 2 shows the key parameters that influence 3D implementation and can provide guidance in estimating the percentage delta power from a corresponding 2D implementation. Figure 7 illustrates the impact on 2D and 3D power for three of these parameters – utilization (UTIL), AR and max fanout (maxFO) (expressed in the figure in terms of pF). Certain parameters such as maxCap do not have significant impact on 3D power, so we do not use these parameters in our modeling. To limit the number of dimensions in our models, we identify the top-10 parameters based on how significantly these parameters affect percentage delta power. We summarize the percentage change in power with the minimum and maximum parameter values in Table 4. We explore sensitivities of power to capacitance scaling in our DoE by (i) running a 2D implementation with 0.7× capacitance scaling for all interconnects, and (ii) varying post-synthesis netlists by changing capacitance using wireload models. We correlate the change in metrics with and without capacitance scaling from both 2D P&R and logic synthesis as part of our model derivations.

The modeling problem of predicting percentage delta power in 3D by only observing metrics from a 2D implementation is nontrivial. Figure 7 as well as our experimental results indicate that 3DIC power is nonunimodal as well as nonmonotonic with different parameters. For example, a large change (i.e., ∼10×) in the number of buffers in 2D between scaled and non-scaled capacitances can lead to a relatively small (i.e., <10%) change in 3D power. Based on our experimental results, the parameters that affect delta power in 3D include WLM scaling, clock period, mix of V_t types, AR, UTIL, maxTran, maxFO, maxCKskew, maxCKlat and maxCKtran.

Table 2: Key parameters used by 3DPE models.

Design metric	Symbol	Description
V_t types	V_t	Mix of threshold voltage libraries = {RVT, RVT & LVT}
Aspect Ratio	AR	{1.0, 1.2, 1.5}
Utilization (%)	UTIL	{50%, 60%}
Max transition	maxTran	{50%, 70%} of max tran in library
Max capacitance	maxCap	{50%, 100%} of max cap in library
Max fanout	maxFO	{5, 10}
Max clock skew	maxCKskew	{0ps, 50ps}
Max clock latency	maxCKlat	{500ps, 2500ps}
Max clock transition	maxCKtran	{20%, 30%} of clock period
Corners	CORNER	PVT = {TT, 0.92V, 25^oC} Analysis = {setup}
WLM scaling	WLMSC	Capacitance scaling = {1.0, 0.7, 0.33}

Table 3: Implementations used in S2D vs. 3D comparisons.

Category	Clock Period	Util	AR	Max Cap	Max Tran	Max FO
I	1.0ns	65%	{1.8, 1.0}	450pF	260ps	10
II	1.5ns	65%	1.0	450pF	{260, 112}ps	10
III	{1.5, 1.0}ns	65%	1.0	450pF	112ps	10
IV	1.0ns	{65%, 55%}	1.0	450pF	260ps	10

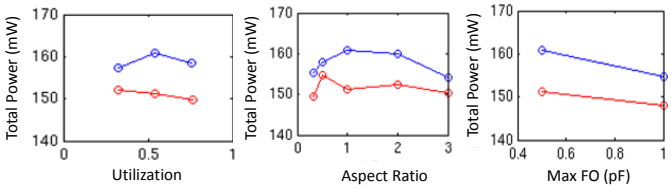


Figure 7: Illustration of implementation parameter impact on 2D (blue lines) and 3D power (red lines).

3.3 Machine Learning Methodology

We develop a model to predict percentage delta power between 2D and 3D implementations of a testcase. From the post-synthesis and post-P&R results of the testcase, we obtain the following parameters.

Table 4: The percentage difference in power with the extreme points.

	UTIL	AR	maxCap	maxTran	maxFO
2D	2.2%	4.4%	3.0%	0.4%	4.0%
S2D	1.5%	3.4%	0.0%	0.6%	2.3%

(i) **Post-synthesis** – number of standard cells, number of buffers and inverters, area of standard cells, internal and leakage power of buffers[5] and non-buffer cells, and net switching power with capacitance in wireload models set to multiple values, and (ii) **Post-P&R** – number of standard cells, number of buffers and inverters, area of standard cells, internal and leakage power of buffers and non-buffer cells, net switching power with capacitance factor set to 1.0× and 0.7×, and wirelength.

The total power in 2D (resp. S2D) implementations is the sum of internal, leakage and net switching power values [22]. For each testcase, we calculate the total power in mW. The S2D implementations are used as a proxy for 3D implementation in our modeling. Table 5 shows the "ground truth" data for deltas in wirelength, number of buffers, and power for S2D versus 2D implementation according to our DoE and testcases (Sections 3.2 and 3.1). Table 2 shows examples of the range of values of our metrics. We create a total of three datasets for modeling – training, validation and testing. Out of all the data points we generate using 2D and S2D P&R flows, we use ∼40% for training, ∼20% for validation, and the remaining ∼40% for testing and reporting errors.

We use Artificial Neural Networks (ANN) as our modeling technique, via the in-built *Matlab vR2013a* toolbox. We use nonlinear modeling because the percentage delta power is non-monotone with respect to the parameters. The complex interactions between parameters are determined automatically by the ANN technique using hidden layers and weights. The hyperparameters [6] we tune are the number of epochs and the number of neurons per hidden layer. We use two hidden layers – one for input and one for output. We vary the number of epochs from 1000 to 5000 in steps of 500 and the number of neurons per hidden layer from one to twice the number of modeling parameters K. We increase accuracy of our models by choosing appropriate hyperparameters such that the range of errors is within a bound. To achieve bounded errors, we search for the number of epochs and the number of neurons that satisfy the following two criteria: (i) the ratio of mean square errors (MSEs) in the training and the validation sets is ≤5, and (ii) add a large multiplicative penalty to suppress outliers (we use 1000×) whenever the range of errors is greater than the bound (we use ∼10% as our error bound and call these data points as outliers). We also perform five-fold cross-validation when training the model. Applying these criteria enables us to develop models that are not overfitted and can generalize to parameter values that are not present in the training dataset. Figure 8 illustrates our modeling flow, which is executed five times due to five-fold cross-validation.[6]

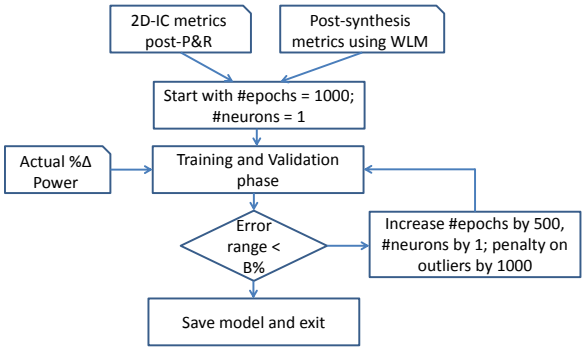


Figure 8: Illustration of our modeling flow.

Table 5: Experimental results of delta outcomes between 2D and S2D.

	2D - S2D			(2D - S2D) / 2D × 100%		
	Min	Max	Mean	Min	Max	Mean
ΔWL (m)	0.03	5.65	1.37	1.95%	35.99%	29.71%
Δ(#buf + #inv)	0.01K	25K	3.6K	0.40%	10.41%	5.05%
ΔTotal power (mW)	-1.40	9.10	2.29	-1.39%	12.72%	3.71%

[5] In the following, we collectively refer to buffers and inverters as "buffers".

[6] The runtime to train our models is four hours on an Intel Xeon E5-2640 2.5 GHz server, using eight threads. This is a one-time cost.

4. MODEL OUTCOMES

We now present our experimental studies. We discuss accuracy results of our 3DPE tool in Section 4.1, robustness and scalability of the 3DPE models in Section 4.2, and model-guided implementation results in Section 4.3.

4.1 Bounded-Error Models

We create three separate models to predict percentage delta (3D or S2D, relative to 2D) for each power component – internal, switching and leakage. We use the predicted values from these models to predict total power in 3DPE. To predict percentage delta internal power, we use seven parameters: (i) internal power from 2D; (ii) ratio of the number of buffers to the total cell count; (iii) delta internal power in 2D with and without capacitance scaling of 0.7× in the technology capacitance tables; (iv) delta internal power in the post-synthesis netlist with and without capacitance scaling by using wireload models (WLMs); (v) the ratio of utilization to cell area; (vi) ratio of memory area to total cell area; and (vii) AR. To predict percentage delta switching power, we use six parameters: (i) switching power from 2D; (ii) the ratio of total wirelength (WL) in 2D to utilization; (iii) delta WL; (iv) delta switching power in 2D with and without capacitance scaling of 0.7× in the technology capacitance tables; (v) delta switching power in the post-synthesis netlist with and without capacitance scaling by using wireload models (WLMs); and (vi) AR. To predict percentage delta leakage power, we use three parameters: (i) leakage power from 2D; (ii) the ratio of low-Vt cell area to total cell area; and (iii) the ratio of memory area to total cell area.

Figure 9 shows our prediction for percentage delta power across our testcases. The solid black line in the middle indicates the line when there is perfect correlation between actual percentage delta power and predicted percentage delta power. The upper and the lower solid lines define the band between maximum and minimum errors. Across our testcases we achieve an error range of ∼9.0%. Figure 10 shows a histogram of error distribution. Only a few outliers are responsible for the maximum and minimum errors. The average error from our total power model is ∼-0.1%.[7]

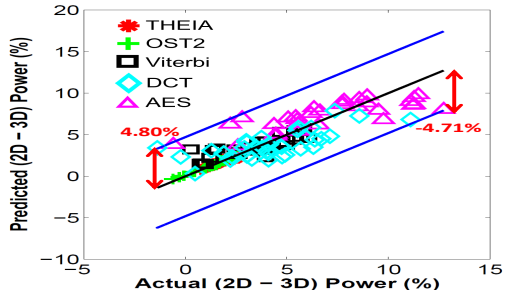


Figure 9: Predicted % delta power vs. actual % delta power between 2D and S2D.

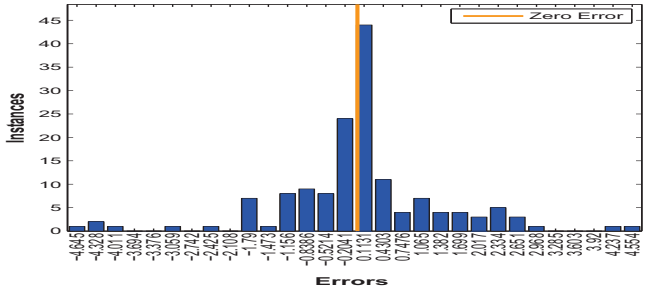


Figure 10: Model error distribution of % delta power.

4.2 Testing of Model Implications

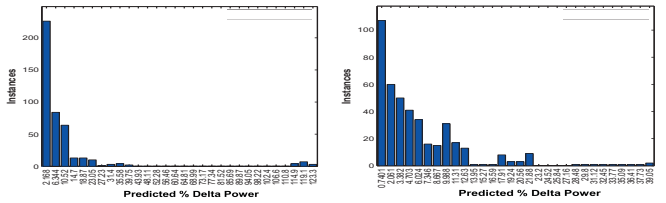
As the time to generate each data point using 2D and S2D flows can be very large (e.g., up to 16 hours for one data point of *OST2*), it is practically impossible to train models with a large range of parameter values. Our models should be scalable and generalizable due to use of cross-validation [6] in our methodology. However, we go a step further with a novel "stress test" of our 3DPE models. (That is, we explicitly test whether the models are capable of returning an unlikely or unreasonable prediction. E.g., if it is possible for our models to predict 90% power benefit from 3DIC, this would cast doubt on the models.)[8] We perform stress testing on our total power model using the following methodology. We vary the following 10 parameters in our models: (i) internal, switching and leakage power values in 2D ($K = 1$, $K = 2$, and $K = 3$); (ii) total WL in 2D ($K = 4$); (iii) utilization ($K = 5$); (iv) number of cells ($K = 6$); (v) total cell area ($K = 7$); (vi) number of buffers ($K = 8$); (vii) ratio of memory area to cell area ($K = 9$); and (viii) maximum transition ($K = 10$). Table 6 shows the distribution of these parameters extracted from our training dataset. We execute the following steps.

- For each parameter i, where $i = 1, 2, ..., 10$, we obtain the mean (μ_i) and standard deviation (σ_i) from the training dataset.
- We construct a test dataset by assuming that each parameter follows a Gaussian distribution with mean and standard deviation respectively indexed into values from the sets μ_i' and σ_i', where $\mu_i' = \{\mu_i, 2\mu_i, ..., 10\mu_i\}$, and $\sigma_i' = \{\sigma_i, 2\sigma_i, ..., 10\sigma_i\}$.
- We index each of these values from sets μ_i' and σ_i' as $s = \{1, 2, ..., |\mu_i'|\}$, and generate a value for each parameter $x_i = \mu_i'(s) + j \times \sigma_i'(s)$, with j varying from -3 to +3 in steps of 0.2.

We generate a total of 434 test data points for the 10 tuples of parameters. Figure 11(a) shows a histogram of percentage predicted delta power. The minimum value is 0.08% (the corresponding bin is 2.17%) and the maximum value is ∼125% (the corresponding bin is 123.3%). The weighted mean of the predicted percentage delta power values is 9.5%. For 14 test data points, our models predict over 100% percentage delta power. These data points have the following attributes which are not practically realizable: (i) the ratio of cell area to the number of cells is larger than the area of the largest cell in the technology library,[9] that is, the number of cells, utilization and the cell area are mismatched; (ii) the ratio of wirelength to the number of cells is less than 50μm, that is, the wirelength and the number of cells are mismatched; and (iii) the maximum transition and/or the maximum fanouts are more than 10× the maximum value in the technology libraries, that is, constraints are mismatched. Figure 11(b) shows a histogram of percentage delta power benefits for data points that are realizable for practical netlists and do not violate constraints with respect to the technology libraries. The maximum possible percentage delta power for these data points is ∼39%, which indicates that our model predictions are close to the values of 3DIC power improvements reported in [10].

Table 6: Distributions of parameters for stress testing.

	$K=1$ (mW)	$K=2$ (mW)	$K=3$ (mW)	$K=4$ (m)	$K=5$ (%)	$K=6$ ($\times 10^3$)	$K=7$ (mm^2)	$K=8$ ($\times 10^3$)	$K=9$	$K=10$ (ps)
μ	71.45	34.97	0.33	4.49	0.56	134.51	0.18	186.48	0.16	200
σ	67.90	30.89	0.45	5.88	0.13	141.77	0.20	286.65	0.22	56



(a) (b)

Figure 11: Percentage predicted delta power distributions (a) when practically unrealizable data points are included, and (b) when only practically realizable data points are included.

4.3 Model-Guided Implementation

An "ultimate" goal of our 3DPE modeling work is to enable fast and accurate design- and implementation-space exploration without actually having to implement a testcase either in 2D or S2D. Toward this end, we explore whether our models can provide guidance to designers as to which classes of testcases are amenable to what kinds of 3D benefits. We conduct two experiments to demonstrate how 3DPE can provide guidance to designers.[10]

[7] The average (resp. maximum) of absolute errors in our internal, switching and leakage power models are respectively -0.42% (resp. 10.9%), -0.07% (resp. 5.6%) and -0.61% (resp. 2.9%). The testing time is ∼one second per every 500 data points.

[8] Our sanity-check approach can be a useful addition to the metamodeling works that have become very popular in the recent IC CAD literature.

[9] In our 28nm FDSOI libraries, the size of the largest cell is 4.4μm^2. The inter-buffer distance is ∼120–150μm [1]. The max transition is 375ps and the max fanout is 20.

[10] Note that although we use one testcase to demonstrate MGI-I and MGI-II, the conclusions drawn are not limited to a specific testcase.

Table 7: Predicted vs. Actual 3D power with high utilization.

Clock Period (*ns*)	AR	UTIL (%)	Predicted % Delta Power (*mW*)	Actual % Delta Power (*mW*)
1.60	1.2	40	2.07	1.82
1.60	1.0	42	1.97	2.14
1.80	1.2	45	2.15	2.46
1.80	1.0	48	2.18	2.88

- **MGI-I** – To predict the WLM scaling *at synthesis* that can lead to the smallest post-P&R power in 3D for a testcase.
- **MGI-II** – To use a low-utilization (small tool runtime) trial 2D implementation to predict the % delta power of a high-utilization (large tool runtime) S2D implementation.

Experiment MGI-I. The goal of this experiment is to create a "properly 3D-aware" netlist by scaling WLM capacitances that can deliver the smallest power in 3D. We create eight WLMs with capacitances $\{1.0, 0.85, 0.70, 0.60, 0.50, 0.45, 0.40, 0.33, 0.3\}pF$. We do not use scaling factor $<0.3pF$, based on the theoretical limit of WL reduction presented in Section 2.3. We use the *AES* testcase, set clock periods to $\{0.8, 0.9, 1.0\}ns$, run synthesis and S2D flow with netlists synthesized with the scaled WLMs, and then obtain 27 data training data points. As part of our model training described in Section 4.1, we comprehend WLM capacitance as a parameter. We now create a test dataset in which WLM capacitances are varied in steps of $0.05pF$ and choose the WLM $W_{best,model}$ that achieves the largest delta power from our models. We run synthesis and S2D flow with WLM $W_{best,model}$ and quantify the cost of misprediction with the WLM $W_{best,actual}$ that delivers the minimum 3D power after implementation using the S2D flow. Figure 12 shows how 3D power changes (albeit not too significantly) with WLM capacitance for the *AES* testcase. S2D always uses $1.0pF$, but the minimum power is achieved with WLM capacitance of $0.45pF$, and the model predicts $0.75pF$. The difference in S2D vs. our models is $\sim 1mW$ or $\sim 5\%$. We see that WLM scaling can achieve smaller 3D power, and that 3DPE models can guide the implementation to achieve within $\sim 1.62\%$ of the minimum power.

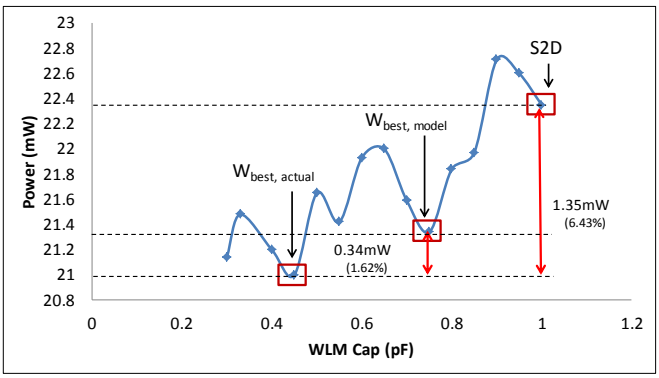


Figure 12: Percentage delta power benefits between actual, model and S2D implementations.

Experiment MGI-II. The goal of this experiment is to predict 3DIC power benefits (relative to 2DIC implementation) when the standard cell utilization is higher than the utilizations used in training the models. High utilizations in large designs such as *THEIA* incur large runtimes of around eight hours per data point. On the other hand, low-utilization runs can be fast but have smaller 3D benefit. Table 7 shows 3DPE modeling accuracy for different combinations of aspect ratios, clock periods and utilizations. The actual % delta power ranges from 1.82% to 2.88%. We implement the testcase with these parameters to quantify the modeling error. 3DPE can very accurately guide high-utilization design-space exploration because it is trained with small testcases (e.g., *AES* and *DCT*) at high utilization and is able to generalize to larger testcases.

5. CONCLUSIONS AND FUTURE WORK

It is difficult to quantify the benefits of 3DIC over a corresponding 2DIC implementation for arbitrary designs because no golden 3DIC flow currently exists. Yet, estimating 3DIC benefits, particularly for the power reduction value proposition, is a critical open issue. We develop a new prediction tool, 3DPE, to predict percentage delta power benefits of 3DIC relative to 2DIC implementation. 3DPE consists of internal, switching, leakage and total power models and these models are very accurate as the error range is bounded to be $\leq 10\%$. Such a tool is useful for designers because it filters out design blocks that can achieve large power benefits in 3D and performs fast design-space exploration to determine various 3DIC implementation parameters. We propose a novel modeling technique that includes WLM scaling, influential parameter identification and bounded errors. We present novel applications/validations that include "stress test" and "model-guided implementation" (MGI). We demonstrate how 3DPE can be used in MGI to predict power benefits of blocks that have high utilization and long runtimes, in a fast and accurate manner. Our ongoing works include: (i) assessing 3DIC benefits of sub-blocks within large designs; (ii) extending 3DPE from block-level to SoC-level; and (iii) developing a "true 3D" flow that iterates between 2D and 3D placements and uses 3DPE to guide the choice of implementation-space parameter values.

Acknowledgments

We thank Prof. Alex Zelikovsky of Georgia State University for discussions leading to the bound of Lemma 1. We also thank Prof. Sungkyu Lim, Shreepad Panth and Moongon Jung of Georgia Tech for their generosity with time and bandwidth so that their 2D, Shrunk2D and 3D flows could be transplanted, and results replicated, in our environment.

6. REFERENCES

[1] T.-B. Chan, K. Han, A. B. Kahng, J.-G. Lee and S. Nath, "OCV-Aware Top-Level Clock Tree Optimization", *Proc. GLSVLSI*, 2014, pp. 33-38.
[2] P. Batude, M. Vinet, A. Pouydebasque, et al., "Advances in 3D CMOS Sequential Integration", *Proc. IEDM*, 2009, pp. 1-4.
[3] P. Batude, T. Ernst, J. Arcamone, et al., "3-D Sequential Integration: A Key Enabling Technology for Heterogeneous Co-Integration of New Function With CMOS", *IEEE JETCAS*, 2(4) (2012), pp. 714-722.
[4] C.-B. Cho, W. Zhang and T. Li, "Thermal Design Space Exploration of 3D Die Stcked Multi-core Processors Using Geospatial-based Predictive Models", *Proc. SPEC Benchmark Workshop on Computer Performance Evaluation and Benchmarking*, 2009, pp. 102-120.
[5] W.-K. Mak and C. Chu, "Rethinking the Wirelength Benefit of 3-D Integration", *IEEE Trans. on VLSI* 20(12) (2012), pp. 2346-2351.
[6] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2009.
[7] M. Jung, T. Song, Y. Wan, Y.-J. Lee, et al., "How to Reduce Power in 3D IC Designs: A Case Study with OpenSPARC T2 Core", *Proc. CICC*, 2013, pp. 1-4.
[8] M. Jung, T. Song, Y. Wan, Y. Peng and S. K. Lim, "On Enhancing Power Benefits in 3D ICs: Block Folding and Bonding Styles Perspective", *Proc. DAC*, 2014, pp. 1-6.
[9] A. B. Kahng, "The ITRS Design Technology and System Drivers Roadmap: Process and Status", *Proc. DAC*, 2013.
[10] D. H. Kim and S. K. Lim, "Through-Silicon-Via-Aware Delay and Power Prediction Model for Buffered Interconnects in 3D ICs", *Proc. SLIP*, 2010, pp. 25-32.
[11] D. H. Kim, K. Athikulwongse, M. Healy, M. Hossain, et al., "3D-MAPS: 3D Massively Parallel Processor with Stacked Memory", *Proc. ISSCC*, 2012, pp. 188-189.
[12] D. H. Kim, S. Mukhopadhyay and S. K. Lim, "TSV-Aware Interconnect Distribution Models for Prediction of Delay and Power Consumption of 3-D Stacked ICs", *IEEE Trans. on CAD* 33(9) (2014), pp. 1384-1395.
[13] D. Milojevic, T. E. Carlson, K. Croes, R. Radojcic, et al., "Automated PathFinding Tool Chain for 3D-Stacked Integrated Circuits: Practical Case Study", *Proc. IEEE 3D System Integration*, 2009, pp. 1-6.
[14] S. Panth, K. Samadi, Y. Du and S. K. Lim, "High-Density Integration of Functional Modules Using Monolithic 3D-IC Technology", *Proc. ASPDAC*, 2013, pp. 681-686.
[15] S. Panth, K. Samadi, Y. Du and S. K. Lim, "Placement-Driven Partitioning for Congestion Mitigation in Monolithic 3D IC Designs", *Proc. ISPD*, 2014, pp. 1-6.
[16] S. Panth, K. Samadi, Y. Du and S. K. Lim, "Design and CAD Methodologies for Low Power Gate-level Monolithic 3D ICs", *Proc. ISLPED*, 2014, pp. 171-176.
[17] T. Thorolfsson, K. Gonsalves and P. D. Franzon, "Design Automation for a 3DIC FFT Processor for Synthetic Aperture Radar: A Case Study", *Proc. DAC*, 2009, pp. 51-56.
[18] F. Toufexis, A. Papanikolaou, D. Soudris, G. Stamoulis and S. Bantas, "Power, Performance and Area Prediction of 3D ICs During Early Stage Design Exploration in 45nm", *Proc. ICECS*, 2011, pp. 715-718.
[19] S. Panth, *personal communications*, 2014.
[20] "3D ICs with TSVs – Design Challenges and Requirements." `http://www.cadence.com/rl/resources/white_papers/3dic_wp.pdf`
[21] Synopsys 32/28nm Generic Library for Teaching IC Design. `http://www.synopsys.com/COMMUNITY/UNIVERSITYPROGRAM/Pages/32-28nm-generic-library.aspx`
[22] Synopsys EDA Tools. `http://www.synopsys.com`
[23] OpenCores. `http://opencores.org/projects/`
[24] OpenSPARC T2. `http://www.oracle.com/technetwork/systems/opensparc/index.html`