Received 28 April 2021; revised 15 June 2021; accepted 2 July 2021.
Date of publication 6 July 2021; date of current version 28 September 2021.
Digital Object Identifier 10.1109/JXCDC.2021.3095104

Machine Learning Integrated Pseudo-3-D Flow for Monolithic 3-D ICs

SAI PENTAPATI[1], BON WOONG KU[2], and SUNGKYU LIM[1] (Senior Member, IEEE)

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

CORRESPONDING AUTHOR: S. PENTAPATI (sai.pentapati@gatech.edu)

This work was supported in part by the DARPA ERI 3DSOC Program under Award HR001118C0096 and in part by the Semiconductor Research Corporation under GRC Task 2929.

ABSTRACT Monolithic 3-D (M3-D) IC design is a manufacturing technique that opens several new possibilities of chip design and exploration for power, performance, area (PPA), and cost benefits. Designing a commercially viable M3-D IC first requires a sign-off timing closure capability. Since the commercial tools lack such capability, several 3-D flows have been proposed that treat 3-D as a 2-D die and use commercial 2-D electronic design automation (EDA) tools for the RTL-to-GDS stage. The conversion between the two stages is done late in the design flow and the conversion is also nontrivial. Here, we propose a machine learning-based prediction algorithm to decrease the discrepancy between the pre and post-partitioned 3-D design using regression models. Our proposed model is circuit-agnostic and its performance with respect to a circuit-dependent model is also studied. Furthermore, more details on the behavior and analysis of the model are considered. Overall, we achieve a significant reduction in the total negative slack (TNS) of the test design ($3\times$–$16\times$) using the machine learning model integrated pseudo-3-D flow at an expense of just -1%–4% increase in total power.

INDEX TERMS Interconnect analysis, monolithic 3-D (M3-D) IC, pseudo-3-D Flows, tier partitioning.

I. INTRODUCTION

MONOLITHIC 3-D (M3-D) IC design is a process in which multiple dies/tiers are sequentially fabricated on top of each other [1], [2] to create a 3-D die stack. The sequential processing helps one to maintain a 3-D contact pitch $\sim$100 nm enabling a high connection density and bandwidth between the adjacent dies. The 3-D vias connecting the two tiers in an M3-D IC are referred to as monolithic intertier vias (MIVs).

The high bandwidth allows for fine-grained gate-level partitioning of an RTL which can lead to various implementations that target wire length reduction, memory latency reduction, etc. In this work, we focus on the gate-level 3-D implementations that target wire length reduction. A two-tiered gate-level 3-D integration has $(1/2)\times$ the footprint of a corresponding 2-D design, and $(1/(2)^{1/2})\times$ the linear dimensions. This leads to an average reduction in wire length of up to 30% in a 3-D design.

Commercial electronic design automation (EDA) tools do not support the design and optimization of an M3-D IC. Due to this, several pseudo-3-D flows are developed that make use of the EDA capabilities to create timing-optimized M3-D ICs. Compact-2-D (C2-D) [3] and Shrunk-2-D [4] are two such flows. A general pseudo-3-D flow follows three main stages to create a 3-D IC: pseudo-3-D design, tier partitioning, and MIV planning, 3-D routing. In the pseudo-3-D stage, a 2-D type floorplan is first designed leveraging the commercial EDA tool capabilities for placement, clock design, timing optimization. In shrunk-2-D, all the linear dimensions of the technology are scaled-down by $(1/(2)^{1/2})\times$ so that the footprint and the resultant x, y placement are more closely related to a 3-D design in the original unscaled technology. As distances between cells are also approximately $(1/(2)^{1/2})\times$ shorter on average, the parasitics replicate the 3-D parasitics and the timing optimization done in pseudo-3-D is useful for 3-D.

In C2-D, the technology dimensions are not scaled as it affects the metal cross section and *RC* parasitics. Instead, the design uses a larger footprint which is $\sim 2\times$ expected 3-D footprint, and the resistance and capacitance values are all derated by $(1/(2)^{1/2})\times$ to make sure the net parasitics in the derated pseudo-3-D stage are similar to the final 3-D stage. By being limited to a single active layer in pseudo-3-D, the conversion to 3-D stage requires placement modification and complete routing overhaul, and therefore affects the final 3-D design quality in terms of power, performance, area (PPA). The placement and routing inconsistencies also affect the shrunk-2-D based flow (although, in shrunk-2-D the technology dimensions are shrunk matching the footprint size to final 3-D).

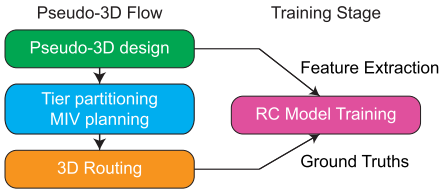

FIGURE 1. Pseudo-3-D flow and training from the corresponding features.

To tackle these problems in the pseudo-3-D flows, we first presented a machine learning integration framework in [5]. A high-level working of the flow is shown in Fig. 1. Better estimations of *RC*s early in the design stage will improve the final PPA in general. In the pseudo-3-D flows for 3-D ICs in particular, this *RC* estimation becomes more important as the timing optimization is exclusively done in the pseudo-3-D stage. The 3-D parasitics depend on a lot of variables as will be discussed later in Section II-B, and so a machine learning framework is well suited to learn the different interactions of the net features to estimate the 3-D R and C values. In this article, we extend the machine learning framework to include the following new items: 1) extend the framework with better testing and training methods; 2) comparing the performance of the circuit agnostic model with a circuit-specific model to understand the performance of these models and their interaction with the EDA tools; and 3) more statistically significant importance analysis of the features.

In this study, we use a total of 12 memoryless logic RTL downloaded from open cores, ISPD contests [6]–[8]. These are designed with a 28-nm commercial technology node. For all the 2-D designs, six metal layers are used for signal routing. 3-D designs have two tiers and a total of 12 signal routing layers (6 per each die). All the machine learning implementation are done with python3.6.

II. RC ANALYSIS

A. RC BREAKDOWN OF A NET

In a GDS layout of an RTL, the nets are complex 3-D structures whose parasitics depend on the exact shape of the net as well as the overall bottom-end-of-line (BEOL) dielectrics, neighboring nets. But such detailed analysis requires a significant amount of time for 3-D extractions and SPICE simulations. So, the commercial EDA tools use several assumptions and simplifications to achieve a trade-off between accuracy and run-time.

Consider a rectangular wire of width W, thickness T, length L, at a distance H from the ground plane. The resistance and ground capacitance are given by $C_{wire} = (\epsilon_d WL/H)$; $R_{wire} = (\rho L/WT)$, where ϵ_d is the dielectric constant of the dielectric between the wire and the ground plane. While the resistance model is fairly simple, the total capacitance is much more complex with contributions from the fringing effect of the ground capacitance, and coupling capacitance. Modern-day IC designs also use multiple layers of metals and the $W, T, H, \epsilon_d, \rho$ can be different for the different metal layers. In an EDA tool, it is not feasible to calculate these capacitances from just the physical dimensions, and material properties. So, an *RC* lookup table is provided by the technology foundry containing precalculated unit length ground capacitance, coupling capacitance, and resistance values of the wires (denoted by lower-case c, r in this article), and the vias at different scenarios (width, thickness, spacing, temperature, etc.) for each metal layer. The total resistance, and capacitance of a net with wires of length l_{M_i} on metal layer M_i, and n_{V_i} number of vias of type V_i is given by

$$C_{net} = \sum_{M_i} c_{M_i} l_{M_i} + \sum_{V_i} c_{V_i} n_{V_i} + XCap \tag{1}$$

$$R_{net} = \sum_{M_i} r_{M_i} l_{M_i} + \sum_{V_i} r_{V_i} n_{V_i} \tag{2}$$

where Xcap is the cross coupling capacitance between the net and the neighboring nets in the design. $c(r)_{M_i}$ is the capacitance (resistance) of a wire of length 1 μm, and $c(r)_{V_i}$ is the capacitance (resistance) of a via of type V_i. Note that there can be multiple types of vias from metal layer M_i to M_{i+1}. Coupling capacitance of the net is dependent on the final routing. While most of the nets have a negligible coupling capacitance, the nets that are routed in congested areas can have majority of the total capacitance as the coupling capacitance.

B. RC EVOLUTION FROM PSEUDO-3-D TO FINAL-3-D DESIGNS

As discussed in Section I C2-D's pseudo-3-D stage works under the assumption that the wire *RC*s scale down by a factor a $(1/(2)^{1/2})$ when the design is converted from pseudo-3-D to 3-D stage. So, the scaling factor is applied in the pseudo-3-D stage. This assumption is true in an ideal case, but the discrete row placement of cells and complex routing algorithms of commercial tools create variations in the scaling factor. Even in a global sense, the overall *RC* reduction is rarely as expected. Furthermore, the global scaling is applied only considering the length reduction portion in (1) and (2). A number of vias on a net is much harder to predict as the routing in pseudo-3-D (six metal layers total) and final-3-D (12 metal layers total) is very different. The contact resistance of vias keeps increasing in smaller technology nodes, and ignoring the via resistance on the overall resistance can cause inconsistencies in the resistance of nets from the pseudo-3-D to final-3-D stages. In the technology node considered here, the unit values for metal layer 4 are as follows: $c_{M_4} \approx 0.20\,\text{fF}\,\mu\text{m}^{-1}$, $c_{V_4} \approx 0.02\,\text{fF}\,\mu\text{m}^{-1}$, $r_{M_4} \approx 10.0\ \Omega/\mu\text{m}$, $r_{V_4} \approx 8.0\ \Omega/\mu\text{m}$. It is clear that the contact resistance is significant even in the relatively older 28-nm node, considering the total wire length and via count in Table 1.

To further analyze the *RC* evolution in design implementation, we design AES-128 circuit in pseudo-3-D and final-3-D stages. Some of the useful metrics from this implementation is shown in Table 1. Note that the wire length, ground capacitance, wire resistance are considerably underestimated in the pseudo-3-D stage. Via count increases by ~17.5%, but the global scaling $[(1/(2)^{1/2})]$ performed is suitable for a ~30% reduction in the via count. This increase in 3-D is due to the halved footprint (or a number of tracks per layer) and twice the number of vertical layers compared to a 2-D or pseudo-3-D implementation.

The scatter plot of the wire parasitics as a function of the routed wire length is in Fig. 2 visualizes a couple of trends.

TABLE 1. Routing statistics of the pseudo-3-D and the final 3-D routed designs.

	Pseudo-3D	Final-3D
Footprint (mm^2)	0.228	0.114
Metal Stack	6 Layers	12 Layers
Wire length† (µm)	1,141,179	1,234,332
Via Count	872,931	1,028,040
Ground Capacitance† (pF)	119.34	140.83
Coupling Capacitance† (pF)	35.20	31.71
Wire Resistance† (MΩ)	15.26	21.23

† Wire length, Capacitance, Resistance values of pseudo-3D are scaled by $\frac{1}{\sqrt{2}}$ to show a clear representation of the estimations

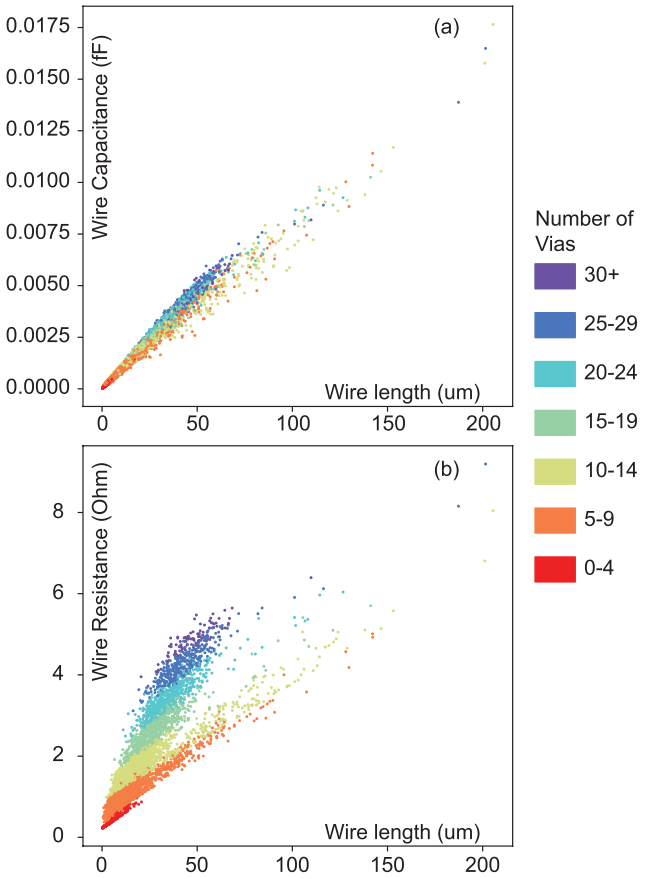

FIGURE 2. (a) Capacitance and (b) resistance of nets in the AES-128 design with respect to routed wire length of the net. The data points are color coded according to the number of vias on each net.

One, the wire resistance can be given by a set of linear functions of the wire length, whose slope is fixed and intercept increases with the number of vias on the net. Two, the via capacitance has a negligible impact on the total wire capacitance. Three, the resistance and capacitance are linear functions of the total wire length, i.e., they do not vary due to different distributions of the total routing on separate metal layers. This is due to the fact that the unit *RC* values for the one to six metal layers in the considered commercial technology node are very close to each other. The difference between routing in pseudo-3-D and final-3-D means that

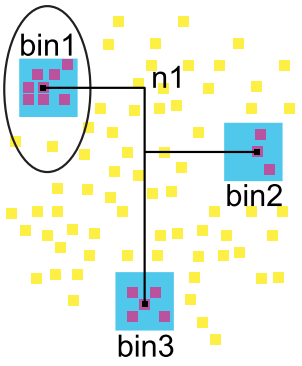

FIGURE 3. Net and its connected cells (smaller squares). The local regions at each endpoint of the net are shaded in blue (larger squares). bin1 has a high cell density and the contained cell will be displaced during legalization after tier partitioning.

the via count in pseudo-3-D cannot be directly used as a proxy for the final-3-D via count. But it provides a new point of information for the machine learning algorithm. In Section III-B, we show that via count indeed has useful information regarding final 3-D parasitics by verifying the null hypothesis probability.

From the stages shown in Fig. 1, the tier partitioning and 3-D routing change layout after the pseudo-3-D stage. Within tier partitioning, cell legalization is performed to get a clean placement solution in two tiers. As mentioned in Section I, the location mapping from a larger 2-D footprint to a halved 3-D footprint in C2-D creates cell overlaps. Additionally in this work, instead of legalizing the cells independently in each tier, we perform an incremental placement similar to the proposed solution in [9]. This allows for a better placement quality as die-by-die legalization does not consider the PPA impact. Based on cell placement and net connectivity, nets undergo different amounts of cell movement during the tier partitioning. Fig. 3 shows an example of a net and neighboring cells. Features such as the local density in a small neighborhood to the end-points of nets can be used to learn the extent of cell movement each net undergoes.

In the next stage of the pseudo-3-D flow—3-D routing—the routing of the nets is fully modified. Based on the routing, nets can be grouped into three different types: 1) single-tier nets; 2) multitier nets; and 3) nets with metal layer borrowing.

Single-tier nets are the ones where the 3-D routing is done entirely within the top or bottom tier. These are expected to undergo the least amount of change from the pseudo-3-D stage as they are still routed within the six signal routing layers.

Multitier nets are the nets connecting cells from different tiers after the partitioning. These have the highest difference in routing between the two design stages, as they need to be routed vertically for proper connectivity. These would have increased wire length and number of vias that will affect capacitance and the resistance of the nets.

Finally, the last group is the nets that use metal-borrowing. Consider a net that is connecting to cells entirely within the top tier. When performing 3-D routing, some of these nets can use metal layers belonging to a different tier. This is called metal layer borrowing, and such nets would have a medium variation in the parasitics in 3-D.

To quantify the impact the tier partitioning and 3-D routing have on the nets and to understand the extent of this impact from different net features, the following metric is useful:

$$\frac{\text{res(cap) of net(s) in pseudo-3-D}}{\text{res(cap) of the net(s) in final-3-D}} = \text{Scaling Error of } R, C$$

represented as $SE_{R(C)}$. This shows how well the scaling in pseudo-3-D corresponds with the final 3-D RCs.

Scaling error can also be defined for a group of nets by using the sum of R and C values in the numerator and denominator of the fraction. $SE_{R(C)} \approx 1$ of a group of nets implies that the scaling done in the pseudo-3-D stage is close to accurate for the group.

$SE_{R(C)} \ll 1$ for a group of nets means that the estimation in pseudo-3-D is much lower than the final 3-D for this group. This results in worse timing after the design is 3-D routed. Identifying such a group of nets using a combination of net is useful in properly applying scaling factors. Most of the nets in a design would fall in this group as the pseudo-3-D is usually optimistic.

$SE_{R(C)} \gg 1$ occurs when the parasitic value in pseudo-3-D is overestimated. Cells on these nets would be oversized in pseudo-3-D. These cells not only consume additional but also manifests as an additional capacitance load to the connected cells.

Grouping the nets using net metrics like routed wire length, fan-out/number of connected cells, number of MIVs, local cell density near the cells, we plot the scaling error variation of these groups in Fig. 4.

In Fig. 4(a), the nets are grouped based on the routed wire length. All the nets with wire length $[x, x+1)$µm are put into group x. Based on this, the average scaling error of R, C, and the MIV counts are plotted. The number of MIVs is dependent on many different features such as cell count, overall connectivity graph, and bin size chosen. In order to observe the impact of just wire length, the other features are kept constant for this plot by only considering nets with fan-out 2 in the implementation of aes-128 with fixed bin-size. This shows how wire length can impact the 3-D routing (specifically the average number of MIVs). The average number of MIVs in each group increases as the pseudo-3-D wire length keeps getting higher. Net groups with at least 1000 nets are considered to reduce volatility in the plot.

More importantly, we observe the scaling error of resistance and capacitance of these groups. This follows a slightly more complex trajectory. The scaling error plots are much smoother by the virtue of the central limit theorem as we consider significantly more number of nets. It is interesting to note that none of the groups have an average scaling error >1 in line with our claim that pseudo-3-D underrepresents the final RC values. Misrepresenting the via calculation causes the resistance to be significantly undervalued. Overall, the scaling error versus wire length plots have two main trends: a steep increase at lower values $0 \leq x < 10$ followed by a saddle-like shape for $10 < x < 60$.

At $0 \leq x < 10$, the scaling error values are the most <1. Since these nets are smaller, small perturbation during legalization and routing changes can cause a relatively significant increase in the final parasitics and so the $SE < 1$. As the nets become relatively large, the net are more likely to be partitioned (as evident from the avg. MIV count plot) and 3-D routing is now going to have a higher impact adding more RCs in the final 3-D that was unaccounted for. This shows us up as the decrease in SE. And as the net length keeps on increasing, the pseudo-3-D RCs increase at a rate higher than the impact of 3-D routing, so the SE increases again. This interaction between the pseudo-3-D RCs and the additional 3-D touring manifests as the saddle shape. With relatively low noise, this allows us to learn a scaling model as a function of routed wire lengths.

Extending a similar analysis to the number of cells connected to a net versus $SE_{R(C)}$ gives us the plot in Fig. 4(b). These plots are close to monotonically decreasing. This shows that a highly connected net is more likely to have a higher discrepancy between the pseudo-3-D and final-3-D stages. This is a direct result of the bin-based Fiduccia-Mattheyses partitioning done in pseudo-3-D flow.

a: BIN-BASED FIDUCCIA-MATTHEYSES PARTITIONING

In this partitioning, placement layout is first divided into smaller rectangular bins and then each bin is partitioned into two tiers such that the area of cells in the two tiers is the same within a tolerance threshold. In any hypergraph partitioning, the nets with high fan-out are more likely to be partitioned. For example, a net with "c" cells connected has 2^c ways being split into two partitions. Apart from the two solutions where all the cells in either of the partitions, the other $2^c - 2$ solutions have a cut-size of 1 net. So, once such a net is forced to not be partitioned, the solution space decreases by a fraction of $\sim 2^c$. But allowing the net to be partitioned leaves the solution space almost unchanged with different configurations achieving the same result. So, it is not a good move to keep a highly partitioned net constrained to a single partition as we might miss the chance of finding a better cut-size solution. So, a highly connected cell is more likely to be partitioned under hypergraph partitioning.

With this knowledge of hypergraph partitioning, it is easy to see the reason for the monotonically decreasing plots in Fig. 4(b). As discussed in the wire length analysis, a partitioned net will have increased parasitics in 3-D and since the groups are not directly dependent on wire length, the numerator (pseudo-3-D parasitic value) cannot compensate for the increase in the 3-D parasitics unlike in Fig. 4(a).

The scaling error due to tier partitioning can also be seen by grouping nets based on the number of MIVs. This is a final 3-D metric and cannot be used in training. But this helps one to understand the scaling error evolution in terms of grouping based on final 3-D parameters. No MIVs on a net imply that it is fully routed in a single tier, and as discussed earlier, such nets have the least deviation from the ideal. This is seen in Fig. 4(c) as the $SE_C \approx 0.95$, $SE_R \approx 0.80$ is highest at #MIVs $= 0$. Both SE_R and SE_C have a significant drop when #MIVs $= 1$. This is because nets with fewer MIVs also have a smaller avg. WL (Fig. 4). When the nets are partitioned its adds vertical routing in 3-D which is particularly significant for small nets. And as resistance is especially large for vias, this causes the significantly large drop for SE_R.

Finally, the number of dense bins per net is analyzed. As discussed in earlier sections, as the number of dense bins per net increases, the legalization distorts the net more. When this value is 0, $SE_C = 1$ showing the group of nets that is closest to ideal in terms of pseudo-3-D scaling. In this case,

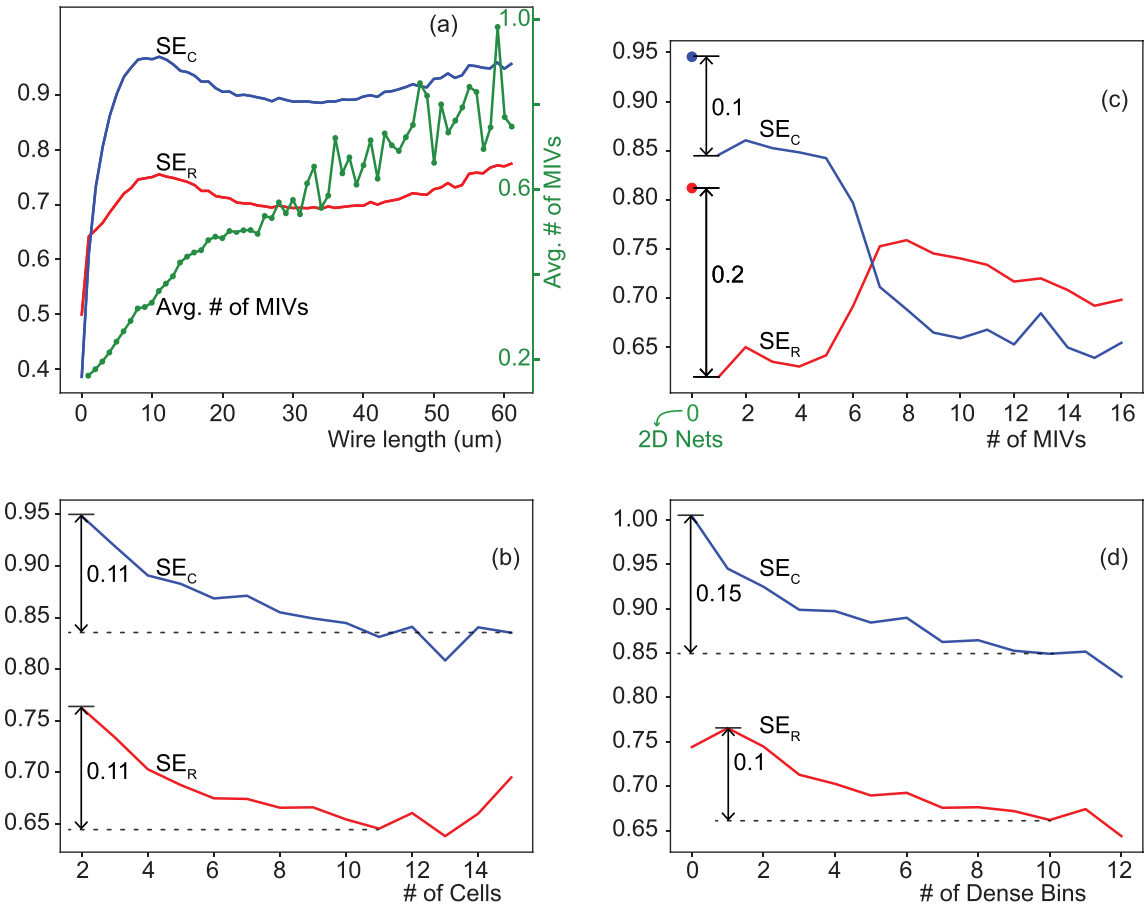

FIGURE 4. **Average *RC* scaling errors with respect to various net features. (a) Wire length. (b) Number of MIVs on the net. (c) Number of cells connected to the net. (d) Number of dense bins of the net.**

each bin at the endpoint is a square centered at the pin with a side of size 3 × Row Height. A bin is considered dense if the density is greater than a certain threshold (which is set to 75%). This again shows a monotonic drop with the range almost as large as the trend for cell count.

III. DESIGN AND LEARNING MODEL IMPLEMENTATION

A. DESIGN IMPLEMENTATION

Training machine learning models require input data to train and the output labels as the target output of the model. In our case, the input data comes from pseudo-3-D design, specifically, the postroute stage of the pseudo-3-D design. This is specially chosen as we can extract proper pseudo-3-D parasitic values as well as routing metrics like wire length via count of nets. We add a few improvements to the 3-D stages of the flow using improvements suggested in [9]. This allows for better legalization with full 3-D connectivity and 3-D congestion-driven placement, and the routing is done with a complete metal stack. This is useful in our model application as the target parasitic value extraction becomes more streamlined. With an independent die-by-die routing, net extraction should be performed for each die separately. By using the full 3-D routing (routing both dies together), net extraction becomes more streamlined leveraging in-built query commands of the commercial EDA tools. So, the target data are the parasitic values of the 3-D routed nets done with full 3-D routing.

The inputs to the model are the net features and design features from the pseudo-3-D stage. Net features differentiate the nets, whereas the design features are different among the type of designs. All the different features used are specified in Table 2. The evolution of a net from pseudo-3-D to final-3-D is design-dependent and, wire dominant and cell dominant designs have very different routing and therefore very different net evolution between the two stages. Similarly global nets that span over a huge fraction of the chip width or height would have varying lengths as the design varies. Similarly, even when a circuit is fixed, different frequency implementations will add additional timing constraints to routing and additional buffers required for timing closure. And finally, tier partitioning has an important variable that changes the partitioning type-bin width. The bin width is used to define bins within which FM min-cut is performed. So, a high bin count (small bin width) can increase the number of nets partitioned. In order to generalize over all these variations, we chose eight different training netlists, four target frequencies per circuit, three bin sizes per frequency totaling 96 implementations for training. Different circuits have a different number of nets, and to make sure some large circuits do not overwhelm the training process, the number of nets per design is capped at a certain value. Four circuits are left out at the training stage that are used during testing.

Fig. 5 shows the overall flow after integrating the *RC* prediction. After the routing stage in pseudo-3-D, pretrained

TABLE 2. Input features used to train the XGBoost model, and their importance and/or explanation.

Feature Name	Significance
	Individual Net Features
Wirelength	Long, medium, short wires have varying average $SE_{R(C)}$
HPBB	Wirelength estimation from placement; not affected by congestion
Net connectivity	Number of cells connected by the net; estimates partitioning probability
Via Count	Number of vias on the net; useful for resistance calculation
Wire Cap	capacitance in pseudo-3D design, has strong effect on final capacitance
Wire Res	resistance in pseudo-3D design, has strong effect on final resistance
Average Local density	Avg. density of the bins of a given net as shown in Fig. 3; quantifies legalization errors
Dense Bins	Number of dense bins of each net; useful for deciding legalization errors
	Full Chip Features (Design Identifiers)
Total WL	Total routed signal wirelength, design identifier
Number of nets	Total number of nets in the design
Number of cells	Total number of cells in the design
Average Fanout	Number of cell pins/ number of nets; design connectivity information
Chip Area	Footprint of the design
Cell Area	Total standard cell area in the design
Utilization	Standard cell density
Bin Count	Number of partitioning bins to be used
Design id	Design name represented as an integer using simple hash function
	Derived Features
net WL/ total WL	Identifying global nets among various designs; detouring, cross coupling capacitance information
HPBB / $\sqrt{\text{Chip area}}$	Identifying global nets among various designs, less affected by pseudo-3D congestion
Wire Cap / Total Cap	Fractional capacitance of net w.r.t. total, importance of net within a design
Wire Res / Total Res	Fractional resistance of net w.r.t. total, importance of net within a design
net WL / Bin size	For a given wirelength, a increase in bin-size would decrease the probability of net being partitioned
HPBB / Bin size	Similar to net WL/ Bin size, but only considers placement information
HPBB + 0.1*VC	A combined HPBB, via count feature

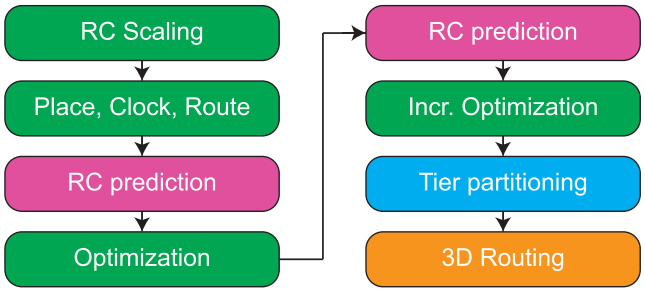

FIGURE 5. Pseudo-3-D flow with integrated *RC* prediction results.

models are loaded and used for *RC* value predictions. These values are annotated to the existing nets and the design is optimized. But postroute optimization has the ability to change the net connectivity and add/delete nets. Completely constraining the net updates at this stage will impact the optimization quality. So, during optimization, nets that do not have the *RC* annotations from the trained model would be introduced to the design by the EDA tool for timing closure. To rectify this, an incremental optimization is performed by reannotating the parasitic values to all the nets in the design followed by an in-place optimization. In this stage, the nets are cells that are fixed in position to minimize the changes to placement and routing structure that could invalidate the *RC* annotation.

B. MACHINE LEARNING MODEL

To find the best learning model, we first consider a couple of different options during the training stage: 1) an XGBoost regressor; 2) XGBoost random forest regressor; and 3) random forest regressor. We chose XGBoost as it is well suited for regression-type problems. *RC* estimation is formulated as a linear regression problem in our work with the least squared sum loss model. Random forest regressors are closely related to the XGBoost regressors, and having more weak tree learners in a forest could be helpful to avoid overfitting data to a single design or net type. Within each model, various hyperparameters are varied and twofold cross validation is used to choose the best combination. Specifically, they are the number of trees, number of features per tree, and max depth of each tree. In general, it is better to have many weak trees to avoid overfitting. Similarly, by randomly choosing a subset of all the features in each tree, the model can be generalized better and performs well with test sets. Moreover, before training any model, we perform feature selection on the 24 selected features. By removing the unnecessary features, overfitting in tree learning models can be avoided. This is done using backward feature selection and null hypothesis test using linear regression.

In backward feature selection, a linear regression model is first fitted based on an initial set of features. p-values are extracted for the features and at each stage, the worst features are discarded if its p-value > 0.05. p-value shows the probability of the null hypothesis, i.e., the probability that a feature does not contribute to the target. In capacitance model

TABLE 3. *R2* **scores and mean squared error of C2-D scaling and the best model for capacitance training.**

	C2D - R2	ML - R2	C2D - MSE	ML - MSE
cordic	0.8284	0.9261	3.00e-7	1.30e-7
des	0.8480	0.9324	6.40e-7	2.80e-7
edit-dist	0.9247	0.9578	1.16e-6	6.50e-7
fpu	0.9289	0.9729	1.38e-6	5.30e-7
ldpc	0.9544	0.9714	3.55e-6	2.22e-6
leon3mp	0.9360	0.9697	3.41e-6	1.62e-6
matrix-mult	0.9208	0.9376	1.17e-6	0.92e-6
netcard	0.9810	0.9912	2.85e-6	1.33e-6
Test Set				
b19	0.9072	0.9704	5.40e-7	1.70e-7
ecg	0.9305	0.9719	6.90e-7	2.80e-7
tate	0.8889	0.9517	1.56e-6	0.68e-6
vga	0.9633	0.9717	3.93e-6	3.03e-6

TABLE 4. Permutation importance using RMSE loss of the eight most important features per model. The RMSE loss of resistance model is 7.31×10^{-2} Ω, and capacitance model loss is 9.22×10^{-4} fF.

Resistance Model		Capacitance Model	
Feature	Importance	Feature	Importance
Via Count	3.08e-1	Wire Cap	4.48e-3
Wire Cap	1.34e-1	Wire Res	4.55e-4
Pin Count	4.16e-2	Wire Length	4.46e-4
Wire Length	3.63e-2	Via Count	6.98e-5
hpbbXwidth	4.47e-3	Pin Count	4.10e-5
hpbbXvc	4.45e-3	hpbbXwidth	9.27e-6
Dense Bins	3.94e-3	viaXhpbb	3.30e-6
Res/Total Res	3.50e-4	Chip area	2.46e-6
# nets	2.66e-4	hpbb	1.82e-6

training, the "design id" was the only feature that had a p-value greater than 0.05 and is removed during model learning. In the resistance model, the wire resistance was the only feature with p-value > 0.05. This is an extremely unintuitive outcome and happens because the influence of vias made the wire resistance less useful and redundant. Changing the design significantly changes the routing patterns (metal usage per layer, vias used, etc.), which has more impact on the resistance model. So "design id" has a smaller p-value and is kept in resistance modeling.

IV. RESULTS

A. TRAINING AND INFERENCE

With the data collected from the previous stages, twofold cross-validation is used on the testing sets and the model with the best cross-validation error is selected. The *R*2 score (explained variance) of the designs is shown in Table 3. Of note are the designs with a low *R*2 score. When *R*2 score equals 1, the model predicts all the variations in the target label. *R*2 score is zero when output is a constant equal to the target mean. The *R*2 score is higher in both training and testing netlists, although the increase per netlist varies. Netlists like vga, ldpc, and netcard already have a relatively high *R*2 score in C2-D stage. Interesting to note is that these three are the wire dominant designs in the netlists considered. So, the slight variances due to 3-D routing are not very significant relative to the overall wire length. MSE is also high for these circuits due to their large parasitics.

In Table 4, the permutation importance is given for the top ten most important features in the model on the test set. In permutation importance, a loss metric is first calculated using a trained model, and then a feature is permuted so that values

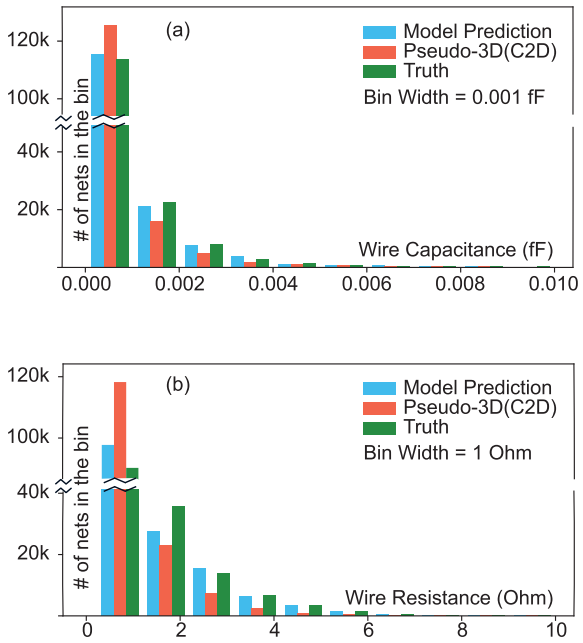

FIGURE 6. ***RC*** **histograms of ML based implementation.**

of the feature are incorrect and the score is recalculated. The difference between the scores is the permutation importance. A large value implies a feature of high importance and a small value implies low importance.

Fig. 6 shows the *RC* histograms of three design stages: Model predicted *RC*s, pseudo-3-D *RC*s, and true 3-D *RC*s. We see that the pseudo-3-D *RC*s deviate more from the ground truth in the lower *RC* ranges leading to smaller *RC*s than final 3-D. Such a design would not meet timing when *RC*s becomes worse in the actual 3-D stage.

B. FULL-CHIP PPA

Irrespective of *RC* prediction, PPA is the most important consideration for a full chip study, and in Table 5 we report the PPA of the four testing netlists used: b19, ecg, tate, vga. b19, and vga are relatively small circuits, and ecg and tate are significantly larger. Three different implementations of PPA are reported: 3-D design using circuit agnostic machine learning model, circuit-specific model, and the C2-D's global *RC* scaling model.

The capacitance error in each implementation is the difference between the total capacitance of its final 3-D and pseudo-3-D stages. −ve capacitance error means that the pseudo-3-D stage predicts a low capacitance value. C2-D flow always underestimates the value leading to worse slack overall. With machine learning models, the error is always positive and slightly in the positive direction. For circuits with large over or underestimates, the power consumption varies based on the method chosen (although this is a very small % of total power). For the small circuits, such as b19 and vga, the power consumption is actually smaller as the machine learning model is employed. This is because the accurate parasitic estimation allows for better optimization in the pseudo-3-D stage.

Total negative slack (TNS) has the highest impact overall as it becomes $\approx 3\times - 16\times$ smaller with the circuit agnostic

TABLE 5. Overall PPA of the test netlists with circuit agnostic ML scaling, C2-D scaling, circuit specific ML scaling models.

		units	b19	ecg	tate	vga
	Frequency	MHz	1250	1500	1000	1250
	Chip Area	μm^2	29446.4	75130.8	116690.2	45624.6
	Cell Count	μm^2	37976	95768	155521	35613
	Cell Area	μm^2	42756.0	107115.5	233599.8	66400.9
	WL	m	0.316	0.851	1.401	1.260
Circuit Agnostic	Cap Error	%	1.22	9.77	2.78	5.36
	Total Power	mW	39.3	107.0	135.0	28.3
	WNS	ns	-0.091	-0.401	-0.441	-0.176
	TNS	ns	-13.1	-67.5	-58.5	-3.1
Compact-2D	Cap Error	%	-6.59	-3.05	-10.23	-3.84
	Total Power	mW	39.8	104.7	131.4	28.4
	WNS	ns	-0.109	-0.282	-0.764	-0.193
	TNS	ns	-80.6	-197.8	-356.1	-48.5
Circuit Specific	Cap Error	%	-0.70	5.96	1.87	2.77
	Total Power	mW	38.9	106.9	135.0	28.1
	WNS	ns	-0.068	-0.409	-1.108	-0.150
	TNS	ns	-15.6	-100.9	-84.5	-10.5

model compared to the C2-D scaling. This is mainly because each net is assigned a better parasitic which contributes a little to the overall slack improvement. Worst negative slack (WNS), on the other hand, depends on just the critical nets in 3-D. These nets are not identifiable from C2-D stages, as the 3-D routing changes the criticality of the paths. The nets with negative timing slack in pseudo-3-D can have a positive slack, and nets with positive slacks in pseudo-3-D can become the new critical paths in final 3-D. To identify the critical paths, a learning model that is tailored for the timing estimation of graph networks should be used and requires high accuracy. But the best solution would be to add a postroute optimization stage in 3-D that can fix the relatively minor TNS and few WNS violations in 3-D.

Finally, we also trained circuit-specific models to check the benefit of having such models. While these models were able to perform better in terms of total capacitance error, they do not have significant improvement in power consumption. With regards to timing, the circuit-specific models have even more volatile WNS. The TNS is better than the C2-D scaling but is not as good as using a general model. This is slightly counter-intuitive as we would assume circuit-specific model would perform better. But they suffer from overfitting and lack of different types of nets during training. Another reason is that these models are trained to learn the parasitics and cannot be directly helpful with overall timing. So the general learning model is as good as circuit-specific models for parasitic estimation and power consumption, but is significantly better in terms of timing.

V. CONCLUSION

In summary, we have presented a machine learning model that can predict final net parasitics at an early stage of the design. We have analyzed several net features and how they impact the parasitic evolution in a pseudo-3-D flow. We formulate new metrics and use them to achieve better circuit, agnostic learning models. Using these models, we were able to achieve higher $R2$ score, lower MSE, better timing. We discussed the issue of critical path estimation in 3-D design and showed that our general model is better than a circuit-specific model. With $3\times-16\times$ TNS reduction on test circuits, integrating these models in the pseudo-3-D flows help us to minimize the number of timing violations and the severity of the violations after routing.

VI. ACKNOWLEDGMENT

Bon Woong Ku was with the Georgia Institute of Technology, Atlanta, GA 30332 USA.

REFERENCES

[1] L. Brunet *et al.*, "Breakthroughs in 3D sequential technology," in *IEDM Tech. Dig.*, Dec. 2018, pp. 7.2.1–7.2.4.

[2] M. M. Shulaker *et al.*, "Three-dimensional integration of nanotechnologies for computing and data storage on a single chip," *Nature*, vol. 547, no. 7661, pp. 74–78, Jul. 2017.

[3] B. W. Ku, K. Chang, and S. K. Lim, "Compact-2D: A physical design methodology to build two-tier gate-level 3-D ICs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 6, pp. 1151–1164, Jun. 2020.

[4] S. Panth, K. Samadi, Y. Du, and S. K. Lim, "Shrunk-2-D: A physical design methodology to build commercial-quality monolithic 3-D ICs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 10, pp. 1716–1724, Oct. 2017.

[5] S. S. K. Pentapati, B. W. Ku, and S. K. Lim, "ML-based wire RC prediction in monolithic 3D ICs with an application to full-chip optimization," in *Proc. Int. Symp. Phys. Design (ISPD)*, Mar. 2021, pp. 75–82.

[6] *Opencores Bemchmarks*. Accessed: Jul. 11, 2021. [Online]. Available: https://opencores.org/projects

[7] M. M. Ozdal, C. Amin, A. Ayupov, S. Burns, G. Wilke, and C. Zhuo, "The ISPD-2012 discrete cell sizing contest and benchmark suite," in *Proc. ACM Int. Symp. Int. Symp. Phys. Design*, 2012, pp. 161–164.

[8] M. M. Ozdal, C. Amin, A. Ayupov, S. M. Burns, G. R. Wilke, and C. Zhuo, "An improved benchmark suite for the ISPD-2013 discrete cell sizing contest," in *Proc. ACM Int. Symp. Int. Symp. Phys. Design*, 2013, pp. 168–170.

[9] S. S. K. Pentapati, K. Chang, V. Gerousis, R. Sengupta, and S. K. Lim, "Pin-3D: A physical synthesis and post-layout optimization flow for heterogeneous monolithic 3D ICs," in *Proc. IEEE/ACM Int. Conf.Comput. Aided Design (ICCAD)*, Nov. 2020, pp. 1–9.