**SLIDE 1: TITLE**

Good morning everyone. My talk is on “New Applications of Learning-Based Modeling in Nanoscale Integrated-Circuit Design”.

**SLIDE 2: OUTLINE**

My talk is structured into three thrusts. Thrust 1 presents three new works on design productivity gains through improved design- and implementation-space exploration.

Thrust 2 presents one new work on improved accuracy of electrical modeling. Thrust 3 presents three new works to optimize design power, energy, project management, and cost.

Thrusts 1 and 2 are related to learning-based modeling, whereas Thrust 3 changes the envelope of what is being modeled or predicted.

**SLIDE 3: PD PICTURE**

Works presented in each thrust can be placed in various parts of IC design flow figure. The two works listed on top apply to system-level optimization and scheduling of design infrastructure. Majority of the works are in the area of physical design that include floorplanning, placement, Clock tree synthesis and routing.

**SLIDE 4: RECAP OF UQE**

Works presented at UQE can be placed under each thrust as follows.

**SLIDE 5: TODO’S FROM UQE**

Out of the TODO’s from UQE, three are done and two are not done due to changed research directions. These include prediction of embedded memory timing failures at floorplan stage, “true 3D” placement, prediction of routability and #metal layers, and reliability-constrained multi-core task scheduling.

**SLIDE 6: NEW AFTER UQE**

New works after UQE are highlighted in blue and are the focus of today’s talk.

**SLIDE 7: PREDICTION AND OPTIMIZATION CONNECTION**

Even though Thrust 3 has no learning-based modeling, yet it is connected to the topic and other two thrusts due to two reasons.

Better prediction leads to more accurate constraints, e.g., upper bounds or requirements, during optimization. Thus, our optimization solutions are better. For example, using past project requirements, we can create a model to predict storage upper bounds. Using requirements from new project, we estimate the current upper bounds and then optimize schedule and resource allocation. Clearly, if the upper bounds are incorrect, the optimization solutions will be incorrect.

Better optimization also leads to better flows, for example, and changes the envelope of what is being predicted or modeled. For example, an optimized 3D flow enables us to obtain accurate ground truth and realistic models.

**SLIDE 8: PRELIMINARIES: Terminologies**

This slide shows preliminaries that define some IC-design related terminologies used in the rest of this talk. These include design space, implementation space, IC blocks, commercial EDA tool examples, and design infrastructure. I also provide definitions of other jargons in the handout.

**SLIDE 9: LIST OF PUBLICATIONS (USED IN THESIS)**

Here is a list of publications used in the thesis. The ones in blue are used in Thrust 1, ones in brown are used in Thrust 2 and the ones in green are used in Thrust 3. Full list of all my used and unused publications are in my webpage.

**SLIDE 10: MOTIVATION: VALUE SCALING GAP**

As part of our groups work on roadmapping effort for International Technology Roadmap for Semiconductors or ITRS, we observe that lithography has continued to deliver “available” Moore’s Law scaling of transistor density growing at 2x/node. However, the “realized density” scaling has slowed down to 1.6x/node roughly around 2009. The picture on the top-right shows this gap which we at UCSD refer to as the “design capability gap”.

Overall, there is slowdown of power-performance-area-cost (PPAC) from process and device scaling. Designers spend area, power and performance resources on reliability, variability, etc. for nanoscale IC design technologies. So, even 10% or 20% improvement in PPAC is a huge deal.

The picture below shows how resources spent on guardbands lead to lost benefits in performance, power, etc. In this work, we develop a chain of models to reduce these large guardbands by enabling incremental PD optimizations.

**SLIDE 11: MOTIVATION: HIGH COSTS, TURNAROUND TIMES**

What prevents design- and implementation-space exploration today?

This figure shows that EDA tool license costs are very high. The design cost of a SOC consumer portable chip in 2013 is $45M. Further, there are no systematic methodologies to enable designers to perform DSE or ISE. To perform DSE with EDA tools implies long runtimes and the exploration is a highly iterative process. In this work, we develop a chain of models that enable fast and accurate DSE, ISE.

**SLIDE 12: OUTLINE**

In Thrust 1, I present the first work related to early-stage slack prediction of embedded memories. This work has been presented in ASP-DAC 2016.

**SLIDE 13: KEY TAKEAWAYS**

Timing closure is time-consuming and complex at advanced nodes as it increases turnaround time. Early prediction of slack can reduce design cost and turnaround time. This problem is difficult because floorplanning with SRAMs is complicated due to congestion, power delivery etc. Also, Multiphysics effects make the problem harder. We describe a novel learning-based methodology to address this problem.

**SLIDE 14: TIMING PRELIMINARIES**

Before I describe the prediction problem and our solution, I will provide background terminologies.

This slide shows timing preliminaries. Setup time is the time for the data to stabilize before the clock edge. Transition or slew time is the time a rising signal takes to transition from 10% voltage to 90% voltage. Load is the sum of pin and wire capacitances. Gate or cell delay is the propagation delay through standard cell input pin to output pin.

**SLIDE 15: PATH SLACK PRELIMINARIES**

Arrival time is the time signal takes to travel from clock pin of launch flip-flop to D-pin of capture flip-flop. It depends on cell and wire delays in the path. Required time is the clock period minus the setup time. Slack is the difference between required and arrival times. In this figure the critical path is shown in grey.

**SLIDE 16: SOC FLOORPLAN**

Floorplan is an arrangement of various blocks in a SoC – processor cores, GPU cores, caches, etc. The left figure is the floorplan of Apple A6 found in iPhone 5. It is fabricated in 32nm and has a die size of 97 mm2. The right figure is the floorplan of Apple A7. It is fabricated in 28nm and the die size is 102 mm2.

**SLIDE 17: CHALLENGE: SENSITIVITY OF SLACK TP SPACING BETWEEN MEMORIES**

Now that I have explained background, I go to the problem of early-stage prediction of timing slack of embedded memories. I discuss two modeling challenges for this problem.

Our experimental results show that the spacing between memories affects post-P&R and multiphysics slack values. The figure in the bottom-left shows our design with five SRAMs placed in the top-left corner of the placement region. The other corners contain other placement and routing blockages; hence standard cells can only be placed in the cross-shaped region in the center of the block. We vary the spacing between SRAMs, that is, the space between these blue-striped boxes in steps of 10 micron from 10 micron to 30 micron.

The slack difference between SRAMs can be larger than 300ps as shown in the bottom-right plot. This is due to congestion and buffer placement. Hence, slack values vary in non-obvious and/or noisy manner when the spacing is changed.

**SLIDE 18: CHALLENGE: ABSTRACTION OF P&R STAGES AND TOOL NOISE**

Another challenge is modeling must abstract multiple stages of the P&R flow as shown in the figure below. EDA tool noise is another factor our modeling must comprehend. As our goal is to predict post-P&R and multiphysics slack values at the floorplan stage, the modeling must therefore comprehend effects of placement to routing, extraction, STA, etc.

Our goal is to derive an approximation function “f” that comprehends the combined effects of the netlist, constraints, P&R stages and tool noise.

**SLIDE 19: MULTIPLE PHYSICS: MAKES PROBLEM HARDER**

We define multiphysics STA as performing STA with more than one physics such as IR, thermal, crosstalk, etc.

Design teams can achieve more accurate timing results by closing multiphysics analysis loops. The figure here shows slack of two SRAMs. The deep blue bar on the left shows slack with no IR is positive 480ps on SRAM #1. By including static IR analysis, the light blue bar here shows that slack become positive 250ps, that is, more pessimistic. By including, dynamic IR analysis, the slack for the same SRAM becomes 20ps as shown by the blue-green bar on the left. We also demonstrate here that more than one dynamic IR loop, shown by the yellow, red and brown bars, can slightly reduce pessimism of only one loop of dynamic IR analysis by 25ps. This is because timing windows change. The key message here is, including more than one physics not only makes the timing analysis more accurate, but also more pessimistic.

However, predicting multiphysics slack at early design stages is very challenging as shown in this plot. In the X-axis, we have 50 different implementations of the same design in which we vary the clock periods and transition time constraints. The red line shows SRAM slack without IR and the blue line shows SRAM slack with dynamic IR. Modeling such a non-uniform trend is difficult.

**SLIDE 20: MULTIPHYSICS ANALYSIS FLOW**

Here, I describe our multiphysics analysis flow. We have developed this flow with significant guidance from our industry colleagues. In this work, we consider only IR drop and crosstalk and perform these analysis using RedHawk and PrimeTime-SI tools, respectively.

The inputs to PTSI are SDC, Verilog netlist, Liberty DBs, parasitic SPEF. PTSI generates a .timing file that contains timing windows of transition times of each pin in the netlist. The input to RedHawk consists of this .timing file from PTSI, Liberty .lib files, DEF, SPEF and technology files. RedHawk generates an IR drop report per instance in the netlist.

We perform STA again using this IR drop map and go around this loop four times. It is possible to include other physics such as temperature and reliability, which we have not explored in this work.

**SLIDE 21: FLOORPLANNING AND SRAM PLACEMENT**

To explore which design parameters affect post-P&R and multiphysics slack, we conduct multiple experiments by varying the floorplan and power delivery network or PDN contexts. This figure shows how we have parameterized our floorplan and SRAM placement. The blue-striped boxes represent SRAMs, the blue box represents buffer screens and green boxes represent blockages that emulate other SRAMs.

We vary the core width and height, SRAM width, height and spacing, blockage width and height, width and height of the routing channels and buffer screen widths.

**SLIDE 22: LIST OF PARAMETERS**

Here is a complete list of our modeling parameters. The prefix “N” in the 1st column denotes a netlist-related parameter. The prefix “FP” denotes a floorplan-related parameter, and the prefix “C” denotes a constraint-related parameter.

The netlist parameters are labeled from N1 through N7, the floorplan parameters are labeled from FP1 through FP11, and the constraints are labeled from C1 through C9. In total, we have 27 parameters.

**SLIDE 23: MODELING TECHNIQUES AND FLOW**

We extract our parameters from netlist, netlist sequential graph, floorplan context and constraints. We obtain ground truth from P&R and multiphysics STA reports. We normalize all our data points to within a range of 0 and 1, both inclusive.

We use one linear – LASSO with L1 regularization and three nonlinear – SVM with RBF kernel, ANN with one input, one output and two hidden layers and Boosting with SVM as weak learners.

We combine the predictions from each of these models using weights. We use a weighting strategy so that when actual negative slack values are predicted as positive in a data point, we retrain our model by increasing the weight for the data point by five times.

We perform five-fold cross-validation during our training phase for each of the modeling techniques.

**SLIDE 24: BOOSTING WITH SVM**

We briefly describe our Boosting implementation with SVM with RBF kernel. This is a new implementation and our contribution.

We build a cascade of weak SVM learners. The learning is weak because we terminate the grid search of hyperparameter values when the predicted slack is within 20% of clock period. Without Boosting, we exit when the predicted slack is within 5% of the clock period.

We adjust the weights at each stage based on the error observed for each data point. The newly weighted data point is an input to the next stage. As we cascade through the stages, errors become small.

We use 40 stages because our experimental results show that beyond 40 stages, there is no significant improvement in error.

We combine the output of each stage using a linear regressor to determine the predicted outputs from Boosting.

**SLIDE 25: POST-P&R SLACK PREDICTION**

I present results of post-P&R slack prediction here. We have a total of 2515 data points. We use 60% of these for training and validation, and the remaining 40% for testing.

This plot shows the effect of our weighting strategy for data points with negative slack values. We show the actual slack values in the X-axis and the error of slack prediction in the Y-axis. When the actual slack is less than -100ps, the error of slack prediction is always negative.

These two plots show our modeling accuracy for post-P&R slack. In the left plot, actual slack is in the X-axis and predicted slack is in the Y-axis. The solid black line denotes perfect correlation. The right plot shows a histogram of error of slack prediction. The solid vertical yellow line denotes zero error. The worst-case error is 224ps and the average error is 4ps. Even though this is very early-stage prediction, the results are surprisingly accurate.

**SLIDE 26: MULTIPHYSICS SLACK PREDICTION**

These plots show the predicted versus actual multiphysics slack values. Recall that we perform multiphysics STA analysis using PTSI by annotating IR drop values for each cell from our RedHawk analysis. The worst-case and average errors are 253ps and 9ps, respectively. These values are larger than post-P&R predictions because predicting multiphysics slack is harder than predicting post-P&R slack.

**SLIDE 27: MODELING FIDELITY**

Here, we show the confusion matrix of our predictions of multiphysics slack on the test set. We report common classification metrics in ML literature, that is, false positives, false negatives, precision and recall. False negatives are 3% of the data points, that is, our model suggests a floorplan needs to be changed when it is actually not required. False positives are 4% of data points, that is, our model deems a floorplan to be good when it is actually bad and can lead to timing failures on SRAMs.

For the data points with positive slack, our recall is 95% for 93% precision. For the data points with negative slack, our recall is 90% for 92.5% precision. We believe our model can provide guidance to designers with high fidelity. Even though this is very early-stage prediction, the results are surprisingly accurate.

**SLIDE 28: SUMMARY**

In conclusion, we note that early stage prediction of timing failure is important and timing closure with multiple analyses are important for complex SOCs for accuracy and faster design turnaround time.

We present a machine learning-based methodology to predict post-P&R and multiphysics slack values within a worst-case error of 253ps.

**SLIDE 29: OUTLINE**

The second work in Thrust 1 predicts 3DIC benefit from 2DIC implementations and has been presented at DAC 2015.

**SLIDE 30: KEY TAKEAWAYS**

3DICs continue Moore’s Law trajectory of value scaling and are fundamental to “More than Moore” idea. Power benefit is the key value proposition of 3D but no tool predicts 3D power benefits from 2D implementations. The problem is difficult because 3D benefits vary with netlist topologies, constraints; implementation space is high-dimensional; lack of a golden 3D flow and a chicken-and-egg loop of trying to embed netlists not created for 3D into 3D. A higher-level chicken-and-egg loop is until people are convinced about 3D benefit, no investment on 3D tool and flow developments will be made. Therefore, a benefit estimation tool is required.

Our solution is to develop a novel learning-based 3D power estimation tool (3DPE) to address this gap.

**SLIDE 31: TOOL COMMANDS AND OPTIONS**

I provide some background terminology now.

Commercial EDA tools provide a plethora of commands and options. Cadence Innovus is a P&R tool. For each physical design stage, the tool provides multiple commands. Synopsys Primetime is a signoff tool and version J-2014.12 has 448 commands. These tool knobs along with constraints make the implementation-space high-dimensional.

**SLIDE 32: SHRUNK2D (S2D): OUR BASELINE 3D FLOW**

This figure shows a classic 2DIC. The height and width of the die are H and W, respectively. When this implementation is changed to 3DIC, we can create two vertically stacked dies. The height and width of each die is divided by the square root of two with respect to those of the 2DIC. The dies are interconnected by vertical interconnects.

Shrunk2D is another way to emulate a 3DIC by doing P&R in a die with the same height and width of the 3DIC. This flow proposed by Panth et al. at Georgia Tech and is the strongest academic “3D” flow today.

**SLIDE 33: “TRUE 3D” OBJECTIVE in APlace3D (A3D)**

For years, our group and collaborators at Qualcomm used the S2D flow as a golden 3D implementation flow. Recently, the APlace implementation from 2004 timeframe and extended it to 3D by a postdoc in 2010. We revisited the implementation and submitted APLace3D (A3D) to ASPDAC-2017 with the postdoc as our co-author. A3D is interesting because it achieves significantly better QoR than S2D and provides opportunity for better predictions – of a 3D power estimation tool here and for routability in 3D that we will see in the next topic.

APlace3D implements a true 3D objective. The figure shows cartoon of a two-tier 3DIC. The bottom tier is Tier 0 and top tier is Tier 1. We also show placement of 4 pins A, B, C, D of a net. Pins A and B are on Tier 0 and pins C and D are on Tier 1.

Our true 3D objective is the weighted sum of half-perimeter wirelength or HPWL0 of the bounding box of pins A, B on Tier 0 and HPWL1 of the bounding box of pins C and D, both shown in green color. The third term is the HPWL of the union of all 4 pins when the two tiers are overlapped. We call this the HPWLov. W1, W2, W3 are user-defined weights. We assume that no signal net crosses between two tiers more than once. This objective comprehends minimization of HPWL on each tier along with the HPWL when both tiers are overlapped.

**SLIDE 34: APLACE3D FLOW**

Our APlace3D flow using commercial P&R and signoff tools is as follows.

We place flip-flops, macros and PI/POs. We perform clock tree synthesis at this stage because we adopt the split at sink strategy proposed by Panth et al. We then invoke A3D to perform 2-tier placement of standard cells. We legalize the placement in a commercial tool. Next, we add VIs a dummy cells in the netlist and invoke A3D again to perform 3-tier placement. We fix the placement of standard cells on the top-most and bottom-most tiers and allow only VI cells to move in the middle tier.

Next, we change the VI cells to PI/POs and place them on each tier. We then perform tier-by-tier legalization, routing and optimization in a commercial P&R tool. Finally, we perform timing and power signoff in a commercial signoff tool.

**SLIDE 35: COMPARISON OF 2D VS. A3D VARIANTS**

We now present our results with the A3D flow. Here we compare 2D and the three variants of A3D, namely, Gordian-L in 3D (A3D-GL3D), weighted wirelength (A3D-WWL) and true 3D (A3D-T3D).

This plot shows normalized WL wrt 2D in the y-axis in 28nm FDSOI. WL reductions are shown by these arrows and the numbers next to them.

This plot shows normalized WL in 28nm LP.

This plot shows normalized power in 28nm FDSOI and this plot shows normalized power in 28nm LP.

Overall, we achieve up to 31% WL reduction and 20% power reduction compared to 2D.

**SLIDE 36: COMPARISON S2D VS. A3D VARIANTS**

In this slide, we compare the strongest academic flow, S2D with the three variants of A3D.

This plot shows normalized WL wrt 2D in the y-axis in 28nm LP. WL reductions are shown by these arrows and the numbers next to them.

This plot shows normalized WL in 28nmFDSOI.

This plot shows normalized power in 28nm LP and this plot shows normalized power in 28nm FDSOI.

Overall, we achieve up to 24% WL reduction and 12% power reduction compared to S2D.

**SLIDE 37: IMPLEMENTATION-SPACE PARAMETERS AND TESTCASES**

Now that we have discussed background and baseline 3D implementation flows, I describe the 3D power benefit estimation problem further. Modeling is difficult due to the high-dimensional space of implementation parameters.

Here is list of various implementation-space parameters we use in our experiments. The parameters span across various constraints, layout contexts and technology choices.

**SLIDE 38: FLOW AND TOP “10” PARAMETERS**

To restrict the dimensionality and runtime of our modeling problem, we seek to explore the 10 most influential parameters.

In our flow, we use engineered WLMs to perform synthesis. Then, we perform both 2D, A3D and Shrunk2D P&R. **We use S2D as a proxy for 3D**.

For both P&R flows, we use scaled RC cap tables. We then extract parameters for modeling.

The top-10 parameters include six constraints such as clock period, max transition time, etc. We also use two implementation and two technology parameters such as utilization, multi-Vt libraries, respectively.

**SLIDE 39: MACHINE LEARNING METHODOLOGY**

With parameters extracted from 2DIC implementation, we perform modeling. We use artificial neural networks to capture the complex interactions between parameters.

We define the ANN architecture with one input and one output layer, plus two hidden layers. We search for the best number of the epochs of back propagation and the number of neurons per layer using the loop here to achieve bounded errors.

We obtain our ground truth from S2D runs.

**SLIDE 40: MODEL ESTIMATE OF DELTA POWER**

We use a wide range of IPs that resembles building blocks of modern SoCs. The table shows the list of our testcases. We use five types of testcases -- CPU, GPU, modem, multimedia and peripheral engine.

This plot shows the actual percentage delta power benefit in the X-axis and the predicted values in the Y-axis for the five types of testcases. We derive separate models for each of the power components – internal, switching and leakage. Then we compose these models to create a model for total power.

We challenge ourselves to predict delta power. Across all our test data points, the worst-case error is 4.8% with S2D flow. With A3D flow, the worst-case error is 5.04% (-5.04%, 4.83%).

**SLIDE 41: MODEL VALIDATIONS**

We do not have ground truth from true 3DIC implementations, so we must test if our 3DPE models are capable of returning unlikely predictions.

We perform “stress testing” of the models. We perform Monte Carlo-like simulations by varying the mean and variance of each parameter in the models.

The figure shows a histogram of percentage predicted delta power. The maximum value is 39% for data points that are practically realizable.

We reject data points that are not practically realizable. For example, data points in which the number of cells, utilization and the cell area are mismatched. Or, the wirelength and the number of cells are mismatched.

**SLIDE 42: MODEL-GUIDED IMPLEMENTATION**

The hypothesis here is 3DPE should guide implementation if the predictions are reliable. We refer to this as model-guided implementation. We test this hypothesis with an implementation here. This figure shows WLM cap in the X-axis and 3D power in the Y-axis. Minimum 3D power is achieved at 0.45pF. Our models predict the cap to be 0.75pF, using which the delta power is 0.34mW. Therefore, 3DPE model guidance is better than S2D by 5%.

**SLIDE 43: SUMMARY**

In summary, power reduction is a key value proposition for 3DICs. Lack of a golden 3D flow makes prediction of 3D power benefits a difficult problem.

We develop the 3DPE tool that predicts the percentage delta power benefits of 3DIC relative to 2DIC implementations. 3DPE is accurate within 5% error.

We also propose stress testing and model-guided implementation approaches with 3DPE.

**SLIDE 44: OUTLINE**

I move on to the last topic in Thrust 1 – Back-end-of-Line or BEOL stack-aware routability prediction from placement. This has been accepted at ICCD 2016 recently.

**SLIDE 45: KEY TAKEAWAYS**

Physical design is complex in advanced nodes due to multiple complex design rules that must be satisfied before tapeout. Currently, PD engineers use congestion maps to predict routability. This is largely an “art” because congestion maps alone cannot predict routability. Furthermore, due to lithography constraints, each metal layer is a sizeable percentage of the wafer cost. If the #metal layers, need to be increased due to incorrect routability prediction, the cost to company will be high.

My work identifies new parameters that comprehend design rule violations or DRCs and enable accurate routability prediction for 2D and 3D ICs using a learning-based methodology. I also demonstrate a novel prediction of Pareto frontiers of max achievable utilization, aspect ratio, #metal layers at iso-performance.

**SLIDE 46: TECHNOLOGY LIBRARIES**

Before I describe the problem further, I provide background. .

Here, we show dimensions of a minimum-sized 2-input NAND cell from 28nm FDSOI libraries. The tightest metal pitch is 0.1 micron for layer M2. The cell comes in 2 sets of track heights – 8T and 12T. Both have 2 Vt flavors – low and regular. The height of the 8T cell is 0.8 um, that is, 8 tracks multiplied by M2 pitch. The height of the 12T cell is 1.2um, that is, 12 tracks multiplied by 0.1um. The widths of the cell are different in 8T and 12T libraries.

The 12T cells also have multiple channel lengths as shown in the bottom table.

**SLIDE 47: MACROS AND VIOLATIONS**

This is the floorplan of Opensparc T2 spc design. The brown blocks are the SRAM macros and the blue rectangles are the standard cells. The white crosses denote violations of design rules flagged by the P&R tool. Most of these violations appear in the channel between macros, which indicates the channel heights must be increased or the macro orientations / placement must be changed.

**SLIDE 48: CONGESTION MAPS CAN BE MISLEADING**

Now that I have described background, I describe the problem of routability prediction further.

The figure on the left shows congestion map at placement stage of a real design. The white regions show low congestion and red regions show high congestion. The figure on the right shows #DRCs at routing stage as white crosses. Many highly congested regions result in few (< 10) DRCs. Note that we can obtain #DRCs only at the routing stage, and not before.

This figure on the left shows congestion map of another design and the figure on the right shows the #DRCs. In this design, few highly congested regions result in many DRCs.

Making predictions of routability by only looking at the congestion map is incorrect.

**SLIDE 49: NEW PARAMETERS**

We identify new parameters from only placement (i.e., no early or trial routing) that correlate well with DRCs. We divide the placement region into grids of 45 x 45 tracks and correlate our new parameters and DRCs within each grid.

Here, we show two such parameters. The top figure shows that sum of incoming and outgoing hyperedges or nets correlate well with #DRCs. It indicates that large #pins of a net within a grid cause DRCs. The bottom figure correlates minimum proximity of any pair of pins to DRCs. It indicates when pins of adjacent cells are too close to each other, then DRCs can occur.

**SLIDE 50: LIST OF MODELING PARAMETERS**

We list all our modeling parameters here. We divide the placement region into 45 x 45 track grids and obtain max, min and coefficient of variation statistics of the following: pin density, min proximity, #complex cells, sum of edges, #buried nets, etc. To differentiate between placements, we also use utilization, aspect ratio, clock period as parameters. To comprehend various BEOL stacks, we use #horizontal and vertical tracks as parameters. We also parametrize routing resources used by the power deliver network and density of vertical interconnects in 3DICs.

**SLIDE 51: INTERPOLATION, EXTRAPOLATION ALGORITHM**

To predict Pareto frontiers from few, i.e., less than 20 placements, we develop an algorithm to interpolate and extrapolate parameter values. We pick three placements and obtain our list of modeling parameters. We pick another placement that we use to fit the model. We use weighted responses from SVM regression and MARS here. We then iterate to improve the model until the modeling error is smaller than an upper bound or we have used up all the 20 placements for modeling. Using this model, we now make predictions of parameter values, given inputs such as utilization, clock period, etc.

**SLIDE 52: PREDICTION OF ROUTABILITY IN 2D (28FDSOI, 8T)**

Now we present our accuracy in predicting if a placement is routable in 28nm FDSOI and 8 track libraries. In the training set, we have 906 routable and 471 unroutable datapoints. In the testing set, we have 1597 routable and 503 unroutable datapoints. The top table shows the confusion matrices of the training and testing sets. The bottom table shows metrics of classification. In the testing set, our accuracy is 87%. Even though the number of unroutable datapoints are few, recall is 90% at 92.7% precision and the negative prediction value is 71%.

**SLIDE 53: PREDICTION OF PARETO FRONTIER**

Here we present predictions of Pareto frontiers of max achievable utilization, aspect ratio, #metal layers at iso-performance, given only 20 placements.

These are the Pareto frontiers of the ARM CORTEXMO design in 2D in 28nm FDSOI. The left figure is the ground truth or actual frontier and the right figure is the predicted frontier. At AR of 1.8, actual says 79% is the max utilization with 5 metal layers, whereas we predict that 78% is the max utilization. So, we are pessimistic by 1%.

These are the Pareto frontiers of the ARM CORTEXMO design in 3D in 28nm FDSOI. For 3D, we have the A3D implementation flow. We show the frontiers of Tier 0 here. The left figure is the actual frontier and the right figure is the predicted frontier. At AR of 2.0, actual says 89% is the max utilization with 5 metal layers, whereas we predict that 88% is the max utilization. So, we are again pessimistic by only 1%.

**SLIDE 54: SUMMARY**

In this work, we demonstrate that new modeling parameters are needed to predict routability, given BEOL stack-aware placement. We devise an interpolation and extrapolation algorithm to predict Pareto frontiers. Our methodology is applicable to 2D and 3D ICs with classification accuracy of greater than 86% across multiple designs and technologies. Our predictions of max utilization are within 2% of the actual max utilization.

**SLIDE 55: OUTLINE**

In Thrust 2, I present one new work that predicts interconnect coupling delay and transition effects. This has been presented at SLIP 2015.

**SLIDE 56: KEY TAKEAWAYS**

Many tools perform timing analysis in both signal integrity or SI mode and non-SI mode. Runtimes in SI mode are 3x as large as runtimes in non-SI mode, also the slack divergence between SI and non-SI is large. The modeling problem is difficult because timers use proprietary timing engines. Alignment of aggressor-victim windows can be complex. We propose learning-based modeling of electrical signals at the gate- and wire-level using a layered modeling approach to address these issues.

I provide only salient highlights of this work as I had presented a very similar flavor of this work in UQE. But, this is a harder problem as we use the same tool for correlation.

**SLIDE 57: SI VS. NON-SI GAP IN STA TOOLS**

Let’s examine how much do SI and non-SI reports of the STA tool differ. This plot shows path slack in SI mode in the X-axis and path slack in non-SI mode in the Y-axis.

The divergence can be as large as 81ps in the critical path at a clock period of 1.0ns.

81ps is equivalent to 4 stages of logic in 28nm, leading to 20% performances difference. This is equal to one node of Moore’s scaling.

**SLIDE 58: SI TO NON-SI CALIBRATION USE CASE**

Instead of using STA tools with SI capability, we can first generate non-SI timing reports, and then use SI to non-SI calibration model to generate SI timing reports. In this way, we do not need STA tool with SI capability

Thus, we only need to spend 100K dollars and maybe 2 hours/per design, instead of 250K and 6hours per run, to get the precision of SI timing analysis.

**SLIDE 59: LIST OF MODELING PARAMETERS**

We develop three models based on our observation of modeling parameters. We first model incremental transition time. Then we model incremental delay due to SI based on transition time model and other parameters. Finally we model SI-aware path delay base on incremental delay model and other parameters.

Here we list all the parameters that are used for modeling, and their respective sources.

These include electrical layout logic structure parameters and constraint. These can be obtained from spef, non-SI report, library and SDC files.

**SLIDE 60: ACCURACY OF PATH DELAY PREDICTION**

We show the path delay prediction. Worst-case absolute error is 8.2ps and the average absolute error is 1.7ps

**SLIDE 61: ROBUSTNESS OF MODELS**

The previous results show acceptable correlations, but what about the ability to predict the unseen data? We implement a JPEG design with different clock period, number of stages and utilization, and we assess prediction of our models on this implementation.

The x-axis is the actual SI incr delay and the y-axis shows the predicted SI incr delay.

The worst-case absolute error is 7.9ps which is 12.3% of the actual SI incr delay, while the average absolute error is 1.6ps, 2.6%.

This indicates that our model can predict other implementations, thus our model is not overfitted.

**SLIDE 62: SUMMARY**

Calibration of non-SI to SI enables cost and runtime saving for SoC design. Because of this, we analyze electrical logic structure and layout parameters between non-SI and SI modes, in order to model the SI from non-SI. We develop machine learning-based models to accurately calibrate non-SI to SI timing.

We achieve worst-case error of 8.2ps in 28nm foundry FDSOI technology. Recent validations on 7nm technology achieve similar quality of predictions. Further validations are ongoing to test robustness of models.

**SLIDE 63: OUTLINE**

Thrust 3 of my thesis presents three works on optimizations of design power, energy, project management, and cost. First, I present a new analytic 3DIC placement that uses a new “true 3D” objective that reduces design power. This work has been submitted to ASP-DAC 2017.

**SLIDE 64: KEY TAKEAWAYS**

As mentioned in Thrust 1, 3DICs offer substantial scaling possibilities for the semiconductor industry. But, no golden 3D flow and tooling exists today. The strongest academic flow, Shrunk2D or S2D, invokes commercial 2D P&R and partition the netlist using minimum #cuts. Our new analytic placement tool, APlace3D or A3D, implements a true 3D objective. Our solutions are routable in a commercial P&R tool and we perform signoff timing and power analyses.

A3D enables better prediction of Pareto frontier of max utilization, aspect ratio and #metal layers in 3D and 3D power benefit.

**SLIDE 65: SUMMARY**

3D is a promising technology but it calls for new true 3D implementation flows. We propose a new analytic 3D placer whose solutions are routable in a commercial P&R tool. We achieve significant WL and power reductions compared to 2D and S2D. Compared to 2D, A3D achieve 31% reduction in WL and 20% reduction in power. Compared to S2D, A3D achieves 24% reduction in WL and 12% reduction in power.

We have validated A3D on multiple technologies and designs.

**SLIDE 66: OUTLINE**

The second work in Thrust 3 is about optimizations of project management and cost. This work has been submitted to TODAES 2016.

**SLIDE 67: KEY TAKEAWAYS**

Large IC design companies spend 100s of millions of dollars per year on design infrastructure to meet tapeout schedules for multiple concurrent projects. The problem is difficult because there are many types of resources, they are limited and must be shared across projects. There are also complex co-constraints between resources. For example, at most two cores can be used for every license of PTSI.

We develop two new mixed integer-linear programming formulations that extend the resource constrained project scheduling problem formulated and solved by Kolisch et al. We achieve significant cost and schedule savings compared to solutions adopted by a top-5 IC design company. We refer to this company as Company X. Better predictions lead to better optimizations as constraints to the optimizer can be predicted accurately (e.g., storage, engineers)

**SLIDE 68: CHALLENGE: MULTIPLE RESOURCE TYPES**

Before describing the problem, I provide background.

IC design companies typically working on multiple simultaneous projects e.g., application processor, audio DSP, video DSP, etc.

Each project has multiple activities such as synthesis, verification, place-and-route, extraction, timing analysis, etc. Each activities consumes one or more resources such as compute cores, storage, memory, EDA tool licenses, engineers, etc.

Co-constraints can exist between resources, e.g., at most 2 compute cores can be used for every PTSI license.

Resources can be of 3 types.

Fully-shared resources are shared across all projects from a common pool. Examples include compute cores in a datacenter.

Segregated or dedicated resources are allocated exclusively to a specific project. Examples include storage and engineers.

Conditionally-shared resources are allocated to each project, but any resource unused by a project may be used by other projects. Examples include EDA tool licenses, engineers.

**SLIDE 69: EXAMPLE: INFEASIBLE ALLOCATION**

Here is an example of infeasible allocation of resources across three projects A, B and C. Each project has one activity and one resource type. The upper bound of fully shared resources is 20. The upper bounds of number of segregated resources projects A, B and C are 5 each. The upper bounds of number of conditionally-shared resources for projects A, B and C are 5 each, as well.

The number of fully shared resources consumed by each project is 3, 2 and 1, respectively, so the sum is less than 20. The number of segregated resources consumed is also less than the upper bounds. The number of conditionally-shared resources consumed by A is 0, by B is 2, but by C is 6. This violates the upper bound. The number of unused conditionally-shared resources consumed by C is 9, which again violates the upper bound of 5.

Conditionally-shared resources are allocated to each project, but any resource unused by a project may be used by other projects. Examples include EDA tool licenses, engineers.

**SLIDE 70: EXAMPLE: FEASIBLE ALLOCATION**

Here is an example of feasible allocation of resources for the same example in the previous slide.

The number of fully shared resources consumed by each project is 3, 2 and 6, respectively, so the sum is less than 11. The number of segregated resources consumed by C is 5. The number of unused conditionally-shared resources consumed by C is 5. So, all constraints are satisfied.

**SLIDE 71: SCHEDULE COST MINIMIZATION (SCM) FORMULATION**

Now that I have described background, I describe our formulations.

The objective of our schedule cost minimization or SCM problem is to minimize the total cost or the sum of schedule penalties of all projects. This is subject to constraints on start and finish times, activity precedence, max number of resources, resource requirements. These are similar to the RCPSP formulation. New are resource co-constraints, stability in resource allocation constraints and tethering forecast resource allocation constraints.

The complexity is O(4NKJ(i)T) variables, where N is the #projects, J(i) is the #activities for each project I, K is the #resources and T is the max duration over all projects.

**SLIDE 72: RESOURCE COST MINIMIZATION (RCM) FORMULATION**

The objective of our resource cost minimization or RCM problem is to minimize the total #resources and cost of all projects. This is subject to constraints on start and finish times, activity precedence, max number of resources, resource requirements. New is the stability in resource allocation constraints.

The complexity is O(NKJ(i)T) variables.

**SLIDE 73: USE CASE: HANDLING LATE-BREAKING BUG**

Using the instance from the previous slide, we demonstrate application to another use case. Project 2 has a late-breaking bug in activity 7, so it delays all activities from 8 to 11. Company X’s solution took 41 extra days to complete. However, our MILP solutions take 34 extra days to complete. That is, we save 7 working days of 1.4 work-weeks. This is significant because Moore’s Law advances by 1% per week.

**SLIDE 74: USE CASE: SCHEDULING TETHERED TO FORECASTS**

Here, we demonstrate another use case in which scheduling is tethered to forecasts. This picture shows resource allocation for one project. The picture on the right shows an allocation of 3 projects. The allocation is infeasible because the #servers during weeks 30-36 exceed the datacenter capacity. Company X could not solve this problem and ended up purchasing 600 additional servers.

The figure at the bottom shows our MILP solution. Our solution meets the schedule without the need for any additional servers. We save cost of 600 servers.

**SLIDE 75: USE CASE: HUMAN RESOURCE ALLOCATION**

Here is a RCM use case using an instance from Company X which has 4 projects, 8 activities per project, four types of engineers and max duration of 16 work-weeks. This figure shows the activity precedence of these 8 activities. The table on the right shows billable man-weeks for each project and activity.

The plot below compares MILP and Company X’s solutions. We reduce #max resources by 37.5%. At a non-US location, the total cost-savings is $5.2M within ½ year project scheduling makespan.

**SLIDE 76: SUMMARY**

Lack of project management tools can impact a IC company’s bottom line by many millions of dollars annually. We develop new formulations by extending the RCPSP formulation. Our solver can be applied for fiscal planning and for “what-if” analyses by program management. Application to industrial instances results in significant cost and schedule savings.

**SLIDE 77: OUTLINE**

The third work in Thrust 3 optimizes design performance. This work has been presented at ISQED 2014.

**SLIDE 78: KEY TAKEAWAYS**

Reliability is a key processor design consideration at advanced nodes to guarantee system lifetime. No existing scheduling policy guarantees both acceptable performance and acceptable throughput. This is because optimal solution requires new ways to perform discretized exhaustive search.

We develop a maximum-value reliability-constrained overdrive frequencies or MVRCOF formulation that guarantees both acceptable performance and acceptable throughput.

**SLIDE 79: RELIABILITY IN MULTI-CORE SYSTEMS**

Modern multicore processors, such as the Intel Xeon, AMD Athlon, etc. operate at multiple operating modes to meet different performance and power requirements. For example, nominal, supply voltage scaling and turbo.

Task scheduling affects how each core in a multicore processor is used. A subset of cores can fail before others.

Applications have different requirements of the number of cores to use. The operating system scheduler packs tasks from applications using some or all of the available processing cores.

The figures show how Applications A on the top and B at the bottom have different requirements for the number of cores during its execution phase.

The figure on the right shows percentage active time on the Y-axis and number of active cores in the X-axis for Application A in blue and Application B in maroon. The green bars show how cores are used when the scheduler packs tasks in a eight-core system. The core usage is roughly a Gaussian distribution with the mean being the total of the available cores divided by two.

**SLIDE 80: OPTIMAL, HEURISTIC VS. BASELINE (RC-LG)**

Here, I present only our key results. This figure shows the objective function value in Y-axis and the testcases in the X-axis. The figure compares the objective function values across optimal, heuristic and the baseline solutions.

Our optimal solutions achieve up to 17.4% higher value of objective function as compared to baseline solutions as shown by the arrows and percentage numbers in white. Our heuristic solutions can be up to 3.3% worse than our optimal solutions as shown by the arrow and percentage number in blue.

**SLIDE 81: SUMMARY**

We formulate and solve a new MVRCOF problem under lifetime reliability constraints.

We develop a MVRCOF solver that implements our optimal and heuristic flows.

Our solutions guarantee both “acceptable performance” and “acceptable throughput” and we empirically demonstrate that our optimal solutions can achieve up to 17.4% greater value in objective function as compared to baseline RC-LG solutions.

**SLIDE 82: PERSPECTIVES: MAINSTREAM VS. PIONEERING**

We classify works presented into mainstream and pioneering. Mainstream works make improvements of existing methods or flows. Pioneering works provide a novel way of approaching the problem.

Mainstream works include routability prediction, 3DIC, project management and task scheduling. In each of these, this thesis makes several new contributions.

Pioneering works include ORION3.0 and signoff timing correlation. In ORION3.0, we demonstrate a novel modeling approach at the architecture-level using post-P&R data to fit models. In signoff timing correlation, we demonstrate approaches to modeling circuit phenomena and model-guided PD optimizations.

**SLIDE 83: PERSPECTIVES: NEW THINKING AND FOLLOW-ON**

Works in this thesis that are suitable for follow-on research include integrating signoff timing correlation models with timing optimization loops, modifying formulation of SCM so that timesteps can change automatically when schedule changes occur, and iterative optimizations to obtain more stable solutions. MVRCOF formulation can be applied to traces of workload from datacenters and solutions can be validated in a datacenter LSF framework. The Pareto frontier prediction work can be extended to perform SoC-level Pareto prediction.

This thesis provides new directions / thinking through some works. Early-stage slack prediction may be adopted for early-stage prediction of power, DRCs, IR drop or other design QoRs. New thinking is also required to reduce the “trial and error” nature of model fitting, e.g., using grid search, incremental modeling, etc. Possibly domain understanding can help here. Signoff timing correlation can be extended to correlate power, reliability. Project management formulations can spur thinking in robustness of scheduling in the face of stochasticity in resources and personnel.

**SLIDE 84: CONCLUSIONS**

In conclusion, this thesis presents wide-ranging opportunities for learning-based modeling in IC design and shows new applications using the three thrusts. We attack qualitatively new challenges and present innovations in modeling and optimization methodologies.

**SLIDE 85: ACKNOWLEDGMENTS**

This thesis would not have been possible without the constant guidance, mentoring and support of my advisor, Prof. Kahng. I thank my committee members – Professors Cheng, Lin, Rao and Saul. I would like to thank my labmates for all their help and support. I thank my mentors, Prof. Lin, Dr. Samadi and Dr. Chan. I would also like to thank all my co-authors, and Shrunk2D authors at GT.