Use of Machine Learning in Radio Frequency Integrated Circuits (RFIC) Development

by

Qiang Cui

Ph.D. Electrical Engineering, University of Central Florida (2013)
M.E. Electronics Science and Technology, Zhejiang University (2008)
B.E. Electronics Science and Technology, Zhejiang University (2006)

Submitted to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

at the

Massachusetts Institute of Technology

February 2020

© 2020 Qiang Cui
All rights reserved

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature of Author________________________ Signature redacted

Qiang Cui
System Design and Management Program
January 28th, 2020

Signature redacted

Certified by________________________

Tomas Palacios
Thesis Supervisor
Professor, MIT Electrical Engineering and Computer Science

Signature redacted

Accepted by ________________________

Joan Rubin
Executive Director, System Design & Management Program

MIT LIBRARIES
MAY 0 5 2022
RECEIVED
ARCHIVES

This page intentionally left blank

Use of Machine Learning in Radio Frequency Integrated Circuits (RFIC) Development

by
Qiang Cui

Submitted to the System Design and Management Program on January 28th, 2020, in Partial Fulfillment of the Requirements for the Degree of Master of Science in Engineering and Management

Abstract

This Master's Thesis starts with an introduction to the radio frequency integrated circuits (RFICs) industry and a discussion on the key problem of the existing RFIC development process: the need for multiple trial and error iterations due to inaccurate simulations. This simulation inaccuracy happens because the existing electronic design automation (EDA) software, and the underlying physics-based IC models, fail to fully capture the nonlinear, frequency-dependent RF parasitic effects. To overcome this problem, in this thesis we propose the use of machine learning in RFIC development. Machine learning uses statistical models to recognize hidden patterns from sample data points, known as "training"; generalize patterns; and make predictions based on new data. In theory, machine learning can capture the nonlinear, frequency-dependent RF parasitic effects very well thanks to the large variety of nonlinear modelling techniques at its disposal, such as polynomial regressions and neural networks. Therefore, this thesis investigates for the first time the feasibility of using machine learning in RFIC development to solve the problem of inaccurate RFIC simulation.

Chapter two describes how to represent and collect the RF and spec data to be able to use them in machine learning. The data needs to be represented in the format of {X: design parameters, Ysim: EDA simulation results, Ytrue: test results}. Ideally, large datasets should be collected by testing fabricated ICs. However, in this thesis we used electromagnetic-enabled mixed mode simulation data as an alternative to actual test data for demonstration purposes. Chapter three summarizes the existing RFIC development flow and describes the three different blocks in the flow where machine learning could be added: (1) between the customer specifications and the circuit design, or specs-to-design; (2) between EDA simulation and circuit fabrication, or simulation-to-fabrication; and (3) between lab test results and design revision, or test-to-re-design. Chapter four studies each block design level in detail by applying two basic machine learning techniques: polynomial regression (PR) and neural networks (NN). Chapter five provides case studies for developing RF switches using machine learning. The results show that machine learning can significantly improve the prediction accuracy, which proves the feasibility of using machine learning in RFIC development.

The research developed in this Thesis has strong potential to impact the RFIC industry. Unlike digital circuit design where the high accuracy of EDA simulations allows for highly automated circuit development, the RFIC design industry suffers from significant simulation inaccuracies. Hence, RFIC development typically requires multiple time-consuming and costly design-fabrication iterations. Some researchers have already used machine learning to improve the step between the initial specifications and design, but those solutions are not really effective because of their large computational complexity. Those researchers in ran hundreds (if not thousands) of simulations using existing EDA tool, and used those simulations to train neural network model so the model can learn how to design circuit. The hundreds of simulations cause the computational complexity. In many cases circuit designs using these techniques take even longer time than existing solutions in the industry. In contrast, this thesis focuses on the use of machine learning to optimize block #2, that is between simulation and fabrication to provide accurate predictions. The task of accurate prediction in this thesis needs less computation resource but provides more helpful to RFIC development. This thesis shows that the simulation accuracy can be improved by 98%, which will dramatically reduce the need for multiple design-fabrication iterations. This improvement means significant time and cost reduction in RFIC products.

Thesis Supervisor: Dr. Tomas Palacios
Professor, MIT Electrical Engineering and Computer Science

Acknowledgments

I first want to express my sincere thanks to Professor Tomas Palacios at MIT EECS, who provided enthusiastic help at every step of this project. This thesis aims to solve a very difficult and complex problem. Its completion would have been impossible without Professor Palacios.

I must also thank the Sandbox Innovation Fund mentors Maureen Boyce and Javier Segovia for their mentoring in this research project. I also appreciate the guidance of Professor Stefanie Jegelka and Abigail Katcoff in machine learning. Dr. Ana Bell and Mr. Kamoya Ikhofua also played an important role in this thesis by helping to answer my questions about coding. I also want to thank Professor Donhee Ham and Dr. Houk Jang from Harvard University for RFIC test assistance.

I want to sincerely thank Joan Robin in the Department of System Design and Management for admitting me to the master program. I truly enjoyed the learning experience that the program has offered me for the past 2 years.

I also want to thank my wife Yan who generously allows me leave work for two years' study. Her love, support and constructive feedbacks make me a better man.

Finally, I want to thank my unborn son. Whenever I feel his kicking from the other side of Yan's belly, the stress caused by thesis writing immediately go away. We are excited to welcome the boy to the world! I hope my son stay positive, enjoy learning, make friends and improve the world.

Table of Contents

List of Figures

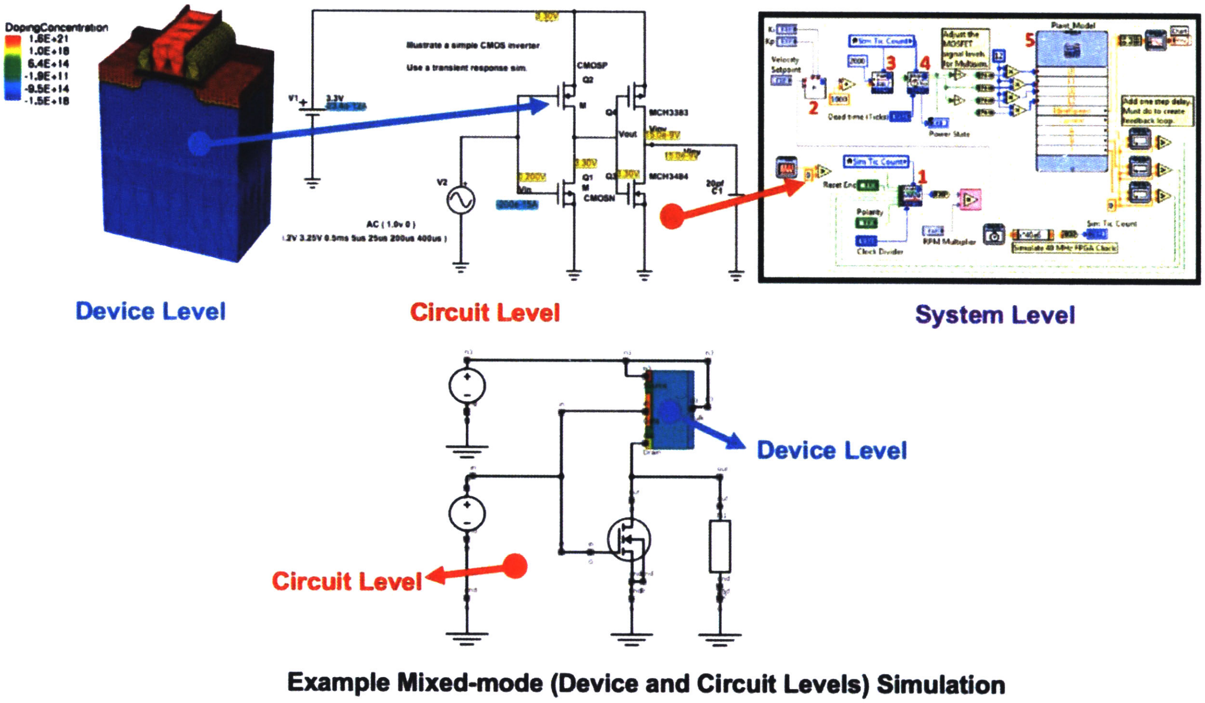
List of Tables

1. Introduction

Radio frequency integrated circuits (RFICs) are key components of most wireless communication systems with applications in cellular, base station, automotive industry, aerospace industry and the Internet of Things (IoT). Since the first handheld cellular phone call [7] was made on April 3, 1973, the cellular communication system has evolved significantly with larger capacity and better quality. Now, the RFIC industry is transitioning from 4th generation of communications (4G) to the 5th generation (5G). The design specs for RFIC companies are more challenging than before because the operational frequencies are higher, the application system is more integrated, and the circuit performance requirements are more stringent. On the other hand, RFIC companies compete with each other fiercely on time-to-market to capture market share. Because of such competitive pressure, good electronic design automation (EDA) software plays a very important role in today's RFIC companies' design activity as the EDA software tools can help engineers verify design parameters, and capture design errors before spending a lot of time and money on mistakes. EDA software also serves as the data management tool in a typical RFIC engineer's daily work as it provides an efficient data management interface between simulation and tests.

EDA has been an independent industry since the early 1980s. Now, this industry becomes highly mature and consolidated after four decades of evolution. The EDA software can solve digital IC design problems very effectively with high prediction accuracy. However, for RFICs, EDA software is much less accurate due to the hard-to-predict parasitic effects at high frequencies (Giga Hertz or above). Generally, the higher the operation frequency is, the lower prediction accuracy EDA provides. Because of the severe prediction inaccuracy, an existing RFIC development project requires multiple trial-and-error fabrication (also called "tape-out") iterations. The problem is that each trial-and-error iteration is very expensive in both time (> 2 months) and cost (can be as high as $2M). Thus, the low EDA prediction accuracy is an important bottleneck of the modern RFIC industry. This problem will be even worse in the future as 5G has expanded the operating frequencies into the millimeter wave range, which is 10 times higher than today's 4G frequencies.

EDA companies know about this problem and they have been working to try to fix it for the last three decades. The first approach was simulation integration. Figure 1 describes three simulation levels (device, circuit and system) and their relation with each other. The EDA companies combined different simulation levels into one mixed-mode simulation, hoping to better model the inter-coupling effects between different parts of the design. For example, they combined the device level simulation and circuit level simulation into a single simulation, hoping to better model inter-coupling effects between the two levels. This simulation integration approach has two problems. First, the existing RF simulation accuracy at any given single level (such as device or circuit level) is not good enough. When we mix simulations from different levels, the inaccuracy will therefore be amplified, which may generate even worse simulation accuracy. Second, the combined simulation will include a large number of coupled physics models, which will make the simulation very time-consuming and unstable. It is not uncommon that a design engineer spends one week to setup the simulation, and the simulation runs for two weeks but fails to converge at 67% of the progress. Nevertheless, the EDA industry continues to pursue this approach hoping they can provide more accurate simulation results. For example, Synopsys has an EDA tool that can run mixed-mode simulations on both the device and the circuit component levels. Cadence has tools to do simulation on both the circuit and the system levels. This approach to simulate across levels has also been reflected in the consolidation trend of the EDA industry. Initially, the EDA industry was fragmented, and different companies had different strengths in different simulation levels (e.g., Company A is good at device simulation, Company B is good at circuit simulation). Due to the trend of integrated simulation, the EDA industry is becoming more and more consolidated, so each company wants to integrate all the design levels into their own simulation ecosystem.



Example Mixed-mode (Device and Circuit Levels) Simulation

Figure 1 Different levels of simulation and mixed-mode simulation example [9~12]

The second approach is focused on increasing modeling complexity, trying to represent parasitic effects using very complex physics-backed mathematical models. EDA companies know that parasitic effects dramatically differ between fabrication technologies. So, they spend a lot of time working with foundries that are developing new fabrication technologies. But the complexity of parasitic effects slows down the development process significantly even though, many times, the parasitic effects are still unpredictable. Due to the complexity of the different models, to revise them to achieve more accurate performance is very time consuming. And also, it is well recognized by industry veterans that even the most complex models today generate very bad predictions in the first couple trials after a new fabrication technology is announced. The root cause for this poor prediction performance is the nonlinear nature of parasitic effects, especially at high operating frequencies. While EDA companies need long time to adapt their software and modes to the newest fabrication technologies, the RFIC companies cannot afford to wait due to competition pressure in the market. Thus, in practice, RFIC engineers bypass detailed EDA simulations for new fabrication technologies. They directly fabricate the IC designs using several trial-and-error design iterations and manually build back-fitting models based on test results. As mentioned before, this trial-and-error design iterations are very expensive in both time and cost.

This thesis attempts to solve this problem using a dramatically different philosophy. See Figure 2, this thesis focuses on prediction of parasitic effects based on the convolution of previous test data, existing simulation results and design parameters. Instead of spending a lot of time building complex, slow yet not-good-enough model for each fabrication technology, this thesis proposes to quickly build machine learning blocks representing the parasitic effects and provide accurate predictions without knowing the exact details of the parasitic mechanism. The proposed solution takes advantage of both semiconductor physics (represented by existing inaccurate simulation results) and machine learning (learning the delta between test results and simulation). Thus, as we will see in the later chapters, this solution has excellent prediction accuracy.

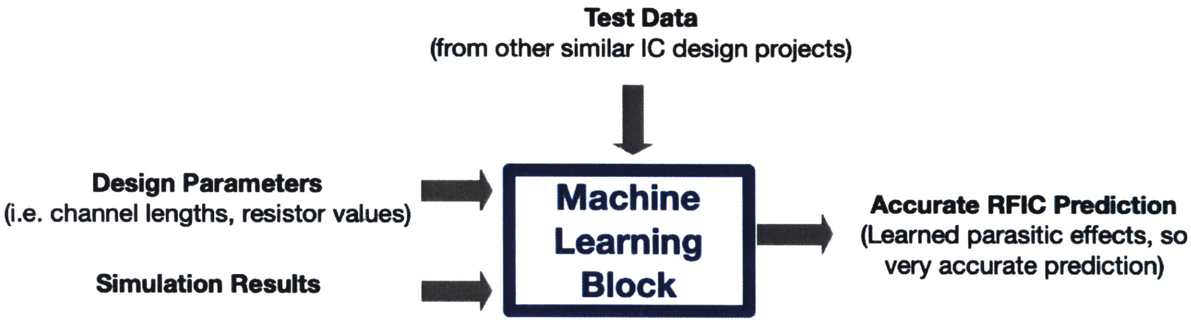

Test Data
(from other similar IC design projects)
Design Parameters
(i.e. channel lengths, resistor values)
Simulation Results
Machine Learning Block
Accurate RFIC Prediction
(Learned parasitic effects, so very accurate prediction)

Figure 2 Proposed Machine Learning Block between Simulation and Fabrication

It is worth mentioning that machine learning and IC design have interacted with each other for a long time. These interactions are of two different types: (1) IC's for machine learning, and (2) The use of machine learning in IC design. Examples for (1) are GPUs from Nvidia and other AI accelerator vendors. Examples for (2) are several register transfer level (RTL) synthesis tools from EDA companies such as Synopsys and Cadence. Those synthesis tools are used in digital circuit design, and provide a first draft of designs based on design specifications, which then engineers can make revisions on. However, for RFICs, to the best of our knowledge there is no commercial synthesis tools except for several design parameters such as simple sub-circuits and automatic layout generation. [1,2] This thesis tries to fill the gap of machine learning for RFIC design.

More specifically, this thesis aims to fix the RFIC simulation inaccuracy problem with the help of machine learning. We do this in a dramatically different way than existing solutions. The existing machine learning for IC design efforts focus on generating schematics and layout based on initial specs, but this thesis focuses on using machine learning to combine test results and simulations to generate predictions with excellent accuracy in new technologies without well-developed models. The success of the approach of this thesis when compared to other machine learning techniques applied to RFIC design is based on (1) Existing RFIC design synthesis (machine learning for RFIC) is very slow due to simulation complexity plus the first draft schematic designs have poor performance which limits its impact. The problem of slow simulation speed is inevitable even with cloud-based computation. This is due to the fact that RFIC designs and simulation models are very complex, which requires very large solution space to meet the specifications. In this case, the machine learning needs to run over hundreds of simulations to learn all the underlying semiconductor physics. (2) On the other hand, the solution proposed in this thesis has strong advantages. The RFIC design companies already have a vast amount of test data. They have invested heavily in generating data through IC fabrication and testing. That data, however, is interpreted today by engineers using just intuition and experience. If we aggregate all the data of the different fabrication technologies and circuit types, it would become a perfect data set for machine learning training, which will then have enough quality to produce accurate results.

This thesis consists of 5 chapters. Chapter 1 (this chapter) introduces the RFIC and EDA industries, the existing problem of RFIC EDA simulation and research motivation. Chapter 2 discusses how to prepare RFIC simulation and test data for machine learning. It describes the methods of data representation and data collections. Chapter 3 reviews the existing RFIC development flow and describes the proposed revision of the RFIC development flow. The proposed development flow includes 3 machine learning blocks. Chapter 4 reviews the 2 machine learning techniques: polynomial regression and neural networks. Chapter 5 studies the RFIC development case that uses machine learning. Chapter 6 summarizes the whole thesis and discusses the future research plan.

2. Data

2.1 Representation

In order to make data suitable for machine learning, we need to process the data into a machine learning friendly format. In general, the prepared RFIC data should have three parts as shown in Figure 3: (1) X: Design parameters such as transistor gate length, device width, inductor turns, metal choices and so on; (2) Ysim: EDA simulation results such as s-parameters, power gain, current consumptions; and (3) Ytrue: Actual test results for RFIC with design parameters listed in (1). The format of Ytrue corresponds to the simulated items in (2) such as s-parameters, power gain, current consumptions. The combination of {X, Ysim, Ytrue} makes one data point with high dimensions. Because different types of RFICs have different set of design specifications, we should use different data representations according to RFIC types. Fortunately, there are only a few types of RFIC components: Switch, Power Amplifiers, Low Noise Amplifiers, Filters and passive Matching Networks. Thus, we just have a few data representations for machine learning, which make the proposed use of machine learning very convenient.

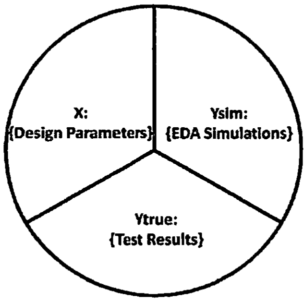


Figure 3 Data Representation

2.2 Collection

Training data collection is important for any machine learning application. There are enough data points in most RFIC companies to feed the machine learning blocks using internal data. For a typical RFIC company, there are about 30 products per year, each product will have about 5 fabrication runs, and we can aggregate the RFIC designs over the past 5 years as the fabrication technology is normally unchanged for several years. This in total will provide 750 high-dimensional data points. Even if we discount this number because of different IC types, there are still enough data points (50~100) to generate accurate prediction accuracy. 50~100 data points are enough because, as it will be shown, our results indicate that the training can converge stably with 45 data points for different RFIC types under different conditions.

In the proposed machine learning solution, the ideal training data points (each data point is an RFIC intellectual property, or IP) would come from the RFIC company. But it is hard to get 50 pieces of IP from RFIC companies for an academic research project like this. To overcome this problem, this thesis generated data points using electromagnetic-enabled simulations.

2.2.1 Electromagnetic-enabled Mixed Mode Simulation

How do we generate a complete data point (especially the test result portion) just using electromagnetic-enabled simulation? I will take a step back to answer this question. The purpose of this thesis is to demonstrate the feasibility of using machine learning in the existing RFIC development process for better prediction accuracy. The reason why current EDA simulation fails in prediction accuracy is that the representations of parasitic effects in simulation and real-world test are different. With electromagnetic-enabled simulation, we can generate two simulations with different representations of parasitic effects but very close semiconductor physics. Thus, we can assign one simulation as "Ysim: EDA simulation results" and the other simulation as "Ytrue: test results". Specifically, I assign Cadence Spectre simulation as "Ysim", and Keysight ADS Momentum Electromagnetic simulation as "Ytrue".

An RF switch will be used as the example circuit to generate simulation-based data points. The RF switch is a good starting point because: (1) it is straightforward enough to start with, so this thesis can focus on verifying the concept of machine-learning assisted simulation method, (2) yet it is complex enough to highlight the RF parasitic challenge [8] (3) moreover, RF switches will be used widely in 5G Millimeter Wave ICs, as these chips will have an architecture based on a phase array beam, hence there significant demand for RF switches.

Figure 4 shows the single pole single throw (SPST) switch designed for 2.7GHz (cellular band) and 5.8GHz (WiFi band). For "X", this thesis included the gate lengths of the series transistor and shut transistor. For "Ysim", this thesis included the scattering parameters (S-parameters) S21 at both 2.7GHz and 5.8GHz using Cadence Spectre. For "Ytrue", I used the mixed-mode simulation of schematics simulator (see Keysight ADS schematic viewer in Figure 6) and electromagnetic simulator (Keysight ADS's momentum viewer in Figure 5). Figure 5 shows the 3D electromagnetic simulation setup of RF switch's routing metal. The "Ysim" and "Ytrue" now have different representation of parasitic effects but same representation of core semiconductor physics model. With help of this data generation methods, this thesis will be able to demonstrate the effectiveness of machine learning in providing much better prediction results.

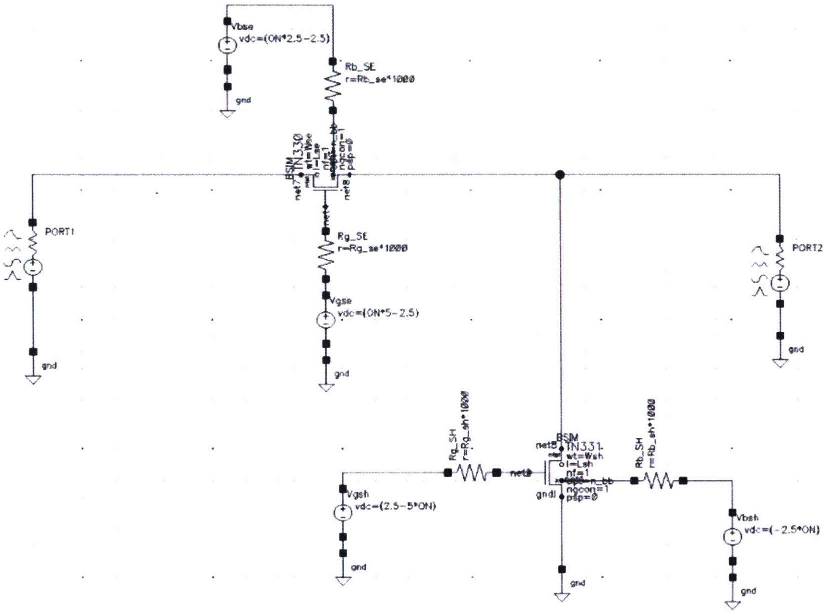


Figure 4 Schematics Simulation in Cadence Spectre ("Ysim")

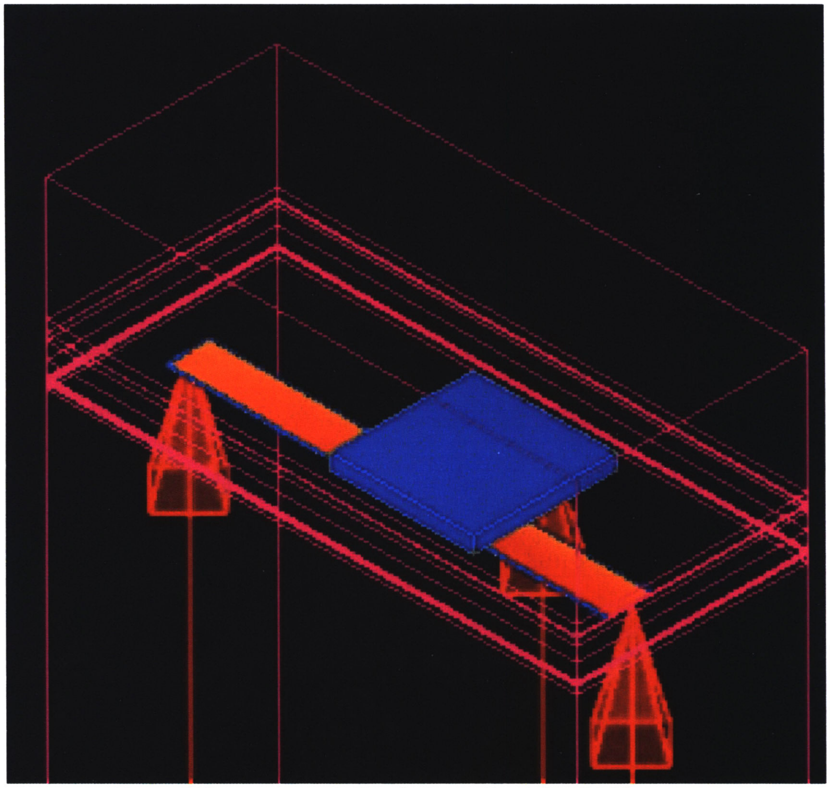


Figure 5 Electromagnetic Simulation in Keysight ADS Momentum

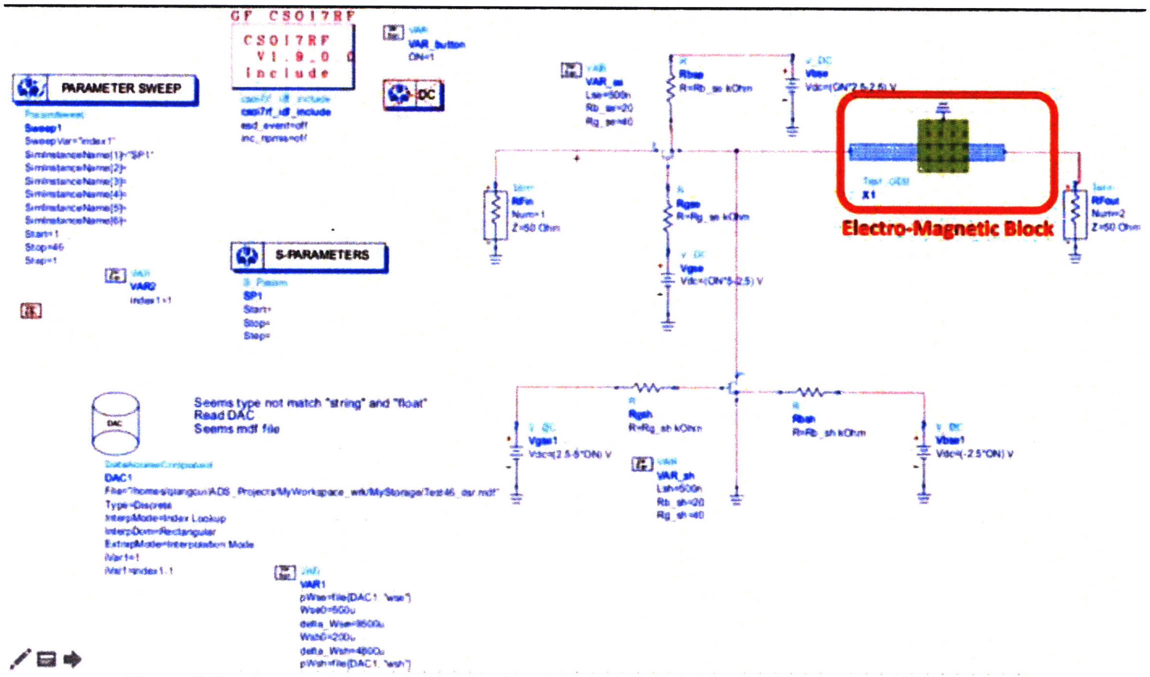


Figure 6 Electromagnetic-enabled Mixed Mode Simulation in Keysight ADS Momentum ("Ytrue")

2.2.2 Actual Fabrication

A better method to generate about 50~100 data points is to fabricate actual chips. But the cost of traditional chip fabrication is beyond the budget available for this thesis. To overcome this, we simplified the RFIC design to significantly reduce the fabrication cost yet provide quality data to validate the effectiveness of machine learning in RFIC development process. First, we found a Multi Project Wafer (MPW) opportunity from Muse semiconductor. This cut the cost by 90% because multiple research teams will share the cost of a wafer. The only down side is that we get much fewer dies than what a traditional RFIC company would get. The MPW approach is not enough as we need 50~100 data points. Moreover, if we use traditional RFIC chip fabrication in MPW fashion, the 90% discounted cost times 50 is a still too expensive. Here comes the second simplification. We designed a laser cut-friendly test structure in Figure 7. We have 3 laser cut friendly input capacitors (left), 3 laser cut friendly output capacitors (right), and 1 laser cut friendly inductor. From this one layout, we have (2^3-1)*(2^3-1)*2=49 configurations which will generate 49 different data points. By doing this, the writer further reduce the cost 98% less than RFIC company. Now, this research can have enough training points from actual chip fabrication with acceptable cost.

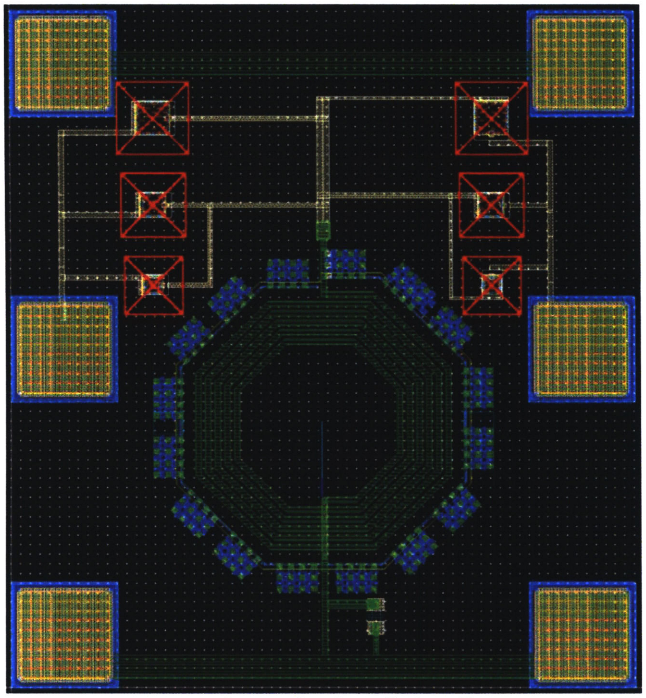


Figure 7 Laser Cut Friendly Test Chip

3. System Design

3.1 Existing RFIC Development Flow

The RFIC design flow has not changed significantly in a long time. The core of the RFIC design flow is trial and error as the EDA simulation is inaccurate. Because RFIC's EDA simulation tools cannot accurately predict chip's performance, a typical RFIC design flow requires 3~5 iterations. Each design iteration is very expensive in both time and cost. In general, the RFIC design flow contains the steps shown in Figure 8.

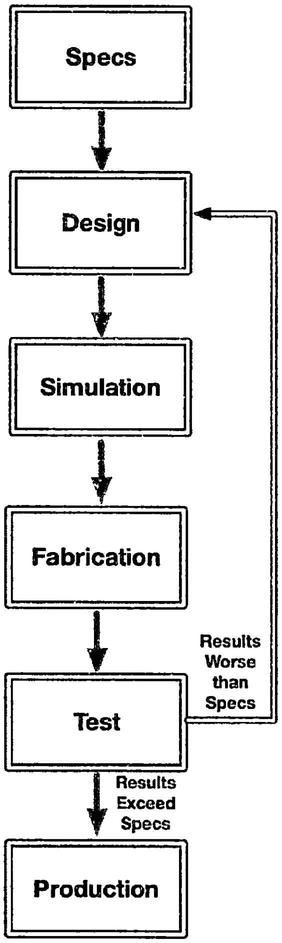


Figure 8 Existing RFIC Development Flow

Specifications: In a typical RFIC design company, the corporate technical marketing team keeps continuous communication with the corporate's customers. One important job of the marketing team is to help their customers define the detailed specifications of an RFIC product. The specification definition takes time (can be as long as 1 year) as the customers themselves may not even know what they want. Customer companies normally work with technical marketing professionals from several RFIC companies at the same time and share specs with all of the RFIC companies. The RFIC companies will start the competition after evaluating the product specifications.

RFIC Product Design: If an RFIC company decides to bid for the specification provided by their customers, it will assemble a team of engineers to design the product. The company assigns design engineers, application engineers, test engineers, technical marketers and project managers into one development team and this development team keeps on communication with customers. The RFIC companies use a divide and conquer method. They divide the RFIC into different functional blocks such as digital controller, analog bias circuit, radio frequency circuit, and module integration. The development team almost always starts from previous RFIC products with similar performance and make necessary revisions. Occasionally (<5% case), they make a new design from scratch if the product specification is dramatically different from all the earlier products.

Simulation: After the initial RFIC design, the engineers want to predict the IC performance before fabrication by using simulations. This step is very important for de-risking the design, as the engineers can use simulation to verify if the circuits will work as intended and to see if there are potential bugs. As the RFIC circuits get more complex, the simulation can take several weeks. Like the product design step, the simulation step also uses divide and conquer method. Engineers divide the simulation job into circuit functional blocks such digital controller part, analog bias circuit part, radio frequency part and module level integration part. Accuracy is crucial in this simulation step as the results directly impact engineers' design decisions and RFIC performance after fabrication. In today's RFIC design industry, simulation is never accurate in the first iteration because of the hard-to-model parasitic effects. The current solution is that engineers manually fit the models [13~15] based on trial and error design iterations, and offset the simulation errors in the next design iteration. If the measurement results after second design iteration are still far off from the fitted models, engineers will try another iteration to fit the models and make the design meet the required specifications. It is worth mentioning that those fitted models can't be generalized to other RFIC designs. For each RFIC design, the engineers need to repeat model fitting efforts using iterations.

Fabrication: Engineers will submit design data to the semiconductor manufacturers (called "foundry" or "fab") for fabrication (called "tape out") after the design and simulation step. The foundry could be inside the organization if the RFIC company is an integrated design manufacturer (such as Qorvo, Analog Devices) or outside organization if the company is a fabless design company (such as Qualcomm). Whether the foundry is from inside or outside, the whole engineering team do nothing at this step but wait. The waiting period is about 2 months depending on fabrication complexity. Sometimes engineers design other products when this product is in fabrication, sometimes they read technical documents for next project or simply take vacation.

Test: The integrated circuits finally come back after several months. The engineers become busy again. They will have technicians test the ICs based on the long list of parameters from the customer specifications. There can be hundreds of test conditions, out of which 20 or so are key test condition. Usually the first version of the design can meet no more than 70% test conditions mostly because the earlier simulation step provides inaccurate predictions. The engineers need to go back to re-design, simulation and come to test again in a trial and error manner. A typical RFIC design requires 3 to 5 iterations due to inaccurate simulations. On the contrary, digital circuit development only needs 1 design iteration thanks to the high accuracy of digital simulations.

Production: If the IC design meets 100% of the design specifications, the engineering team will move on to mass production step. Now they need to worry about the process variation and wide distribution of the test results. Sometimes, they will need to go back to the re-design step if the issue in mass production is due to a design problem. For example, if the top metal or metal capacitors have cracking issues, they may change the design into smaller piece of metal or capacitors. At the production step, the engineering team hands over the product to the product management team in the RFIC company. The development ends here.

The biggest problem of the existing RFIC development process is that it requires multiple iterations from test back to design (see Figure 8). The number of design iterations in RF IC development is much larger than in digital IC development, because RFIC simulation is very inaccurate due to hard-to-predict parasitic effects. If we can improve the RF simulation accuracy to the accuracy level in digital IC simulation, we can significantly speed up the design process, reduce the time to market and development cost. This thesis will use machine learning in the RFIC development process to significantly improve RF simulation so that we can solve this fundamental problem in RFIC development.

3.2 Proposed RFIC Development Flow

A proposed RFIC development process is shown in Figure 9. The revised development process has three machine learning blocks (highlighted in blue) added. This thesis describes all the three blocks but focuses on the second block between simulation and fabrication. This machine learning block #2 provides much more accurate simulation based on existing simulation, test data and current design. The three machine learning blocks are trained with the same set of data formatted as Figure 3. We can aggregate the high dimensional data points at company level or even industry level if a data sharing agreement is reached between companies. Even if the data aggregation only happens at the company level, a typical RFIC design company has enough data points to effectively train machine learning blocks because there are many design re-use cases in the IC company plus many designs are sharing the same fabrication process.

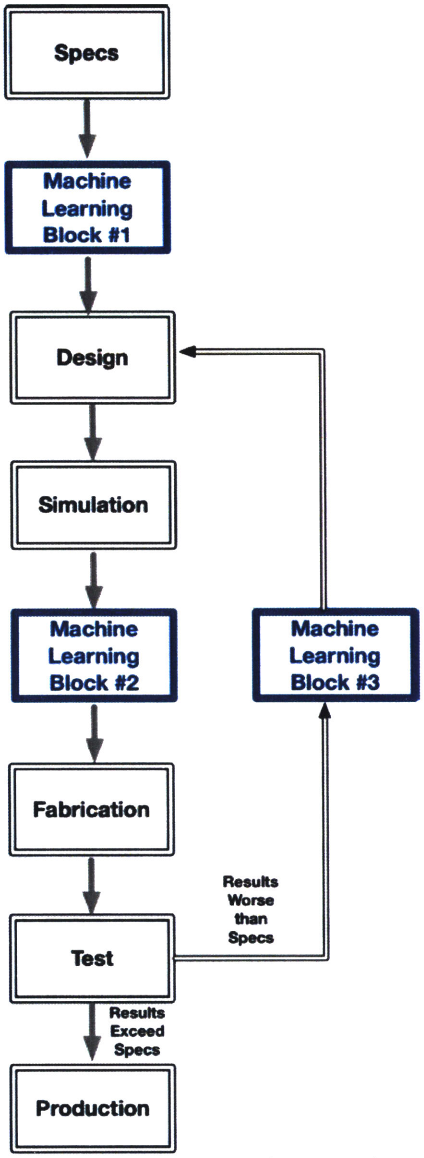


Figure 9 Machine Learning Powered RFIC Development Flow

The three machine learning blocks have the same structure as shown in Figure 10. The three blocks are trained with the same set of data with the {X, Ysim, Ytrue} format in Figure 3. But they may have different models such as different designs of neural networks. The three machine learning blocks all use two pieces (two inputs) of information to predict one piece (one output) of information.

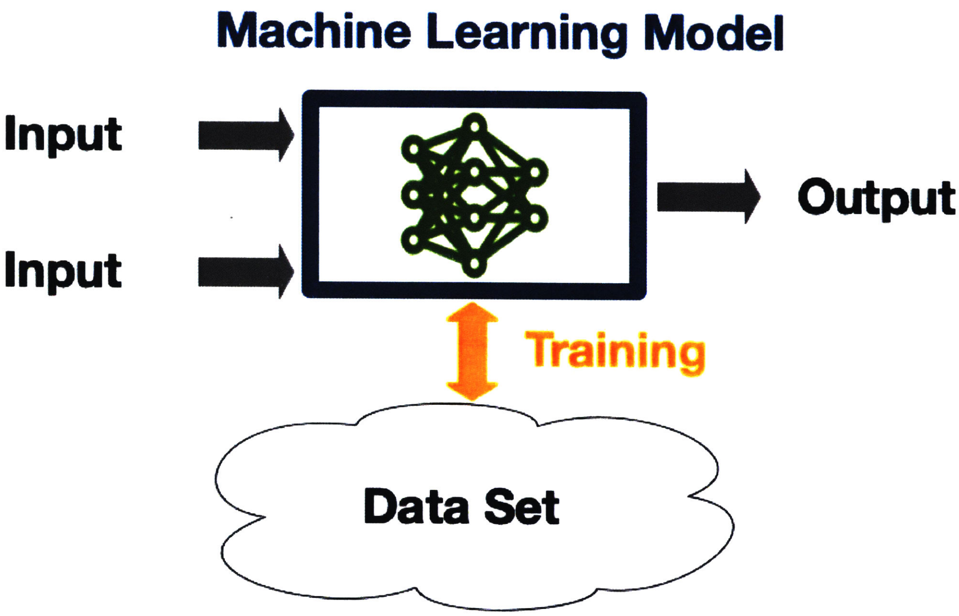


Figure 10 Structures of Machine Learning Blocks in Figure 9

3.2.1 Block #1: Specs-to-Design

Machine learning block #1 fits between specs and design steps. This block is trained on the shared data set which has {X : Design Parameters, Ysim : EDA Simulations, Ytrue : Test Results} as data format. See Figure 11, this block has two inputs (Ysim, Ytrue) and one output (X). It is worth mentioning that X, Ysim and Ytrue, as inputs or outputs of the machine learning block, may have different meanings than what is suggested by its naming. For machine learning block #1, Input Ysim means EDA Simulations (same meaning as suggested by naming), but Input Ytrue means specifications (different than the meaning suggested by naming) and Output X means suggested design parameters (not the meaning of existing design parameters suggested by naming). The Inputs and Outputs keep the labels of "X", "Ysim" and "Ytrue" so that they can match to the data set format for prediction. After customized training, this machine learning block will be able to suggest its output based on inputs with a reasonable quality. There are many matured machine learning blocks (also called synthesis tools) in digital ICs area but only a few in

RFICs. For example, Guo Zhang and Hao He from MIT CSAIL Professor Dina Katabi's lab are doing some research to automate analog circuit design based on massive number of simulations. [3] Hanrui Wang in MIT Professor Song Han's group and Professor David Pan from University of Texas also have some machine learning for analog IC design automation papers published. [4,5] In those publications, the researchers generate hundreds (if not thousands) of data points using computer-aided simulations, which makes their training process very slow. The machine learning block #1 in this thesis uses a very different approach than the existing research of machine learning for RFIC design. This block is trained on data set which has both design parameters ("X"), EDA simulations ("Ysim") and test results ("Ytrue"). This block predicts the output ("X", suggested design parameters) using two inputs ("Ysim", EDA simulations; "Ytrue", specifications). Because machine learning block #1 aggregated previous test data as training data set, it does not need the time-consuming massive simulations required in [3~5].

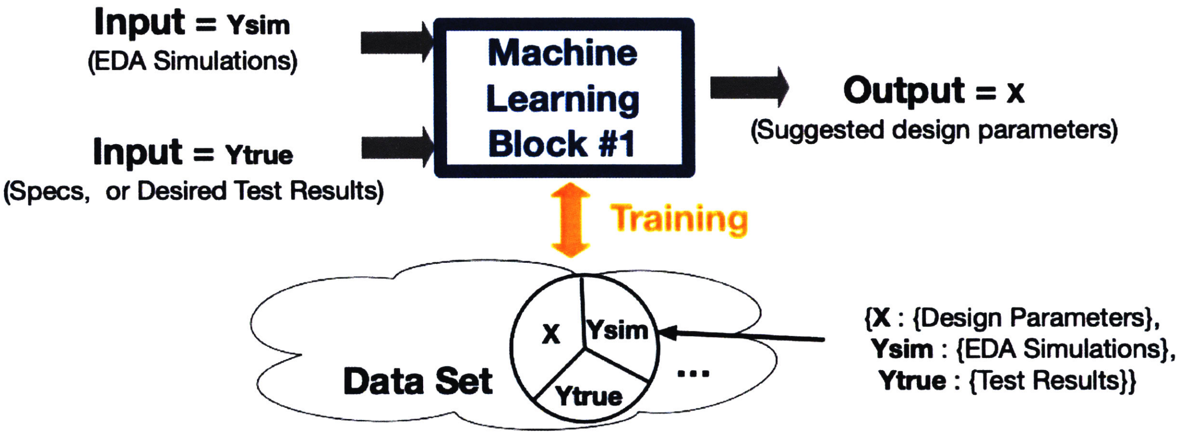

Input = Ysim
(EDA Simulations)
Input = Ytrue
(Specs, or Desired Test Results)
Machine Learning Block #1
Output = x
(Suggested design parameters)
Training
Data Set
X
Ysim
Ytrue
...
{X : {Design Parameters},
Ysim : {EDA Simulations},
Ytrue : {Test Results}}

Figure 11 Machine Learning Block #1, between Specs and Design

3.2.2 Block #2: Simulation to Fabrication

Figure 12 shows the structure of the machine learning block #2 between the simulation and fabrication steps. This machine learning block uses the same data set for training. The block predicts Output ("Ytrue", predicted test results) by two Inputs ("X", design parameters and

"Ysim", EDA simulations) without actual fabrication. As will be shown in the case study of chapter 5, this machine learning block can provide very accurate predictions after customized training on reasonable number of data points (less than a hundred). On the contrast, the trainings on reference [3,4,5] requires large number of simulation results. The reason why trainings in reference [3,4,5] require more data points is that the machine learning approaches in [3,4,5] have very different learning objectives than machine learning block #2 here. The machine learning block #2 just need to learn the difference between test results and simulation results. The difference (delta part) is just parasitic effects, which can be well represented by frequency-dependent nonlinear LRC networks. Though parasitic effects are hard to model using physics-backed math equations, they are relatively easy to fit using simple machine learning techniques such as polynomial regression and neural networks in section 0. Therefore, machine learning block #2 only needs less than a hundred data points for training. On the contrast, the machine learning block's learning objectives in reference [3,4,5] is the entire RFIC design, which includes all the complex semiconductor physics and circuit knowledge, thus they require at least thousands of simulation points. As will be shown in case study in section 0, machine learning block #2 can provide excellent prediction accuracy. This level of prediction accuracy is very helpful to reduce the required number of design iterations which means significant time and cost reduction for the RFIC design industry.

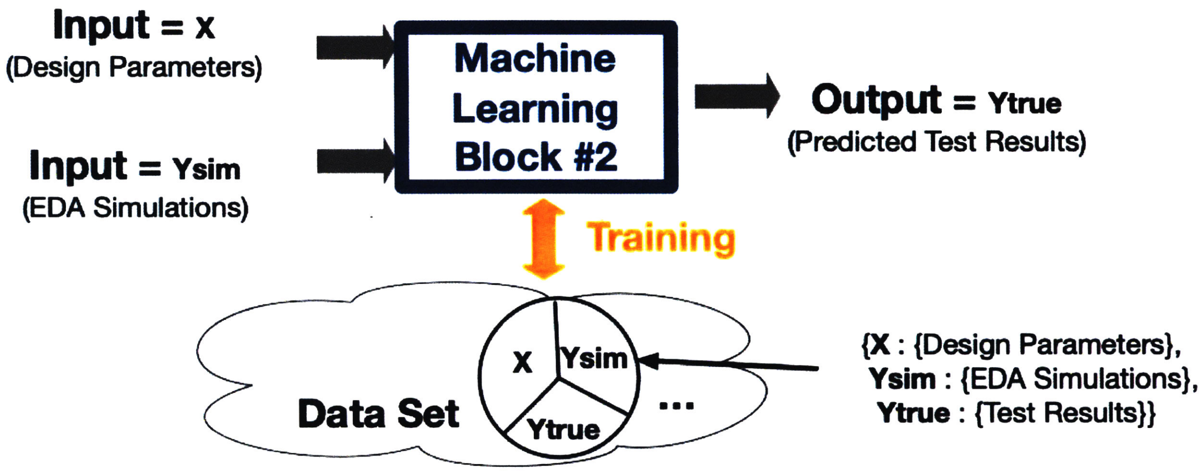


Figure 12 Machine Learning Block #2, between Simulation and Fabrication

3.2.3 Block #3: Test to re-Design

Figure 13 shows the structure of machine learning block #3 at the rework loop from test to re-design steps. It uses the same data set as the other 2 machine learning blocks. Similar to block #1, this machine block #3 has “Ysim” (EDA simulations) and “Ytrue” (specifications, or desired test results) as two Inputs and “X” (suggested design revisions) as the Output. The main difference from machine learning block #1 is that the newly acquired test data (see Figure 13) is added to the data set and is assigned heavier weight during the re-training. Because the newly acquired data point is directly relevant to the desired design revision, the prediction of machine learning block #3 is theoretically more effective than machine learning block #1. In Figure 13, the newly acquired data point is added to data set for re-training the machine learning block. An alternative approach is to feed the newly acquired data point into machine learning block #3 as the third Input (not drawn). This thesis did not compare the effectiveness of machine learning block #3 in Figure 13 and its alternative, but has listed it as future work.

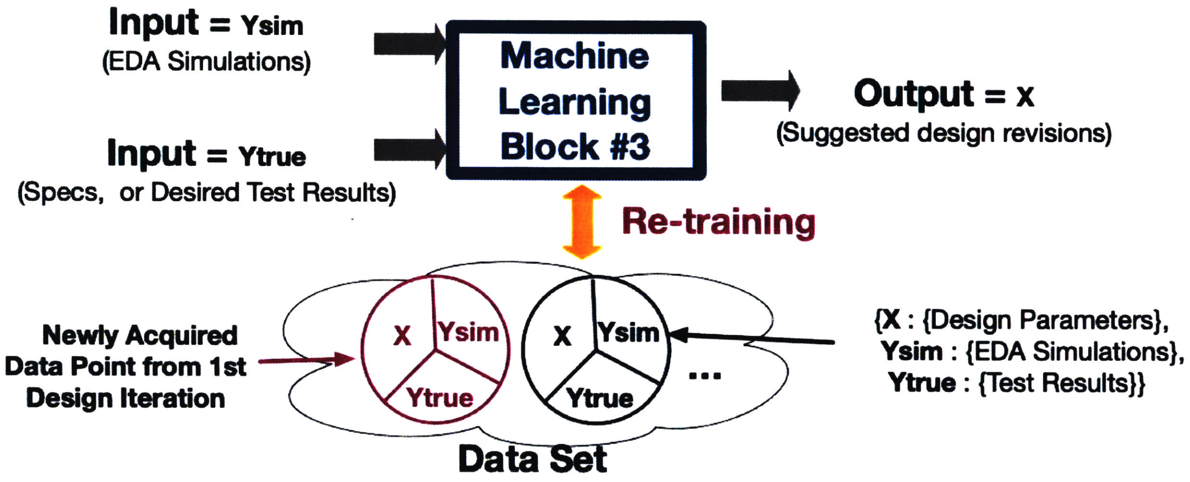

Input = Ysim
(EDA Simulations)
Input = Ytrue
(Specs, or Desired Test Results)
Machine Learning Block #3
Output = x
(Suggested design revisions)
Re-training
Newly Acquired Data Point from 1st Design Iteration
X
Ysim
Ytrue
X
Ysim
Ytrue
...
{X : {Design Parameters},
Ysim : {EDA Simulations},
Ytrue : {Test Results}}
Data Set

Figure 13 Machine Learning Block #3, between Test and re-Design

3.2.1 Input and Output of Three Machine Learning Blocks

In the proposed machine learning-powered RFIC development flow, there are three machine learning blocks: between specs and design, between simulation and fabrication, and between test and re-design. The three blocks share the same data set but use different inputs to predict the output. Table 1 summarizes the Input and Output configurations for the 3 blocks. The naming of "X", "Ysim" and "Yture" is tailored for the shared data set. For different machine learning blocks, the naming (such as "Ytrue" for ML Block #1) may have slightly different meaning (such as desired test results, or specifications for ML Block #1). This table aims to clarify the meaning of Input and Output of three blocks.

Table 1 Summary of Input and Output for Three Machine Learning Blocks

	"X"	"Ysim"	"Ytrue"
Data Set	**design parameters:** parameters such as gate lengths, capacitor sizes.	**EDA simulations:** such as s-parameters, power gain.	**test results:** actual test results such as s-parameters, gain.
ML Block #1	**Output**: design suggestions such as how to set gate lengths.	**Input**: EDA simulations such as s-parameters.	**Input**: desired test results, or specifications
ML Block #2	**Input**: design parameters such as gate lengths, capacitor sizes.	**Input**: EDA simulations such as s-parameters.	**Output**: Predicted RFIC test results such as s-parameters. Generally, it is much more accurate than simulation.
*ML Block #3	**Output**: design suggestions such as how to set gate lengths.	**Input**: EDA simulations such as s-parameters.	**Input**: desired test results, or "desired specifications"

*Note: ML Block #1 and ML Block #3 have same input, output setting but ML Block #3 has one more relevant data points in data set

4. Block Design

This thesis uses two simple machine learning techniques: polynomial regression and neural networks for implanting the machine learning block. As will be shown in chapter 5, both regression and neural networks provide significant prediction accuracy. RFIC companies could use even more sophisticated algorithms (Long short-term memory, random forest, and so on) to customize the machine learning block based on their circuit types and fabrication technologies, and more sophisticated machine learning blocks should provide even better prediction performance than regression and neural networks in this thesis. However, the work demonstrated in this thesis is just a proof of concept. The thesis will not discuss those advanced machine learning algorithms since this thesis focuses on feasibility research of machine learning in RFIC development flow instead of machine learning optimization.

4.1 Polynomial Regression

Polynomial regression is a basic machine learning technique that analyzes the relationship between different independent variables (commonly named as "x1", "x2" and so on) and the dependent variable (commonly named as "y") by polynomial equations. Regression often comes with regularization, a technique that adds additional weight into modeling to represent the system offset. There are two types of regressions: linear polynomial and nonlinear polynomial regression. Linear regression is the regression without polynomial transformation but with regularization. Nonlinear regression has both polynomial transformation and regularization. In the machine learning training implemented in this Thesis, the polynomial order was limited to less than 5, and regularization between 0 and 20 with 0.1 as sweeping step. The data was cross validated by data using 90% of the data as training and 10% as validation. The root mean square error (RSME) was used as performance indicator for the training.

One interesting observation about linear regression is that it is the technique most similar to engineers' interpreting efforts in today's RFIC development flow. In today's trial and error RFIC development flow, engineers interpret the discrepancy between test and simulation in a linear fashion. For example, if the power output of the RF power amplifier is 10% less than the required specification, engineers would increase the device size linearly by 10%. This intuitive correction will not always work because in reality, the RFIC performance is nonlinear. This nonlinearity can be easily captured by nonlinear regression machine learning techniques (such as neural networks) but very hard for today's engineers to interpret in a linear regression manner.

4.2 Neural Networks

Artificial neural network is a network of neurons that calculate the output based on inputs, weights and activation functions. Its name comes from the weak similarity between the mathematical equations that govern its behavior and biological neural networks. The most fundamental building block of neural networks is the single "Neuron" (also called "Perceptron" if the activation function is the step function) shown in Figure 14. A single "Neuron" has one constant unit (1) and n inputs (x1~xn). w0 represents the weight for constant unit, and w1~wn represents the weights for inputs x1~xn. The neuron first sums up x1*w1+..+xn*wn+w0, then feeds a output activation function to determine if the output is 1 or 0. Neuron networks is very powerful in predicting RFIC parameters because each building blocks can represent different degrees of nonlinearity hence can model the nonlinear parasitic effects much better than humans can. That is why we will see the significant prediction improvement in chapter 5.

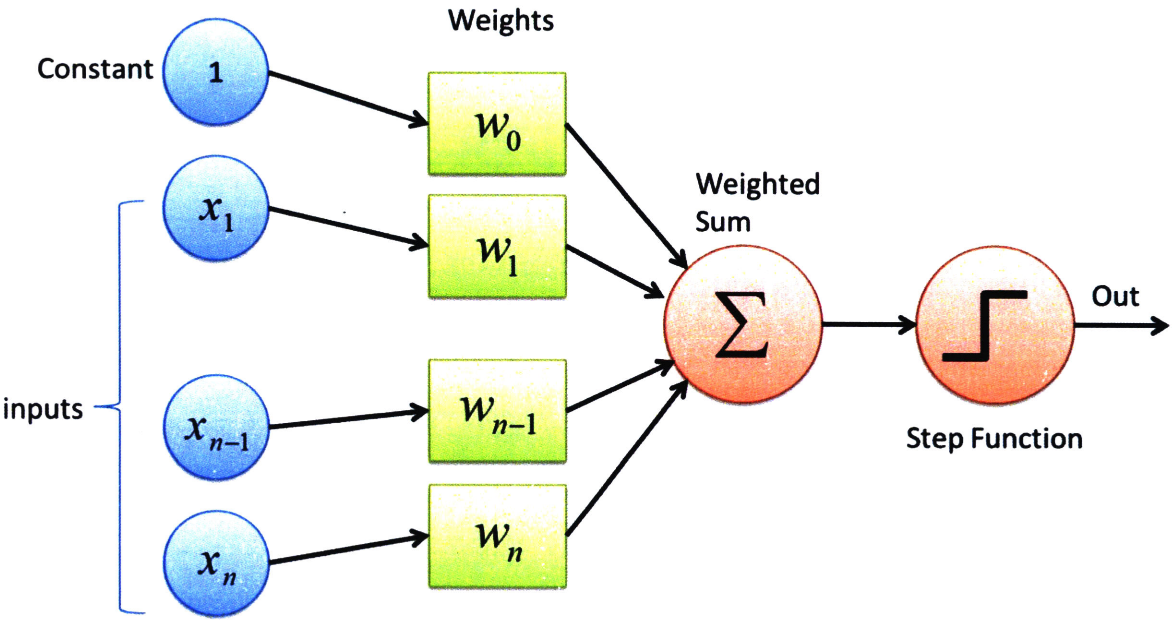


Figure 14 Building Block of Neural Networks, "Perceptron" or single "Neuron" [6]

Figure 15 and Table 2 show the dimensions of different feedforward neural networks (NNs). In this thesis, the NN were customized for different types of RFIC circuits (such as switches, PAs, LNAs, matching networks or other circuits) and different ranges of operating frequencies. Since the learning problem in the machine learning block #2 in Figure 12 is a regression problem, the activation function in the output layer was set to be linear. But the activation functions in the hidden layers were set as rectified linear unit (ReLU) to enable the nonlinear representation of RFIC parasitic effects.

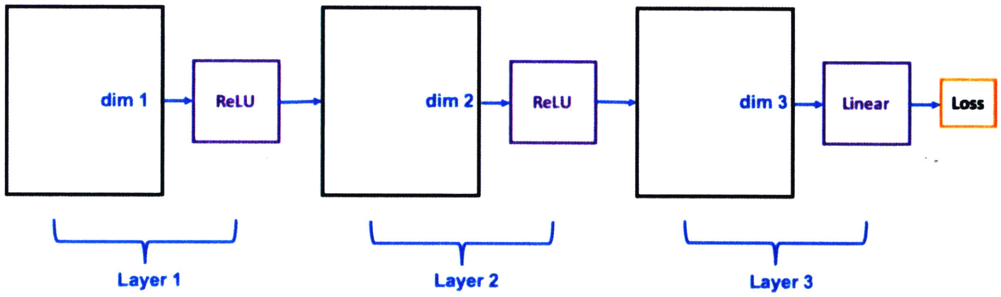
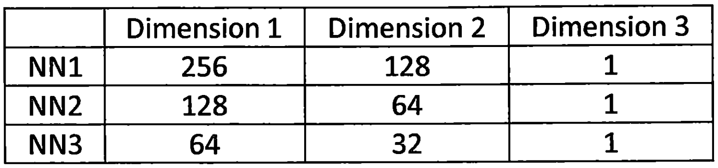


Figure 15 Block Diagram of Neural Networks

Table 2 Dimensions of Neural Networks



	Dimension 1	Dimension 2	Dimension 3
NN1	256	128	1
NN2	128	64	1
NN3	64	32	1

5. Case Study

This chapter will demonstrate the effectiveness of the machine learning block #2 (simulation to fabrication) with the case of RF switch. The "test" data (Ytrue) comes from the electromagnetic enabled mixed-mode simulations described in section 2.2.1. Though the data is non-ideal, it is real enough to demonstrate the effectiveness of machine learning block #2 because it has completely different parasitic effects from schematic-only simulations (Ysim). This situation is very similar to the real world, where the test data has completely different parasitic effects from simulation.

5.1 Circuit

Figure 16 shows the first case study: a single pole single throw (SPST) RF switch. There is a series MOSFET transistor on the top left and a shunt MOSFET transistor on the bottom. The two transistors operate in a complementary way and they are biased at gate terminal and body terminal through resistors (gate resistors Rg_se, Rg_sh and body resistors Rb_se, Rb_sh). The Vdd is set at 2.5V. See Table 3, when this SPST switch operates at ON state, series MOSFET is ON and shunt MOSFET is OFF. The controlling voltages are Vgse=2.5V, Vbse=0V, Vgsh=-2.5V, Vbsh=-2.5V. When the switch operates in the OFF state, the series MOSFET is OFF and the shunt MOSFET is ON. The controlling voltages are Vgse=-2.5V, Vbse=-2.5V, Vgsh=2.5V, Vbsh=0. The series shunt MOSFET structure and negative charge pump voltage (-Vdd=-2.5V) are used here for better signal isolation between RFin and RFout.

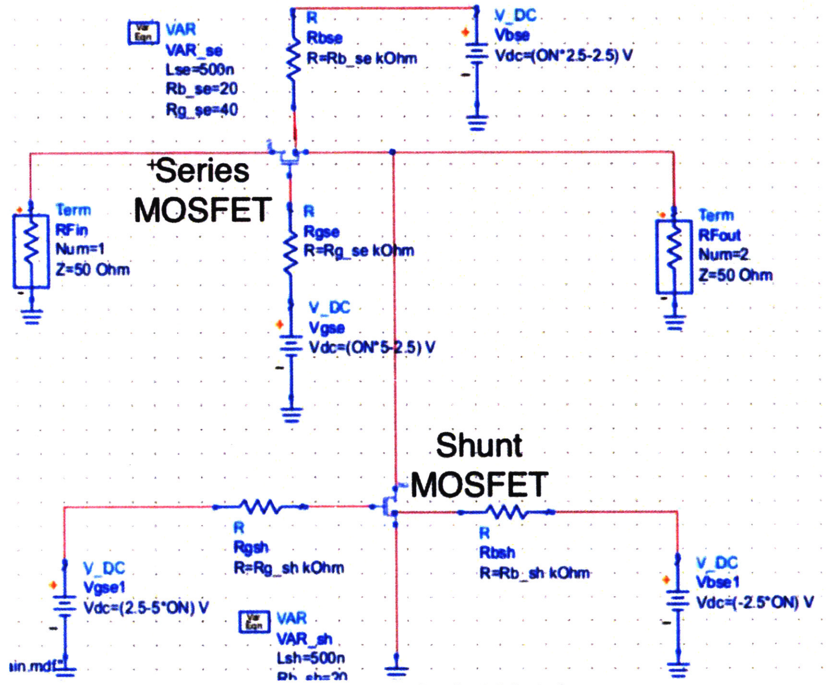


Figure 16 Case Study: RF Switch

Table 3 Bias Voltages of RF Switch

	RF Switch ON	RF Switch OFF
Series MOSFET	ON	OFF
Vgse	2.5V	-2.5V
Vbse	0V	-2.5V
Shunt MOSFET	OFF	ON
Vgsh	-2.5V	2.5V
Vbsh	-2.5V	0V

5.2 Parameters

This case uses data collected from electromagnetic-enabled mix mode simulation. The data is formatted as {X, Ysim, Ytrue} shown in Figure 3. For this RF switch case, the parameters are listed in Table 4. For X (design parameters), I choose the channel length of series MOSFET and shut MOSFET as parameters because channel length is the most important feature that determines the RF Switch's electrical performance. For Ysim (EDA simulation results), I use S21's magnitude, dB representation, imaginary part, real part and phase in Cadence Spectre simulator as they cover the typical parameters that RFIC engineers watch in the design. For Ytrue (test results), I use same set of S21 parameters in Ysim (such as magnitude and so on) collected from electromagnetic enabled simulations in Keysight ADS momentum.

Table 4 List of Parameters

Parameter Name	Parameter Type	Description
pWse	X	Normalized channel length of series transistor
pWsh	X	Normalized channel length of shunt transistor
CDS_mag	Ysim	CDS simulation result of the gain (S21) magnitude
CDS_dB	Ysim	Decibel version value of the gain (S21)
CDS_img	Ysim	The imaginary part of simulation gain (S21) magnitude
CDS_real	Ysim	The real part of simulation gain (S21) magnitude
CDS_phase	Ysim	The phase of simulation gain (S21) magnitude
EM_mag	Ytrue	Test of gain (S21) magnitude
EM_dB	Ytrue	Decibel version test of gain (S21)
EM_img	Ytrue	The imaginary part of test gain (S21)
EM_real	Ytrue	The real part of test gain (S21) magnitude
EM_phase	Ytrue	The phase of test gain (S21) magnitude
delta_mag	Ytrue	The difference between test and simulation result
Delta_dB	Ytrue	The difference between decibel version test and simulation result

5.3 Prediction Tasks

Table 5 listed 15 basic prediction tasks for the RF Switch. For each task, we can predict "Ytrue" based on "X" (if any) and "Ysim" (if any) using machine learning block #2 trained on aggregated {X, Ysim, Ytrue} data set. Let us take task 4 as an example. The input "X" represents the design parameters pWse (normalized channel length of series transistor) and pWsh (normalized channel length of shunt transistor). The input EDA simulation result "Ysim" is CDS_mag (Electromagnetic simulated gain (S21) magnitude in Cadence Spectre environment). And the output test result "Ytrue" is the electromagnet-enabled simulation (decibel version) of magnitude EM_mag. Those prediction tasks provide the method to check the effectiveness of the machine learning block #2 in RFIC development flow.

It is worth noting that the size of {X, Ysim, Ytrue} data set is intentionally set at 50 though many more data points can be collected from electromagnetic-enabled mix mode simulation. Here is the reasoning for that: A typical RFIC company has at least 50 data points available for one kind of RFICs. This thesis intentionally uses the worst-case situation to demonstrate effectiveness of machine learning in the real world. If the data set size is relaxed to be larger, the prediction accuracy will generally be better but will saturate at certain number depending on data distribution and the circuit condition to be predicted.

Table 5 Prediction Tasks

Task ID	"X"	"Ysim"	To be predicted: "Ytrue"
1	n/a	CDS_mag	EM_mag
2	n/a	CDS_dB	EM_dB
3	pWse, pWsh	n/a	delta_mag
4	pWse, pWsh	CDS_mag	EM_mag
5	pWse, pWsh	CDS_mag	delta_mag
6	pWse, pWsh	n/a	delta_dB
7	pWse, pWsh	CDS_dB	EM_dB
8	pWse, pWsh	CDS_dB	delta_dB
9	pWse, pWsh	CDS_img	EM_img
10	n/a	CDS_img	EM_img
11	pWse, pWsh	CDS_real	EM_real
12	n/a	CDS_real	EM_real
13	pWse, pWsh	CDS_phase	EM_phase
14	n/a	CDS_phase	EM_phase
15	pWse, pWsh	CDS_img	delta_img

5.4 Results

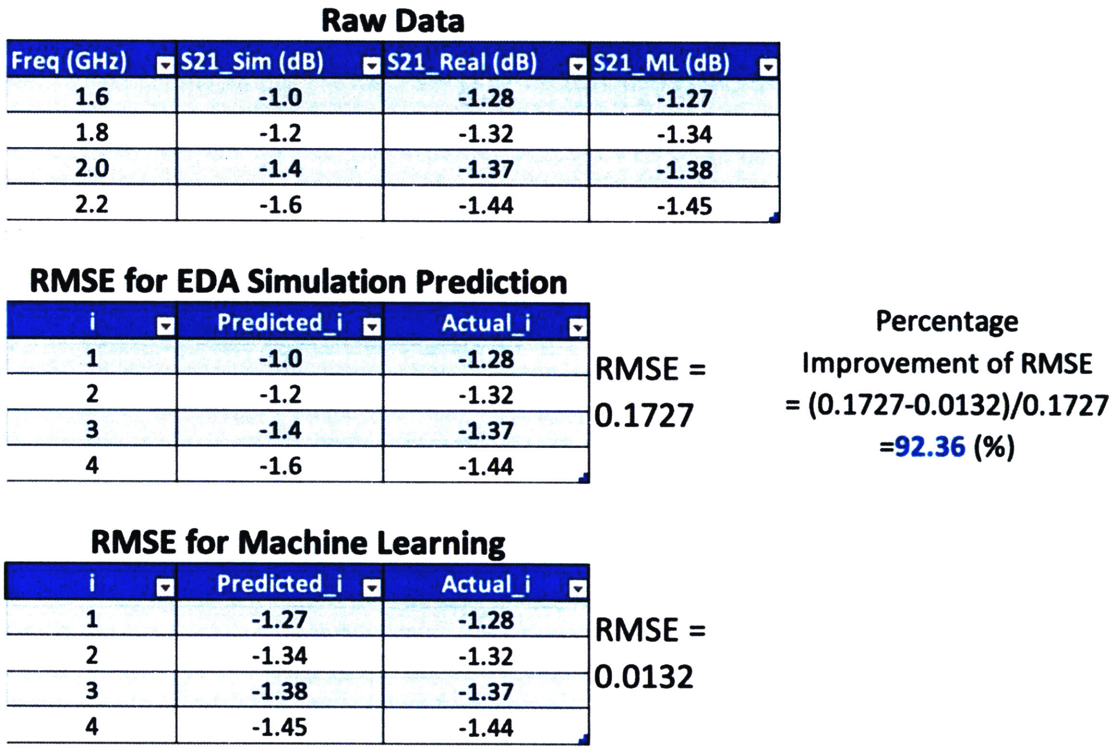
As briefly mentioned in chapter 5, this thesis uses 4 machine learning techniques: polynomial regression, and neural networks with 3 different dimensions (Table 2). And the predictions are made at both cellular band frequency (2.7GHz) and WiFi band frequency (5.8GHz). Therefore, there are 120 cells in Table 6. In each cell there is a number between 0 and 100, which is the percentage improvement of root mean square error (RMSE) representing prediction accuracy with machine learning versus EDA simulation only. Figure 17 explains the calculation of percentage improvement of RMSE in details. Basically, this calculation uses percentage improvement to compare the RMSE with the machine learning and RMSE with only EDA simulations.

Table 6 Simulation RMSE Improvement (%) using Linear Regression (LR) with less than 5 orders and Neural Networks (NN)

	Cellular Band Frequency (2.7GHz)				WiFi Band Frequency (5.8GHz)			
Task ID	PR(%)	NN1 (%)	NN2 (%)	NN3 (%)	PR(%)	NN1 (%)	NN2 (%)	NN3 (%)
1	**99.90**	99.10	98.96	98.59	**98.93**	98.90	98.85	98.89
2	**99.88**	99.62	99.68	99.66	98.51	**99.62**	99.58	99.44
3	1.17	1.34	1.33	**2.24**	93.15	**98.67**	98.34	98.37
4	**99.92**	98.97	99.07	98.53	99.26	**99.35**	98.95	98.74
5	**1.24**	1.03	0.17	1.00	97.47	**98.65**	97.58	98.02
6	1.39	1.50	1.46	**1.53**	88.87	**99.79**	99.71	99.50
7	**99.95**	99.76	99.68	99.25	98.96	99.53	**99.83**	99.65
8	**1.50**	1.50	1.28	1.61	94.83	**99.75**	99.69	99.62
9	**99.97**	99.85	99.65	99.63	99.14	**99.82**	99.79	99.69
10	**99.92**	99.73	99.71	99.73	93.78	**95.11**	93.77	93.76
11	**99.90**	98.96	98.83	98.85	86.65	**98.73**	97.08	96.79
12	**99.83**	98.95	98.64	98.91	72.68	**96.85**	96.80	96.02
13	**99.99**	99.78	99.64	99.56	**99.73**	99.63	99.73	99.51
14	**99.99**	99.82	99.86	99.56	99.18	**99.94**	99.91	99.87
15	1.39	1.39	0.92	**1.62**	89.10	98.02	**98.61**	96.81

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

(a)



Raw Data

Freq (GHz)	S21_Sim (dB)	S21_Real (dB)	S21_ML (dB)
1.6	-1.0	-1.28	-1.27
1.8	-1.2	-1.32	-1.34
2.0	-1.4	-1.37	-1.38
2.2	-1.6	-1.44	-1.45

RMSE for EDA Simulation Prediction

i	Predicted_i	Actual_i
1	-1.0	-1.28
2	-1.2	-1.32
3	-1.4	-1.37
4	-1.6	-1.44

RMSE = 0.1727

Percentage Improvement of RMSE
= (0.1727-0.0132)/0.1727
=92.36 (%)

RMSE for Machine Learning

i	Predicted_i	Actual_i
1	-1.27	-1.28
2	-1.34	-1.32
3	-1.38	-1.37
4	-1.45	-1.44

RMSE = 0.0132

(b)

Figure 17 Calculation of percentage improvement of Root Mean Square Error (RMSE) (a) Definition of RMSE (b) Calculation Example demonstrating Percentage Improvement of RMSE

For all the 15 prediction tasks, the machine learning block #2 makes better prediction than original EDA simulation results. Except for task 3, 6, 8, 15 (explanations will be given below), most of the tasks will have more than 98% improvement of accuracy. This is an exciting result. Typically, the RFIC simulation has RMSE of about 10%. An improvement of 98% means the RMSE is only 0.2%, which means the RFIC development has very high chance to be successful after first design-fabrication iteration. If one-time pass happens, this will save about 80% of development time and cost!

Table 7 Best ML Methods for different parameters under different frequency

Prediction	Cellular Band Frequency (2GHz)			WiFi Band Frequency (5.8GHz)		
"Ytrue"	**"X" and "Ysim"**	**Best ML**	**% Improv.**	**"X" and "Ysim"**	**Best ML**	**% Improv.**
EM_mag	pWse, pWsh, CDS_mag	LR	99.92	pWse, pWsh, CDS_mag	NN1	99.35
EM_dB	pWse, pWsh, CDS_dB	LR	99.95	pWse, pWsh, CDS_dB	NN2	99.83
EM_real	pWse, pWsh, CDS_real	LR	99.9	pWse, pWsh, CDS_real	NN1	98.73
EM_img	pWse, pWsh, CDS_img	LR	99.97	pWse, pWsh, CDS_img	NN1	99.82
EM_phase	pWse, pWsh, CDS_phase	LR	99.99	CDS_phase	NN1	99.94
delta_mag	pWse, pWsh	NN3	2.24	pWse, pWsh	NN1	98.67

delta_dB	pWse, pWsh, CDS_dB	NN3	1.61	pWse, pWsh	NN1	99.79
delta_img	pWse, pWsh, CDS_img	LR	1.39	pWse, pWsh, CDS_img	NN2	98.61

To find more insights, this thesis rearranged the predictions into Figure 7, where only the best prediction methods (X, Ysim to Ytrue) for each prediction task are listed. There are several interesting findings of machine learning based prediction performance.

Finding 1: There is significant improvement if we feed machine learning models with both existing EDA simulations ("Ysim") and geometric design parameters ("X"), regardless of frequency bands or what parameter to predict.

Explanation 1: In order to predict an integrated circuit's parameters (such as EM_mag: the measured Ytrue of gain magnitude), we need to consider 3 possible factors: (1) underlying semiconductor physics, which is represented by input simulation (such as CDS_mag), (2) design parameters input by engineers (such as pWse, pWsh), and (3) frequency-dependent parasitic effects (represented by learned weights "Th"s). In the industry, engineers only considered (1) and (2), which is well modelled in their electron design automation (EDA) simulation. They also tried to represent (3) using Maxwell's equations with uniform parameters. However, we believe (3) is where the major discrepancy shows up in existing simulation method because parasitic is not linear! On the other hand, the proposed machine learning techniques (both polynomial regressions and neural networks) are perfect tools to represent non-linear frequency-dependent parasitic effects. That is why we see significantly improved RMSE result for the held-out test data.

Finding 2: For lower frequency (2.7GHz) data set, polynomial regression provides best RMSE improvement, but for higher frequency (5.8GHz), neural networks provides much better RMSE improvement.

Explanation 2: As mentioned in explanation 1, the machine learning blocks try to learn the frequency-dependent parasitic effects. The parasitic effects are highly dependent on frequency. In RF semiconductor area, one important parasitic effect is called "harmonics". Harmonics has different orders, 1st order, 2nd order, 3rd order etc. At lower frequency, 1st order is dominant. As frequency moves higher, higher order (2,3,…,n,..) harmonics will be more dominant. Neural networks have intrinsically better prediction ability on non-linearity for higher order "harmonics" prediction. Therefore, we see at 5.8GHz data sets, neural networks demonstrated better prediction than polynomial regression. We are in the transition from 4G to 5G. One important change in 5G is that it expands from sub-6GHz frequency range to mmWave frequency range (27GHz, 37GHz). I expect that customized neural networks at higher frequency will provide excellent prediction results for radio frequency integrated circuit design at 5G era.

Finding 3: For lower frequency (2.7GHz) data set, the delta_mag and delta_dB of the gain difference ("delta_part") between test data (Ytrue) and simulation prediction improvement is marginal but very remarkable in high frequency (5.8GHz) data.

Explanation 3: This is an interesting phenomenon, probably caused by "harmonics" found in RF semiconductor physics. At lower frequency, the existing EDA simulation is less off from the test result, the "delta part" is mostly influenced by the 1st order of "harmonics" and absolute value of "delta" is small hence could be impacted by test (here is electromagnetic-enabled enabled mix. Mode simulation) convergence noise floor. So, it is harder to improve the prediction RMSE. On the other hand, as frequency increases to 5.8GHz, the gain difference ("delta_part") between test (Ytrue) and simulation prediction improvement is large enough, and also dominated by higher order of "harmonics". Therefore, the simulation improvement is much higher. I expect the machine learning will help mmWave frequency integrated circuits design much more than sub-6GHz frequency.

Finding 4: For high frequency WiFi Band (5.8GHz), high dimensional neural networks (NN1) provides better improvement than other low dimensional neural networks (NN2 and NN3).

Explanation 4: At higher operating frequency (5.8GHz), the higher order harmonics cause non-linear discrepancy between simulation and test results. Such harmonics come from different pieces of integrated circuits design such as input stage, output stage, Vdd, grounds, inter-stage nodes etc. A typical integrated circuit has hundreds of net list nodes (pieces of integrated circuits). Each net list node can have tens of orders of harmonics impact. So, we need around 1,000 weights to account those parasitic effects. That is why for the higher frequency WiFi Band (5.8GHz), we get best prediction improvement from NN1, which has larger number of tunable weights. This means as integrated circuit gets more complex, deeper neural networks with more weights may be needed to predict better results.

6. Conclusions

This thesis proved the feasibility of using machine learning to improve RFIC development process. Chapter 1 introduces the RFIC design and EDA industry. Chapter 2 explained how to generate and structure the {X, Ysim, Ytrue} data set. Chapter 3 first reviewed the existing RFIC design flows and its key problem: highly iterative trial-and-error development process due to EDA simulation inaccuracy. Then, the existing RFIC development is revised by adding 3 machine learning blocks (specs to design, simulation to fabrication, and test to design). Chapter 4 described the machine learning techniques in each block. Finally, Chapter 5 demonstrated the effectiveness of machine learning using RF Switch as the example. For RF Switch example, Cadence Spectre simulation is used as EDA simulation results "Ysim" and ADS Keysight Momentum's Electro-magnetic enabled simulation is used as "Ytrue" for demonstration purpose. Four machine learning methods (one polynomial regression, and three different neural network designs) are used to 15 prediction tasks at both cellular band frequency (2.7GHz) and WiFi band frequency (5.8GHz). It is clearly that machine learning trained on early tape-out data can dramatically improve the simulation accuracy by > 98%. At higher frequency, neural networks are more powerful than polynomial regression.

We plan to continue this project after graduation. For this project, I have assembled a team of electrical engineers, machine learning engineers, business developers as well as industry mentors. The team has gets accepted by MassChallenge and received pre-seed funding from MIT Sandbox and National Science Foundation (NSF) i-Corps program. The future work is to get real world RFIC data to improve machine learning blocks. We will start to collaborate with university IC labs to aggregate RFIC data, and use the data to improve machine learning blocks in different fabrication technologies. With the better machine learning blocks, the writer will then be able to approach RFIC design companies for more RFIC data and customize the machine learning blocks for each company.

References

[1] A. Olofsson, "Silicon Compilers-Version 2.0," *keynote, Proc. ISPD,* 2018.

[2] Hao He , Guo Zhang , Jack W. Holloway , and Dina Katabi "End-to-end Learning for Distributed Circuit Design," presented at the Workshop on ML for Systems at The Thirty-second Conference on Neural Information Processing Systems, 2018.

[3] B. Xu *et al.*, "Device Layer-Aware Analytical Placement for Analog Circuits," in *Proceedings of the 2019 International Symposium on Physical Design*, 2019: ACM, pp. 19-26.

[4] H. Wang, J. Yang, H.-S. Lee, and S. Han, "Learning to Design Circuits," *arXiv preprint arXiv:1812.02734,* 2018.

[5] A. L. Ressler, "A circuit grammar for operational amplifier design," Massachusetts Institute of Technology, 1984.

[6] @SagarSharma4244, "What the Hell is Perceptron?," 2019-10-11 2019.

[7] W. E. Gibson, "The First Cell Phone Call Was Made 45 Years Ago," 2020.

[8] C. Wang *et al.*, "A study of interferences inside an RF switch array in 45nm SOI CMOS," in *2017 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*, 2017: IEEE, pp. 1-3.

[9] "Semiconductor process simulation - Wikipedia," 2020.

[10] @e14Community, "Development Tools and Solutions," 2020.

[11] "Figure 8: Closed loop system simulation of the brushed DC motor..." https://www.researchgate.net/figure/Closed-loop-system-simulation-of-the-brushed-DC-motor-controller-The-LabVIEW-Control_fig8_268981258 (accessed.

[12] "Mixed-mode simulation." http://www.globaltcad.com/en/solutions/introductory-tutorials/mixed-mode-part-i/00-mixed-mode.html (accessed.

[13] H. F. Chau and D. Pavlidis, "A physics- based fitting and extrapolation method for measured impact ionization coefficients in III- V semiconductors," *Journal of applied physics,* vol. 72, no. 2, pp. 531-538, 1992.

[14] "A short example of equivalent circuit fitting with real data." http://lacey.se/science/eis/fitting-with-real-data/ (accessed.

[15] "Model Fitting." https://courses.washington.edu/matlab1/ModelFitting.html (accessed.