A Practical Methodology to Waive Marginal Timing Violations using Machine Learning

by

Rajat Kumar

Submitted in partial fulfilment of the requirements

for the degree of Master of Technology in Electronics and Communication Engineering

with specialization in VLSI and Embedded Systems

Under the supervision of

Dr. Sneh Saurabh

Certificate

This is to certify that the thesis titled **"A Practical Methodology to Waive Marginal Timing Violations using Machine Learning"** submitted by **Rajat Kumar** to the Indraprastha Institute of Information Technology Delhi, for the award of the Master of Technology, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

July, 2022

Supervisor Name: Dr. Sneh Saurabh
Department of Electronics and Communication Engineering
Indraprastha Institute of Information Technology
New Delhi, 110020

Abstract

Achieving timing closure is a challenging task, and it becomes more complicated due to the artificial pessimism in the traditional timing models of the flip-flops. During the signoff stages, we can alleviate this problem by waiving marginal timing violations with the help of more accurate flip-flop timing models and careful analysis of the failing endpoints. In this work, we propose to develop ANN-based and SVM-based timing models for flip-flops. We demonstrate that the errors in ANN-based models and SVM-based models are less than 2% and 1%, respectively, compared to the golden SPICE results. Further, we propose a three-tiered filtering mechanism to waive marginal timing violations. It employs an ANN-based timing model to filter violations using predicted clock-to-Q delay. Then, it uses an SVM-based timing model to ensure that the marginally failing flip-flop can correctly capture the data. Finally, it checks whether surplus slack is available in the fanout of the marginally failing flip-flop that allows waiving that violation. We demonstrate the utility and robustness of the proposed methodology on TAU CONTEST'19 benchmark circuits and validated the results with SPICE simulations.

Acknowledgements

The work for this thesis was carried out at IIIT-Delhi, India, during the year 2021-2022. First and foremost, I would like to express my sincere gratitude to my research advisor Dr. Sneh Saurabh for regular guidance and encouragement throughout the journey of this work. Without his support, this work would never have been a successful one. My sincere thanks also go to all the supporting staff for their prompt help and support whenever needed.

This work is dedicated to my family. I am grateful to my parents for encouraging me at all times.

I am also thankful to my peers Debjit, Animesh, and Gagandeep who supported me to resolve some peculiar issues and were always available for any support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We ensure the timing closure of a design using static timing analysis (STA) tools. These tools perform timing analysis with the help of timing models of the flip-flops existing in a technology library in Synopsys Liberty format [1]. The timing model of a flip-flop typically consists of separate two-dimensional lookup tables for the setup time (ST), the hold time (HT), and the clock-to-Q (C2Q) delays obtained using SPICE simulations [1–4]. For a safe circuit operation, we take conservative values for these attributes. As a result, timing analysis using traditional flip-flop timing models is often pessimistic.

The artificial pessimism of the timing models complicates the task of achieving timing closure [5]. During the sign-off stages, we often need to tackle marginal timing violations. We can fix marginal timing violations by making suitable design changes. However, fixing these violations can add a few more iterations to the design cycle and adversely impact the design schedule. Moreover, fixing these marginal timing violations can incur unnecessary penalties of area, power, and other figures of merit.

As an alternative, we can waive off some marginal timing violations after careful analysis. A marginal timing violation often increases the C2Q delay, and we can tolerate such an increase depending on the circuit condition. However, we need a more accurate flip-flop timing model for analyzing marginal timing violations. Additionally, we need to devise a technique to filter such marginal timing violations safely.

1.1 Traditional Methods to fix timing closure problems

In the past, researchers have proposed setup-hold pessimism reduction (SHPR) techniques and flexible timing models for flip-flops to alleviate timing closure problems [6–14]. These techniques model the interdependence between the timing attributes (ST, HT, C2Q delays) using linear, polynomial, or piece-wise linear functions. Since the timing behavior of a flip-flop (e.g., C2Q delay) is highly non-linear when the data changes close to the clock edge, these models cannot fully utilize the flexibility in the flip-flop timing models. As an alternative, we propose to employ machine learning (ML) based flip-flop timing models in this work because they can efficiently capture the complex dependencies on several input features. In contrast to the traditional models and other SHPR techniques, we do not explicitly model the ST and HT of a flip-flop. Instead, we use trained ML models to classify a flip-flop as violating/non-violating based on the circuit conditions.

Filtering out timing violations needs to be carried out very carefully. Wrongly filtered timing violations can lead to a chip failure. However, ML-based models are statistical and can produce some errors. Therefore, we need to design a violation filtering mechanism that does not produce false negatives. In this work, we have adopted a three-tiered violation filtering framework that is robust and minimizes the risk of waiving off a real timing violation. First, we use an artificial neural network (ANN) model to predict the C2Q delay of a marginally failing flip-flop and ensure that the delay is within a specified tolerance. Second, we use a support vector machine (SVM) model to check whether the marginally failing flip-flop can correctly capture the data. Third, we check whether there is sufficient slack available in the fanout of the marginally failing flip-flop. This check ensures that the increased delay of the marginally failing flip-flop does not lead to any extra timing violations in its fanout. We waive the timing violations of a flip-flop only when it satisfies all these three criteria.

We demonstrate the effectiveness of the proposed technique on benchmark circuits of TAU CONTEST'19 [15]. We use NANGATE 15 nm open-cell libraries and NCSU FreePDK15 models for implementing these circuits and developing an ML-based flip-flop timing model [22]. Using golden SPICE simulations on these designs, we verify that the proposed technique never produces a false negative result. Additionally, we show that we can filter more timing violations as the tolerances are increased. However, it

cannot filter any more violations beyond some tolerance because real timing failures start to show up. Hence, the proposed technique is robust and efficient for filtering marginal timing violations. Moreover, the proposed technique is versatile and can be enhanced to incorporate the effects of various other factors such as temperature, voltage, and process-induced variations. Additionally, the continued advancement in the ML technology can allow us to adopt a similar methodology for filtering marginal violations arising due to signal integrity issues and IR drop in the supply lines [16–20].

1.2 Contribution of this work

In one of our previous works, we have examined the dependencies of C2Q delay on different circuit conditions [4]. Additionally, we developed an ANN-based flip-flop timing model that computes C2Q delay more accurately than the traditional timing models [4]. In this work, we have employed the same ANN-based flip-flop model in the first-tier filtering. However, an ANN-based flip-flop model can produce some false negatives due to the high user-defined tolerances on the C2Q delay and the statistical nature of the ML model. Hence, in this work, we have developed an SVM-based flip-flop model that treats the timing check problem as a classification problem. We use the SVM-based flip-flop model in the second-tier filtering. Ideally, we can use only one of these ML models to filter marginal timing violations. However, we propose to employ both these models to minimize the risk of waiving a real failure. Finally, we also check whether there is sufficient slack on the output of the violating flip-flop such that no new violation is created by its increased C2Q delay. Thus, the proposed methodology can transfer the surplus slack from the output side of the flip-flop to the input side wherever possible. The results produced by the proposed technique on the benchmark circuits agree well with the golden SPICE simulations. We find that the proposed technique can waive 25% of the marginal timing violations on average.

1.3 Thesis Organization

The thesis is organized as follows. In Chapter 2, we explain the basic terminologies used in this paper. In Chapter 3, we described the ANN-based and SVM-based flip-flop timing models. Chapter 4 explains the three-tiered marginal timing violation waiving framework. Chapter 5 presents the results obtained using the proposed methodology on the benchmark circuits. Finally, Chapter 6 concludes the work.

Chapter 2

Timing Behavior of a flip-flop

In this section, we have briefly described the timing behavior of flip-flops and the relevant terminologies.

2.1 Definitions

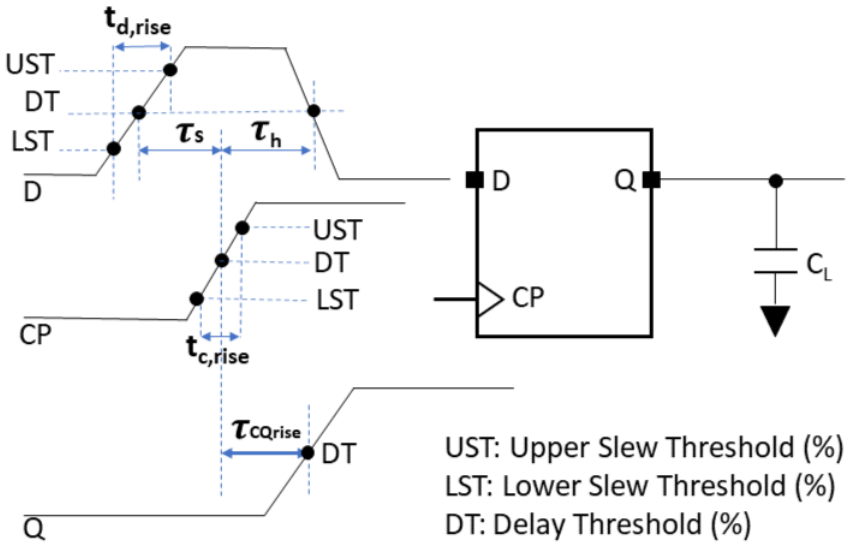


Figure 2.1: Definition of circuit parameters in a flip-flop.

Consider the D flip-flop shown in Fig. 2.1. It drives a load capacitance C_L

connected to the Q-pin. We define the rise slew as the time a signal takes to transition from the lower slew threshold (LST) to the upper slew threshold (UST). We denote the slew at the D-pin of the flip-flop as $t_{d,rise}$ and at the clock pin as $t_{c,rise}$ [4]. We define rise delay ($\tau_{CQ,rise}$) as the time interval between the signal crossing the delay threshold (DT) at the clock pin to the signal crossing the delay threshold at the Q pin. The thresholds and the transition points are mentioned in the technology libraries. For the falling signal, the delays and the slews are defined analogously. The time interval between the signal arriving at the D pin and the next clock edge arriving at the clock pin is defined as setup skew (τ_s). The signals can arrive at the D pin through different paths. We consider the last arriving signal at the D pin to measure the setup skew [4]. We define hold skew (τ_h) as the time interval between the signal changing value at the D pin after the clock signal has arrived at the clock pin. We consider the earliest change in the signal value at the D pin to measure the hold skew [4].

2.2 Dependence of C2Q Delay on Circuit Parameters

The traditional flip-flop timing model defines setup and hold constraints for a flip-flop that demarcate a window around the clock edge where the transition in data is forbidden. To derive setup time (T_s) and hold time (T_h), the counterpart skews are set to a large constant value. This constraint on the data edge allows the traditional timing model to model C2Q delay as a function of only $t_{c,rise}$ and C_L. However, this constraint also reduces the flip-flop timing space for the traditional STA tools. Thus, the restricted flip-flop timing model is safe but unnecessarily pessimistic. The existing SHPR techniques attempt to widen the timing space of the flip-flop by modeling the dependence of T_s and T_h and reduce some pessimism. However, experimental results have shown that a proper data capture for a flip-flop and its C2Q delay primarily depend on the following circuit parameters [4]:

- Data slew: $t_{d,rise}$ or $t_{d,fall}$ depending on transition being considered.
- Clock slew: $t_{c,rise}$ or $t_{c,fall}$ depending on the flip-flop type.
- Setup skew (τ_s)

- Hold skew (τ_h)
- Load capacitance (C_L)

Therefore, we need to capture the above dependency in the flip-flop timing model to achieve greater accuracy and utilize the entire safe operating region for a flip-flop. The proposed ML-based models are developed to accomplish this objective, as explained in the following sections.

Chapter 3

ML-based Timing Models

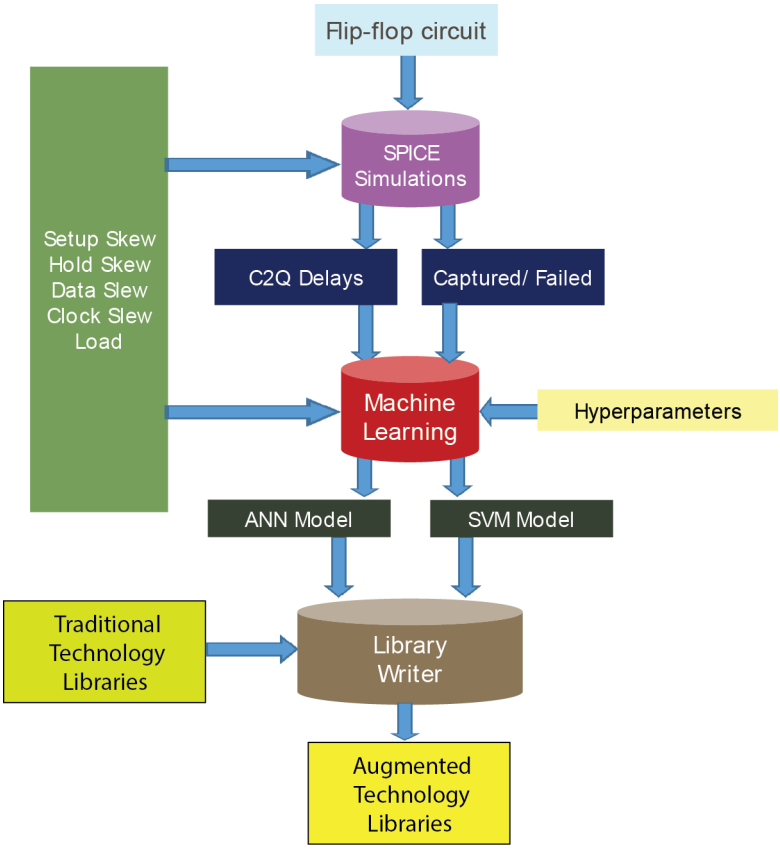


Figure 3.1: Model creation framework.

Fig. 3.1 shows the framework of creating ML-based flip-flop timing model. First, we perform SPICE simulations for each flip-flop existing in a technology library.

We vary the input waveform and the circuit conditions to gather sufficient training and test data. Then, we create ML-based timing models using the results of these SPICE simulations. We adjust the hyperparameters to improve the accuracy of the ML models. Finally, we save the ML-based models to be used for waiving timing violations. We provide the details of the proposed ML-based models in the following paragraphs.

3.1 ANN-based Timing Model

Based on the dependency of the C2Q delay on the circuit condition described above, we can write as follows [4]:

$$\tau_{CQ,rise} = F(t_{d,rise}, t_{c,rise}, \tau_s, \tau_h, C_L) \tag{3.1}$$

$$\tau_{CQ,fall} = G(t_{d,fall}, t_{c,rise}, \tau_s, \tau_h, C_L) \tag{3.2}$$

where F and G are functions that we can model efficiently using ANN. We extract C2Q delay using SPICE simulations for varying data slew, clock slew, setup skew, hold skew, and load capacitance for a given flip-flop and transition. We varied these parameters within a timing space where the flip-flop latches the data correctly. We have varied the data slew, clock slew, and capacitance for the range defined in the traditional technology library. We have taken finer steps in regions where setup skew and hold skew are small since the C2Q delay increases sharply in these regions.

The input features to the ANN model are the attributes on which F and G depend, i.e., $t_{d,rise}$ or $t_{d,fall}$, $t_{c,rise}$ or $t_{c,fall}$, τ_s, τ_h, and C_L. Feature scaling of all the input parameters is done to have a mean value of zero and standard deviation as one. We partition the collected data into two sets: 80% as the training data and 20% data points as the test data. The ANN produces C2Q delay based on the value of the above five input features. We trained separate ANN models for different transition types (rise and fall) using 20K data samples. We used normalized root means square error (NRMSE) to evaluate the model's accuracy. The loss function used to train the model was mean squared error (MSE). We tuned various hyperparameters, such as the learning rate, number of epochs, and batch size, to improve the accuracy of the ANN model [21]. We experimented with different ANN architectures to assess the trade-off between accuracy and complexity.

We present the results in Tab. 3.1. The notation [5,30,60,25,1] means that there are five neurons present in the input layer, then there are three hidden layers with 30, 60, and 25 neurons each, and one neuron in the output layer. We observe that the accuracy increases as the number of layers in the ANN model increase since the error percentage decreases. However, we observe undesirable over-fitting in overly complex ANN architectures. Based on these considerations, we choose [5,30,60,25,1] architecture for the rest of this paper. The ANN-based model has errors less than 2% compared with the golden SPICE results.

Table 3.1: Accuracy Trade-off in ANN

Architecture	***NRMSE(Rise)(%)***	***NRMSE(fall)(%)***
[5,8,8,1]	2.6	2.6
[5,15,10,1]	2.4	2.5
[5,8,8,5,1]	2.4	2.7
[5,17,17,10,1]	1.9	2.2
[5,28,28,25,1]	1.7	2.2
[5,32,32,25,1]	2.3	2.4
[5,30,60,15,1]	2.1	1.5
[5,30,60,25,1]	**1.5**	**1.4**
[5,32,64,25,1]	1.7	2.2
[5,30,60,25,10,1]	1.9	2.2

3.2 SVM-based Timing Model

We create an SVM-based flip-flop model that can determine whether a flip-flop can correctly capture data for the given circuit condition. We use the same input features (i.e., $t_{d,rise}$ or $t_{d,fall}$, $t_{c,rise}$ or $t_{c,fall}$, τ_s, τ_h, and C_L) for training the SVM model as for the ANN model. We gather training and test data by performing SPICE simulations and varying the input features appropriately. An SVM works by determining a hyperplane or boundary that can separate the classes. We labeled the points for which the flip-flop captured the data correctly as 1 and labeled the rest as 0. We took finer steps in the region where the flip-flop is likely to fail in capturing the input data.

We experimented with linear and radial basis function (RBF) kernels in this work. We also tuned the hyperparameters C and *gamma* to improve classification accu-

racy. The parameter C is the cost of misclassification. A large C gives low bias and high variance. The hyperparameter *gamma* gives curvature weight to the decision boundary for the RBF kernel. A small gamma gives a pointed bump in the higher dimensions, and a large gamma gives a softer, broader bump. The results of tuning these hyperparameters are shown in Tab. II, separately for the rise and fall transitions. We observed that RBF delivered better accuracy than linear kernels. For the RBF, the accuracy increases by increasing C till $C = 50$, and then it starts decreasing due to over-fitting. On increasing *gamma*, the accuracy increases till $gamma = 10$, and then it remains pretty constant. Hence, we choose the SVM model with RBF kernel, $C = 50$, and $gamma = 10$ in this work.

Table 3.2: Tuning accuracy of the SVM model.

Kernel	***C***	***gamma***	***Accuracy(rise) (%)***	***Accuracy(fall) (%)***
Linear	1	scale	79.7	87.1
Linear	10	scale	80.5	95.9
Linear	20	scale	80.6	96.9
Linear	50	scale	80.6	97.1
RBF	1	10	95.4	92.3
RBF	10	10	98.9	97.8
RBF	20	10	99.2	98.4
RBF	30	10	99.2	98.7
RBF	40	10	99.3	98.8
RBF	**50**	**10**	**99.4**	**99.1**
RBF	70	10	99.3	99.0
RBF	50	0.01	91.2	90.8
RBF	50	0.1	96.9	96.3
RBF	**50**	**10**	**99.4**	**99.1**
RBF	50	15	99.4	99.1
RBF	50	20	99.4	99.1

Chapter 4

Timing Violation Waiver

Fig. 4.1 shows the framework of the proposed violation waiver methodology. Firstly,

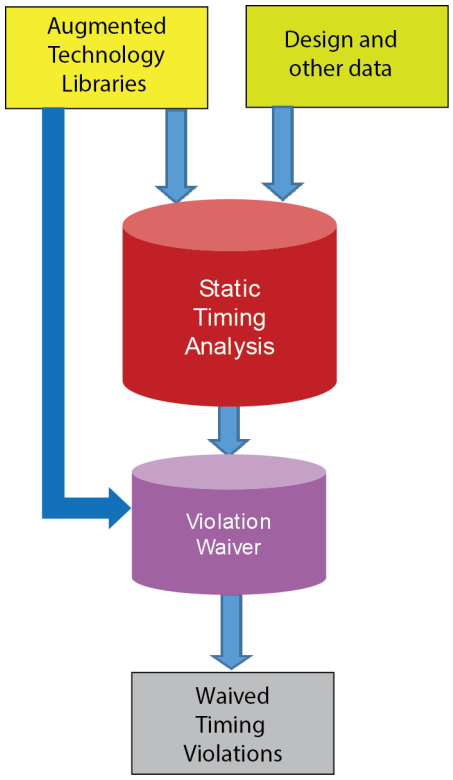


Figure 4.1: Framework of proposed violation waiver methodology.

we carry out the STA using traditional timing libraries and obtain the violation reports for the given design. Then, the violation waiver extracts information from the STA tool

that it requires for analyzing the failing paths and waiving a violation. It also reads the augmented timing libraries that contain the proposed ML-based models. Subsequently, it analyzes each violating timing endpoint with the help of proposed ML-based models and waives the violation if it is not a real timing failure. We illustrate the method of waiving a violation in Fig. 4.2.

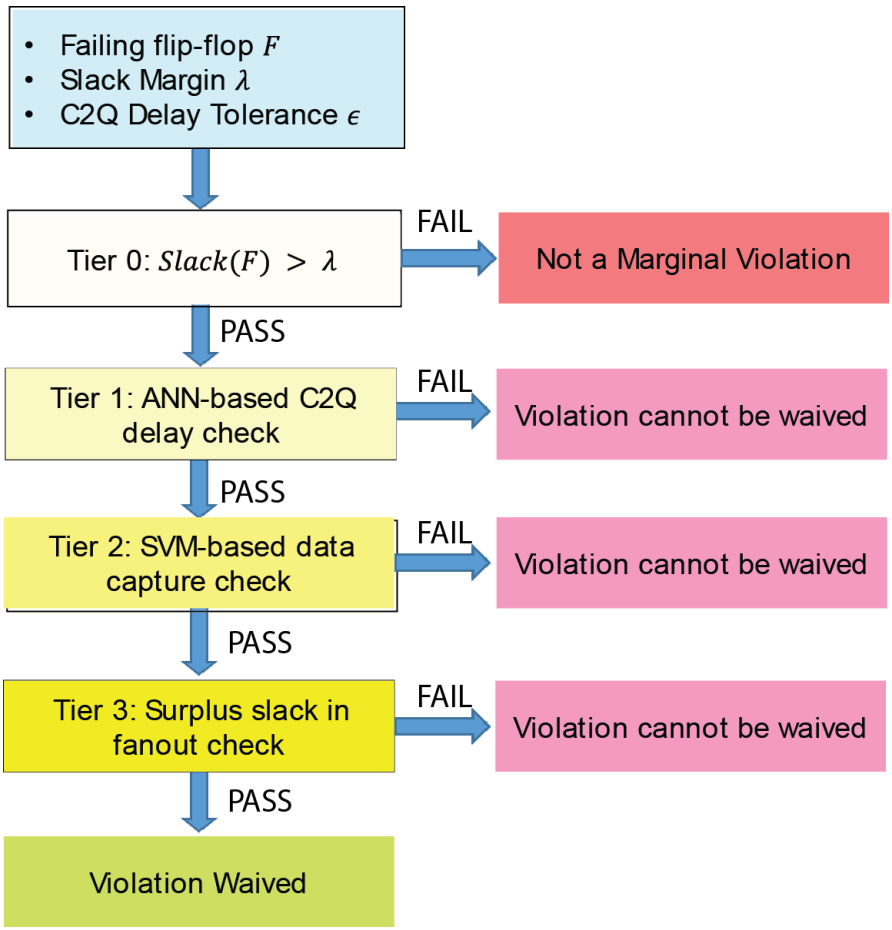


Figure 4.2: Method of waiving a given violation.

We consider each failing endpoint sequentially for waiving. We take slack margin (λ) and C2Q delay tolerance (ϵ) as input parameters from the designer. The slack margin λ defines the window of marginal timing violation. A timing violation is considered marginal if it is greater than λ. For example, if $\lambda = -10ps$, we consider timing violations with slack s as marginal if $-10ps < s < 0$. We consider only the marginal timing violations for waiving. A violation is waived if it passes all three tiers of checking, as illustrated in Fig. 4.2. We describe these checking in the following paragraphs.

4.1 ANN-based Checking:

In the traditional timing analysis of a flip-flop, a setup/hold violation occurs when the data changes in the forbidden window defined by its setup/hold time constraints. For a failing flip-flop, three problems can occur: a) C2Q delay increases from the nominal C2Q delay, b) flip-flop can reach a metastable state and then settle to a stable 0/1 value in a non-deterministic manner, and c) flip-flop can miss the data entirely. If the setup/hold violation in a flip-flop leads to only a C2Q delay increase, we can tolerate the violation in some cases. We use the ANN-based model to compute the C2Q delay more accurately than the traditional flip-flop model and filter those flip-flops that can potentially tolerate the violations.

In this work, we use a C2Q delay tolerance ϵ to define the maximum allowed percentage increase in the C2Q delay over the nominal delay. If the ANN-computed C2Q delay of a flip-flop is within the above ϵ limit, we say that the failing endpoint passes Tier 1.

For computing ANN-based C2Q delay, we extract the information of data slew, clock slew, setup skew, hold skew, and load capacitance from the STA tool. We take the worst-case value of these attributes by enquiring from the timer. Since STA already performed a worst-case analysis, the violation waiver need not recompute them. We pass these attributes to the ANN model, and the ANN model returns the computed delay. Note that this task is not time-consuming since the ANN model needs to perform only some arithmetic operations. We check whether the ANN-computed delay is within the ϵ tolerance of the nominal delay. If it is, we move to the Tier 2 checking; else, we declare that we cannot waive this violation.

4.2 SVM-based Checking:

The ML-based models are probabilistic. We have shown in Tab. 3.1 that ANN-computed C2Q delay can have around 2% errors compared to the golden SPICE delay, on average. Therefore, to reduce the risk of waiving a real violation, we use the SVM-based model for

Tier 2 checking. We pass the worst-case data slew, clock slew, setup skew, hold skew, and load capacitance obtained from the STA tool to the SVM model. The SVM-based model determines whether the flip-flop will be able to capture data or not. If it can capture the data, we move to the Tier 3 checking; else, we declare that we cannot waive this violation.

4.3 Surplus Slack in Fanout Check:

Consider a failing flip-flop $F1$ that launches data. Assume that the data launched by $F1$ is captured by the flip-flop $F2$. Assume that $F1$ passes Tier 1 and Tier 2 criteria. However, the increased C2Q delay of $F1$ can fail $F2$ if $F2$ is marginally safe. Moreover, the increased delay of $F1$ can make the slack of $F2$ more negative if $F2$ was already failing. However, if $F2$ has a sufficiently positive slack, we can tolerate the increased C2Q delay of $F1$. In effect, we can transfer the surplus slack of $F2$ to $F1$ and safely waive the violation at $F1$ in this case. We ensure that the circuit condition allows such surplus slack transfer in the Tier 3 check.

We perform a Tier 3 check by considering the flip-flop $F2$ that captures the data from the failing flip-flop $F1$ and exhibits the minimum slack. We obtain $F2$ with the help of the STA tool that has already performed the worst-case analysis. We compute the increased delay of $F1$ as follows:

$$\Delta = C2Q_{ANN} - C2Q_{NOM} \tag{4.1}$$

where $C2Q_{ANN}$ is the C2Q delay computed using ANN and $C2Q_{NOM}$ is the nominal C2Q delay obtained using the STA tool (that uses the traditional technology library). If the slack of $F2$ is more than Δ, the increased C2Q delay of $F1$ will not cause a violation in $F2$, and we consider $F1$ to pass Tier 3. If a failing flip-flop passes the Tier 3 check, we waive the corresponding violation; else, we declare that we cannot waive it.

Chapter 5

Results

We demonstrate the application of the proposed methodology on a few benchmark circuits of TAU CONTEST'19 [15]. We have implemented these circuits using NANGATE 15 nm open-cell libraries and NCSU FreePDK15 models. We developed ANN-based and SVM-based timing models for the flip-flops in these libraries. Then, we experimented with the violation waiving methodology described in the previous section on these designs. First, we show the result of the proposed methodology on a few paths of these circuits. Then, we present the results of the violation waiver for the entire circuit.

5.1 Waiving Violation

A portion of circuit from *leon3mp_iccad* is shown in Fig. 5.1. The flip-flop $memarr\_reg\_425\_\_20\_\_u0$ exhibits a timing violation (slack=-5.4 ps). With $\lambda=-25$ps, we consider it for waiving violation. First, we compute the ANN-based C2Q delay for this flip-flop and find it to be 16.3 ps. The nominal C2Q delay obtained using the STA tool is 15.2 ps. For an $\epsilon=25\%$, the Tier 1 checking condition is fulfilled. Then, we verify using the SVM-based timing model that the failing flip-flop can latch the data correctly. Next, we compute the increase in delay $\Delta=16.3-15.2=1.1$ ps. The minimum slack in the fanout of the failing flip-flop is exhibited by $data\_\_0\_\_4\_\_u0$ (slack=645.3 ps). Since there is sufficient surplus slack on the output side, Tier 3 condition is also

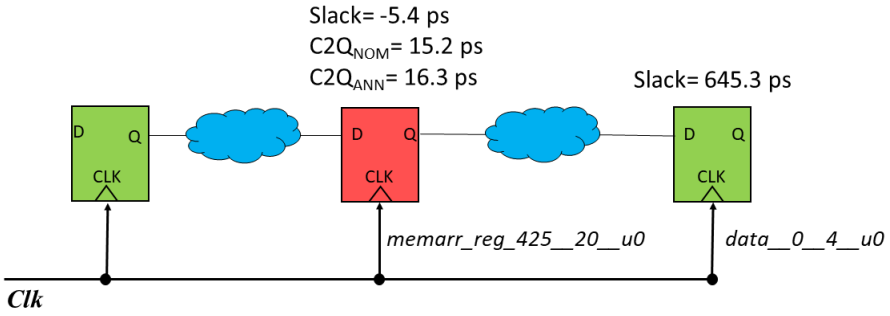


Figure 5.1: A portion of circuit from leon3mp_iccad.

fulfilled. Hence, we can waive this violation.

We validate the result produced by the proposed methodology using SPICE simulations. The SPICE simulations show that the failing flip-flop can latch the data correctly. Additionally, the SPICE-computed C2Q delay is 16.0 ps, and the ANN-computed delay agrees well with it.

5.2 Detecting capture problem using SVM-based model

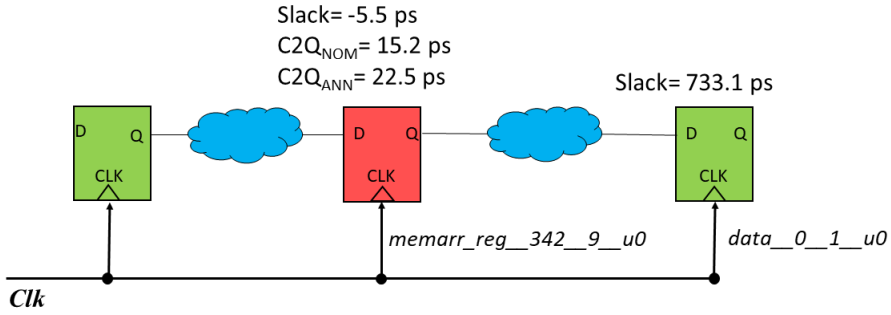


Figure 5.2: A portion of circuit from leon3mp_iccad.

A portion of circuit from *leon3mp_iccad* is shown in Fig. 5.2. The flip-flop $memarr\_reg\_342\_\_9\_\_u0$ exhibits a timing violation (slack=-5.5 ps). With λ=-25ps, we consider it for waiving violation. First, we compute the ANN-based C2Q delay for this flip-flop and find it to be 22.5 ps. The nominal C2Q delay obtained using the STA tool is 15.2 ps. For an ϵ=50%, the Tier 1 checking condition is fulfilled. Then, we

check using the SVM-based timing model whether the failing flip-flop will be able to latch the data correctly. However, we find that the SVM-based model predicts that the flip-flop will be unable to latch this data. Therefore, the failing point does not pass Tier 2, and we do not waive this violation. The increase in delay for this flip-flop is Δ=22.5−15.2=7.3 ps, but since it fails Tier 2 check, we do not check for Tier 3.

We validate the result produced by the proposed methodology using SPICE simulations. The SPICE simulations also show that the failing flip-flop will be unable to latch the data correctly.

5.3 Detecting violation in the fanout

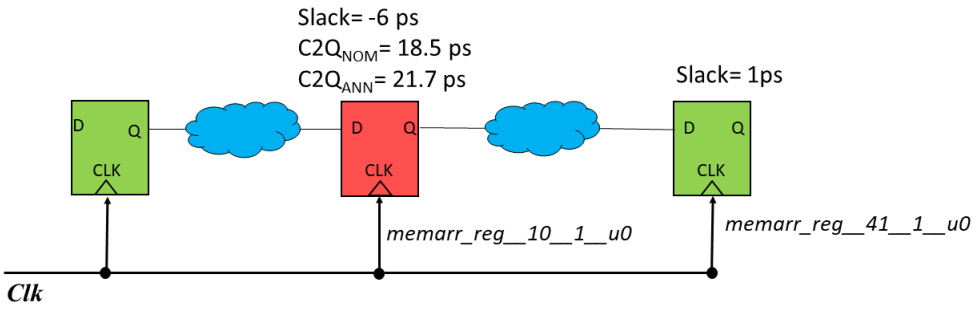


Figure 5.3: A portion of circuit from leon3mp_iccad.

A portion of circuit from *leon3mp_iccad* is shown in Fig. 5.3. The flip-flop $memarr\_reg\_10\_\_1\_\_u0$ exhibits a timing violation (slack=−6 ps). With λ=−25ps, we consider it for waiving violation. First, we compute the ANN-based C2Q delay for this flip-flop and find it to be 21.7 ps. The nominal C2Q delay obtained using the STA tool is 18.5 ps. For an ϵ=40%, the Tier 1 checking condition is fulfilled. Then, we verify using the SVM-based timing model that the failing flip-flop will be able to latch the data correctly. Next, we compute the increase in delay Δ=21.7−18.5=3.2 ps. The minimum slack in the fanout of the failing flip-flop is exhibited by $memarr\_reg\_41\_\_1\_\_u0$ (slack=1 ps). Hence, the increase in C2Q delay due to marginal timing violation will lead to a violation in the flip-flop $memarr\_reg\_41\_\_1\_\_u0$ since it does not have sufficient surplus slack (required surplus is 3.2 ps, while it has a surplus of only 1 ps). Therefore, the above violation fails Tier 3, and we do not waive this violation in the proposed methodology. We report the flip-flops that pass Tier 1 and Tier 2 but fail Tier 3 separately to the

designer so that it can be checked whether some surplus slack can be generated in the fanout of the failing flip-flop.

5.4 Results on benchmark circuits

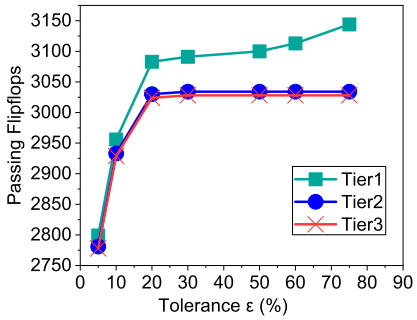
Next, we present the result of the proposed methodology on a few designs taken from the TAU contest 2019 mapped to NANGATE 15 nm open-cell libraries [22]. First, we develop the ANN-based and SVM-based flip-flop timing models for these libraries, as described in the previous sections. Then, we perform STA using the traditional technology library for each design and examine the marginal timing violations using the three-tier approach. We report the filtering percentage for each tier for these designs and demonstrate their effectiveness in the following paragraphs. We compute the filter percentage for a given tier using the following formula:

$$Filter\ \% = \left(\frac{X - Y}{X}\right) \times 100\ \% \tag{5.1}$$

where X is the total number of violations provided as an input to that tier, and Y is the total number of violations that passes that tier. A high filter percentage indicates that a high fraction of violations given as an input to that tier is found to be real violations. We have varied the designer-controlled parameters slack margin (λ) and delay tolerance (ϵ) to show their impact on the proposed mechanism of waiving violation.

Table 5.1: Effect of delay tolerance ϵ in leon3mp_iccad (#instances=1.25M, #flip-flops=109K, $\lambda = -25ps$, #violations=24.5K, #marginal violations=11.6K)

	Tier 1		**Tier 2**		**Tier 3**		**SPICE**
ϵ (%)	***Pass***	***Filter (%)***	***Pass***	***Filter (%)***	***Pass***	**Filter (%)**	**Mismatch**
5	2799	88.62	2781	0.64	2778	0.11	0
10	2956	87.98	2933	0.78	2929	0.14	0
20	3083	87.45	3030	1.72	3024	0.19	0
30	3091	87.43	3034	1.84	3028	0.19	0
50	3100	87.39	3034	2.12	3028	0.19	0
60	3113	87.33	3034	2.54	3028	0.19	0
75	3144	87.21	3034	3.49	3028	0.19	0



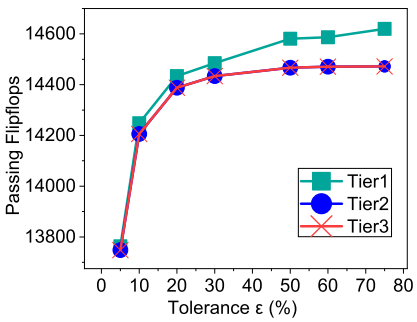
Tab. 5.1 shows the impact of changing delay tolerance ϵ on the design

leon3mp_iccad. For λ =−25 ps, 11619 flip-flops are found to have marginal timing violations. As we increase the value of ϵ, we tolerate more C2Q delay increase above the nominal C2Q delay, and more flip-flops pass ANN-based Tier 1 checking. However, some flip-flops that pass Tier 1 can have real timing violations, especially when we keep ϵ higher. We filter those false negatives in Tier 2. We observe that as we increase ϵ, less filtering is done by Tier 1, while more filtering is done by Tier 2. When ϵ is made greater than 30%, all the additional violations passed by Tier 1 get filtered by Tier 2, and we obtain a constant number of violations (3034) that pass both Tier 1 and Tier 2. These results indicate that even when Tier 1 does not filter real violations well, Tier 2 performs robust filtering. Further, some filtering is done in Tier 3 for cases where we do not have sufficient surplus slack in the fanout of the violating flip-flop. In the sign-off stage, these cases can be treated differently. Besides fixing a timing violation on the input side (D-pin side of the failing flip-flop), we can fix a violation that fails Tier 3 by generating sufficient slack on the output side (Q-pin side of the failing flip-flop) also. We reuse the ANN-based delay computed in Tier 1 in Tier 3. When ϵ is made greater than 30%, Tier 3 receives a fixed number of violations from Tier 2, and the filter percentage in Tier 3 becomes constant.

We verified using SPICE simulations that all the violations waived by this method are correct. Additionally, we verified that the violations filtered by the SVM-based model (Tier 2) are real violations and need to be filtered.

Table 5.2: Effect of delay tolerance ϵ in leon2_iccad (#instances=1.62M, #flip-flops=149K, $\lambda = -25ps$, #violations=94.4K #marginal violations=59.6K)

ϵ (%)	**Tier 1**		**Tier 2**		**Tier 3**		**SPICE**
	Pass	***Filter (%)***	***Pass***	***Filter (%)***	***Pass***	**Filter (%)**	**Mismatch**
5	13763	85.42	13748	0.11	13748	0	0
10	14248	84.90	14206	0.29	14206	0	0
20	14434	84.71	14388	0.32	14388	0	0
30	14485	84.65	14434	0.35	14434	0	0
50	14581	84.54	14467	0.78	14467	0	0
60	14587	84.54	14471	0.79	14471	0	0
75	14620	84.50	14472	1.01	14472	0	0



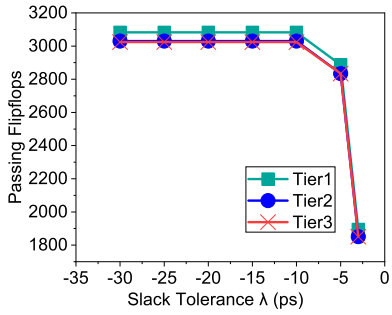
Tab. 5.2 shows the impact of changing ϵ on the design *leon2_iccad*. For λ=−25 ps, we observe 59554 marginal timing violations. We notice a similar trend in Tier 1 and Tier 2 filtering. However, no filtering is done in Tier 3 because all the flip-flops

that passed Tier 2 had sufficient surplus slacks in the fanout. We verify the correctness of the waived violations by comparing them with the golden SPICE simulations.

Tab. 5.3 shows the impact of changing slack margin λ on the design *leon3mp_iccad*, keeping ϵ fixed at 20%. As we increase the negative slack margin, we consider more failing flip-flops for waiving. As more failing flip-flops are received by Tier 1, initially, more violations pass Tier 1. However, when λ is reduced below -10 ps, flip-flops having large negative slacks are received by Tier 1. These flip-flops show significantly higher C2Q delay than the nominal C2Q delay and are filtered out by Tier 1 since ϵ is fixed at 20%. Hence, the number of violations passed by Tier 1 remains constant even when we reduce λ below -10 ps. Tier 2 performs some filtering on the violations. However, since Tier 1 filtering works very well for large negative slack violations, the filtering percentage of Tier 2 decreases with the decrease in λ. Some filtering is performed by Tier 3 also, depending on whether there is sufficient surplus slack on the fanout of the violating flip-flop. We verified the correctness of the waived violations by comparing them with the golden SPICE simulations.

Table 5.3: Effect of slack margin λ in leon3mp_iccad (#instances: 1.25M, #flip-flops: 109K, $\epsilon = 20\%$, #violations= 24.5K)

λ (ps)	Marginal Violations	Tier 1		Tier 2		Tier 3		SPICE Mismatch
		Pass	*Filter (%)*	*Pass*	*Filter (%)*	*Pass*	Filter (%)	
-3	1940	1894	92.29	1851	2.27	1851	0	0
-5	3173	2887	88.25	2834	1.84	2834	0	0
-10	5661	3083	87.45	3030	1.72	3024	0.19	0
-15	7827	3083	87.45	3030	1.72	3024	0.19	0
-20	9802	3083	87.45	3030	1.72	3024	0.19	0
-25	11619	3083	87.45	3030	1.72	3024	0.19	0
-30	13379	3083	87.45	3030	1.72	3024	0.19	0



Tab. 5.4 shows the results for some designs for varying ϵ and λ=-25 ps. We observe that the proposed methodology can waive 23%- 80% of marginal timing violations. Hence, the proposed methodology can significantly reduce the effort to achieve timing closure, especially during the last stages of a design flow when we want to eliminate the last few timing violations under a tight tape-out schedule. Moreover, SPICE simulations confirm that no real failure escapes in the proposed methodology. Additionally, the runtime required to waive these violations is very small (around 3 minutes to process 146K marginal violations). Thus, the results demonstrate that the proposed methodology effec-

tively removes artificial pessimism in timing analysis and is robust enough to be deployed in practical design flows.

Table 5.4: Results on a few benchmark designs (λ=−25 ps) (Run on 1.38 GHz 64-b Linux machine with 8-GB RAM)

S.No	Design Name	No.of FF/ Instances	Marginal Vio./ Total Violations	ϵ (%)	Tier 1 Pass	Tier 2 Pass	Tier 3 Pass	SPICE Mismatch	Waived Marg. Violations (%)	Runtime (s)
1	usb_funct	1.8K/31.6K	112/200	5	89	89	86	0	77	0.15
			112/200	10	92	92	89	0	80	0.15
2	vga_lcd	17K/0.14M	1761/8188	5	1294	1294	1294	0	73	2.04
			1761/8188	10	1308	1308	1308	0	74	2.04
3	leon3mp_iccad	109K/1.25M	11619/24584	5	2799	2781	2778	0	23	13.41
			11619/24584	10	2956	2933	2929	0	25	13.42
4	leon2_iccad	149K/1.62M	59554/94368	5	13763	13748	13748	0	23	77.15
			59554/94368	10	14248	14206	14206	0	24	77.17
			Total Marginal Vio.= 146092				Total Waived Vio.= 36438		Total Waived Vio.(%)= 25	Total= 185.53

Chapter 6

Conclusions and Future work

This work demonstrates the effectiveness of the ML-based flip-flop timing models in waiving marginal timing violations flagged by traditional STA tools. The data required for training/testing ML-based models can be collected during traditional library characterization. However, the proposed methodology requires additional effort in training and tuning the ML-based models. Nevertheless, we need to create the ML-based models only once while characterizing the library. Since we reuse the same library for multiple designs, this effort gets amortized over many designs, and the overall cost of creating ML-based models is reduced. Moreover, the proposed three-tier filtering mechanism can be easily integrated with the traditional STA tool and requires minimal changes in the design flows. Additionally, the runtime required to waive marginal timing violations is negligible, making the proposed methodology fast and efficient. In this work, we have shown results for only setup analysis. However, it can easily be extended to the hold analysis. Additionally, in the future, we can enhance the ML-based models to account for other phenomena such as PVT variations, SI analysis, and IR-drop analysis.

Bibliography

[1] Liberty library modeling [online] Available: `https://www.synopsys.com/community/interoperability-programs/tap-in.html` [Accessed: 29-May-2022].

[2] R. W. Phelps, "Advanced library characterization for high-performance ASIC," in *Proc. Fourth Annual IEEE International ASIC Conference and Exhibit*, Sep. 1991, pp. 15-3.1–15-3.4.

[3] M. A. Cirit, "Characterizing a VLSI standard cell library," in *Proc. of the IEEE Custom Integrated Circuits Conf.*, May 1991, pp. 25-7.1–25-7.4.

[4] M. Agarwal and S. Saurabh, "An Efficient Timing Model of Flip-Flops Based on Artificial Neural Network," *2021 ACM/IEEE 3rd Workshop on Machine Learning for CAD (MLCAD), 2021, pp. 1-6.*

[5] S. Saurabh, H. Shah and S. Singh, "Timing Closure Problem: Review of Challenges at Advanced Process Nodes and Solutions", *IETE Technical Review*, pp. 1-14, 2018.

[6] A. Kahng and H. Lee, "Timing margin recovery with flexible flip-flop timing model," *IEEE International Symposium on Quality Electronic Designs*, 2014.

[7] G. Rao and E. Howick, "Apparatus for optimized constraint characterization with degradation options and associated methods," June 24 2003, US Patent 6,584,598.

[8] E. Salman, A. Dasdan, F. Taraporevala, K. Kucukcakar, and E. G. Friedman, "Exploiting Setup–Hold-Time Interdependence in Static Timing Analysis," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 6, pp. 1114-1125, June 2007.

[9] E. Salman, A. Dasdan, F. Taraporevala, K. Kucukcakar, and E. G. Friedman. "Pessimism reduction in static timing analysis using interdependent setup and hold times," in *7th International Symposium on Quality Electronic Design (ISQED'06)*, pp. 6, 2006.

[10] S. Srivastava and J. Roychowdhury, "Interdependent latch setup/hold time characterization via Euler-Newton curve tracing on state-transition equations," *ACM/IEEE Design Automation Conference*, pp. 136-141, June 2007.

[11] S. Srivastava and J. Roychowdhury, "Independent and interdependent latch setup/hold time characterization via Newton-Raphson solution and Euler curve tracking of state-transition equations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 5, 2008.

[12] E. Salman and Eby G. Friedman. "Utilizing interdependent timing constraints to enhance robustness in synchronous circuits," *Microelectronics Journal*, 43, no. 2 (2012): 119-127.

[13] S. Hatami, H. Abrishami, and M. Pedram, "Statistical timing analysis of flip-flops considering codependent setup and hold times," *ACM Great Lakes Symposium on VLSI*, pp. 101-106, May 2008.

[14] Chen, Ning, Bing Li, and Ulf Schlichtmann. "Iterative timing analysis based on nonlinear and interdependent flip-flop modelling," *IET Circuits, Devices & Systems 6*, no. 5, pp. 330-337, 2012.

[15] TAU Contest 2019 (Resources). [online] Available: https://sites.google.com/view/tau-contest-2019/home/resources [Accessed: 29-May-2022].

[16] Fabian Pedregosa, Ga¨el Varoquaux, et. al, "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning research*, vol. 12, pp. 28252830, Nov. 2011.

[17] A. B. Kahng, "New directions for learning-based IC design tools and methodologies," *23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 405-410, 2018.

[18] M. Abadi, A. Agarwal, P. Barham, et. al, "TensorFlow: Large-scale machine learning on heterogeneous systems", Software available from tensorflow.org, 2015.

[19] O.V.S. Shashank Ram and S. Saurabh, "Modeling Multiple Input Switching in Timing Analysis using Machine Learning," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 4, pp. 723-734, April 2021.

[20] S. Saurabh, P. Jain, M. Agarwal, and O.V.S. Shashank Ram, "Applications of Machine Learning in VLSI Design," in *VLSI and Hardware Implementations Using Modern Machine Learning Methods*, pp. 125-139, CRC Press, 2022.

[21] S. J. Pan and Q. Yang, "A survey on Transfer Learning", *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, Oct. 2010.

[22] M. Martins, J. M. Matos, R. P. Ribas, A. Reis, G. Schlinker, L. Rech and J. Michelsen, "Open Cell Library in 15nm FreePDK Technology", in *Proc. of the International Symposium on Physical Design (ISPD)*, 2015.

Publication related to this work

This work has been submitted for publication in the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems journal and is currently under review.

Brief Description of Author

Rajat Kumar holds a degree of B.Tech in Electrical Engineering from Rajkiya Engineering College, Kannauj - Affiliated to APJ Abdul Kalam Technical University, Lucknow in the year 2019. He completed his M.Tech. in Electronics and Communication Engineering from Indraprastha Institute of Information Technology, Delhi, in 2022. His interests lie in the application of ML in VLSI Design. He is working as an intern at Synopsys India Pvt Ltd, Noida. He can be reached at rajat20321@iiitd.ac.in or rajatkmr768@gmail.com.