Crosstalk Prediction in Integrated Circuits Based on Machine Learning Techniques

Gor Abgaryan

October 9, 2023

Crosstalk Prediction In Integrated Circuits Based On Machine Learning Techniques

Gor Abgaryan
Institute of Physics
Yerevan State University
Synopsys Armenia Educational Department
Yerevan, Armenia
abgaryan.gor98@gmail.com

***Abstract*—Unintentional signal coupling between adjacent wires known as crosstalk is a common problem in integrated circuits (IC) and became major with operating frequencies rise and circuit dimensions decrease. Performance decline, signal distortion, and functional failures could all result from this phenomenon. Hence, having reliable crosstalk prediction and reduction mechanisms is a crucial aspect of IC design.**

Machine learning (ML) is currently a widely utilized technique in prediction algorithms. The suggested approach combines crosstalk analysis and ML to explore ways to predict crosstalk and reduce disturbances in ICs taking as input the physical design of IC.

Training data for the ML model is collected from the parsing algorithm of IC information. Experiments are done for different types of designs (standard cells, memories, etc.). As a result, the trained ML model provides approximately 90% pass rate.

***Keywords*—Crosstalk, crosstalk prediction, signal integrity, machine learning, deep learning, neural network.**

I. Introduction

Crosstalk is a common occurrence in ICs, especially as operating frequencies continue to rise, circuit dimensions continue to decrease, and more devices are organized in ICs. A signal switching of one net creates crosstalk noise at its neighboring nets by capacitive coupling. Noise amplitude can reach up to 30% of the design source voltage. Crosstalk may cause data corruption, timing mistakes, increased power consumption, and even leakage of secure information and functional failures in ICs [1], [2]. As a result, controlling and reducing crosstalk has grown to be an essential part of contemporary IC design to produce high-performance and dependable electronic systems.

There are several works introducing an overview of the crosstalk phenomenon, crosstalk identification and estimation, proposing a crosstalk analysis model [3]- [8], as well as multiple works proposing crosstalk reduction and prediction techniques including clock tree optimization, repeater insertion, shielding, skewing, etc [9]- [14].

The clock tree optimization is proposed for crosstalk reduction [9]. A new approach to clock tree synthesis was proposed with non-default rules for clock nets. The combination of non-default rules allowed for the reduction of crosstalk in circuits. The rules affect on distribution of clock nets and the cells, which decreases the routing congestion.

Crosstalk noise can be reduced via shielding, which is both practical and common [10]. Placing ground or power lines alongside a victim signal line is a ubiquitous technique for shielding that lowers noise and delays uncertainty (Fig. 1). When a shield is added, the crosstalk between two connected interconnects is frequently ignored, which significantly underestimates the coupling noise.

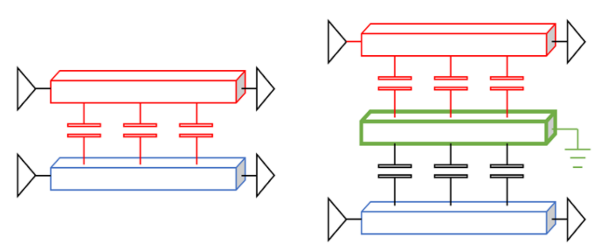

Fig. 1. Shielding in interconnects

In order to sustain the signal strength in lengthy interconnects, repeater insertion is a strategy [10]. The time delays brought on by lengthy interconnects in Very large-scaled integrated (VLSI) circuits might be decreased by using repeaters. Crosstalk is further reduced upon signal restoration at the repeater node as the time delay is decreased and the signal is also restored after each repeater (Fig. 2).

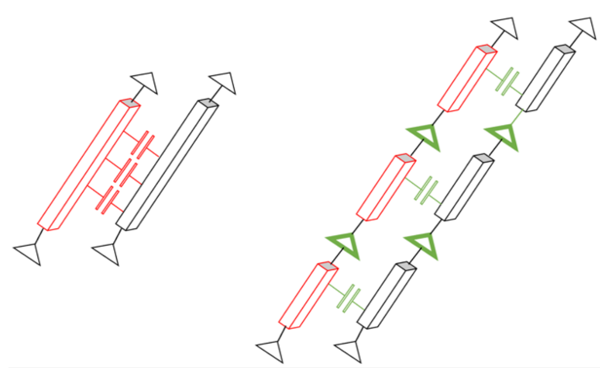

Fig. 2. Repeater insertion in interconnects

When drivers switched in different directions as opposed to the same direction, there was a greater propagation delay and crosstalk noise. Skewing is a static delay that is added to signal propagation. Thus, the skewing of drivers led to a faster switchover time, which in turn decreased crosstalk [10].

Besides crosstalk reduction, a crosstalk prediction mechanism is proposed as well [14]. The experimental data of crosstalk critical net categories is shown with a Venn diagram (Fig. 3).

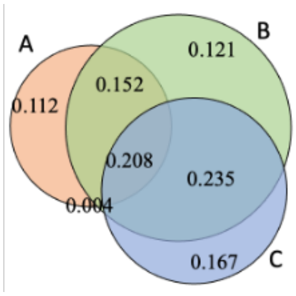

Fig. 3. The Venn diagram of three crosstalk-critical nets with:
A: Nets with large coupling capacitance,
B: Nets with large crosstalk inducted noise,
C: Nets with long incremental delay

The crosstalk modeling flow is constructed based on the experimental results (Fig 4).

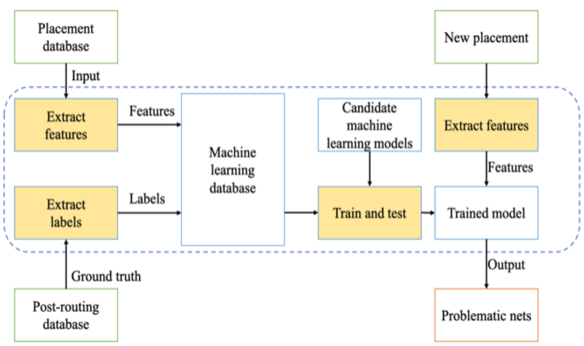

Fig. 4. Crosstalk modeling flow

The placement and post-routing databases are used to extract input features and ground truth data, which together create the ML database. The most efficient feature sets and the best models for three crosstalk classification problems are identified through training and evaluating the prediction performance of candidate ML models using the labeled data stored in the ML database. These feature sets and models can be used to quickly identify problematic nets in new placement instances.

A part of artificial intelligence called machine/deep learning (ML/DL) enables computers to learn patterns from data, adapt, and enhance their performance over time. It has transformed the field of IC by allowing systems to make predictions and judgments based on data [15].

II. The Proposed Crosstalk Prediction Method and Implementation

The inputs of the proposed mechanism are the design GDSII file and the layer mapping file for the given GDSII. As an output, the model will detect whether there are crosstalk effects for the selected pair of metals. Overall flow has two main parts:

- data collection (Fig. 5)
- ML/DL model construction and training (Fig. 6)

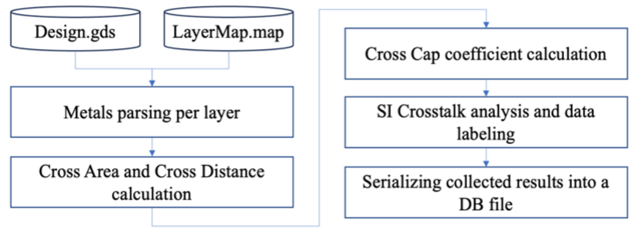

Fig. 5. Data collection

The data collection flow contains several iterations. The metals parsing iteration contains the metal layer and coordinates detection and extraction from GDSII and LayerMap files. As an output, not only the information of metals' coordinates, but their mutual positions as well are available. Cross area and cross distance calculation is done based on the output of the first iteration and calculated for each layer. Next is the cross-cap coefficient calculation. Crossing area and distance between metal pairs allow for calculating the crossing capacitance coefficient between them. The main iteration of the data collection flow is Signal Integrity (SI) analysis and data labeling. This iteration runs the SI analysis. The output information is used to label pairs of metals per layer and determine which neighbor pairs have crosstalk. The identification values for crosstalk existence or absence between gathered pairs are logical "1" and "0" accordingly. Currently, the SI analyzer will take care of the signal switching activity or shielded metals, including their existence or absence in the output information.

The final step of data collection is serializing collected results into a database (DB) file. The DB file has XLSX/CSV format file which will be used for ML model training and verification.

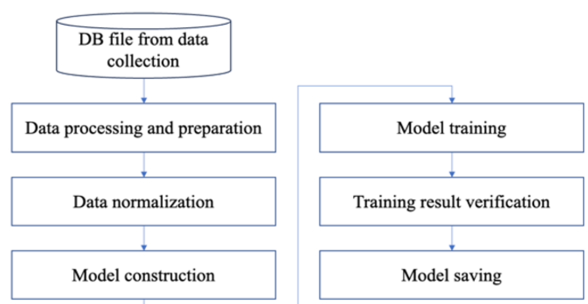

Fig. 6. ML/DL model construction and training

The ML/DL model construction and training flow contains several iterations:

A. Data processing and preparation

This iteration takes the output DB file from the data collection flow as input and starts data processing and preparation, including duplicate, incomplete data drops and string type data processing. For last, one-hot encoding method is used to convert string to digital format.

B. Data normalization

The results of data processing and preparation are normalized to be an input for the neural network model. The linear scaling normalization technique is used [16]. This will transform features to a specific range ([0, 1]), ensuring all features have the same scale.

C. Model construction

One of the valuable parts of neural network model construction is a suitable selection of neuron activation functions [17]. During model construction two types of neuron activation functions are used: Parametric ReLU and sigmoid (Fig. 7).

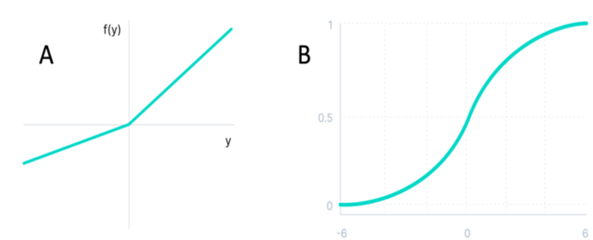

Fig. 7. A) Parametric ReLU function, B) sigmoid function

The next part of model construction is building the model architecture. Three hidden layers have been used for the DL model. The model type is a branched neural network (Fig. 8).

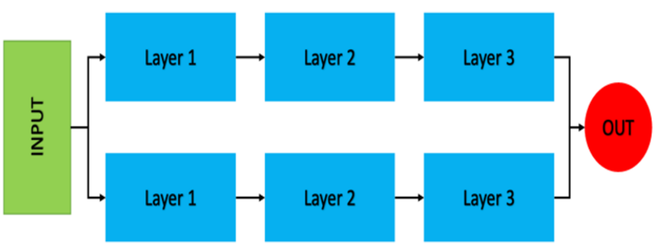

Fig. 8. Model architecture

There are several reasons for the current architecture selection:

- Feature specialization: Each branch concentrates on a distinct attribute, boosting the model's capacity to identify a variety of complex patterns.
- Hierarchical learning: The model can understand complex interactions because different branches learn at varying levels of abstraction.
- Enhanced representation: The overall representation of the input data in the model is enhanced by the fusion of outputs from distinct branches.
- Improved generalization: Reduced over-fitting improves the model's performance on unobserved data due to specialized branches.
- Resource efficiency: The best possible use of computing resources is achieved through parallel processing.
- Flexibility: Branches can be added or changed to accommodate changing data circumstances or application needs.

D. Model training, result verification, saving

The training application is an automated flow which encapsulates the entire logical sequence of the results' extraction via ML model (Fig. 9).

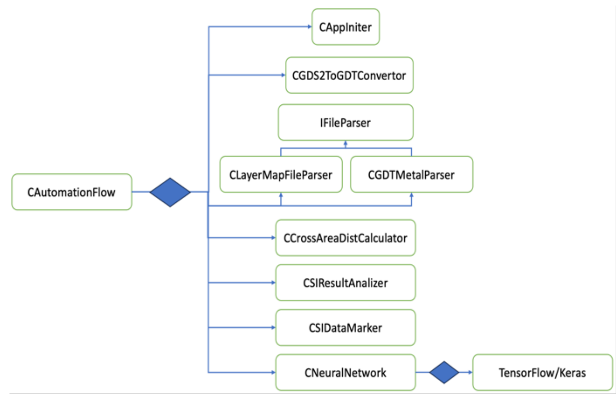

Fig. 9. The application's high level design UML diagram

Each step of the automation flow represented as an independent software module (Fig. 10), which can be modified and replaced with another one fast and easily due to application's modular structure.

```
flow = CAutomationFlow()
flow.init(config_file_path)
flow.gds2gdt(gds_path, gdt_path)
flow.parseLayerMapFile(layerMap.map, parsed_layes_output_path)
flow.metalExtraction(parsed_layes_output_path)
flow.metalsCrossAreaDistanceCalculation()
flow.SIAnalyzes(eda_tool_script, si_analyzes_results_path)
flow.dataLabeling()
flow.modelConstructionAndTraining()
flow.modelSaving('crosstalkmodel.h5')
```

Fig. 10. Pseudo-code for execution flow of application

The usage flow of the model reads the saved model from encrypted container (docker), providing the GDSII and LayerMap files to the predictor tool.

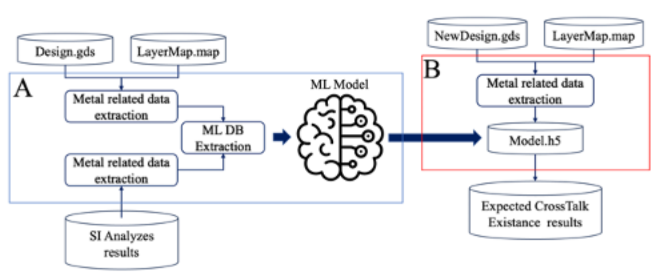

Fig. 11. The proposed method implementation flow with:
A) Data collection and model training,
B) Model evaluation and crosstalk prediction

After the entire data processing flow as data collection (Fig. 1), the processed data must be provided to the trained model. The result is crosstalk existence prediction for neighbor metal pairs. The tool will exclude the pairs that do not have crosstalk effect and will generate report with crosstalk affected metal pairs for each metal layer in design (Fig. 11).

III. Model Training Results

The model training had been done with about 90% pass rate. Mini-batch training technique was used to train the model due to large training data set. Training was done on different data and address sized embedded memories and had been successfully tested on another bunch of memories and standard cells. During training process, some utilities, such as TensorFlow-TensorBoard were used to log and extract training related information, especially training accuracy (Fig. 12) and loss function (Fig. 13) curves.

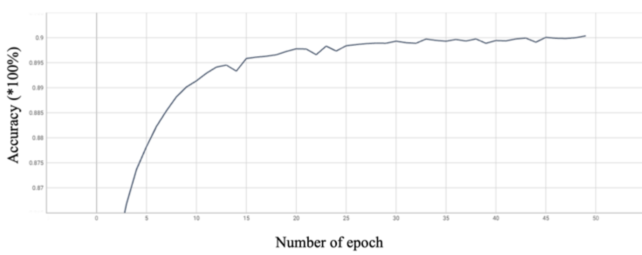

Fig. 12. Training accuracy curve

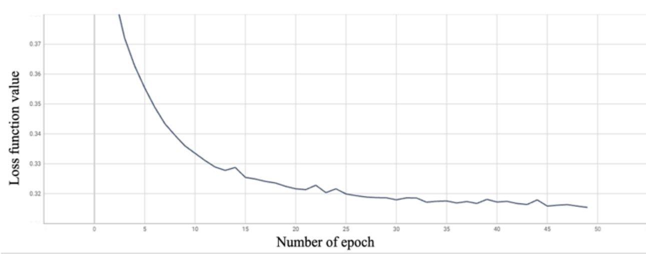

Fig. 13. Loss function curve

Training validation data set accuracy curve as well had been extracted (Fig. 14).

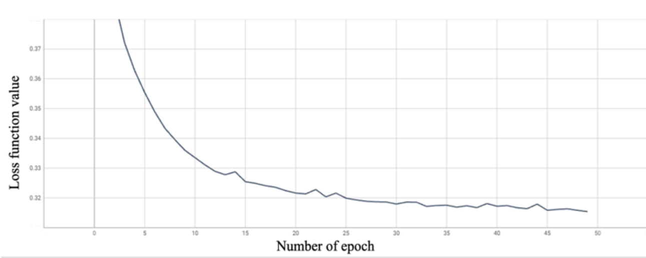

Fig. 14. Model training validation accuracy curve

IV. Model Evaluation Results

The model evaluation accuracy was checked for several designs (Table 1) with the same clock frequency value. The results show that 90% training pass rate remains for other designs with the same design parameters.

TABLE I
Model evaluation results for several designs

Design	**Metal pair count**	**Crosstalk count**	**Right predictions**	**Accuracy (%)**
sram_4x4	52318	3105	2826	91,01
sram_8x8	114726	7819	6880	87,99
sram_64x16	1198216	23402	20827	89,00
sram_64x64	1784573	31115	27847	89,50
sram_128x8	1865918	33121	30153	91,04
sram_256x128	7675412	81272	72900	89,70
100 STD cells	15415	282	269	95.39

V. Discussion About Further Optimization and Improvements for the Proposed Crosstalk Prediction Method

Currently the model considers the GDSII file for the designs that contain only straight metals per metal layer due to nowadays technology nodes limitations. But there are a lot of technology nodes which are actual up today but takes smaller part of the chip design market. Such nodes accept folded metal usage in the design, so to cover this part of designs, the purposed solution and model structure/training flow can be enhanced.

The next possible enhancement point is more proper and deep usage of the GDSII parsed output file. That file gives opportunity to extract the port list of the cells. The additional data can be provided to the flow which will include the mapping of input signal names and their functionalities (Tags). Based on that information the training data set can be enriched by adding port related data, which will include data type and data switching activity.

Currently the model supports crosstalk detection only for metals per layer as mentioned above. The new input will enable crosstalk prediction based on the connected metals between layers. Some of newer technology nodes allow having metal cut in the GDSII output file which is presented with separate layer. Different metals per layer will be presented as a single metal and cut layers. The further improvement will consider these cut layers as well and consider the metal layer as a separate metal.

VI. Comparison Of the Proposed Crosstalk Prediction Method with Existing Methods

The routing-free crosstalk prediction method has been already implemented [14]. The inputs of that method are the designs placement and post-routing databases, out of which the machine learning DB is created, which becomes training material for ML model. The advantage of the mentioned method is predicting the possible crosstalk effects before having the full GDSII file for the design. This gives opportunity to have maximum crosstalk aware placement due to the early staged detected results.

Meanwhile, the proposed crosstalk prediction method uses fully designed GDSII file, which allows detecting possible crosstalk for the entire design hierarchies, including routing related crosstalk effects. As well, GDSII file at any stage of digital design flow may be used as an input.

Modern place and route family tools usually use the computational methods to calculate crosstalk effect between nets or pins. Computational methods require resources such as CPU/GPU and memory, and the entire design crosstalk detection take huge time. The market requires that the time to fully implement a chip be shorter. Nevertheless, the computational methods provide certain results, over time the run-time of computational crosstalk detection methods will no longer meet market requirements. The proposed method has much higher performance than computational methods, however it provides the possibilities of crosstalk existence.

VII. Conclusion

In conclusion, the ML technique-based method is introduced and implemented for crosstalk prediction in integrated circuits. By providing the circuit GDSII file and layer mapping file, the crosstalk effect can be predicted with approximately 90% pass rate. With the technology node change, only metals parsing per layer iteration from data generation flow may be changed, and the rest of the iterations can be the same, which is the flexibility and versatility of the method. The potential use cases of the model are place and route family tools. The predicted possible crosstalk points can be transferred to that tool with corresponding format (due to flow's flexibility) and the tool can automatically resolve crosstalk related issues. The model can be potentially used in the signal integrity analyzer tools as well as a crosstalk prediction engine.

References

[1] T. Jerse, "Introduction to Crosstalk," 2017 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI), Washington, DC, USA, 2017, pp. 1-45, doi: 10.1109/ISEMC.2017.8078014.

[2] D. Bucur, "Crosstalk - An overview for high speed design," 2013 21st Telecommunications Forum Telfor (TELFOR), Belgrade, Serbia, 2013, pp. 701-704, doi: 10.1109/TELFOR.2013.6716325.

[3] B. K. Kaushik and S. Sarkar, "Crosstalk Analysis for a CMOS-Gate-Driven Coupled Interconnects," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 6, pp. 1150-1154, June 2008, doi: 10.1109/TCAD.2008.923259.

[4] Xiao, Fengchao & Nakada, Yohei & Murano, Kimitoshi & Kami, Yoshio. (2007). Crosstalk analysis model for trace crossing split ground plane and its reduction by stitching capacitor. Electronics and Communications in Japan (Part II: Electronics). 90. 26 - 34, doi: https://doi.org/10.1002/ecjb.20338.

[5] Norte, Andrew. (2015). Near-End Crosstalk Considerations For Coupled Microstriplines, doi: 10.13140/RG.2.1.4903.3684.

[6] Norte, Andrew. (2015). Far-End Crosstalk Considerations For Coupled Microstriplines With Even And Odd Mode Switching, doi: 10.13140/RG.2.1.1495.4969.

[7] Papandreou, Nikolaos & Antonakopoulos, Theodore. (2005). Far-End Crosstalk Identification Method Based on Channel Training Sequences. Instrumentation and Measurement, IEEE Transactions on. 54. 2204 – 2212, doi: 10.1109/TIM.2005.858578.

[8] Maniruzzaman, Md & Ahmed, Shakil & Fattah, Galib & Toma, Rafia. (2015). Estimation of crosstalk noise for RLC interconnects in deep submicron VLSI circuit. 1-4, doi: 10.1109/ICEEICT.2015.7307503.

[9] G. Abgaryan, A. Galstyan, G. Harutyunyan, "Reducing the consequences of the wire interconnections in integrated circuits by optimizing the clock tree", 2023, v76.1-93, doi: 10.53297/0002306X-2023.

[10] Kaur, Damanpreet & Sulochana, V.. (2013). Crosstalk Minimization in VLSI Interconnects. International Journal of Reconfigurable and Embedded Systems (IJRES). 2. 89, doi: 10.11591/ijres.v2.i2.pp89-98.

[11] Lee, J.-H. A Novel Meander Split Power/Ground Plane Reducing Crosstalk of Traces Crossing Over. Electronics 2019, 8, 1041, doi: https://doi.org/10.3390/electronics8091041

[12] Ginis, George & Peng, Chia-Ning. (2006). Alien Crosstalk Cancellation for Multipair Digital Subscriber Line Systems. EURASIP journal on advances in signal processing. 2006, doi: 10.1155/ASP/2006/16828.

[13] P. Milosevic, J. E. Schutt-Ainé and W. T. Beyene, "Crosstalk mitigation of high-speed interconnects with discontinuities using modal signaling," 19th Topical Meeting on Electrical Performance of Electronic Packaging and Systems, Austin, TX, USA, 2010, pp. 21-24, doi: 10.1109/EPEPS.2010.5642534.

[14] Rongjian Liang, Zhiyao Xie, Jinwook Jung, Vishnavi Chauha, Yiran Chen, Jiang Hu, Hua Xiang, and Gi-Joon Nam. 2020. Routing-free crosstalk prediction. In Proceedings of the 39th International Conference on Computer-Aided Design (ICCAD '20). Association for Computing Machinery, New York, NY, USA, Article 163, 1–9, doi: https://doi.org/10.1145/3400302.3415712.

[15] "Introduction to Machine Learning Models." Keboola, https://www.keboola.com/blog/introduction-to-machine-learning-models

[16] "Normalization in Data Preparation." Google Developers, https://developers.google.com/machine-learning/data-prep/transform/normalization

[17] "Neural Networks Activation Functions." V7 Labs, url: https://www.v7labs.com/blog/neural-networks-activation-functions#h1