Tackling Signal Electromigration with Learning-Based Detection and Multistage Mitigation

Wei Ye
ECE Department, UT Austin
weiye@utexas.edu

Mohamed Baker Alawieh
ECE Department, UT Austin
mohdbaker@utexas.edu

Yibo Lin
ECE Department, UT Austin
yibolin@utexas.edu

David Z. Pan
ECE Department, UT Austin
dpan@ece.utexas.edu

ABSTRACT

With the continuous scaling of integrated circuit (IC) technologies, electromigration (EM) prevails as one of the major reliability challenges facing the design of robust circuits. With such aggressive scaling in advanced technology nodes, signal nets experience high switching frequency, which further exacerbates the signal EM effect. Traditionally, signal EM fixing approaches analyze EM violations after the routing stage and repair is attempted via iterative incremental routing or cell resizing techniques. However, these "EM-analysis-then fix" approaches are ill-equipped when faced with the ever-growing EM violations in advanced technology nodes. In this work, we propose a novel signal EM handling framework that (i) incorporates EM detection and fixing techniques into earlier stages of the physical design process, and (ii) integrates machine learning based detection alongside a multistage mitigation. Experimental results demonstrate that our framework can achieve 15× speedup when compared to the state-of-the-art EDA tool while achieving similar performance in terms of EM mitigation and overhead.

1 INTRODUCTION

As integrated circuit (IC) technologies continue to scale, electromigration (EM) comes forth as one of the prominent reliability issues challenging the design of robust circuits [1]. Complex chip functionalities have been made possible by virtue of increasing transistor densities and aggressive scaling of interconnects. However, these two factors bring along higher current densities in metal wires, a phenomenon that further exacerbates EM. Particularly, high current densities lead to the *migration* of atoms in metal wires resulting in opens and shorts over time [2]. Hence, the continuous drive toward extreme scaling will keep compounding the EM problem especially for signal nets that are expected to switch at gigahertz speed, making EM design closure a challenging task [3, 4].

Addressing the EM challenge requires a two-step process: (i) violations detection and (ii) EM mitigation. Conventionally, EM checking tools are invoked after the detailed routing stage [5, 6]. These tools compare the current densities in metal wires with technology-specific design rules to detect EM violations. Next, the violations are fixed with engineering change order (ECO) efforts [7]. EM checking tools leverage post-routing information to detect violations, which consequently limits the efficiency of their mitigation techniques. In the routing phase, the locations of standard cells and the corresponding current distribution are already fixed and the traditional fixing approaches such as wire widening and cell resizing are not effective enough to handle the ever-growing number of violations in signal wires [4]. In fact, the methodology of "EM-analysis-then-fix" is becoming obsolete at advanced nodes [8], which makes it of vital importance to incorporate EM detection and fixing techniques into earlier stages of physical design (PD).

Two clear benefits are associated with such early stage EM handling. First, the number of EM violations can be decreased as the result of using a larger set of mitigation techniques. Second, introducing early stage mitigation techniques can help reduce the resulting overhead when compared to post-routing fixing techniques. Thus, moving the EM detection and resolving steps to earlier stages of the physical design can help in reducing runtime or the number of iterations needed for design closure. Towards this goal, and given the critical role placement plays in current distribution, we propose a placement-based EM detection and adjustment framework.

EM failure has been dealt with at different design stages, including placement [9] and routing [10–14]. However, the focus has been concentrated on applying optimization techniques after EM violations are already detected, or using approximation methods to guess possible violations. In this work, we propose a novel EM hotspot detection and mitigation framework based on information available at the placement phase. In particular, three main steps constitute our proposed approach. As a first step, a classification model is trained through machine learning techniques to detect EM hotspots based on features extracted from the placement scheme. This model can be trained using data obtained from designs where EM hotspots are already known, and then, it can be applied to detect hotspots in new designs. In addition to its main role in hotspot prediction, the model helps identify the placement-based features that are critical for hotspot identification. Knowing these features is fundamental for constructing effective EM adjustment techniques at the placement stage.

In the second step, the placement scheme is adjusted by incorporating EM hotspot mitigation mechanism in the cost function of the placement problem. This mechanism incorporates the detection model information about critical features to address the EM

ASPDAC '19, January 21–24, 2019, Tokyo, Japan

ACM ISBN 978-1-4503-6007-4/19/01…$15.00
https://doi.org/10.1145/3287624.3287688

hotspots. At the end of this step, a new placement is obtained. As a last step, the classification model is used again to detect hotspots still present after placement adjustment and non-default routing (NDR) rules are applied to address these hotspots in the routing stage.

Our main contributions are summarized as follows:

- A novel EM hotspot detection and mitigation framework has been proposed based on information available at the placement stage, which enables early-stage EM handling and reduces iterative EM fixing cost.
- An accurate cascaded logistic regression model is proposed to detect signal EM hotspots at the placement stage.
- A multistage EM mitigation approach is proposed to address the problematic nets detected by the classification model.
- Experimental results demonstrate that our framework can achieve nearly the same EM violation reduction as the state-of-the-art PD tool while achieving 15× speedup.

The rest of this paper is organized as follows. Section 2 gives an overview of the proposed framework. Sections 3 and 4 provide a detailed explanation of the EM detection model building process and mitigation techniques. Section 5 demonstrates the effectiveness of our approaches with comprehensive results, followed by conclusion in Section 6.

2 PROPOSED APPROACH: AN OVERVIEW

The key idea of our proposed approach is to leverage machine learning techniques to detect EM hotspots at placement stage and exploit the trained models to guide EM mitigation. An overview of the framework is presented in Figure 1.

Figure 1(a) shows the process of training an EM hotspot prediction model. Starting from the input netlist of the training set, a physical design (PD) tool is used to get the placement result. Next, routing and EM evaluation are performed to get the EM hotspots in the designs. Finally, the placement information is used along with the EM hotspot results to train a classification model for EM detection.

Figure 1(b) demonstrates the application of the EM detection and mitigation framework. After having a trained model for EM detection, PD tool is used to do placement, and then EM hotspots are predicted using the classification model given the placement-related features. Next, placement is incrementally updated to mitigate predicted EM hotspots. Then, the EM detection model is used again to detect remaining hotspots that are finally routed using NDR rules.

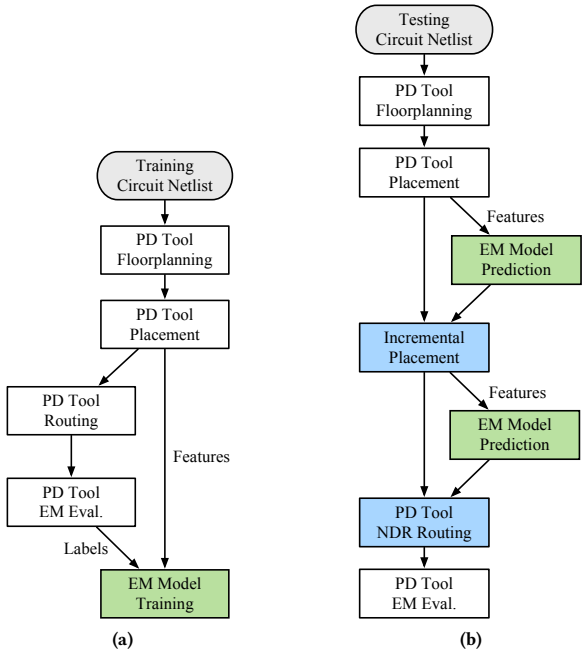

Figure 1: An overview of the hotspot detection model training (a) and its application in EM detection and mitigation (b) is shown.

3 MACHINE LEARNING FOR EM DETECTION

3.1 Features Extraction

Despite the fact that the current profile for the design is not available at the placement stage, multiple features that are highly correlated with the current can be crafted. To elaborate on this, we consider the three nets in Figure 2, A, B and C. One can expect net A to have the highest current density. This is mainly because, unlike the 2-pin nets B and C, A is a 6-pin net connected with two large cells. On the other hand, net C is the one least prone to EM. In practice, although both B and C are 2-pin nets, Figure 2 clearly shows that the neighborhoods around the pins of net B are more congested (i.e., high pin density). This in turn can lead to detours when routing net B; hence, longer wires, large wire capacitance and higher current.

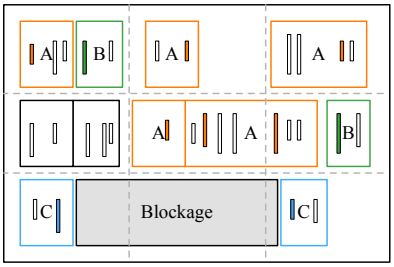

Figure 2: An illustration of a placement scheme with three nets is shown.

In our approach, we extract a set of features from the placement to be used for training the model for EM detection. These features can be divided into two categories: (i) net-specific features and (ii) neighborhood related features. The net-specific features used are the following:

(1) Net half-parameter wirelength (HPWL)
(2) Number of net pins
(3) Net switching activity
(4) Maximum fall transition time
(5) Maximum capacitance
(6) Circuit frequency

On the other hand, neighborhood related features are used to capture information about possible congestion around net pins. To define these features, the placement region is divided into a grid with fixed window size as shown by the gray-colored grid in

Figure 2. Then, for each net, a set of features is defined over all grid windows containing pins connected to the net.

Using Figure 2 as an example, we consider the feature defined as the average number of pins. To compute this feature for net A, we first identify the grid windows containing pins of net A which are the three windows in the first row, and second and third window in the second row counting from the left. Then, we average the number of pins in the five windows counting all pins in the windows, not only those connected to net A. This results in a feature value equal to $\frac{21}{5}$ for net A. Computing the same feature for nets B and C gives 5 and 2 respectively. The full list of neighborhood related features used is as follows:

(1) Average number of pins
(2) Average number of cells
(3) Average cell area
(4) Average area capacity (space not occupied by blocks)
(5) Average number of placement sites

It is important to note that all the features mentioned above can be extracted without any knowledge about the final routing scheme. Moreover, with the exception of switching activity that can be obtained through high-level hardware simulation, all features can be extracted from the placement scheme.

3.2 Data Preparation

Starting from the labeled training set, features defined in the previous section are extracted resulting in a feature vector with a Boolean class label for each net in the design. Two important characteristics of the resulting dataset should be examined. First, the dataset is significantly imbalanced. In other words, the EM hotspot class (H) is enormously outnumbered by the non-hotspot class (NH). Secondly, the different features have different ranges of values. For instance, HPWL has a wider range of possible values compared to the number of pins. These two characteristics can affect the training process and the interpretability of the model, and hence, they should be addressed before training.

In the scenario where the two classes are imbalanced, the training is expected to be biased towards the objective of learning the larger class while neglecting the errors in predicting the smaller one. Among the methods used to address such bias is class weighting where higher weights are given to instances in the smaller class when formulating the training objective. This can be done by associating different costs with mispredicting instances from different classes; i.e., mispredicting an instance from the smaller class is associated with higher cost compared to mispredicting an instance from the larger one.

On the other hand, having features with different ranges of values can affect both the model training and its interpretability. During training, numerical issues arising from such case can cause convergence problems. In addition, in distance-based classification models, different ranges of values can result in unwanted weighting for the features. Moreover, having features with different ranges makes the task of interpreting any model more challenging. For example, important features in a trained model are usually inferred from the weight given to each feature after the training phase. For the case where all features have similar ranges, it suffices to compare the absolute values of the weights to judge upon the importance of the features. However, with features taking values in different ranges, this comparison does not hold any more. Therefore, a normalization step is done to map all features to the $[0, 1]$ interval to ensure they all have the same weight when training the EM detection model.

3.3 Cascaded Model for False Alarm Avoidance

The EM detection problem can be cast into a classification problem. In practice, a wide range of classification models are available for use, and these models vary in their complexity and application space [15]. Two important characteristics of the EM detection application contribute to the decision upon the classification model to use. First, the problem is a binary classification problem (i.e., two class problem) with relatively small number of features. Secondly, the EM detection model is a part of an EM detection and mitigation framework. Hence, in addition to the detection task, we are interested in analyzing the trained model to arrive at the features contributing the most to the prediction decision. Knowing these features plays a significant role in the EM mitigation process described in the next section. Therefore, the interpretability of the trained model is critical from this perspective.

In practice, as the complexity of the classification model increases, interpretabilty becomes more challenging. And since the problem at hand is low-dimensional, we choose to use logistic regression [15, 16] as the classification model. Such model is known to behave well with binary classification problems and its regression coefficients can be used to interpret the importance of the different features.

As will be demonstrated in the result section, logistic regression can achieve high EM detection accuracy at a small false alarm rate. However, by examining the overall flow of the EM detection and migration framework and the relative number of H and NH instances, false alarm rate should be addressed from a different perspective. Technically, in a general classification problem, correctly labeling 99% of the target group (H in our case) with 3% false alarm rate can be acceptable. However, given that the two groups are unbalanced, even a 3% false alarm rate can result in a number of false alarms that is a multiple of that of H instances.

Hence, with such number of false alarms, mitigation techniques will perform a large number of unnecessary changes to the placement and routing schemes; thus, introducing additional overheads. To address this issue, we introduce the two-stage detection approach shown in Figure 3. In the first stage, a classification model M1 is trained to detect EM hotspots using all the nets in the training dataset. After the first stage, all nets with NH prediction will be labeled as NH without further processing. For nets labeled H by M1, a new model, M2, is trained to prune out false alarms. M2 is trained using nets in the training dataset labeled H by M1. For those nets going through the second stage, the final label will be the prediction of M2.

In practice, when two models are trained, inference for new nets can be done in a way analogous to the training process. First, an initial prediction is obtained by applying M1, and if the prediction is NH the net is given that as the final label. Otherwise, a new prediction is obtained from M2, and the final label is that generated by M2.

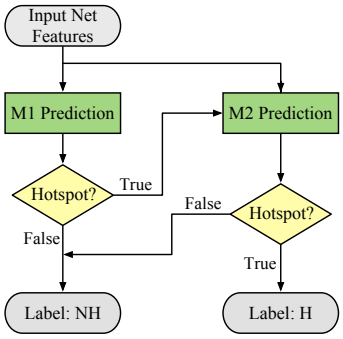

Figure 3: The flow of the two stage detection approach is shown.

This proposed approach helps reduce the number of false alarms while preserving the interpretability characteristic of the model. This translates to reducing the overhead incurred by the mitigation process.

4 MACHINE LEARNING GUIDED EM MITIGATION

4.1 Placement Adjustment

Besides its main role in detecting nets susceptible to EM failures, the trained EM detection model points out the potential directions to mitigate them The coefficients in the trained model indicate that wirelength and cell density, the two features that can be optimized in the given placement, contribute significantly to EM severity. Therefore, we propose an incremental placement approach to mitigate signal EM violations with minimal perturbation to the layout. The major purpose of this technique is to achieve selective wirelength reduction and cell density improvement.

Similar to a timing-driven placement [17], a net n_i is assigned a weight w_i based on its EM criticality. The higher the weight assigned to a given net is, the more is the push by the placer to reduce its wirelength. Considering cell density as well, the cost for a cell move is defined as:

$$\text{wHPWL}(1 + \alpha \cdot c_d), \tag{1}$$

where wHPWL is the weighted wirelength sum of all the nets connected to this cell, i.e., $\text{wHPWL} = \sum_i w_i \, \text{HPWL}(i)$, and c_d denotes the cell density cost computed according to [18, 19].

The incremental placement scheme is summarized in Algorithm 1. After detailed placement, PD tools are able to output high-quality placement results in term of timing, power, and routability of a design, which serve as the starting point for our signal EM optimization. As a first step, the trained EM prediction model is used to detect the set of EM hotspot nets $\mathcal{H}$ in the input placement scheme. Next, the set of cells $\mathcal{C}$ connected by the nets $\mathcal{H}$ is identified and reordered by their area. At this stage, the objective of the proposed incremental placer is to move the cells in $\mathcal{C}$ in a way to minimize the wirelength of the nets in $\mathcal{H}$ and mitigate the cell density around the target cells.

The principal idea is to find a search region for a cell in the placement region and move the cell to the best location in this region. Different from the classical optimal region calculation method [20], we use the weighted median to compute the optimal region for cell move since the nets have different weights in the current scheme. Then, the optimal region is extended to larger search region.

Algorithm 1 Cell Move

Input: Initial placement, predicted EM hotspot nets $\mathcal{H}$;
1: $\mathcal{C} \leftarrow$ set of cells connected by $\mathcal{H}$;
2: Reorder $\mathcal{C}$;
3: **repeat**
4: **for** $c_i \in \mathcal{C}$ **do**
5: Determine the search region of c_i;
6: Move c_i to the position that minimizes the objective;
7: **end for**
8: **until** converged or maximum iteration reached
9: Legalize placement;

We perform wirelength optimization to improve both wirelength and density until less than 1% of the target cells are moved in an iteration or the maximum number of iterations is reached. After that, legalization is performed to remove possible overlaps.

4.2 Non-Default Routing for EM Adjustment

While the aforementioned incremental placement algorithm is tailored to address the EM hotspots, it does not guarantee the mitigation of all detected hotspots. In other words, some EM hotspots can be still present after the incremental placement adjustment stage. For example, for the nets with high fanouts, the current flowing through the main metal branch drives large capacitive loads, therefore, improving wirelength is not effective enough to solely resolve the current issue.

However, we can still utilize the EM prediction model after the proposed incremental placement. That is, we can set the router to route those predicted hotspot nets with wider widths to avoid iterative fixing. Practically, this option is readily available in many PD tools through the non-default routing (NDR) rule option. As the name implies, NDR applies non-default routing geometries to those selected nets in the design based on user specification; i.e., instead of the default single-width single-spacing (1W1S) scheme, a user set scheme can be used to route specific nets in the design. This option is leveraged to address the EM hotspots detected by the model in Section 3.3 after incremental placement using a double-width single-spacing (2W1S) NDR rule.

5 EXPERIMENTAL RESULTS

Throughout the experiments, TSMC 40nm CMOS physical design kit (PDK) [21] was used for evaluating the efficacy of our proposed framework. Moreover, slow process, voltage and temperature (PVT) corners were used to generate a worst-case EM environment. The five benchmark circuit netlists used are taken from ICCAD 2014 placement contest [19] and OpenCores [22] respectively. In addition, physical design was performed using Synopsys IC Compiler (ICC) 2017 [7].

5.1 EM Prediction Model Comparison

Among the five available designs, three were used to train the hotspot detection model, while the remaining two were used for testing. The training data set consisting of designs `b19`, `ecg`, and

mmm contains a total of 426152 nets of which 2681 are hotspots. Meanwhile, designs med and vga with 298197 nets, including 648 hotspots, are used for testing.

The confusion matrix summarizing the evaluation of the model on the testing data when a single stage logistic regression (M1) was used is shown in Table 1. While the results demonstrate high true positive rate, the number of false alarms is more than 10× the number of actual hotspots. On the other hand, Table 2 shows the confusion matrix when the cascaded model (M1+M2) described in Section 3.3 was used. One can notice a reduction of 65% in the number of false alarms at the cost of missing 21 of the hotspots. This cascaded model provides a compromise between the high accuracy of the hotspot detection and the overhead induced from fixing nets wrongly labeled as hotspots. The details will be demonstrated in Section 5.3.

Table 1: Confusion matrix of M1.

	NH	H
$\widehat{\text{NH}}$	290084	2
$\widehat{\text{H}}$	7465	646

Table 2: Confusion matrix of M1+M2.

	NH	H
$\widehat{\text{NH}}$	295178	23
$\widehat{\text{H}}$	2371	625

5.2 Incremental Placement + NDR

As mentioned earlier, our mitigation flow consists of two steps: incremental placement and NDR. We performed the two mitigation techniques on the five designs to verify the effectiveness. The placement algorithm in Section 4.1 was implemented in C++. During EM mitigation at the placement stage, at most 6 incremental placement iterations were allowed in the experiments, and the parameter α in formulation (1) was set to 1. We set the same weight w for all the hotspot nets as $w = 2000/|\mathcal{H}|$, while keeping unity weight for NH nets. The information of the detected hotspots is provided to ICC for performing NDR.

To demonstrate the efficacy of each individual component in the proposed multistage mitigation framework, we run several flows as described below and the results are summarized in Table 3. We first run clock tree synthesis (CTS) and default routing on the initial placement generated by the PD tool, and the number of the final EM violations without any repair approaches is shown under the column "Initial". Second, we performed incremental placement with the actual EM hotspots being known, and then run CTS and routing. The number of final EM violations under this flow is under column "Incr. place". Lastly, we run the incremental placement, CTS and NDR routing and reported the final number of EM violations under the column "Incr. place + NDR route". Note that the target EM violations to repair for all the flow shown in Table 3 are reported from ICC EM evaluation. It can be observed that incremental placement solely reduces 37.1% of the violations on average, and incremental placement and NDR routing reduces 74.1% violations, which is about the same performance achieved when using EM fixing in PD tool (73.1%).

Table 3: Comparison of PD tool EM repair, the incremental placement flow, and incremental placement combined with NDR flow in terms of final EM violations is shown.

Design	Initial	PD tool	Incr. place	Incr. place + NDR
b19	302	104	260	108
ecg	225	3	70	21
mmm	2,154	1,637	1,997	1,320
med	252	6	34	6
vga	396	80	360	83
Avg. improve	—	73.1%	37.1%	74.1%

5.3 Framework Validation

The EM detection model was trained on three benchmarks, b19, ecg and mmm, and we integrated the trained model into the proposed framework, which is applied to the five benchmarks. Table 4 reports the number of EM violations, routed wirelength (Wirelength), net area (Area), overall runtime, worst negative timing slack (WNS), and total negative timing slack (TNS) at the end of detailed routing in different flows. "Initial" denotes the default PD flow without any EM fixing attempts, while "PD fixing" denotes using wire widening throughout during the fixing stage. "M1" and "M1+M2" represent the proposed EM detection and mitigation flow with M1 and M1+M2 as the EM prediction model respectively.

We can see that the proposed M1+M2 flow fixes about the same number of EM violations as the PD fixing flow. It also achieves 15× speedup compared with the PD fixing flow. The runtime decomposition for the PD tool fixing flow and our proposed flow with the cascaded model is shown in Figure 4. One can see that, compared to PD fixing flow whose runtime is dominated by post-route EM fixing, the M1+M2 flow can perform the incremental placement in less than 10 seconds and NDR takes nearly the same runtime as the default routing.

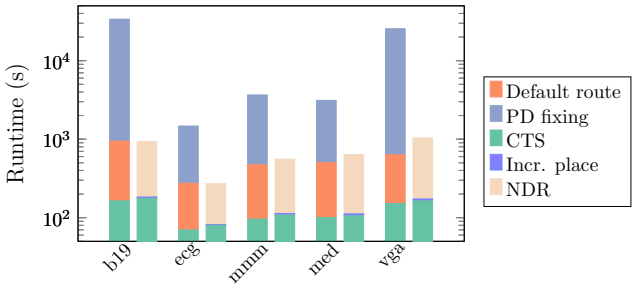

Figure 4: Runtime comparison between the PD fixing flow (CTS + default route + PD fixing) and the proposed M1+M2 flow (incr. place + CTS + NDR) is shown.

6 CONCLUSION

In this work, we propose a novel EM hotspot detection and mitigation framework using learning-based detection and multistage mitigation. Utilizing features extracted from the placement, a classification model is proposed to detect EM hotspots in the design. In addition, an incremental placement strategy is proposed to mitigate the detected EM hotspots. EM hotspots still present after the

Table 4: Comparison of EM violation reduction, metal wirelength and area overhead, and timing impact for the designs produced by the conventional EM fixing flow and our proposed methodology using machine learning trained model.

Design	Flow	#EM Vio.	Wirelength (um)	Area (um^2)	Runtime (s)	WNS (ns)	TNS (ns)
	Initial	302	2,242,990	165,284	943.4	-0.11	-0.85
b19	PD fixing	104	2,260,090	188,171	33,865.1	-0.09	-3.83
Nets: 219,289	M1	116	2,368,201	221,096	1,293.5	-0.09	-0.61
	M1+M2	120	2,248,412	174,432	935.5	-0.10	-0.77
	Initial	225	873,557	63,050	274.1	-0.17	-81.00
ecg	PD fixing	3	874,541	63,420	1,470.1	-0.17	-81.97
Nets: 48,337	M1	3	996,912	67,072	420.6	-0.23	-83.2
	M1+M2	9	884,589	65,727	270.7	-0.20	-91.88
	Initial	2,154	1,823,239	132,646	471.8	-0.15	-9.43
mmm	PD fixing	1,637	1,824,374	138,799	3,663.9	-0.15	-9.43
Nets: 158,526	M1	1,245	1,872,680	191,545	724.4	-0.16	-10.52
	M1+M2	1,364	1,847,248	143,328	556.5	-0.16	-10.83
	Initial	252	2,638,638	190,443	504.3	-0.19	-135.53
med	PD fixing	6	2,642,620	199,231	3,124.5	-0.19	-141.38
Nets: 133,222	M1	11	2,746,344	260,535	12,971.9	-0.23	-161.12
	M1+M2	11	2,655,013	207,188	635.2	-0.22	-148.43
	Initial	396	3,169,437	227,432	633.7	-0.21	-116.74
vga	PD fixing	80	3,227,050	268,042	25,529.2	-0.23	-144.39
Nets: 164,975	M1	84	3,306,214	300,918	1,113.9	-0.17	-84.71
	M1+M2	87	3,228,308	272,295	1,038.8	-0.18	-86.02
	PD fixing	0.269	1.006	1.083	19.10	0.983	1.760
Ratio wrt initial	M1	0.246	1.062	1.307	6.38	1.052	0.955
	M1+M2	0.267	1.011	1.093	1.21	1.087	1.004

placement-stage mitigation are addressed through the NDR scheme in the routing stage. Contrary to conventional EM mitigation flows, the proposed approach addresses the EM problem at an earlier stage in the PD process resulting in faster closure and versatile mitigation techniques.

ACKNOWLEDGMENT

This work is supported in part by HiSilicon, and the authors would like to thank Dr. Liang Peng, Canhui Zhan, Yongsheng Sun and Yiwei Fu for helpful discussions.

REFERENCES

[1] J. Lienig, "Electromigration and its impact on physical design in future technologies," in *ACM International Symposium on Physical Design (ISPD)*, 2013, pp. 33–40.
[2] J. R. Black, "Electromigration — a brief survey and some recent results," *IEEE Transactions on Electron Devices (TED)*, vol. 16, no. 4, pp. 338–347, 1969.
[3] A. B. Kahng, S. Nath, and T. S. Rosing, "On potential design impacts of electromigration awareness," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2013, pp. 527–532.
[4] "Addressing signal electromigration (EM) in today's complex digital designs," https://www.eetimes.com/document.asp?docid=1280370.
[5] B. Li, P. Muller, J. Warnock, L. Sigal, and D. Badami, "A case study of electromigration reliability: From design point to system operations," in *IEEE International Reliability Physics Symposium (IRPS)*, 2015, pp. 2D.1.1–2D.1.6.
[6] "ANSYS RedHawk," https://www.ansys.com/products/semiconductors/ansys-redhawk.
[7] "Synopsys IC Compiler," http://www.synopsys.com.
[8] M.-K. Hsu, N. Katta, H. Y.-H. Lin, K. T.-H. Lin, K. H. Tam, and K. C.-H. Wang, "Design and manufacturing process co-optimization in nano-technology," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2014, pp. 574–581.
[9] W. Ye, Y. Lin, X. Xu, W. Li, Y. Fu, Y. Sun, C. Zhan, and D. Z. Pan, "Placement mitigation techniques for power grid electromigration," in *IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, 2017.
[10] J. Lienig, "Introduction to electromigration-aware physical design," in *ACM International Symposium on Physical Design (ISPD)*, 2006, pp. 39–46.
[11] X. Chen, C. Liao, T. Wei, and S. Hu, "An interconnect reliability-driven routing technique for electromigration failure avoidance," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, vol. 9, no. 5, pp. 770–776, 2012.
[12] I. H.-R. Jiang, H.-Y. Chang, and C.-L. Chang, "WiT: optimal wiring topology for electromigration avoidance," *IEEE Transactions on Very Large Scale Integration Systems (TVLSI)*, vol. 20, no. 4, pp. 581–592, 2012.
[13] H.-B. Chen, S. X.-D. Tan, V. Sukharev, X. Huang, and T. Kim, "Interconnect reliability modeling and analysis for multi-branch interconnect trees," in *ACM/IEEE Design Automation Conference (DAC)*, 2015, pp. 90:1–90:6.
[14] J. Pak, B. Yu, and D. Z. Pan, "Electromigration-aware redundant via insertion," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2015, pp. 544–549.
[15] C. M. Bishop *et al.*, *Pattern Recognition and Machine Learning*. Springer New York, 2006, vol. 4, no. 4.
[16] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.
[17] C. J. Alpert, D. P. Mehta, and S. S. Sapatnekar, *Handbook of Algorithms for Physical Design Automation*. CRC press, 2008.
[18] M.-C. Kim, N. Viswanathan, Z. Li, and C. Alpert, "ICCAD-2013 CAD contest in placement finishing and benchmark suite," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2013, pp. 268–270.
[19] M.-C. Kim, J. Hu, and N. Viswanathan, "ICCAD-2014 CAD contest in incremental timing-driven placement and benchmark suite," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2014, pp. 361–366.
[20] M. Pan, N. Viswanathan, and C. Chu, "An efficient and effective detailed placement algorithm," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2005, pp. 48–55.
[21] "TSMC 40nm Technology," http://www.tsmc.com/english/dedicatedFoundry/technology/40nm.htm.
[22] "OpenCores," https://opencores.org.