2021 26th IEEE European Test Symposium (ETS)
- Hot-Topic Paper -

Unsupervised Learning in Test Generation for Digital Integrated Circuits

Soham Roy, Spencer K. Millican, and Vishwani D. Agrawal
Department of Electrical and Computer Engineering
Auburn University, Auburn, AL 36849-5201
{*szr0075, millican, agrawvd*}*@auburn.edu*

2021 IEEE European Test Symposium (ETS) | 978-1-6654-1849-2/20/$31.00 ©2021 IEEE | DOI: 10.1109/ETS50041.2021.9465403

***Abstract*—The exponential complexity of automatic test pattern generation (ATPG) necessitates the use of heuristics in making choices during test generation. However, in practice no single heuristic fits all situations. Unsupervised learning can combine any number of known heuristics, such as input-output distance (logic depths), gate type, fanout information, and testability measures like Controllability and Observability Program (COP) and Sandia Controllability/Observability Analysis Program (SCOAP) through principal component (PC) analysis, and then the major PC can guide ATPG choices. This study combines three heuristics, distance, COP, and SCOAP. Some heuristic data are complemented and two major PC are obtained. These PC guide backtrace directions in a PODEM ATPG program. For most circuits, the number of backtracks either matches the best of the three heuristics or is lower than all.**

***Index Terms*—ATPG, digital testing, unsupervised learning, principal component analysis, tracing, heuristics**

I. Introduction

Depending on context, the definition of machine intelligence (MI) is, either 1) data-driven trends in deep learning, or 2) mimicking human cognitive capacities. Such biologically-mimicking intelligence aims to solve cognitive tasks by replacing human decisions or expensive analytical processes. MI has found numerous applications across scientific domains such as computer vision, speech recognition, forensic analysis, autonomous vehicles, and many more. Integrated circuit (IC) testing is one such field, as existing literature testifies [1]–[6]. Robust, dependable, and resilient IC manufacturing requires substantial investment, but much data in IC manufacturing processes remains unexplored. Data mining can extract meaningful correlations using MI, leading to changes away from conventional test techniques, and result in cost-effective and better test quality.

II. Prior Work

Nearly two decades ago, artificial neural networks (ANNs) were first applied to test generation: they modeled digital integrated circuits with neurons representing the values of signals [7]. When this ANN model is modified by an injected fault, the minimum energy state (i.e., the stable state) of the network produces a test in the form of primary input (PI) states. However, using this model for automatic test pattern generation (ATPG) requires a physical ANN or a software model, which causes practicality issues: the ANN energy function, being non-linear, has many local minima making the test search difficult for some faults.

Backtrace guidance for tracing the best path is an important problem in ATPG [8]. The use of human intuition, as *heuristics*, has long been practiced. But, it is realized that no single heuristic works best for all situations and proposals for using multiple heuristics in a program can be expensive [9], [10].

Recent work [11] has demonstrated that MI using a supervised learning algorithm in the form of an ANN combines multiple heuristics. This ANN can then replace any heuristic in PODEM [8] and speedup the ATPG. However, that work [11] lacked a formal ANN training method. Further work [12] produced a structured and formal methodology to train the ANN, effectively improving the previously reported performance [11]. But, combining several heuristics increases the training data volume that overloads the ANN to the extent that its efficiency suffers.

Principal component (PC) analysis (PCA) [13], [14] can combine training features to enhance supervised learning of ANNs in the MI-based PODEM ATPG. The concept is still under research, since ANN training time and complexity may add to ATPG cost from the use of PCs. This work uses unsupervised learning to guide a test generator that outperforms conventional heuristic-based PODEM ATPG in terms of reduced backtracks and CPU time, with no dependency on any ANN or training.

III. Present Contribution

Machine intelligence (MI) has two phases: learning from problem-specific data and then using that knowledge to solve problems. In supervised learning, these phases may be ANN training and ANN guidance. In contrast, unsupervised learning would use statistical tools such as PCA [13], [14] and k-means clustering [15]–[18]. In the first phase, the tool analyzes the problem-specific data to extract relevant characteristics, which in the second phase, directly help to solve problems.

We apply unsupervised learning to the ATPG problem using the principal component analysis (PCA) as the statistical tool.

A. Dimensionality Reduction

One could say that data has become more precious and expensive than crude oil. To meet the challenges of storage and computation, data mining and pruning techniques are sought. In spite of the discovery of the principal component analysis (PCA) and its extensions almost a century ago [13], [14], its demand burgeoned when computer-based applications spread across multiple disciplines. PCA is a statistical technique that drastically reduces the dimensionality of data through new variables known as principal components (PC). Each PC is a linear function of the original data, which maximizes the variance of uncorrelated data while preserving statistical information. Evaluating PCs instead of original data narrows down

decision making to an eigenvalue/eigenvector (also known as eigen decomposition method) or singular value decomposition (SVD) [16]. PCs can be chosen based on either a covariance matrix or correlation matrix, and the choices are independent of any pre-defined functions [16]. In short, PCA is descriptive and adaptive rather than inferential.

This study uses SVD to obtain PCs based on a parameter known as *explained variance* [19]. PCs represent the same amount of information as the original data, i.e., the original data can be restored from the PCs. Moreover, the total variances of the original and transformed data are the same but are redistributed unequally among the PCs. The first PC (also known as the major PC) has the highest variance, as shown in Fig. 1. The standard quality measure *explained variance* π_j of the jth PC is the ratio of its variance λ_j to the total variance (sum of variances of all PCs):

$$\pi_j = \frac{\lambda_j}{\sum_{i=1}^{p} \lambda_i}$$

where, p is the number of PCs and λ_i is the individual variance of ith PC. The progressive nature of PCs means that a proportion of total explained variance for a subset S of q PCs is expressed as a percentage of the total variance: $\sum_{i \in S} \pi_i$. It is a common practice to set a threshold for this total variance to decide how many PCs to use; only first one, two, or three PCs may be required. However, there are circumstances, such as outlier detection [16] or image analysis, where the last few PCs may be of interest.

B. Multi-Heuristic Guidance for ATPG

This study combines multiple heuristics, such as, shortest distance D to primary inputs [8], COP controllabilities [20], and SCOAP testability measures [21] in the present illustration. From 0 and 1 signal probabilities of COP, CC0 and CC1, we only use CC1 because of their complete dependence on each other as CC0 = 1 − CC1. SCOAP 0 and 1 combinational controllabilities are denoted here as SC0 and SC1. For every node in the circuit four quantities, D, CC1, SC1, and SC0, are computed using known linear-time algorithms [8], [20], [21]. Each quantities is normalized to [0,1] range with respect to its maximum value over all nodes.

The same heuristic data have been used in supervised learning-based PODEM ATPG [11], [12]. There, a backtrace choices are directed by a trained artificial neural network (ANN) that computes relative metrics for the available nodes. Highest metric implies best chance of finding a test without a backtrack. For training the ANN, the information about how each heuristic influences success is derived from ATPG runs on sample circuits. In the unsupervised learning system, however, no ANN is used. Instead, multiple heuristic data are combined through PCA for directly guiding the backtrace.

C. Principal Component Analysis (PCA)

In combining heuristics, it is necessary that they work cooperatively without contradicting each other in comparing the effectiveness of inputs of a logic gate while justifying the output value. Table I shows how individual heuristic works. For example, consider a backtrace through an AND gate with two or more inputs being guided by D. To justify the output value 0, the backtrace must take the input closest to primary inputs (PI) [8]. In Table I, this is indicated by "min" under D for AND gate and value = 0. To justify a 1 at the output, the backtrace follows the input with highest D, shown as "max". We observe that the four heuristics do not agree for any of the gates. Hence, if combined by PCA, the major component (P) cannot be given guidance criteria.

TABLE I
HEURISTIC-BASED INPUT SELECTION CRITERIA FOR BACKTRACING THROUGH A GATE TO JUSTIFY OUTPUT VALUE.

Gate	**D**		**CC1**		**SC1**		**SC0**		**P**	
***Value*→**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
AND	min	max	min	min	max	max	min	min	**?**	**?**
NAND	max	min	min	min	max	max	min	min	**?**	**?**
OR	max	min	max	max	min	min	max	max	**?**	**?**
NOR	min	max	max	max	min	min	max	max	**?**	**?**

TABLE II
PRINCIPAL COMPONENTS (P0 AND P1) FOR GATE OUTPUT = 0 AND 1. *Italicized* DECISION CRITERION (*min* OR *max*) SHOWS COMPLEMENTED HEURISTIC DATA TO ACHIEVE SYNCHRONIZATION.

Gate	**D**		**CC1**		**SC1**		**SC0**		**P0**	**P1**
***Value*→**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
AND	min	max	min	*max*	*min*	max	min	*max*	**min**	**max**
NAND	max	min	*max*	min	max	*min*	*max*	min	**max**	**min**
OR	max	min	max	*min*	*max*	min	*max*	*min*	**max**	**min**
NOR	min	max	*min*	max	min	*max*	*min*	max	**min**	**max**

Table II takes a two-step approach to overcome the above difficulty. First, selected heuristic data are complemented. For example, when AND gate output is 1, at its inputs, CC1 is replaced with 1 − CC1 and SC0, with 1 − SC0. Also, when the AND gate output is 0, SC1 is replaced by 1 − SC1. This reverses the corresponding backtrace criteria now shown in *italics*. Similar changes are made for NAND, OR and NOR gates, giving complete synchronization of the choice criteria for all heuristics. However, it necessitates separate PCAs for gate outputs 0 and 1, respectively, requiring two major PCs, P0 and P1, with corresponding backtrace criteria (see Table II).

D. Preprocessing and ATPG

To run ATPG, circuit netlist is preprocessed to compute four values for each signal node, namely, D [8], CC1 [20], and SCOAP combinational measures SC0 and SC1 [21]. Complemented values are computed according to Table II and P0 and P1 are from PC analyses for all gate outputs assumed as 0 and 1, respectively. PCA results for ISCAS'85 [22] and ITC'99 [23] benchmarks are shown in Figures 1 and 2. The blue bars show the major PCs, P0 and P1.

IV. EXPERIMENTAL RESULTS

A workstation containing an Intel i7-8700 processor and 8 GB of RAM performed all experiments. Tools were implemented in C++ using the MSVC++ 14.15 compiler with maximum performance optimization, and all PCA activities were executed in Python. PODEM ATPG [8] is reproduced along with event-driven fault simulation [24] such that any heuristic (distance [8], COP [20], SCOAP [21], or PC) can be applied across ISCAS'85 [22] and ITC'99 [23] benchmark circuits without favoring a single heuristic.

This study will surely polarize electronic design automation (EDA) vendor mindset to deploy MI in their ATPG software.

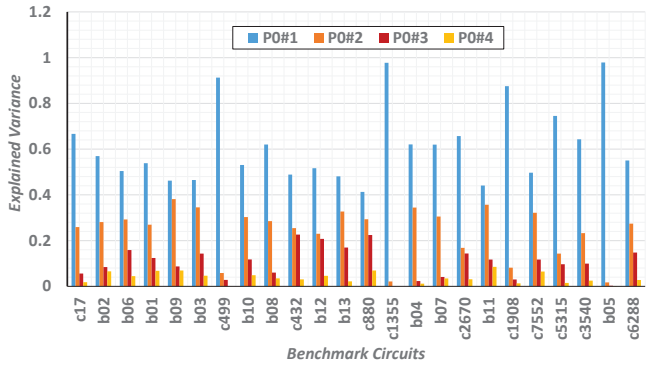

Fig. 1. PCA for ISCAS'85 and ITC'99 benchmarks. Heuristic data are complimented according to Table II assuming 0 output for all gates. The major PC, P0, is shown in blue.

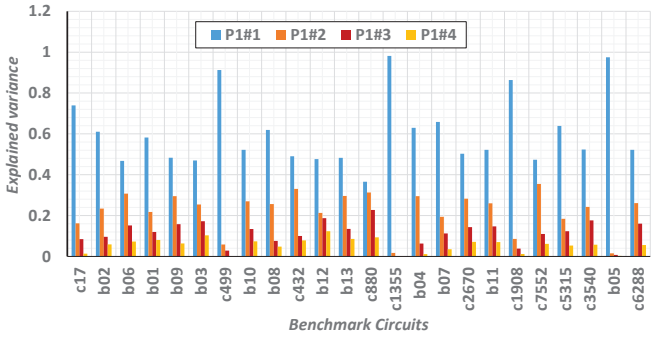

Fig. 2. PCA for ISCAS'85 and ITC'99 benchmarks. Heuristic data are complimented according to Table II assuming 1 output for all gates. The major PC, P1, is shown in blue.

However, EDA vendors hesitate to divulge their program source code, and it is impossible to conduct research-based experiments using the executable. Therefore, this study prefers to run the experiments using the in-house EDA tools.

Experiments used testable and redundant faults to prove the efficacy of guidance provided by PCA to PODEM ATPG [8]. Figure 3 (circuits are arranged by logic depths) and Figure 4 show relevant findings on ATPG CPU time (ms) and the number of backtracks with respect to distance [8], COP [20], SCOAP [21], and PCA (P0 and P1 guidance).

This experiment demonstrates how combining multiple heuristics into a linear combination through PCA can achieve better ATPG CPU time and fewer backtracks compared to conventional single heuristics. Circuits c1355, c2670, c3540, b04, b11, b08, c499 and c6288 show improvement of reduced backtracks, but these circuits needed more backtraces, thus CPU time increased. Most frequently, PCA is the best guidance for ATPG, but when it is not it is never the worst performing. Circuits c17, b02, b01, and b06 have no reconvergent fanouts, and therefore have no scope for reducing backtracks. Circuit c880 is the good example of zero backtracks in PCA-based PODEM ATPG compared to other conventional heuristics, which is significant in terms of the ability to achieve no backtracks.

V. Discussion and Future Work

MI, big data, and data mining are hot-topics with ample media coverage, upcoming start-up companies, and outstanding mergers and acquisitions. In the past few decades, MI

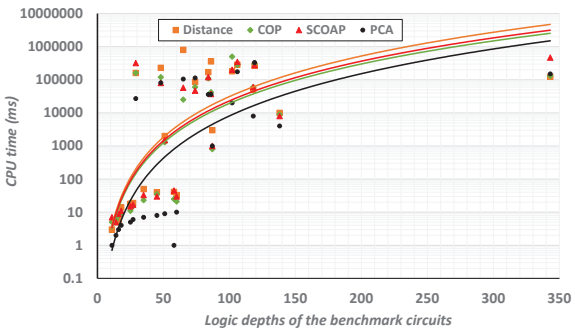

Fig. 3. CPU time for detecting all faults with ATPG using conventional heuristics and PCA.

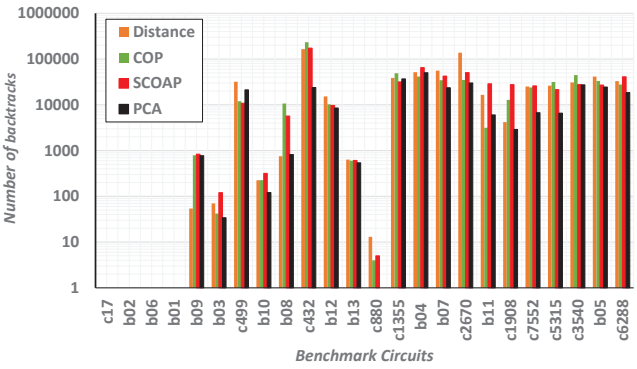

Fig. 4. Total backtracks for detecting all faults with ATPG using conventional heuristics and PCA. Four circuits on the left required no backtracks and c880 required no backtracks only when PCA was used.

allowed the extraction and use of domain-specific knowledge to solve computationally hard problems. VLSI design and test have benefited too: MI has been in use for analog, digital, and memory testing, along with emerging technology-based device test and hardware security [3]. A recent discovery of solving test generation problem using MI opened ample research avenues [11], [12]. Nevertheless, proposed MI techniques are supervised and need ANNs that must be trained and used in place of heuristics. The result is fewer erroneous backtraces (i.e., fewer backtracks) and more efficient ATPG. But, the increase in the volume of heuristic data may overburden the ANNs, making it more complex, leading to increased training time. This study used no ANN, and instead opted for a trivial statistical analysis technique, i.e., PCA, which can be used with the least additional cost to implement PODEM ATPG.

This PCA-based ATPG implementation outperforms conventional heuristics in terms of backtracks and CPU time. Circuit c880 (although small) shows the possibility of no backtracks using a linear combination of multiple features. Though previous work [9], [10] has reported that no single heuristic performs well in all cases, the present results show that the major PC combines multiple heuristics effectively and either outperforms or matches the best standalone heuristic on most circuits with few exceptions, as shown in Figures 3 and 4. Although performance improvements diminish for larger circuits, further improvement may be possible with more ANN features (reminiscent of the recent work [11]), such as, reconverging signal characteristics, fanout information, etc.

PCA-based ATPG is promising and also opens up new

avenues and future research directions. First, reconvergent fanout-free circuits do not have any backtracks, but backtraces in reconvergent fanout-free circuits can still be reduced and lead to more efficient tests requiring fewer PI assignments. Second, eliminating backtracks has a cost in CPU time; finding a 'sweet-spot' may be feasible where one can get the optimized number of backtracks for minimal CPU time. Third, one can detect redundant faults quickly to expedite ATPG using MI-guided ATPG. Fourth, MI was used in backtracing guidance of PODEM ATPG, but never used for D-Frontier drive selection in PODEM ATPG to witness more ATPG performance improvement. Fifth, using k-means clustering as the second technique of unsupervised learning and comparing it against PCA may give some interesting observations. Sixth, state-of-the-art ATPG tools find challenges in detecting some faults in a circuit either due to circuit size or atypical fault characteristics. Seventh, this study uses academic benchmark circuits instead of large industry-standard circuits; this study's authors believe that the trends of MI-guided ATPG performance are encouraging and likely to apply to larger circuits and show immense capabilities in future.

Finally, the efficacy can always be improved by maximizing the *explained variance*. The usual practice of PCA is to keep only the first $k < p$ principal components, where k is the dimension of transformed subspace that comprises of PCs and p is the dimension of the original space. PCA is an orthogonal transformation that projects data from a p-dimensional space to a k-dimensional subspace, and the remaining $p-k$ dimensions vanish in this kind of projection. It is rational to minimize variability in those $p-k$ directions and maximize the variance of the first k variables as the total variance of both p and k dimensional spaces is constant. Fig. 1 shows that the major PC (or the first PC) of some circuits do not reach close to 1, which signifies that there is room to maximize the *explained variance* of such circuits by adding more isomorphic features to the original dataset. In the future, one will make choices in PCA based on specific objectives: 1) PCA is a orthogonal transformation and will need maximum variance in the first k components and minimum variance in $p-k$ components; 2) choosing the first k components for maximum variability; and 3) choosing large k to reduce information loss and variance of $p-k$ components.

VI. Conclusion

For the first time, this study attempted to integrate unsupervised learning of MI with PODEM ATPG to demonstrate the effectiveness of ATPG in terms of reduction of backtracks and ATPG CPU time. This study used PCA to combine multiple features. A linear transformation projects conventional ATPG backtracing heuristics into a new major PC (the first PC), which is considered to be the carrier of maximum variance and replaces the traditional single-heuristic guidance of ATPG.

The conjecture about zero backtracks [11], [12] is supported once again by benchmark circuit c880 (although small from any standard), which gives a possibility (light at the end of the tunnel) of an ATPG with no backtracks. However, this claim may be too ambitious for large circuits with reconverging fanouts. For those cases, finding a 'sweet-spot' in the CPU time versus backtrack curve would be an acceptable choice with minimal ATPG CPU time despite backtracks.

References

[1] M. Pradhan and B. B. Bhattacharya, "A Survey of Digital Circuit Testing in the Light of Machine Learning," *WIREs Data Mining Knowl. Discov.*, pp. 1–18, 2020.

[2] H. Stratigopoulos, "Machine learning applications in IC testing," in *Proc. IEEE 23rd European Test Symposium (ETS)*, 2018, pp. 1–10.

[3] S. Roy, S. K. Millican, and V. D. Agrawal, "Special Session – Machine Learning in Test: A Survey of Analog, Digital, Memory, and RF Integrated Circuits," in *Proc. IEEE VLSI Test Symposium (VTS'21)*, 2021, pp. 1–10.

[4] S. Roy, B. Stiene, S. K. Millican, and V. D. Agrawal, "Improved Random Pattern Delay Fault Coverage Using Inversion Test Points," in *Proc. IEEE 28th North Atlantic Test Workshop (NATW)*, 2019, pp. 206–211.

[5] S. Roy, B. Stiene, S. K. Millican, and V. D. Agrawal, "Improved Pseudo-Random Fault Coverage Through Inversions: a Study on Test Point Architectures," *J. Electron. Test*, vol. 36, no. 1, p. 123–133, Feb. 2020.

[6] S. K. Millican, Y. Sun, S. Roy, and V. D. Agrawal, "Applying Neural Networks to Delay Fault Testing: Test Point Insertion and Random Circuit Training," in *Proc. IEEE 28th Asian Test Symposium (ATS)*, 2019, pp. 13–18.

[7] S. T. Chakradhar, V. D. Agrawal, and M. L. Bushnell, *Neural Models and Algorithms for Digital Testing*. Springer, 1991.

[8] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits," *IEEE Transactions on Computers*, vol. C-30, pp. 215–222, 1981.

[9] J. H. Patel and S. Patel, "What Heuristics are Best for PODEM?" in *Proc. First International Workshop on VLSI Design*, 1985, pp. 1–20.

[10] S. Patel and J. Patel, "Effectiveness of Heuristics Measures for Automatic Test Pattern Generation," in *Proceedings of the 23rd ACM/IEEE Design Automation Conference*, 1986, pp. 547–552.

[11] S. Roy, S. K. Millican, and V. D. Agrawal, "Machine Intelligence for Efficient Test Pattern Generation," in *Proceedings of the IEEE International Test Conference*, Washington D.C, Nov. 2020.

[12] S. Roy, S. K. Millican, and V. D. Agrawal, "Training Neural Network for Machine Intelligence in Automatic Test Pattern Generator," 2021, Proc. 34th International Conference on VLSI Design.

[13] K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[14] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology,*, vol. 24, no. 6, pp. 417–441, 1933.

[15] J. E. Jackson, *A User's Guide to Principal Components*. Wiley, 1991.

[16] I. Jolliffe, *Principal Component Analysis*. Springer-Verlag, 2002.

[17] K. Diamantaras and S. Kung, *Principal Component Neural Networks: Theory and Applications*. Wiley, 1996.

[18] B. Flury, *Common Principal Components and Related Models*. Wiley, 1988.

[19] J. Guo, G. James, E. Levina, G. Michailidis, and J. Zhu, "Principal Component Analysis With Sparse Fused Loadings," *Journal of Computational and Graphical Statistics*, vol. 19, no. 4, pp. 930–946, 2010.

[20] F. Brglez, "On Testability Analysis of Combinational Circuits," *Proc. International Symp. Circuits and Systems*, pp. 221–225, 1984.

[21] L. Goldstein, "Controllability/Observability Analysis of Digital Circuits," *IEEE Trans. on Circuits and Systems*, vol. 26, pp. 685–693, 1979.

[22] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Targeted Translator in FORTRAN," *Proceedings of the IEEE Int. Symposium on Circuits and Systems (ISCAS)*, pp. 677–692, June 1985.

[23] F. Corno, M. S. Reorda, and G. Squillero, "RT-Level ITC'99 Benchmarks and First ATPG Results," *IEEE Design & Test of Computers*, vol. 17, pp. 44–53, Jul. 2000.

[24] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Springer, 2000.