Machine Learning-Based Solution Predicting Hardware Faults

R Ganesan1*, T Prabahar Godwin James2, P Perumal3, and Pardeep Kumar4
1Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamil Nadu, India
2Department of Computer Science and Engineering, Sri Sairam Institute of Technology, Chennai, Tamil Nadu, India
3Department of Computer Science and Engineering, Sri Ramakrishna Engineering College, Coimbatore, Tamil Nadu, India
4Department of Artificial Intelligence, Anurag University, Hyderabad, Telangana, India
Email: rganesan1978@gmail.com1*, prabahar.cse@sairamit.edu.in2, perumalp@srec.ac.in3, pardeepsep2@gmail.com4

Abstract. Constant exploration of better performance and energy consumption has led to several hardware-based generators being used in interdependent SoC. Such generators abuse the intrinsic contrast of activities, and they also forgive inconsistencies in their outputs, such as image and optical machine learning applications. At about the same time owing to process growing and control constraints, irreversible faults are escalating, leading to incorrect outputs. Strategy to resolve this issue, which uses machine learning training tools to improve the impact of permanent fix in hardware accelerators that can withstand incorrect outputs. The suggested financial benefits do not need any accelerator details and are highly flexible with low capital areas. Furthermore, earlier work is very time intensive as it involves the introduction of fault pairs into the gate netlists to test their effect on the outputs. To solve these problems, this paper proposed to improve representation first by increasing the degree of complexity from the degree of both the gate to a stage of actions. The power to make segments and sub of the same contextual definition with specific features by establishing numerous propagation commands in the supply chain context involves. Test data demonstrate that our proposed approach is a simple and reliable way of generating different designs to secure the device.

Keywords: Predicting Hardware Faults, AI, ML, Deep learning, Soc.

1. Introduction
Two types of defects that usually exist in equipment are temporary and persistent. First one does no lasting harm to a connection, while the second causes the system to be permanently damaged or destroyed. In the production and lifespan dependability of computer chips (ICs), method optimisation has increasingly critical challenges. Future VLSI architecture, mostly due to the construction in system physics or the electronics industry, is likely related to a balance with handling product defects and cost. The error continues for the whole lifespan of the system in the event of irreversible defects. Therefore it is important to establish strategies that can account for all of these irreversible faults with minimum fixed costs to benefit from Moore's rule. The traditional solution is to thoroughly test greater test time and therefore expense for lasting faults. Faulty circuitry is usually either discarded or, in some situations, partly damaged after testing, dependent on the magnitude of the defects found [1-5].
If the number of full-time faults increases, there is an incredibly prominent expense associated with yield and output loss due to the rise in switching frequency and growing uncertainty. Thus at the design time, it makes perfect sense to solve this problem to create relief circuitry that can minimise the influence of rough defects on the ICs. These apps are intrinsically immune to inaccuracies or approximations and are thus tolerant of any lack of consistency or optimality in their measured performance. Significant generators in modern dynamic homogenous SoCs can perform these tasks more effectively by leveraging their accelerators’ intrinsic parallel processing. In this article, we suggest an answer to error correction that can account for circuitry of any complexity, such as those typically used in dedicated computer propulsion systems. This work specifically targets force users because they often accept numerical errors and because they are usually the key components that distinguish amongst various IC suppliers. By reflecting on the discrepancy in the flawed and perfect performance for CPU cores that can accept errors in their operations, we suggest a modular method for constructing error compensation modules. We conduct gate-level studies for various accelerators of various complexities and test various statistical methods of deep classification to construct the penalty logic.

2. Background Work

Significant work has been done on identifying and fixing hardware defects. Most of these techniques rely on the efficiency of a single circuit or system using functional and data replication. For test-related applications, parity trees have been used widely. While most prior research focuses on the identification of temporary faults, these article concentrates on trying to compensate and for impact mostly on output of the temporary weaknesses mostly on system. Expense, conventional circuit or complication technologies in terms of accessibility, energy and efficiency, like dual-module replication for excusing circuitry failures, are too costly; potentially boost poor validity solutions[6-9]. Computational noise sensitivity solutions have been introduced for Responsible for the distribution where time mistakes are required to occur, and a computational error detection framework is then changed. Mistakes are resolved using a collective error's frequency method to find the probability ratio for and output bit. A metric to calculate the sensitivity of code to guaranteed faults was developed by the authors of[10-17], and a mapping strategy was proposed to increasing the fault rate.

Comparison of the flawed circuit to a fault-free central solution. The authors of [18-21] also use supervised-learning dependent error mitigation to analyse the treatment of irreversible errors. However, the methodology focuses on simple reference techniques to forecast the entire performance. The downside to quite an approach would be that the number of features expected is strongly equal to the value of specimens, resulting in a high workload area for broader circuitry. Artificial intelligence algorithms’ precision falls as the list of predictor features grows[23-27]. The likelihood of a projected
sample resulting in a wrong boundary reduces. This article discusses an approach to developing compensate logic for dedicated computer generators capable of using classifiers for approximation computation. We will not need to produce bit-wise effectiveness of the developed process within each output, which brings more consistency to a technique than using quantitative or Causal inference approaches. We think that our methodology is exactly equivalent to earlier system.

3. Model Of Fault Recovery For Estimated Hardware Accelerators

Many recent researches are heavily dependent on implementation based on compensating for errors. With the amount of faulty outputs, our suggested solution is independent of service and scales so well. Also it is particularly strongly adapted as the cause of the fault depends solely on the parameters, not on the current state of the circuit or atmospheric conditions[22-26], because as amount of incorrect outcomes made observable only at acceleration inputs is comparatively small refers to the total number of test training samples. Our suggested system utilises the incorrect outputs as an advantage for fault repayment. The suggested reimbursement principle contains a tuneable module, rendering it modular to account for several recurrent faults. Although the adjustment principle may not fully produce the total error is mitigated, so it is especially suitable for embedded processors that can tolerate output errors.

Usually, the source nodes that house the fault have to be located to address a structural fault. The key issue is that all the thread can be identified for the existence and place of temporary defects. This assumes that even in the voltage applied, any remedy concept would need as many essential feedback as any harm to the body nodes. Even this is not realistic. The reimbursement rationale involves ties to approximately application domains from regional sources and regional outlets, such as those targeted in this work, as seen in Fig. 1, thereby decreasing and streamlining the chain's reach effective. Also, the SMURF device IOs are just the global current and voltage signals of the accelerator.

All of the dedicated hardware that is compliant with estimated computation easily create the existing based incentive architecture. Fig. 2 shows the SMURF system description, which is subdivided into two key components: the tuneable control tree framework and the control adder module. Two key modules are comprised of the tuneable tree this a multiplier block and a rotary encoder tree. There are special vectors in the founder block that result in false outputs. These lines are called Error Distance Values ED, and the next subsection explains them in detail. The tunable tree component is fixed and needs to be calibrated to allow multiple faults in each unique device's acceleration by selecting a range of parameters. This forest floor size is independent of the accelerator's difficulty and reflects only on the input data set.

The very last element is the n-bit circuit, with n being the number of compressor outputs. By applying a correction vector (EDSMU RF) to the defective value Frequency Defective, the circuit compensates the defective outcomes:

\[
\text{Output SMU RF} = \text{Output Faulty} + \text{EDSMU RF}
\]
Figure 2. hardware fault representation of SMURF

Compares favourably to the defective error lacking adjustment decreases the output error (ESMU RF) so that ESMU RF < E Faulty. This stage of output also increases directly proportional to the set of inputs. It thus leads to the usability of the logic of the challenge findings, an example shown in fig 3. Given the step towards homogeneous SoC systems with different hardware accelerators, another significant resource constraint for the reimbursement module collective part given the movement towards homogenous SoC systems featuring different hardware accelerators. The field’s cost will then be amortised by the distribution of the reward module between these accelerators. The next chapter explains how learning-based approaches are used to do this tuning, also summarised in methodology.

Figure 3. SMURF Tree
To test our suggested approach, we have selected several accelerators. To obtain their gate-netlists, these were synthesised using Yosys. Faults have been introduced into the seven benchmark circuits’ main entrance to represent lasting faults. This was tested using Modelismusing 10000 randomised parameters before and after the fault was added. The machine learning process is done using WEKA. For larger accelerators to produce reliable recovery models, a larger test set can be needed.

Throughout this research, we expand this task and offer a fast and effective predictor of mortality analysis to predict the diversity just after HLS among different sub so that no fault injecting becomes needed, thus dramatically accelerating the development of the scalable framework. It can be shown that by raising the level of abstraction from RTL to the behavioural level, a much greater range of locally accepted can be created as the pragmas allow entirely new microarchitectures to be produced.

4. Results
We equate distinct predictive approaches in this last series of experiments. In specific, vs Linear Regression and Rep Tree machine learning algorithms were previously described by the Hoeffding tree. The same benchmarks and the same defects as previously mentioned were used in both situations. We specifically note that there is a lower MAE value in the SMURF model. For Linear regression and RGB, the MAE values are similar to each other. However, one very interesting finding is that 90% of cases, the current Hoeffding tree model is reliably very precise across all parameters, indicating a substantially stable methodology. On the opposite, the matrix project resulted in incorrect results, just 90% of the cases, although the Rep Tree results in 87% on the comparison.

Table 1. Memory Usage

| Benchmarks | DCT | RGB | MAC |
|------------|-----|-----|-----|
| Memory/Area Overhead % | 6.4% | 3.9% | 3.4% |

Eventually, for different implementations of our suggested compensation logic technique, Table 1 displays the overhead field ratio. For most projects, the overhead areas are on average, 5%. Given the increase in error detection obtained, we assume this overlap is appropriate.

5. Conclusion
To minimize the burden of severe failures in cpu cores that accept the certain lot of mistakes by the use of classifiers, we developed a simple reduced grades due for permanent hardware failures. For many broad studies, designers also performed entrance testing, multi-stage circuits for various complex CPU cores and provided extensive research data, demonstrating significant standard errors. Our suggested approach is based on reimbursing the impact of faults on production by estimating the relative variance of production instead of forecasting the complete production practically. It has been shown that the suggested solution increases the exact outcome by 50 percent and decreases the current total failure rate by 90 percent with such a time complexity of 5 percent. The suggested solution has the extra benefit of treating the unit like a computer machine. The difficulty of a fast fix is not dependent on the size and scope of the task. It is just dependent on the number of variables which are independent and dependent. Our suggested approach is adjustable because the size does not decrease with the length and variety of the application, including its SMURF module.

References
[1] Ragmani, A., Elomri, A., Abghour, N., Moussaid, K., Rida, M., & Badidi, E. (2020). Adaptive fault-tolerant model for improving cloud computing performance using artificial neural network. *Procedia Computer Science, 170*, 929-934.
[2] Abdalla, A. S., & Marojevic, V. (2020). Machine Learning-Assisted UAV Operations with UTM: Requirements, Challenges, and Solutions. *arXiv preprint arXiv:2006.14544*. 
[3] Benacchio, T., Bonaventura, L., Altenbernd, M., Cantwell, C., Gillard, M., Giraud, L., ... & Wedi, N. (2020). Deliverable 1.3: Report on local data recovery approaches suitable for the weather and climate prediction (No. SAND2020-3622R). Sandia National Lab. (SNL-CA), Livermore, CA (United States).

[4] Moorthi, S. (2020). An Approach to Emulate and Validate the Effects of Single Event Upsets using the PREDICT FUTRE Hardware Integrated Framework. Defence Science Journal, 70(3).

[5] Wang, X., Jiang, L., & Chakrabarty, K. (2020, April). LSTM-based analysis of temporally- and spatially-correlated signatures for intermittent fault detection. In 2020 IEEE 38th VLSI Test Symposium (VTS) (pp. 1-6). IEEE.

[6] Pandey, P., Basu, P., Chakrabarty, K., & Roy, S. (2020). GreenTPU: Predictive design paradigm for improving timing error resilience of a near-threshold tensor processing unit. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 28(7), 1557-1566.

[7] Bender, E., Bernstein, J. B., & Bensoussan, A. (2020). Reliability prediction of FinFET FPGAs by MTOL. Microelectronics Reliability, 114, 113809.

[8] Kale, S. D., & Patil, S. C. (2021). Need for Predictive Data Analytics in Cold Chain Management. In Advances in VLSI and Embedded Systems (pp. 115-129). Springer, Singapore.

[9] District, G., & Prades, A. Self-learning and Efficient Health-status Analysis for a Core Router System.

[10] Ibrahim, Y., Wang, H., Liu, J., Wei, J., Chen, L., Rech, P., ... & Guo, G. (2020). Soft errors in DNN accelerators: A comprehensive review. Microelectronics Reliability, 115, 113969.

[11] Anik, M. T. H., Danger, J. L., Guilley, S., & Karimi, N. (2020). Detecting Failures and Attacks via Digital Sensors. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

[12] Vijayaragavan, V., ME, S. D., & ME, M. Y. Reduction of Resource Usage in FPGA by Implementing Back Propagation Algorithm.

[13] Manimegalai, C. T., Gauni, S., & Kalimuthu, K. (2017). Smart traffic management using a building integrated photovoltaic (BIPV) powered Li-Fi system on IoT. Electronic Government, an International Journal, 13(2), 168-181.

[14] Moradi, M., & Denil, J. (2020). Machine-learning assisted model-implemented fault injection.

[15] Manimegalai, C. T., Gauni, S., Kalimuthu, K., & Bashkaran, R. (2020). Design and analysis of turbo-coded DCM-OFDM ultra-wideband system with radio-over-fibre and wireless transmission in underwater communication. Journal of Optics, 49(1), 140-146.

[16] Gauni, S., Manimegalai, C. T., Kalimuthu, K., & Anjanasree, M. (2018). Detection of breathing and heart rate using a simple vital radio system. International Journal of Engineering & Technology, 7(2.8), 311-314.

[17] Manimegalai, C. T., Kalimuthu, K., Gauni, S., & Kumar, R. (2016). Enhanced Power Control Algorithm in Cognitive Radio for Multimedia Communication. Indian Journal of Science and Technology, 9, 25.

[18] Loknath, G., & Umapathy, K. (2016). 5G wireless mesh network 802.11 s load balancing architecture for 802.11 Bgn radio-PCI interface. Procedia Computer Science, 87, 252-257.

[19] Krishnamoorthy, R., & Umapathy, K. (2018, February). Design and Implementation of Microstrip Antenna For Energy Harvesting Charging Low Power Devices. In 2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB) (pp. 1-3). IEEE.

[20] Sivakumar, G. P., & Umapathy, K. (2017, February). Closed-loop fuzzy logic controlled class-F3 amplifier system with an improved dynamic response. In 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT) (pp. 1-4). IEEE.

[21] Raj, N. G., & Umapathy, K. (2016, February). A distributed fair channel using MAC for the 5G wireless network. In 2016 International Conference on Information Communication and Embedded Systems (ICICES) (pp. 1-5). IEEE.
[22] Umapathy, K., Balaji, V., Duraisamy, V., & Saravanakumar, S. S. (2015). PERFORMANCE OF WAVELET BASED MEDICAL IMAGE FUSION ON FPGA USING HIGH-LEVEL LANGUAGE C. *Jurnal Teknologi*, 76(12).

[23] Balaji, V., Umapathy, N., Duraisamy, V., Umapathy, K., Venkatesan, P., & Saravanakumar, S. (2015). ENHANCING VARYING OVERHEAD AD HOC ON-DEMAND DISTANCE VECTOR WITH ARTIFICIAL ANTS. *Jurnal Teknologi*, 77(28).

[24] Akillandeswari, A. (2014). A Multiplier-Less Lifting Scheme Based DWT Structure. *Computers and Software*, 203.

[25] Gauni, S., Manimegalai, C. T., Kalimuthu, K., Kaushik, V. C. S., & Rao, T. R. (2018). Optical channel analysis of turbo coded MIMO-OFDM system for visible light communication. In *Optical And Microwave Technologies* (pp. 161-174). Springer, Singapore.

[26] Uthayakumar, C., & Sarukesi, K. (2011, December). An adaptive e-Learning system using knowledge management. In *3rd International Conference on Trendz in Information Sciences & Computing (TISC2011)* (pp. 70-74). IEEE.

[27] Nagaraju, V., Bapu, B. R. T. & Janani, B. (2015). Design of Log Periodic T-Antenna for Operation in S-Band. *Indian Journal of Science and Technology*, 8, 98-102.