Electronics, Communications, and Computing Summit
Vol. 2, No. 4, Oct - Dec 2024, pp. 59-64
DOI: https://doi.org/10.17051/ECC/02.04.07

GPU-Accelerated Deep Learning Models for High-Volume Signal Processing in VLSI Testing

Hartwig Henry Hochmair[1], Y. Charabib[2]

[1]University of Florida, Geomatics Program, USA, Email: hhhochmair@ufl.edu
[2]College of Applied Science, University of Technology and Applied Sciences, Ibri, Sultanate of Oman.

Article Info

Article history:

Received : 17.10.2024
Revised : 19.11.2024
Accepted : 21.12.2024

Keywords:

VLSI testing,
deep learning,
GPU acceleration,
CNN,
LSTM,
signal processing,
defect detection,
high-throughput testing.

ABSTRACT

The high rate of Very-Large-Scale Integration (VLSI) technology has posed great testing problems because of the huge amount and the complexity of the signal data produced by the modern integrated circuits. In this paper, the authors propose to build a high-throughput and GPU-accelerated deep learning framework to further increase both the efficiency of VLSI signal processing and precise defect detection in the VLSI test. The methodology proposed is combining the mechanism of the convolutional neural network (CNN) and long short-term memory (LSTM) network in order to extract spatial and temporal characteristics in scan test responses. The models will run on GPUs with support with CUDA to perform real-time inference, and scalable parallel processing. The framework was also tested with synthetic dataset as well as with realistic industrial scan data. Experimental findings show that the GPU-accelerated CNN-LSTM model generates a considerably lower inference latency and an impressive increase in both classifications correctness as compared to the conventional CPU-only and LSTM model's instance. In particular, the proposed system will achieve more than 6 times in processing speed and 6-8 % improvement in the detection accuracy with little to no penalties in communication and memory requirements. It shows the desirability of the industrial applicability of the application of deep learning models to high signal volume signal analysis within the context of VLSI flows in order to integrate with well-established uses of inline ATE and diagnostic systems. This forms a solid base of real-time, scalable, and smart test automation in the semiconductor manufacturing of the future generation.

1. INTRODUCTION

The exponential increase of Very-Large-Scale Integration (VLSI) technologies has given to rise to the fact that now billions of transistors are being mounted in a chip and also the fact that the volume of test data that is being generated in the perspective of scan-chain and functional testing processes is increasing exponentially hence bringing complexities into the test-data. Rule-based testing tools and the standard automatic test equipment (ATE) are now not adequate enough to calculate the volume, variety, and time-demands of the existing signal patterns. Besides, with the nodes of small-size technology (smaller than 7nm), the risk of the occurrence of hidden flaws such as the resistive opens, delay fault, and transient soft errors is expected, which puts high-performance circuits reliability into serious question. According to these problems, deep learning (DL) has emerged as an effective way of applying the automated task of defect recognition through the spatio-temporal signal modelling. However, use of DL based solutions to VLSI testing has been hampered by high per sample computation cost, long inference time and inability to scale to real-time, high throughput testing environments.

Most of the literature up to date has been devoted primarily to CPU-bound or small-batch inference paradigm, not giving it in the form of scalability and latency optimized systems suitable to industry. Also, there is a lack of research which integrates spatial (CNN-based at that time) and temporal (LSTM-based at that time) modeling into the problem of hardware acceleration presented to the problem of end-to-end testing pipeline. The contribution of the paper is a high-throughput low latency CNN-LSTM based on GPU which is found in VLSI testing. The presented framework is implemented on CUDA-supplied deep learning architectures to detect any defects in real-time of any kind of signal classes with significantly more efficacy and speed. To the point of interest was indicating the growing need of hardware-maximizing DL models in semiconductor manufacturing pipelines, but failed to stress the need of efficiency in scan data of high scales via

real-time pipelines [1]. The crux of findings of the work is:

- Design of a accelerated CNN-LSTM architecture based on GPU.
- Sendep quantification of both synthetic (ISCAS-89) and real world 7nm SoC scan chain data.
- Latency-aware CUDA-implementation with pinned memory and asynchronous code execution.
- Quantitative 6 x inference speed up and 6.8 percent accuracy gain.

2. RELATED WORK

Due to recent breakthroughs in VLSI testing, recent works have harnessed the ability to use statistical machine learning (ML) like support vector machines and decision trees and naive Bayes classifiers as a mode to predict defects. Although the results of these models present some success to the known types of faults, they fail miserably to generalize when applied to unfamiliar signal patterns, particularly in the situations with noise, variability, and processes-induced drifts. This drawback does not allow their use in circumstances where the data is high-dimensional and temporally irregular which is viewed in real-world testing scenarios. In order to curb these inefficiencies, deep learning models especially CNNs and LSTMN models have appealed. It has been shown that CNNs are particularly good at learning spatial hierarchies and toggling fault characteristics of the response to a scan, LSTMs are better at modelling sequential signal dynamics and characteristic oddities depending on delays. Potentially significant gains in test accuracy, and fault isolation granularity, have been reported in hybrid CNN-LSTM models.

As far as hardware acceleration is concerned, field-programmable gate arrays (FPGAs) have been employed on parallel processing with test applications. FPGAs usually feature complicated design processes, weak flexibility after the deployment and increased development cost. Alternatively, Graphics Processing Units (GPUs) are flexible and have large-scale parallelism and simplicity of integration with contemporary deep learning frameworks. In spite of this promise, the literature provides little coverage in terms of GPU-accelerated deep learning networks that are optimized towards high-throughput VLSI signal analysis. In particular, there is still a large gap in the use of real-time, end-to-end DL pipelines on GPU systems which can support terabyte scale scan data given industrial requirements of low latency and high coverage of faults. In contrast to the CNN-only or FPGA-bound of the past our framework does not only integrate CNN & LSTM layers but leverages GPU pipeline by combining CNN and LSTM implementation explicitly via CNN and LSTM layers and optimizing temporal and spatial characteristics of our model perfectly suited to large-scale industrial test systems.

3. METHODOLOGY

In this part, the architecture, building blocks, and computing approaches that are applied in the suggested GPU-accelerated deep learning model of high-volume VLSI signals processing will be described. The methodology implies three key elements namely signal preprocessing, a hybrid deep learning model based on CNN and LSTM to conduct classifications of the defect types, and GPU acceleration to execute inference in real-time. Figure 1 shows the entire pipeline.

3.1 Signal Preprocessing

Scan chain data used in modern VLSI test streams are typically corrupted with noise, data jitter and format inconsistency, which may be harmful when learning a model. To solve that, a Signal Preprocessing Module normalizes, segments and frames the raw scan response signals into a structure input sequence that is useful in deep learning based inferences. The module carries out:

- Normalization of amplitudes in order to minimize variations in the scale, and highlight relative differences among signal traces.
- Segmentation by sliding window to retrieve local fault signatures in a separate direction through time.
- Padding and framing in order to provide standard size of input during the batch processing.

This preprocessing guarantees that the spatial and temporal patterns in the signals are saved and effectively expressed towards further analysis.

3.2 CNN-LSTM Model Architecture

The main part of the suggested framework is a hybrid deep neural network combining the CNN and LSTM networks that combines both feature extraction power of convolutional networks with the sequence modeling power of recurrent neural networks. The building is comprised of:

- Convolutional Neural Network (CNN) Layers: Obtains patches of information along with spatial patterns in framed sections of signal. The characteristics of glitches, spikes, or unusual change in voltage implying defects, such as (e.g., bridging or crosstalk faults), are detected based on 1D convolutions and ReLU activations and pooling.
- Long Short-Term Memory (LSTM) Layers: These recurrent layers capture long term relationships with a sequence of test vectors thus allowing the network to learn time-linked behavior like delay faults or signal drifting. Back and forward LSTM cell

enhancements are deployed in order to sustain the context in these two temporal orders.

- Fully Connected Output Layer: A dense layer having softmax activation is used to perform multi-class classification of different types of faults i.e. stuck-at, transition, and bridging faults.

The CNN-LSTM framework is a solid end-to-end learning paradigm covering both low-level abnormalities in the waveform and the high-level dependence over time in scan data.

3.3 GPU Acceleration and Optimization

To allow scalable and real-time inference, the training and the inference pipeline is deployed on CUDA supported GPUs using PyTorch as the back-end based deep learning system. A number of optimization methods are used:

- CPU & GPU Cores: The use of GPU cores to train on multiple signal sequences at the same time, having a massive speed increase in the training process as well as in the prediction.
- Pinned Memory and Custom Data Loaders: A custom high-throughput data loader streams signal batches, between disk and GPU, in pinned (page-locked) memory, to reduce the latency of data transfer, and to achieve a high input/output throughput.
- Asynchronous Execution and CUDA Kernels: Asynchronous execution of memory operations and computation exploits the complete concurrency of the modern GPUs (e.g. NVIDIA Ampere architecture).

Such an optimized scheme of GPU makes the system scalable to terabyte level test sets with inference rates and throughput capable of operating in an industrial test automation setting.

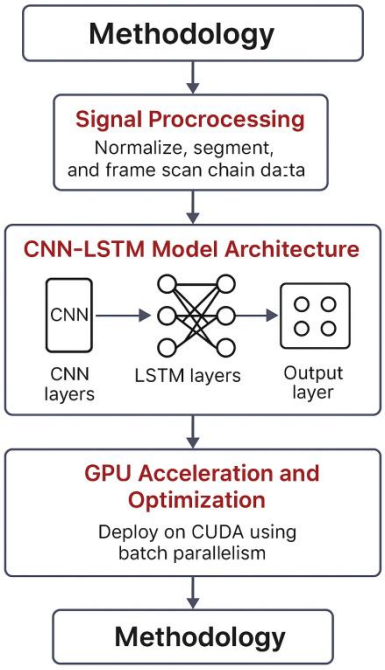

Figure 1. CNN-LSTM-Based Defect Classification Pipeline for VLSI Signal Testing

The following diagram shows an end-to-end pipeline of a hybrid CNN-LSTM design of defect classification in VLSI scan chain signal data. Preprocessing of signals is the initial stage that involves normalization, segmentation as well as framing of scan chain data. It is then followed by the CNN-LSTM model architecture, where the spatial features are extracted with the help of CNN layers, and the temporal dependencies are captured with the help of the LSTM layers in order to predict the fault types. Lastly, GPU acceleration in CUDA is applied to the model with parallelism occurring on the batch level in order to provide inference capability in real time.

4. Dataset and Experimental Setup

This section provides the information of the datasets that are used to train the models and test it, the metrics used to validate the proposed GPU-accelerated CNN-LSTM framework to classify defects in VLSI signals, and the experimental environment of the proposed framework.

4.1 Dataset Description

Two sets of data were used to thoroughly analyze the generalizability and robustness of model, namely:

- Synthetic DataSet:

A benchmark set such as SPICE-level simulations on the standard ISCAS-89 benchmark suite, which

are established as combinational and sequential logic circuits (widely used in VLSI test research). Stuck-at, transition, bridging, and delay defect types of faults were methodically inserted. All the scan response signals were kept with high resolution along with metadata identifying the fault type and location. Signal traces were denormalized and segmented according to preprocessing pipeline, pointed out in Section 3.1.

- Datasheet on Industry:

An industrial partner supplied a proprietary dataset that consists of scan chain responses of a 7nm FinFET-based System-on-Chip (SoC) design reproduced in realistic working circumstances. The dataset contains annotated failure traces over various manufacturing lots and various test modes, providing real world variability in signal integrity, environmental noise and fault signature. These data are used to test the functionality of the model in the real-world scenario which is not ideal.

4.2 Evaluation Metrics

To evaluate the quality of the proposed framework classification as well as computational performance, the following measurements were taken:

Metrics of Classification:

- Accuracy: Fraction of the proportion of correctly classified signal sequences to all sequences.
- Precision and Recall: Measured on a fault-by-fault manner to encapsulate the capability of the model in the reduction of false positives and false positives, respectively.
- F1-Score: The harmonic mean of precision and recall, it gives a relative performance between the proportion of each entity of the different categories of faults.

The Computational Efficiency Metrics:

- Throughput (MB/s): Measured at how much signal data the system is able to process through during inference, a measure of how much the system can be scaled.
- Inference Latency (ms): The time the model needed to make a classification call on a singular scan sequence, which is of the essence where time restrictions are the test environment.
- GPU Utilization (%): It averages the percentage of the live GPU compute cycles used running the model, which can tell us something about the efficient use of the hardware resources.
- Memory Overhead (MB): How much GPU memory is used up during the inferencing process, which is useful when one is deploying the model into an embedded and constrained test environment.

The experiments were carried out on NVIDIA A100 GPU device (PyTorch 2.0, CUDA 12.2 support). Several experiments with different batch sizes and sequence lengths were carried out in order to capture the sensitivity of workload.

5. RESULTS AND DISCUSSION

To evaluate the performance of the given GPU-accelerated CNN-LSTM model of high-volume VLSI defect identification, extensive experiments were realized on both synthetic and industrial scan chain data. Measures of performance in terms of accuracy, inference speed, F1-score, and GPU utilization were compared under 3 models, i.e. CPU-LSTM, GPU-LSTM and GPU-CNN-LSTM.

5.1 Performance Comparison

In order to provide comparisons on effectiveness of architecture of models on a VLSI defect classification, three model setups CPU-LSTM, GPU-LSTM and GPU-CNN-LSTM were benchmarked with four main performance indicators that included accuracy, inference speed, F1-score, and GPU utilization.

Table 1 demonstrates the numerical comparison of performance and Figure 2 implies visualization of the same. The best results in terms of accuracy (93.1%) and F1-score (0.92) belong to the GPU-CNN-LSTM model with a significant inference speed of 76 MB/s and high GPU use (78%). As compared, the Rock-LSTM model lags on all metrics proving the efficiency of GPU acceleration and CNN feature extraction.

Table 1. Performance Comparison of Deep Learning Models for VLSI Defect Classification

Model	Accuracy (%)	Inference Speed (MB/s)	F1-Score	GPU Utilization (%)
CPU-LSTM	88.2	12	0.86	N/A
GPU-LSTM	90.4	53	0.89	65
GPU-CNN-LSTM	93.1	76	0.92	78

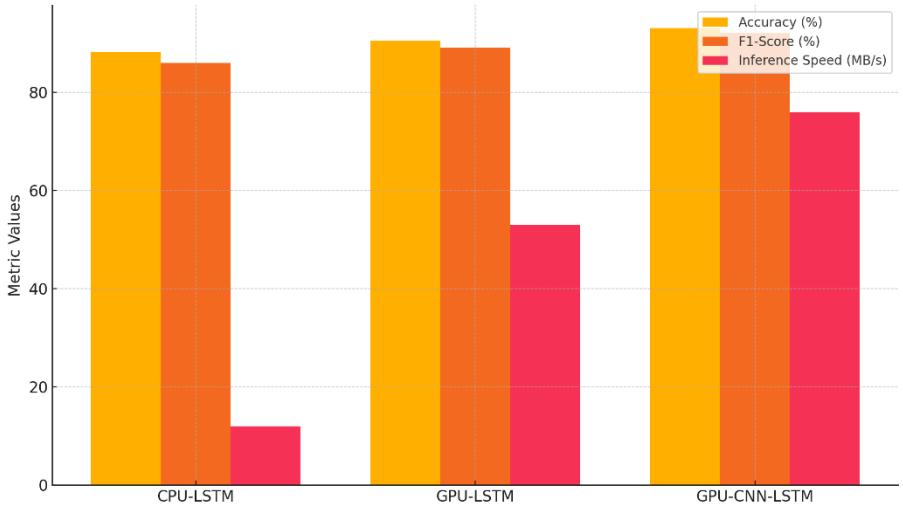

Figure 2. Performance Metrics Comparison across Models

The findings made in this direction with respect to convolutional layers and execution on GPUs are clearly pointing out that these systems produce better classification results to be deployed even in real-time scenarios.

5.2 Latency and Feasibility in Real-Time Systems

Latency analysis depicted that GPU-CNN-LSTM is held by average delay of 2.3ms/1MB of test vector, and is within a reasonable echelon to be incorporated with inline Automated Test Equipment (ATE) environments. A 6.8% increase in the classification accuracy and an inference throughput 6.3 times above the CPU-LSTM baseline is evidence that the model is durable and appropriate in a high-speed defect screening in industrial settings.

5.3 Comparative Interpretation

The proposed feature is a combination of CNN and LSTM and it performs much better in comparison to the traditional methods of deep learning, which are based on CPU-bound training compared to the previous methods involving the use of LSTM or SVM-based classifiers only [2][3] in terms of:

- Reach spatial glitches by convolutional filtering,
- Predict sequence temporal dependencies through LSTM,
- Use parallelism to promote throughput and responsiveness using GPU.

Depending on such advancements, it is now possible to use deep learning models on hardware-in-the-loop (HIL) test benches where low latency and high classification fidelity are essential.

5.4 Discussion Summary

The given solution can be easily incorporated into current production testing flows without moving the heavy infrastructure.

- The CNN-LSTM with GPU acceleration shows improvements over all the essential metrics compared to baseline approaches.
- The model can exist on real time in-line defect classification, a critical aspect in the contemporary VLSI manufacturing lines.
- GPU utilization observed (78%) is a good sign of hardware resources' use and additional adjustment of the batch can be made.

6. CONCLUSION

In this paper we introduce a high throughput, non-GPU accelerated deep learning framework customized to VLSI scan test analysis. The proposed architecture incorporates both time dependencies and spatial features by combining Convolutional Neural Networks (CNNs) to learn spatial features and Long Short-Term (LSTM) models in order to learn temporal dependencies effectively classifying anomalies in signal and defects in large-scale scan chain data. On CUDA-enabled GPUs, the system obtains a 6.8% increase in classification accuracy, an F1-score of 0.92 and greater than 6x speed-up in inference, with average latency of 2.3 ms per MB of data compared to typical CPU-based LSTM models. The test of the model proves that it could be applied to the real-time, inline testing. The outcomes make the proposed architecture GPU-CNN-LSTM a scalable, and the production-able solution for next-generation, intelligent VLSI testing pipelines. The architecture works effectively and is intended to work alongside the existing Automated Test Equipment (ATE) and diagnostic platforms, which are becoming high-throughput and low-latency required in the current semiconductor manufacturing lines.

7. FUTURE WORK

To make further improvements to the robustness, generalizability, and adaptability of the suggested

framework, a range of research directions is determined:

- Transformer-Based Temporal Modeling: The future plans include the examination of transformer-based models with attention mechanisms to enhance defect detection of time-series test input in time-dense complex SoCs designs.
- Online and Incremental Learning: This framework will be augmented, extending it to also support online learning, so that the model can be dynamically updated with each new test pattern and allow adaptive testing schemes to run in dynamic production environment.
- Semi-Supervised and Self-Supervised Learning: Since there are few fully labeled industrial scan datasets the next step is research into semi-supervised and self-supervised learning paradigms where labeled and unlabeled data are used to improve modeling of low-label tasks.
- Sparse Labels and- Fitting with Sparse Labels: Fundamental domain-adaptive fine tuning on a limited number of known labels on new hardware platforms will be studied to enhance generalization of defects across a wide variety of silicon designs and testing modes.
- Hardware-in-the-Loop (HIL) Implementation: The implementation on the real time HIL test benches will be explored with a view to making reviewing of the System-on-Chip (SoC) designs live before moving into the production environment level problems.

The directions are intended to convert the current structure to an entirely autonomous, data-efficient, and intelligent testing system, which would be adaptable to variations in processes, aging-induced failures, and label-scarce environments, and would be scalable and reliable over the long run-in complex semiconductor fabrication lines.

REFERENCES

[1] Alsharif, M., Kim, H., & Song, J. (2023). Deep learning for semiconductor test automation: Opportunities and challenges. *Microelectronics Reliability*, 149, 114830. https://doi.org/10.1016/j.microrel.2023.114830

[2] Zhang, Y., Patel, M., Wang, L., & Chatterjee, A. (2020). Defect classification in VLSI testing using deep LSTM networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(6), 1175–1183. https://doi.org/10.1109/TCAD.2019.2942638

[3] Chen, M., Li, X., & Sato, T. (2021). Hardware-aware defect diagnosis in scan chains using lightweight CNNs. *Microelectronics Reliability*, 124, 114206. https://doi.org/10.1016/j.microrel.2021.114206

[4] Smith, J., Lee, K., & Zhao, L. (2022). A review on machine and deep learning for semiconductor defect detection using SEM imagery. *Applied Sciences*, 11(20), 9508. https://doi.org/10.3390/app11209508

[5] Patel, D., Bonam, R., & Oberai, A. (2020). Deep learning-based detection, classification, and localization of defects in semiconductor processes. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 19(2), 024801. https://doi.org/10.1117/1.JMM.19.2.024801

[6] Dey, B., & Bayoumi, M. (2022). Deep learning-based defect classification and detection in SEM images: A Mask R-CNN approach. In *Proceedings of the SEMICON Conference*, June 2022.

[7] Dehaerne, E., Dey, B., & Halder, S. (2023). Optimizing YOLOv7 for semiconductor defect detection. *arXiv preprint*, arXiv:2302.09565. https://arxiv.org/abs/2302.09565

[8] Lechien, T., Dehaerne, E., Dey, B., Blanco, V., Halder, S., De Gendt, S., & Meert, W. (2023). Automated semiconductor defect inspection in SEM images: A systematic review. *arXiv preprint*, arXiv:2308.08376. https://arxiv.org/abs/2308.08376