Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

Cluster-based test vector re-ordering for reduced power dissipation in digital circuits

M. Navin Kumar, S. Sophia & B. Nivedetha

To cite this article: M. Navin Kumar, S. Sophia & B. Nivedetha (2023) Cluster-based test vector re-ordering for reduced power dissipation in digital circuits, Automatika, 64:4, 1141-1147, DOI: 10.1080/00051144.2023.2251230

To link to this article: https://doi.org/10.1080/00051144.2023.2251230

Published online: 29 Aug 2023.

AUTOMATIKA
2023, VOL. 64, NO. 4, 1141–1147
https://doi.org/10.1080/00051144.2023.2251230

Cluster-based test vector re-ordering for reduced power dissipation in digital circuits

M. Navin Kumar[a], S. Sophia[b] and B. Nivedetha[c]

[a]Dept. of ECE, Sri Krishna College of Technology, Coimbatore, India; [b]Dept. of ECE, Sri Krishna College of Engineering and Technology, Coimbatore, India; [c]Dept. of EEE, PSG College of Technology, Coimbatore, India

ABSTRACT

Optimizing testing power is a paramount concern in modern digital circuit design, particularly as the intricacy of circuits continues to rise. This issue becomes even more pronounced with the scaling down of feature sizes due to advancements in process technology. The increase in testing power dissipation, beyond the regular operational state, raises red flags for both designers and test engineers. This paper introduces a novel cluster-based approach, shedding light on the efficient reordering of test vectors to mitigate the heightened switching activity. This technique aims to reduce the power consumption during testing, while still ensuring efficient error detection, maintaining the test period, and preserving the original order of the scan chain. Importantly, this reordering method does not introduce any additional area or test time overhead, minimizing the risks associated with these factors. The potential impact of this approach is demonstrated through concrete examples drawn from ISCAS'89 benchmark circuits. The results showcase a notable 11% reduction in switching activity, all while preserving fault coverage levels.

ARTICLE HISTORY
Received 23 June 2023
Accepted 18 August 2023

KEYWORDS
Switching activity; low power testing; cluster based reordering

Introduction

In the dynamic realm of modern VLSI circuits, the complexity surge presents an array of challenges, with power dissipation during testing topping the list. The phenomenon of amplified switching activity during VLSI circuit testing, compared to regular operations, sets the stage for a critical conundrum. As capacitive loads charge and discharge, power dissipation balloons, with the magnitude of activity determine the extent of the increase. Moreover, the testing phase introduces another concern: abrupt power surges that can potentially disrupt the testing process and jeopardize the integrity of the circuit under test (CUT). Excessive power dissipation, particularly during the capture cycle of the scan chain, raises a binary dilemma – under-testing or over-testing – both undermining the quality of outcomes leading to yield loss [1]. The necessity of skilfully managing power dissipation during CUT testing is thus undeniable. In the realm of modern digital circuit design, the optimization of testing power has become pivotal, especially as the scale of circuits continues to escalate. This study addresses the challenge of reducing power consumption during testing without compromising error rates and the integrity of the scan chain order.

To tackle the issue of managing test power, various methods are projected in the existing works. The more switching action in circuit logic elements is primarily caused by the repeated toggling of scan cells [2], triggered by input test vectors and their related scan-out responses. This activity is mainly attributed to consecutive complementary bits within the scan cells. As the activity propagates via the combinational circuit, there is a significant increase in dynamic power dissipation. One approach to mitigate this issue is to reorder the scan cells. However, this results in considerable increase of routing hardware and computation time, particularly as the complexity of the CUT increases [3].

Another approach to minimize test power is by inserting inverters linking scan cells, which helps in minimizing switching movement throughout the scan shift operation. However, this introduces additional area and cost overhead over other technique for reducing test power is modifying the scanning method by incorporating extra logic gates. On the other hand, the scan chain can be divided into numerous sub-chains. Although this approach can be effective, it requires extra complex control circuit with additional I/O pins. Examining every bit of the configuration information requires additional test clock cycles. Furthermore, test vectors reorder themselves to minimize switching activity during test mode without altering the scan configuration. However, this technique is susceptible to increased computation time due to the growing complexity of the CUT [15–18].

Scan chains work by incorporating additional scan flip-flops into the circuit design. These scan flip-flops are connected in a chain, with each flip-flop connected

CONTACT M. Navin Kumar navinkumar.m@skct.edu.in Department of Electronics and Communication Engineering, Sri Krishna College of Technology, Coimbatore, Tamilnadu, India

to the next using a scan enable signal. During normal operation, the scan enable signal is deactivated, and the scan flip-flops pass the data through without any interference. However, during the testing process, the scan enable signal is activated, causing the scan flip-flops to hold and observe the current state of the circuit. This allows each flip-flop in the chain to be examined individually, making it easier to detect any manufacturing defects or errors in the combinational logic. By incorporating scan chains diagnosis principle into the design, engineers can easily apply test patterns and observe the behaviour and output of each flip-flop in the digital circuit. This simplifies the testing process and helps to identify any faults that may be present. Scan chain reordering is an important optimization technique that involves rearranging the order of flip-flops in the scan chain based on their physical placement in the design.

This paper presents a novel approach to determine the overall switching movement, which specifically correlates with the actual power dissipation of the circuit. To achieve a maximum defect coverage value for the circuit being tested, ATPG tools like ATALANTA are employed to generate test vectors. These test vectors are then reordered using Kruskal's algorithm. However, as the complexity of the circuit increases, computing the order becomes more challenging with Kruskal's algorithm. To address this, the entire set of test vectors is divided into smaller clusters, enabling easier reordering of the test vectors within each cluster. To get the final test vector order, the clusters rearrange themselves via the similar process. This study presents a novel approach utilizing Kruskal's algorithm to reorder test vectors at the cluster level. The primary contribution lies in achieving substantial reduction in power consumption during testing. By focusing on the rearrangement of test vectors, this approach ensures that the critical balance between power reduction and accurate fault detection is maintained.

The proposed approach offers practitioners the flexibility to navigate the trade-off between computation time and test power savings. By adjusting the number of sub-clusters, the study empowers users to tailor their strategy based on project-specific time constraints and power-saving objectives. This dynamic adaptability contributes to the practicality of the methodology and its potential integration into real-world scenarios.

In summary, the proposed work suggests a novel method to reorder the test vectors at the cluster level, effectively reducing test power while keeping the scan chain architecture unchanged. This approach maintains high error detection, test period, and the order of the scan chain, eliminating the risk of area overhead or test time overhead.

MST in Kruskal's algorithm

The minimum spanning tree (MST) size is the number of edges in the tree that connects all vertices of a graph with the minimum possible total edge weight. In the context of Kruskal's algorithm, the MST size is equal to the number of vertices minus one. For instance, circuit "s5378" has 257 test vectors. Therefore, the MST size for this circuit would be 257.

Kruskal's algorithm avoids cycles by employing the concept of a disjoint set data structure, often implemented using techniques like Union-Find. This data structure keeps track of connected components within the graph. When Kruskal's algorithm considers an edge, it checks if the vertices of that edge belong to different connected components. If they do, the edge is added to the MST. If they belong to the same connected component, adding the edge would create a cycle, which is undesirable. Hence, the algorithm avoids adding such edges to prevent cycles in the MST.

Here's a high-level overview of how Kruskal's algorithm is implemented:

Initialization: Sort all the edges of the graph in non-decreasing order of their weights.

Disjoint Set Initialization: Create a disjoint set for each vertex. Each vertex initially belongs to its own set.

Main Loop: Iterate through the sorted edges. For each edge:

Check if adding the edge would create a cycle. This is done by checking if the two vertices of the edge belong to the same set in the disjoint set data structure. If the vertices are not in the same set, add the edge to the MST. This means joining the sets of the two vertices by performing a union operation in the disjoint set data structure.

Termination: Continue this process until you have added (V – 1) edges to the MST, where V is the number of vertices.

Output: The final set of edges in the MST represents the optimal sequence or arrangement based on the algorithm's criteria (such as minimal edge weights). In this proposed work, we have multiple circuits with multiple input nodes and logic gates which have been analysed with the MST in Kruskal's algorithm to further enhance the accuracy of the system.

Power consumption measurement using switching activity

Power consumption in a circuit is predominantly influenced by the switching activity of the scan cells, as it involves the discharging and charging of capacitive loads. Consequently, as the fan-out increases, power dissipation also escalates ultimately. The switching activity can be computed using Hamming distance

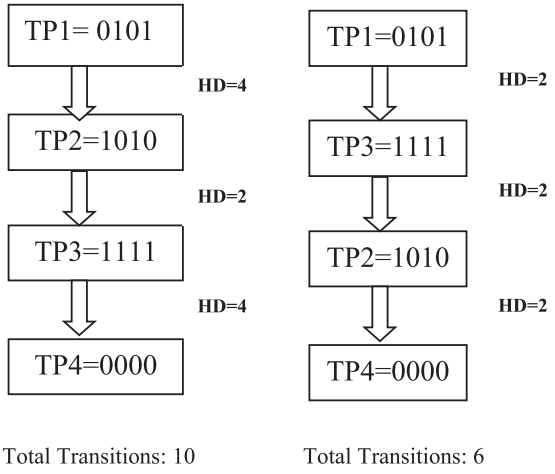

Figure 1. Hamming distance calculation.

as shown in Figure 1, total transition is reduced to 6 from 10 by suitably reordering it [6–8].

Figure 1 demonstrates the reduction in switching transitions achieved by reordering the test patterns. By optimizing the order of test patterns, the Hamming distances are minimized, resulting in fewer transitions during testing. This leads to a reduction in power consumption, as fewer transitions mean less charging and discharging of capacitive loads, which is a significant contributor to power dissipation in digital circuits.

In static CMOS gates, dynamic switching power typically surpasses leakage power and crossbar power by a considerable margin. Therefore, by multiplying the switching activity at a specific node of the circuit under test (CUT) by its corresponding fan-out, a more accurate estimate of the total power dissipation can be obtained [9,10].

The primary objective to mitigate the adverse impact of switching activity during testing, is intricately linked to power consumption. The proposed approach is meticulously designed to yield a twofold benefit – lowering power dissipation while concurrently upholding the standard of test quality. This research contribution holds substantial implications within the larger picture of energy and power modelling. By providing a tangible and innovative technique for optimizing power consumption during the testing phase, this research addresses a critical concern in the field of digital circuit design. This not only underscores the relevance of this research but also positions it as a pragmatic solution that holds potential to substantially enhance overall circuit efficiency and performance. The incorporation of cluster-based reordering with Kruskal's algorithm, showcases the focused and concrete approach of this research to effectively minimizing power consumption during testing, thus supporting the context of energy and power modelling.

Various methods for test vector reordering

Various techniques can be used to rearrange test vectors based on a function that quantifies the overall switching activity in a given set of test patterns.

Particle Swarm Optimization (PSO) is a metaheuristic algorithm that uses a population of particles to search for optimal solutions. PSO has been used effectively for a variety of optimization problems, including test vector reordering. In test vector reordering, the goal is to find an order for the test vectors that minimizes the number of test cases required to achieve a desired level of coverage. PSO can be used to find effective reordering solutions by representing each test vector as a particle and moving the particles through the search space according to the algorithm. However, PSO can be slow to converge for large problems. This is because the search space for test vector reordering is very large, with m! possible orderings for m test vectors [11].

Genetic Algorithm-Simulated Annealing method uses a combination of Genetic algorithm-simulated annealing approach to minimize switching movement by reducing the Hamming distance between successive test patterns and improves device reliability. The advantages of the approach include a significant reduction in switching activity and improved performance and convergence rate due to the use of simulated annealing. However, the further tuning of parameters are required [5].

In TSP based Reordering method, the reordering in test vector sequencing is approached using Travelling Salesman Problem (TSP). Here each test vector is treated as a city, and the entire switching activity required for conversion between test vectors is considered as the distance or cost of travel between cities. This analogy allows us to create a graph with test vectors as vertices and switching activity as edge weights [4]. To construct the graph, calculate the switching activity between each pair of test vectors generated by ATALANTA. This information is used to form a completely connected undirected graph, where each vertex represents a test vector, and the edges represent the transitions between them. The edge labels indicate the switching activity cost in both directions [12–14].

Kruskal's algorithm, a greedy algorithm can be used in the graph to find the minimum spanning tree (MST) which represents the optimal sequence of test vectors, The MST is a spanning tree that covers all the vertices with the minimum possible total cost.

It is crucial to note that solving the TSP is known as an NP-hard problem, meaning it is computationally challenging to determine the exact optimal solution in a reasonable time frame. Hence, specialized approaches such as heuristics or greedy methods are mostly employed to approximate the optimal solution

efficiently [24,25]. Scan chain masking for diagnosis has been analysed by Kundu et. al in.

By utilizing the TSP-based reordering method, find an ordering of test vectors that reduces the overall test power consumption. This optimization can significantly improve the efficiency and effectiveness of testing procedures, ultimately enhancing the performance and reliability of the tested circuit.

Utilizing Kruskal's algorithm for test vector reordering

Kruskal's algorithm is a greedy approach to constructing a minimum spanning tree in a weighted graph. It selects the smallest edge that connects two different components at each step, aiming to minimize the total weight of edges. While it doesn't always guarantee the optimal solution, it's widely used and efficient in practice. This can be effectively utilized in the context of reordering test patterns to determine the optimal sequence. Kruskal's algorithm uses the disjoint-set data structure to detect cycles and maintain connected components. It requires sorting edges by weight, but time complexity can be optimized with the disjoint-set operations.

The algorithm begins by selecting a root vertex (r) as the starting point for constructing the MST. It connects isolated vertices one by one to form a tree, known as Tree A. The next vertex is chosen from the ones connected to r based on the minimum cost required to reach it, and these selected vertices are added to Tree A.

The remaining set of vertices (V) that are not yet part of Tree A is then evaluated. For each isolated vertex that can be directly reached from Tree A, the associated cost is examined. The vertices with the lowest cost are chosen and incorporated into Tree A. This process continues until the set V becomes empty, resulting in the completion of the MST, referred to as Tree A.

This approach ensures that the test vectors are sequenced in a manner that minimizes the overall cost or distance between them, contributing to improved efficiency and effectiveness in testing procedures.

Algorithm:

(1) Sort the edges of G in non-decreasing order based on their switching activity.
(2) Initialize an empty list to store the MST edges.
(3) Create a disjoint set data structure to keep track of connected components.
(4) Select an initial vertex r as the root vertex.
(5) Add r to the MST.
(6) For each vertex v in G. vertices, excluding r:
    (a) Add v to the MST.
    (b) Connect v to r with the edge (r, v) having the minimum switching activity.
(7) Mark all vertices in G as not yet included in the MST.
(8) While there are vertices not yet included in the MST:
    (a) Select the vertex u with the minimum switching activity to be added to the MST.
    (b) Connect u to the MST with the edge (u, v) having the minimum switching activity, where v is a vertex already included in the MST.
(9) The resulting list of MST edges represents the optimal order for reordering the test vectors.

Test vector reordering at the cluster level

In large circuits, a substantial amount of test vectors are often needed to achieve satisfactory fault coverage. However, attempting to reorder a large amount of test patterns by devising a graph can lead to computational infeasibility and high memory requirements. Additionally, selecting an appropriate root node becomes a challenge when dealing with a high amount of test patterns or vertices in the graph. To address these issues, it is necessary to limit the test set size [19–21].

Consider a circuit with N test vectors, which we treat as a single cluster called Cluster-0. Instead of applying reordering directly to Cluster-0, divide it into h same-sized sub-cluster and use the similar mechanism of reordering to each unit of sub-cluster. Every sub-cluster now contains N/h test vectors. By doing this, the complexity of finding an approximate Minimum Spanning Tree (MST) within a cluster of size N is reduced to finding an MST within cluster of size N/h.

After reordering every sub-cluster using the Kruskal's algorithm, the sub-clusters among themselves are rearranged using the same approach. This process yields a final order for both the sub-cluster and the test patterns in each sub-cluster. By combining this reordered sub-cluster, we obtain the completely ordered set of test patterns or vectors. K means clustering approach is preferred over hierarchical clustering because of its computational efficiency and convergence [22,23].

For instance, let's consider an unordered cluster of test vectors, Cluster-0, obtained from ATALANTA, as depicted in Figure 2. By applying the Kruskal's algorithm, we can achieve a Minimum Spanning Tree (MST) for the test vector set, resulting in an ordered set of test vectors stored in Cluster-0.

Alternatively, we can divide Cluster-0 into four unordered sub-Clusters: Cluster-01, Cluster-02, Cluster-03, and Cluster-04. Applying the Kruskal's algorithm to each sub-cluster will produce MSTs within each Cluster, yielding ordered sub-cluster: Cluster-03, Cluster-02, Cluster-04, and Cluster-01. However, we cannot simply concatenate these sub- clusters. The order of the sub-clusters must be determined based on the minimum cost of transitioning between them.

To establish the inter-cluster level reordering, apply Kruskal's algorithm once more amongst the sub-clusters. This process helps to find the optimal order

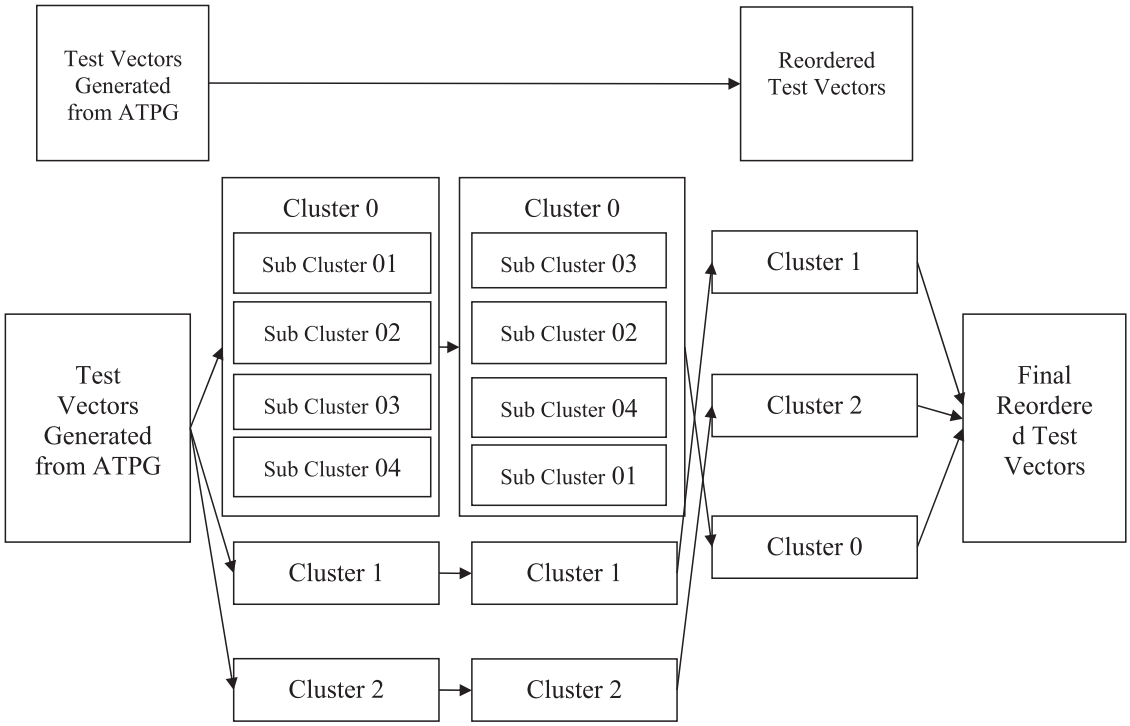

Figure 2. Cluster based reordering.

of the sub-clusters. Next, combine the sub-clusters according to the obtained orders, considering both the order among the sub-clusters and the order of the test patterns within each sub-cluster. This procedure yields the final ordered clusters, Cluster-1. By adopting cluster-level reordering, amount of the test set is limited, resulting in savings in rearranging time, computational complexity, and memory. However, it's important to note that the overall solution quality may be compromised.

Experimental findings

To evaluate the effectiveness of the proposed reordering technique, experiments were conducted using ISCAS'89 standard circuits. It was considered as full scan chains circuits were employed. Different reordering methods were applied to these circuits, and the total switching activity was measured. The results presented in Table 1 demonstrate that reordering based on the Kruskal's algorithm yielded the lowest switching activity.

As an illustrative example, let's consider the circuit s5378, comprising 214 input nodes and 2794 logic gates. In the pursuit of achieving comprehensive fault coverage, a total of 257 distinct test patterns were meticulously generated using the ATALANTA Automatic Test Pattern Generation (ATPG) tool. Reordering these patterns with the Kruskal's algorithm resulted in an overall switching activity of 48,764,768, which are lesser compared to previous approaches.

Therefore, Kruskal's algorithm outperformed PSO and Genetic algorithm based reordering techniques. To

Table 1. Comparison of switching activity using various methods of TVR.

Circuits	Number of test vectors	Switching activity from various reordering techniques			
		ATALANTA	PSO	GA-SA	Proposed
s526	65	194052	191224	189626	188542
s838	101	296485	294559	291032	290364
s953	94	426750	423091	417840	416784
s1196	140	1618159	1599437	1575380	1573432
s1488	127	741411	724116	695202	694386
s5378	257	49319413	49265211	48959214	48764768
s9234	379	78456678	76384756	73674588	72754346
s13207	480	87654382	85425842	83749326	81438274
s15850	437	85463826	83837492	81684624	80473684
s38584	653	298673142	276843268	264754826	252346382

mitigate the computational burden, sub-cluster level reordering was employed. The findings in Table 2 indicate that when the real test pattern set was divided into more sub-clusters (upto 4 clusters were tested), there was a slight degradation in the reordering solution with an increasing number of Sub-clusters. However, as shown in Table 2, the computation time significantly decreased by employing Sub-clusters. For instance, the s5378 circuit exhibited a 2.21% reduction in switching activity when treated as a single block. When divided into 4 Sub-clusters., the reduction was 1.84%. However, the computation time for a single cluster decreased from 53,116 s to 37,432 s for 4 Sub-clusters (Figure 3).

These experimental results validate the effectiveness of the proposed reordering technique in reducing switching activity and highlight the trade-off between solution quality and computation time (Figure 4).

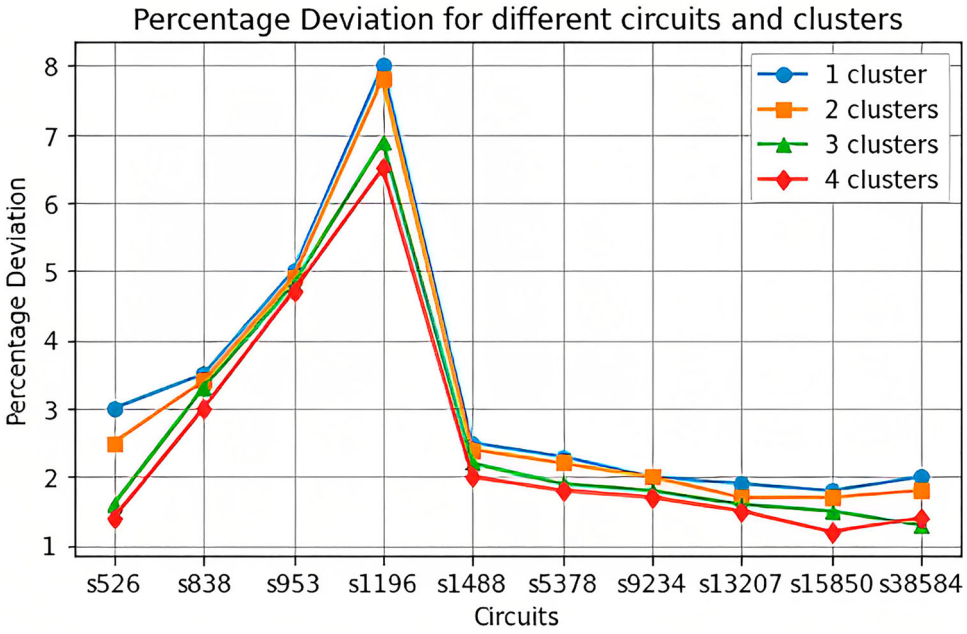

Figure 3. Deviation in switching activity with different values of clusters.

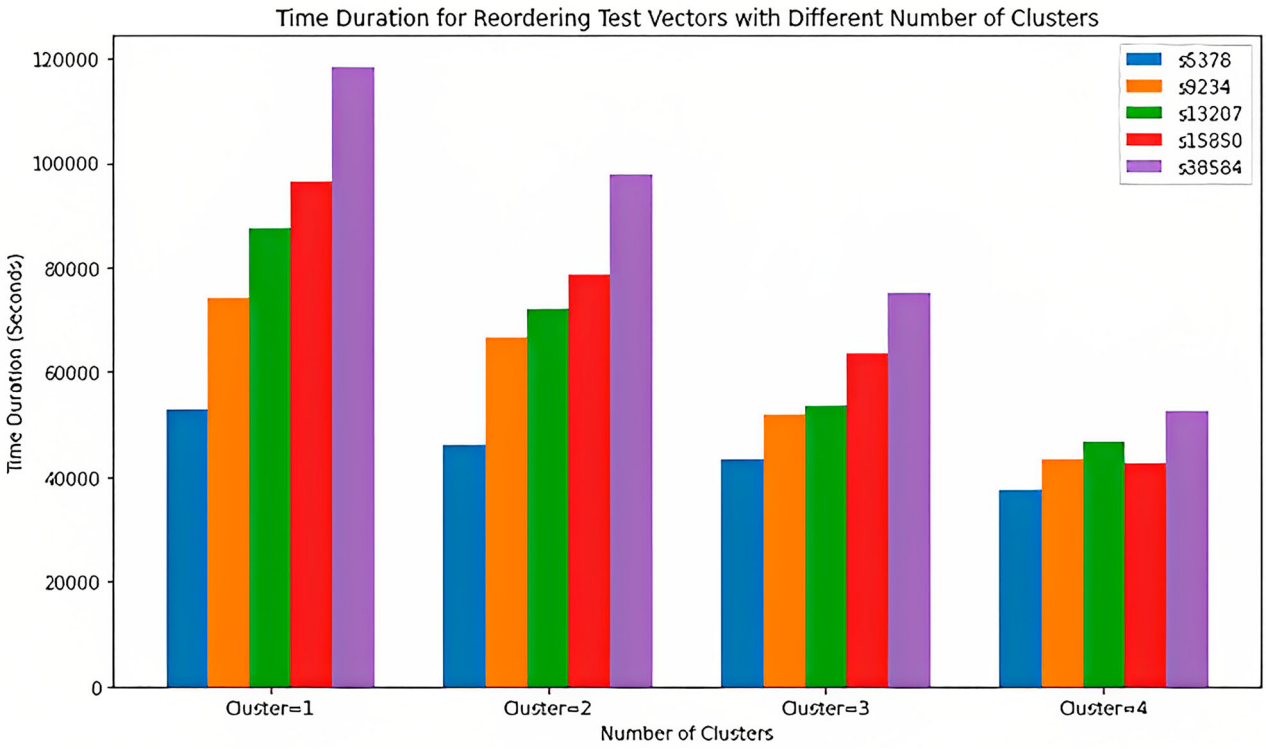

Figure 4. Switching activity computation time for various clusters.

Conclusion

This research presents a novel approach employing Kruskal's algorithm to reorder test vectors at the cluster level. The proposed methodology effectively reduces power consumption during the testing process while preserving the scan chain's original order and maintaining error rates. Comparative analysis against similar techniques like PSO and Genetic algorithms demonstrates the superiority of the proposed approach, showcasing an impressive 11% reduction in switching activity. Considering time constraints and the objective of minimizing test power, the technique offers flexibility through the manipulation of sub-clusters. Increasing sub-clusters benefits scenarios with tighter time constraints, whereas fewer sub-clusters provide greater power savings when time resources are more abundant. This adaptability allows users to tailor their strategy to the specific trade-offs between computation time

Table 2. Computational times of switching activity with varying clusters.

	Time duration for reordering test vectors (in seconds) with no. of clusters			
Circuits	Cluster = 1	Cluster = 2	Cluster = 3	Cluster = 4
s526	174	89	51	40
s838	170	82	49	34
s953	342	201	121	90
s1196	821	409	358	271
s1488	832	411	369	278
s5378	53116	46286	43458	37432
s9234	74282	66824	51937	43268
s13207	87696	72258	53598	46854
s15850	96432	78638	63639	42873
s38584	118458	97954	75349	52563

and test power savings, yielding substantial advancements in test power reduction while optimizing available resources.

Looking ahead, the application of Kruskal's algorithm in this context holds promising potential for further refinement and expansion. Future work could delve into exploring hybrid approaches that combine Kruskal's algorithm with other optimization techniques, amplifying the benefits of both. Additionally, investigating the scalability of the proposed methodology to larger and more intricate circuits will contribute to its robustness and applicability in real-world scenarios, ultimately advancing the field of power-efficient testing in modern digital circuits.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Singh B, Narang SB, Khosla A. Test power optimization with reordering of genetic test vectors for VLSI circuits. Acta Tech Nap Electr Telecommun. 2012;53:2.

[2] Almukhaizim S, AlQuraishi E, Sinanoglu O. Test power reduction via deterministic alignment of stimulus and response bits. J Low Power Electr. 2011;7(4):573–584. doi:10.1166/jolpe.2011.1152

[3] Maity H, Khatua K, Chattopadhyay S, et al. A new test vector reordering technique for low power combinational circuit testing. In 2020 International Symposium on Devices, Circuits and Systems (ISDCS). IEEE, 2020. p. 1–6.

[4] Song T, Liang H, Huang Z, et al. Chip test pattern reordering method using adaptive test to reduce cost for testing of ICs. IEICE Electron Express. 2021;18:2.

[5] Reddy KV, Bharath MV, Suhag AK, et al. Test vector reordering by using hybrid genetic algorithm-simulated annealing for lower switching activity. In 2018 4th International Conference on Computing Communication and Automation (ICCCA). IEEE, 2018, p. 1–6.

[6] Paramasivam K, Gunavathi K, Arulmozhivarman A. Functional distance based test vector reordering for low power testing of VLSI circuits. J Adv Res Dyn Contr Syst. 2017;9(16):1126–1137.

[7] Zhang M, Kuang J, Huang J. Double hamming distance-based 2d reordering method for scan-in powerreduction and test pattern compression. Electr Lett. 2020;56 (7):352–354. doi:10.1049/el.2019.3225

[8] Bushnell ML, Agrawal VD. Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits. Springer; 2000.

[9] Girard P, Landrault C, Pravossoudovitch S, et al. Reducing power consumption during test application by test vector ordering. In: Proc. of International Symposium on Circuits and Systems. 1998. p. 296–299.

[10] Girard P. Survey of low-power testing of VLSI circuits. IEEE Design Test. 2002;19(3):82–92. doi:10.1109/MDT.2002.1003802

[11] Paramasivam K. Reordering algorithm for minimization of test power in VLSI circuits. Eng Lett. 2007;14: 78–83.

[12] Dabholkar V, Chakravarty S, Pomeranz I, et al. Techniques for reducing power dissipation during test application in full scan circuits. IEEE Trans CAD. 1998;17(12):1325–1333. doi:10.1109/43.736572

[13] Roy S, Gupta IS, Pal A. Artificial intelligence approach to test vector reordering for dynamic power reduction during VLSI testing. TENCON, IEEE Region 10 Conference. November 2008.

[14] Basker P, Arulmurugan A. Survey of low power testing of VLSI circuits. Sci J Circ Syst Signal Process. Feb. 2013;2(2):67–74.

[15] Sudha AM. Low power testing of VLSI circuits using test vector reordering. Proc Int J Electr Energy. 2014;2:284–288. doi:10.12720/ijoee.2.4.284-288

[16] Nicolici N, Wen X. Embedded tutorial on low power test. Proc.12th IEEE European Test Symp., p. 202–210, 2007. doi:10.1109/ETS.2007.22.

[17] Wang J, Shao J, Li Y, et al. Using ant colony optimization for test vector reordering. In Proc. Of IEEE symposium on Industrial Electronics and Applications, 2009, p. 52–55.

[18] Hashempour H, Lombardi F. Evaluation of heuristic techniques for test vector ordering. In Proc. of the GLSVLSI, p. 96–99, 2004. doi:10.1145/988952.988976.

[19] Jadeja S, Mathuria P, Popat J. A review on low power testing techniques. Proc of International Journal of Advanced Networking Applications. 2015: 70–72.

[20] Pomeranz I. Balancing the numbers of detected faults for improved test set quality. IEEE Trans Comput Aided Des Integr Circ Syst. 2016;35(2):337–341. doi:10.1109/TCAD.2015.2460463

[21] Moghaddam EK, Rajski J, Kassab M, et al. At-speed scan test with low switching activity. In Proc. VLSI Test Symp., 2010, p. 177-182.

[22] Mehta U, Dasgupta K, Devashrayee N. Survey of test data compression techniques emphasizing code based scheme. Proceeding of IEEE 12th Euromicro Conference on Digital System Design (DSD09). 2009.

[23] Nivedetha B, Vennila I. Ffbks: fuzzy fingerprint biometric key based security schema for wireless sensor networks. Comput Commun. 2020;150:94–102. doi:10.1016/j.comcom.2019.11.007

[24] Wang L-T, Wu C-W, Wen X. Design for testabilityVLSI test principles and architectures. San Francisco: Morgan Kaufmann; 2006. p. 37–103.

[25] Ravi S, Chickermane V, Chakravadhanula K. Testing low power integrated circuits: challenges, solutions, and industry practices. 27th International Conference on VLSI Design and 13th International Conference on Embedded Systems, Jan.2014, p. 5–6.