A Review of Structural Testing Methods for ASIC based AI Accelerators

Umair Saeed[*†], Irfan Ali Tunio[*†], Majid Hussain[††], Fayaz Ahmed Memon[†††], Ayaz Ahmed Hoshu[†] and Ghulam Hussain[†]

umair.saeed25@quest.edu.pk, engr.irfan.ali@quest.edu.pk, majidhussain@quest.edu.pk engr_fayaz@quest.edu.pk, ayaz.soomro@quest.edu.pk, engr.ghulam.hussain@quest.edu.pk

[†]Electronic Engineering Department, Quaid-e-Awam University of Engineering Science & Technology,
Campus Larkana, 77150, Pakistan

[††]Electronic Engineering Department, Quaid-e-Awam University of Engineering Science & Technology,
Nawabshah, 64750, Pakistan

[†††]Computer Systems Engineering, Quaid-e-Awam University of Engineering Science & Technology,
Nawabshah, 64750, Pakistan

Abstract

Implementing conventional DFT solution for arrays of DNN accelerators having large number of processing elements (PEs), without considering architectural characteristics of PEs may incur overwhelming test overheads. Recent DFT based techniques have utilized the homogeneity and dataflow of arrays at PE-level and Core-level for obtaining reduction in; test pattern volume, test time, test power and ATPG runtime. This paper reviews these contemporary test solutions for ASIC based DNN accelerators. Mainly, the proposed test architectures, pattern application method with their objectives are reviewed. It is observed that exploitation of architectural characteristic such as homogeneity and dataflow of PEs/ arrays results in reduced test overheads.

Keywords:

Accelerator, Artificial Intelligence, Design-for-Testability, DNN, testing.

1. Introduction

Recently, it has been reported that AI based electronic devices will see about 18 percent annual growth over the next few years. And AI based electronic devices may generate around $67 billion in revenue by the year 2025 with a 20% share of the total market [1]. Recent advancements in classification accuracy of Deep Neural Networks (DNNs) in various domains i.e., speech recognition [2], image recognition [3] and sound recognition has allowed integration in various application platforms such as datacenters [4], automotives [5] and other edge applications etc. Specifically, these real-time applications execute inference operations, where already trained weights are used. Implementing this inference operation involves loading trained weights inside layers, where activation and trained weights are multiplied to generate weighted sums [6]. These weighted sums are then accumulated in an Accumulator unit. A non-linear function like ReLu is performed over this accumulated sum response [7]. This matrix-multiplication based inference operations is physically embedded on hardware accelerators. These hardware accelerators consist of array of 1000s of identical PEs to perform multiply-and-accumulate (MAC) operations over input activations and trained weights. The advancement of AI applications is coupled with improvement in these accelerator architectures.

These accelerators may be implemented on application specific integrated circuits (ASICs) [8], field programmable gate arrays (FPGAs) [9], or graphics processing unit (GPUs) [10]. Mainstream tech companies such as Google Inc., Tesla Inc., NVIDIA Inc., Graphcore, Enflame etc. have designed their custom DNN accelerators for optimized performance for their respective application domains [4,11,12,13,14]. Because of the reduced energy consumption (with less frequent memory access) and lesser data bandwidth, most ASIC based accelerators use spatial dataflow among interconnected PEs. This results in efficiency of throughput and energy consumption. For this reason, academia and industry have focused on developing specialized architectures of accelerators with focus on increasing throughput and reducing energy consumption [15,16,17,18].

Structural testing is an essential phase during manufacturing of any electronic systems. And with built-in testing method, periodical testing during infield operation can be performed to ensure functional and structural integrity of the system. Previously, it was shown that because of their learning algorithms, deep learning-based operations are inherently error resilient [19,20,[21]. But recent work has signified the impact of permanent faults i.e., stuck-at faults on the inference classification accuracy. It was shown that stuck-at faults in 0.005%

Manuscript received January 5, 2023
Manuscript revised January 20, 2023
https://doi.org/10.22937/IJCSNS.2023.23.1.14

faulty PEs /MACs degrades the classification accuracy from 74.13% to 39.69% [22]. Also, it is demonstrated in another study that 0.0003% faults in an accelerator drops the classification accuracy from 97.4% to 7.75% [23]. The reason for this drastic drop in accuracy is because of permanent faults can affect the higher order bits of th e weighted sum of the MAC, which increases the error for generated weighted sum [22,[23]. Such observations substantiate the need for structural test solutions for reliable production and operation of DNN accelerators.

Full scan-based EDT [24], Hierarchical ATPG [25], LBIST [26,[27] are primary off-the-shelf test solutions for testing of permanent faults. Incorporating these conventional test solutions can expedite time-to-market constraint for rapidly growing AI industry, specifically for ASIC based DNN accelerators. However, implementing such solutions on array level (with 1000s of PEs) can be inefficient in terms of test data volume, test time, test power and area overhead [28]. If not efficiently addressed, these test constraints can severely affect the overall manufacturing cost of AI accelerator. And if an accelerator is to be tested infield, then the BIST test power consumption can determine the package cost (for cooling mechanism) of the accelerator. Hence, the testing solution must be a cost-effective solution. In this paper, we first present conventional test solution based on state-of-the-art. Then novel test solutions are presented, these test solutions exploit the homogeneity/ regularity and dataflow of interconnected PEs of the array. The resulting novel test architectures are efficient in terms of test data volume, test time and test power. The rest of the paper is organized as follows.

- Section 2 briefly presents ASIC based DNN accelerators for which the test solutions were developed.
- Section 3 presents Core-level testability solutions based on state-of-the-art EDT, Hierarchical ATPG and LBIST.
- Section 4 presents techniques based on PE-level testability.
- Section 5 summarizes the paper.

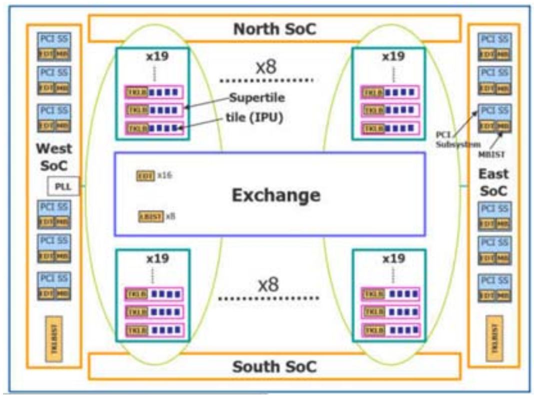


Fig. 2 Graphcore's AI SoC [29]

2. ASIC based DNN Accelerators

In this section, we will highlight the architectural characteristics of DNN accelerators for which the DFT solutions were implemented.

2.1 Graphcore's AI chip

The shows the architecture of Graphcore's AI chip[29]. Where a 'tile' is the basic processing unit of this chip. This tile contains an *intelligence processing unit* (IPU) and associated memory units. Due to high density of the chip, multiple tiles are grouped together to form a *Supertile*. It's a massive SoC, where an interconnect unit called the *Exchange* facilitates the communication among these Supertiles.

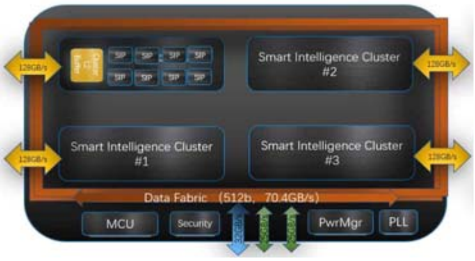

128GB/s
SIP
Smart Intelligence Cluster #2
128GB/s
128GB/s
Smart Intelligence Cluster #1
Smart Intelligence Cluster #3
128GB/s
Data Fabric (512b, 70.4GB/s)
MCU
Security
PwrMgr
PLL

Fig. 1 Enflame's DTU [30]

2.2 Enflame's DTU

Fig. 2 shows *Enflame's deep-thinking unit* (DTU), This DTU is also used as *convolution neural network* (CNN) inference acceleration [30]. It is specifically designed for Datacenters. It contains 4 *Smart Intelligence Clusters* (SICs). Where, an SIC contains 8 identical processing units called *Smart Intelligent Processors* (SIPs). Each SIP has 8 identical DSP cores along with other cores.

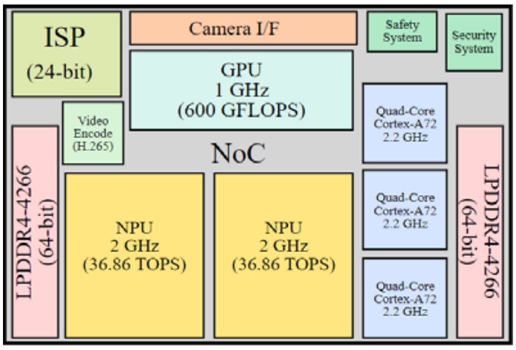


Fig. 1 Tesla's FSD chip [11]

2.3 Tesla's NPU based FSD

Tesla Inc. have custom designed their own *full self-driving* (FSD) chip for their autonomous vehicles [11]. It includes 2 neural processing units (NPU), shown in Fig. 3. This NPU is a 96x96 MAC array. The MAC array is used for inference acceleration for CNNs. A single MAC array contains 9216 PEs. Each PE shares connection with activation and weight memory to load the activation inputs and trained weights respectively. The PEs are interconnected to allow transfer of weighted sum output in cascaded manner via each column of PEs.

2.4 Google's TPU

Google Inc. was the first one to implement its own custom accelerator for inference application at their Datacenters [4]. This tensor processing unit (TPU) contains a MAC array of 256x256 systolically interconnected 65536 PEs, shown in Fig. 4. The architecture of the MAC array is weight-stationary systolic architecture [31]. It is specifically designed for CNN inference which utilizes trained weights reusability. The trained weights are loaded in systolic manner from the weight memory. After the array is loaded with trained wights, operational cycles are used to load the activation inputs and to generate and shift the weighted sums through PEs. In this systolic accelerator, a permanent fault manifested in the Datapath inside a PE can propagate systolically to other PEs. This will exacerbate the classification accuracy by affecting other PEs and their results to a significant level [23]. As it will be shown in the next section, most of the DFT based test solutions are proposed for this TPU and mainly exploit the systolic dataflow for application of test patterns.

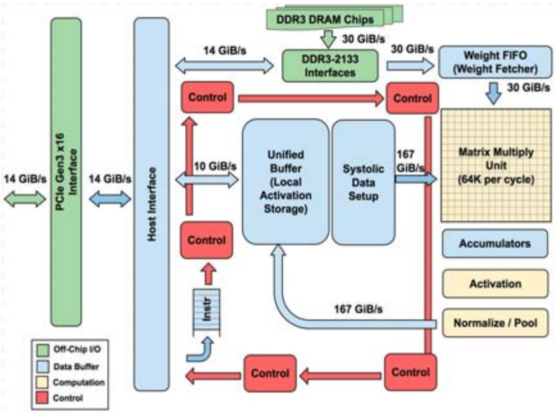


Fig. 2. Google's weight stationary systolic TPU [4]

3. Core-level test solutions

The DNN accelerators present homogeneity at different levels i.e., Core (group of many PEs) and PE. In this section, case studies will be presented, which address the testability at Core-levels. These Core-level solutions provide ease of testing by utilizing state-of-the-art full scan based EDT, wrapper logic and Hierarchical ATPG.

3.1 Graphcore's Hierarchical ATPG based solution

A case study for testing solution is presented for Graphcore's DNN accelerators in [29]. With a huge number of transistors i.e., 24 billion, the objective of the DFT insertion was set for faster time-to-market with less test power, test pins and scan data volume. To achieve this, Hierarchical ATPG was enabled for a single supertile and the ATPG is repeated for remaining tiles. This reduces the test data volume and the overall test time. As shown in Fig. 5, the DFT was inserted at supertile level with logic BIST and memory BIST, which was reused 304 times for the entire chip (for 304 Superiles).

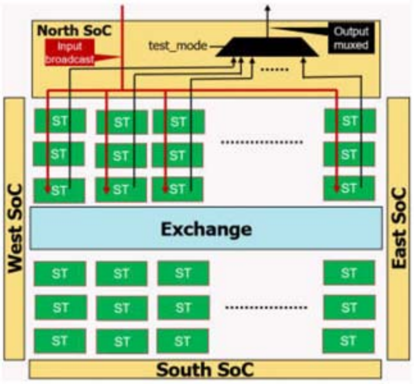
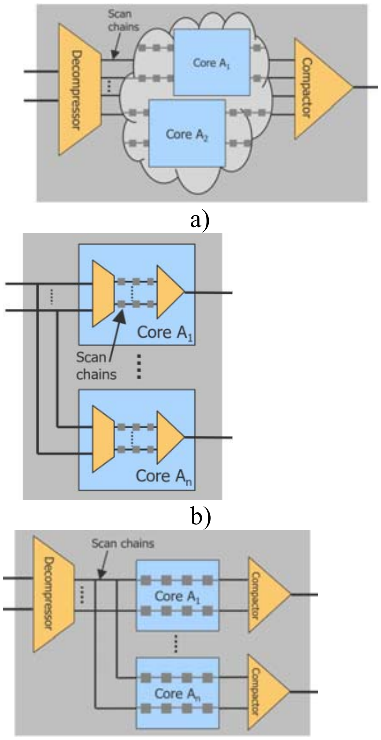


Fig. 6. DFT logic for enabling scan out from ST [29]

Using BIST modules with only seeds and signatures storage memory requirement, allowed reduction in test data volume. Majority of faults for supertiles were tested with LBIST whereas the remaining faults were tested with EDT (to ensure higher fault coverage). Moreover, to reduce the test pins at input, channel broadcasting was used for identical cores for application of test patterns. For reducing the output pin requirement, test output is done with MUX having non-overlapping modes, this is shown in Fig. 6. The test power was controlled with insertion of low toggle rate patterns in BIST and with clock gating.

3.2 Enflame's EDT based Solution.

A similar approach is observed for Enflame's 14 billion transistor chip [30]. To find an optimal test solution, different methods are explored as shown in Fig. 7. In the first method, the homogeneity of cores is not used for ATPG generation and DFT insertion. And the whole module is subjected to test pattern generation with EDT, shown in Fig. 7(a). This black-box approach results in lesser routing congestion but higher number of patterns, with repeated pattern generation for same faults (in identical cores). In the second approach the identical nature of cores is used for EDT logic and ATPG generation. The patterns are applied with simultaneous broadcasting of the same test patterns to input channels of identical cores, shown in Fig. 7(b). Compared with the previous approach this approach results in lower ATPG pattern count and runtime. Since the faults in cores are same then same scan patterns can be applied to each core. This approach is used to reduce the EDT logic overhead, shown in Fig. 7(c). But due to internal broadcasting, the routing congestion is high in this case.



a)

b)

c)

Fig. 7. Three possible configurations for EDT implemenation [30]

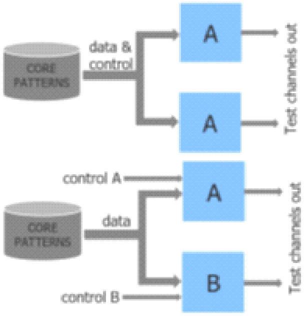


Fig. 8. Pattern broadcasting for identical and non-identical cores [30]

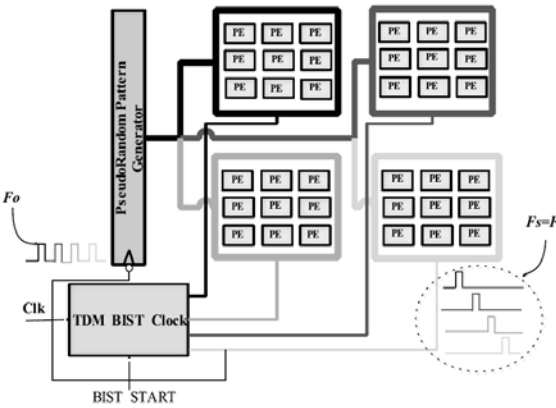


Fig. 9. Time multiplexed pattern loading into multiple sub arrays [32]

EDT is used at core level for test data compression. To alleviate the issue of limited test pins, pattern sharing is used. This can be done for identical and non-identical cores. The test patterns are broadcasted simultaneously to identical cores, number of such cores is limited by the shift power budget. Fig. 8 shows the broadcasting configuration for cores with isolation wrappers. This configuration results in reduced test data and short ATPG runtime. While other configurations like scan chain sharing increases routing overhead and congestion and applying EDT to non-shared cores can increase ATPG runtime and test data volume.

3.3 LBIST based test solution

Automotive electronics is a rapidly expanding domain for DNN accelerators for application such as autonomous driving. For such automotive DNN accelerators, an infield test solution such as LBIST is necessary to commence periodical testing and ensure reliable operation as per ISO 26262 [32]. With its built-in LFSR based pseudorandom pattern generator and MISR based signature analysis, the LBIST provides ease of integration into automotive hardware as a safety mechanism [27]. However, the pseudorandom nature of LBIST scan patterns suffer from some drawbacks, such as low fault coverage, increased shift power and test time and aliasing error.

To overcome these LBIST related issues, a time division multiplexing based LBIST is presented with concurrent error checking circuitry for DNN accelerator array testing, shown in Fig. 9. The array module is sub divided into homogenous smaller sub arrays, where arrays are clocked with time-multiplexed clock cycles during scan operation. An array partitioning algorithm is used, with peak power as the main constraint. Since patterns are applied to a smaller array, a proportional reduction in the number of test patterns, test time and peak shift power is achieved. Also, the proposed method uses combinational logic for concurrent error checking, which is free from aliasing problem. Compared to MISR, where error is detected at the end of the shift-out operation, this method provides much faster detection of faults in case of mismatch i.e. presence of faults.

3.4 Summary of Core-level testing approach

Using state-of-the-art like LBIST, Hierarchical ATPG and EDT allows for faster DFT implementation. Moreover, these case studies have also exploited the regularity/ homogeneity of identical cores to reduce the test data volume and lesser ATPG run time. The former feature will require less ATE memory requirement allowing more units to be tested simultaneously. Whereas the latter factor impacts the time-to-market and consequently the cost of the accelerator. However, this DFT approach needs additional overheads like wrapper logic, EDT controller logic, clock control logic. This additional logic may be a smaller percentage of a massive chip such as Graphcore's and Enflame's but this excessive DFT logic overhead may not be an efficient solution for edge-based DNN accelerators with their smaller size and power footprint. Moreover, it is observed that classification accuracy is largely affected by stuck-at faults of the MAC circuitry inside a PE [22],[23]. So high fault coverage at PE-level is desirable for comprehensive testing because using LBIST on core-level may not ensure high coverage at PE-level. And with this shortcoming, some critical faults can escape and may deteriorate the accuracy by affecting the high order bits of MAC circuit of a PE.

4. PE-level testability based solutions

The above test solutions have exploited the homogeneity on Core-levels with multiple identical PEs inside each core. However, the smallest replicable unit is the PE and testability can be confined to these PEs, which can further optimize the ATPG effort and reduce the test overheads. C-testing is a class of array testing method, where constant number of functional patterns are used to detect single faulty cell in the array of identical elements [33]-[39]. In the context of DNN accelerator with thousands of identical PEs, constant number of structural test pattern generated for one PE/ MAC can be applied to whole array. This will reduce the testability effort i.e., ATPG effort, test time and test data volume to a single PE/ MAC. In this section, such PE-level based testability techniques are presented.

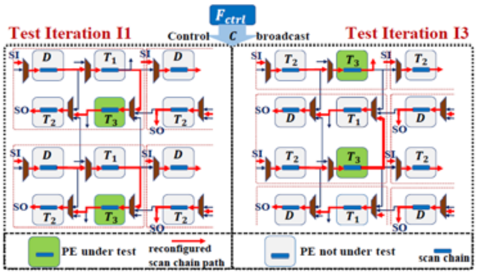


Fig. 10. DFT logic for enabling checkerboard style testing [40].

4.1 Checkerboard-style C-testing

The concept of structural C-testing has been realized for TPU's systolic array by generating structural test patterns for smallest replicable unit i.e., PE [40]. This approach minimizes the ATPG effort and culminates wrapper and EDT logic overhead. As the ATPG generates the scan patterns for a single PE, the solution is scalable for any array size. The PEs of the array are not usually wrapped by flops. So, this technique uses flops of the adjacent PEs (in addition to flops of the tested PE) for application of test pattern and to capture test responses. With reconfigurable scan architecture, the method uses the systolic dataflow for application of test patterns. The scan chains are inserted at the PE-level with a one scan chain per PE. Additional MUX logic is used to allow the iterations based scan-in and scan-out operations as shown in Fig. 10. This enables scan pattern applications from four adjacent PEs with four iterations in checker-board style. So, alternate PEs and in rows and columns are tested simultaneously in each iteration (shown in Fig. 11).

4.2 Partial scan based testing

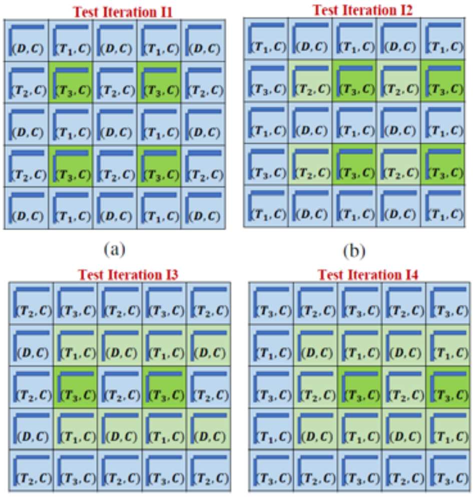

Test Iteration I1
(D,C) (T_1,C) (D,C) (T_1,C) (D,C)
(T_2,C) (T_3,C) (T_2,C) (T_3,C) (T_2,C)
Test Iteration I2
(T_1,C) (D,C) (T_1,C) (D,C) (T_1,C)
(T_3,C) (T_2,C) (T_3,C) (T_2,C) (T_3,C)
(a)
(b)
Test Iteration I3
(T_2,C) (T_3,C) (T_2,C) (T_3,C) (T_2,C)
(D,C) (T_1,C) (D,C) (T_1,C) (D,C)
Test Iteration I4
(T_3,C) (T_2,C) (T_3,C) (T_2,C) (T_3,C)
(T_1,C) (D,C) (T_1,C) (D,C) (T_1,C)

Fig. 11. Checkerboard styled iterations [40].

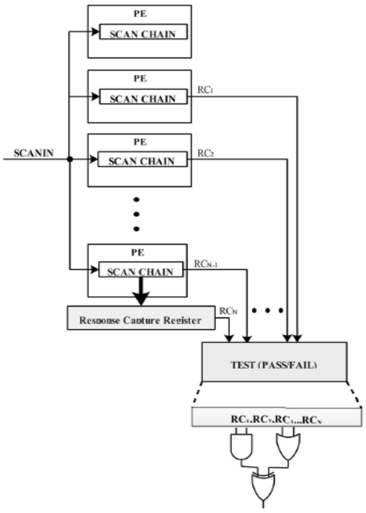


Fig. 12. Partial scan based scan-in, scan-out with comparison logic for stuck-at fault testing [41].

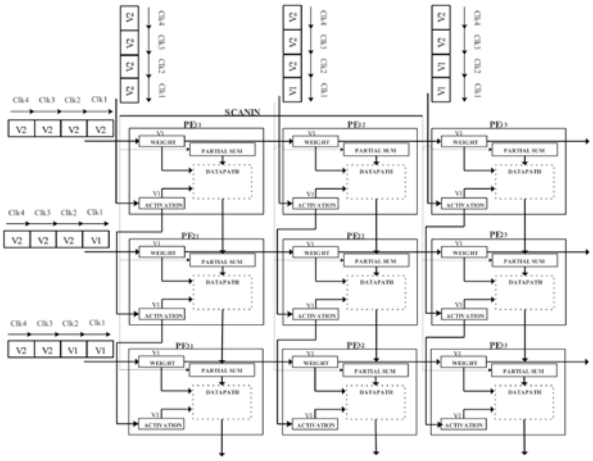


Fig. 13. Delay fault testing for partial scan method [41].

This technique also uses the systolic flow to deliver partial scan test pattern on PE-level [41]. The patterns are generated for combinational multiplier and adder logic of the PE. As shown in Fig. 12 & Fig. 13. A portion of the test pattern is applied functionally via activation and weight register to the combinational logic. And the remaining portion of scan test pattern is broadcasted to scan chains of summation circuitry inside each PE. The proposed method is also able to perform functional test on PE-level for functional test patterns. Since only a subset of scan flops are used, a significant reduction in test time and test power is observed for stuck-at and transition test patterns. The patterns are generated for combinational MAC circuitry of PE, which further reduces the ATPG effort compared to [40]. Fault coverage of this method is 100%. The activation and weight test patterns are functionally loaded from activation and weight memories; this allows testing of primary inputs of boundary PEs. For test response checking an integral combinational comparison logic is used to detect mismatch among PE responses, this comparison is based on the probability that majority of the PEs are non-faulty. Since this is a partial scan approach, full scan based conventional delay fault testing methods cannot be used. To enable delay fault testing, an innovative LoC based method is proposed. In this method, vector pairs are loaded into PE's registers in a systolic manner. Thereby, launching transitions and capturing their responses systolically.

A major portion of combinational logic i.e., the multiplier is directly connected to activation and weight registers inside a PE. With full scan, serial shift of scan patterns through these scan elements causes higher switching power in each PE. Combined for the whole array, an exponential increase in shift power will occur. So, in comparison to full scan, with thepartial scan method a major test power overhead is saved by using non-scan activation and weight register. To ensure C-testability of the proposed method, the array is sub-divided into smaller groups of PEs. Since the method uses unconventional test pattern application method a novel delay fault testing method based on Launch-on-Capture (LoC) is proposed, shown in Fig. 13. As the partial scan based technique generates patterns for the MAC only so compared with [40] it requires less number of patterns and required test time.

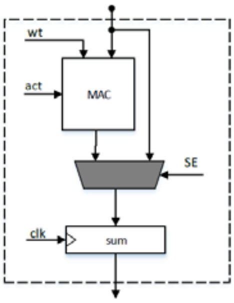


Fig. 14. Parallel pattern application to a non-scan PE with the additional MUX [42].

4.3 Non-scan based structural testing

The combinational logic of a MAC is a main component that affects the classification accuracy. Because of this fact a non-scan based test pattern application method is presented to test combinational logic of multiplier and adder circuitry of each PE in [42]. With systolic flow, w eight register and activation register are already controllable. However, partial sum capturing register is not controllable because it is connected to the output of the MAC only. For application of test patterns, sum register is made controllable with an additional MUX logic as shown in Fig. 14. Since patterns are applied via memory units and test patterns are applied functionally, all the primary inputs, primary outputs, and functional paths (interconnect among PEs) of the entire array are tested, this results in 100% coverage.

Moreover, it is known that the scan test patterns incur high switching activity during serial shift operations. This high activity gives rise to high switching power in combinational logic of PEs. So, by loading test patterns is parallel with the aid of systolic flow results in lesser test switching and peak power. Hence, compared to array level full scan DFT, this technique with its parallel pattern loading is more test time and test power efficient.

4.4 YAOTA

Further optimization of test pattern generation is done in *YAOTA*: Yield and Accuracy aware Optimum Test of DNN Accelerators [43]. Practically, for inference operation a large proportion of the weights is assigned to zero weights and can be pruned. This fact signifies that not every PE in an array have same level of impact and contribution on classification accuracy. And even in the presence of some faulty PEs the accelerator can still be used for non-critical applications. Based on this fact, a fault rate aware testing is proposed, where MAC faults are categorized based on their criticality assessment. To assess the criticality of faults, faults are injected into gate-level netlist combinational adder-multiplier logic. As shown in Fig. 15., the faults affecting the logic cone for the MSBs are regarded as critical and faults affecting the logic cone of LSBs are regarded as non-critical.

Then PEs are applied with structural test pattern in broadcast manner to all PEs in two phases i.e., i) to detect PEs with non-critical faults and ii) PEs with critical faults. It can be seen in Fig that positional weight of the MAC output bits has varied impact on accuracy. Faults in logic cone of the MSBs have more impact and are regarded as critical faults. Since accelerators with non-critical PEs ca still be used, an enhancement in yield for accelerators can be achieved by using the accelerators having non-critical faulty PEs.

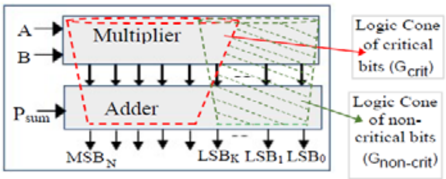

A
B
Multiplier
Psum
Adder
MSBN
LSBK LSB1 LSB0
Logic Cone of critical bits (Gcrit)
Logic Cone of non-critical bits (Gnon-crit)

Fig. 15. MSB and LSB logic cone separation in a MAC circuit based on criticality [43].

4.5 Master-Slave based testing

The full scan DFT provides maximum observability and controllability for detection of stuck-at faults. Additionally, full scan support for Launch-on-Capture (LoC) and Launch-on-Shift (LoS) allows for detection of delay faults. These features make the full scan a comprehensive test solution. However, a major disadvantage is the serial scan shift power consumption of scan shift operations. To implement a full scan based DFT along with reduced test power, a novel technique is presented in [44]. As show in Fig. 16., PEs are grouped into sub arrays. This grouping of PEs into sub arrays is done by considering (not exceeding) peak and shift power limitations. Each sub array has one Master PE and multiple Slave PEs. The Master directly loads the pattern from ATE and consequent responses are checked with ATE. The Slave PEs take input test pattern from the adjacent Master PEs. This allows parallel pattern application into majority of PEs, thereby reducing the overall test time without exceeding test power limits.

This technique enables fault localization on PE-level along with the test compaction. The sequence for fault localization is initiated in response to detection of faulty PE in a sub array. Since the array is divided into smaller sub arrays and sub arrays are tested simultaneously, the time to detect the faulty PE is shortened. This fault localization can aid in fault diagnosis to improve the yield and allow pruning for faulty PEs [44]-[48]. Also, the Master PE in each sub array is tested and compared with external ATE. So, comparing the responses of the Slave PEs against the Master's response results in zero aliasing probability.

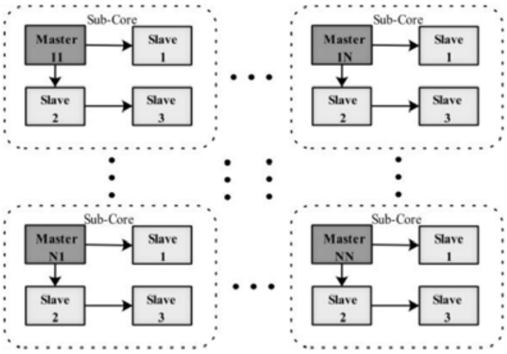


Fig. 16. Master-Slave configuration of PEs in a subarray to enable pattern loading [44].

4.6 Summary of PE-level based solutions

Using same test patterns for a group of or all of PEs result in test data volume reduction. The application method of these patterns is subject to constraints such as test time, test power, routing congestion. With systolic architecture based accelerators, there is a means for test dataflow among neighboring elements, which minimizes the requirement for extra test input pins to access a large number of PEs. Generating and applying test patterns on PE-level ensures maximum fault coverage of the whole array and reduces the ATPG runtime. Further optimization of test patterns can be done to target critical faults of MAC. The test data volume is further reduced by enabling built-in response checking by detecting mismatch in responses of identical PEs. By addressing testability at PE-level fault localization is enabled, which can provide pruning of faulty PEs and fault diagnosis to improve the yield. However, these solutions are proposed for structural testing at manufacturing stage of DNN accelerators. For real time applications such as autonomous automotives an infield testing method is necessary.

5. Summary and future directions

Permanent faults in the array of identical PEs can affect the yield and classification accuracy of the DNN accelerators. Testing these faults can ensure; i) higher yield, which results in reduced cost production of these accelerators and ii) reliable operation by detecting the faults affecting the classification accuracy. Testing methods are evolving with the evolution in architecture of DNN accelerators, with each one prioritizing different constraint i.e., yield, test time, test power, area overhead and test data volume. However, there is a need for a generic test solution for array module of accelerators. Which can exploit the homogeneity of the array for pattern applications and response checking. The solution should be configurable enough to target various kind of array architecture i.e., weight stationary, row stationary, non-systolic, SIMD. Also, it should be capable of exploiting homogeneity at Core and PE levels.

References

[1] G. Batra et. al., Artificial-intelligence hardware: New opportunities for semiconductor companies, McKinsey & Company, January 2019

[2] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams et al., Recent advances in deep learning for speech research at Microsoft, in ICASSP, 2013.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in NIPS, 2012.

[4] N. P. Jouppi, C. Young, and et al., In-datacenter performance analysis of a tensor processing unit, in Proc. of the 44th Annual International Symposium on Computer Architecture, 2017, Conference Proceedings, pp. 1–12.

[5] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, Deepdriving: Learning affordance for direct perception in autonomous driving, in ICCV, 2015.

[6] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," in Proc. IEEE, vol. 105, no. 12, pp. 2295-2329, Dec. 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in ICCV, 2015.

[8] Raju Machupalli, Masum Hossain, Mrinal Mandal, Review of ASIC accelerators for deep neural network, Microprocessors and Microsystems, Volume 89,2022,

[9] Mittal, S. A survey of FPGA-based accelerators for convolutional neural networks. Neural Comput. & Applic. 32, 1109–1139 (2020).

[10] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi and J. Kepner, Survey and Benchmarking of Machine Learning Accelerators, 2019 IEEE High Performance Extreme Computing Conference (HPEC), 2019, pp. 1-9.

[11] P. J. Bannon, ``Accelerated Mathematical Engine,'' U.S. Patent 0 026 078 A1, Sep. 20, 2017.

[12] NVIDIA. (2019). JETSON TX2 High Performance AI at the Edge.[Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-tx2/

[13] H. K. Lau, J. Ferguson, E. Griffiths, R. Singhal, and L. Harison, Enabling DFT and fast silicon bring-up for massive AI chip - case study, in International Test Conference (ITC), 2019, Conference Proceedings.

[14] H. Ma, R. Guo, Q. Jing, J. Han, Y. Huang, R. Singhal, W. Yang, X.Wen, and F. Meng, ``A case study of testing strategy for AI SoC,'' in Proc. IEEE Int. Test Conf. Asia (ITC-Asia), Tokyo, Japan, Sep. 2019, pp. 61-66.

[15] Y.-H. Chen, T.-J. Yang, J. S. Emer, and V. Sze, "Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices," IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 9, no. 2, pp. 292-308, Jun. 2019.

[16] M. Sankaradas, V. Jakkula, S. Cadambi, S. Chakradhar, I. Durdanovic, E. Cosatto, and H. P. Graf, "A massively parallel coprocessor for convolutional neural networks," in Proc. 20th IEEE Int. Conf. Appl. Specific Syst., Archit. Processors, USA, Jul. 2009, pp. 53-60.

[17] L. Cavigelli, D. Gschwend, C. Mayer, S.Willi, B. Muheim, and L. Benini, "Origami: A convolutional network accelerator," in Proc. 25th Ed. Great Lakes Symp. (VLSI), 2015, pp. 199-204.

[18] S. Chakradhar, M. Sankaradas, V. Jakkula, and S. Cadambi, "A dynamically configurable coprocessor for convolutional neural networks," in Proc. 37th Annu. Int. Symp. Comput. Archit. (ISCA), 2010, pp. 247-257.

[19] A. Gebregiorgis et al., Error propagation aware timing relaxation for approximate near threshold computing, in Proceedings of the 54th Annual Design Automation Conference (DAC), pp. 1–6, 2017.

[20] Li, Guanpeng, Siva Kumar Sastry Hari, Michael Sullivan, Timothy Tsai, Karthik Pattabiraman, Joel Emer, and Stephen W. Keckler. Understanding error propagation in deep learning neural network (DNN) accelerators and applications. in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1-12. 2017.

[21] Younis Ibrahim, Haibin Wang, Junyang Liu, Jinghe Wei, Li Chen, Paolo Rech, Khalid Adam, Gang Guo, Soft errors in

DNN accelerators: A comprehensive review, Microelectronics Reliability, Vol. 115, 2020.

[22] J. J. Zhang, T. Gu, K. Basu, and S. Garg, ``Analyzing and mitigating the impact of permanent faults on a systolic array based neural network accelerator," in Proc. IEEE 36th VLSI Test Symp. (VTS), USA, Apr. 2018, pp. 1-6.

[23] S. Kundu, S. Banerjee, A. Raha, S. Natarajan and K. Basu, Toward Functional Safety of Systolic Array-Based Deep Learning Hardware Accelerators, in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 3, pp. 485-498, March 2021.

[24] J. Rajski et al., Embedded deterministic test for low cost manufacturing test, Proceedings. International Test Conference (ITC), 2002, pp. 301-310,

[25] B. Lu et al., The test cost reduction benefits of combining a hierarchical DFT methodology with EDT channel sharing—A case study, 2018 13th International Conference on Design & Technology of Integrated Systems In Nanoscale Era (DTIS), Taormina, 2018, pp. 1-4.

[26] George, Kiran, and Chien-In Henry Chen. Logic built-in self-test for core-based designs on system-on-a-chip. IEEE Transactions on Instrumentation and Measurement 58, no. 5 (2009): 1495-1504.

[27] N. Mukherjee et al., Time and Area Optimized Testing of Automotive ICs, in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 1, pp. 76-88, Jan. 2021.

[28] R. Singhal, ``AI chip DFT techniques for aggressive time-to-market," Mentor, Siemens Bus., White Paper, 2019.

[29] H. K. Lau, J. Ferguson, E. Griffiths, R. Singhal, and L. Harison, Enabling DFT and fast silicon bring-up for massive AI chip - case study, in International Test Conference, 2019, Conference Proceedings.

[30] H. Ma et al., A Case Study of Testing Strategy for AI SoC, 2019 IEEE International Test Conference in Asia (ITC-Asia), 2019, pp. 61-66,

[31] Kung, Why systolic architectures?, in Computer, vol. 15, no. 1, pp. 37-46, Jan. 1982

[32] U. S. Solangi, M. Ibtesam and S. Park, Time multiplexed LBIST for in-field testing of automotive AI accelerators, IEICE Electronics Express, 2021, Volume 18, Issue 24, Pages 20210451

[33] W. H. Kautz, ``Testing for faults in combinational cellular logic arrays," in Proc. 8th Annu. Symp. Switching Automata Theory (SWAT), Austin, TX, USA, 1967, pp. 161-174,

[34] A. D. Friedman, ``Easily testable iterative systems," IEEE Trans. Comput., vol. C-22, no. 12, pp. 1061-1064, Dec. 1973,

[35] C.-H. Sung, ``Testable sequential cellular arrays," IEEE Trans. Comput.,.vol. C-25, no. 1, pp. 11-18, Jan. 1976,

[36] H. Elhuni, A. Vergis, and L. Kinney, ``C-testability of two-dimensional iterative arrays," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. CAD-5, no. 4, pp. 573-581, Oct. 1986, doi: 10.1109/TCAD.1986.1270228.

[37] F. Lombardi, ``On a new class of C-testable systolic arrays," Integration, vol. 8, pp. 269-283, Dec. 1989,

[38] W. R. Moore and V. Bawa, ``Testability of a VLSI systolic array," in Proc. 11th Eur. Solid-State Circuits Conf. (ESSCIRC), Toulouse, France, Sep. 1985, pp. 271-276,

[39] S.-K. Lu, J.-C. Wang, and C.-W. Wu, ``C-testable design techniques for iterative logic arrays," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 3, no. 1, pp. 146-152, Mar. 1995,

[40] A. Chaudhuri, C. Liu, X. Fan and K. Chakrabarty, C-Testing and Efficient Fault Localization for AI Accelerators, in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2021

[41] U. S. Solangi, M. Ibtesam, M. A. Ansari, J. Kim and S. Park, Test Architecture for Systolic Array of Edge-Based AI Accelerator, in IEEE Access, vol. 9, pp. 96700-96710, 2021,

[42] M. Ibtesam, U. S. Solangi, J. Kim, M. A. Ansari and S. Park, Reliable Test Architecture With Test Cost Reduction for Systolic-Based DNN Accelerators, in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 3, pp. 1537-1541, March 2022, doi: 10.1109/TCSII.2021.3108415.

[43] M. Sadi and U. Guin, Test and Yield Loss Reduction of AI and Deep Learning Accelerators, in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 41, no. 1, pp. 104-115, Jan. 2022, doi: 10.1109/TCAD.2021.3051841.

[44] U. S. Solangi, M. Ibtesam and S. Park, Master-slave based test cost reduction method for DNN accelerators, IEICE Electronics Express, 2021, Volume 18, Issue 24, Pages 20210425

[45] S. Han et al., Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, arXiv preprint arXiv:1510.00149, 2015.

[46] J. Yu et al., Scalpel: Customizing dnn pruning to the underlying hardware parallelism, in Proceedings of the 44th Annual International Symposium on Computer Architecture, pp. 548–560, ACM, 2017.

[47] H. Li et al., Pruning filters for efficient convnets, arXiv preprint arXiv:1608.08710, 2016.

[48] S. Anwar et al., Structured pruning of deep convolutional neural networks, ACM Journal on Emerging Technologies in Computing Systems (JETC), vol. 13, no. 3, p. 32, 2017.

[49] P. Molchanov et al., Pruning convolutional neural networks for resource efficient inference, 2016.