IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 30, 2021, accepted June 21, 2021, date of publication July 5, 2021, date of current version July 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3094741

Test Architecture for Systolic Array of Edge-Based AI Accelerator

UMAIR SAEED SOLANGI[1,2], **MUHAMMAD IBTESAM**[1], **MUHAMMAD ADIL ANSARI**[2], **JINUK KIM**[1], **AND SUNGJU PARK**[1], **(Senior Member, IEEE)**

[1]Department of Computer Science and Engineering, Hanyang University, Seoul 04763, South Korea
[2]Department of Electronic Engineering, Quaid-e-Awam University of Engineering Science and Technology, Larkana 77150, Pakistan

Corresponding author: Sungju Park (paksj@hanyang.ac.kr)

This work was supported in part by the Higher Education Commission, Government of Pakistan, under the Scholarship Program Faculty Development of University of Engineering Science and Technology Pakistan/University of Engineering and Technology (UESTPs/UETs), in part by the BK21 FOUR (Fostering Outstanding Universities for Research) funded by the Ministry of Education (MOE), South Korea, and in part by the National Research Foundation of Korea (NRF).

ABSTRACT The application diversity and evolution of AI accelerator architectures require innovative DFT solutions to address issues such as test time, test power, performance and area overhead. Full scan DFT, because of its enhanced controllability and observability, is an industrial de facto test strategy. However, it may not yield an optimal test solution with stringent design constraints of edge-based AI accelerators. In this paper, a novel test architecture based on selective-partial scan is proposed for performance, power and area (PPA) overhead constrained edge-based systolic AI accelerator. In this architecture, the structural test patterns are applied partly in functional manner, which reduces the testability problem of an array to that of a single processing element (PE); thus, resulting in reduced test time and test data volume. Moreover, a delay fault testing method based on Launch-on-Capture is presented for the partial scan based proposed architecture. Experimental results show that proposed architecture is efficient in terms of test power and test time when compared to full scan DFT.

INDEX TERMS Design for testability, systolic arrays, TAM, testing.

I. INTRODUCTION

Currently, most of the artificial intelligence (AI) applications are running on clouds/datacenters. However, with an enormous amount of data being produced by the consumers, there is a growing need for edge AI accelerators. The edge computing offers a cost-effective and low data bandwidth solution by bringing data processing local to the source of data, with improvement in the response time [1], [2]. Recent AI resurgence has been due to deep neural networks (DNNs), which process more hidden layers and result in increased classification accuracy [3]. Moreover, there is a growing interest in convergence of DNN processing capabilities with the edge computing devices to enhance application paradigm [4], [5]. NVIDIA, Google and Tesla have already introduced specialized accelerators for edge inference applications [6]–[8] with smaller physical and power footprint.

Several DNN hardware accelerators are being developed for inference tasks on application specific integrated circuits (ASIC) [9]–[11]. ASIC based AI hardware accelerators usually favor spatial dataflow architectures, which enable transfer of data between neighboring processing elements (PEs). This pipelined dataflow avoids the need for frequent memory read operations that result in energy optimization [10]. Variants of weight-stationary systolic array are used to accelerate the CNN inference with low power consumption in [12]–[14]. Essentially, a weight-stationary systolic array allows reusability of weights in implementing subsequent layers of DNN. This architecture has also been adopted by Google Inc. for their industrial Tensor Processing Unit (TPU) [9] due to its low bandwidth feature.

Recent study has shown that error resilience of AI is insufficient to overcome the effects of stuck-at-faults for weight-stationary systolic array. As only 0.005% faulty PEs can degrade the classification accuracy for up to 74.13% [15]. The reason for such drop in accuracy is that the stuck-at-faults frequently affect the higher order bits of the MAC output. This shows that in addition to yield enhancement, the fault coverage (FC) is also crucial for reliable DNN operation. Moreover, the edge-based AI hardware requires small physical and power footprint. The main limiting factor with scalability of full scan DFT approach in terms of test overhead is

The associate editor coordinating the review of this manuscript and approving it for publication was Luca Cassano.

the addition of an extra scan MUX logic (per Flip Flop) and additional routing (for scan chain stitching). Which in case of an accelerator, multiplies with increasing array size of the accelerator and may exceed allowed limits of size and power for an edge based AI accelerator. A full scan based C-testing approach was proposed in [16], where testability effort has been confined to single PE. This C-testing approach results an improvement in test time and test pattern reduction. In this paper, we propose partial scan based DFT architecture having low overhead (PPA) for edge-based AI hardware. The key contributions of this paper are;

- Investigations of conventional test solutions for systolic array; Sequential ATPG and Full scan are first implemented for weight-stationary systolic array (based on TPU model) with FC analysis and associated test overhead.
- A test architecture based on partial scan and systolic pattern loading with a built-in checking circuitry is proposed for weight-stationary systolic array (based on TPU model).
- Partial broadcasting is proposed for test pattern loading (for test time synchronization) for arrays of different sizes ($>16 \times 16$). Test cost of the proposed test architecture is presented and compared with full scan.
- A delay fault testing method based on Launch-on-Capture is presented for the proposed architecture.
- Evaluation of the proposed method is also performed in comparison with Checkerboard based full scan method [16].

The remaining paper is organized as follows. In Section 2, various array testing methods are discussed. Section 3 briefly introduces Google's TPU model that was used for implementation of this work. In Section 4, we present our analysis of conventional test methods. Section 5 presents the details of the proposed test architecture and its operation. Section 6 gives the details for the proposed solution for at-speed testing with partial scan based test architecture. In Section 7, the results for associated experiments are given. Finally, we present the conclusion in Section 8.

II. RELATED WORK

Testing of iterative arrays have been previously studied with C-testability, which is primarily based on functional testing with constant number of test patterns to test each PE [17], [18]. Friedman [18] presented a theory for modified C-testability based on the function of the processing cell, which detects single faulty cell of an array. Sung [19] presented sufficient conditions to ensure testability of unilateral and bilateral arrays for detection of a single faulty unit. Elhuni *et al.* [20] have shown that the test pattern length can be made independent of the size of the array, but this method is limited to one dimensional iterative array. Lombardi [21] has extended the C-testability approach to systolic arrays provided there are additional patterns to be used for testing the sequential cells (FFs) of a processing unit. These patterns ensure every possible transition as sufficient condition to test the sequential cells. Moore and Bawa [22] presented testing method for a bit-level unilateral systolic array, where length of the test vectors increases with the size of an array. It uses a row comparator for each column for generating test pass/fail result, thereby compressing the test response for the array. The main limitation of C-testability based functional testing is the detection of a single faulty cell from the whole array. BIST solutions (based on single cell fault model) for array multipliers with deterministic (constant) patterns are presented in [23] and [24] in which MUX logic is introduced as a DFT solution to switch between functional and test mode.

Besides, strategies for testing identical cores have been proposed. Giles *et al.* [25] have addressed the testing of multiple identical cores by providing a scalable parallel test access mechanism (TAM) architecture. In this architecture, the response paths from each core are pipelined through comparators in order to compare the response of each core with a core, which is already tested by the ATE. Han *et al.* [26] proposed a TAM architecture for multiple identical cores that uses majority voting for checking test response of each core and the majority response is cross checked with the ATE response. The key takeaway is that majority of the cores will be matched to the expected response and can distinguish the minority cores with faulty response through majority analyzer. A method for concurrent error checking between neighboring elements in a systolic array is presented in [27]. This requires additional XOR logic for output comparisons between neighboring elements and may result in an increased test area overhead.

Ma *et al.* [28] have tested an AI based SoC by broadcasting test patterns by embedded deterministic test (EDT) to the identical cores to reduce test time. These cores are isolated by IEEE 1500 wrappers and are tested by means of comparator in subsequent test modes. However, this testing approach results in a very high routing congestion due to input channel broadcasting, and due to the hardware overhead associated with EDT, it is not an optimum solution for the edge-based AI accelerator. Moreover, this state-of-the-art solution uses full scan DFT approach, which may not be a suitable solution for systolic array. The reason is that the circuit connectivity may not allow each FF to provide same level of controllability and observability, which is the case for most of the pipeline flow-based accelerators with unidirectional connections. A framework for functional criticality based stuck-at fault analysis for inference applications is presented in [29]. This machine learning based gate-level netlist analysis prior to manufacturing test to target location specific structural faults for testing may optimize the test generation by specifying the test points/ test pattern generation for these critical locations. However, this machine learning based analysis may add to the time-to-market constraint and affect the overall test cost.

Recently, a C-testing approach based on full scan DFT is proposed in [16]. Homogeneity of PE is exploited for testing sub-arrays in multiple iterations, which are executed in

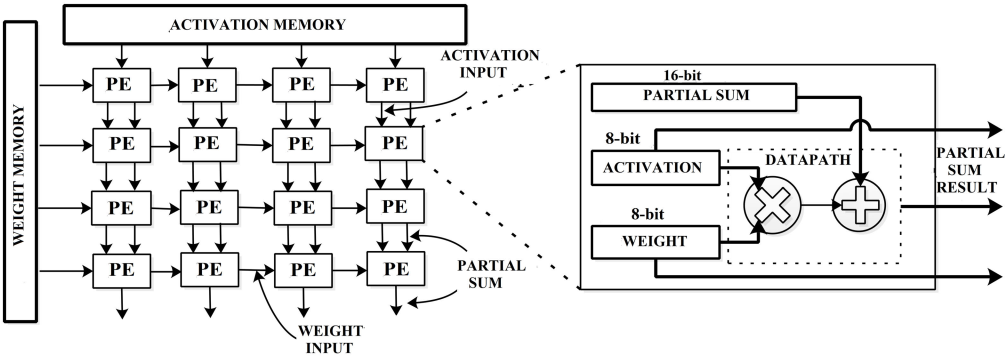


FIGURE 1. Implemented TPU's Systolic Array model.

checkerboard style. Compared with industrial EDT approach, this proposed method results in improved test time and since test patterns are generated on PE level, the number of patterns is also reduced. This avoids the need for ATPG effort for full array. Unlike previous work, our proposed architecture enables the concurrent detection of multiple fault cells compared to single fault model of C-testing approach. The ATPG effort has been reduced to generation of the test patterns only for combinational logic of PE. Also, it requires less ATE involvement by checking the response by the built-in test pass/fail logic. In the proposed test architecture, the test patterns are loaded systolically as well as by broadcasting to partial scan chains. The test response of all PEs is compared for test pass/fail signal for the array. Consequently, in comparison to the full scan test, the test time and test power are significantly reduced; moreover, the area overhead is also reduced.

III. SYSTOLIC ARRAY-BASED MATRIX MULTIPLICATION UNIT

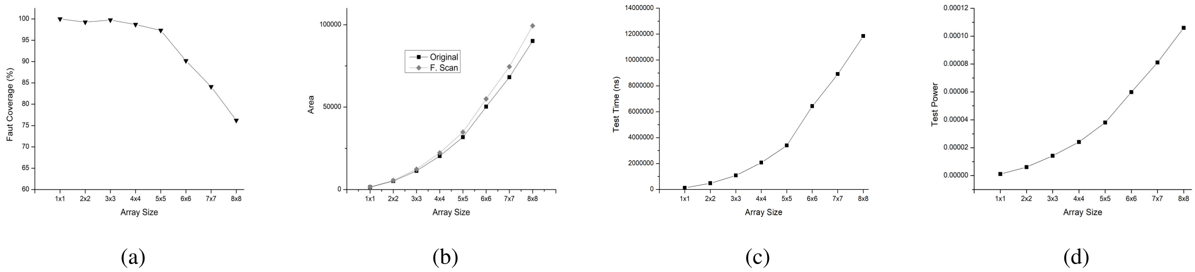
In order to perform real time inference operation on streaming data, the accelerator needs to perform frequent data read operations. This read operation is more energy consuming as it needs to access the memory. Therefore, such read operations in an edge-based accelerator may not be a suitable choice due to its limited energy resource. On the other hand, the accelerators with spatial connection, like systolic array, the connectivity between neighboring cells requires much less energy and low bandwidth [12]–[14]. Google's TPU is mainly used in the Clouds/Datacenters for inference applications with 256×256 Matrix Multiply Unit (MMU) as inference engine. Whereas its Edge version with the smaller array size and power consumption (2 Watts) uses quantized weight bits, e.g., 8-bits [7]. TPU's MMU is based on weight-stationary systolic array to allow reusability of weights in subsequent layers of the DNN. Each PE of the array generates a partial sum and are accumulated at the end of each column in an accumulator.

During the normal operation of an MMU, initially, the pre-trained weights are fed from weight memory and are systolically shifted across the corresponding PE row of the array. The weight is stored in the weight register of the PE, as depicted in Fig 1. Subsequently, the activation inputs are fed from the activation memory and are systolically shifted across the corresponding PE column along with the generated partial sum. Each PE performs multiplication between activation and weight inputs. This product is then added to the partial sum that is generated by the preceding PE to generate the partial sum for the succeeding PE. In a PE, the partial sum is generated by the combinational datapath (multiplier and summation circuitry) and is captured by the partial sum register of the succeeding PE (along the column). Due to industrial significance of TPU, we have implemented the proposed DFT architecture and developed the Verilog model of this systolic MMU based on [30]–[33].

IV. CONVENTIONAL TEST SOLUTIONS FOR TPU'S SYSTOLIC ARRAY

To find an optimal test solution for the systolic TPU, first, the gate-level netlist of the verilog model is subjected to sequential ATPG, as it incurs no DFT hardware overhead. For this, Tetramax ATPG is used to obtain the FC and sequential ATPG patterns. It is observed that with an increasing array size, the FC degrades, as shown in Fig. 2a. This happens due to an increasing depth of sequential path. Since inference-based classification accuracy of the DNN operation is heavily affected by the degradation of FC, the sequential ATPG testing is not a suitable test solution for the systolic array-based accelerator.

For full scan test, all the FFs are replaced with muxed scan FFs and it was synthesized with Synopsys Design Compiler. This conventional approach offers FC near to 100%. However, with full scan testing, the area overhead, test time and test power increase with increasing array size. The DFT area/logic overhead increases mainly due to increasing number of scan MUX logic and routing overhead, as shown in Fig. 2b. The test time increases due to increasing number of scan cells, as shown in Fig. 2c, and the test power increases due to serial scan shift of test data, as shown in Fig. 2d. This makes full scan method unscalable and infeasible approach for testing an edge-based AI hardware, which has smaller



(a) (b) (c) (d)

FIGURE 2. Issues with conventional Test Solution (a)FC drop with increasing array size (b) Increase in area overhead for Full Scan DFT (c)Increasing test time with increasing array size (d) Test power for increasing array size.

physical and power footprint. If the array size is smaller, e.g. up to 8 × 8, full scan implementation is done with single scan chain. For the array sizes greater than 8 × 8, e.g., 16 × 16 and 32 × 32, multiple scan chains are synthesized. The number of scan chains is configured to allow the same scan chain length for 16 × 16 and 32 × 32 as there are in 8 × 8 array, i.e. 4 scan chains for 16 × 16 array and 16 scan chains for 32 × 32 array. This is done to restrict the scanin and scanout pins.

V. PROPOSED TEST ARCHITECTURE

Since the array contains identical PEs, it would be an effective approach to confine the test efforts to a single PE. To achieve this, the components of PE are separately observed. A PE consists of sequential cells (activation, weight and partial sum register) and a combinational datapath. For the synthesized model, in a single PE, the combinational datapath comprises most of the logic (55%) and interconnect (57%), also it contributes to over 90% of the computation in a DNN layer. Moreover, the stuck-at faults in the datapath severely affect the classification accuracy in inference applications. Because of this structural and computational significance, datapath circuit is considered exclusively for fault detection. In our model, the datapath consists of an 8-bit multiplier and a 16-bit summation circuitry, as shown in Fig. 1. Tetramax ATPG provided 100% coverage with only 15 test patterns for this datapath circuit (tested separately). The test pattern length was 32-bit wide; 16-bits for partial sum input, 8-bits for activation and 8-bits for weight registers. The test architecture is developed to allow the application of these test patterns to each PE separately yet simultaneously. Constraining testability to a single PE allows reduction in test data volume, as only 16 test patterns will be used to test array of any size.

It is depicted in Fig. 1 that the activation and weight registers have pipelined connectivity; thus, these registers only allow applying the test patterns from primary inputs and they cannot capture any test response from any datapath circuitry. Whereas, the content of partial sum register of a PE is transferred to the succeeding PE's partial sum register. This spatial connection enables the capture of test response of a PE into the succeeding PE. For this reason, only the partial sum register is synthesized with scan chain to provide essential observability for the captured response. The structural test patterns for datapath unit are applied in functional manner via activation and weight registers, and the test response is captured by the partial sum register's scan chain of the succeeding PE in a column. A single column of PEs with the proposed architecture is shown in Fig. 3. The Scan input 'SCANIN'is broadcast to all the PEs of the array to load each scan chain in parallel. The capture response from each scan chain ($RC_1, RC_2, \cdots, RC_N$) is shifted to a Built-in checking circuitry that compares the response bits from each scan chain and generates pass/fail signal. Since last PE in a column is connected to the accumulator, to capture the response of the last PE of each column, partial sum from the last PE is loaded into 'Response Capture Register'as shown in Fig. 3.

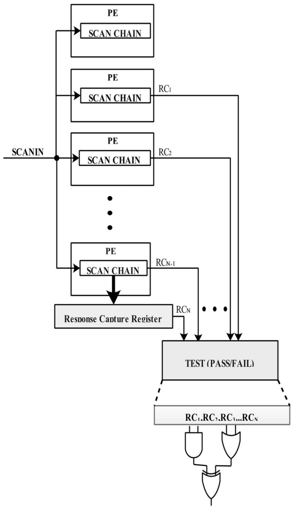


FIGURE 3. A single column implementation with proposed architecture.

A. TEST PATTERN APPLICATION WITH BUILT-IN CHECKING

It is assumed that the memory unit feeding the test patterns is already tested. We propose to use on-chip memory to store the 15 test patterns that will be loaded into the activation and weight memory. The approach is based on deterministic BIST techniques, where compressed patterns/seeds are stored in the on-chip ROM. For the PEs directly connected to the both memories, the activation and weight part of the datapath's test patterns are applied in a single test clock cycle. On the successive test clock cycles, the same test pattern will be

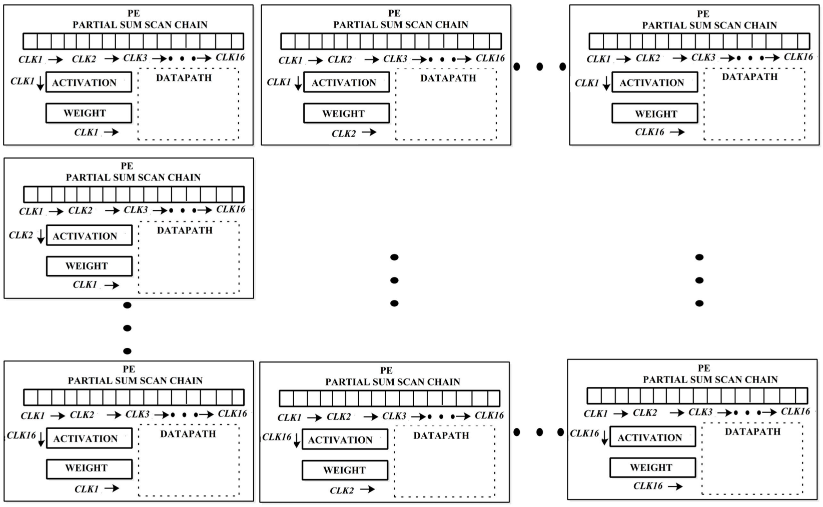


FIGURE 4. Number of clocks required for parallel and serial loading of a single test pattern in a 16 × 16 array.

loaded into the rest of the PEs by systolic shifting. To do that, the enable logic signal of the weight register FFs is ORed with test mode signal to allow weight shifting during the test mode. For the partial sum portion of the datapath input pattern, the test pattern is serially shifted-in with multiple (equal to the number of partial sum FFs in a single PE) test clock cycles for each PE by the shared scanin pin. Fig. 4 shows the loading of the test patterns for a 16×16 array and the number of required clock cycles for the input pattern loading in the registers.

During the functional mode, while capturing the test response of a datapath into the succeeding PE, the activation and weight inputs are held constant in the memory (activation and weight) units. Again, during the test mode, the captured test response is unloaded serially with simultaneous shifting in the next test pattern from the partial sum scan chain, while activation and weight inputs are applied systolically. If, during the test response unloading, any bitwise mismatch between the test responses of various PEs occurs, it is registered as a Test FAIL flag by the built-in checking circuitry and the testing will be ended.

Since in the proposed technique, the test patterns are applied in functional manner, the test response against a test pattern for all the PEs must be same in case of no stuck-at fault. However, the presence of a stuck-at fault in datapath unit and registers will result in a mismatched datapath response compared to the response of other non-faulty PEs in a column. The proposed architecture has an integral built-in checking circuitry that acts as a comparator as shown in Fig. 3, which has a combinational logic. In the built-in checking circuitry, serial scan out of each PE is shared with AND and OR gate input that can detect any single bit mismatch among any number of output response streams from partial sum scan chains by raising the flag to '1'. The flag gives false indication only when all the PEs have same faulty response. All the response flags from built-in checking circuits are ORed for the detection of faults of whole array.

Moreover, successive pattern application with systolic dataflow ensures that each FF of the activation and weight registers goes through all possible transitions, ensuring the testability conditions for the FFs. So, if there is a stuck-at fault in any FF of the registers (activaiton, weight, partial sum) in a PE, it would result in a different input pattern being applied to the datapath circuit and its/their response will cause a mismatch at built-in checking circuitry when compared with other PE responses. The FFs of the partial sum register are tested by applying the scan chain pattern (..001100..) at the start of serial shifting.

In addition to the structural testing, the proposed architecture enables the functional testing of the PEs, where functional patterns can be loaded into the activation and weight registers through memory units systolically. Subsequently, each datapath's functional response is captured by the partial sum scan chain and shifted out concurrently from each PE. These responses can be compared and checked by the built-in checking circuit.

B. PARTIAL BROADCASTING OF INPUT PATTERNS

For the array size of more than 16×16 PEs, the partial broadcast is proposed in which the array is divided into multiple blocks of 16×16 PEs. Instead of broadcasting the test patterns to all the PEs of the array, broadcasting is done to the first PE of each block. Each block will require exactly 16 cycles to load the test patterns into the partial sum scan chains (serially) and into the 16 activation and weight registers (systolically). Since broadcasting allows the loading/unloading of patterns for each block in 16 cycles, the whole array requires only 16 cycles to load the pattern. Hence this broadcast technique improves the testing time of the whole array.

This broadcasting can also be used for the array sizes which are not multiple of 16. For example, a 24×24 array can be divided into 4 blocks of 12×12 as shown in Fig. 5. The only restriction is that a block cannot be greater than 16×16 as the

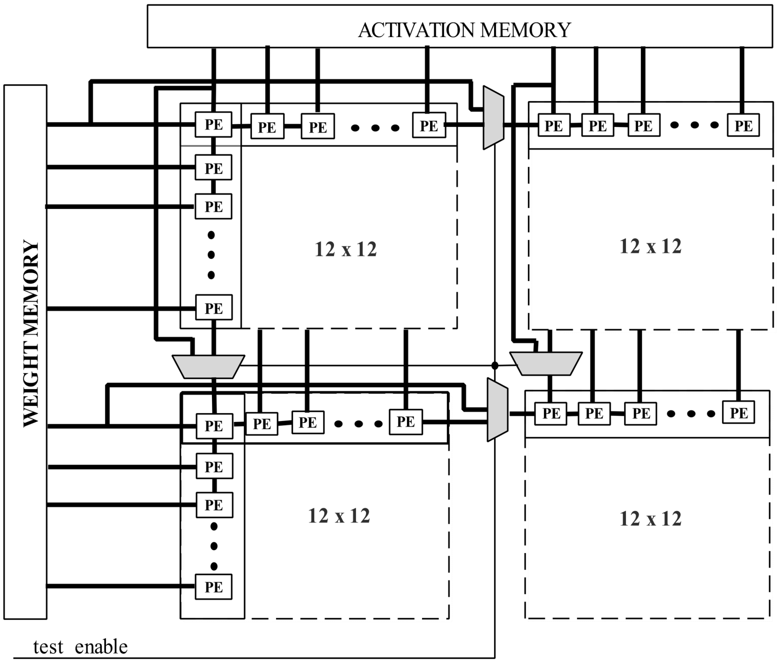


FIGURE 5. An Example of partial broadcasting for array (24 × 24).

pattern loading requires exactly 16 cycles. In that case, first block is always made of 16 × 16. This broadcasting is enabled only during test mode. This way the array size would not be a factor to affect the overall test time. And number of test cycles (eventually test time) determined by the size of the partial sum register. Furthermore, this broadcasting of input test patterns allows the scalability of the proposed architecture with increasing array size. Fig. 5 shows an example of partial broadcast implementation. Where MUX logic is inserted at boundaries of 16 × 16 block to allow input pattern sharing to the first PE of the other block directly from memory units.

VI. AT-SPEED TESTING WITH PROPOSED ARCHITECTURE

To enable at-speed testing of the TPU systolic array, vector pairs for launching the transitions are extracted for the datapath combinational logic (such as done for stuck-at faults). As the activation and weight registers are non-scan flip flops the vectors are applied at-speed from the memory in the systolic pipelined method. The application of vectors is depicted in Fig. 6. First the vector V1 is loaded into activation and weight register systolically, while concurrently loading the partial sum scan chain input (serially) in 16 clock cycles. After the vector V1 is loaded into each PE, test enable is deactivated to allow at-speed launching of transition with vector V2. It may be noted the vector V2 for activation and weight register is loaded from memory units while the transition vector for at-speed testing of summation circuitry is launched (generated functionally) from the datapath logic of the previous PE (vertically connected). With systolic loading, the activation and weight register receive same transition launching vector V2. While the transition launching vector for summation circuitry is from vertically preceding adjacent PE (for example, from PE_{11} to PE_{21} and PE_{22} to PE_{32}).

Based on the proposed test architecture an implementation for a 3 × 3 array is presented in Fig. 6. The launch vector V2 is timed to match the systolic loading into adjacent PEs. For example, at first clock cycle the vector V2 (activation and weight) is loaded into PE11 only and on the 2^{nd} cycle vector V2 is loaded systolically into PE_{12} (activation) and PE_{21} (weight) from PE_{11}. While remaining part of V2 is loaded from the memory into PE_{12} (weight) and PE_{21} (activation) on the 2^{nd} cycle. The 2^{nd} cycle will capture the response form PE_{11} into PE_{21} (vertically adjacent) in the partial sum register as shown in the timing diagram and this captured response will launch a transition for the summation circuitry of PE_{21} as shown in timing diagram Fig. 7. A series of launch/capture at-speed clocks (in addition to shift-in/shift out clocks for loading V1) is applied to test the whole array for a single vector pair (V1:V2). The number of these at-speed clocks is dependent upon the size of the array, i.e., for $n \times n$ array $n + n$ at-speed clocks are required for a single transition pattern in addition to 16 clock cycles for scanning in and scanning out at scan shift frequency.

For a 3 × 3 array, after 6 cycles, the response from each column is collected at the response capture registers. But each column's response is captured at different cycles. For column 1, since there are 3 PEs 4^{th} cycle will capture the response. For column 2, 5^{th} cycle captures the response and 6^{th} cycle will capture the response for column 3. Each capture response register is clocked at these specified clock cycles. This is enabled by the clock control circuitry shown in Fig. 8. This control is required to prohibit any fault to be masked (change any faulty response value to be corrected) that can

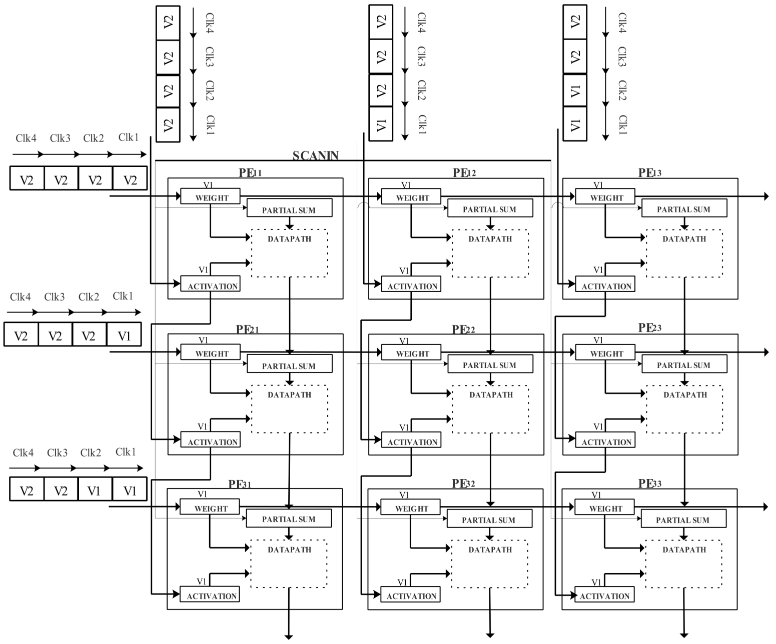


FIGURE 6. Vector loading into a 3 × 3 systolic array for transition delay testing.

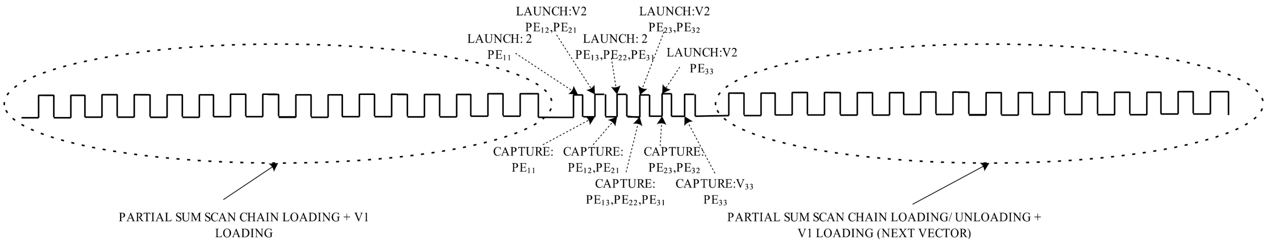


FIGURE 7. Timing diagram with Launch, Capture, Scanin and Scanout operations for a 3 × 3 array.

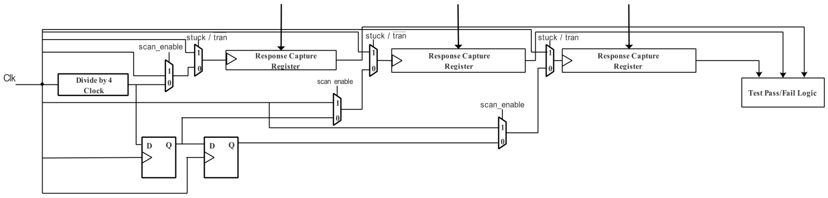


FIGURE 8. Clock synchronized response capture unit for a 3 × 3 array.

happen due to multiple capture cycles. For example, if any fault occurs in the response from PE_{11} in the 2^{nd} cycle the 3^{rd} can mask this fault into PE_{21} summation circuitry. The 5^{th} and 6^{th} cycle may mask this fault for the whole column. So, to stop masking of this fault response capture, the response of column 1 is locked at 4^{th} cycle. As this fault changes the launch vector for (vertical) adjacent PE's summation circuit (in column 1), a different transition vector is propagated through that column (1) as compared to other columns (2 3), resulting in different response. The partial sum registers of PEs other than first PE in each column is connected with partial sum response of vertically adjacent PEs. This will result in partial sum registers launching different transition vector value through that column. However, the launch vector from partial sum registers of horizontally adjacent PEs is same. Hence the built-in checking is done among the column responses to detect this mismatch in responses, unlike proposed scan testing method (in Fig. 3), where response is compared among each PE. Here the column response is the response from the last PE in that column. After the 6^{th} th cycle, response from each response capture register is serially unloaded concurrently into the built-in checking circuit, which is a separate unit than the one that is used for stuck-at fault testing. Like stuck-at fault response checking, built-in checking circuit will detect any mismatch among the column responses. A signal textquotesingle $stuck/(\overline{tran})$', switches between stuck-at fault testing and transition testing mode. This signal also switches between Test/Pass fail logic unit of stuck-at faults testing and Test/Pass fail logic unit of transition faults testing. The overhead for the response capturing unit is 1.6% for a 3 × 3 array and since this unit is implemented per column of the whole array, its overhead decreases with the array size. This is because the response capture unit increases linearly unlike the exponential increase in logic of the whole array.

VII. EXPERIMENTS AND RESULTS

A. AREA OVERHEAD

The systolic TPU model for various array sizes is synthesized with the Design Compiler on SAED 32nm Library. With proposed synthesized design there is DFT logic area overhead reduction of approx. 11% in sequential area, 9% in interconnect area and around 4% in total area compared to full scan DFT given in TABLE 1. It is evident that total area overhead is marginally improved as compared to the sequential and routing area overhead. This is because, most of the total area is taken up by the combinational logic of datapath circuit. The area overhead of the test pass/fail logic for a 32 × 32 array is 0.1%.

TABLE 1. Reduction in area overhead.

AREA	PE_Full	PE_Partial	Area red. (%)
Combinational area	1001.58	999.54	0.2
Sequential area	301.92	270.4	10.44
Net Interconnect area	219.79	200.42	8.81
Cell area	1303.5	1269.95	2.57
Total area	1523.29	1470.38	3.47

B. PERFORMANCE OVERHEAD

Full scan FFs always introduce performance penalty to the original circuit due to additional MUX logic and fanout in the critical path. The synthesized proposed design has no scan chain fanout for activation and weight register flip flops. This represents an average of 26.44% reduction in fanout capacitance compared to the full scan flip flops. This results in reduced delay and less dynamic power (αCV^2f) for sequential cells (FFs). Moreover, there is no additional propagation delay of scan MUX logic in activation and weight registers.

C. TEST TIME

For evaluation of test time for the arrays with proposed design. Gate-level netlist simulations were performed on Modelsim with test frequency of 10 MHz. In addition to 15 test patterns to test the datapath, a scan chain test pattern ($\cdots 00110 \cdots$) is included to test the partial sum scan chain. With these gate-level netlist simulations the test time improvement for the proposed design is compared to full scan design in TABLE 2. As test application time for the whole array (in the proposed architecture) is matched with the test time of a single PE, it remains the same for any array size and there is growing improvement in the test time with an increasing array size. For an array of less than 16 PEs, 16 clock pulses will still be applied to activation and weight registers to synchronize with 16 clock pulses for shift-in and shift-out operation of each partial sum scan chain. For this reason, the test patterns data from memory is held constant for 17 (16 for shift in and 1 for capture) cycles. Test time for multiple scan chain-based arrays (16 × 16 and 32 × 32) with same scan chain length does not improve due to increased number of patterns, which is due to relative increase in combinational logic.

The table 3 implements the full scan based array testing as proposed in [34]. Which mainly considers time-to-market as the main constraint and proposes to use broadcasting of the test patterns to test multiple identical modules simultaneously. The proposed architecture maintains advantage over full scan DFT with various array module implementations, where the whole array (full scan) is divided into smaller sub-modules for pattern broadcasting/sharing.

D. TEST POWER

As with full scan DFT, shift power of serial scan shifting of test patterns depend on the length of scan chain and size of the combinational logic. This results in the increase in shift power with increment in array size. For the proposed architecture, value change dump (VCD) files were generated by the gate-level netlist simulation for obtaining event driven test power of the whole array with Synopsys Primepower. As the proposed architecture uses a smaller number of patterns with serial scan chain length limited to 16 for each PE, there is improvement in serial shift power compared to full scan DFT, as shown in TABLE 2 and TABLE 3. Compared to Full scan DFT where a single pattern may cause multiple transitions during serial shift operations at single scan FF (of activation and weight register). In the proposed architecture, a single pattern loading may cause only a single (at maximum) transition at non-scan FF (activation and weight register). Also, the partial scan is connected to summation circuitry of the datapath, whereas full scan chain is connected to the whole datapath unit (summation and multiplier circuitry). This results in lesser dynamic power consumption in the combinational logic during the scan shift operation in the proposed architecture. This reduction in number of transitions combined with smaller scan chain per PE and lower scan shift power per pattern causes a proportional reduction in the overall test power.

Moreover, with limited power footprint, edge-based AI devices are more vulnerable to peak test power (maximum power consumed at any single test clock cycle), as it may cause reliability issues (like hotspots). Since the proposed architecture uses partial scan, the number of scan FFs capturing the response at a single test clock cycle and number of transitions (during serial shift) occurring at any test clock cycle is reduced. Both factors contribute to reduction in peak power. This reduction in peak test power improves the

TABLE 2. Results for the test time, power and peak power of the proposed architecture against test pin constrained f.scan.

Module	Full Scan			Partial Scan			Test time red. (%)	Test Power red. (%)	Peak Power red. (%)
	Test Pin Constrained								
	Test Time (ns)	Test Power (W)	Test Peak Power (W)	Test Time (ns)	Test Power (W)	Test Peak Power (W)			
1 × 1	121900	1.12E-06	8.77E-03	29000	7.89E-07	4.80E-03	76.21	29.36	45.3
2 × 2	475800	6.09E-06	2.41E-02	29000	3.16E-06	1.92E-02	93.91	48.14	20.37
3 × 3	1083400	1.42E-05	8.26E-02	29000	7.10E-06	4.32E-02	97.32	49.99	47.72
4 × 4	2074600	2.41E-05	1.51E-01	29000	1.26E-05	7.68E-02	98.6	47.63	49.03
5 × 5	3398700	3.81E-05	2.66E-01	29000	1.97E-05	1.20E-01	99.15	48.2	54.87
6 × 6	6436600	5.98E-05	4.70E-01	29000	2.84E-05	1.73E-01	99.55	52.52	63.24
7 × 7	8915300	8.11E-05	7.42E-01	29000	3.87E-05	2.35E-01	99.67	52.36	68.32
8 × 8	11849400	1.06E-04	1.09E+00	29000	5.05E-05	3.07E-01	99.76	52.36	71.73
16 × 16	14388600	4.30E-04	5.14E+00	29000	2.02E-04	1.23E+00	99.8	53.04	76.09
32 × 32	21794600	1.77E-03	2.18E+01	29000	8.08E-04	4.91E+00	99.87	54.3	77.51

TABLE 3. Results for test time, power and peak power of the proposed architecture against test time constrained f.scan [34].

Module	Full Scan			Test time red. (%)	Test Power red. (%)	Peak Power red. (%)
	Test Time Constrained [34]					
	Test Time (ns)	Test Power (W)	Test Peak Power (W)			
1 × 1	121900	1.12E-06	8.77E-03	76.21	29.36	45.3
2 × 2=4(1 × 1)	121900	4.47E-06	3.50E-02	76.21	29.27	45.14
3 × 3=9(1 × 1)	121900	1.00E-05	7.89E-02	76.21	29.37	45.24
4 × 4=4(2 × 2)/16(1 × 1)	475800/121900	2.434E-05/1.787E-05	9.64E-02/1.40E-01	93.91/76.21	48.23/41.82	20.33/45.26
5 × 5=25(1 × 1)	121900	2.79E-05	2.19E-01	76.21	29.46	45.25
6 × 6=4(3 × 3)/9(2 × 2)/36(1 × 1)	1083400/475800/121900	5.68E-05/5.47E-05/4.02E-05	3.30E-01/2.16E-01/3.15E-01	97.32/93.91/76.21	50/48.14/29.53	47.57/20.23/45.07
7 × 7=49(1 × 1)	121900	5.47E-05	4.29E-01	76.21	29.3	45.22
8 × 8=4(4 × 4)/16(2 × 2)/64(1 × 1)	2074600/475800/121900	9.64E-05/9.737E-05/7.148E-05	6.02E-01/3.85E-01/5.61E-01	98.60/93.91/76.21	47.63/48.13/29.35	49.03/20.38/50.61
16 × 16=4(8 × 8)/16(4 × 4)/64(2 × 2)	11849400/2074600/475800	4.24E-04/3.85E-04/3.71E-04	4.34/2.4E-01/1.54E-01	99.76/98.60/93.91	52.36/47.62/48.07	71.68/48.75/20.12
32 × 32=16(8 × 8)/64(4 × 4)/256(2 × 2)	11849400/2074600/475800	1.69E-03/1.54E-03/1.48E-03	17.36/9.63E-01/6.16	99.76/98.60/93.91	52.35/47.53/48.13	71.80/49.01/20.29

reliability of the hardware. For full scan arrays 16 × 16 and 32 × 32, multiple scan chains do not alleviate test power and peak power consumption, mainly because of additional scan chains (compared to 8 × 8) and increased power consumption in combinational logic. While for proposed partial scan-based arrays there is improvement over their full scan counterparts. The total power overhead of the test pass/fail logic for the whole duration of pattern application in our proposed architecture for a 32 × 32 array is 4.6%.

E. TEST POWER AND TEST TIME FOR AT-SPEED TESTING

For at-speed testing, the proposed architecture is simulated on Modelsim with its custom flow for transition patterns (vector pairs). From Tetramax ATPG, 16 transition vector pairs were generated for datapath combinational logic. Test power for customized pattern flow for various array size is estimated with Prime Power from the associated VCD file, given in Table 4. In full scan based arrays, one scan chain per PE is synthesized to restrict the delay-based testing to single PE, as done for the proposed architecture. The full scan delay testing is done with LoC method. The number of patterns with array size increases as shown in Table 4. This increase in number of patterns with increasing array size results in increasing test power and test time for full scan. Since with the proposed partial scan based architecture, the number of patterns is fixed i.e. 16, it results in increasing test time improvement. While maintaining advantage over the shift power, as (per PE) only half of the scan elements are shifting the patterns, when compared to full scan DFT.

F. CHECKERBOARD FULL SCAN TEST METHOD

The proposed partial scan method is also evaluated in parallel with the checkerboard method [16]. A 32-bit partial sum register based TPU model is considered for proposed partial scan method because the checkerboard method uses the 32-bit partial sum register. ATPG was performed for stuck-at and

TABLE 4. Results for test time and test power for at-speed testing against f.scan LoC test.

Module	Partial Scan		Full Scan			Test Power Red. (%)	Test Time Red. (%)
	Test Power (W)	Test Time (ns)	Test Patterns	Test Power (W)	Test Time (ns)		
1 × 1	9.49E-08	27232	54	1.41E-06	181608	93.29	85.01
2 × 2	3.85E-07	27264	58	4.98E-06	194816	92.25	86.01
3 × 3	8.50E-07	27296	88	7.69E-06	293876	88.94	90.71
4 × 4	1.50E-06	27328	90	1.24E-05	300480	87.92	90.91
5 × 5	2.32E-06	27360	98	1.87E-05	326896	87.56	91.63
6 × 6	3.33E-06	27392	116	2.76E-05	386332	87.93	92.91
7 × 7	4.51E-06	27424	121	3.60E-05	402842	87.45	93.19
8 × 8	5.88E-06	27456	133	4.30E-05	442466	86.32	93.79

TABLE 5. Comparison with checkerboard [16].

Module	Checkerboard Test [16] (SAF)			Partial Scan (SAF)			Checkerboard Test [16] (TF)			Partial Scan (TF)		
	no. of Patterns	T.C	Test Cycles	no. of Patterns	T.C	Test Cycles	no. of Patterns	T.C	Test Cycles	T.C	no. of Patterns	Test Cycles
8 × 8	32	93.3	2560	15	100	527	41	71.5	3280	85.7	17	848
16 × 16	32	96.6	2560	15	100	527	41	79.1	3280	85.7	17	1120
32 × 32	32	98.3	2560	15	100	527	41	83	3280	85.7	17	1664
64 × 64	32	99.2	2560	15	100	527	41	84.9	3280	85.7	17	1664
128 × 128	32	99.6	2560	15	100	527	41	85.9	3280	85.7	17	1664
256 × 256	32	99.8	2560	15	100	527	41	86.4	3280	85.7	17	1664

transition faults for this 32-bit model. Since the ATPG effort is limited to Datapath logic, So the number of patterns is less than the checkerboard method. Also, the PE has only partial sum register as scan register, it results in a smaller number of test cycles. The proposed partial broadcasting method allows the test time improvement for arrays larger than 32×32 (as the scan chain length is 32 now) as mentioned in section V-B. The results are shown in Table 5.

VIII. CONCLUSION

Regardless of the application area of an electronic system, test cost/overhead presents a major design problem due to its implications on the overall system cost and operation. Implementing de facto test techniques such as full scan DFT may not yield a cost-effective solution for overhead constrained edge computing devices. In this paper, an efficient and scalable test solution is proposed for weight-stationary systolic array for an edge-based AI hardware. The proposed architecture addresses the testability on PE level of the whole array. This architecture specific solution leads to an efficient testing approach. Due to improvement of test time and test power with increasing array size, this architecture is also well-suited for large-scale accelerators of Clouds/Datacenters.

REFERENCES

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

[2] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Gener. Comput. Syst.*, vol. 97, pp. 219–235, Aug. 2019, doi: 10.1016/j.future.2019.02.050.

[3] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017, doi: 10.1109/JPROC.2017.2761740.

[4] F. Wang, M. Zhang, X. Wang, X. Ma, and J. Liu, "Deep learning for edge computing applications: A state-of-the-art survey," *IEEE Access*, vol. 8, pp. 58322–58336, 2020, doi: 10.1109/ACCESS.2020.2982411.

[5] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019, doi: 10.1109/JPROC.2019.2921977.

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

[7] Google. (2019). *Edge TPU—Run Inference at Edge*. [Online]. Available: https://cloud.google.com/edgetpu/

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

[9] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, N. Boden, A. Borchers, and R. Boyle, "In-datacenter performance analysis of a tensor processing unit," in *Proc. 44th Annu. Int. Symp. Comput. Archit.*, Toronto, ON, Canada, Jun. 2017, pp. 1–12, doi: 10.1145/3079856.3080246.

[10] 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, doi: 10.1109/JETCAS.2019.2910232.

[11] T. Chen, Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Temam, "DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning," in *Proc. ASPLOS*, 2014, pp. 269–284.

[12] 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*, Boston, MA, USA, Jul. 2009, pp. 53–60, doi: 10.1109/ASAP.2009.25.

[13] 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.

[14] 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.

[15] 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)*, San Francisco, CA, USA, Apr. 2018, pp. 1–6, doi: 10.1109/VTS.2018.8368656.

[16] A. Chaudhuri, C. Liu, X. Fan, and K. Chakrabarty, "C-testing of AI accelerators*," in *Proc. IEEE 29th Asian Test Symp. (ATS)*, Nov. 2020, pp. 1–6, doi: 10.1109/ATS49688.2020.9301581.

[17] 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, doi: 10.1109/FOCS.1967.33.

[18] A. D. Friedman, "Easily testable iterative systems," *IEEE Trans. Comput.*, vol. C-22, no. 12, pp. 1061–1064, Dec. 1973, doi: 10.1109/T-C.1973.223651.

[19] C.-H. Sung, "Testable sequential cellular arrays," *IEEE Trans. Comput.*, vol. C-25, no. 1, pp. 11–18, Jan. 1976, doi: 10.1109/TC.1976.5009199.

[20] 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.

[21] F. Lombardi, "On a new class of C-testable systolic arrays," *Integration*, vol. 8, pp. 269–283, Dec. 1989, doi: 10.1016/0167-9260(89)90020-5.

[22] 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, doi: 10.1109/ESSCIRC.1985.5468108.

[23] 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, doi: 10.1109/92.365462.

[24] D. Gizopoulos, A. Paschalis, and Y. Zorian, "An effective built-in self-test scheme for parallel multipliers," *IEEE Trans. Comput.*, vol. 48, no. 9, pp. 936–950, Sep. 1999, doi: 10.1109/12.795222.

[25] G. Giles, J. Wang, A. Sehgal, K. J. Balakrishnan, and J. Wingfield, "Test access mechanism for multiple identical cores," in *Proc. Int. Test Conf.*, Austin, TX, USA, Nov. 2009, pp. 1–10, doi: 10.1109/TEST.2009.5355560.

[26] T. Han, I. Choi, and S. Kang, "Majority-based test access mechanism for parallel testing of multiple identical cores," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 8, pp. 1439–1447, Aug. 2015, doi: 10.1109/TVLSI.2014.2341674.

[27] M. Cheong, I. Lee, and S. Kang, "A test methodology for neural computing unit," in *Proc. Int. SoC Design Conf. (ISOCC)*, Daegu, South Korea, Nov. 2018, pp. 11–12, doi: 10.1109/ISOCC.2018.8649896.

[28] 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, doi: 10.1109/ITC-Asia.2019.00024.

[29] A. Chaudhuri, J. Talukdar, F. Su, and K. Chakrabarty, "Functional criticality classification of structural faults in AI accelerators," in *Proc. IEEE Int. Test Conf. (ITC)*, Nov. 2020, pp. 1–5, doi: 10.1109/ITC44778.2020.9325272.

[30] J. Ross, N. Jouppi, A. Phelps, R. Young, T. Norrie, G. Thorson, and D. Luu, "Neural network processor," U.S. Patent 9 747 546 B2, May 21, 2015.

[31] J. Ross and A. Phelps, "Computing convolutions using a neural network processor," U.S. Patent 9 697 463 B2, May 21, 2015.

[32] J. Ross, "Prefetching weights for use in a neural network processor," U.S. Patent 9 805 304 B2, May 21, 2015.

[33] J. Ross and G. Thorson, "Rotating data for neural network computations," U.S. Patent 9 747 548 B2, May 2015.

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

UMAIR SAEED SOLANGI received the bachelor's degree in electronic engineering and the master's degree in embedded systems from Mehran University, Pakistan. He is currently doing Ph.D. research in the field of design for testability with Hanyang University, ERICA, South Korea. He is also an Assistant Professor with the Department of Electronic Engineering (at a public sector university), Pakistan. Other research interests include embedded systems, low power design, and digital logic design. He was a recipient of the Ph.D. Scholarship by the Higher Education Commission, Pakistan.

MUHAMMAD IBTESAM received the B.Sc. degree in electrical engineering from the University of Engineering and Technology at Taxila, Taxila, Pakistan. He is currently pursuing the combined M.S. and Ph.D. degree in computer science and engineering with Hanyang University. His research interests include the design for testability (DFT), low power 3-D IC/SiP testing and low power TAM designs for AI accelerators. He was a recipient of M.S.-Ph.D. Scholarship by the Higher Education Commission, Pakistan.

MUHAMMAD ADIL ANSARI received the B.E. degree in electronic engineering from the Mehran University of Engineering and Technology (UET), Pakistan, in 2006, and the M.S. and Ph.D. degrees in computer science and engineering from Hanyang University, South Korea, in 2010 and 2016, respectively. He worked as an Operations Engineer with Pakistan Telecommunication Company Ltd., from 2006 to 2008, and served as a Lecturer for the COMSATS Institute of Information Technology, Pakistan, from 2010 to 2011. He is currently with Quaid-e-Awam University, Pakistan, as an Assistant Professor, from 2011 to 2018, where he has been an Associate Professor, since 2018. His research interests include design-for-testability of digital stacked and non-stacked integrated circuits.

JINUK KIM received the B.S. degree in computer science and engineering from Hanyang University, South Korea, in 2015, where he is currently pursuing the combined M.S. and Ph.D. degree in computer science and engineering. His research interests include design-for-testability (DFT), memory ECC, memory test, and 3-D IC / SiP (system-in-package) testing.

SUNGJU PARK (Senior Member, IEEE) received the B.S. degree in electronic engineering from Hanyang University, South Korea, in 1983, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Massachusetts, USA, in 1988 and 1992, respectively. From 1983 to 1986, he was with Gold Star Company, South Korea. From 1992 to 1995, he served for IBM Microelectronics, Endicott, NY, USA, as a Development Staff, in-charge of boundary scan and LSSD scan design. Since 1995, he has been a Professor with the Department of Computer Science and Engineering, Hanyang University. His research interests include the area of VLSI testing, including scan design, built-in self-test, test pattern generation, fault simulation, and synthesis of test. Additional interests include graph theory and design verification. He is a member of the Institute of Electronics Engineers of Korea, the Korea Information Science Society, and the Institute of Electronics and Information and Communication Engineers.