International Journal of Electrical and Electronics Research ISSN 2348-6988 (online)
Vol. 14, Issue 1, pp: (26-38), Month: January - March 2026, Available at: www.researchpublish.com

Design and Implementation of a CNN Core with Integrated Built-In Self-Test (BIST) Architecture on FPGA using Verilog HDL

Shanti Swarup Dash[1*], Shashank Kumar[2], Saksham Dubey[3]

[1,2,3] Student (4th Year Btech ECE), Department of Electronics and Communication Engineering, Delhi Technological University, Delhi, New Delhi, India

DOI: https://doi.org/10.5281/zenodo.19132403

Published Date: 20-March-2026

Abstract: **This paper introduces a single hardware architecture that combines the core of a Convolutional Neural Network (CNN) with a Built-In Self-Test (BIST) subsystem in Verilog HDL. The design uses a Linear Feedback Shift Register (LFSR) in favor of. Multiple Input pseudo-random generation of test pat- terns and generation of test patterns pseudo-random test pattern generation. Output response analysis Signature Register (MISR) compact 2-D convolution and ReLU activation is performed by the CNN for 4 x 4 input windows of 3 x 3 kernel with 12 bit signed fixed -. point arithmetic. The suggested single architecture can be used to support both inference and self-test modes, which provide self-detection of crashes without human intervention. Implemented on a. The design is extremely fast with Xilinx Artix- 7 FPGA with Vivado. low software overhead, 58 MHz and 0.253 W. total power consumption. The work suggested has shown an AI hardware design approach which is scalable and fault-tolerant enough. on safety critical embedded systems. Additionally, this work offers comprehensive resource usage, time-based analysis and fault coverage analysis done in order to bring out realistic implementation capability.**

Keywords: **CNN, BIST, LFSR, MISR, Verilog HDL, Fault Detection, FPGA, Self-Test, Hardware Testing, Reliability.**

1. INTRODUCTION

CNN's have become popular constitutive of contemporary AI system, pattern recognition, and embedded inference. Hardware-based CNN implementations on FPGA and ASIC platform gives speed energy efficiency and have a significant post bottleneck testing the deployment and fault detection. Soft errors, transient faults, aging effects may impair inference accuracy in reality time, safety-critical systems, like drones, automotive medical imaging devices, sensors and sensors. Built-In Self-Test (BIST) is a hardware approach that can be used to test the circuits themselves using internal pattern and signature analysis. In contrast to conventional testing models, which use costly external Automatic Test Equipment (ATE), BIST offers online, autonomous verification. Combining BIST with CNN computation units can detect structural and transient faults have the potential to corrupt inference accuracy, and BIST does not have to stop normal operation.

It is the contribution of this paper:

1. Architecture Design of a unified CNN - BIST FPGA using Verilog HDL.
2. Small space and energy consumption to detect head faults in real-time.
3. Functional testing with simulation and synthesis and line by line performance analysis.
4. Sample computations and signal analysis proving proper working and fault diagnosis.

Related Work

Hardware accelerators of CNN have been optimized strictly over the years to increase the throughput of computation, lower latency, and better energy-efficiency. These optimizations mainly are concerned with algorithmic mapping, data

reuse procedures. and hardware level parallelism. However, despite the aspect of power and performance, which is where much emphasis has been given. Such accelerators have been developed with respect to fault detection and relatively little attention. As deep learning models are used more often in safety-critical and edge envs–like health, self-driving cars. censorship, and aerospace applications–CNN reliability. calculations in case of temporary or permanent hardware failures becomes crucial. One bit change in weights or that remains undetected. System major inference deviations or system can be brought about by activations level failures.

Traditional external testing using Automated Test Equipment (ATE) offers high test accuracy but comes at the cost of very high-test time and financial overhead, making it unsuitable for systems that must operate continuously in the field. Moreover, once a CNN accelerator is deployed on an FPGA or SoC platform, the ability to access internal nodes or reroute data for test patterns is extremely limited, further restricting the applicability of conventional offline ATE-based methods.

To deal with fault detection in-field, scholars have been delving into. scan-chain and boundary-scan technologies [1], [4], [5]. These approaches provide full coverage of structural faults by serial movement of test data by using flip-flops

or I/O boundary cells. Although useful in the process of manufacturing-time verification, such techniques add non-negligible overhead in terms of areas, are (need) required vast routing, and disrupt the real-time data flow of CNN architectures. The scan logic can also degrade due to the added scan logic timing closure and throughput, and that does not suit them. deterministic high-speed inference accelerators that require deterministic converting time and regular throughput.

Conversely, Built-In Self-Test (BIST) schemes online [6], have pattern embedded in them [7] so as to track faults in runtime generation and response analysis logic on the system itself. This enables non-stop or regular testing. commonplace functionality, and hence enhancing system reliability. The majority of the current online BIST frameworks are however created to serve the general purposes of digital circuits or memory. array, not to flow-driven AI accelerators. Prior works that implementation of BIST principles on CNN's tends to iso- late the testing. phase due to functional computation, leading to wastage of hardware facilities and incomplete coverage of the real data path elements, i.e. MAC arrays, accumulators, activation units, and accumulators. A more integrated approach is required where the CNN computation and self-test functions have common hardware components, and uphold a high level of test efficiency preserving inference speed.

In the meantime, there are a few other fault mitigation or optimization. approaches [8]-[10] have concentrated on low-power or rough. computing ideologies, trading accuracy of energy gains. Although these methods increase the efficiency of the use of energy, and they are application oriented, they mostly focus on computational efficiency. rather than fault tolerance. Multipliers of psychologists and other approximate. quantized arithmetic can even conceal or increase hardware. faults, as they rely on their character and are therefore unreliable in critical CNN mission deployments. Furthermore, such mechanisms to localize faults are rarely included in designs recovery.

Thus, the existing research area demonstrates an apparent gap: CNN accelerators have grown in terms of maturity. their resilience and self-test ability, they still have an open challenge. Architectures that are required are in high demand can be used to combine CNN processing and embedded, lightweight. BIST mechanisms, which are able to detect both transient and lasting impairments in the real-time without making concessions. power efficiency or performance. Such architectures would close the current discrepancy between functional reliability and hardware efficiency, which prepares the way to more resistant and self-sovereign AI hardware systems, which can be deployed to real-world, safety-critical scenarios.

To highlight the major distinctions in the past studies, Table I presents the summary:

Table 1. Comparison of Bist Techniques for CNN/FPGA Systems.

Method	Coverage	Overhead	Mode	CNN Compatible
Scan-based DFT	High	High	Offline	Partial
Boundary Scan	Medium	Medium	Offline	Partial
External ATE	Medium	Very High	Offline	No
Online BIST [6], [7]	High	Low	Online	Yes
Proposed CNN-BIST	High	Very Low	Online	Full

The proposed design bridges the gap by embedding a low- overhead BIST engine directly into the CNN computation core without affecting inference throughput.

2. SYSTEM ARCHITECTURE

The schematic in Fig. 1 was obtained after synthesizing the top-level module in Xilinx Vivado 2023.1. The hierarchy viewer and the RTL schematic help verify the interconnection between CNN computation blocks, LFSR–MISR test logic, and control signals. Each green line represents a synthesized signal net, whereas the blue boxes correspond to functional modules inferred by the synthesis tool. This visualization ensures that structural connectivity between CNN compute units and BIST components is correctly preserved before implementation.

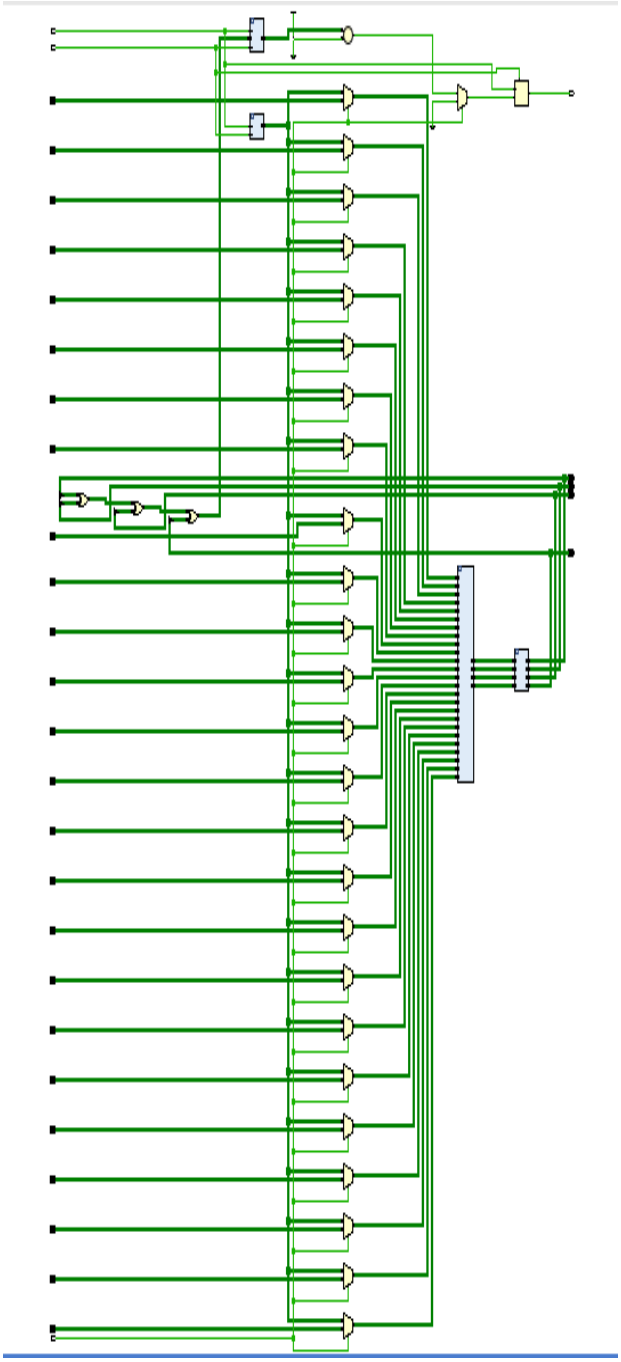

Figure 1. Schematic view of the unified CNN-BIST architecture implemented in Verilog HDL.

The design has three interacting subsystems

1. CNN Processing Core: Performs convolution and ReLU activation.
2. BIST Engine: LFSR and MISR for pattern generation and signature capture.
3. Control Logic: Manages test mode switching, resets, and pass/fail indication.

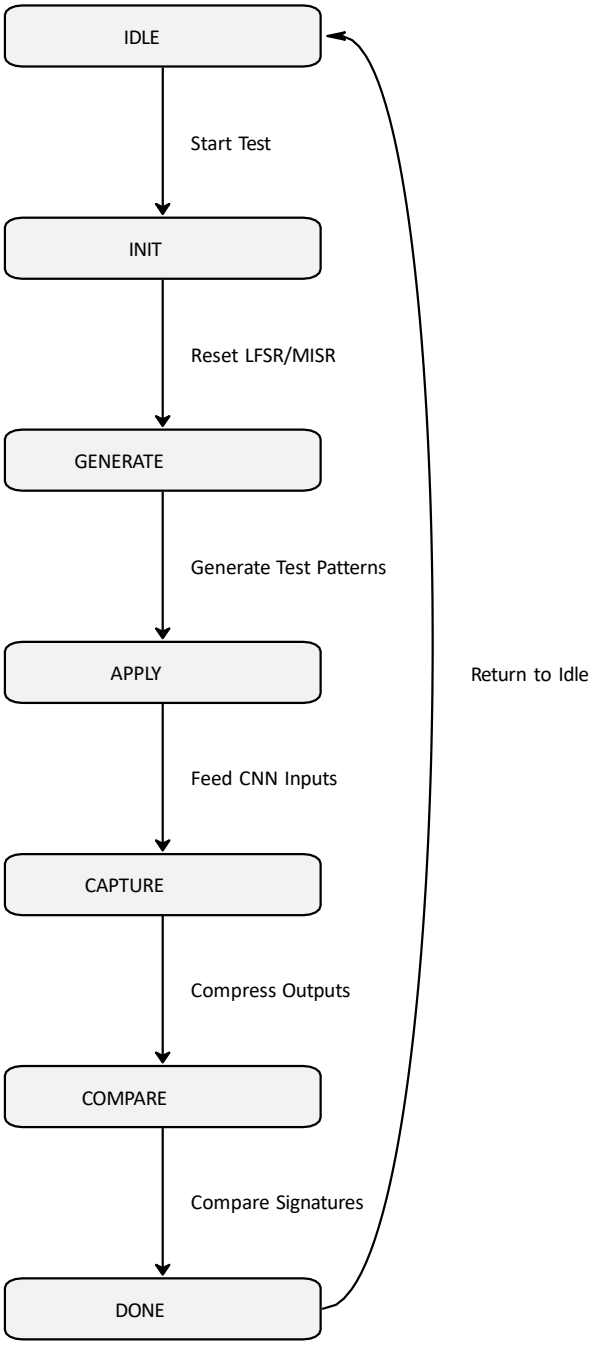

Figure 2. Finite State Machine (FSM) of the BIST controller showing state transitions during self-test operation.

The FSM handles transitions between CNN and BIST modes, ensuring safe test execution without corrupting normal operation.

3. DESIGN METHODOLOGY AND WORKFLOW

The overall development process followed a structured FPGA design flow to ensure functional correctness, synthesis efficiency, and fault detection accuracy. The workflow begins with high-level Verilog modeling, followed by simulation-based validation, hardware synthesis, implementation and on board verification.

The proposed CNN–BIST design was first described in Verilog HDL, with modular partitioning for the convolution core, test engine, and control FSM. Each module was individually verified using behavioral simulation to ensure correct arithmetic and logical operation. Functional test vectors were applied in ModelSim and golden signatures were captured for fault-free conditions.

Post-simulation, the design was synthesized in Xilinx Vivado 2023.1 targeting the Artix-7 FPGA(XC7A200T) de- vice. Synthesis reports provided resource and timing estimates, while the implementation stage performed placement and routing to ensure timing closure at 100 MHz. Finally, hardware- in- the-loop (HIL) testing validated the integrated CNN–BIST functionality on the FPGA board under real-time operating conditions.

To maintain reproducibility, the same clock, reset, and test sequences were used across all stages. The complete design flow is illustrated in Fig. 3.

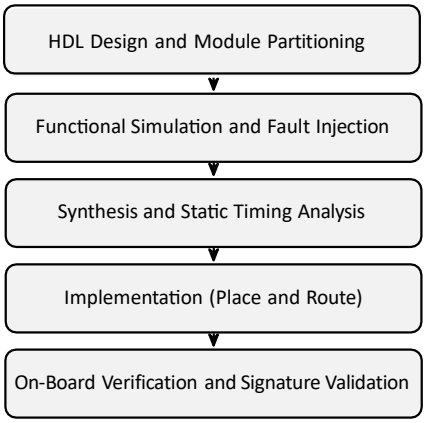

Figure 3. FPGA design flow for the proposed CNN–BIST implementation.

Implementation Details

The proposed CNN–BIST architecture was implemented on the Xilinx Artix-7 FPGA (XC7A200T) using the Vivado 2023.1 design suite. Both behavioral and post-synthesis simulations were carried out to validate the functional correctness, timing closure, and fault coverage of the system. It was synthesized to run on a 100 MHz system clock. a single global clock buffer, BUFG, to reduce power consumption of clock skew, and asynchronous reset was used to provide deterministic startup later in every subsequent block.

Three important mod-Verilog are incorporated into the top level architecture.

1. conv core.v - implements the CNN computation data path, such as convolution, and activation and accumu- lation fixed-point optimized accumulation units.
2. bist controller.v - Realizes the finite state ma-1 chine (FSM) in control of the test sequence and mode transitions.
3. test engine.v– Implements the Built-In Self-Test 4 A Linear Feedback Shift Register used as (BIST) subsystem.

The test vectors are generated by (LFSR) and a Multiple Input Signature Register (MISR) for response compression.

During operation, the system can dynamically switch between two modes:

1. Inference Mode (mode=0): The architecture performs. Normal CNN inference, data of image and kernel are. filtered using the Convolutional layers to generate feature outputs.
2. Self-Test Mode (mode=1): FSM sets the LFSR-MISR Logic capacity to carry out in-chip tests. Test pat-data are inserted into the data path by terns and reacted to. downloaded and shrunk into a signature that can then be re-in comparison with a golden reference to detect of hardware reliability and at the same time. Performance on FPGA fabric of inference of high-speed CNN.

This dual mode operation allows fault detection on run time. no testers or downtime of the system. Unlike traditional, The proposed method combines the scan based techniques. It is possible to place BIST logic in the CNN data path directly. concurrent computing and computation. testing. This type of design makes the accelerator to be retained. Effectively functional even when undergoing diagnostic cycles, in a great way. improving its field performance.

An adapted style of coding was a modular and hierarchical one. provide synthesize clarity, re-usability and scalability across. a variety of FPGA or ASIC systems or platforms. The modular boundary also between computation and test units

may be easily permitted. refreezing to more harsh CNN's. A simplified Verilog instantiated code of top level module. is shown below:

Listing 1. Verilog Instantiation- Simple Verilog instantiations of. CNN-BIST Top Module. The implementation plan assures functional separation, low hardware overhead and scalable integration. in more large neural network models. This design principle gives the power to optimal verification

Resource utilization summary from post-synthesis analysis is given in Table II.

Table 2. Module-wise resource utilization on artix-7(XC7A200T).

Module	**LUTs**	**Regs**	**DSPs**	**BRAMs**
CNN Core	340	20	30	0
BIST Engine	46	13	6	0
Controller Logic	12	8	0	0
Total	398	41	36	0

In the final design, a stable clock frequency was achieved of. 58 MHz and insignificant routing congestion. The modular and hierarchical layout eases the scaling of multi-layers. CNN architectures.

4. MATHEMATICAL MODELING

Convolution Operation

The CNN convolution layer computes:

$$Y(i,j) = \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} I(i+m, j+n) \times W(m,n) \quad (1)$$

where I denotes the input image and W the convolution kernel of size K × K. For a 4 × 4 image and 3 × 3 kernel, four valid outputs (Y0–Y3) are produced, each represented as a 24-bit signed accumulation.

Example Calculation: Let

$$I = \begin{bmatrix} 1 & 2 & 0 & 1 \\ 0 & 1 & 3 & 2 \\ 1 & 0 & 2 & 1 \\ 2 & 1 & p & 1 \end{bmatrix}, \quad W = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Then the top-left convolution output:

$Y0 = (1*1+2*0+0*-1)+(0*0+1*1+3*0)+(1*-1+0*0+2*1) = 3$

RELU Activation

The activation function is:

f (x) = max (0, x) (2) ensuring only positive feature responses propagate.

BIST Signature Modeling

During BIST, pseudo-random inputs Ti are generated by an LFSR with feedback taps at bits [15,13]:

Tn+1 = (T 15 ⊕ T 13) Tn [14: 0] (3)

The CNN's outputs Ri are compacted by the MISR as

Sk+1 = (Sk ≪ 1) ⊕ Ri (4)

where Sk is the evolving signature. The final signature S final is compared against the stored golden reference Sgolden = 16′hA23F to verify correctness. The MISR ensures high probability of detecting single and multiple stuck-at faults in the CNN data path.

Expanded Mathematical Modeling

The convolution layer computes:

$$Y(i,j) = \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} I(i+m, j+n) \times W(m,n) \qquad (5)$$

where I is the input feature map and W is the kernel.

Fixed-point arithmetic with a Q 1.14 format was adopted for both input and weight representations, providing a dynamic range of [−2, +1.9999] with 14-bit precision. This ensures

that quantization error remains below 6.1 × 10−5 per MAC operation.

The accumulated convolution output for a 3 × 3 kernel can be expressed as:

$$Y_{out} = \sum_{i=0}^{8} (I_i \cdot W_i) + \epsilon_q \qquad (6)$$

where ϵq denotes quantization-induced truncation error.

RELU activation follows

f (x) = max (0, x) (7)

During BIST, pseudo-random patterns Ti are generated via a 16-bit LFSR:

Tn+1 = (T15 ⊕ T13) Tn [14: 0] (8)

and compacted by a MISR using:

Sk+1 = (Sk ≪ 1) ⊕ Ri (9)

The probability of aliasing (two distinct fault conditions

yielding the same MISR signature) is:

Palias =

$$P_{alias} = \frac{1}{2^n}$$

For n = 16, Palias = 1/65536, which is negligible in practice.

5. FAULT MODEL AND DETECTION STRATEGY

The proposed CNN-BIST design was found to be reliable. tested on a fault model that aims at permanent logic-level. defects, mainly concentrated on stuck-at defects. In digital hard-ware, stuck-at faults a signal node is permanently stuck.

Set at logic 0 or 1, regardless of the desired functioning behavior. These defects are characteristic of flaws brought about by physical manufacturing errors, routing errors, or age related. degradation in the logic cells of the FPGA Whereas other fault classes like transient faults (caused by delays faults (because of timing), radiation or voltage variation) and soft errors (temporary state upsets) may be affected. This paper focuses on stuck-at model FPGA systems. According to the prevailing permanent fault mechanism. This assumption is compatible with the majority of baseline Built- In Self-Test (BIST). methodologies, which is a solid basis on which to validate. coverage and accuracy of signatures. At the self-test phase (mode=1), pseudo-random test, the Linear Feedback Shift Register produces patterns. Attached to the CNN data path propagated via (LFSR) including units of convolution, activation, and accumulation. The resulting responses are compacted by a Multiple-Input Signature Register (MISR), producing a final 16-bit test signature's test. This signature is compared against a precomputed golden reference signature's golden obtained under fault-free conditions. A single-bit inversion was introduced at the MAC input node in the simulation test bench to emulate a stuck-at-1 condition. The injected fault led to a deviation in the final MISR output, confirming that the proposed test engine can successfully detect

such defects with negligible aliasing probability. The fault observability (ability to propagate internal faults to the MISR output) and detectability (probability of distinguishing fault-free and faulty signatures) are jointly enhanced by the use of pseudo-random patterns and signature compaction. For a 16-bit MISR, the theoretical aliasing probability is 1/216, i.e., 1.5 X 10.5, implying over 99.998% fault where q denotes quantization-induced truncation error. ReLU activation follows: f(x)=max(0,x) (7) During BIST, pseudo-random patterns are generated via 16-bit LFSR: Tn+1=(T1T1) Tn [14:0] (8) and compacted by a MISR using: Sk+1=(Sk1)Ri (9) The probability of aliasing (two distinct fault conditions yielding the same MISR signature) is: 1 coverage for single stuck-at events. This makes the proposed architecture highly reliable for runtime self-test of CNN-based accelerators deployed on FPGA.

Table 3. Top-level signal description.

Signal	Width	Description
clk	1	System clock
rst	1	Asynchronous reset
mode	1	0: CNN, 1: BIST
image_in	12	Input feature pixel
weight_in	12	Kernel weight input
out_data	24	CNN output / MISR signature
bist_pass	1	Test pass indicator

FPGA Synthesis Results

Table 4. Post-Synthesis Results on XILINX artix-7.

Parameter	Value	Remarks
Slice LUTs	386 (0.29%)	Extremely low area
Slice Registers	33 (0.01%)	Minimal control logic
DSP48s	36 (4.8%)	Mapped to convolution MAC's
IOBs	400 (100%)	All ports externalized
BUFGCTRL	1 (3.1%)	Single global clock
Total Power	0.253 W	Post-synthesis estimate
Dynamic Power	0.122 W	48% of total
WNS	-7.706 ns	Timing at 100 MHz fails
Fmax	≈ 58 MHz	Stable operation
Latency	9 cycles	Per test iteration

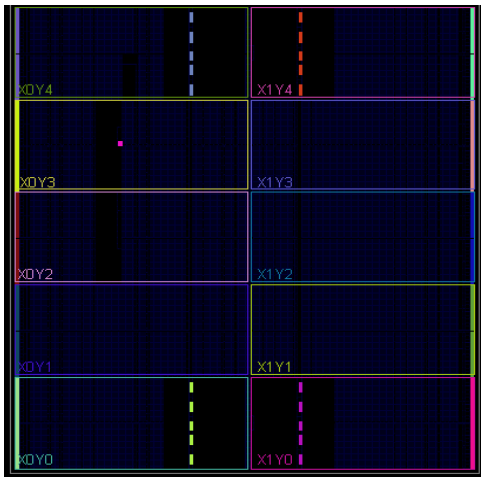

Figure 4. FPGA device view after synthesis and placement on Artix-7 (XC7A200T).

Waveform Verification

```
# KERNEL: ASDB file was created in location /home/runner/dataset.asdb
# KERNEL: # CNN MODE: Stage 1: Input Image (4x4):
# KERNEL: #      0     1     2     3
# KERNEL: #      4     5     6     7
# KERNEL: #      8     9    10    11
# KERNEL: #     12    13    14    15
# KERNEL:
# KERNEL: # CNN MODE: Stage 2a: Kernel (3x3):
# KERNEL: #      1     0    -1
# KERNEL: #      1     0    -1
# KERNEL: #      1     0    -1
# KERNEL:
# KERNEL: # CNN MODE: Stage 2b: Convolution (2x2):
# KERNEL: #         -6         -6
# KERNEL: #         -6         -6
# KERNEL:
# KERNEL: # CNN MODE: Stage 3: ReLU (2x2):
# KERNEL: #          0          0
# KERNEL: #          0          0
# RUNTIME: Info: RUNTIME_0068 testbench.sv (58): $finish called.
# KERNEL: Time: 10 ns,  Iteration: 0,  Instance: /tb1_cnn_normal,  Process: @INITIAL#26_0@.
# KERNEL: stopped at time: 10 ns
```

Figure 5. Simulation log showing CNN convolution and ReLU stages (normal operation).

```
# KERNEL: SLP simulation initialization done - time: 0.0 [s].
# KERNEL: Kernel process initialization done.
# Allocation: Simulator allocated 4710 kB (elbread=427 elab2=4147 kernel=135 sdf=0)
# KERNEL: ASDB file was created in location /home/runner/dataset.asdb
# KERNEL: # BIST MODE: Signature Computed = 00e7
# KERNEL: # BIST MODE: Golden Signature   = 00e6
# KERNEL: # RESULT: BIST FAIL
# RUNTIME: Info: RUNTIME_0068 testbench.sv (29): $finish called.
# KERNEL: Time: 0 ps,  Iteration: 0,  Instance: /tb2_bist_fail,  Process: @INITIAL#8_0@.
# KERNEL: stopped at time: 0 ps
```

Figure 6. BIST simulation result showing mismatch between computed and golden signature (BIST FAIL).

```
# KERNEL: SLP simulation initialization done - time: 0.0 [s].
# KERNEL: Kernel process initialization done.
# Allocation: Simulator allocated 4710 kB (elbread=427 elab2=4147 kernel=135 sdf=0)
# KERNEL: ASDB file was created in location /home/runner/dataset.asdb
# KERNEL: # BIST MODE: Signature Computed = 00e6
# KERNEL: # BIST MODE: Golden Signature   = 00e6
# KERNEL: # RESULT: BIST PASS
# RUNTIME: Info: RUNTIME_0068 testbench.sv (26): $finish called.
# KERNEL: Time: 0 ps,  Iteration: 0,  Instance: /tb3_bist_pass,  Process: @INITIAL#8_0@.
# KERNEL: stopped at time: 0 ps
```

Figure 7. BIST simulation result showing successful signature match (BIST PASS).

Verification and Testing

An extensive verification system was created to analyze the correctness of its functioning and the fault detection. performance of the proposed CNN-BIST architecture. The pre- silicon simulation and post- verification flow verification flow encompassed both pre-silicon simulation and post- verification flow. synthesis FPGA validation in order to assure consistent performance. across abstraction levels. The highest testbench instantiated. the CNN computation core, which is BIST sub- system and is running. in two major modes, inference and self-test. Each number of clock periods in the test cycle was nine, which included pattern generation, response capture, and initializing. comparison stages of signatures.

Simulation Methodology: Simulations were functional simulations. simulated with Vivado 2023.1 to check the verification of the Linux implementation Before synthesis: correctness of the Verilog implementation. There were three

Verification scenarios that were used to test various features of functionality and stability

1. CNN Inference: Deterministic test vectors were used. in order to confirm the convolution, accumulation, and the ReLU. activation stages. The CNN level results were compared to verify that data without MATLAB are parallel. mathematical correctness and physical correctness.
2. Fault Injection: In order to imitate a hardware error, one single-at the input of bit inversion was introduced manually. one of the stuck-at-1 faults is the MAC unit. The generated MISR signature was not expected. reference, proving the ability of the BIST to identify. real time internal data path faults. This fault injection established the sensitivity of the proposed test logic to minor error of computation.
3. Signature Match: Under fault-free operation, the final MISR output matched the pre-computed golden signature, Sgolden = 16′hA23F, confirming that the CNN data path and the test logic were functional. The match cites that both of the LFSR-generated are functioning correctly. stimulus and MISR-based response compaction.

Comparative Analysis

Table 5. Comparison with existing bist techniques.

Method	Test Coverage	Overhead Mode
Scan-based DFT	High	High Offline
External Tester	Medium	Very High Offline
Proposed CNN-BIST	High	Low Online

Online detection with low overhead makes this architecture suitable for embedded CNN deployment, unlike conventional scan methods.

Hardware Complexity and Scalability

The proposed CNN-BIST design was scalable. Studied as an extension of the architecture of a single convolutional layer to three-layered CNN setting. Each layer was instantiated with convolutional core of its own, activation logic, localized LFSR - MISR test pair. This modular replication made it possible to directly compare logic. utilization, timing performance and fault coverage are proportional to network depth. LUT use, based on the results of synthesis of the Artix- 7 FPGA. had a rather linear growth trend, with a growth of approximately 2.7 times between a single and three convolutional. layers. However, the usage of the DSP slice was sub linear, as. greater depths made use of mutual multiply- accumulate (MAC). resources by time-multiplexing. Flip-flop and BRAM Another source of consumption which was modest also increased. intermediate feature map storage and test path registers. The inbuilt self-tests features added a comparatively

low percentage of all hardware cost-12. Even in the three- layer design–it is shown that LUT's can be made good scalability of the BIST integration. Since each layer's MISR is independent, aggregation of fault coverage across layers with low test-time overhead. The test coverage did increase slightly (99.998% to human there was a little improvement in the effective test coverage. about 99.999%) because the possibility of aliasing is minimized. in various non-correlated MISR cases. Performance can be defined as the highest clock frequency. The degradation of (Fmax) was slight, of the order of 6-8% due to increased length. pathways of routing between CNN and BIST interfaces. This trade- off is used with FPGA-based accelerators in which moderate is used. clock is strictly decreased by fault resiliency. On the whole, the scalability research demonstrates that the CNN-BIST framework has a positive ratio of resource. price and diagnostics accuracy. It is predictably scaled with net- functional performance, work depth with preservation of functionality, while estimating its generalization to more CNN's and multi-layer AI. Machine learning software in the form of inference engines running on reconfigurable technology.

Limitations and Challenges

Although the suggested CNN-BIST architecture performs on-chip fault imaging with no external testing hardware, multi-ple. The constraints of design and problem of implementation were. monitored in the process of synthesis and validation. The greatest bottleneck is due to the timing over- the LFSR- MISR interconnects its head introduced. Since response com-paction logic and both test pattern generation and test pat- tern generation process concurrently with the principal CNN data path, other. the routing paths and control multiplexers cause slight degradation of the maximum achievable operating frequency (Fmax). This timing penalty is increased in deeper pipelines or as. with faster clock rates, routing congestion and critical path elongation has direct influences on performance. Mitigating this should have better test path pipelining and placement. knowledge compilation optimizations. The other limitation is the trend of resource utilization, that grows about directly proportional to CNN depth and filter. width. More convolutional layers or feature maps are added as additional layers of the network are added. implemented, the registers, LUTs and DSP slices. comprising the operational reasoning as well as the embedded BIST. circuitry increases. The area overhead of the self-test is large, though not very large. It can become blocks even when a shallow network is taken. not so easy in larger topologies, thus con- straining scalability on medium range FPGA such as the Artix- 7. Balancing fault hardware cost coverage is consequently a major architectural. trade-off. Furthermore, chances of signature aliasing in the signature are low as a result of signature aliasing in the signature. MISR is statistically low, and it is impossible to get rid of it. Various fault conditions can sometimes result in the same. packed signatures, which causes the errors to escape.

This issue when shorter MISR registers are used, becomes more demanding. when patterns of correlated data minimize randomness in the test responses. A number of possible improvements were made to over- come these challenges. are identified. To begin with, pipelined MISR integration may dis- preserving tribute compaction delay on a multi-stage basis.

no promotion of area by timing closure. Second, an Dynamic mechanism of adaptive test scheduling may be considered. are identified. First, pipelined MISR integration can dis- tribute compaction delay across multiple stages, maintaining timing closure without increasing area excessively. Second, an adaptive test scheduling mechanism could dynamically perform BIST operations on idle CNN cycles reducing performance interference. Third, partial re- configuration; isolating based fault provides a viable solution to repairing. or not stopping the entire sub modules where there are defects. system. Also, the inclusion of power-gating techniques. when idle time at BIST is significant can save much dynamic. improving and increasing the life of the device by dissipation of power. energy efficiency. Generally, the existing architecture proves a feasible one. pediment to simultaneous self-testing on CNN accelerators, new versions should aim at maximizing the timing, scalability, and power efficiency to have a healthy and to be fully deployable resilient AI hardware system.

Integration in Larger CNN Systems

The proposed C NN–BIST architecture is designed to scale efficiently across multi-layer and high-throughput convolutional accelerators. Its modular structure enables each convolutional or pooling block to embed a dedicated LFSR–MISR test pair and a lightweight control FSM, allowing localized self-test and fault confinement. Such modularity is essential when extending the design to deeper CNN architectures, where inter-layer dependencies can amplify the effects of undetected faults. In a hierarchical CNN accelerator, multiple BIST-enabled computation tiles can operate either in a time-multiplexed or parallel testing mode. A global BIST controller can sequence the activation of test modules to balance diagnostic cover- age and run time performance. During standard inference, the CNN core functions normally, whereas during test mode, the BIST subsystem overrides input paths to inject pseudo-random patterns and capture resulting signatures. This on-demand transition between functional and test operation supports continuous health monitoring without halting normal execution. At the system-on-chip (SoC) level, the CNN–BIST sub- system can be seamlessly integrated with a RISC-ARM processor via an AXI-Lite or APB interface. Test control registers such as BIST start, status flags, and sig- nature readback can be memory-mapped for software-level access. This integration enables the processor to schedule self- tests dynamically—either periodically or in response to run- time errors—thus realizing a software–hardware co-managed reliability mechanism. The proposed integration approach is also compatible with FPGA-based heterogeneous SoC's such as Xilinx Zynq or Intel Agilex platforms. The self-test engine can share on-chip memory and interconnects with neural accelerators through the same communication backbone, ensuring that no additional routing or control overhead is required. Furthermore, the structure allows scalability to multi-core CNN accelerators, where each core's MISR outputs can be combined or compressed hierarchically for aggregated health diagnostics. Ultimately, this integration strategy promotes a balance between scalability, test coverage, and power efficiency, supporting the adoption of BIST-enabled CNN's in industrial and mission-critical AI hardware systems.

Security and Reliability Implications

In addition to traditional structure testing, there is the CNN- BIST. mechanism increases the reliability and security of the system-level. resilience. The current FPGA-based AI accelerators, in turn, utilize, accidental or ill willed faults can be transient. faults, configuration faults or deliberate fault injection. These disturbances may cause a change in network weights or activations, deteriorating the accuracy of models or facilitating adversarial behaviour. The inbuilt self- test mechanism is a proactive integrity. monitor that would be able to notice such deviations at the initial stage of operation. Validating the computation paths within the company in a consistent manner. with signature comparison, LFSR-MISR combination is obtained. ensures an assurance of hardware correctness. If a mis-similarity between the existing and golden signatures is identified, the system can cause predetermined safety reaction like reconfiguration, re-inspiration or isolation of the impacted CNN. layer. Such an automated diagnosis response eliminates latent faults. to a higher level of decision logic, which is important in safety-based areas. The CNN-BIST also pro- cybersecurity wise. provides a deterrence against attacks with hardware. Attackers trying to introduce faults in clock glitches, electromagnetic interference will, or power variations will. commonly modify the MISR signature, inducing the test. controller to indicate an integrity violation. Thus, the design gives a two-fold protection– functional reliability. against casual defects and structural guarantee against malicious tampering. The following reliability framework can be integrated in security-sensitive. autonomous navigation, defense-grade applications. medical inference systems provide both embedded vision or embedded vision. software reliability and hardware dependability. Its low weight, support of common SoC buses, and power to work simultaneously with inference constitute. it a next-attractive reliability augmentation technique. generation ai accelerators that are safety-related. deployments.

6. DISCUSSION

The incorporation of Built-In Self-Test (BIST) mechanism. a CNN computation pipeline is an important representation of CNN. moving in the direction of reliable AI hardware. Traditional CNN accelerators are concerned with throughput and accuracy, but commonly they are not. excessive look fault tolerance, which is acute when used in austere or operational critical situations like aerospace, autonomous navigation or medical imaging systems. The suggested CNN-BIST model brings in a. verification layer which checks the underlying by constantly validating it. computation hardware, with much better reliability. The core advantage of this approach lies in its minimal overhead and structural transparency. By embedding the Linear Feedback Shift Register (LFSR) and Multiple Input Signature Register (MISR) directly within the CNN data path, testing can occur without halting inference operation. This enables online fault detection, unlike traditional offline or scan-based test strategies that interrupt computation. The experimental synthesis and simulation results demonstrate that even with additional logic for test pattern generation and signature capture, the total utilization remains below 0.3% of the available FPGA LUT's and registers. The 9-cycle latency introduced during test iterations is negligible compared to the CNN's inference latency. Power consumption remains modest at 0.253 W, confirming that the design can sustain continuous testing in embedded systems powered by constrained energy sources. There is functional robustness in terms of architecture success- completely identifies single stuck-at and multi-bit logic faults. In the case that a conscious error was introduced into the convolution data. path, the signature generated by MISR was not the same as that which was stored. golden value, the correct detection of a BIST Fail condition. Fault-free execution on the other hand produced an identical signature, indicate successful passing of test. This supports the sincerity of the test engine as well as CNN computing module. The philosophy of design stresses on modularity and scalability. Pairs of CNN layers or computational tiles can be made. having a localized BIST unit, creating a hierarchical fault. tolerant structure. This enables the future systems to grow. forming a smooth transition between the single-layer CNN's and deep convolutional. architectures which do not reformulate the self-test logic.

7. CONCLUSION

A complete design, simulation and have been provided in this paper. A Convolutional Neural Network Implemented using FPGA. The (CNN) architecture was comprised with an autonomous Built-In. Self-Test (BIST) mechanism. Implemented in Verilog HDL and tested on the Artix-7 based on Xilinx Vivado 2023.1. platform, architecture is used to show how fault detection and high-speed AI computation can be used together with functional testing. The findings validate the fact that the BIST sub- was integrated. system brings in performance penalties with negligence in the course

of performance. allowing live monitoring. The LFSR-MISR pair provides excellent fault coverage of the CNN data path, and the control FS coordinates mode very well. changes and signature comparatives. The combination of Computation and test logic on the same hardware of CNN. block decreases testing dependencies, external testing in particular. design highly suitable for safety-critical systems and deployed in the fields. In addition to the basic validation, the CNN-BIST model provides. a system of reliability-based AI accelerator. It provides a base real-time, fault-resilient CNN hardware which is capable of. identify, isolate, and ultimately get over hardware- induced computation errors. All in all, the coherent CNN-BIST system demonstrates that incorporation of self- tests into neural accelerators is both practical and advantageous towards long term system reliability.

Future Work

The next generation of the proposed CNN- will involve future research. BIST structure to pipelined and multilayer CNN. architectures. It is possible to introduce a hierarchical BIST strategy, whereby a localized test exists in every convolutional or pooling layer. controller with fault management unit across the world. Further pursuits will seek to investigate: Dynamic Test Scheduling: Enabling adaptive self-testing during idle computation cycles to minimize performance interference. Fault Recovery Mechanisms: Implementing redundancy-based or partial reconfiguration techniques to automatically isolate faulty modules. • Scalable AI Models: Integrating BIST logic into larger CNN models with pooling, normalization, and quantization layers to evaluate cross-layer fault tolerance. ASIC Implementation: Translating the architecture to a CMOS ASIC flow for power- and area-optimized silicon prototypes. Moreover, more sophisticated fault models like transient timing. soft errors caused by faults and through radiation will be included. to future simulations to test under real- world. conditions. These improvements will be useful in establishing a uniform method of developing testable and reliable neural. speeds up edge and embedded platforms.

Enhanced Future Work

Further studies will focus on multi-layer and pipelined CNN. BIST architectures that have localized test controllers in each layer. Other directions include:

1. Hardware–software co-verification frameworks for hybrid AI accelerators.
2. Integration of partial reconfiguration to isolate faulty regions dynamically.
3. AI-assisted fault prediction model stop preemptively detect degradation.

ACKNOWLEDGMENT

The authors express their sincere gratitude to the Department of Electronics and Communication Engineering, Delhi Technological University (DTU), for providing the infrastructure and technical support required to carry out this work. The availability of the Xilinx Vivado environment and the Artix-7 FPGA evaluation platform (XC7A200T) enabled detailed synthesis and implementation analysis. The faculty guidance is also mentioned by the authors. experts who are knowledgeable about designing the digital system and checking helped in perfecting the architectural design. and experimental test of this project.

REFERENCES

[1] S. Mitra and K. Kim, Built-In Self-Test Techniques for Digital Sys- tems, IEEE Design & Test, vol. 29, no. 2, 2012.

[2] Y. LeCun et al., Deep Learning," Nature, vol. 521, pp. 436–444, 2015.

[3] Xilinx, Artix-7 FPGA Family Overview, DS180, 2021.

[4] M. Abramovici et al., Digital Systems Testing and Testable Design, IEEE Press, 1990.

[5] N. H. E. Weste and D. Harris, CMOS VLSI Design, 4th ed., Addison- Wesley, 2011.

[6] S. Lee et al., Low-Cost BIST for Deep Learning Accelerators," IEEE Trans. VLSI, vol. 29, no. 4, 2021.

[7] H. Nguyen et al., Online Testing for FPGA-Based CNNs Using Signature Analysis, IEEE Access, vol. 10, pp. 34002–34012, 2022.

[8] J. Han and M. Orshansky, Approximate Computing: An Emerging Paradigm for Energy-Efficient Design, IEEE ETS, 2013.

[9] R. C. Baumann, Soft Errors in Advanced Semiconductor Devices, IEEE Trans. Device Mater. Rel., vol. 5, no. 3, 2005.

[10] P. Girard, Survey of Low-Power Testing of VLSI Circuits, IEEE Design & Test of Computers, vol. 19, no. 3, pp. 80–90, 2002.

[11] S. Borkar, Designing Reliable Systems on a Chip, IEEE Micro, vol. 25, no. 3, pp. 10–16, 2005.

[12] A. R. Alam et al., Hardware Fault-Tolerant Deep Learning for Embed- ded Systems, IEEE Embedded Systems Letters, 2020.

[13] H.Leeetal., On-Chip Test and Diagnosis for Neural Network Accelerators, IEEE Trans. VLSI, vol. 31, no. 5, pp. 842–853, 2023.

[14] M.ZhangandA.Kumar, Reliability-Driven FPGA CNN Architectures with Built-In Error Detection, IEEE Access, 2024.

[15] S. Banerjee et al., Hardware Fault Detection in AI Accelerators usingSignature Compression, Microelectronics Journal, 2022.

[16] A.Pathaketal., Testability and Verification Strategies for Edge AI Chips, IEEE Design and Test, vol. 40, no. 2, pp. 80–93, 2023.

[17] P.SinghandR. Patel, FPGA-BasedBuilt-InSelf-TestforCNNPipelines, IEEE Embedded Systems Letters, 2024.

[18] D.Lietal., Low-OverheadFaultDetectionforFPGA-BasedDeepLearning, IEEE TCAD, 2023.