

저작권양도서

(Copyright Transfer Form)

소속 : 인하대학교 전기전자공학부

논문제목 : 고효율 Systolic Array 테스트를 위한 병렬 패턴 주입 기법

저자1(성명, 학번) : 김가현, 12211733

저자2(성명, 학번) : 장준영, 12191529

본인은 상기 논문을 2025학년도 2학기 ‘전기전자 종합설계’ 결과 논문으로 제출하고자 합니다. 본 논문의 내용은 저자가 직접 연구한 결과인 것과 이전에 출판된 적이 없음을 확인합니다. 또한 공저자와 더불어 인하대학교 전기전자공학부에서 발간하는 논문집에 본 논문을 수록하는 것을 허락하며 제반 저작권을 전기전자공학부에 양도합니다.

2025년 12월 8일

저자1 : 김가현 (서명)

저자2 : 장준영 (서명)

전기전자공학부장 귀하

고효율 Systolic Array 테스트를 위한 병렬 패턴 주입 기법

Broadcast-Input Comparison-Scan Architecture for High-Efficient Systolic Array Testing

김가현, 장준영
이영우(서명)

(Ga-Hyeon Kim, Jun-Young Jang and Young-woo Lee)

Abstract: This paper proposes a BICS-BIST (Broadcast-Input Comparison-Scan Built-In Self-Test) architecture to enhance the test efficiency of Systolic Arrays, which are core computational units in AI accelerators. The conventional serial scan-based STRAIT architecture faces structural limitations where test time increases linearly with array size and parallel input is infeasible due to its serial data injection (Shift-in) method. To address these issues, the proposed BICS-BIST architecture adopts a method that broadcasts test data to all Processing Elements (PEs) in parallel and replaces the conventional serial scan with a vertical parallel shift mechanism. By performing In-Situ Comparison within PEs and compressing the vertical column outputs via bitwise-OR logic, it minimizes the test cycles required for result readout. Experimental verification on a 16x16 Systolic Array using a Xilinx Artix-7 FPGA demonstrated that the proposed architecture achieved a 45.7% reduction in test latency (from 560 cycles to 304 cycles) compared to the conventional STRAIT. Furthermore, it proved high reliability by effectively detecting not only Single Stuck-at Faults but also Multi-Burst Faults and Transition Delay Faults during high-speed operation. Although hardware resource (LUT) utilization increased due to the parallel logic implementation, the proposed architecture confirmed its potential as a test solution for next-generation high-performance AI semiconductors by demonstrating clear advantages in terms of test throughput and fault coverage.

Keywords: Systolic Array, Built-In Self-Test, AI Accelerator, Parallel Testing, Fault Tolerance

I. 서론

1. 연구 배경

인공지능(AI) 모델의 성능이 지속적으로 향상됨에 따라, 대규모 행렬 곱셈 연산을 고속으로 처리할 수 있는 하드웨어 가속기의 수요가 급증하고 있다. 이러한 요구에 대응하기 위해 개발된 Systolic Array 구조는, 높은 연산 병렬성과 데이터 재사용 효율을 바탕으로 다양한 AI 하드웨어 시스템에서 핵심 연산 블록으로 채택되고 있다. 대표적인 예로 Google TPU(Tensor Processing Unit)는 Systolic Array 기반의 행렬 곱셈 유닛을 활용하여 딥러닝 연산을 효율적으로 수행한다.

하지만 Systolic Array는 수많은 Processing Element(PE)로 구성되어 있어, 하나의 PE에 발생한 결함이 전체 연산 결과에 심각한 영향을 미칠 수 있다. 특히 각 PE가 파이프라인처럼 연산 흐름을 구성하고 있어, 특정 PE의 오류는 다음 연산 단계로 전파되며 최종 출력에 오류를 유발한다. 이에 따라, 하드웨어 가속기의 신뢰성을 보장하기 위해 각 구성 요소의 정상 동작 여부를 시스템 내부에서 스스로 검증할 수 있는 Built-In Self-Test(BIST) 기술의 중요성이 부각되고 있

다.

대표적인 구조로는 STRAIT(Scan Test and Repair Architecture for AI Tensor Core)가 있다. STRAIT는 각 PE 내부에 시프트 레지스터(Shift Register)와 캡처 레지스터(Capture Register)를 포함한 Scan Chain을 구성하여, 테스트 벡터를 시리얼로 주입(shift-in)하고, 연산 결과를 캡처한 후, 시리얼로 출력(scan-out)하면서 기대 값과 비교하는 구조다. 이러한 방식은 기존 DFT(Design-for-Testability) 기법을 Systolic Array 구조에 효과적으로 적용할 수 있다는 장점이 있지만, Scan Shift 과정에 많은 클럭이 소요되며, 병렬 입력이 불가능하다는 구조적 제약으로 인해 테스트 시간이 배열 크기에 따라 선형적으로 증가하는 단점이 있다.

본 연구에서는 STRAIT 구조의 시간 소모와 구조적 제약을 개선하기 위해, 테스트 데이터를 각 PE에 병렬로 주입하고, 연산 결과를 병렬로 비교하는 BICS-BIST(Broadcast-Input Comparison-Scan BIST) 구조를 제안한다. 제안하는 구조는 전체 테스트 시간을 줄이는 동시에 설계 복잡도와 회로 자원 사용량을 줄이는 것을 목표로 한다. 특히 본 연구에서는 메모리 셀 자체의 결함 검출을 다루는 Memory BIST(MBIST)는 구현

대상에서 제외하고, Systolic Array의 연산 로직에 대한 Logic BIST(LBIST)만을 중점적으로 구현하였다.

2. 연구 목적 및 구성

본 연구의 목적은 기존 STRAIT 구조의 한계를 극복할 수 있는 새로운 BIST 아키텍처인 BICS-BIST를 제안하고, 이를 RTL 수준에서 설계 및 시뮬레이션 하여 성능을 정량적으로 비교 분석하는 데 있다. STRAIT 구조는 테스트 데이터를 직렬 입력 방식으로 전달하는 특성상, Systolic Array의 크기가 증가할수록 테스트에 필요한 클럭 수와 자원이 선형적으로 증가하는 문제를 안고 있다. 특히 모든 PE에 동일한 테스트 벡터를 전달해야 하는 경우에도, 데이터를 순차적으로 이동시켜야 하기 때문에 테스트 시간 최적화에 제약이 존재한다.

이러한 문제를 해결하기 위해 본 연구에서는 병렬 입력 기반의 BIST 구조인 BICS-BIST(Broadcast-Input Comparison-Scan BIST)를 고안하였다. 이 구조는 테스트 데이터를 각 PE에 동시에 병렬로 주입한 후, Capture 및 Compare 과정을 거치는 방식으로 테스트를 수행하여, 전체 테스트 시간을 줄이고 설계 복잡도를 낮추는 것을 목표로 한다.

본 논문의 구성은 다음과 같다. 제1장에서는 연구의 필요성과 목적을 제시하고, 제2장에서는 Systolic Array 와 기존 STRAIT 구조의 기술적 특징 및 한계를 분석한다. 제3장에서는 이를 개선하기 위해 제안하는 병렬 입력 기반 BICS-BIST의 아키텍처와 핵심 동작 원리를 다룬다. 제4장에서는 제안 구조를 RTL 수준에서 구현 및 검증하고 기존 구조와의 성능 비교 결과를 제시한다. 끝으로 제5장에서는 실험 결과에 대한 고찰과 함께 본 연구의 최종 결론 및 향후 발전 방향을 논한다.

II. 기존 구조 및 관련 연구

1. Systolic Array 구조

Systolic Array는 규칙적인 격자 형태로 구성된 연산 유닛(Processing Element, 이하 PE)들이 시간의 흐름에 따라 일정한 방향으로 데이터를 흐르게 하면서 연산을 수행하는 구조이다. 이 구조는 1979년 H. T. Kung에 의해 처음 제안되었으며, 이후 고속 행렬 연산, 신호처리, 딥러닝 가속기 등 다양한 분야에서 널리 활용되고 있다. 특히, 데이터 병렬성과 지역성(Locality)을 동시에 만족할 수 있다는 점에서 하드웨어 효율이 뛰어나고, 연산 처리 속도와 에너지 효율 측면에서도 매우 우수한 구조로 평가받고 있다.

Systolic Array는 일반적으로 2차원 형태의 PE 배열로 구성되며, 각 PE는 인접한 PE로부터 데이터를 전달 받아 연산을 수행한 후, 그 결과를 다시 다른 방향의 PE로 전파하는 방식으로 동작한다. 가장 대표적인 사용 예는 행렬 곱셈(Matrix Multiplication)이다. 예를 들어, 두 행렬 A ($M \times K$)와 B ($K \times N$)를 곱하여 결과 행렬 C ($M \times N$)를 계산할 때, A의 요소는 왼쪽에서 오른쪽으

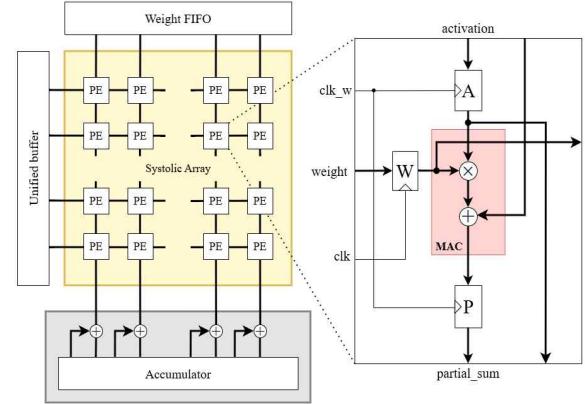


그림 1. 일반적인 Systolic Array-based AI 가속기 구조 [2]

Figure 1. Typical Systolic Array-based AI Accelerator Structure [2]

로, B의 요소는 위쪽에서 아래쪽으로 각각 이동하며, 각 PE는 자신이 담당하는 결과 행렬 C의 원소 하나를 계산하는 역할을 수행한다. 이때 PE는 입력된 A의 요소와 B의 요소를 곱하고, 이전 단계에서 계산된 partial sum을 누적하여 최종 결과를 생성한다.

하나의 PE는 주로 세 가지 기능을 수행한다. 첫째, 입력된 두 데이터(Activation과 Weight)에 대한 곱셈 연산(Multiply), 둘째, 이전 단계의 partial sum을 더하는 덧셈 연산(Accumulate), 셋째, 결과를 다음 PE로 전달하는 전파(Propagation) 기능이다. 이와 같은 연산 방식은 MAC(Multiply-Accumulate) 구조로 요약되며, CNN과 같은 딥러닝 모델에서 핵심 연산으로 활용된다. Systolic Array는 클럭 사이클마다 각 PE가 입력 데이터를 받아 MAC 연산을 동시에 수행하기 때문에 높은 병렬성을 구현할 수 있으며, 인접한 PE 간의 데이터 전달만으로 동작이 가능하므로 외부 메모리 접근을 최소화할 수 있어 에너지 효율 또한 높다.

Systolic Array의 동작은 입력 데이터를 시계열적으로 전달한다는 점에서 큰 장점을 갖는다. Activation은 일반적으로 왼쪽에서 오른쪽으로, Weight는 위쪽에서 아래쪽으로 한 클럭마다 한 칸씩 이동하며, 각 PE는 이 데이터를 받아 MAC 연산을 수행한다. 이를 통해 전체 구조는 일정한 클럭 사이클마다 각 위치에서 동시에 연산을 수행하게 되며, 구조적으로 매우 높은 병렬성을 가지게 된다.

이러한 특성 때문에 Systolic Array는 하드웨어 딥러닝 가속기에서 가장 핵심적인 구조로 채택되고 있으며, 대표적인 예로는 Google의 TPU(Tensor Processing Unit), NVIDIA의 Tensor Core 등이 있다. 이러한 시스템들의 기반이 되는 일반적인 Systolic Array AI 가속기 구조는 그림 1과 같다.

하지만 높은 병렬성과 연산 밀도를 가진 Systolic

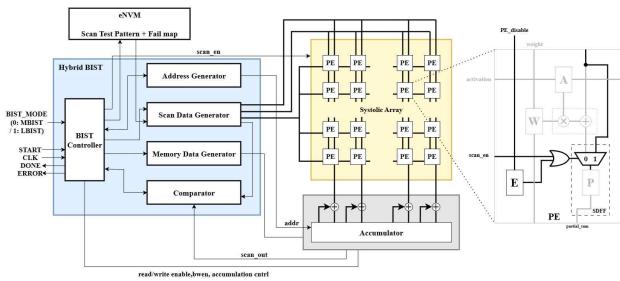


그림 2. STRAIT의 BIST 구조도 [2]

Figure 2. BIST structure diagram of STRAIT [2]

Array 구조는, 동시에 많은 수의 PE를 포함하기 때문에, 하드웨어 결함(Fault)에 매우 민감하다는 단점도 가지고 있다. 하나의 PE에 발생한 오류가 전체 행렬 연산 결과에 영향을 미칠 수 있으며, 이러한 결함을 빠르게 탐지하고 복구할 수 있는 테스트 구조가 반드시 필요하다. 이에 따라 최근에는 Systolic Array 구조 내부에 BIST(Built-In Self-Test)를 통합하여, 외부 테스트 장비 없이도 자체적으로 결함을 진단할 수 있는 구조가 적극적으로 연구되고 있다.

2. STRAIT 기반 BIST 구조

Systolic Array는 수많은 PE로 구성되어 있고, 이를 각각은 MAC 연산을 수행하는 핵심 단위이기 때문에, 하나의 PE라도 오작동할 경우 전체 행렬 연산 결과에 심각한 영향을 미칠 수 있다. 이에 따라 Systolic Array의 신뢰성을 확보하기 위해 결함 탐지를 위한 내장 자가 테스트 기법인 BIST(Built-In Self-Test)의 적용이 필수적이다. 최근 발표된 STRAIT(Scan Test and Repair Architecture for AI Tensor Core)는 이러한 요구에 대응하기 위해 제안된 대표적인 BIST 아키텍처 중 하나이며, 그 전체적인 구조는 그림 2와 같다.

STRAIT는 각 PE 내부에 scan chain을 삽입하여 테스트 데이터를 주입하고, 내부 연산 결과를 캡처한 뒤, 이를 기대값과 비교하는 방식으로 결함을 탐지한다. STRAIT 구조에서 가장 핵심적인 특징은 테스트 벡터를 각 PE에 shift 방식으로 전달한다는 점이다. 이 구조에서는 test mode로 진입한 후, scan enable 신호(scan_en)를 활성화하여, 테스트 데이터를 scan-in 방식으로 한 비트씩 순차적으로 이동시키며 각 PE의 입력에 도달하게 한다.

이 과정은 scan shift process라고 하며, Systolic Array 내 모든 PE에 동일한 테스트 벡터를 공급할 수 있는 장점을 제공한다. Scan shift process가 완료되면, scan_en 신호를 비활성화하고 capture 신호를 통해 PE 내부의 연산 결과를 레지스터에 저장하는 scan capture process가 수행된다. 이후, 이 레지스터의 값은 Main Comparator로 전달되어, 예상 결과(expected value)와 비교된다. 만약 실제 결과와 기댓값이 일치하지 않으면 에러 플래그가 활성화되며, 최종적으로 모든 PE의 에

러 상태를 집계하여 총 에러 개수를 산출하게 된다.

STRAIT 구조는 설계가 비교적 단순하며, 기존 DFT 설계 흐름과 유사한 방식으로 구성되어 있어 실제 구현이 용이하다는 장점을 가진다. 특히 각 PE가 동일한 scan 구조를 가지기 때문에, 전체 구조가 규칙적이고 모듈화 되어 있다는 점에서 복잡한 AI 가속기 시스템에 적용하기에 적합하다.

3. STRAIT 구조의 한계

하지만 STRAIT 구조는 테스트 데이터의 전달 방식이 serial 방식에 의존하고 있다는 근본적인 한계를 내포하고 있다. 테스트 데이터가 scan chain을 따라 순차적으로 이동해야하기 때문에, 전체 테스트 시간은 scan chain의 길이에 따라 선형적으로 증가한다. 예를 들어, 4x4 구조의 Systolic Array에서는 PE당 3개의 테스트 벡터를 주입해야 한다면, 총 $3 \times 16 = 48$ 클럭 이상의 shift 과정이 필요하게 된다. 이로 인해 전체 테스트 시간이 길어지고, 테스트 종 회로가 long-hold 상태로 유지되어야 하므로 전력 소비 측면에서도 비효율적일 수 있다. 또한, shift-in 시 데이터가 각 PE에 동시에 주입되지 않기 때문에, 테스트 중간 단계에서의 예측 불가능한 출력이 발생할 수 있으며, 이는 테스트 정확도에 영향을 줄 가능성도 존재한다.

이러한 문제를 해결하고자 본 연구에서는 STRAIT의 scan shift 구조를 제거하고, 테스트 데이터를 각 PE에 동시에 병렬 주입하는 BICS-BIST 구조를 제안한다. BICS-BIST 구조는 scan chain이 아닌 broadcast 방식으로 데이터를 주입하여 테스트 시간을 획기적으로 줄이는 것을 목표로 하며, 다음 장에서 그 구조 및 동작 방식에 대해 자세히 설명한다.

III. 제안하는 구조: BICS-BIST

1. 설계 개요

본 장에서는 기존 STRAIT 구조의 한계를 개선하고자 본 연구에서 제안하는 새로운 병렬 BIST 구조인 BICS-BIST(Broadcast-Input Comparison-Scan Built-In Self-Test)의 설계 개요를 서술한다. 제안하는 구조는 Systolic Array의 테스트 효율을 높이기 위해 scan chain 기반의 순차적 테스트 데이터 주입 방식을 제거하고, 그 대신 테스트 데이터를 병렬로 각 PE에 동시에 전달하는 방식을 채택하였다. 이로써 전체 테스트 시간을 획기적으로 줄이고, 복잡한 scan shift 제어 로직을 단순화하여 회로 자원 사용량을 절감할 수 있다.

기존 STRAIT 구조에서는 테스트 데이터를 scan-in 방식으로 전달하기 위해 각 PE 내부에 shift register를 포함한 scan chain을 삽입해야 했다. 이 구조는 데이터가 한 클럭씩 이동하며 순차적으로 각 PE에 도달하므로, 전체 Systolic Array 크기에 비례하여 테스트 시간이 선형적으로 증가하는 문제가 있었다. 반면, BICS-BIST 구조는 모든 테스트 데이터를 한 클럭에 모

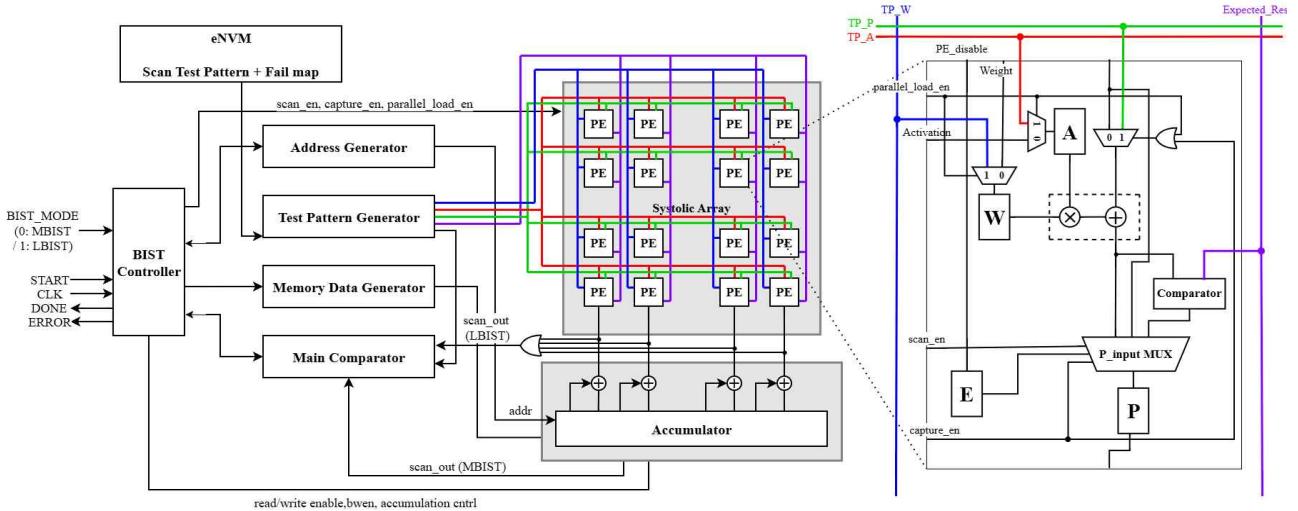


그림 3. BICS-BIST 구조도
Figure 3. BICS-BIST structure diagram

는 PE에 동시에 주입하는 broadcast 방식을 사용하므로, 테스트 시간 측면에서 scan shift 구조에 비해 우수한 성능을 보일 수 있다.

BICS-BIST 구조의 핵심은 총 네 단계의 동작으로 요약된다. 첫 번째는 초기화 단계(Initialization)로, 클럭 및 리셋 신호를 통해 전체 시스템을 안정된 초기 상태로 설정하는 과정이다. 두 번째는 테스트 벡터 주입 단계(Test Pattern Injection)로, Test Pattern Generator를 통해 생성된 activation 및 weight 테스트 벡터가 각각 TP_A와 TP_W 경로를 통해 모든 PE에 병렬로 입력된다. 세 번째는 연산/비교 및 에러 캡처 단계(Operation, Comparison & Error Capture)로, 주입된 테스트 벡터를 바탕으로 각 PE가 MAC 연산을 수행하고, 그 결과를 기댓값과 즉시 비교한 후, 1비트 에러 플래그를 내부 레지스터에 캡처하는 과정이다. 마지막으로 네 번째는 에러 시프트 아웃 및 최종 카운트 단계(Error Shift-Out & Final Count)로, 캡처된 에러 플래그를 scan chain을 따라 직렬로 shift-out하고, Main Comparator를 통해 최종적으로 총 에러 개수를 계산한다.

제안하는 BICS-BIST 구조는 전체 제어 흐름을 하나의 Finite State Machine(FSM)으로 제어하며, 각 단계에서 필요한 enable 신호들을 순차적으로 생성한다. FSM은 총 여섯 개의 상태(IDLE, LOAD, CAPTURE, SHIFT, PATTERN_INC, DONE)로 구성되어 있으며, 각 단계에서 필요한 enable 신호들을 순차적으로 생성한다. 특히, 테스트 패턴 주입이 한 클럭에 완료되기 때문에 FSM의 전체 상태 수를 최소화할 수 있고, 상태 간 전이 조건도 간결하게 구성할 수 있다는 장점이 있다. 이를 통해 제어 로직의 복잡성을 줄이는 동시에, 테스트 처리 속도를 높일 수 있다.

BICS-BIST 구조는 이론적으로 scan-in 기반 구조에 비해 훨씬 빠른 테스트를 제공할 수 있지만, 실제 구현에서는 제어 타이밍, 파이프라인 동기화, 비교기 구조

등 다양한 요소가 전체 성능에 영향을 미친다. 따라서 본 구조는 Verilog로 구현한 후 Vivado 환경에서 STRAIT 구조와 동일한 조건으로 시뮬레이션을 진행하여 비교 평가하였으며, 테스트 시간, FSM 상태 수, 회로 자원 사용량 등을 중심으로 그 성능을 분석하였다. 이러한 비교 분석을 통해 본 구조의 효용성과 한계점을 명확히 파악하고, 향후 개선 방향을 도출하는 것이 본 장 이후의 주요 목표이다.

2. 시스템 구성 및 모듈 설명

본 연구에서 제안하는 BICS-BIST 시스템은 Systolic Array 기반 AI 가속기의 신뢰성을 확보하기 위해, 연산 로직(Logic)과 메모리(Memory) 결합을 모두 테스트할 수 있는 통합 BIST 아키텍처로 설계되었다. 그림 3은 전체 시스템 구조를 나타내며, 핵심 제어 모듈인 BIST Controller를 중심으로 TPG, Address Generator, Systolic Array, Main Comparator가 유기적으로 연결된다. 특히, BIST Controller는 외부 신호(BIST_MODE)에 따라 동작 모드를 전환한다. MBIST 모드에서는 Accumulator RAM 제어 신호를 활성화하여 메모리를 테스트하고, LBIST 모드에서는 이를 비활성화하고 Systolic Array의 로직 결합 검출에 집중한다. 본 논문에서는 핵심 연산 부하를 담당하는 Systolic Array의 검증이 주 목적이므로, LBIST 모드에서의 BICS 구조 구현 및 성능을 중점적으로 기술한다.

2-1. Test Pattern Generator

Test Pattern Generator(TPG)는 Systolic Array의 BIST에 사용될 테스트 벡터를 생성하는 모듈로, 본 구현에서는 미리 저장된 결정론적 패턴을 출력하는 Test Pattern ROM 형태로 설계되었다¹. 이는 Verilog의 case 문을 이용한 조합 논리 회로로 구현되며, Activation(TP_A), Weight(TP_W), Partial Sum 입력

(TP_P), 그리고 예상 결과값(Expected_Result)을 개별 ROM이 아닌 단일 ROM 모듈 내에서 주소(Address)에 매핑하여 동시에 출력하도록 통합 구성된다. 생성된 테스트 벡터들은 별도의 중계 모듈 없이 Systolic Array의 각 입력 포트로 직접 broadcast되며 이는 Stuck-At(SA) Fault 모델과 Transition Delay(TD) Fault 모델 시나리오에 맞춰 사전에 계산된 테스트 패턴을 제공한다. 특히, 기존의 TP_A와 TP_W 경로뿐만 아니라 P 경로의 입력을 위한 테스트 벡터(P_in)가 추가되어, 무작위 패턴으로는 검증하기 힘든 연산의 Corner Case와 타이밍 마진을 더욱 효과적으로 테스트할 수 있다.

2-2. Address Generator

Address Generator는 Test Pattern ROM에 저장된 테스트 벡터들을 순차적으로 로드하기 위한 주소 생성 모듈로, 동기식 카운터(Synchronous Counter)로 설계되었다. BIST Controller FSM으로부터 rom_addr_en 신호를 입력받아 동작하며, FSM이 S_PATTERN_INC 상태로 전이할 때마다 내부 카운터 값을 1씩 증가시켜 다음 테스트 패턴의 주소를 생성한다.

생성된 주소는 단일 통합 모듈로 구성된 Test Pattern ROM으로 전달되며, ROM은 해당 주소에 매핑된 Activation(TP_A), Weight(TP_W), Partial Sum(TP_P), 그리고 Expected_Result 데이터를 동시에 병렬로 출력하여 Systolic Array로 공급한다. 이는 모드에 따라 데이터를 선택적으로 읽어오는 것이 아니라, 주소 지정 한 번으로 필요한 모든 테스트 벡터 세트를 한꺼번에 확보하는 구조이다.

또한, FSM 내부의 패턴 카운터가 설정된 총 패턴 수 (NUM_PATTERNS)에 도달하면(pattern_done), FSM은 이를 감지하여 테스트를 종료하는 S_DONE 상태로 전이하고 Address Generator의 동작을 멈추도록 설계하였다.

2-3. Systolic Array Module

Systolic Array 모듈은 본 구조에서 테스트의 핵심 대상이 되는 연산 블록으로, 규칙적인 2차원 배열 형태로 구성된 다수의 Processing Element(PE)로 이루어져 있다. 각 PE는 입력으로 주어진 activation 값과 weight 값을 곱하고, 이를 테스트 모드 시 병렬 주입되는 P_in 값(또는 이전 단계의 partial sum)과 누적하는 MAC(Multiply-Accumulate) 연산을 수행한다.

본 구조의 가장 큰 특징은 In-Situ Comparison 방식의 도입이다. 각 PE는 연산 결과를 레지스터에 저장하기 이전에, 내부의 Internal Comparator를 통해 Expected_Result와 즉시 비교한다. 비교 결과 생성된 1-bit Fail Flag는 이후 Capture 단계에서 P 레지스터에 저장된다.

이는 기존 STRAIT 구조와 달리, 테스트 데이터 주입 과정에서 Scan Chain을 이용한 Shift-in 방식을 제거하고, broadcast 방식을 통해 각 PE에 병렬로 데이터를 주입하도록 설계한 것이다. 이를 통해 데이터 로드에

소요되는 클럭 사이클을 획기적으로 줄이고, 전체 테스트 시간을 단축함과 동시에 구조적 간소화를 달성하였다.

2-4. Main Comparator

Main Comparator는 Shift 단계에서 Systolic Array로부터 출력되는 결함 신호를 모니터링하여 시스템의 최종 결함 여부를 판별하는 모듈이다. 본 구조에서는 테스트 병렬성을 극대화하기 위해, 각 PE의 Fail Flag가 P 경로를 통해 수직 방향으로 Shift된다. 이때 최하단 행으로 배출되는 각 열의 출력값들은 하단의 Bitwise-OR로 직을 통해 단일 비트로 압축되어 Main Comparator로 전달된다.

따라서 Main Comparator는 별도의 데이터 비교 연산 없이, 이 압축된 스트림(scan_out_data)에 대해 Error Latch 로직을 수행한다. 즉, scan_en 신호가 활성화되어 데이터가 수직으로 Shift-out 되는 동안, 압축된 출력 버스에서 0이 아닌 값(Logic High)이 단 한 번이라도 감지되면 final_error 신호를 1로 천이시키고, 리셋 신호가 인가되기 전까지 해당 상태를 지속적으로 유지한다. 최종적으로 FSM이 DONE 상태에 도달했을 때 이 final_error 값이 0이면 Pass, 1이면 Fail로 판정하여 상위 시스템에 보고한다. 이는 하드웨어 자원을 많이 소모하는 Error Counting 로직을 제거하고, 시스템의 결함 존재 유무만을 효율적으로 판단하도록 설계된 경량화 구조이다.

이처럼 제안된 BICS-BIST 구조는 각 구성 요소들이 고도로 병렬화되어 있으며, 복잡한 외부 비교기나 카운터를 제거하고 In-Situ Comparison과 Vertical Parallel Scan 방식을 채택함으로써, 테스트 속도와 하드웨어 리소스 효율성 측면에서 STRAIT 구조 대비 유리한 특성을 가진다. 다음 절에서는 각 모듈의 FSM 구성과 동작 타이밍을 포함한 전체 시스템의 제어 흐름을 상세히 기술한다.

3. FSM 동작 방식

본 연구의 BIST Controller는 통합 아키텍처 설계를 반영하여, 메모리 테스트(MBIST)와 로직 테스트(LBIST)를 모두 제어할 수 있는 유연한 Finite State Machine(FSM)으로 구현되었다. FSM은 그림 4와 같이 총 여섯 개의 상태(IDLE, LOAD, CAPTURE, SHIFT, PATTERN_INC, DONE)로 구성된다.

설계된 FSM은 MBIST 수행을 위한 RAM 제어 신호 생성 로직을 포함하고 있으나, 본 논문의 검증 대상인 LBIST 모드에서는 해당 신호들이 비활성화(Masking)되거나 Main Comparator 경로로 우회하도록 제어된다. 이를 통해 불필요한 메모리 접근 전력을 차단하고, 로직 테스트의 처리 속도를 극대화하였다. 각 상태별 상세 동작은 다음과 같다.

IDLE 상태는 시스템 리셋 직후 진입하며, 전체 시스템을 안정된 초기 상태로 설정하기 위한 단계이다. 이 단계에서는 상태 레지스터 및 내부 카운터

(Shift/Pattern Counter)가 초기화되며, rom_addr_rst 신호를 통해 Test Pattern ROM의 주소 포인터도 0으로 리셋된다. IDLE 상태는 외부로부터 start 신호가 High로 입력되기 전까지 유지되며, 해당 신호가 감지되면 다음 상태인 LOAD 상태로 전이된다.

LOAD 상태는 테스트 벡터를 각 PE에 병렬로 전달하는 단계이다. FSM은 이 상태에서 parallel_load_en 신호를 출력하여, Test Pattern ROM으로부터 읽어온 Activation(TP_A), Weight(TP_W), Partial Sum(TP_P) 데이터를 Systolic Array 내부의 모든 PE에 동시에 주입한다. 이 과정은 단 한 클럭 만에 수행되며, 데이터 주입이 완료되면 FSM은 즉시 CAPTURE 상태로 전이된다.

CAPTURE 상태는 PE 내부의 연산 결과를 캡처하는 단계이다. 이 상태에서 bist_capture_en 신호가 활성화되면, 각 PE는 내부적으로 수행한 In-Situ Comparison의 결과(Fail Flag)를 P 레지스터에 저장한다. Load 단계와 마찬가지로 단일 클럭 사이클 내에 수행되며, 캡처가 완료되면 FSM은 결과 판독을 위한 SHIFT 상태로 진입한다.

SHIFT 상태는 캡처된 결과 값을 직렬로 Scan-out하여 결합 여부를 판별하는 단계이다. 이 상태는 ARRAY_SIZE(N)만큼의 클럭 사이클 동안 유지된다. 이 기간 동안 scan_en 신호가 활성화되면, Systolic Array 내부의 모든 PE가 저장하고 있는 Fail Flag가 기준의 P 경로를 통해 수직 아래 방향으로 시프트(Vertical Shift)된다. 최하단 행으로 배출되는 각 열의 결과값들은 하단의 Bitwise-OR 로직을 통해 단일 비트로 압축되어 Main Comparator로 실시간 전달되며, 이를 통해 0이 아닌 값(에러)이 존재하는지 검사된다. 내부 Shift 카운터가 N에 도달하여 모든 데이터의 이동이 완료(shift_done)되면, FSM은 다음 패턴 준비를 위해 PATTERN_INC 상태로 전이된다.

PATTERN_INC 상태는 다음 테스트 패턴을 적용할지 여부를 결정하는 단계이다. 이 상태에서는 rom_addr_en 신호를 통해 ROM 주소를 증가시켜 다음 테스트 패턴을 준비한다. 동시에 내부 Pattern 카운터를 확인하여, 설정된 총 패턴 수(NUM_PATTERNS)에 도달했는지 검사한다. 만약 아직 수행할 패턴이 남아있다면 FSM은 다시 LOAD 상태로 돌아가 테스트 사이클을 반복하고, 모든 패턴의 테스트가 완료되었다면 (pattern_done) DONE 상태로 진입한다.

FSM의 마지막 단계인 DONE 상태에서는 전체 BIST 과정이 완료되었음을 알리기 위해 bist_done 신호를 출력한다. FSM은 start 신호가 비활성화될 때까지 이 상태를 유지하며, 이후 start가 Low가 되면 다시 초기 상태인 IDLE 상태로 복귀하여 다음 명령을 대기하게 된다.

이러한 FSM 구조는 'Load-Capture-Shift'로 이어지는 테스트 사이클을 지정된 패턴 수만큼 자동으로 반복 수행하도록 설계되어 있어, 별도의 외부 개입 없이 효율적인 다중 패턴 테스트가 가능하다.

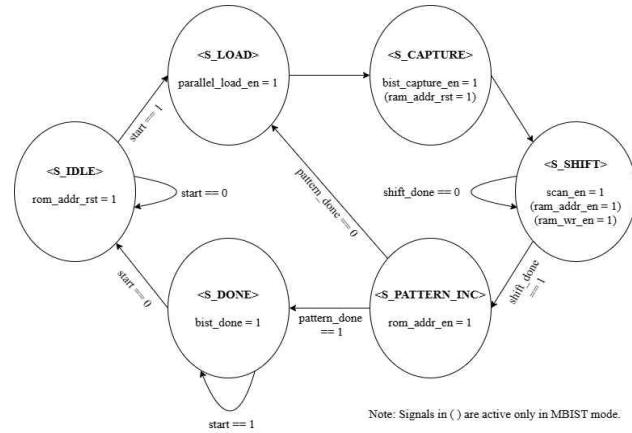


그림 4. BICS-BIST Controller의 FSM 다이어그램
Figure 4. FSM diagram of BICS-BIST Controller

4. 세부 테스트 동작 원리

제안하는 BICS-BIST 구조는 Broadcast 입력 방식을 통해 Stuck-At (SA) Fault뿐만 아니라 고속 동작 시의 Transition Delay (TD) Fault까지 효율적으로 검출할 수 있다. 각 테스트 모드에서의 PE 내부 동작 메커니즘은 다음과 같다.

SA 테스트는 회로 내부의 노드가 0 또는 1로 고정되는 결함을 탐지하기 위한 정적 테스트 과정으로, 그림 5와 같이 세 단계로 수행된다. Parallel Pattern Load (a) 단계에서는 parallel_load_en 신호가 활성화되어 TP_A, TP_W가 레지스터로 로드된다. 동시에 Pre-Adder MUX가 활성화되어 TP_P 값이 덧셈기 입력으로 직접 주입된다. 이는 기존 직렬 방식에서는 불가능했던 덧셈기 로직의 독립적인 제어를 가능하게 한다. Execution & Comparison (b) 단계에서는 로드된 테스트 벡터들을 기반으로 MAC 연산 $(A \times W) + P_{in}$ 이 수행된다. 연산 결과는 즉시 Internal Comparator로 전달되어 Expected_Result와 비교된다. Result Capture (c) 단계에서는 capture_en 신호를 통해 비교 결과(Fail Flag)를 P 레지스터에 저장한다. 이때 Pre-Adder MUX는 캡처 신호에 의해 테스트 경로 연결을 유지하여 연산의 안정성을 보장한다.

TD 테스트는 신호의 전파 지연으로 인한 Setup Time Violation을 검출하기 위한 동적 테스트로, 그림 6과 같은 Launch-on-Load 방식을 따른다. Initialization (a) 단계에서는 레지스터에 초기 벡터(V1)가 저장되어 있고, 입력단에는 다음 벡터(V2)가 대기하는 상태를 형성한다. Launch Event (b) 단계에서 클럭 엣지와 함께 레지스터 값이 V1에서 V2로 급격히 Transition한다. 이 변화는 Critical Path(그림 내 L 경로)인 곱셈기와 덧셈기를 거쳐 전파되며, 이 과정이 한 클럭 주기 내에 완료되는지를 검증한다. At-speed Capture (c) 단계에서는 다음 클럭에서 연산 결과를 캡처한다. 이때 입력 MUX가 기능 경로로 전환되더라도 레지스터의 출력은 유지

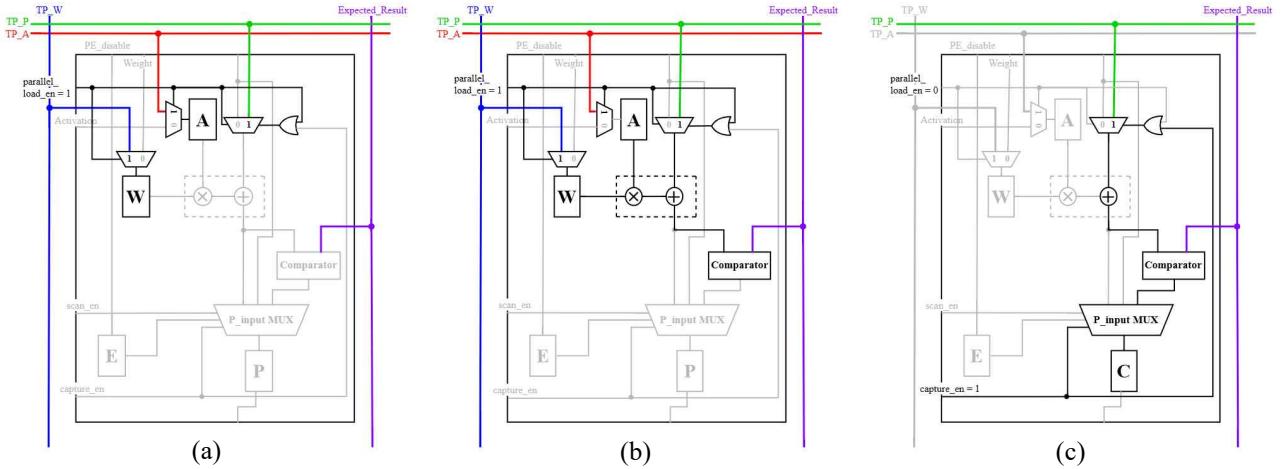


그림 5. BICS-BIST의 SA 테스트 (a) 병렬 패턴 로드, (b) 연산 및 비교, (c) 결과 캡처

Figure 5. SA test of BICS-BIST (a) Parallel pattern load, (b) Execution & comparison, and (c) Result capture

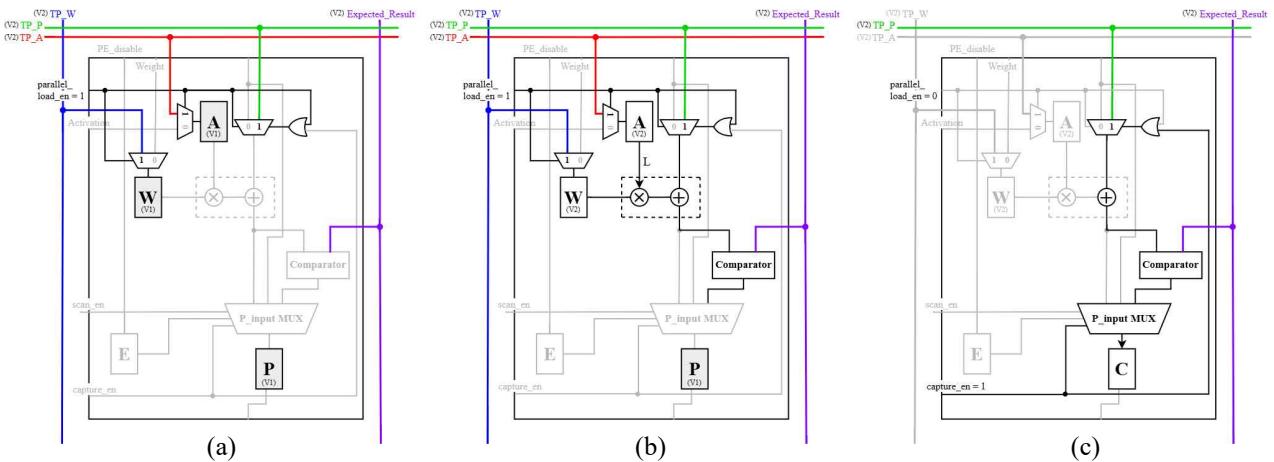


그림 6. BICS-BIST의 TD 테스트 (a) 초기화, (b) 런치 이벤트, (c) At-speed 캡처

Figure 6. TD test of BICS-BIST (a) Initialization, (b) Launch event, (c) At-speed capture

되므로 연산 유효성은 보존된다. 최종적으로 캡처된 결과(C)는 스캔 체인을 통해 외부로 출력된다.

본 제안 구조는 PE 간의 연결선 지연 검사는 배제하고, 단일 PE 내부의 로직 검증에 집중하도록 설계되었다. 이는 Systolic Array 기반 가속기의 전체 동작 속도를 결정짓는 Dominant Path가 물리적 거리가 짧은 PE 간 배선이 아니라, 복잡한 로직 게이트로 구성된 PE 내부의 MAC 유닛이기 때문이다. 따라서 시스템의 최대 동작 주파수를 제한하는 Critical Path인 MAC 유닛을 직접 타격하여 검증하는 방식만으로도 고속 동작에 대한 신뢰성을 충분히 확보할 수 있다.

IV. 실험

1. 실험 환경

본 연구에서 제안한 BICS-BIST 구조의 기능 검증 및 성능 평가는 Xilinx Vivado Design Suite(2022.2)를 활용하여 수행하였다. 하드웨어 설계 언어로는 Verilog-2001 표준을 준수하여 RTL 설계를 진행하였으며, 기능 검증을 위해 Vivado Simulator (XSim)를 이용한 Behavioral Simulation을 수행하였다. Synthesis 및 Implementation을 위한 Target Device는 Xilinx Artix-7 계열의 FPGA를 선정하였으며, 시스템 클럭 주기는 10ns(100MHz)를 목표로 설정하였다. 특히, STRAIT와 BICS-BIST 두 구조 간의 공정한 비교를 위해, Out-of-Context (OOC) 모드를 적용하여 Synthesis를 수행하였다.

Power Consumption 분석은 합성된 Netlist를 기반으로 Vivado의 Power Report 기능을 사용하여 측정하였으며, Dynamic Power와 Static Power를 포함한 Total On-Chip Power를 비교 지표로 삼았다. Timing Analysis 또한 동일한 Constraints 하에서 수행하여, 각 구조의 Critical Path와 최대 동작 주파수(F_{max})를 산출하였다.

Power Consumption 분석은 합성된 Netlist를 기반으로 Vivado의 Power Report 기능을 사용하여 측정하였으며, Dynamic Power와 Static Power를 포함한 Total On-Chip Power를 비교 지표로 삼았다. Timing Analysis 또한 동일한 Constraints 하에서 수행하여, 각 구조의 Critical Path와 최대 동작 주파수(F_{max})를 산출하였다.

표 1. STRAIT와 BICS-BIST의 PPA 및 효율성 비교 요약

Table 1. Summary of PPA and Efficiency Comparison between STRAIT and BICS-BIST

구분	성능 지표	STRAIT	BICS-BIST	변화율
Hardware Cost	Area (LUTs)	1,246	42,511	+3,311%
	Area (FFs)	523	12,664	+2,321%
	Power (Total)	0.136W	1.011W	+643%
Speed	Max Freq (F_{max})	149.35MHz	119.86MHz	-19.74%
Efficiency	Test Latency	560 Cycles	304 Cycles	-45.71
	Execution Time	3.71 μ s	2.54 μ s	-32.3%

2. 실험 데이터 및 검증 시나리오

본 실험의 DUT(Design Under Test)인 Systolic Array는 16x16 크기의 배열로 구성하였으며, 각 PE는 8-bit 정수형(INT8) 데이터를 처리하도록 설계하였다. 테스트의 목적은 Systolic Array의 연산 로직이 정상적으로 동작하는지 검증하고, 결함 발생 시 이를 BIST 회로가 올바르게 탐지하는지 확인하는 데 있다. 실험은 크게 기능 검증(Functional Test)과 결합 주입 시나리오(Fault Injection Scenarios) 두 가지로 진행되었다.

2-1. Functional Test

무작위 난수 생성 방식의 불확실성을 배제하기 위해, Test Pattern ROM에 미리 정의된 Deterministic Pattern을 사용하였다. 검증 패턴은 양수 간 연산, 음수 혼합 연산, 0 포함 연산 등 Corner Case를 포함하여 총 16개의 테스트 벡터로 구성되었다. 이를 통해 MAC 연산의 Overflow 처리 및 Sign Extension 로직의 정상 동작 여부를 RTL 시뮬레이션 레벨에서 전수 검사하였다.

2-2. Fault Injection Scenarios

본 연구에서는 제안 구조의 Fault Coverage를 검증하고자 Verilog의 force 명령어를 활용하여 세 가지 유형의 결함을 주입하였다.

우선, 임의의 PE 출력 노드가 VDD(Stuck-at-1) 혹은 GND(Stuck-at-0)로 단락되는 Single Stuck-at Fault를 적용하였다.

또한, 실제 칩의 전원 라인이나 클럭 트리 문제로 발생할 수 있는 결함을 반영하여, 특정 행(Row)이나 열(Column)의 다수 PE가 동시에 오작동하는 Multi-Burst Fault를 모사하였다.

마지막으로 Transition Delay Fault 검증을 위해, 입력 패턴이 0x00에서 0xFF로 급변하는 구간에 강제로 신호 지연을 주입하여 Setup Time Violation 상황을 유도하였다.

이때 각 결함 신호는 BIST의 Capture 동작이 확실하게 수행되도록 100 클럭 사이클 동안 유지하였다.

3. 실험 결과 및 비교

본 절에서는 기존 STRAIT 구조와 제안하는 BICS-BIST 구조의 성능을 정량적으로 비교 분석한다. 비교 지표로는 하드웨어 자원 사용량(Hardware Cost), 전력 소모(Power Consumption), 동작 속도(F_{max}), 그리고 테스트 효율성(Test Efficiency)을 사용하였다.

3-1. 하드웨어 자원 사용량 및 동작 속도

표 1에 나타난 바와 같이, 제안하는 BICS-BIST 구조는 기존 STRAIT 대비 LUT 사용량이 약 34배, 레지스터(FF) 사용량이 약 24배 증가하였다. 이러한 면적 증가는 주로 테스트 패턴을 저장하기 위한 ROM이 최적화되지 않은 FPGA 내부의 Distributed Logic으로 합성되었기 때문이다. 이에 대한 상세한 원인 분석과 실제 상용 칩 설계 시의 해결책은 5.2절에서 논의한다.

더불어, 제안하는 BICS-BIST 구조는 기존 STRAIT 와 달리 각 PE 내부에 32-bit 크기의 Internal Comparator를 추가하여 In-Situ Comparison을 수행하도록 설계되었다. 기존 구조는 연산 결과를 외부로 Scan-out한 후 Main Comparator에서 순차적으로 비교하지만, 제안 구조는 Internal Comparator를 통해 연산 직후 즉시 결함 여부를 판단한다. 이로 인해 PE 당 로직 사용량이 증가하여 전체 LUT 및 FF 사용량 증가에 기여하였다. 하지만 이는 병렬성을 극대화하여 테스트 속도를 획기적으로 높이기 위한 필수적인 Trade-off로 분석된다.

또한, 타이밍 분석 결과 BICS-BIST의 최대 동작 주파수(F_{max})는 119.86 MHz로, STRAIT(149.34 MHz) 대비 약 19.7% 감소하였다. 구체적인 F_{max} 값은 타이밍 분석 리포트의 WNS(Worst Negative Slack)를 바탕으로 아래의 수식을 통해 도출되었다

$$F_{max} = \frac{1}{Target\ Clock\ Period - WNS}$$

이러한 동작 속도의 감소는 병렬 데이터 주입 및 비교를 위한 추가적인 Routing 자원과 제어 로직이 데이터 경로에 삽입되면서 Critical Path가 다소 길어졌기 때-

표 2. 결합 주입 시나리오별 탐지 결과 요약

Table 2. Summary of Detection Results by Fault Injection Scenarios

시나리오	결합 유형	상태	결과/비고
Case 1	Normal Operation (Benchmark)	PASS	BICS: 304 Cycles vs STRAIT: 560 Cycles
Case 2-A	Single Fault at PE[0][0] (Stuck-at-1)	Detected	Success
Case 2-B	Row Burst Fault at Row 2 (Stuck-at-0)	Detected	Success
Case 2-C	Column Burst Fault at Col 4 (Stuck-at-1)	Detected	Success
Case 3	Transition Delay Fault (Slow-to-Rise)	Detected	Success (ROM Addr 6)

문이다. 그러나 측정된 값은 목표 동작 주파수인 100MHz를 충분히 상회하므로, 실제 시스템 적용에 있어 Timing Margin은 안정적으로 확보된 것으로 판단된다.

3-2. 테스트 효율성

가장 유의미한 성과는 테스트 효율성에서 확인되었다. STRAIT 구조는 Serial Scan 방식으로 인해 테스트 완료까지 560 클럭 사이클이 소요되었으나, BICS-BIST는 Broadcast Input 방식을 적용하여 304 클럭 사이클만에 테스트를 완료하였다. 이를 실제 물리적인 시간 (Execution Time, 100MHz 기준)으로 환산하면, BICS-BIST는 $3.04\mu s$ 만에 테스트를 수행하여 STRAIT($5.60\mu s$) 대비 약 1.84배(45.7% 시간 단축) 더 빠른 실제 처리 속도를 달성하였다.

이는 배열의 크기가 커질수록 스캔 체인 길이가 늘어나는 STRAIT와 달리, BICS-BIST는 배열 크기에 영향을 받지 않는 병렬 주입 구조를 가지므로 대규모 시스템일수록 그 효율성이 더욱 극대화될 것임을 시사한다.

3-3. 결합 탐지 신뢰성 검증

제안하는 BICS-BIST 구조의 신뢰성을 입증하기 위해 앞서 설계한 세 가지 결합 주입 시나리오에 대한 시뮬레이션을 수행하였다. 표 2는 그 결과를 요약한 표다.

실험 결과, BICS-BIST는 단일 PE의 고장(Case 2-A) 뿐만 아니라, 특정 행(Row)이나 열(Column) 전체에 걸쳐 다발적인 오류가 발생하는 상황(Case 2-B, 2-C)에서도 모든 결함을 정확하게 탐지(DETECTED)하고 최종 결과로 Fail 신호를 출력함을 확인하였다.

특히 Case 2-B (Row Burst Fault)의 성공적인 검출은, 각 PE의 1-bit Fail Flag가 시프트 레지스터를 통해 수직 방향으로 전파되는 과정에서 데이터 충돌 없이 정상적으로 동작함을 증명한다. 또한 Case 2-C (Column Burst Fault)의 검출 성공은, 다수의 에러가 동시에 발생하여 최종 출력단의 Bitwise-OR 로직에 입력되더라도, 에러 신호(Logic High)가 소실되지 않고 확실하게

최종 에러 카운터에 반영됨을 시사한다.

더 나아가 Case 3 (Transition Delay Fault) 실험을 통해, at-speed 동작 시 신호가 제시간에 도착하지 못하는 지연 결함까지 성공적으로 포착함을 확인하였다.

이를 통해, BICS-BIST 구조가 SA, TD Fault 모두 검출할 수 있는 높은 신뢰성을 갖추었음을 실험적으로 입증하였다.

4. 실험 결과 해석 및 Trade-off 분석

본 연구의 실험 결과를 종합해 볼 때, 제안하는 BICS-BIST 구조는 하드웨어 자원(Area)과 동작 주파수(Speed)에서의 손실(Penalty)을 감수하고, 테스트 처리량(Throughput)과 신뢰성(Reliability)을 극대화하는 공학적 Trade-off를 명확히 보여준다.

가장 주목할 부분은 단순 하드웨어 동작 속도 (F_{max})는 기존 STRAIT 구조가 약 19.7% 더 빠르지만, 실제 전체 테스트를 완료하는 데 걸리는 Execution Time은 BICS-BIST가 약 1.48배 더 빠르다는 것이다. 이는 STRAIT 구조가 데이터를 한 비트씩 순차적으로 주입하는 비효율적인 Scan-Shift 동작에 많은 시간을 허비하는 반면, BICS-BIST는 아키텍처 레벨에서의 병렬성을 극대화하여 불필요한 대기 시간을 제거했기 때문이다.

또한, 각 PE 내부에서 수행되는 In-Situ Comparison 기법의 도입은 테스트 데이터의 Output Bandwidth 효율성을 획기적으로 개선하였다. 기존 STRAIT 구조는 결합 검출을 위해 32-bit의 연산 결과를 모두 Scan-out해야 했으므로, 배열 크기와 데이터 비트 수에 비례하여 스캔 시간이 급증하는 병목 현상이 존재했다. 반면, BICS-BIST는 PE 내부에서 즉시 비교를 수행하고 1-bit의 Fail Flag만을 캡처하여 Scan-out한다. 이는 스캔 체인을 통해 이동해야 하는 데이터의 양을 1/32로(32-bit 데이터 기준) 대폭 감소시키는 효과를 가져왔으며, 결과적으로 Test Latency 단축과 스캔 전력 소모 감소에 크게 기여한 것으로 분석된다.

결과적으로 본 연구는 하드웨어의 단순 클럭 속도보

다는 아키텍처의 구조적 효율성이 시스템의 전체 테스트 성능에 더 결정적인 영향을 미칠 수 있음을 입증하였다. 한편, 실험 결과에서 면적 및 전력 오버헤드가 크게 측정된 주원인은 FPGA 합성 과정에서 ROM이 최적화되지 않은 LUT 기반 로직으로 구현되었기 때문이다. 실제 상용 칩(ASIC) 설계에서는 이 부분이 고밀도 SRAM이나 외부 메모리 인터페이스로 대체될 것으로, 제어 로직 자체의 순수 오버헤드는 본 실험 결과보다 현저히 낮을 것으로 판단된다.

V. 결론

본 연구에서 제안한 BICS-BIST 구조는 기존 STRAIT 구조의 구조적 제약 사항을 개선하고, Test Efficiency를 극대화하는 데 중점을 두었다. 가장 큰 장점은 테스트 시간의 획기적인 단축에 있으며, 테스트 데이터를 모든 PE에 동시에 주입하는 병렬 데이터 주입 방식을 채택함으로써 Test Latency를 획기적으로 45.7% 단축하고, 시스템의 유휴 시간을 최소화하는 효과를 가져왔다. 이러한 속도 이점은 STRAIT 구조가 직렬 스캔 방식의 고유한 특성으로 인해 시간 소모를 필연적으로 수반하는 Scan-Shift 동작에 비해, BICS-BIST 가 아키텍처 레벨에서의 병렬성을 극대화하여 불필요한 대기 시간을 제거했기 때문에 달성되었다.

BICS-BIST는 또한 Scan Bandwidth 최적화를 성공적으로 달성하였다. 이는 In-Situ Comparison 방식을 통해 PE 내부에서 32-bit 연산 결과를 1-bit Fail Flag로 압축하여 출력하기 때문에 가능하다. 구조적으로 BICS-BIST는 복잡한 Scan Chain 제어 로직 없이 Address Generator와 Comparator 기반의 간결하고 직관적인 구조로 설계되었으며, 1-bit Fail Flag 스캔 방식은 기존 STRAIT의 Diagnosis Logic Chain과 호환되므로, 기존의 결함 위치 추적 알고리즘을 그대로 활용할 수 있다는 강점을 가진다.

실험을 통해 제안된 구조가 하드웨어 자원의 증가(Area)를 수반함에도 불구하고, 테스트 시간을 획기적으로 단축시키고 고속 동작 시의 지연 결합까지 효과적으로 탐지할 수 있음이 입증되었다. 이는 하드웨어의 물리적 한계를 아키텍처 레벨의 병렬성과 데이터 압축 효율성으로 극복한 사례로서, 제안 구조의 실효성을 확인하였다.

하지만 본 연구의 한계점을 보완하고 실용성을 높여 차세대 AI 반도체 시스템의 테스트 기술 발전에 기여하기 위해 다음과 같은 후속 연구가 필요하다. 첫째, 메모리 최적화 연구이다. 현재 LUT로 구현된 ROM을 FPGA 내부의 Block RAM(BRAM)이나 외부 DRAM 인터페이스로 대체하여 면적 효율을 개선하는 설계가 요구된다. 둘째, 대규모 확장성 검증이다. 본 연구의 16x16 배열을 넘어 64x64, 128x128 이상의 대규모 Systolic Array에 적용할 경우 발생할 수 있는 Fan-out 및 신호 지연(IR-Drop 등) 문제를 해결하기 위해, 전체 배열을 여러 구역(Zone)으로 나누어 순차적으로 테스

트하는 분할 테스트 기법 도입을 고려해야 한다.

결론적으로, BICS-BIST 아키텍처는 인공지능 가속 기의 핵심인 Systolic Array의 신뢰성을 확보하기 위해, 기존의 직렬 스캔 방식의 구조적 제약을 해소한 병렬 입력 기반 BIST 구조의 타당성을 입증하였다. 이는 향후 고성능·고신뢰성이 요구되는 차세대 AI 반도체 시스템의 테스트 기술 발전에 기여할 수 있을 것으로 기대된다.

참고문헌

- [1] R. Xu, S. Ma, Y. Guo, and D. S. Li, "A Survey of Design and Optimization for Systolic Array-based DNN Accelerators," *ACM Comput. Surv.*, vol. 56, no. 1, pp. 1-37, Aug., 2023.
- [2] H. Lee, J. Kim, J. Park, and S. Kang, "STRAIT: Self-Test and Self-Recovery for AI Accelerator," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 9, pp. 3092-3104, Sept., 2023.
- [3] U. S. Solangi, M. Y. Wako, and S. M. Park, "Test Architecture for Systolic Array of Edge-Based AI Accelerator," *IEEE Access*, vol. 9, pp. 96700-96710, Jul. 2021.
- [4] J. Kim, H. Lee, and S. Kang, "A New Zero-Overhead Test Method for Low-Power AI Accelerators," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 71, no. 5, pp. 2649-2653, May 2024.
- [5] P. R. S. and N. M. S., "BIST Design and Implementation for a Fixed-Point Arithmetic MAC Unit within a Systolic Array," *2023 IEEE International Conference on Signal Processing, Computation and Control (ISPCC)*, Solan, India, pp. 1-6, May 2023.
- [6] Q. Zhang, C. Liu, B. Liu, H. Huang, Y. Wang, H. Li, and X. Li, "Cross-Layer Optimization for Fault-Tolerant Deep Learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43, no. 1, pp. 120-133, Jan. 2024.
- [7] Z. Zhang, K. Basu, and R. Karri, "Fault-Tolerant Systolic Array Based Accelerators for Deep Neural Network Execution," *IEEE Design & Test*, vol. 36, no. 5, pp. 44-53, Oct. 2019.
- [8] C. De Sio, S. Azimi, and L. Sterpone, "Dependable DNN Accelerator for Safety-Critical Systems: A Review on the Aging Perspective," *IEEE Transactions on Emerging Topics in Computing*, vol. 11, no. 2, pp. 320-333, April-June 2023.
- [9] Y. H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks,"

- IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127-138, Jan. 2017.
- [10] N. P. Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit," *Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA)*, Toronto, ON, Canada, pp. 1-12, Jun. 2017.
- [11] M. Capra, B. Bussolino, A. Marchisio, G. Masera, M. Martina, and M. Shafique, "Hardware and Software Optimizations for Accelerating Deep Neural Networks: Survey of Current Trends, Challenges, and the Road Ahead," *IEEE Access*, vol. 8, pp. 225134-225180, Nov. 2020.
- [12] S. Mittal, "A Survey on FPGA-based Accelerators for Convolutional Neural Networks," *Neural Computing and Applications*, vol. 32, no. 4, pp. 1109–1139, Feb. 2020.



김 가 혼

2021년~현재 인하대학교 전기전자공학부 재학 중. 관심분야는 디지털 회로 설계 및 검증.



장 준 영

2019년~현재 인하대학교 전기전자공학부 재학 중. 관심분야는 디지털 시스템 설계 및 임베디드 S/W.

인하대학교
카피킬러캠퍼스 표절 검사
결과 확인서

지도교수	본인
성명	성명
서명	서명

이메일	12211733	표절률	3%
성명	자필로 기재하세요		
소속	전기전자공학부		
검사번호	00337269070		
검사명	2025년 2학기 전기전자종합설계 논문		
문서명	전기전자종합설계_김가현_장준영_최종.hwp		
비교범위	[현재첨부문서] [카피킬러 DB]		
평가 유형	<input checked="" type="checkbox"/> 카피킬러 <input checked="" type="checkbox"/> GPT킬러	발급형태	<input checked="" type="checkbox"/> 기본보기 <input checked="" type="checkbox"/> 요약보기 <input checked="" type="checkbox"/> 상세보기
발급일자	2025.12.08 20:10	검사일자	2025.12.08 20:07
비고			

평가 설정

인용/출처 포함 문장	검사 대상 제외	법령/경전 포함 문장	검사 대상 제외	목차/참고문헌	검사 대상 제외
-------------	----------	-------------	----------	---------	----------

표절기준

어절	6	문장	1
----	---	----	---

검토 의견

검토 의견을 입력하는 대형 텍스트 필드입니다.

본 확인서는 12211733 사용자가 카피킬러에서 표절검사를 수행한 표절분석 결과에 대한 문서로 카피킬러 표절 시스템이 자동으로 생성한 자료입니다. 문서 작성 기준이 각 학교, 기관마다 다르므로 최종 평가자의 표절평가 결과와는 다를 수 있습니다.

분석 정보

표절률	전체 결과	동일 문장	의심 문장	GPT 의심 문장	인용/출처	법령/경전
3%	23	1	22	0	0	0

비교 문서 정보

번호	표절률	출처정보
1	1%	[카피킬러 DB] Copykiller <ul style="list-style-type: none"> 파일명 : 대학IT교육경쟁력강화_대학IT전공역량강화_NEXT사업(IT학과) 저자 : 오범환 발행 : 2006
2	1%	[카피킬러 DB] Copykiller <ul style="list-style-type: none"> 파일명 : 대학IT교육경쟁력강화_대학IT전공역량강화_NEXT사업(IT학과) 저자 : 오범환;김덕경;김주희 발행 : 2006-12
3	1%	[카피킬러 DB] ice.inha.ac.kr <ul style="list-style-type: none"> 파일명 : 이규홍 - 인하대학교 정보통신공학부
4	1%	[카피킬러 DB] blog.naver.com <ul style="list-style-type: none"> 파일명 : 엔비디아 CUDA가 이끄는 AI 혁명: HW-SW 융합 : 네이버 블로그
5	1%	[카피킬러 DB] ice.inha.ac.kr <ul style="list-style-type: none"> 파일명 : 별지 서식 7 - 인하대학교 정보통신공학부
6	1%	[카피킬러 DB] blog.naver.com <ul style="list-style-type: none"> 파일명 : [영진전문대 컴퓨터정보계열] ■ 11 장 - Layout Manager 발행 : naver
7	1%	[카피킬러 DB] eschdo.blog.me <ul style="list-style-type: none"> 파일명 : [영진전문대 컴퓨터정보계열] ■ 11 장 - Layout Manager 발행 : naver
8	1%	[카피킬러 DB] Copykiller <ul style="list-style-type: none"> 파일명 : Analysis of dynamic H₂¹⁵O PET data by factor analysis and independent component analysis 저자 : 안지영 발행 : 서울 : 이화여자대학교 대학원, 2004.8
9	1%	[카피킬러 DB] cafe.naver.com <ul style="list-style-type: none"> 파일명 : 성추행수사 혐의 중대한 사건에서 어떻게 살아남을 수 있을까요? 저자 : 변호사 즉시상담 발행 : 법률사무소 영웅 - 형사, 성범죄, 이혼, 상속, 민사, 부동산 법률카페, 20241026022835
10	1%	[카피킬러 DB] www.copykiller.com <ul style="list-style-type: none"> 발행 : 중앙대학교, 2025.04.
11	1%	[카피킬러 DB] Copykiller <ul style="list-style-type: none"> 파일명 : 시추공 탄성파 초동주시 기록의 정보정 연구 저자 : Lee, Doo-Sung 발행 : 2011, vol.14, no.2, pp. 146-151 (6 pages)
12	1%	[카피킬러 DB] www.copykiller.com <ul style="list-style-type: none"> 발행 : 중앙대학교, 2025.04.
13	1%	[카피킬러 DB] www.helilot.net <ul style="list-style-type: none"> 파일명 : [3D 스캐너] 무작정 따라 하면 왕초보도 3D 전문가! - 산업 전문 미디어 He....

14	1%	[카피킬러 DB] www.yna.co.kr <ul style="list-style-type: none"> 파일명 : 여소야대 극복 印尼 조코위, 공무원 100만명 감축 개혁드라이브 연합 ... 발행 : www.yna.co.kr
15	1%	[카피킬러 DB] Copykiller <ul style="list-style-type: none"> 파일명 : A Study on the Static Correction for the First Arrival Travel-time of the Cross-well Seismic Data 저자 : Doo Sung Lee 발행 : 2011
16	1%	[카피킬러 DB] m.hello7.net <ul style="list-style-type: none"> 파일명 : [3D 스캐너] 무작정 따라 하면 왕초보도 3D 전문가! - 첨단
17	1%	[카피킬러 DB] blog.naver.com <ul style="list-style-type: none"> 파일명 : 원자 물질의 화학 반응: 실험과 과학적 측정의 중요성 : 네이버 블로그
18	1%	[카피킬러 DB] Copykiller <ul style="list-style-type: none"> 파일명 : 컴퓨터 교과 수업을 위한 문제중심학습 사이트의 구축 저자 : 임명희 발행 : 춘천 : 강원대학교 교육대학원, 2007.2
19	1%	[카피킬러 DB] ice.inha.ac.kr <ul style="list-style-type: none"> 파일명 : 별지 서식 8 - 인하대학교 정보통신공학부
20	1%	[카피킬러 DB] ice.inha.ac.kr <ul style="list-style-type: none"> 파일명 : 저작권 양도 서정보통신공학부장 귀하 - 인하대학교 정보 ...
21	1%	[카피킬러 DB] ice.inha.ac.kr <ul style="list-style-type: none"> 파일명 : 별지 서식 7 - 인하대학교 정보통신공학부
22	1%	[카피킬러 DB] Copykiller <ul style="list-style-type: none"> 파일명 : 경미한주취소란자에대한효율적대응방안연구 저자 : 김호철 발행 : 2008-09-01~2011-02-28
23	1%	[카피킬러 DB] Copykiller <ul style="list-style-type: none"> 파일명 : 중화권 주식시장의 동조화 현상 분석 = Analysis of co-movements among the greater China stock markets (다운로드) 저자 : 호연 중화권 주식시장의 동조화 현상 분석 = Analysis of co-movements among the greater China stock markets
24	1%	[카피킬러 DB] blog.naver.com <ul style="list-style-type: none"> 파일명 : 브랜드자산 및 성과 진단 모델에 관한 실증 연구
25	1%	[카피킬러 DB] Copykiller <ul style="list-style-type: none"> 파일명 : 도심지 내 중·저층 주요시설물(학교, 병원 등)의 응급복구 내진보강기술 개발 저자 : 권민호 발행 : 2014.11
26	1%	[카피킬러 DB] www.copykiller.com <ul style="list-style-type: none"> 발행 : 숭실대학교 대학원, 2014.04.
27	1%	[카피킬러 DB] www.copykiller.com <ul style="list-style-type: none"> 발행 : 숭실대학교 대학원, 2024.09.
28	1%	[카피킬러 DB] www.copykiller.com <ul style="list-style-type: none"> 발행 : 숭실대학교 대학원, 2024.10.
29	1%	[카피킬러 DB] www.copykiller.com <ul style="list-style-type: none"> 발행 : 중앙대학교, 2024.11.

검사 문서

문장표절률: 38%

저작권 양도서 (Copyright Transfer Form) 소속 : 인하대학교
전기전자공학부 논문제목 : 고효율 Systolic Array 테스트를 위한
병렬 패턴 주입기법 저자1(성명, 학번) : 김가현, 12211733

문장표절률: 100%

본 논문의 내용은 저자가 직접 연구한 결과인 것과 이전에 출판된 적
이 없음을 확인합니다.

문장표절률: 56%

또한 공저자와 더불어 인하대학교 전기전자공학부에서 **발간하는 논**
문집에 본 논문을 수록하는 것을 허락하며 제반 저작권을 전기전자
공학부에 양도합니다.

문장표절률: 42%

대표적인 예로 Google TPU(Tensor Processing Unit)는
Systolic Array 기반의 행렬 곱셈 유닛을 활용하여 딥러닝 연산을
효율적으로 수행한다.

비교 문서

[ice.inha.ac.kr] 별지 서식 8 - 인하대학교 정보통신공학부

(별지 서식 7) 저작권 양도서 (Copyright Transfer Form) 소속
: 인하대학교 정보통신공학부 성명 : 학번 : 논문제목 : 본인은

[ice.inha.ac.kr] 이규홍 - 인하대학교 정보통신공학부

저작권 양도서 (Copyright Transfer Form) 소속 : 인하대학교
정보통신공학부 저자(들) 이름 : 이

[ice.inha.ac.kr] 별지 서식 7 - 인하대학교 정보통신공학부

본 논문의 내용은 저자가 직접 연구한 결과인 것과 이전에 출판된 적
이 없음을 확인합니다

[ice.inha.ac.kr] 이규홍 - 인하대학교 정보통신공학부

본 논문의 내용은 저자가 직접 연구한 결과인 것과 이전에 출판된 적
이 없음을 확인합니다

[ice.inha.ac.kr] 별지 서식 7 - 인하대학교 정보통신공학부

또한 공저자와 더불어 인하대학교 정보통신공학부에서 **발간하는 논**
문집에 본 논문을 수록하는 것을 허락하며 제반 저작권을 정보통신
공학부에 양도합니다.

[Copykiller] 대학IT교육경쟁력강화_대학IT전공역량강화_NEXT사업
(IT학과)

저자 : 오범환

발행 : 2006

또한 공저자와 더불어 인하대학교 정보통신공학부에서 **발간하는 논**
문집에 본 논문을 수록하는 것을 허락하며 제반 저작권을 정보통신
공학부에 양도합니다. 년 월 일

[blog.naver.com] 엔비디아 CUDA가 이끄는 AI 혁명: HW-SW 융합
: 네이버 블로그

엔비디아의 시장 점유율을 위협하고 있습니다. **대표적인 예로,**
Google은 TPU(Tensor Processing Unit)를 통해 자사의 클라
우드 서비스에서

[Copykiller] [논문]인공지능의 에너지 효율화와 엣지 컴퓨팅

저자 : 김말희 허태욱 이일우

발행 : 2025

높은 성능과 전력 효율을 제공한다. **Google TPU(Tensor**
Processing Unit)는 텐서 연산을 가속하여 AI 모델

문장표절률: 27%

하지만 Systolic Array는 수많은 Processing Element(PE)로 구성되어 있어, 하나의 PE에 발생한 결함이 전체 연산 결과에 심각한 영향을 미칠 수 있다.

[[blog.naver.com](#)] 원자 물질의 화학 반응: 실험과 과학적 측정의 중요성 : 네이버 블로그

수 있는 뜻밖의 사고는 실험 결과에 심각한 영향을 미칠 수 있다 그러한 사고 중 하나가 바로

문장표절률: 27%

이에 따라, 하드웨어 가속기의 신뢰성을 보장하기 위해 각 구성요소의 정상동작 여부를 시스템 내부에서 스스로 검증할 수 있는 Built-In Self-Test(BIST) 기술의 중요성이 부각되고 있다.

[[cafe.naver.com](#)] 성추행수사 혐의 중대한 사건에서 어떻게 살아남을 수 있을까요?

저자 : 변호사 즉시상담

발행 : 법률사무소 영웅 - 형사, 성범죄, 이혼, 상속, 민사, 부동산 법률카페, 20241026022835

않을 수 있으며, 이는 사건의 결과에 심각한 영향을 미칠 수 있다.

[[www.copykiller.com](#)] 1101244042 김용호 A built-In self-Test scheme for 3D RAMs.hwp

발행 : 숭실대학교 대학원, 2014.04.

는 고장난 메모리 셀을 찾을 수 있는 Built-in self test(BIST)로직과 수리가능한 여분 메모리를 분석하고

[Copykiller] 디지털 오버샘플링을 이용한 1Gbps 데이터 복원 회로 =

저자 : 박준영

발행 : 2000

Chip안에서 Test를 수행할 수 있는 Built In Self Test 에 대한 연구가 필요하며 DLL

[Copykiller] 벤처산업의 성과와 과제 - 부문별 주요 논점 및 제도개선 과제 -

저자 : EH9KUybj5/gp0zRreM9pSg==

발행 : 2002

과거 대기 업 위주의 성장 구조의 한계를 극복할 수 있는 새로운 희망으로 떠올랐습니다. 이러한 변화에 부응하여

[Copykiller] 환율과 기초여건 간 관계에 대한 연구: 시장심리를 중심으로

저자 : btO7J+mcgrcYshVuVoo5Jw==

발행 : 2021

연구는 제한적이다. 2. 연구의 의의 및 구성 본 연구의 목적은 기존 정형화된 환율 예측모형과 더불어 외환시장의

문장표절률: 41%

본 논문의 구성은 다음과 같다. 제1장에서는 연구의 필요성과 목적을 제시하고, 제2장에서는 Systolic Array와 기존 STRAIT BIST 구조의 기술적 특징 및 한계를 분석한다.

[Copykiller] 중화권 주식시장의 동조화 현상 분석 = Analysis of co-movements among the greater China stock markets (다운로드)

저자 : 호연 중화권 주식시장의 동조화 현상 분석 = Analysis of co-movements among the greater China stock markets

본 논문의 구성은 다음과 같다. 제1장에서는 연구의 필요성과 목적을 밝히고 논문의 구성에 대해 기술한다

[Copykiller] 컴퓨터 교과 수업을 위한 문제중심학습 사이트의 구축

저자 : 임명희

발행 : 춘천 : 강원대학교 교육대학원, 2007.2

점이 있다. 4. 논문의 구성 본 논문의 구성은 다음과 같다. 제1장에서는 연구의 필요성과 목적, 연구 내용 및 방법, 연구의

[kportal.kipris.or.kr] 네임 기반 인-네트워크 프로세싱 방법 및 시스템

이때, 함수의 러닝 인스턴스는 필요한 데이터를 전달받아 연산을 수행한 후 그 결과를 유저에게 전달할 수 있다. 상술한

[www.a-ha.io] 전자 계산기의 작동 원리가 궁금합니다 | 궁금할 땐, 아하!

미리 정해진 계산 순서에 따라 연산을 수행한 후 그 결과를 다시 우리가 알아볼 수 있는 십진수로

[blog.naver.com] 엔비디아 CUDA가 이끄는 AI 혁명: HW-SW 융합 : 네이버 블로그

엔비디아의 시장 점유율을 위협하고 있습니다. 대표적인 예로, Google은 TPU(Tensor Processing Unit)를 통해 자사의 클라우드 서비스에서

[Copykiller] 영상 데이터 전송을 위한 LPDDR2 메모리 컨트롤러의 설계

저자 : 강진수 영상 데이터 전송을 위한 LPDDR2 메모리 컨트롤러의 설계

이 외에도 메모리 컨트롤러의 BIST(Built In Self Test)를 위한 모듈, 메모리 컨트롤러의 셋

[www.pinpointnews.co.kr] 엑시콘, 후공정 검사 장비로 실적 안정화 성장 동력 동시에 확보 < 증권·IB < 금융·증권 < 기사본문 - 핀포인트뉴스

형태의 저장장치를 대상으로 기능 테스트와 BIST(Built In Self Test)를 수행, SSD의 정상 작동

문장표절률: 15%

Figure 2. BIST structurediagram of STRAIT [2] 2. STRAIT
기반 BIST 구조 Systolic Array는 수많은 PE로 구성되어 있고, 이들 각각은 MAC 연산을 수행하는 핵심 단위이기 때문에, 하나의 PE라도 오작동할 경우 전체 행렬 연산 결과에 심각한 영향을 미칠 수 있다.

[[blog.naver.com](#)] 원자 물질의 화학 반응: 실험과 과학적 측정의 중요성 : 네이버 블로그

수 있는 뜻밖의 사고는 실험 결과에 심각한 영향을 미칠 수 있다 그러한 사고 중 하나가 바로

[[cafe.naver.com](#)] 성추행수사 혐의 중대한 사건에서 어떻게 살아남을 수 있을까요?

저자 : 변호사 즉시상담

발행 : 법률사무소 영웅 - 형사, 성범죄, 이혼, 상속, 민사, 부동산 법률카페, 20241026022835

않을 수 있으며, 이는 사건의 결과에 심각한 영향을 미칠 수 있다.

문장표절률: 29%

이에 따라 Systolic Array의 신뢰성을 확보하기 위해 결함 탐지를 위한 내장자가 테스트 기법인 **BIST(Built-In Self-Test)**의 적용이 필수적이다.

[[blog.naver.com](#)] [공유] SoC 설계시 고려사항 3

발행 : naver

그러한 예의 하나로 칩내부의 설계검증으로 **BIST(Built-In Self-Test)**의 On-Chip Test 나 바운더리

[[patents.google.com](#)] KR100187871B1 - 다수의 메모리를 검사하기 위한 비스트(bist)검사기 및 검사방법 - Google Patents

다수의 메모리를 동시에 검사하기 위한 **BIST(Built-In Self-Test)**의 상위 레벨 블럭도. 제2a

[[www.globalwellpcba.com](#)] 인서킷 테스트란 무엇인가? - GlobalWellPCBA

따라 핀이 마모되거나 정렬이 틀어질 수 있으며, 이는 테스트의 정확도에 영향을 미칩니다. 시스템이 올바르게 작동하는지 확인하려면

문장표절률: 20%

또한, shift-in 시 데이터가 각 PE에 동시에 주입되지 않기 때문에, 테스트 중간 단계에서의 예측 불가능한 출력이 발생할 수 있으며, 이는 테스트 정확도에 영향을 줄 가능성도 존재한다.

[[www.andongmin.com](#)] 서버와 클라이언트 관계 이해 - 안동민 개발 노트

네트워크 지연, 서버 처리 시간 등 다양한 요소가 전체 성능에 영향을 미칩니다. 최근에는 서비스 아키텍처와 같은

문장표절률: 18%

특히, 기존의 TP_A와 TP_W 경로뿐만 아니라 P 경로의 입력을 위한 테스트 벡터(P_in)가 추가되어, 무작위 패턴으로는 검증하기 힘든 연산의 Corner Case와 타이밍 마진을 더욱 효과적으로 테스트 할 수 있다.

[[Copykiller](#)] Analysis of Chinese Consumers' Intention to Engagement in Taobao Metaverse Shopping through S-O-R Model

저자 : 리우밍위에 Pansoo Kim

발행 : 2024

등과 같은 다양 한 측면을 더욱 효과적으로 테스트 할 수 있다 (Hair et al., 2010). 설문지에서

문장표절률: 29%

따라서 시스템의 최대 동작 주파수를 제한하는 Critical Path인 MAC 유닛을 직접 타격하여 검증하는 방식만으로도 고속 동작에 대한 신뢰성을 충분히 확보 할 수 있다.(a)

[Copykiller] 도심지 내 중·저층 주요시설물(학교, 병원 등)의 응급복구
내진보강기술 개발

저자 : 권민호

발행 : 2014

발휘하고, 외부의 환경조건에 관계없이 결속력에 대한 신뢰성을 충분히 확보 할 수 있다. 또한 간편한 시공으로 실제

문장표절률: 27%

또한, 실제 칩의전원 라인이나 클럭 트리 문제로 발생할 수 있는 결함을 반영하여, 특정 행(Row)이나 열(Column)의 다수 PE가 동시에 오작동하는 Multi-Burst Fault를 모사하였다.

[Copykiller] 도심지 내 중·저층 주요시설물(학교, 병원 등)의 응급복구
내진보강기술 개발

저자 : 권민호

발행 : 2014.11

발휘하고, 외부의 환경조건에 관계없이 결속력에 대한 신뢰성을 충분히 확보 할 수 있다. 또한 간편한 시공으로 실제

[eschdo.blog.me] [영진전문대 컴퓨터정보계열] ■ 11 장 - Layout

Manager

발행 : naver

열(column)의 경우도 마찬가지이다. 특정 행(row)이나 열(column)의 weightx, weighty 값이 '0'으로 column)은 컨테이너의 크기가 변해도 행(row)이나 열(column)의 폭이나 높이가 변하지 않는다. 보통의

[blog.naver.com] [영진전문대 컴퓨터정보계열] ■ 11 장 - Layout

Manager

발행 : naver

열(column)의 경우도 마찬가지이다. 특정 행(row)이나 열(column)의 weightx, weighty 값이 '0'으로 column)은 컨테이너의 크기가 변해도 행(row)이나 열(column)의 폭이나 높이가 변하지 않는다. 보통의

문장표절률: 33%

3. 실험 결과 및 비교 본 절에서는 기존 STRAIT 구조와 제안하는 BICS-BIST 구조의 성능을 정량적으로 비교 분석한다.

[Copykiller] 3차원 영상의 시점변화와 악의적 공격에 대한 디지털 워터마킹 = Digital watermarking on 3D image for view-point change and malicious attacks

저자 : 김보라

발행 : 2015

s experiment results 5-2. 실험 결과 및 비교 본 절에서는 제안한 방법과 기존의 방법에 대해

[Copykiller] 3차원 영상의 시점변화와 악의적 공격에 대한 디지털 워터마킹 = Digital watermarking on 3D image for view-point change and malicious attacks

저자 : 김보라

발행 : 2015

s experiment results 5-2. 실험 결과 및 비교 본 절에서는 제안한 방법과 기존의 방법에 대해

문장표절률: 14%

실험 결과, BICS-BIST는 단일 PE의 고장(Case 2-A)뿐만 아니라, 특정 행(Row)이나 열(Column) 전체에 걸쳐 다발적인 오류가 발생하는 상황(Case 2-B, 2-C)에서도 모든 결함을 정확하게 탐지(DETECTED)하고 최종 결과로 Fail 신호를 출력함을 확인하였다.

[eschoo.blog.me] [영진전문대 컴퓨터정보계열] ■ 11 장 - Layout Manager

발행 : naver

열(column)의 경우도 마찬가지이다. 특정 행(row)이나 열(column)의 weightx, weighty 값이 '0'

문장표절률: 30%

결과적으로 본 연구는 하드웨어의 단순 클럭 속도보다는 아키텍처의 구조적 효율성이 시스템의 전체 테스트 성능에 더 결정적인 영향을 미칠수 있음을 입증하였다.

[blog.naver.com] [영진전문대 컴퓨터정보계열] ■ 11 장 - Layout Manager

발행 : naver

열(column)의 경우도 마찬가지이다. 특정 행(row)이나 열(column)의 weightx, weighty 값이 '0'

문장표절률: 29%

하지만 본 연구의 한계점을 보완하고 실용성을 높여 차세대 AI 반도체 시스템의 테스트 기술 발전에 기여하기 위해 다음과 같은 후속 연구가 필요하다.

[Copykiller] 2017년 상반기 연구·실험·기술

세메스 제품의 기술 발전에 기여하기 위해 다음과 같은 역량을 키워 왔습니다. 1. 다양한 분야에서의

문장표절률: 41%

이는 향후 고성능·고신뢰성이 요구되는 차세대 AI 반도체 시스템의 테스트 기술 발전에 기여할 수 있을 것으로 기대된다 .

[Copykiller] 2017년 상반기 연구·실험·기술

세메스 제품의 기술 발전에 기여하기 위해 다음과 같은 역량을 키워 왔습니다. 1. 다양한 분야에서의

[www.copykiller.com] 디지털 계측 기반 철근 간격 실시간 모니터링 및 View Synthesis를 활용한 정밀 측정 고도화_도서관업로드.pdf

발행 : 숭실대학교 대학원, 2024.12.

건설 품질 및 안전 관리 기술의 발전에 기여할 수 있을 것으로 판단된다.

[www.copykiller.com] 14. 학위논문양식들(국문)(A4)_목차 본문 _20240927.hwp

발행 : 숭실대학교 대학원, 2024.09.

실시간 데이터 분석 및 처리 기술의 발전에 기여할 수 있을 것으로 기대된다.