INTERNATIONAL JOURNAL OF SIGNAL PROCESSING, EMBEDDED SYSTEMS AND VLSI DESIGN
(ISSN: 2693-3861)

Volume 05, Issue 01, 2025, pages 10-34
Published Date: - 09-05-2025
Doi: -https://doi.org/10.55640/ijvsli-05-01-03

Design-for-Test (DFT) strategies for high-performance computing and graphics chips

Vikas Nagaraj
MTS at Advanced Micro Device(AMD), San Jose, California, USA

ABSTRACT

With the architecture complexity of silicon in high-performance computing (HPC) and graphics processing units (GPUs) growing, reliability, scalability, and first-time-right silicon cannot be achieved without the introduction of advanced Design for Test (DFT) methodologies. This paper addresses the peculiarities of DFT magnetization to cope with the characteristics of HPC and GPU environment issues: massive parallelism, depth pipelining, multi-clock, power domains, and rising thermal and power density. It covers basic techniques, including scan-based testing, built-in self-test (BIST), logic BIST (LBIST), and a modular and hierarchical test planning framework. Additionally, the paper studies the related key infrastructural pieces, such as test access mechanisms (IJTAG, IEEE 1500), remote debug orchestration, and centralized test control units. Additionally, emerging trends like AI/ML-enabled ATPG, in-field telemetry, predictive maintenance, and DFT innovations in the contexts of chipset-based and 3D-integrated architecture alter the test requirements for the overall multi-die system. It provides best practices in early DFT planning, modular IP reuse, scan chain optimization, and power-aware test pattern generation to obtain high test coverage while maintaining silicon performance. This work presents actionable insights for high-yield silicon design and validation in the next-generation compute platform landscape. It is aimed at silicon architects, DFT engineers, and verification professionals.

KEYWORDS

Design-for-Test (DFT), High-Performance Computing (HPC), GPU Validation, Built-In Self-Test (BIST), Semiconductor Test Automation

INTRODUCTION

Introduction to Design-for-Test (DFT) in HPC and Graphics Chips

Design for Test (DFT) has become an inescapable discipline in semiconductor engineering, owing to the increased complexity and scale of integrated circuits (ICs) with billions of transistors. In HPC and GPU contexts, DFT lies at the cutting edge of the functional correctness of chips, fault tolerance, and production viability for chips that operate at the edge of speed, power, and concurrency. Tests of these modern chips (which commonly appear in data centers, AI training, and rendering engines) require robust test strategies that can reveal subtle defects without sacrificing their time to market and yield targets.

DFT can be considered as an attempt to enhance a chip's testability on the architecture of the chip, that is, in the absence of the fabrication process. Using scan chains, boundary scan cells, BIST modules, and compression logic, the engineers can target previously unreachable design features, such as those associated with testing the board with external automated test equipment (ATE) or embedded diagnostic tools. Without these mechanisms, it would be both costly and inefficient as well as practically impractical to verify at a nanometer process node like 5nm or 3nm. Due to the power and thermal noise, new fault types are introduced with shrinking geometries of transistors, bridging faults, open circuits, and transient errors. High coverage of these faults can be detected on a fine scale with precise and scalable DFT strategies.

HPC and GPU architectures both uniquely complicate the design and testing problems. Unlike general-purpose CPUs, massive parallelism, hundreds to thousands of concurrent execution threads, deeply pipelined compute engines, and distributed memory systems are incorporated. In addition, they operate over many domains of clock and voltage frequency, often exceeding several gigahertz. While essential for performance, this architectural richness poses significant difficulties for test vector generation, fault propagation, and observability. In addition, as the chipset packaging becomes the standard and interconnects like PCIe, HBM, and GDDR become advanced, the test strategy should cover signal integrity and fault coverage across interfaces.

Thermal considerations make the picture even more difficult. Power consumption of HPC and GPU chips is significant, and temperature-related failures like electro migration, time-dependent dielectric breakdown (TDDB), or thermal runaway may not lead to failure if no operating stress is applied. Thermal awareness testing and monitoring of DFT is becoming essential. On top of that, the huge amounts of data center silicon require test strategies that achieve the highest possible throughput while sacrificing as little coverage or defect detection accuracy. Unfortunately, failure due to undetected faults in the field can result in catastrophic failures, expensive product recalls, and loss of reputation, specifically when chips are employed in mission-critical AI infrastructure or real-time visualization systems. This article presents a comprehensive search of DFT strategies tailored for high-performance computing and GPU chips. Fundamental principles and traditional techniques are introduced and deep-dived into architecture-specific challenges, memory test strategy, scan-based methods, and modular DFT frameworks. It also explores how innovations such as hierarchical testing, debug infrastructures, and first-time-right design philosophies transform the DFT landscape.

Fundamentals of DFT Methodologies

Table 1: ***Key DFT Techniques and Methods***

Technique	**Description**
Scan Chains	Serially connected flip-flops for testing, turning sequential circuits into combinational ones for easier testing.
Built-In Self-Test (BIST)	Embedded pattern generators and analyzers for internal testing without external access.
Boundary Scan (JTAG)	Provides standardized serial access for testing, especially when direct probing is impractical.

Core DFT Principles: Controllability and Observability

The backbone of reliable silicon validation and silicon production yield optimization is the Design for Test (DFT) methodology (Jiang et al., 2021). These methodologies are not monolithic; they have evolved given the complexity of integrated circuits, moving from simple test hooks to well-connected, system-aware infrastructures. The underlying ideas of all DFT strategies are based on controllability and observability. The notion of controllability means that an internal circuit node can be set to a desired logical state during testing, and observability is the ability to monitor the internal state through observable outputs from a circuit under test. These principles guarantee robust stimulation and detection of faults, even within deeply embedded blocks of the functional block.

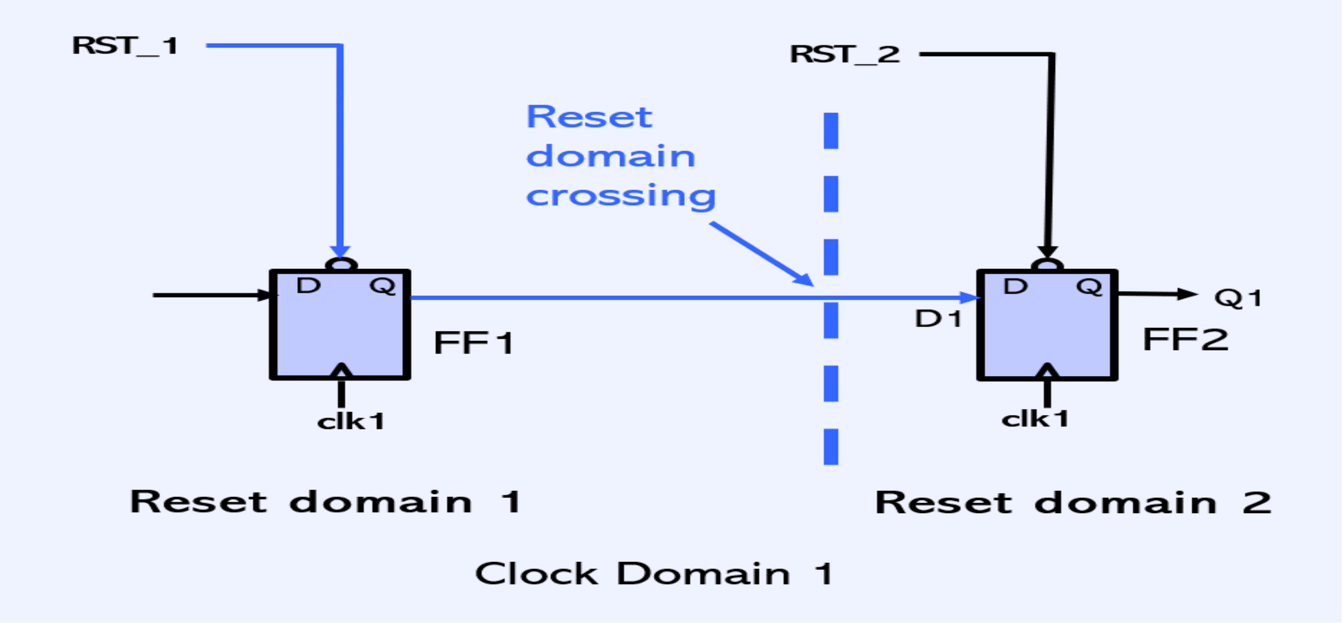

Figure 1: Reset Domain Crossing Sign-Off

Scan Chains, BIST, Boundary Scan

Scan design is one of the earliest and most commonly used DFT techniques in which digital design flip-flops are restructured to form a serially connected chain of scan test structures. These flip-flops shift the test patterns in and out during test mode; therefore, a sequential circuit is converted to a combinational one during the test mode for test monitoring and analysis. However, full-scan designs maximize the test coverage at the expense of the area and power overhead. However, the techniques in partial scan reduce the resources consumed by backing out only a subset of the design, but at the expense of reduced fault observability. Structural testing approaches like stuck-at-fault testing, transition fault testing, and path delay analysis could be performed using scan-based testing for nanometer-scale devices.

An important aspect of DFT, particularly when external tests are not feasible, is the Built-In Self-Test (BIST) technique. This is possible since BIST allows a chip to test its functionality using embedded pattern generators (e.g., Linear Feedback Shift Registers) and signature analyzers. For example, logic BIST (LBIST) targets combinational and sequential logic, while memory BIST (MBIST) focuses on RAM and ROM blocks. These mechanisms are particularly relevant to remote systems, mission-critical applications, and high-volume production, where external test access may be difficult or expensive. BIST also supports power-on self-test (POST), and this growing need applies to safety-critical sectors such as automotive and aerospace (Abotbol et al., 2022).

IEEE 1149.1, known as JTAG (commonly called boundary scan testing), provides a standardized serial test access mechanism from which test data can be shifted through a device's boundary scan cells. As such, this technique is invaluable when board-level testing is needed and allows for probe access to devices that would otherwise be physically non-probeable due to advanced packaging and dense integration. JTAG is also a multipurpose DFT interface with an accompanying capability to provide device programming, field diagnostics, debugging, and other functions.

Evolution of DFT with Technology Scaling and SoC Complexity

Traditional DFT methods alone are insufficient as the complexity of the System Chip (SoC) design increases, and compression is increasingly used in modern chips to reduce the area and time penalties for full scan implementation. Test compression reduces test data volume, which is based on reducing the amount of test data on a chip through on-chip decompressors and response compactors. Embedded Deterministic Test (EDT) and Xpress compression increase ATE efficiency by decreasing the number of patterns while improving fault coverage, which aligns with the principles highlighted in dual sourcing strategies aimed at optimizing resource use and system reliability (Goel & Bhramhabhatt, 2024).

Technology scaling has also influenced DFT. As process nodes decline to 5nm and below, fault models must become more complex to include such subtle physical defects as bridging faults, resistive opens, and even more complex layout-dependent anomalies. Low power design techniques such as power gating and clock gating bring in new obstacles, including untestable paths and false fault detection; hence, power-aware DFT insertion and validation become necessary. Also, the DFT challenges of multi-voltage domains and dynamic frequency scaling require the DFT solutions to vary and meet the industry requirements across various conditions. At the same time, have minimum coverage and minimum overhead. As the integration of CPUs, GPUs, accelerators, and memory moves toward heterogeneous integration, which includes the components on a single die or chipset, the necessity for modularity and hierarchy of DFT has been emphasized. Test wrappers are employed in such an environment to standardize interfaces and isolate faults in IP Blocks. The wrappers make the reuse of IP easier, simplify the integration effort, and allow for core-level debugging independent of system-level mapping. The DFT methods continue to be developed in conjunction with technological developments of semiconductors (Oba & Kumagai, 2018). These techniques are now crucial for the availability of high-level, reliable, testability, and manufacturability required on today's HPC and GPU silicon. Hierarchical and compressed DFT architectures and power integrate efficiently enough to allow high yield and first-time-right silicon for designs of any complexity.

Architecture-Specific Testing Challenges in HPC and GPUs

The most complex silicon platforms in modern computing are high-performance computing (HPC) and graphics processing units (GPUs). These chips are designed to provide enormous computational throughput and integrate vast numbers of execution units, entire memory systems, and rich interconnects. These features achieve performance but do so at the expense of testing challenges that are more involved than the conventional design for the test (DFT). As a result, it is necessary to develop specialized strategies in this environment that are tuned to the nature of parallelism, clocking, and thermal dynamics in such architectures for extremely high test coverage, at-speed validation, and defect isolation.

Table 2: ***DFT Challenges in HPC and GPUs***

Challenge	**Description**
Deep Pipelines & High Concurrency	Managing test coverage across multiple threads and deeply pipelined architectures.
Multi-Clock Domains & Synchronization	Handling timing and synchronization issues across different clock speeds and domains.
Thermal Faults & Power Density	Addressing transient faults and thermal issues in power-dense environments.

Dealing with Deep Pipelines and High Concurrency

The defining characteristic of HPC and GPU architectures is their tremendously high levels of concurrency (Cini & Yalcin, 2020). For example, GPUs may be built with tens of thousands of ALUs executing threads in parallel over hundreds of thousands across multiple cores and warps. Like HPC chips, HPC chips have deeply pipelined vector engines and parallel instruction streams. However, these are also exactly the architectural traits that can make functional validation in post-silicon testing very undesirable. The exponential growth in state space complexity makes traditional ATPG (Automatic Test Pattern Generation) tools struggle to handle such parallelism. Moreover, the timing of the fault detection depends on how the fault is propagated through multiple concurrent pipeline paths, and structural failure in one part of the pipeline can appear subtly through various paths. To cope with this, DFT has to be designed carefully to use localized scan chains, modular BIST, and isolation mechanisms that isolate one execution unit against faults in other execution units—an approach that mirrors the strategies used in handling real-time big data challenges through scalable solutions (Dhanagari, 2024).

The high pipelining degree also creates additional latency between the input stimulus and the observable output. This leads to the introduction of delays in fault observation windows, and such means for capture are required to be time-aligned. Pipeline integrity must be preserved during scan mode, i.e., back pressure must not occur, hurdles must not happen, and hazards and deadlocks must not affect the test pattern shifting. Therefore, DFT logics for these architectures typically include enhanced control FSMs and hold logic to control pipelined state transitions during test application.

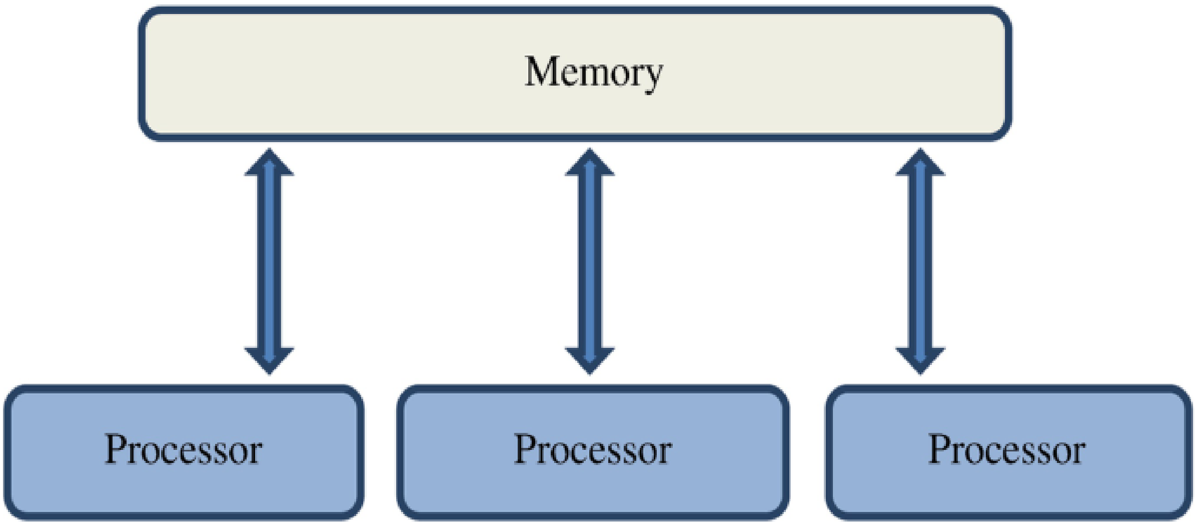

*Figure 2: **High-Performance Computing***

Multi-Clock Domains and Synchronization Errors

The proliferation of clock domains is another major obstacle to testing HPC and GPU chips (Tiwari et al., 2015). The designs operate at different frequencies to achieve each functional block's highest possible power and performance. To be specific, memory controllers or interconnects May work simultaneously at the compute cores, which run at 2.5 GHz, or they may be asynchronous or work at lower rates. The traditional cross or flip-flop may cause synchronization problems, detestability risks, and timing-related failures, which cannot be covered through conventional logic tests.

Due to the nature of the multiple asynchronous clock domains, each CDC insertion must be done, especially from a DFT perspective. These may include handshake synchronizers, FIFOs, and metastability capture elements that can be observed under test mode. Moreover, timing faults, for example, hold violations or setup failures, manifest only under real operating frequencies, and at such frequencies, at-speed testing is critical. The issues are addressed with clock-gating-aware scan insertion and the use of launch-on capture or lead-on shift testing. These enable on-the-fly validation of signal timing at speed while keeping scan_CHAIN observability. Additionally, on-chip clock generation and phase-locked loops (PLLs) have to be tested or skipped in scan mode to ensure deterministic timing.

Power Density Issues and Thermal Fault Detection

HPC and GPU chips have high power density, producing high thermal gradients over the die. Variations may produce transient fault patterns in data that only appear under one heat stress or peak current draw. For example, such occurrences are electro migration in metal interconnects, thermal runaway, and time-dependent dielectric breakdown (TDDB) in gate oxides. These fault types are presented, which are difficult to detect with conventional cold test methods (Kim & Katipamula, 2018).

This has to be overcome, and the importance of non-thermal-aware DFT strategies is growing. The built-in thermal sensors distributed across the die are monitored during burn-in or power-on self-tests. Other DFT strategies even help to incorporate power-aware ATPG, the generation of test patterns that mimic real workload-induced switching activity to provide real power and thermal relation case situations during validation. Also, dynamic voltage and frequency scaling (DVFS) has additional challenges related to variability in circuit timing and power profile due to the continuous switching in the presence of DVFS. DFTs must be robust over operating points and can identify temperature-sensitive faults without causing false positives. Factory tests supplement field testing mechanisms, such as error logging and predictive failure analytics, which capture latent defects not seen in the early stages of screening.

Scan-Based DFT for Complex Logic Blocks

Digital logic validation using Scan-based Design for Test (DFT) techniques continues to be an important evaluation methodology and, in particular, is critical for testing sophisticated functional blocks, such as arithmetic units, control logic, and interconnect fabrics, that are prevalent in large high-performance computing (HPC) and graphics processing unit (GPU) chips. The technique is a fundamental concept of scan-based testing that has been used for decades. Still, its implementation in recent deep submicron, high-concurrency environments has required a lot of evolution. Due to these chips' increasing complexity, integration density, and timing sensitivity, scan insertion, scan compression, and scan architecture optimization become imperative to derive effective, low-cost, and high-coverage testing.

Table 3: ***Scan-Based Testing Methods***

Method	**Description**	**Pros**	**Cons**
Full Scan	Converts most flip-flops into scan cells for complete testability.	High fault coverage.	Increased area, power, and timing overhead.
Partial Scan	Only a subset of flip-flops are in the scan chain.	Reduced overhead and resource consumption.	Reduced fault coverage

Full-Scan vs. Partial-Scan Methodologies

The essence of scan-based testing is turning flip-flops into scan cells and connecting them to the chain of serial shift registers. In a full scan design, almost all of the flip-flops are turned into scan cells; hence, there is complete controllability and observability of internal states. Such an approach simplifies the generation of tests along with their corresponding high fault coverage using the automatic test pattern generation (ATPG) tools. For example, it is especially useful in logic-heavy subsystems such as execution units, schedulers, and decode/dispatch logic for GPUs and HPC cores, where integrating robust and secure practices—akin to those applied in CI/CD pipelines with security tools like SAST, DAST, and SCA—can enhance design reliability (Konneru, 2021).

Full-scan techniques, however, cost. Such extra mux logic at each flip-flop entails area, timing, and power overhead (Sontakke & Dickhoff, 2023). Scan insertion can also disturb timing closure and increase clock loading, even in highly pipelined designs. However, to alleviate these issues, designers may use partial scan methodology where only a part of the flip flop is placed in the scan path, those flip flops are usually located in areas difficult to control or observe. A partial scan reduces area and power cost, but reduces the fault coverage and complicates the ATPG because of the remaining sequential logic. The choice between full and partial scans based on design architecture, timing margins, power budget, and test coverage goals requires a complicated tradeoff analysis. For HPC and GPU chips, fault coverage expectations beyond 98% full scan are still preferred for critical computing blocks. In contrast, the partial scan can be used for peripheral or low-risk logic.

Scan Compression (EDT, Xpress, and Other Techniques)

Test data volume and test time have become major bottlenecks as design sizes have ballooned into the billions of gates with the use of traditional scan-based testing (Cheng & DeGiorgio, 2020). The limitation of these is what reasonable modern chips mitigate using scan compression techniques, reducing the number of bits by which test patterns are applied and captured. This helps improve test efficiency and reduces testing time and cost for automated test equipment (ATE). Embedded Deterministic Test (EDT) is one of the most widely used methods. It utilizes on-chip decompressors to decompress test patterns into full scan inputs. It allows high fault coverage at a substantially lower cost per bit. Likewise, Xpress compression and other proprietary methods provide robust test compression ratios and support complex clocking and power domains.

Scan compression often involves output response compactors, which combine the output responses of one or more scan chains to the observation points (Janicki et al., 2020). Therefore, these compactors must be designed to avoid aliasing, i.e., multiple faults generate the same signature. Advanced techniques included in VFLIP, such as MISR (Multiple Input Signature Register) based compaction, are used to achieve accuracy without sacrificing throughput. Furthermore, the compression logic must be designed carefully to be compatible with data, power, and clock domain separation. It must be carefully coordinated with timing constraints and routing congestion between DFT architects and physical design teams, since scan paths must not be scan-accessible.

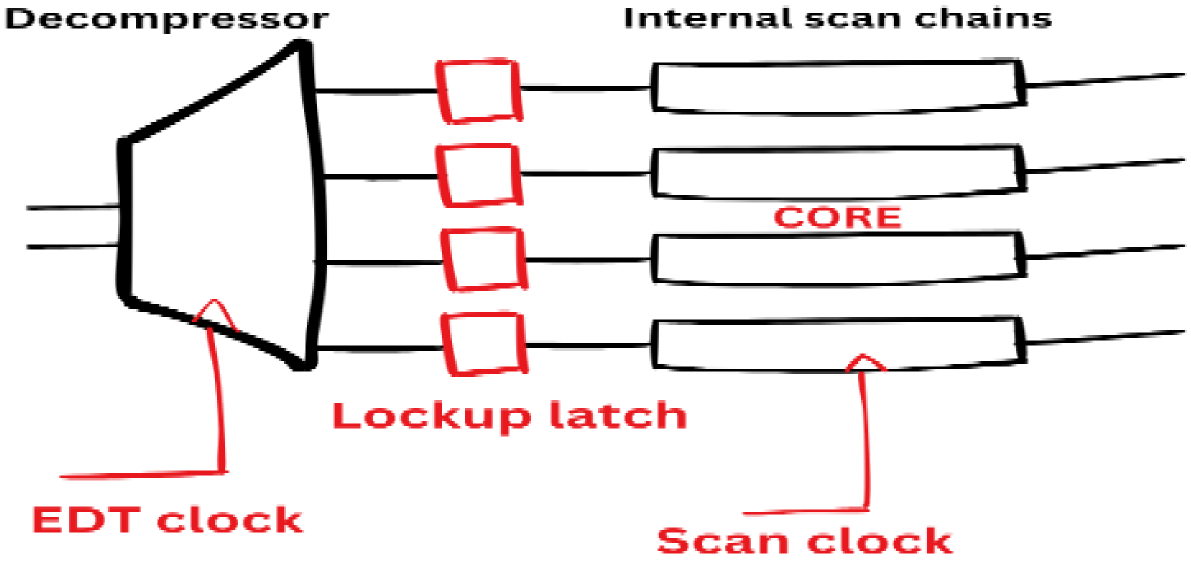

*Figure 3: **Test Compression***

Performance, Area, and Timing Impact

Much needs to be known about the performance-critical paths in the architecture in order to integrate scan logic into HPC and GPU designs. Scan muxes introduce timing delays in latency-sensitive pipelines, such as floating point units or load/store queues, which impact cycle time. As a result, the static and dynamic switching and timing violations in the scan enable signals, control FSMs, and clock gating logic must be reduced. From the silicon real estate point of view, scan logic and compression consume some additional percentage points. This may be fine for smaller chips, but as the die size goes up, with every square millimeter considered valuable in HPC and GPU dies, DFT architects have to minimize the overhead through sophisticated scan chain partitioning and hierarchical scan insertion.

Another important issue is the amount of power consumed during scan testing. Scan shifting is likely to activate many flip-flops at a time and, therefore, can generate large IR drops, ground spikes, or thermal spikes. Scan patterns generated by power-aware ATPG tools do not necessarily switch activity to values that severely affect the switching power, and the scan chain balancing technique evenly distributes the switching activity over logic blocks—similar in concept to how scheduled and balanced system interventions can improve outcomes in critical environments (Sardana, 2022). Timing closure also requires that scan paths of designs at or above 2 GHz GHz do not introduce additional critical paths or race conditions (Sur et al., 2016). By aligning scan architecture with floorplan and clock tree synthesis objectives, risks like these can be mitigated through the use of tools like DFT-aware place and route and scan reordering.

Memory DFT and Redundancy Schemes

Memory structures are an order of magnitude or more silicon real estate in HPC and GPU architectures. They include small register files to large embedded SRAMs, high bandwidth DRAM interfaces, and custom cache arrays. Memory testing and redundancy management are crucial in a full DFT strategy since memory devices are so dense and sensitive to manufacturing defects. Memory faults, with the soft error being one of them, bridging being another, or cell leakage also not last, can cause catastrophic system errors. Hence, robust memory DFT is a must on the first silicon.

Table 4: ***Memory DFT and Redundancy Techniques***

Technique	**Description**	**Challenges**
March Algorithms	Test memory arrays with address-ordered sequences (e.g., March C, March SS).	The complexity of testing dynamic RAMs like HBM and DRAM.
Memory BIST (MBIST)	Automates the memory testing process using built-in test engines.	Managing complex memory topologies and fault isolation.
Redundancy & ECC	Uses spare memory cells and error-correction codes to handle faulty memory.	Managing memory defects in large, high-density systems.

Testing Static and Dynamic Memory Arrays

Embedded memories present unique challenges in testing as they have a regular structure, low external observability, and high cell density. Conversely, memories must be tested using techniques that can reach every word line, bit line, and sense amplifier path, and memory testing is no exception. Static Random-Access Memories (SRAMs) that are the basis of cache hierarchies and register files are typically state-of-the-art tested with March algorithms: a class of address-ordered test sequences providing detection of stuck-at, coupling, transition, and address decoder faults (Nyati, 2018). Depending on the fault coverage and timing constraints, commonly used variants include March C, March B, and March SS. An embedded controller executes these algorithms, which apply read/write operations to certain sequences' memory cells, supporting the broader goal of fault tolerance in complex, event-driven architectures (Chavan, 2024).

Due to refresh requirements and sense amplifier timing, testing dynamic RAMs (DRAMs), including high-bandwidth memory (HBM) used in HPC and AI accelerators, is inherently more difficult. Furthermore, DRAM testing must verify row and column decoding, charge retention, and access timing, all typically done during manufacturing tests but now supported by built-in test (BIST) capabilities in 2.5D and 3D stacked configurations (Wang et al., 2015). Other emerging memories, such as MRAM, ReRAM, and eDRAM, are also emerging, and some are starting to appear in HPC/GPU markets, but are still in the niche. They incorporate fault models for write disturbance, retention loss, and endurance degradation, all of which are utilized in modern ATPG and memory BIST techniques.

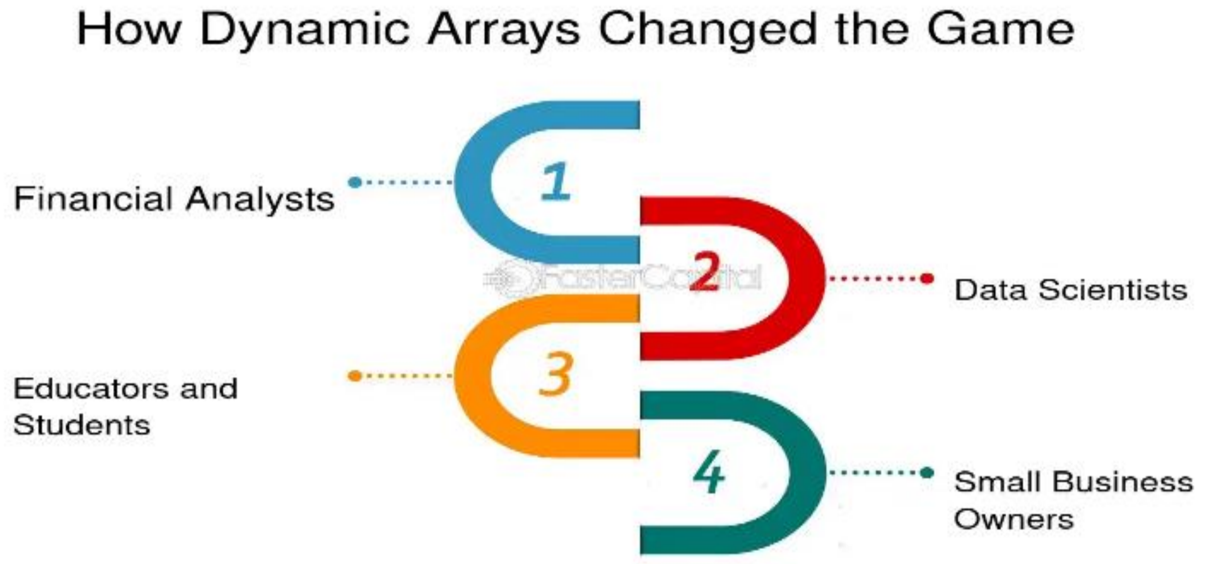

Figure 4: ***Dynamic Arrays: Dynamic Arrays in Excel***

March Algorithms and Memory BIST (MBIST)

Most modern designs use Memory Built-In Self-Test (MBIST) engines to automate and accelerate memory validation. The MBIST controller generates the March-based test sequences internally, interfaces with memory blocks, and compares output responses received from blocks to expected values. This approach significantly reduces application time and test costs and improves testing throughput since no significant external test vector storage is required.

MBIST architectures are very configurable, and engineers can configure the test complexity, number of iterations, and fault models depending on each memory instance's role and criticality. For example, if L1 caches and register files are closely tied to the control logic, these will be more highly tested, given that they will influence system stability. On the other hand, L2/L3 caches can be made larger and less constrained compared to the above test regime, especially if protected by Error Correcting Codes (ECC) able to correct single and a few multi-bit faults. Modern MBIST designs include programmability to choose different March algorithms and operating conditions, repair interfaces for redundancy fusing, and BIST chaining for sequential or parallel testing of multiple memory instances. In the HPC and GPU domains, where embedded memory arrays coexist hundreds of times, hierarchical MBIST insertions and intelligent chaining are necessary to reduce the test time and the overall power during the execution of the memory tests (Kong et al., 2021).

Redundancy Analysis and ECC Integration

Even for rigorous testing, manufacturing defects are still shown in memory arrays because of process variations, lithography limits, and random defect distribution. To address this, memory DFT often uses redundancy schemes to add spare rows and columns to the memory layout. However, any faulty cells identified during the test are joined with other spares through a redundancy fusing process, mapping out their location. Typically, redundancy analysis is done based on built-in logic to sense and log fail addresses, which are then used by repair tools to recode address decoders, re-cable, or recast multiplexor paths. Global repair data is stored in some architectures via programmable fuses (fuses or laser fuses), while in other cases, the repair data is stored in a set of shadow registers or reconfigurable logic, aligning with broader efforts to optimize system performance through intelligent architectures (Singh, 2022).

Hardware redundancy is also increasingly combined with Error-Correcting Codes (ECC)—for example, GPU memory subsystems and HPC caches. ECC can correct single-bit errors on the fly and detect multiple-bit errors. Possible implementations include SECDED (Single Error Correction, Double Error Detection) and more advanced BCH or LDPC codes in the high-reliability domains. When combined, BIST, redundancy, and ECC take a multi-layer approach to memory fault management (Vitucci et al., 2023). BIST provides rapid diagnosis/screening, redundancy for in-fabric defects repairs, and in-field fault tolerance for in-operation. System uptime is essential to mission-critical HPC applications, where mission-critical would also include data integrity and computational accuracy.

Built-In Self-Test (BIST) and Logic BIST (LBIST) Strategies

Built-In Self-Test (BIST) is one of the most powerful and versatile test techniques available in the Design for Test (DFT) arsenal. It allows semiconductor devices to perform internal or Built-in self-testing (BIST) to determine their integrity using resources within the device. BIST mechanisms become increasingly essential for silicon validation, manufacturing yield improvement, and in-the-field diagnostics when chip complexity, parallelism, and packaging density make it less feasible, or even unfeasible, to test externally in HPC or GPU environments (Marwala, 2024).

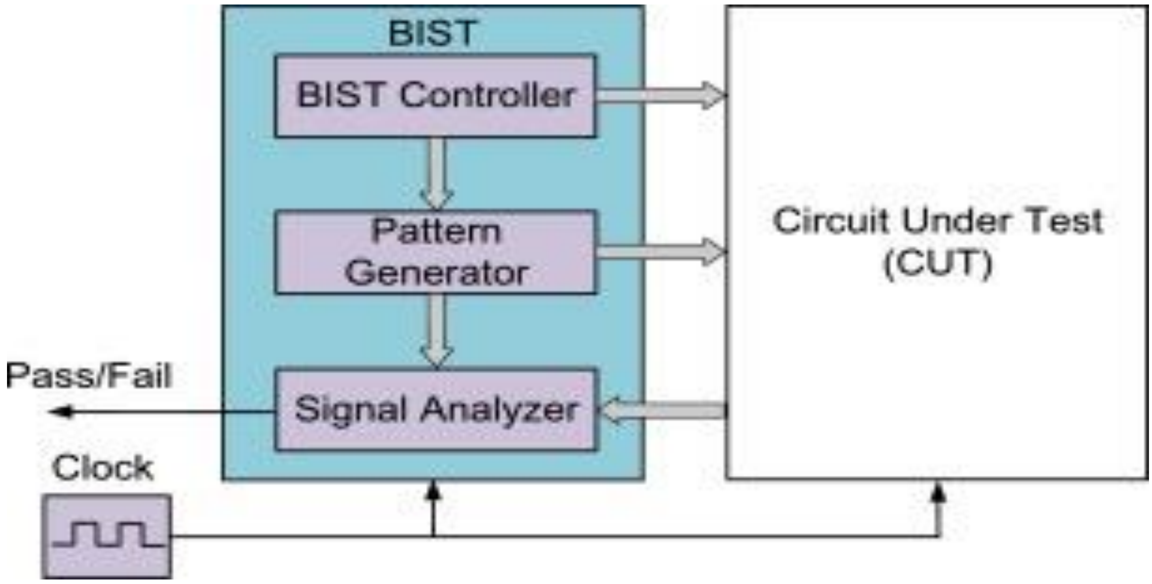

Figure 5: built-in self-test

Differentiating Structural and Functional BIST

They generally fall into the sphere of structural or functional BIST techniques. Structural BIST is dedicated to identifying the physical level defects in digital logic and memory structures using independent test paths (exercise). Typically, this type of circuit is implemented as Logic BIST (LBIST) or Memory BIST (MBIST), targeting, for example, stuck-at, transition, bridging, and path-delay types of faults. Structural BIST is needed to detect and incorporate manufacturing defects into the chip's RTL or gate-level design.

The difference here is that Functional BIST runs at the behavioral level using a high-level working load or microcode to check end-to-end data flow, instruction execution, and pipeline behavior. However, although it takes more time to execute and sometimes does not provide fine-grain fault localization, functional BIST complements the structural BIST by acting as a guarantor of all the interactions of system components, also working as intended. For example, functional BIST is helpful for power-on self-tests (power-on self-tests) or firmware-based diagnostics of GPU shader units, thread dispatch engines, and dataflow networks (Mazumdar, 2017). In general, both approaches are used in practice. Therefore, structural BIST offers coverage against manufacturing-related defects, whereas functional BIST offers runtime confidence in functional correctness, particularly in safety or mission-critical domains.

Application in ALUs, FPUs, Caches, and Control Logic

Logic BIST (LBIST) has been particularly useful in testing large blocks such as Arithmetic Logic Units (ALUs), Floating Point Units (FPUs), instruction schedulers, and Control Finite State Machines (FSMs) typical in high-performance computing (HPC) processors, and GPUs. Since these functional units have very high frequencies of operation and are prone to timing-sensitive faults and logic hazards, they offer themselves as at-speed testing candidates. It is common to have many such core components in a typical LBIST. Inside is a pseudo-random pattern generator (PRPG), usually an LFSR implemented for the test stimuli. A Multiple Input Signature Register (MISR) compacts the output responses from the logic under test to form a fault signature. Additionally, the control logic controls scan chain configuration and pattern sequencing for deterministic test pattern application and observation.

LBIST executes these patterns at functional speed to capture the timing-related defects, such as setup and hold violations, which static scan might not capture (Bhatelia, 2017). This is crucial to logic blocks that operate above 2–3 GHz, where minor variations in timing result in functional errors when the logic block operates under real conditions. Test reusability is also supported through LBIST's ability to be periodically activated in the field to monitor for aging-related degradation and emerging defects during the product's operational lifetime. However, MBIST is used to test caches and small memory arrays, whereas LBIST is critical to validate any parity and ECC that encircles those memories to assure that error detection and correction mechanisms will perform properly during access cycles. Furthermore, LBIST makes testing asynchronous control logic, such as state machines, instruction decoders, and interrupt controllers, very effective because they are hard to test exhaustively with standard functional testing. Testing for these components is event-driven, but LBIST can efficiently provide flexible and dynamic times.

Online BIST for Mission-Critical Runtime Diagnostics

With the increased deployment of chips in mission-critical applications (autonomous systems, data center AI accelerators, and scientific computing), the need for run-time (or online) BIST is growing substantially. Self-test operations are executable during a system's idle period, a scheduled maintenance window, or within a redundant failover configuration so that a constant level of hardware reliability is maintained during the device's operational life. Runtime BIST is designed as non-destructive to have scan chains, thus activated during low loads to minimize the impact on performance. Typically, it uses a segmented BIST architecture capable of testing individual partitions of the chip independently of each other and thus continues to operate the rest of the system while performing a test. The use of this granular testing approach enables a combination of comprehensive fault coverage of the diagnostic routines while being able to not interfering with mission-critical functions, echoing the tailored, non-intrusive strategies seen in adaptive AI-driven systems (Karwa, 2023).

Dynamic thresholding in multiple input signature registers (MISRs) is a key feature of online BIST. It provides for minor variation due to environmental effects, temperature, or voltage drift, allowing for the inability to distinguish between real faults and benign deviations. This increases accuracy (and avoids false positives) at runtime tests. Periodic runtime fault detection is a regulatory mandate, and the use case for which one of the most relevant online BIST applications is here—systems that must be recognized as meeting Safety Integrity Level (SIL) or Automotive Safety Integrity Level (ASIL) certification. In such applications, built-in diagnostics monitor system health continuously and can activate recovery actions, isolate bad modules, or perform reconfiguration processes automatically when some anomalies are detected. Runtime BIST is used in cloud service providers for their predictive maintenance strategy beyond safety-critical domains. They analyze self-test logs to look for early signs of wear or failure on components and can proactively replace or reroute workloads before errors altogether disrupt service. Taking these proactive steps in advance doesn't just improve system uptime and reliability; it also decreases the number of unplanned outages and minimizes the impact on service level.

Hierarchical and Modular DFT for Scalability

With HPC and GPU designs expanding into the multi-billion transistor range, SoC complexity now exceeds the levels that require a fundamentally hierarchical and modular Design for Test (DFT). However, in traditional flat DFT methods, the increase of manufacturers, the size, and the interconnect density of chips imply unacceptable test time, effort, and verification complexity. To address this challenge, two hierarchical and modular DFT techniques are used that support scalable, reusable, and integration-friendly low-cost test architectures suitable with current enabling chip partitioning schemes, including IP-based design and chipset-based architecture (Okasaka et al., 2016)

*Table 5: **Hierarchical DFT for Scalability***

Hierarchy Level	**Function**
Core Level	Verification of individual IP blocks using self-contained DFT logic (scan chains, BIST).
Cluster Level	Groups multiple cores/subsystems, shares common test access mechanisms.
System Level	Manages test mode configuration, scan distribution, and test sequencing for the entire die.

Core, Cluster, and System-Level Hierarchical Testing

Hierarchical DFT strategies partition the test infrastructure across multiple levels of the design hierarchy—core level, cluster level, and top system level. At the core level, each IP block (give shader core, alu cluster, or media engine) is equipped with all the self-contained DFT logic (scan chains, BIST engines, and test controller) at its core. Therefore, this approach for localizing independent verification of functional blocks and early test development is feasible even before full chip integration. In the typical use case of many cores or subsystems clustered at the test access hierarchy level, the cores or subsystems are grouped at the cluster level and share common test access mechanisms (cell-based test transport, scan chain routing hubs). Parallel or sequential testing of these clusters is possible to control power consumption and test scheduling.

At the system level, global DFT logic enables test mode configuration, scan enable distribution, compression control, and test sequencing at the entire die (Koenemann, 2018). Hierarchical DFT also simplifies test timing closure since each level can be validated independently from the others, significantly reducing the effort in managing full-chip scan insertion in a monolithic environment. This layered approach makes particular sense in chips with heterogeneous compute units, that is, chips having GPUs, CPUs, other AI accelerators, or network processors. By enabling independent test development and validation across each hierarchical level, such a platform allows engineers to speed up silicon bring-up and isolate defects rapidly during post-silicon debugging.

Reusability of IP Blocks in SoCs.

IP reusability is a cornerstone of SoC development, and modular DFT design promotes it at the level of individual IP modules. Many pre-verified intellectual property (IP) blocks (whether internal or from third parties) incorporate embedded DFT logic conforming to standardized test protocols. Allowing IPs to seamlessly integrate into the SoC without implementing the scan logic or test control infrastructure will facilitate the reuse of existing scan-enabled IPs in a SoC. To enable this, IP blocks are commonly wrapped with DFT test interfaces, including scan ports, BIST control pins, and status signals. They all adhere to standard protocols such as IEEE 1500 (for core-level tests) or IEEE 1687 (for instrument access) and are plug-and-play compatible within the SoC test environment.

Modular DFT also shortens the time to verify the design because test patterns for each IP can be generated, simulated, and validated individually. The parallel development cycles and the collaboration of distributed design teams are enabled. Modular DFT in production enables selective retesting of updated or re-spun IPs without re-generating full chip tests to save time and resources. Modular test planning also improves pattern reuse and instantiation for HPC and GPU systems, where hundreds of replicated compute tiles or memory banks must be tested. This saves storage and application overhead for test data.

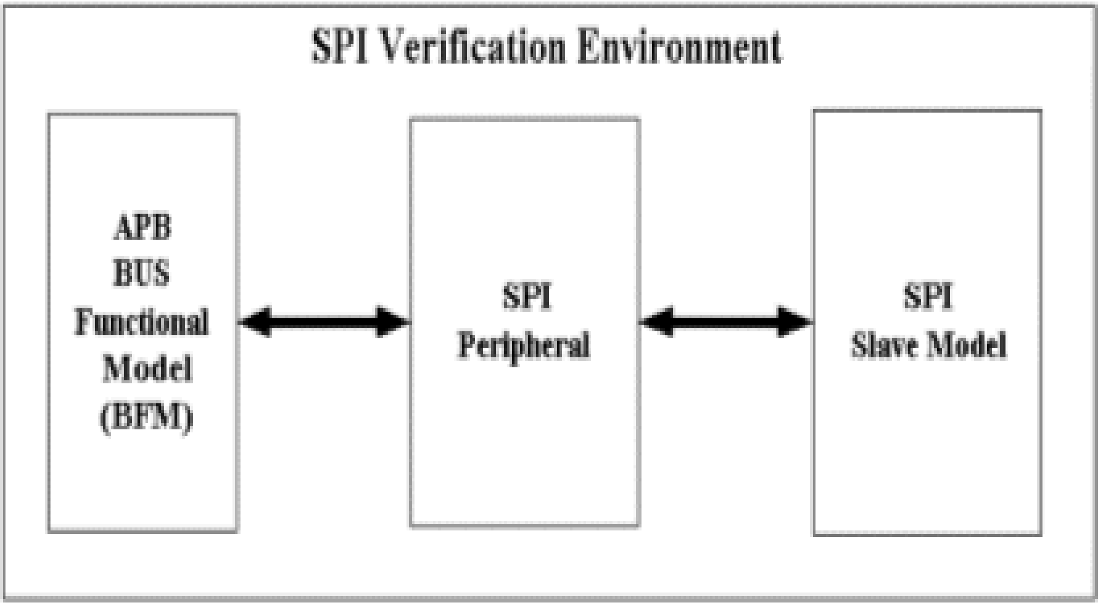

Figure 6: Developing a Reusable IP Platform

Plug-and-Play Test Wrappers for Scalable Integration

Plug-and-play test wrappers facilitate easy integration of modular intellectual property (IP) blocks into systems-on-chip (SoC) designs. These wrappers are abstraction layers that separate the overall SoC-level controller, which integrates the test with the internal Design for Test logic (DFT) in the module. These wrappers are translators between standardized interfaces and internal scan or BIST (Built-In Self-Test) configurations to simplify test integration between heterogeneous components. The main advantages of test wrappers are scan isolation for independent activating and deactivating scan paths, clock domain adaptation to safely provide asynchronous scan operations, and interface protocol bridging to unite different infrastructures of tests. With these wrappers, the scan multiplexers and isolation cells control the connectivity of the scan chains according to the selected test mode, and the clock domain crossing (CDC) adapters guarantee undistorted signal transitions from one domain to the other with different timing characteristics. Also, in addition to testing reconfigurable interfaces so that IP blocks run with multiple scan configurations, such as daisy chain or parallel scans based on the overall SoC test strategy,

A well-designed test wrapper allows for testing an IP block either as an independent module or within a complete test sequence without having to change the block's internal design. This flexibility proves advantageous in chipset architectures, which rely on rigorous testing of each die and a group of interconnected dies. These wrappers, die-to-die interface testing, boundary scan chaining, and inter-die BIST coordination are verified for multi-die systems (Gulve et al., 2022). In addition, late-stage product configuration is enhanced using plug-and-play test infrastructure, which provides a significant advantage in a highly diversified product family. For instance, consider various GPU SKUs that differ in the number of active compute cores or memory partitions. Wrapper-based Design-for-Test (DFT) techniques allow designers to selectively enable or restrict test access to these units at configuration time, thereby minimizing the need for changes to the underlying logic. This approach aligns with principles of dynamic system adaptation, as seen in memory inference models that manage resource allocation based on runtime contexts (Raju, 2017).

Test Access Mechanisms and Debug Infrastructure

The issue of embedding and efficiently accessing and controlling integrated circuit test logic becomes more difficult as integrated circuits scale in size and complexity across many computing species (e.g., high-performance computing and graphics processing units). Test architecture can be thought of as a system that adds scan chains and BIST modules, but, for example, inserting scan chains or BIST modules is not sufficient. The DFT architecture must feature robust, scalable test access mechanisms (TAMs) and debug infrastructure that can deliver test stimuli, capture test responses, and orchestrate complex test sequences when required. As such, they serve as a basis for both production testing and post-silicon validation and provide precise fault localization, speed of test execution, and low failure diagnosis delay.

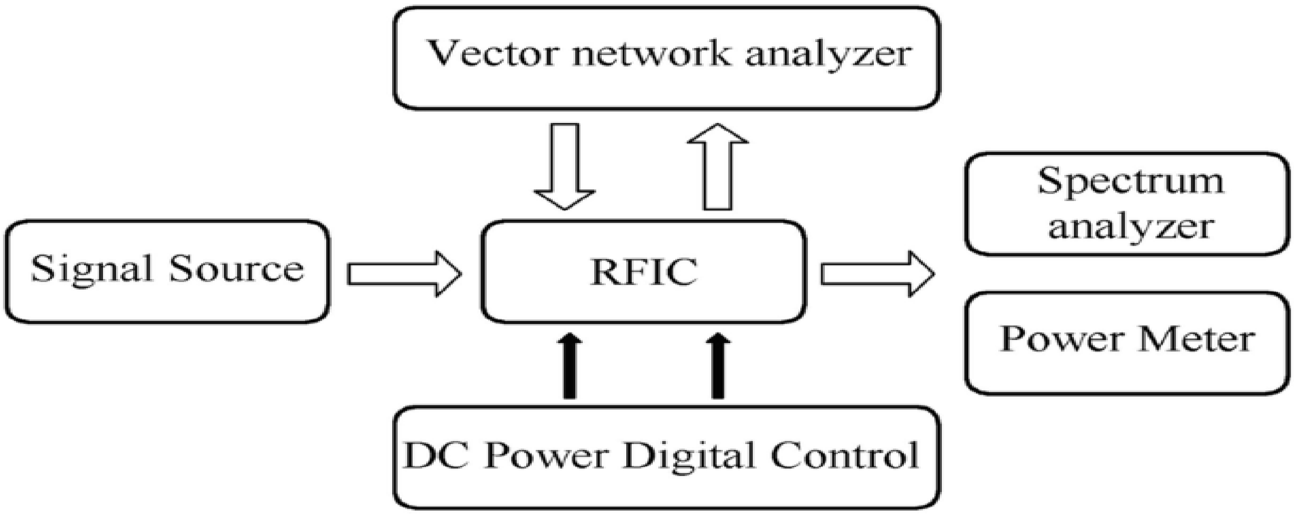

Figure 7: ***Integrated Circuit Testing Technology***

IJTAG (IEEE 1687) and IEEE 1500 Standards

Two key standards underpin modern TAMs: IEEE 1687 (IJTAG) and IEEE 1500. Since HPC chips and GPUs are modular and hierarchical, these frameworks enable structured access to embedded test and instrumentation logic. IEEE 1500 is a core-level test standard that embodies a wrapper architecture for an intellectual property (IP) block. Boundary scan cells, a test access port, and control logic that allows individual testing of an IP module are included in each wrapper. Test vector delivery via a wrapper serial port (WSP) or wrapper parallel port (WPP) supports IEEE 1500's flexibility for performance and area trade-offs. IEEE 1500 provides a means to enable parallel access to multiple test modules in an environment with numerous test modules, such as GPUs, without redesigning the core test infrastructure, which reflects the kind of modular and comparative approach also seen in evaluation frameworks such as those used for image captioning techniques.

Internal JTAG (IJTAG) extends JTAG by providing dynamic access to embedded monitors, sensors, and BIST engines in the IEEE 1687 form (Laisne et al., 2020). IJTAG allows a scalable instrument access network wherein test resources are connected using reconfigurable paths controlled by segment insertion bits (SIB). The hierarchical approach allows for the activation of instruments only when they are needed and thus minimizes fixed scan path overhead. These standards enable structured testing of replicated cores, voltage monitors, performance counters, and temperature sensors in HPC and GPU environments. They also facilitate software-driven validation, where embedded firmware or drivers initiate self-test routines during system bring-up or maintenance cycles. In high-stakes contexts such as surveillance and healthcare domains increasingly reliant on AI, such structured validation is vital to ensuring both performance and ethical integrity in public-facing systems (Singh, 2024).

Embedded Instrumentation and Scan Reuse

Continuous monitoring and testing of logic is referred to as embedded instrumentation. These include on-chip logic analyzers, performance counters, power monitors, and clock jitter detectors. Unlike conventional DFT logic, these instruments are active during functional and test modes and thus provide a uniform view of chip behavior under real workloads. Embedded instruments are integrated into the modern debug infrastructures into the IJTAG or custom buses, from which data can be collected and commands sent across software interfaces. During post-silicon validation, these instruments are critical when defects, such as race conditions, timing glitches, and hard-to-trigger faults, can only be seen at corner workloads.

Scanning chain reuse also maximizes the utility of existing DFT structures. During functional operation, scan cells are reused as data capture probes, with different scan patterns used at test and debug times. This reuse enables designers to monitor internal states with very little area overhead and without intrusive probe insertion. Scan reuse allows for event-triggered trace capture and retrospective analysis in GPUs for debugging shader pipelines or memory controllers that demand deep visibility. Built-in logic analyzers and compression units, combined with the gigabytes of debug data collected, can be filtered without overpopulating external interfaces for engineers.

Centralized Scheduling and Debug Orchestration

With the increase in the modularity and hierarchy of Design-for-Test (DFT) logic, it is essential to have centralized coordination to ensure efficient and reliable testing throughout the entire system-on-chip (SoC). Typically, this coordination is performed by a Test Control Unit (TCU) or DFT manager, coordinating key activities such as test sequence execution, scan chain selection, test clock generation, and power domain management. At the center of HPC and GPU SoC peripherals, the TCU interfaces to concurrent and sequential test engines, BIST engines, and the IJTAG (IEEE 1687) network.

Centralized scheduling is necessary to optimize test time and control power consumption during pattern application. The intelligent sequencing of test operations by the Test Control Unit (TCU) reduces the likelihood of IR drop and thermal violations by ensuring that high-power, high-switching functional blocks are not energized simultaneously. This approach aligns with strategies used in managing consistency in microservices, where careful orchestration is essential to avoid resource contention and ensure efficient data delivery (Chavan, 2021). Additionally, TCUs facilitate the sharing of test access buses without creating conflicts and ensure test data reaches the appropriate units at the correct time. Modern DFT infrastructures are also tightly coupled with debugging orchestration tools for test and test coordination (Kovács et al., 2024). Since these tools work with design verification environments, engineers can use them to initiate test routines, stop and control tests, and analyze failure data. Post-silicon is when one can diagnose design issues and assess the design's health quickly. This ability to correlate test outcomes with the simulation waveform is especially critical in reducing design time and improving yield.

Centralized orchestration even takes on greater importance in the contexts of chipset-based and 3D integrated circuit (3D-IC) architectures. Now, test and debug coordination must span interdie interfaces embedding through silicon vias (TSVs), interposers, and die-to-die links that require a new set of test requirements. Thus, these multi-die environments have synchronized cross-die scan chaining, timing calibration, and signal integrity validation dependency, all controlled by distributed debug units under centralized TCU management. In such systems, the TCU guarantees that each die operates within specified parameters to provide system-level awareness, maintainable data flow validation, and interconnect reliability.

First-Time-Right Silicon: DFT's Role in Tape-Out Success

First-time right (FTR) silicon is a technical goal and a business imperative in high-performance computing (HPC) and GPU silicon. For the design re-spin that produces a new mask set, cycles through the foundry, and involves engineering time, the cost can easily run into millions of dollars while also resulting in lost market opportunity. In this equation, however, Design for Test (DFT) plays a crucial role in minimizing the risk of latent or undetected defects in the fabricated silicon and ensuring it functions as intended. DFT contributes to advancing high-performance devices from design validation through post-silicon analysis to narrow down potential issues that become errors when they escalate into expensive failures, thereby minimizing yield ramp times, smooth bring-up, and fast time to volume.

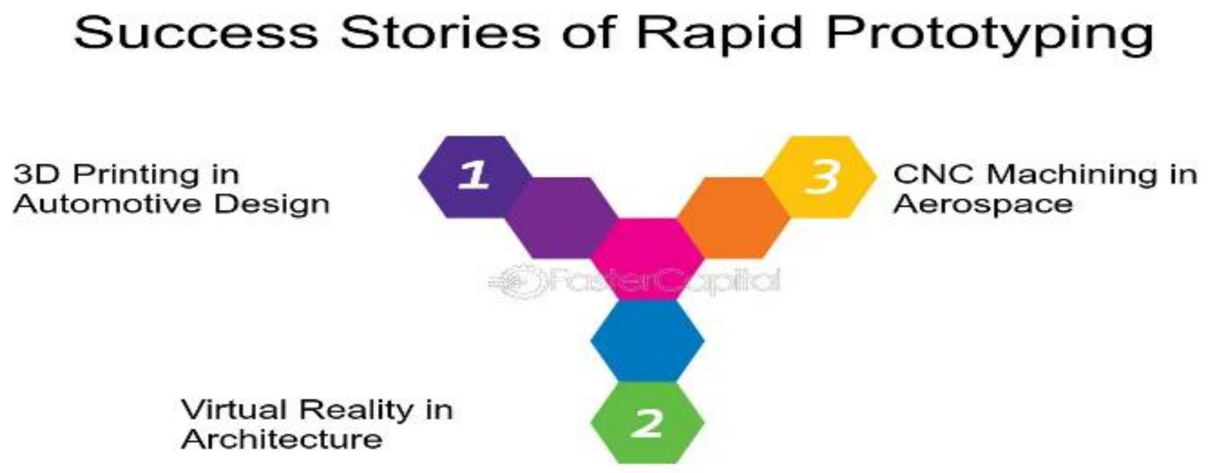
Success Stories of Rapid Prototyping
3D Printing in Automotive Design
1
3
CNC Machining in Aerospace
FasterCapital
Virtual Reality in Architecture
2

Figure 8: ***Time-saving Methods***

Increasing Yield through Better Fault Modeling

DFT's main contribution to FTR silicon is improving the manufacturing test coverage and yield analysis. Basic stuck-at and transition models for fault modeling have evolved into increasingly more elaborate representations, such as bridging faults, resistive opens, cell-aware faults, and layout-aware defects. At the top end, these ATPG (Automatic Test Pattern Generation) vectors are generated based on real physical failures of the process variation and design marginality in similar hardware. In HPC or GPU chips based on 7nm, 5nm, or even 3nm nodes, as the pitch shrinks, so does the area between structures, increasing the probability of random manufacturing defects. Without sufficient DFT, the defects cannot be detected and can sneak out into the field, causing silent data corruption, intermittent failures, or catastrophic system errors.

DFT enables in-field screening through structural testing with high coverage, built-in self-test (BIST), and logic monitoring. These mechanisms ensure that hard (permanent) and soft (temperature-dependent, transient) faults fail and are caught before a device goes to its production environment. Furthermore, DFT logic can also support diagnostic test modes, allowing the yield engineers to perform failure analysis and identify localized root causes. This capability is critical to yielding ramp efforts, including those in early silicon samples. It is relied upon for silicon debugging, increasing test coverage, and tuning process parameters in real time.

Simulation-to-Silicon Correlation and Test Validation

The simulation-to-silicon correlation is the process by which FTR silicon must match simulation and emulation results with real-world chip behavior. DFT allows test validation paths to be measured directly, accelerating this by an order of two. State capture can also be accomplished via scan chains, enabling real-time observation and capture of the internal state, which can be compared with the expected RTL output. Pattern replay is a common test validation strategy that employs the usage of ATPG or functional tests in simulation and subsequently in post-silicon environments via JTAG or IJTAG interfaces. If the observed behavior is inconsistent with simulated behavior, logic bugs could be one source, toolchain problems are a second possibility, and the third reason could be silicon-level variation, such as clock skew, voltage instability, or incompatibilities between the currently synthesized and routed logic.

DFT schemes such as trace buffers, on-chip logic analyzers, and trigger logic assist in isolating root causes by providing functional history up to and including a detected failure event. During the bring-up phase, this observability is crucial to reduce debug time from weeks to days. If failure extraction can be automated via tools integrated with IJTAG, IEEE 1500, and scan test infrastructure, the extracted falling waveforms can be deeply correlated with simulation models. In HPC environments, where execution units are replicated and run under fine-grained power and clock management, such correlation is necessary to validate proper operation under dynamic conditions. Only through such test-enabled infrastructure is real silicon behavior traceable, and it is thereby possible to ensure accurate cross-domain synchronization, correct voltage scaling behavior, and stable timing margins.

Real-World Case Examples of Avoiding Costly Respins

DFT has value and several real-world examples of achieving the first-time-right silicon in the semiconductor industry. One GPU vendor, whose family of products includes many designs, found a race condition within its shader pipelines that would have otherwise led to pixel corruption under high thermal loads. The vendor used modular LBIST with hierarchical test access to pinpoint the issue. The defect was brought up early and isolated with scan capture replay and on-chip debug instrumentation. In a second instance, an HPC server-grade processor developer combined MBIST and redundancy logic with cell-aware ATPG to recover yields in silicon ramp time. Detailed DFT blocks, some of which were advanced, were used to localize the fault to a specific memory compiler variant before the necessary retape, thus allowing quick corrective action in physical design.

The third example demonstrated the importance of testing for chip-to-chip interconnects with a chipset-based AI accelerator, using inter-die scan chaining and test compression to validate the inter-die interface. Without DFT visibility into die interfaces, subtle timing mismatches would have gone undetected and required a retape that cost upwards of a million dollars. These cases stress that DFT is not the posterior function of robust and forward-thinking DFT design; it is a critical part of the silicon success strategy (Chen et al., 2024).

Future Trends in DFT for HPC and AI Accelerators

Changes in high-performance computing (HPC) and artificial intelligence (AI) hardware evolution continue to drive traditional Design for Test (DFT) paradigms out of resistance. A new DFT methodology exists to cope with these chips' increasing heterogeneity, data-centricity, and power awareness. Three forces increasingly shaping future trends in DFT are artificial intelligence and machine learning (AI/ML) enabled test automation, in-field telemetry and analytics, and advanced packaging technologies (chiplets and 3D-ICs). These innovations will be how testability, reliability, and validation are engineered into next-generation compute platforms.

Table 6: ***Future Trends in DFT for HPC and AI Accelerators***

Trend	**Description**
AI/ML in Test Automation	Use of machine learning to generate test patterns, improve compaction, and reduce test time.
In-Field Telemetry	Continuous monitoring of chip health and predictive maintenance to prevent failures.
Advanced Packaging (3D-ICs)	DFT for heterogeneous chip stacking and inter-die connections in advanced packaging.

AI/ML in Automated Test Generation and Fault Analysis

Integrating AI and machine learning into DFT is one of the most promising trends that aim to extend the capability of traditional test development. AI/ML algorithms that analyze RTL structures, power grids, and fault injection simulations may create intelligently generated high-coverage test patterns. These systems input historical test data, yield logs, and silicon debug traces, learning to predict likely defect sites to generate optimal pattern sets (Sardana, 2022). As a result, adaptive ATPG becomes possible, which utilizes learning models to adapt the test strategies to a given node or node topology. AI can help in test compaction, pattern selection, and scan chain balancing to shorten test time at the same test coverage. Additionally, ML models are extensively used to relate failure signatures to defect localization in different silicon samples. In high-volume HPC and AI accelerators required to perform rapid iteration, this accelerates silicon bring-up and yield learning. AI-enhanced DFT has also been integrated into EDA toolchains for use in the early stages of synthesis and place and route. The tools provide DFT insertion points that minimize congestion, improve observability, and minimize scan path length while considering future debug and power profiling.

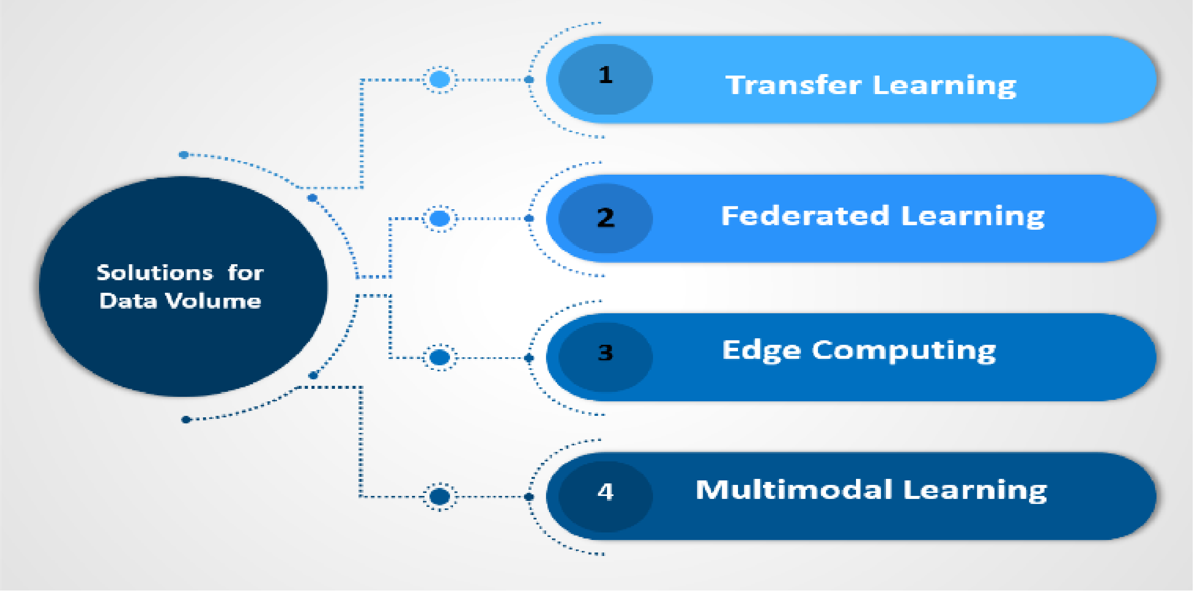

Figure 9: Proposed solutions for challenges of data volume

In-Field Telemetry and Post-Silicon Debug Analytics

HPC and AI chips work in mission-critical environments (such as cloud data centers, autonomous systems, and edge inference engines) and cannot be limited to factory testing in terms of reliability. This has also spawned interest in field telemetry, where ever-present monitors continuously track the chip's aging, health, thermal behavior, and fault events (Dhanagari, 2024). In the field, DFT helps in predictive maintenance by understanding degradation patterns and usage-induced faults before they manifest as system-level faults (Nair et al., 2017). Now, RAS (Reliability, Availability, and Serviceability) architectures are starting to integrate logging systems that are DFT (Design for Test) aware, triggering proactive recovery mechanisms or workload migration.

Post-silicon debugs analytics builds these models of the latent defect behavior in tandem with aggregated logs of thousands of deployed chips. These analytics give the design and manufacturing team valuable feedback to use in improving the continuous testing process and defect mitigation strategy. Most modern chips now include BIST or LBIST engines that periodically run tests during idle cycles. Reference signatures are compared against the obtained results, and anomalies are logged for later analysis. Such runtime validation is necessary for autonomous or aerospace applications to fulfill regulatory and safety certification requirements.

DFT Innovations for Chiplets, 3D-ICs, and Advanced Packaging

The most revolutionary trend in silicon architecture history is shifting from chipset-based designs to 3D integrated circuits (3D ICs). Advanced packaging methodologies, which differ from the conventional practice of building a monolithic die, assemble various smaller chips called chipsets to form a single system. Such units enable higher-speed die-to-die interconnection with high scalability, design flexibility, and heterogeneous integration between the processing, memory, and I/O domains. However, this type of evolution poses new challenges for Design for Test (DFT). Strategies for the testability of chipset-based systems must consider each chip's testability separately in chip fabrication, stacked form, and then post-assembled as part of the complete package. Then, if interconnect integrity (i.e., die-to-die links) and thermal behavior need to be verified, vertical stacking can cause hot spots, compromise heat dissipation, and affect test coverage and fault manifestation.

Careful DFT is also needed for power delivery and clock synchronization across dies. The timing and voltage of stacked dies vary for unpredictable failure modes, and DFT solutions need incapable signal routing and system stability across dynamic operating conditions. To overcome these challenges, new industry standards (IEEE 1838) for DFT address this need for 3D-ICs (Kumar, 2019). The protocols of IEEE 1838 are defined for Silicon Via (TSV) testing, test escape routing, and die identification, which allows standardized access and scheduling among multiple dies in a stack. To validate System in-package (SiP) configurations with digital, analog, RF, and memory components, complementary enhancements, such as boundary scan extensions, programmable test routers, and interposer-level BIST, are becoming essential.

As HPC and AI accelerators become increasingly heterogeneous in the face of test and verification challenges, DFT cannot continue operating at the chip level and should be accompanied by a system test paradigm. Therefore, future DFT strategies must support vertical test planning, thermal-aware scan path design, and partitioned test scheduling compatible with power and thermal constraints in a vertically stacked substrate. Such innovations are needed to ensure that next-generation silicon's quality, reliability, and yield will be maintained for increasingly dense and interlinked compute platforms (Kim, 2015).

CONCLUSION

The growth in the demand for computational performance has proceeded exponentially, driven by workloads in high-performance computing (HPC), artificial intelligence (AI) inference, scientific simulations, and immersive graphics, while the complexity of underpinning silicon architectures has also grown exponentially. With modern HPC and GPU chips comprising billions of transistors, dozens of power domains and clock domains, and subsystems toward latency, throughput, and energy efficiency, developing and testing powerful new algorithms isn't easy. In this complexity, silicon reliability, testability, and manufacturability have become more important than ever, but ensuring these properties has become increasingly difficult. This mission rests on the core of this: Design for Test (DFT), which has evolved from a collection of 'add-on' features into a deeply integrated discipline enabling the design and success of advanced silicon.

But DFT now provides a defense against premature failure at the manufacturing step, supports triggering silicon debug in the field, and offers post-silicon debug assurance into the product life cycle. It is the technical basis for the first-time-right silicon, test escape reduction, and provision of yield learning. Even though communication, memory, and signal integrity continue to challenge electronic design, fundamental testing methodologies such as scan-based testing, built-in self-test, logic BIST, and memory, DFT is still very important; in contrast, the new methods –fabless DFT, embedded instrumentation, and AI-aided pattern generation –are exploring the edges of what may be viable in present and future large-scale integration.

The key to fully using the advantages of DFT is to incorporate best practice tools into the chip development flow at the inception. DFT planning should start at the architectural level instead of being postponed to back-end implementation to guarantee that test requirements naturally align with the design hierarchy and performance goals. Hierarchical and modular DFT simplifies integration, and such an approach also allows the reuse of verified intellectual property (IP) blocks, which speeds up time-to-market. Due to operating stress scenarios and the need to capture subtle single timing-dependent defects, power, and timing-aware ATPG must be used. As evident in both the DFT and physical implementation perspectives, their DFT-aware adaptation is equally necessary to avoid scan congestion and routing conflicts that result in the degradation of critical path performance.

Post-silicon infrastructure is equally vital. Debug hooks in compute cores, memory subsystems, and interconnect fabrics should allow visibility of runtime behavior and root cause analysis through IEEE 1687 (IJTAG) access networks, trace buffers, and trigger logic. In chipset and 3D die integration, DFT should validate the individual dies and ensure device and inter-die link integrity and performance, as well as die-to-die scan chains and shared power and clock domains. The strategies make system-level testability scalable over vertically integrated package technologies.

The future of DFT is to evolve along with the state of the art in packaging, AI-based design automation, and field analytics. Machine learning-driven intelligent test systems will generate the test content on the fly, adapt to the in-field conditions, and learn from silicon performance trends. The system will have embedded telemetry with predictive maintenance tools, ensuring real-time insights into the system's health status and making it more reliable. Cross-die DFT frameworks will facilitate horizontally seamless validation of complex chipsets and 3D-IC architectures to bridge the gap between the assurance of individual components and the system. DFT will ultimately never be seen as a post-design necessity; it must be integrated into the silicon fabric from the get-go. On the integrated level of HPC and GPU, DFT is more than a test strategy: It is the stage upon which resilient, high-performance, scalable computing at Petascale will be achieved in the era of data-intensive computing.

REFERENCES

1. Abotbol, Y., Dror, S., Tshagharyan, G., Harutyunyan, G., & Zorian, Y. (2022). In-field test solution for enhancing safety in automotive applications. *Microelectronics Reliability*, *137*, 114774.
2. Bhatelia, S. H. (2017). *Scan Analysis & Coverage Improvement forLeading ProcessTechnology* (Doctoral dissertation, Institute of Technology).
3. Chavan, A. (2021). Eventual consistency vs. strong consistency: Making the right choice in microservices. International Journal of Software and Applications, 14(3), 45-56. https://ijsra.net/content/eventual-consistency-vs-strong-consistency-making-right-choice-microservices
4. Chavan, A. (2024). Fault-tolerant event-driven systems: Techniques and best practices. Journal of Engineering and Applied Sciences Technology, 6, E167. http://doi.org/10.47363/JEAST/2024(6)E167
5. Chen, S., Tong, X., Huo, Y., Liu, S., Yin, Y., Tan, M. L., ... & Ji, W. (2024). Piezoelectric biomaterials inspired by nature for applications in biomedicine and nanotechnology. *Advanced Materials*, *36*(35), 2406192.
6. Cheng, X., & DeGiorgio, M. (2020). Flexible mixture model approaches that accommodate footprint size variability for robust detection of balancing selection. *Molecular biology and evolution*, *37*(11), 3267-3291.
7. Cini, N., & Yalcin, G. (2020). A methodology for comparing the reliability of GPU-based and CPU-based HPCs. *ACM Computing Surveys (CSUR)*, *53*(1), 1-33.
8. Dhanagari, M. R. (2024). MongoDB and data consistency: Bridging the gap between performance and reliability. *Journal of Computer Science and Technology Studies, 6*(2), 183-198. https://doi.org/10.32996/jcsts.2024.6.2.21
9. Dhanagari, M. R. (2024). Scaling with MongoDB: Solutions for handling big data in real-time. *Journal of Computer Science and Technology Studies, 6*(5), 246-264. https://doi.org/10.32996/jcsts.2024.6.5.20
10. Goel, G., & Bhramhabhatt, R. (2024). Dual sourcing strategies. *International Journal of Science and Research Archive*, 13(2), 2155. https://doi.org/10.30574/ijsra.2024.13.2.2155
11. Gulve, R., Bade, D. P., Kulkarni, S., Ricchetti, M., & Cron, A. (2022, July). Test methodology automation for multi-die package realization. In *2022 IEEE International Test Conference India (ITC India)* (pp. 1-5). IEEE.
12. Janicki, J., Mrugalski, G., Stelmach, A., & Urban, S. (2020, November). Scan Chain Diagnosis-Driven Test Response Compactor. In *2020 IEEE 29th Asian Test Symposium (ATS)* (pp. 1-6). IEEE.

13. Jiang, D., Lin, W., & Raghavan, N. (2021). Semiconductor manufacturing final test yield optimization and wafer acceptance test parameter inverse design using multi-objective optimization algorithms. *Ieee Access*, *9*, 137655-137666.

14. Karwa, K. (2023). AI-powered career coaching: Evaluating feedback tools for design students. Indian Journal of Economics & Business. https://www.ashwinanokha.com/ijeb-v22-4-2023.php

15. Kim, K. (2015, February). 1.1 silicon technologies and solutions for the data-driven world. In *2015 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers* (pp. 1-7). IEEE.

16. Kim, W., & Katipamula, S. (2018). A review of fault detection and diagnostics methods for building systems. *Science and Technology for the Built Environment*, *24*(1), 3-21.

17. Koenemann, B. (2018). Design-for-test. In *EDA for IC System Design, Verification, and Testing* (pp. 21-1). CRC Press.

18. Kong, T. N., Alias, N. E., Hamzah, A., Kamisian, I., Tan, M. P., Sheikh, U. U., & Wahab, Y. A. (2021, August). An efficient march (5n) FSM-based memory built-in self test (MBIST) architecture. In *2021 IEEE Regional Symposium on Micro and Nanoelectronics (RSM)* (pp. 76-79). IEEE.

19. Konneru, N. M. K. (2021). Integrating security into CI/CD pipelines: A DevSecOps approach with SAST, DAST, and SCA tools. *International Journal of Science and Research Archive*. Retrieved from https://ijsra.net/content/role-notification-scheduling-improving-patient

20. Kovács, J., Ligetfalvi, B., & Lovas, R. (2024). Automated debugging mechanisms for orchestrated cloud infrastructures with active control and global evaluation. *IEEE Access*.

21. Kumar, A. (2019). The convergence of predictive analytics in driving business intelligence and enhancing DevOps efficiency. International Journal of Computational Engineering and Management, 6(6), 118-142. Retrieved from https://ijcem.in/wp-content/uploads/THE-CONVERGENCE-OF-PREDICTIVE-ANALYTICS-IN-DRIVING-BUSINESS-INTELLIGENCE-AND-ENHANCING-DEVOPS-EFFICIENCY.pdf

22. Laisne, M., Crouch, A., Portolan, M., Keim, M., von Staudt, H. M., Abdalwahab, M., ... & Rearick, J. (2020, November). Modeling novel non-JTAG IEEE 1687-like architectures. In *2020 IEEE International Test Conference (ITC)* (pp. 1-10). IEEE.

23. Marwala, T. (2024). CPUs Versus GPUs. In *The Balancing Problem in the Governance of Artificial Intelligence* (pp. 137-152). Singapore: Springer Nature Singapore.

24. Mazumdar, S. (2017). An Efficient NoC-based Framework To Improve Dataflow Thread Management At Runtime.

25. Nair, R., Nayak, C., Watkins, L., Fairbanks, K. D., Memon, K., Wang, P., & Robinson, W. H. (2017). The resource usage viewpoint of industrial control system security: an inference-based intrusion detection system. In *Cybersecurity for Industry 4.0: Analysis for Design and Manufacturing* (pp. 195-223). Cham: Springer International Publishing.

26. Nyati, S. (2018). Revolutionizing LTL carrier operations: A comprehensive analysis of an algorithm-driven pickup and delivery dispatching solution. International Journal of Science and Research (IJSR), 7(2), 1659-1666. Retrieved from https://www.ijsr.net/getabstract.php?paperid=SR24203183637

27. Oba, F., & Kumagai, Y. (2018). Design and exploration of semiconductors from first principles: A review of recent advances. *Applied Physics Express*, *11*(6), 060101.

28. Okasaka, S., Weiler, R. J., Keusgen, W., Pudeyev, A., Maltsev, A., Karls, I., & Sakaguchi, K. (2016). Proof-of-concept of a millimeter-wave integrated heterogeneous network for 5G cellular. *Sensors*, *16*(9), 1362.

29. Raju, R. K. (2017). Dynamic memory inference network for natural language inference. International Journal of Science and Research (IJSR), 6(2). https://www.ijsr.net/archive/v6i2/SR24926091431.pdf

30. Sardana, J. (2022). Scalable systems for healthcare communication: A design perspective. *International Journal of Science and Research Archive*. https://doi.org/10.30574/ijsra.2022.7.2.0253

31. Sardana, J. (2022). The role of notification scheduling in improving patient outcomes. *International Journal of Science and Research Archive*. Retrieved from https://ijsra.net/content/role-notification-scheduling-improving-patient

32. Singh, V. (2022). Visual question answering using transformer architectures: Applying transformer models to improve performance in VQA tasks. Journal of Artificial Intelligence and Cognitive Computing, 1(E228). https://doi.org/10.47363/JAICC/2022(1)E228

33. Singh, V. (2024). Ethical considerations in deploying AI systems in public domains: Addressing the ethical challenges of using AI in areas like surveillance and healthcare. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*. https://turcomat.org/index.php/turkbilmat/article/view/14959

34. Sontakke, V., & Dickhoff, J. (2023). Developments in scan shift power reduction: a survey. *Bulletin of Electrical Engineering and Informatics*, *12*(6), 3402-3415.

35. Sur, S., Zhang, X., Ramanathan, P., & Chandra, R. (2016). {BeamSpy}: Enabling robust 60 {GHz} links under blockage. In *13th USENIX symposium on networked systems design and implementation (NSDI 16)* (pp. 193-206).

36. Tiwari, D., Gupta, S., Rogers, J., Maxwell, D., Rech, P., Vazhkudai, S., ... & Bland, A. (2015, February). Understanding GPU errors on large-scale HPC systems and the implications for system design and operation. In *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)* (pp. 331-342). IEEE.

37. Vitucci, C., Sundmark, D., Danielsson, J., Jägemar, M., Larsson, A., & Nolte, T. (2023, November). Run time memory error recovery process in networking system. In *2023 7th International Conference on System Reliability and Safety (ICSRS)* (pp. 590-597). IEEE.

38. Wang, R., Chakrabarty, K., & Bhawmik, S. (2015). Built-in self-test and test scheduling for interposer-based 2.5 D IC. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, *20*(4), 1-24.