Transport and Communications Science Journal, Vol. 77, Issue 09 (11/2026), 1138-1154

Transport and Communications Science Journal

TIMING BOTTLENECK CHARACTERIZATION IN AN ITERATIVE KECCAK-F[1600]/SHA3-256 CORE: AN FPGA/ASIC STUDY

Nguyen Thanh Ngoc*, Nguyen Dac Khiem, Doan Duc Duy, Tran Trong Hieu, Nguyen Anh Tai, Chu Thi Ngoc Quynh

Academy of Cryptography Techniques, 141 Chien Thang Street, Thanh Liet, Hanoi, Vietnam

ARTICLE INFO

TYPE: Research Article
Received: 20/04/2026
Revised: 02/11/2026
Accepted: 09/11/2026
Published online: 15/11/2026
https://doi.org/10.47869/tcsj.77.9.2
* *Corresponding author*
Email: thanhngoc@actvn.edu.vn

Abstract. The Keccak-f[1600] permutation underlies the Secure Hash Algorithm 3 (SHA-3) family of cryptographic hash functions, widely used for data integrity and, increasingly, in post-quantum cryptographic hardware. Iterative implementations reuse one round datapath for all 24 rounds through a wide feedback multiplexer, a resource-efficient choice whose timing has not previously been measured with post-route accuracy. This study analyzes an iterative SHA3-256 core implemented on a Xilinx Artix-7 field-programmable gate array (FPGA) and cross-checked on a Virtex-7 device, comparing post-synthesis and post-route timing and evaluating two fan-out-reduction techniques and a multicycle-path exception. The same design was also taken through a complete register-transfer level (RTL) to GDSII flow on a 45 nm process as an application-specific integrated circuit (ASIC), using an in-house-characterized standard-cell library and a widely used public library, with functional correctness verified against reference test vectors. Post-synthesis estimates underestimate the critical-path delay of the feedback-select path by up to 2.8×, none of the three mitigation techniques improves the operating frequency, and the bottleneck is architecture-induced rather than technology-dependent. The in-house library achieved a signed-off implementation with zero design-rule check (DRC) and layout-versus-schematic (LVS) violations, whereas cell-level verification found manufacturing-grid violations in the public library that prevented a comparable signoff.

Keywords: Keccak, SHA-3, FPGA, ASIC, timing closure, standard-cell library.

@ 2026 University of Transport and Communications

1. INTRODUCTION

The Keccak-f[1600] permutation forms the core of the Secure Hash Algorithm 3 (SHA-3) family of cryptographic hash functions, standardized by the National Institute of Standards and Technology (NIST) in Federal Information Processing Standards (FIPS) Publication 202 [1], following the original Keccak specification [2]. The permutation applies 24 rounds, each composed of five sub-transformations (θ, ρ, π, χ, ι), to a 1600-bit internal state. Hardware acceleration of this permutation remains an active research area for both conventional embedded security and post-quantum cryptographic systems: the NIST-standardized module-lattice signature and key-encapsulation schemes (ML-DSA, the Module-Lattice-Based Digital Signature Algorithm derived from Dilithium, and ML-KEM, the Module-Lattice-Based Key-Encapsulation Mechanism derived from Kyber) internally invoke SHA-3 and its extendable-output functions (SHAKE) for expansion, sampling, and transcript hashing, so that the Keccak permutation is frequently the dominant hardware cost in a post-quantum accelerator [3-9].

A widely used class of hardware architecture for Keccak-f[1600] is the iterative, feedback-based design [2, 10]: a single round datapath is instantiated once and reused for all 24 rounds, with a multiplexer selecting between newly padded input data and the feedback state from the previous round. This architecture minimizes resource usage relative to unrolled or fully pipelined alternatives [10, 11], making it attractive for resource-constrained field-programmable gate array (FPGA) platforms [12] and area-constrained application-specific integrated circuit (ASIC) blocks. However, the feedback-select signal must fan out to the full 1600-bit state register, and the resulting routing structure can dominate the critical timing path in ways not always visible at early design stages - a possibility that, as Section 2 shows, has not been directly measured in the reported iterative-Keccak literature.

This paper investigates that timing bottleneck directly, using post-route (not post-synthesis) measurement as the primary source of evidence throughout, and additionally evaluates the same design through a complete ASIC flow to characterize the practical consequences of standard-cell library choice for academic physical-design studies. Section 2 reviews related literature and distills the specific gaps this work addresses; Section 3 describes the proposed architecture, verification methodology, and the interventions evaluated on the feedback-select path; Section 4 reports FPGA and ASIC results; Section 5 concludes.

2. RELATED WORK, RESEARCH GAP, AND CONTRIBUTIONS

2.1. Iterative and Pipelined Keccak/SHA-3 Architectures

A substantial body of work targets FPGA throughput improvement for Keccak/SHA-3 through pipelining and architectural restructuring. Sideris, Sanida, and Dasygenis reported a NIOS-II-coupled implementation [13], a two-stage pipelined cryptoprocessor [14], and an enhanced high-throughput/low-area architecture exploiting round-constant (RC) sparsity - only 7 of 64 bit positions are ever non-zero across all 24 rounds, independently confirmed here - to simplify the RC generator [15, 16], and an FPGA SHA-3 accelerator for image-processing workloads [17]. Putra, Natan, and Istiyanto mapped Keccak's bitwise operations onto Xilinx digital signal processing (DSP48) slices with an eight-stage pipeline, trading a lower 107.979

MHz clock for reduced look-up table (LUT) and flip-flop (FF) pressure [18]. Abdulsamad and Répáš targeted the same Artix-7 family addressed here, reporting 210 MHz and 1.35 Gbps with loop unrolling, clock gating, and pipeline balancing, at 0.94 W [12]. Kermiche, Saoudi, and Drouiche reported a five-stage pipelined SHA-3 design that processes five messages in succession, with a maximum frequency of 676.18 MHz and a throughput of 27.82 Gbps [19]. SHA-3 deployment on the RISC-V (reduced instruction set computer, fifth generation) architecture using the Arty A7-35T board has also been reported in a Vietnamese conference study [20]; that work concerns RISC-V-based deployment, whereas the present study concerns the timing behavior of the Keccak datapath itself. Earlier throughput-oriented implementations [10, 21, 22] and a broader comparative survey of Keccak SHA-3 implementations [11] are not restated here, as their results were not independently re-verified in this work.

These studies share a common evaluation practice: reported maximum operating frequency (Fmax) figures are taken from the vendor timing report at whatever stage the authors chose to characterize, and the pipelined or DSP-mapped restructuring is itself the proposed fix for low throughput. None of [10, 12-16, 18, 19, 21, 22] reports, for an iterative single-round-per-cycle architecture, a direct measured comparison between the post-synthesis and post-route delay of the feedback-multiplexer structure - the feature pipelining and unrolling implicitly work around. This comparison is provided in Section 4.1.

2.2. SHA-3/Keccak in Post-Quantum Cryptographic Hardware

SHA-3's role has expanded beyond its original mandate as several NIST-standardized post-quantum schemes depend on it internally. Aikata et al. present Kali, a unified hardware core sharing resources between Kyber and Dilithium in which Keccak is a shared, heavily reused block [6]. Dedicated Keccak-for-Kyber wrappers [7] and RISC-V-coupled accelerator studies integrating Keccak/SHA-3 as a fixed-function unit [8] follow the same pattern, as does post-quantum signature acceleration on RISC-V where SHA-3/SHAKE is again a shared, latency-critical resource [9]. A recent lightweight unified SHA-3/SHAKE architecture with a fault-resilient state, integrated into a RISC-V processor at under 8% area overhead, follows the same building-block integration pattern [23]. In software, Keccak is reported as one of the dominant cycle costs alongside Kyber and Dilithium on RISC-V vector extensions [24] and on the 16-bit MSP430 [25]. SHA-3 execution time has also been benchmarked against BLAKE3 and SHA-256 on a Raspberry Pi board in a study published in this journal [26]. This body of work motivates why Keccak-f[1600] efficiency remains relevant beyond classical hashing, but treats the permutation as a reused building block rather than the primary object of timing-closure analysis - the gap addressed here.

2.3. Reliability of Pre-Route Timing Estimates (General Electronic Design Automation Literature)

Outside the Keccak-specific literature, the electronic design automation (EDA) community has separately documented that post-synthesis or post-placement timing estimates diverge from final, routed timing, particularly under routing congestion. Studies bridging global-route and detailed-route stages report measurable wire-delay divergence from pin-access constraints and design rules not modeled at earlier stages, both in the open-source OpenROAD flow and a

commercial 45 nm flow [27, 28]. A recent timing-driven placement study for large heterogeneous FPGAs reports an average post-routing critical-path prediction error of 8.3% for their placer versus 4.4% for commercial Vivado 2024.2 and 11.4% for a prior academic baseline [29]. A broader survey likewise frames post-route optimization as the stage where residual, routing-specific violations first become fully visible [30]. These studies confirm the pre-route/post-route gap as a recognized, general phenomenon, but it has not previously been quantified for the specific case of a wide feedback multiplexer in an iterative cryptographic permutation core, nor tied to a testable remedy. Sections 3.2-3.4 and 4.1 provide that quantification and remedy.

2.4. Standard-Cell Library Selection for Academic ASIC Feasibility Studies

Academic ASIC feasibility studies commonly rely on either a widely distributed open-source standard-cell library or an in-house-characterized one. The FreePDK45 process design kit (PDK), developed jointly by North Carolina State University (NCSU) and Oklahoma State University (OSU) as a predictive, non-fabricable 45 nm node for research and education, is the most widely cited resource of this kind [31, 32], and the accompanying OSU Standard Cell Library (gscl45nm) is correspondingly one of the most widely reused open digital cell sets at this node. Independent extensions of the same FreePDK45 base (ASCEnD-FreePDK45 [33, 34]; the mflowgen FreePDK45/Nangate design-kit assembly [35]) corroborate that assembling a usable, design-rule-check (DRC) and layout-versus-schematic (LVS) clean open-source cell set at this node has historically required manual correction: the mflowgen project's public change log documents net-naming conflicts and a layer-map correction needed to “match the Calibre DRC/LVS rule decks and stdcells.gds layer numbers” [35] - friction independent of the present authors' own findings. A separate characterization effort further corroborates that a verification-clean FreePDK45-class flow has repeatedly required dedicated engineering effort rather than being available out-of-the-box [36].

Despite this history, DRC/LVS status for the distributed standard-cell layouts themselves, as opposed to a full routed chip, is rarely reported in the academic ASIC-flow literature that uses these libraries for a specific application (e.g. a cryptographic core) - DRC is typically reported only for the final routed design if at all, and a defect intrinsic to the cell layouts is easily conflated with a routing-stage violation. Section 4.4 reports a direct, cell-level DRC check performed in isolation from the full-chip place-and-route (P&R) flow, with methodological confounds (database units, layer mapping) explicitly identified and eliminated before drawing a conclusion - a level of verification not reported in the reviewed application-level literature for this library.

2.5. Summary of Identified Gaps

- Gap 1 (architecture): No reviewed iterative-Keccak study reports a direct, measured post-synthesis-vs-post-route comparison for the feedback-select fan-out structure, nor tests fan-out mitigation against post-route measurement for this structure (§2.1).
- Gap 2 (generality of the timing gap): The pre-route/post-route timing-estimate gap is documented generally in the EDA literature (§2.3) but not connected quantitatively to this architectural structure or a corresponding timing-exception remedy.

- Gap 3 (verification thoroughness): Reported iterative-Keccak testbenches are not, in general, documented as exercising back-to-back (no-reset) operation - a condition that stresses controller re-arming logic and complements a single reset-then-hash test.
- Gap 4 (library verification): Open-source standard-cell libraries at the FreePDK45/45 nm node have documented layer-convention and verification friction (§2.4), yet cell-level DRC signoff of the distributed layouts, isolated from the P&R flow with confounds controlled for, is not reported in the literature that consumes these libraries.

2.6. Contributions of This Work

This work does not propose a new Keccak permutation architecture, and none of the text, figures, or numerical results of the cited studies are reproduced. The contributions below are original measurements and methodological artifacts produced for this study. Specifically, this paper contributes:

- A direct, measured quantification (Section 4.1) of the post-synthesis-to-post-route timing gap for the feedback-select fan-out network of a baseline iterative Keccak-f[1600] core.
- A characterization, by post-route measurement, of three targeted interventions on the feedback-select path (Sections 3.3-3.4): two fan-out-reduction techniques and a multicycle-path exception valid for the signal's once-per-operation transition. None improves operating frequency; the multicycle exception relaxes the targeted path but the binding constraint migrates elsewhere, evidence the limitation is distributed and wire-length-dominated rather than a localized fan-out effect.
- A cross-device (Artix-7 vs. Virtex-7) validation (Section 4.2) isolating architecture-induced from technology-induced timing limitation.
- A functional-verification methodology (Section 3.5) that re-applies the same NIST test-vector set at the register-transfer level (RTL), post-synthesis, and post-route stages, and additionally exercises back-to-back (no-reset) operation.
- A complete, signed-off RTL-to-GDSII ASIC implementation (Section 4.3) on an in-house-characterized 45 nm standard-cell library, reported with zero DRC/LVS violations and staged logical-equivalence checking.
- An isolated, confound-controlled cell-level DRC verification of a widely used public reference library (Section 4.4), reported together with an explicit statement of what remains open (Section 4.6).

3. PROPOSED METHOD

3.1. Baseline Iterative Architecture

The baseline architecture follows the conventional iterative, feedback-based organization for Keccak-f[1600]: a single instantiation of the round function (θ, ρ, π, χ, ι) is applied once per clock cycle, and its output is fed back through a 1600-bit-wide multiplexer that selects, at the start of each hash operation, between newly padded input data and the recirculating state from

the previous round. A dedicated controller tracks the round index (0 to 23) and asserts a truncation/output stage once the final round has completed. This architecture was selected, in preference to unrolled or multi-context interleaved alternatives such as [18], specifically for its low resource footprint on cost-constrained FPGA targets; Section 4 reports the resource cost actually measured for this choice. The overall datapath organization is shown in Figure 1.

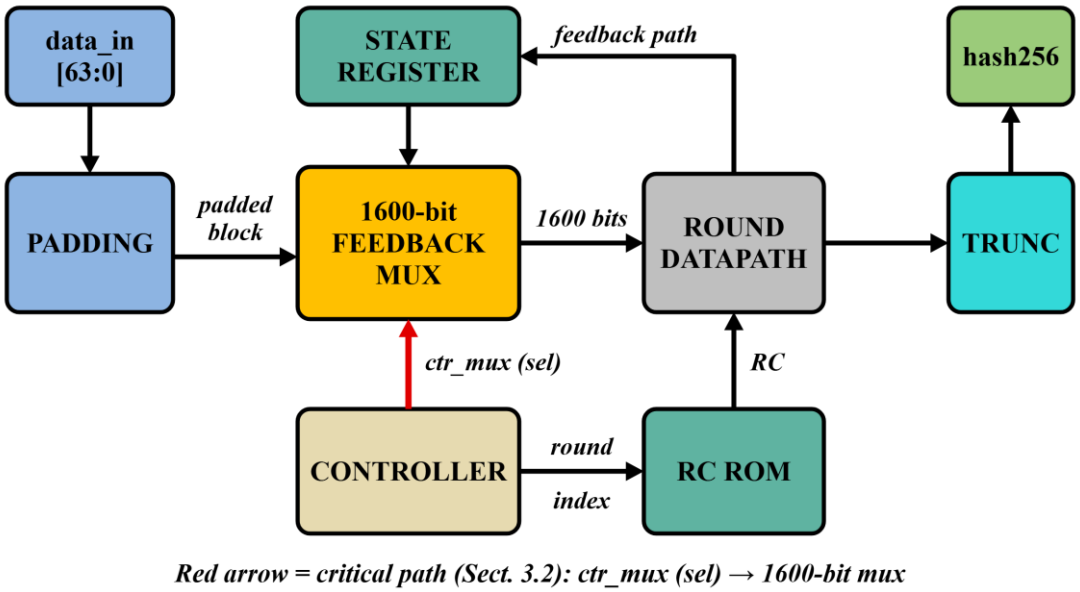


Fig. 1. Block diagram of the iterative Keccak-f[1600] datapath with the critical path annotated (Section 3.2).

3.2. Root-Cause Analysis of the Timing Bottleneck

Post-route timing analysis identifies the critical path as originating at the feedback-select register and terminating at the input-multiplexer output register, spanning the full 1600-bit fan-out network (measured fan-out: 201). Table 1 compares the net-delay component of this path as estimated at the post-synthesis stage versus the value measured after placement and routing.

Table 1. Post-synthesis vs. post-route net delay on the feedback-select-to-register path.

Stage	Net delay (ns)	Ratio vs. post-synthesis
Post-synthesis (pre-placement estimate)	1.92	1.0× (baseline)
Post-route (measured)	5.41	2.8×

This discrepancy is consistent in direction, though larger in magnitude, with the general pre-route/post-route gaps reported in the EDA literature [27-30]. It indicates that, for this architecture class, post-synthesis timing estimates should not be relied upon: the routed net delay of the feedback-select path, not its logic depth, dominates the critical path. As Section 4.2 confirms (net delay accounting for 82.5-89.6% of total path delay) and Section 4.6 further supports (routing congestion near zero on the affected net), this net-delay dominance stems from the physical wire length required to distribute the feedback-select signal across its placed loads, not localized congestion - meaning techniques that reduce fan-out capacitance without shortening the wiring would not be expected to help.

3.3. Fan-out-Reduction Attempts (Negative Results)

Two conventional fan-out-mitigation techniques were evaluated: a manually coded two-level buffer tree (protected from optimization via KEEP/DONT_TOUCH), and an automatic MAX_FANOUT placement constraint (in Xilinx Design Constraints, XDC, format) letting the synthesis tool determine replication and placement. Both were evaluated by post-route measurement on the feedback-select path characterized in Section 3.2; both reduced its post-route setup slack relative to the unmodified baseline (Table 2), confirming that neither improves the targeted structure's timing.

Table 2. Post-route setup slack on the feedback-select path under fan-out-reduction and multicycle-path interventions.

Configuration	Sel_mux path setup slack (post-route)	Outcome vs. baseline
Baseline (no fan-out mitigation)	4.185 ns	-
Manual 2-level buffer tree (KEEP/DONT_TOUCH)	3.855 ns	Worse (-0.330 ns slack, -7.9%)
Automatic MAX_FANOUT XDC constraint	2.607 ns	Worse (-1.578 ns slack, -37.7%)

Note: setup-slack values were measured on the sel_mux_reg-to-MUX_REG path (Section 3.2) under a fixed, controlled RTL/XDC configuration per row, isolating each technique's effect from unrelated placement variation. A higher slack is better; both techniques reduced it. In the manual-buffer-tree case, the design-level worst negative slack migrated to an unrelated high-fan-out control net rather than improving - indicating that local fan-out replication redistributes rather than removes the limitation, consistent with Section 3.2's wire-length interpretation.

3.4. Multicycle-Path Timing Exception

Unlike the continuously toggling pipeline-control signals typically targeted by multicycle-path exceptions in the pipelined designs of Section 2.1 [18], the feedback-select signal here transitions exactly once per hash operation - at the boundary between the input-loading and 24-round permutation phases - and remains stable thereafter. This makes a multicycle-path exception logically valid: the destination register (and the round-index register) is held stable for one additional clock cycle at that transition, using a dedicated hold-enable signal generated by the round-counting controller and distributed to the register-holding logic and round-tracking finite-state machines (FSMs), keeping data and round-count synchronized. The exception was declared as set_multicycle_path -setup 2 / -hold 1 on the corresponding path, with functional correctness under the relaxed timing confirmed against NIST reference vectors (Section 3.5). Applied alone, however, it did not improve the design-level worst negative slack: post-route measurement (Table 2) shows relaxing the feedback-select path removes it from the worst-path set, but the binding constraint migrates to other high-fan-out control nets (notably a hold-enable distribution net with fan-out of order 10^3) that the exception does not cover. Operating frequency is therefore governed not by the single feedback-select path but by a population of wide, physically distributed nets; relaxing or buffering any one shifts the bottleneck rather than removing it - together with Section 3.3's results, the basis for the wire-length-dominated characterization of Section 3.2, revisited in Section 4.

3.5. Functional Verification Methodology

Functional correctness of the RTL, and of every downstream netlist derived from it, was verified against golden SHA3-256 digests generated with Python's hashlib.sha3_256 reference implementation. Six test messages were used: three with an external reset asserted before each hash operation, and three presented consecutively without an intervening reset, to exercise the controller's re-arming behavior between back-to-back operations. This condition, identified in Section 2.5 (Gap 3) as generally under-documented in comparable testbenches, confirms that internal round-counters and multicycle hold logic return to a consistent state without external intervention. These six vectors do not cover the full NIST Known-Answer-Test (KAT) suite; broader coverage (e.g., empty-message, multi-block cases) is left as future verification work.

The same test-vector set and pass/fail comparison were re-applied at each stage of the flow - RTL simulation, post-synthesis gate-level simulation, and post-route gate-level simulation with back-annotated Standard Delay Format (SDF) timing - so that a discrepancy introduced at any single stage would be caught before being carried forward. This staged re-verification is summarized in Figure 2.

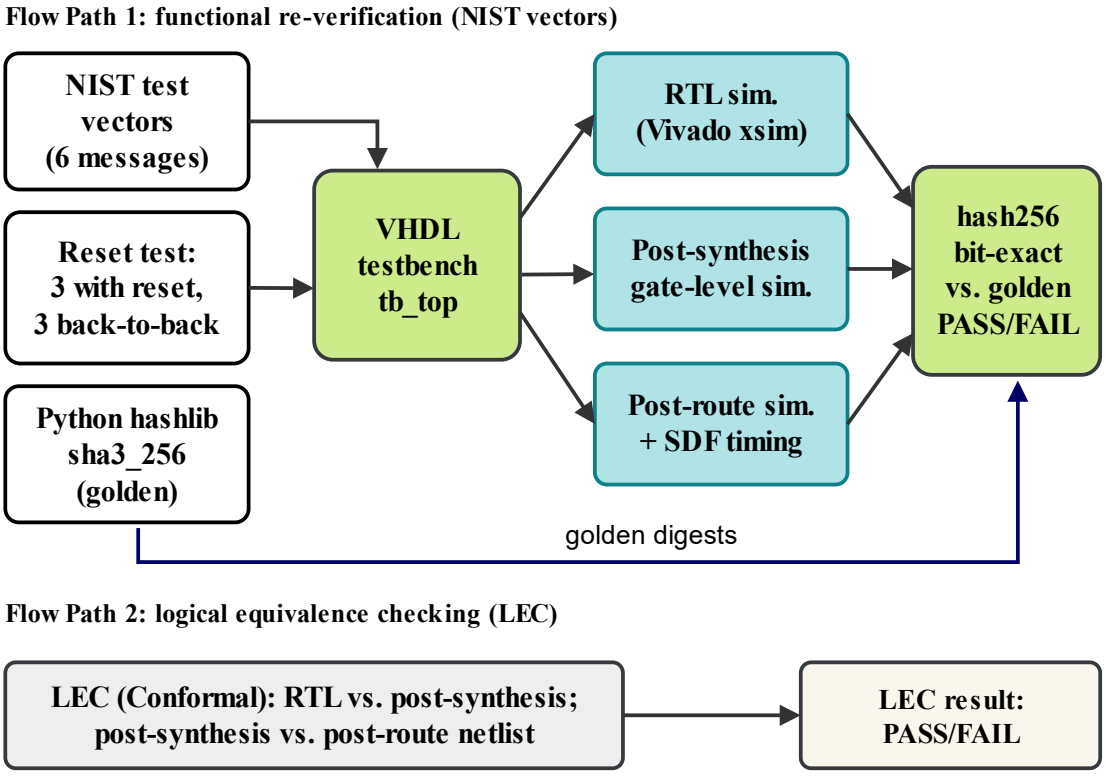


Figure. 2. Functional and logical verification flow: NIST-vector re-testing (Flow Path 1) and logical equivalence checking (Flow Path 2) (Section 3.5).

In addition to functional (value-level) equivalence, logical equivalence checking (LEC) was performed between each pair of adjacent design representations (RTL vs. post-synthesis netlist; post-synthesis vs. post-place-and-route netlist) using Cadence Conformal, confirming that no unintended logic transformation was introduced during synthesis or physical implementation. This is reported together with the ASIC results in Section 4.3.

4. RESULTS

4.1. FPGA Implementation and Functional Verification

The design was implemented on a Digilent Arty A7-100T board (Xilinx XC7A100T-CSG324-1) using Xilinx Vivado. Post-route static timing analysis identifies the feedback-select-to-register path (Section 3.2) as the dominant timing-limiting path; a secondary path introduced solely by an on-chip debug harness (a free-running linear-feedback shift register (LFSR) and XOR-reduction network added to prevent synthesis-time removal of the datapath) is excluded from the reported core timing, as it does not belong to the Keccak datapath. Excluding this path, the measured worst negative slack (WNS) at a 10 ns clock constraint is 4.185 ns, corresponding to an estimated Fmax of approximately 172 MHz - within the range reported for other resource-constrained, non-pipelined Artix-7-class SHA-3 designs [12, 18]. This Fmax is derived as 1/(T_constraint - WNS) from the single 10 ns report, not confirmed by re-implementing near that point; it is an estimate, not a directly measured operating point.

Functional correctness was verified against NIST SHA3-256 reference digests for six test vectors as described in Section 3.5: three with external reset, three consecutive without an intervening reset. All six test vectors produced digests matching the reference values. Figure 3 shows the RTL simulation waveform for this sequence, and Figure 4 the corresponding pass/fail transcript against the golden reference.

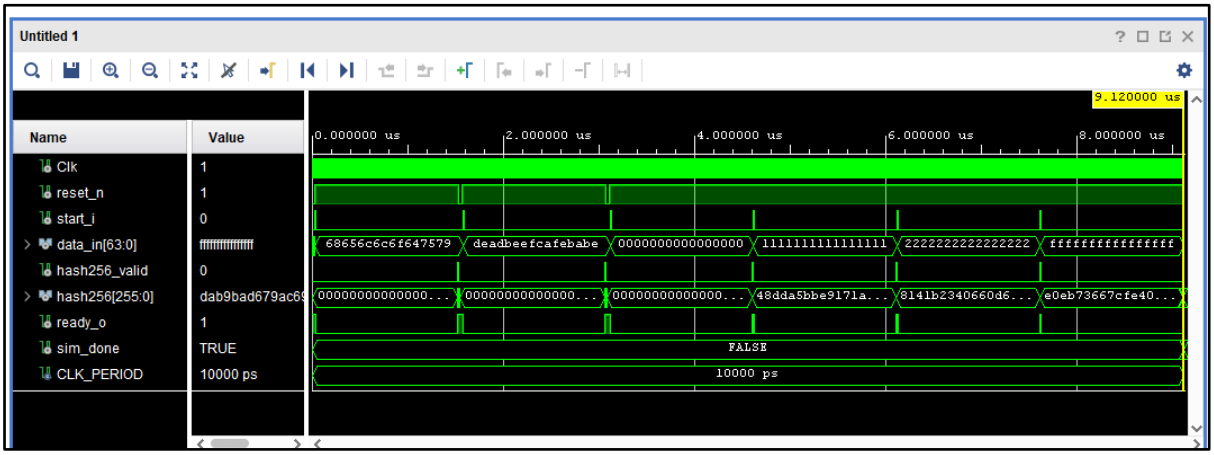


Figure. 3. RTL simulation waveform of the six-vector test sequence, confirming controller re-arming without external reset.

```
RUN 1 @1525 ns in=0x68656C6C6F647579 => 0xC1EC0FA4...153480FB
RUN 2 @3065 ns in=0xDEADBEEFCAFEBABE => 0x947C159A...950B41B7
RUN 3 @4605 ns in=0x0000000000000000 => 0x48DDA5BB...33AEE9DF
RUN 4* @6105 ns in=0x1111111111111111 => 0x8141B234...215E79D0
RUN 5* @7605 ns in=0x2222222222222222 => 0xE0EB7366...CF870A53
RUN 6* @9105 ns in=0xFFFFFFFFFFFFFFFF => 0xDAB9BAD6...B6819EB9
=== DONE: 6 runs, 0 fail(s) ===
* consecutive run, no reset
```

Figure. 4. Simulation transcript confirming bit-exact match against the golden SHA3-256 reference for all six test vectors (Section 3.5).

Table 3. Post-implementation resource utilization report (Artix-7 XC7A100T-CSG324-1; LUTRAM: LUT used as distributed memory, IO: input/output pins, BUFG: global clock buffer).

Resource	Utilization	Available	Utilization %
LUT	3,519	63,400	5.55%
LUTRAM	4	19,000	0.02%
FF	10,773	126,800	8.50%
IO	6	210	2.86%
BUFG	1	32	3.13%
Fmax (estimated)	≈ 172 MHz	-	-
WNS @ 10 ns constraint (core path, debug harness excluded)	4.185 ns	-	-

4.2. Cross-Device Validation

To determine whether the timing bottleneck is an artifact of the target FPGA technology or a structural property of the architecture, the same RTL was synthesized and implemented, without modification, for a Xilinx Virtex-7 XC7VX485T-2FFG1761C device (as on the VC707 evaluation platform). The post-route critical path originates from the same source and destination registers as on Artix-7, with an identical measured fan-out of 201. Logic delay decreased by ≈ 57% relative to Artix-7, consistent with Virtex-7's higher-performance fabric; net (routing) delay decreased by only 21-27% and remained dominant on both devices (82.5% on Artix-7, 89.6% on Virtex-7) - indicating the bottleneck originates predominantly from the physical routing structure and is not substantially alleviated by higher-performance silicon alone. Table 4 summarizes the comparison.

Table 4. Cross-device comparison of fan-out and delay-share on the critical path.

Device	Fan-out	Logic delay share	Net (routing) delay share
Artix-7 XC7A100T	201	17.5%	82.5%
Virtex-7 XC7VX485T	201	10.4%	89.6%

4.3. ASIC Implementation and Standard-Cell Library Verification

All ASIC results here were obtained on FreePDK45, a predictive, non-fabricable academic PDK (Section 2.4); absolute figures are therefore illustrative of relative comparison, not validated silicon. The design was evaluated through a complete RTL-to-GDSII flow on this 45 nm process, using Cadence Genus for logic synthesis and Cadence Innovus for placement and routing. The primary target was an in-house-characterized standard-cell library (kmaLIB, 24 cells characterized using Cadence Liberate 23.1.6 with Spectre across fast (FF), nominal (NOM), and slow (SS) process corners). Table 5 summarizes the post-signoff results: the full flow completed with positive setup slack and zero DRC and LVS violations. Figure 5 shows the post-route layout of the core.

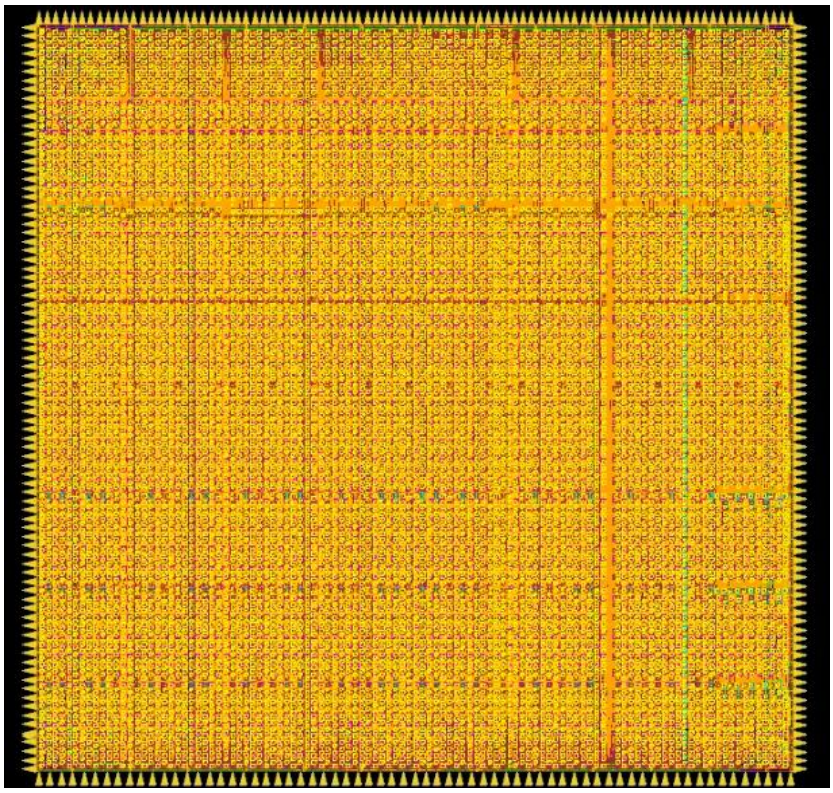


Figure. 5. Post-route layout of the Keccak SHA3-256 core (kmaLIB; die area ≈ 391.6 × 389.8 μm).

Table 5. FPGA/ASIC implementation summary (kmaLIB vs. OSU_SOC).

Metric	**kmaLIB (final, RTL-to-GDSII, signoff)**	**OSU_SOC gscl45nm (synthesis stage; P&R not signed off - see §4.4)**
Stage	Post-P&R (signoff)	Post-synthesis
Cell count	45,389	45,381
Area (μm²)	90,445	172,041 (synthesis-stage; not comparable, see Section 4.4)
Setup WNS (ns)	+1.894	+5.652 (pre-route estimate)
Hold WNS (ns)	0.000	n/a (pre-route)
Power (mW @ 100 MHz)	30.54	22.14 (synthesis-stage; not comparable)
DRC / LVS	0 violations / CORRECT	Not signed off (§4.4)

Functional verification of the ASIC implementation followed the methodology of Section 3.5 at each stage, and was additionally cross-checked using logical equivalence checking (Cadence Conformal) between RTL and post-synthesis, and between post-synthesis and post-place-and-route netlists (evaluated incrementally at five P&R checkpoints: post-placement, post-pre-CTS (CTS: clock tree synthesis), post-CTS, post-route optimization, and post-signoff). All checks passed with zero non-equivalent points at every checkpoint.

4.4. Standard-Cell Library Comparison: Cell-Level DRC Verification

A parallel attempt was made to reproduce the same flow using the publicly available OSU Standard Cell Library for FreePDK45 (gscl45nm) [32], the library whose broader ecosystem history of layer-convention and verification friction is documented in Section 2.4 [31-36]. Before interpreting any P&R-stage result, an independent DRC was performed directly on the standard-cell GDSII layouts supplied with the library, outside the context of the full-chip flow, so as to characterize library-intrinsic layout quality in isolation.

Two potential methodological confounds were identified and eliminated before drawing any conclusion. First, a database-unit mismatch was excluded by verifying that all layout files used a consistent 1 nm database unit. Second, a layer-assignment mismatch between the checking rule deck and the library's original GDSII layer convention was identified and corrected; this correction alone reduced the violation count on the most complex cell examined (DFFPOSX1, a positive-edge D flip-flop) from 1,284 to 898 - confirming layer mapping was a genuine, partially confounding factor, though it did not eliminate the residual violations. With database precision, rule deck, and manufacturing-grid definition (2.5 nm) confirmed identical to the settings under which kmaLIB was independently verified as fully DRC-clean, the remaining violations could no longer be attributed to a units, layer-map, or grid-definition mismatch. Table 6 reports the DRC result for three representative OSU_SOC cells spanning increasing structural complexity: a simple inverter (INVX1), a two-input NAND gate (NAND2X1), and a positive-edge D flip-flop (DFFPOSX1). In all three cells, the dominant violations belong to the manufacturing-grid (off-grid) check group, with the contact off-grid check against the 2.5 nm manufacturing grid being the single largest contributor in every case.

Table 6. Cell-level DRC results for three representative OSU_SOC cells, with matching same-cell-type kmaLIB results.

Cell	Type	Total violations	Off-grid contact (dominant check)	Other notable checks	kmaLIB (same cell type)
INVX1	Inverter	128	68	-	0
NAND2X1	2-input NAND	126	64	-	0
DFFPOSX1	D flip-flop	898	388	Contact.1/2/5, Metal2.3 (≤1 each)	0*

To rule out structural complexity as the explanation, the same three cell types were checked in kmaLIB under the identical deck; all three (INVX1, NAND2X1, D flip-flop) returned zero violations, matching BUFX1. Caveat: kmaLIB's flip-flop is named DFF, not DFFPOSX1 - functionally equivalent, but not yet formally confirmed, hence the asterisk in Table 6.

The off-grid nature of the violations was further confirmed at the coordinate level: contact geometries were displaced from the 2.5 nm manufacturing grid by a fixed, non-zero offset when parsed independently of the DRC tool. Because these violations are intrinsic to the distributed cell layouts - present in the unmodified source GDSII and confirmed to persist after eliminating the units and layer-mapping confounds - they cannot be remedied by P&R adjustments, and any routed result built upon these cells would inherit them. For this reason, post-P&R figures for gscl45nm are not reported alongside the signed-off kmaLIB numbers in Table 5.

This outcome is consistent with the broader, independently documented history of manufacturing-grid and layer-convention friction in the FreePDK45 open-source ecosystem noted in Section 2.4 [35]. Whether the residual violations are best characterized as a genuine layout defect or a systematic grid-convention mismatch with the NCSU FreePDK45 rule deck is beyond what DRC counts alone can settle; this work reports the measured, reproducible result and its methodological controls, leaving the underlying characterization as an open question. Regardless, the practical implication is the same: the public library, as distributed, did not permit a clean physical-design closure without cell-level layout intervention within this study's scope.

4.5. Comparison with Prior Work: Scope and Methodology

The studies reviewed in Section 2.1 [12-16, 18, 19] adopt pipelined or partially unrolled datapaths trading FPGA resource usage for higher throughput, whereas this work adopts a single-round-per-cycle iterative datapath (Section 3.1) - a direct Fmax/throughput comparison would therefore conflate two different architecture classes. The results of Sections 4.1-4.4 instead concern the *reliability of timing analysis* and *implementation completeness* for that architecture class, not a throughput contest. Table 7 therefore compares scope and verification methodology rather than raw performance; where a pipelined study reports higher Fmax or throughput this is stated directly - the studies in [12, 15, 16, 18] are the throughput/frequency leaders among the reviewed FPGA-only Keccak/SHA-3 implementations, consistent with their pipelined/unrolled design intent, and the present work does not contest that ranking.

Table 7. Scope and methodology comparison with prior FPGA/SHA-3 studies (Section 2.1).

Study	Architecture class	Target device(s)	Primary reported performance metric	Post-route vs. post-synthesis gap	Cross-device validation	Complete ASIC (RTL-to-GDSII) flow	Cell-level standard-cell DRC verification
Sideris, Sanida & Dasygenis, 2020-2024 [13-16]	Pipelined, multi-message throughput	Virtex-5/Artix-7/Virtex-6/Virtex-7 (separately)	Throughput up to 19.5-36.4 Gbps depending on device	Not reported	Not reported	Not reported	Not reported
Putra, Natan & Istiyanto, 2025, *IIUM Eng. J.* [18]	8-stage pipelined, DSP48-mapped	Artix-7-class	Fmax ≈ 107.979 MHz	Not reported	Not reported	Not reported	Not reported
Abdulsamad & Répáš, 2026, *Computers* [12]	Fully pipelined + partial loop unrolling + clock gating	Artix-7	210 MHz, 1.35 Gbps, 0.94 W	Not reported	Not reported	Not reported	Not reported
This work	**Iterative, single-round-per-cycle**	**Artix-7; Virtex-7 for cross-check**	**Fmax ≈ 172 MHz; LUT 3,519 (5.55%)**	**Yes - 2.8× (1.92→5.41 ns)**	**Yes (Artix-7 vs. Virtex-7)**	**Yes - signed off, kmaLIB, 0 DRC/LVS**	**Yes - confound-controlled DRC (OSU_SOC & kmaLIB)**

Note: figures for "This work" are drawn from Sections 4.1-4.4. "Not reported" means the metric falls outside that study's stated goal (typically pipelined/DSP throughput or Fmax), not a claim of infeasibility or lower quality - a scope difference, not a ranking; the pipelined studies' throughput/Fmax lead (Section 4.5) stands unchallenged.

As Table 7 shows, the throughput/Fmax advantage of [12, 15, 16, 18] reflects their pipelined architecture choice and is not disputed here. The contribution of this work lies in the columns to the right - a measured post-synthesis-to-post-route gap, controlled cross-device validation, a complete signed-off ASIC flow, and cell-level DRC verification - none of which the reviewed FPGA-only studies report (Section 2.6).

Given the ≈ 172 MHz ceiling here versus 210 MHz-1.35 Gbps for pipelined alternatives [12], the iterative architecture suits area/LUT-FF-constrained targets rather than throughput-critical ones. Since the bottleneck is wire-length-, not logic-, dominated (Section 3.2), pipelining is likely a more effective remedy than further fan-out mitigation.

4.6. Limitations and Open Items

The following items are noted for completeness: the first remains open, the second has since been resolved (Section 4.4), and the third bounds the generality of the measured timing gap:

- The place-and-route attempt on gscl45nm did not fully converge to a clean Innovus-native verify_drc pass within the routing budget used here. Two candidate confounds were ruled out: a power/ground net-naming inconsistency (both flows use library-native net names) and routing congestion (NanoRoute reported 0.00 congestion on five of six routing layers, negligible on the sixth). With both eliminated, the residual behavior is consistent with Section 4.4's cell-level DRC finding: Innovus's in-loop DRC (a reduced rule subset for engineering-change-order (ECO) convergence) resolved routing-stage violations to a small residual, whereas signoff-grade Calibre verification with the full rule deck reported substantially more - compatible with a fixed per-instance defect propagating across cell instances, though this does not on its own distinguish a genuine layout defect from a grid-convention mismatch; conclusive attribution is left as future work.
- [Resolved] The same-cell-type kmaLIB comparison (Table 6) is complete: INVX1, NAND2X1, and DFF all returned zero violations under the identical deck, matching BUFX1 and eliminating structural complexity as a confound. Residual caveat: kmaLIB's flip-flop cell is named DFF, not DFFPOSX1; formal functional-equivalence between them has not yet been confirmed.
- The 2.8× gap (Section 4.1) was measured only on the feedback-select path, not on other high-fan-out nets or designs; the ratio is not a general constant - only its direction (post-synthesis underestimates net delay) is expected to generalize (Section 2.3).

5. CONCLUSION

This paper presented a measurement-driven analysis of the timing bottleneck inherent to iterative Keccak-f[1600] architectures: the routing structure required by the 1600-bit feedback-select fan-out, not logic depth, dominates the critical path, and this cannot be reliably assessed from post-synthesis estimates alone (up to 2.8× underestimation observed on this path; the ratio is not claimed to generalize across designs). Three targeted interventions - two fan-out-reduction techniques and a multicycle-path exception exploiting the feedback-select signal's low transition rate - were each evaluated by post-route measurement, and none improved the operating frequency: the fan-out techniques reduced the targeted path's slack, while the multicycle exception (functionally verified under both reset and back-to-back conditions) relaxed that path but left the design-level worst slack unchanged as the constraint migrated to other wide control nets. These results characterize the bottleneck as a distributed, wire-length-

dominated limitation rather than a localized effect removable by a point technique. Cross-device implementation on a higher-performance FPGA fabric confirmed the bottleneck to be architectural rather than technology-specific.

A complete RTL-to-GDSII ASIC flow using an in-house-characterized 45 nm standard-cell library achieved signoff with positive timing slack and zero DRC/LVS violations. A parallel attempt using a widely used public reference library revealed a consistent class of off-grid violations intrinsic to that library's distributed layouts (confound-controlled cell-level DRC, Section 4.4) - consistent with previously documented friction in the same open-source ecosystem. A matching same-cell-type check found zero violations across all four cell types examined, ruling out structural complexity as an alternative explanation - a practical caution for future academic ASIC studies relying on the public library. Future work includes the P&R convergence analysis flagged in Section 4.6, and extending the cross-device timing study to the ASIC domain via multi-corner characterization.

REFERENCES

[1]. National Institute of Standards and Technology, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, FIPS PUB 202 (2015).

[2]. G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, The Keccak reference, Keccak Team (2011).

[3]. National Institute of Standards and Technology, Module-Lattice-Based Digital Signature Standard, FIPS PUB 204 (2024).

[4]. National Institute of Standards and Technology, Module-Lattice-Based Key-Encapsulation Mechanism Standard, FIPS PUB 203 (2024).

[5]. R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, D. Stehlé, CRYSTALS-Kyber (version 3.02) - Submission to Round 3 of the NIST PQC Project (2021).

[6]. Aikata, A. C. Mert, M. Imran, S. Pagliarini, S. S. Roy, KaLi: A crystal for post-quantum security using Kyber and Dilithium, IEEE Transactions on Circuits and Systems I: Regular Papers, 70(2) (2022) 747-758.

[7]. A. Dolmeta, M. Martina, G. Masera, Hardware architecture for CRYSTALS-Kyber post-quantum cryptographic SHA-3 primitives, Proc. 18th Conf. Ph.D. Research in Microelectronics and Electronics (PRIME), Valencia, Spain (2023) 209-212.

[8]. T. Kamucheka, A. Nelson, D. Andrews, M. Huang, A Masked Pure-Hardware Implementation of Kyber Cryptographic Algorithm, Proc. Int. Conf. Field-Programmable Technology (ICFPT) (2022).

[9]. P. Karl, J. Schupp, T. Fritzmann, G. Sigl, Post-Quantum Signatures on RISC-V with Hardware Acceleration, ACM Transactions on Embedded Computing Systems, 23(2) (2024) 30.

[10]. M. Sundal, R. Chaves, Efficient FPGA implementation of the SHA-3 hash function, Proc. IEEE Computer Society Annual Symposium on VLSI (ISVLSI) (2017) 86-91. https://doi.org/10.1109/ISVLSI.2017.24

[11]. A. Dolmeta, M. Martina, G. Masera, Comparative Study of Keccak SHA-3 Implementations, Cryptography, 7(4) (2023) 60. https://doi.org/10.3390/cryptography7040060

[12]. A. Abdulsamad, S. R. Répáš, Design of an Energy-Efficient SHA-3 Accelerator on Artix-7 FPGA for Secure Network Applications, Computers, 15(1) (2026) 3. https://doi.org/10.3390/computers15010003

[13]. A. Sideris, T. Sanida, M. Dasygenis, High throughput implementation of the Keccak hash function using the NIOS-II processor, Technologies, 8(1) (2020) 15. https://doi.org/10.3390/technologies8010015

[14]. A. Sideris, T. Sanida, M. Dasygenis, High throughput pipelined implementation of the SHA-3 cryptoprocessor, Proc. 32nd Int. Conf. Microelectronics (ICM) (2020) 1-4. https://doi.org/10.1109/ICM50269.2020.9331803

[15]. A. Sideris, T. Sanida, M. Dasygenis, A Novel Hardware Architecture for Enhancing the Keccak Hash Function in FPGA Devices, Information, 14(9) (2023) 475. https://doi.org/10.3390/info14090475

[16]. A. Sideris, T. Sanida, M. Dasygenis, Hardware acceleration design of the SHA-3 for high throughput and low area on FPGA, Journal of Cryptographic Engineering, 14 (2024) 193-205.

[17]. A. Sideris, T. Sanida, D. Tsiktsiris, M. Dasygenis, Acceleration of Image Processing with SHA-3 (Keccak) Algorithm using FPGA, Journal of Engineering Research and Sciences, 1(7) (2022) 20-28. https://doi.org/10.55708/js0107004

[18]. A. E. Putra, O. Natan, J. E. Istiyanto, Optimizing FPGA Resource Allocation for SHA-3 Using DSP48 and Pipelining Techniques, IIUM Engineering Journal, 26(1) (2025) 240-253. https://doaj.org/article/b9cca29bd44441cdb2ebf90f7b146ff6

[19]. A. Kermiche, M. Saoudi, A. Drouiche, High throughput pipelined implementation of SHA3 hash algorithm on FPGA, Journal of Cryptographic Engineering, 15(2) (2025) 15. https://doi.org/10.1007/s13389-025-00379-3

[20]. Lục Như Quỳnh et al., Triển khai thuật toán băm SHA-3 dựa trên kiến trúc RISC-V sử dụng phần cứng Arty A7-35T, Hội thảo khoa học "Công nghệ mới và ứng dụng trong lĩnh vực Điện - Điện tử - Tự động hóa", Trường Đại học Giao thông vận tải (2026).

[21]. A. Akın, A. Aysu, O. C. Ulusel, E. Savaş, Efficient hardware implementations of high throughput SHA-3 candidates keccak, luffa and blue midnight wish for single- and multi-message hashing, Proc. 3rd Int. Conf. Security of Information and Networks (SIN) (2010) 168-177.

[22]. G. S. Athanasiou, G.-P. Makkas, G. Theodoridis, High throughput pipelined FPGA implementation of the new SHA-3 cryptographic hash algorithm, Proc. 6th Int. Symp. Communications, Control and Signal Processing (ISCCSP) (2014) 538-541.

[23]. C. Ewert, A. Sharma Poudel, M. Ayache, A. Neskovic, R. Buchty, M. Berekovic, S. Berndt, S. Mulhem, Lightweight Unified SHA-3/SHAKE Architecture with a Fault-Resilient State, arXiv:2512.03616 (2025).

[24]. J. Zhang, Y. Yan, J. Huang, Ç. K. Koç, Optimized Software Implementation of Keccak, Kyber, and Dilithium on RV{32,64}IM{B}{V}, IACR Transactions on Cryptographic Hardware and Embedded Systems, 2025(1) (2025) 632-655. https://doi.org/10.46586/tches.v2025.i1.632-655

[25]. D. Shin, Y. Kim, A. Khalid, M. O'Neill, S. C. Seo, Optimized Implementations of Keccak, Kyber, and Dilithium on the MSP430 Microcontroller, IACR Transactions on Cryptographic Hardware and Embedded Systems, 2026(2) (2026) 409-440. https://doi.org/10.46586/tches.v2026.i2.409-440

[26]. Đàm Đức Trung, Đặng Tuấn Anh, Thực thi thuật toán băm BLAKE3 trên bo mạch phát triển Raspberry Pi, Tạp chí Khoa học Giao thông vận tải, 76(5) (2025) 701-712. https://doi.org/10.47869/tcsj.76.5.5

[27]. V. A. Chhabria, W. Jiang, A. B. Kahng, S. S. Sapatnekar, From Global Route to Detailed Route: ML for Fast and Accurate Wire Parasitics and Timing Prediction, Proc. ACM/IEEE Workshop on Machine Learning for CAD (MLCAD) (2022). https://doi.org/10.1145/3551901.3556475

[28]. V. A. Chhabria, W. Jiang, A. B. Kahng, S. S. Sapatnekar, A Machine Learning Approach to Improving Timing Consistency between Global Route and Detailed Route, ACM Transactions on Design Automation of Electronic Systems, 29(1) (2023) 18. https://doi.org/10.1145/3626959

[29]. H. Jiang, Y. Guo, S. Guo, H. Liu, X. Li, N. Wang, Z. Di, Critical Path Aware Timing-Driven Global Placement for Large-Scale Heterogeneous FPGAs, arXiv:2512.00038 (2025).

[30]. M. Darvishi, Practical Timing Closure in FPGA and ASIC Designs: Methods, Challenges, and Case Studies, arXiv:2510.26985 (2025).

[31]. J. E. Stine, I. Castellanos, M. Wood, J. Henson, F. Love, W. R. Davis, P. D. Franzon, M. Bucher, S. Basavarajaiah, J. Oh, R. Jenkal, FreePDK: An Open-Source Variation-Aware Design Kit, Proc. IEEE Int. Conf. Microelectronic Systems Education (MSE) (2007) 173-174. https://doi.org/10.1109/MSE.2007.44

[32]. J. Grad, J. E. Stine, A standard cell library for student projects, Proc. IEEE Int. Conf. Microelectronic Systems Education (MSE) (2003) 98-99. https://doi.org/10.1109/MSE.2003.1205272

[33]. C. H. M. Oliveira, M. T. Moreira, R. A. Guazzelli, N. L. V. Calazans, ASCEnD-FreePDK45: An open source standard cell library for asynchronous design, Proc. IEEE Int. Conf. Electronics, Circuits and Systems (ICECS) (2016) 652-655. https://doi.org/10.1109/ICECS.2016.7841286

[34]. M. L. L. Sartori, M. T. Moreira, N. L. V. Calazans, ASCEnD-FreePDK45: a free standard cell library for SDDS-NCL circuits (GitHub repository). <https://github.com/marlls1989/ascend-freepdk45>, (accessed 3 October 2026).

[35]. C. Torng, freepdk-45nm: ASIC design kit for FreePDK45 + Nangate for use with mflowgen (GitHub repository). <https://github.com/mflowgen/freepdk-45nm>, 2019, (accessed 3 October 2026).

[36]. R. Thapa, S. Ataei, J. E. Stine, WIP: Open-source standard cell characterization process flow on 45 nm (FreePDK45), 0.18 µm, 0.25 µm, 0.35 µm and 0.5 µm, Proc. IEEE Int. Conf. Microelectronic Systems Education (MSE) (2017) 5-6. https://doi.org/10.1109/MSE.2017.7945072