International Journal for Modern Trends in Science and Technology
Volume 11, Issue 05, pages 154-157
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol11issue05.html
DOI: https://doi.org/10.5281/zenodo.15271230

Compact Implementation of SHA3-256 Bit

V Swarna Latha[1], Kolagatla Narendra[2], Batta Jyothi[3], Desaboyina Naga Ganga Praveen[4], Karanam Venkata Ramana[5]

Amrita Sai Institute of Science and Technology, Andhra Pradesh, India.
[1]swarna.j56@gmail.com, [2]narendrakolagatla45@gmail.com, [3]battajyothi2021@gmail.com, [4]desaboinapraveen3@gmail.com, [5]Karanamvenkatramana385@gmail.com

To Cite this Article
V Swarna Latha, Kolagatla Narendra, Batta Jyothi, Desaboyina Naga Ganga Praveen & Karanam Venkata Ramana (2025). Compact Implementation of SHA3-256 Bit. International Journal for Modern Trends in Science and Technology, 11(05), 154-157. https://doi.org/10.5281/zenodo.15271230

Article Info
Received: 26 March 2025; Accepted: 19 April 2025.; Published: 23 April 2025.

KEYWORDS

ABSTRACT

In the era of internet and computer networking the need for security have increased rapidly. Various crypto algorithms are used for secured data transmission and reception through the network, of which hash function possess a key role in various cryptographic protocols. Keccak algorithm is the winner of SHA-3 competition conducted by NIST. SHA-3 consists of different variant such as 224, 256, 384 and 512 bit. This paper discussesthedesignandimplementationofSHA-3256-bitcore.Thecoreisdesignedusing Verilog HDL and prototyped using Xilinx Virtex6FPGA

1. INTRODUCTION

The SHA3-256, part of the SHA-3 family standardized by NIST, is a cryptographic hash function based on the Keccak sponge construction. Unlike earlier SHA-2 algorithms, SHA3-256 is designed with a fundamentally different structure, offering increased resistance against certain cryptographic attacks. It produces a fixed 256-bit (32-byte) output hash, regardless of the input size, making it suitable for applications like digital signatures, message authentication, and data integrity checks.

At its core, SHA3-256 uses the sponge construction, which divides the hashing process into two main phases: absorb and squeeze. In the absorbing phase, input data is XORed into the state in fixed-size blocks and mixed using the Keccak-f [1600] permutation. Once the entire message has been absorbed, the squeezing phase begins, where the output hash is extracted from the transformed state.

The Keccak-f[1600] permutation is an on linear transformation applied to a 5×5 matrix of 64-bitwords (totaling 1600 bits). It operates over 24 rounds, each consisting of five steps: theta (mixing columns), rho (rotating bits), pi (rearranging positions), chi (nonlinear mixing), and iota (adding a round constant). This permutation is responsible for the diffusion and avalanche effect that gives SHA3-256 its cryptographic strength.

In implementation, SHA3-256 can be compactly realized by carefully managing bit-level operations, padding the input with a domain-specific suffix (0x06), and correctly applying the Keccak-f permutation to process each message block. Its clean structure and reliance on simple bitwise operations make it not only secure but also suitable for lightweight and efficient software or hard implementations.

One of the defining features of SHA3-256 is its flexibility and resilience. UnlikeSHA-2, which relies on Merkle–Damgård construction, SHA-3's sponge structure allows variable-length output and inherent resistance to length-extension attacks. Additionally, the separation of capacity and rate in the sponge model enables fine-tuning of the security margin, which is particularly useful for different application requirements, from high-security environments to constrained embedded systems.

In practice, a compact implementation of SHA3-256 involves managing only a few core operations: XOR, rotation, bitwise logic, and array indexing. This makes the algorithm relatively easy to implement in low-level languages like C, and efficient in hardware environments such as FPGAs or ASICs. Even in high-level languages like Python, a minimal implementation can be both concise and educational, illustrating key cryptographic principles while maintaining functional correctness and standard compliance.

SHA3-256 has gained widespread adoption in both academic and industrial applications due to its robust design and proven security. It was the result of a public competition hosted by NIST, which evaluated multiple candidates over several years before selecting Keccak for its simplicity, flexibility, and resistance to known crypto graphic vulnerabilities. Unlike previous hash functions, Keccak was built from the ground up with security as a first-class principle, rather than as an extension or evolution of older algorithms.

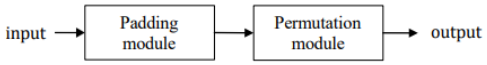
From a developer's perspective, SHA3-256 is often preferred in systems where future-proofing and versatility are critical. For instance, it is used in secure bootloaders, blockchain applications, digital signatures, and zero-knowledge proofs. Its ability to be customized through variations like SHAKE (extendable-output functions) allows it to adapt beyond traditional hash roles into stream-like crypto graphic applications. Its compact logical so makes it ideal for resource-limited plat for mssuch as microcontrollers or smart cards.

Moreover, SHA3-256 helps address some of the limitations observed in the SHA-2 family, including structural similarities across variants that posed theoretical concerns. With SHA3, those concerns are mitigated thanks to its non-Merkle–Damgård structure and sponge design, which inherently prevents many attack vectors ,including length extension and collision attacks that have challenged earlier hash designs. As result, SHA3-256 is not only a drop-in replacement in many contexts but often a strategic upgrade for long-term security and performance.

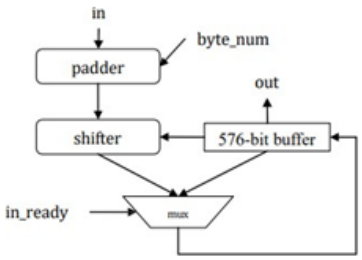
2. ARCHITECTURE

2.1 Architecture of the core:

The architecture depicted as follows is of the whole core. Two cores implemented in this project have the same architecture.

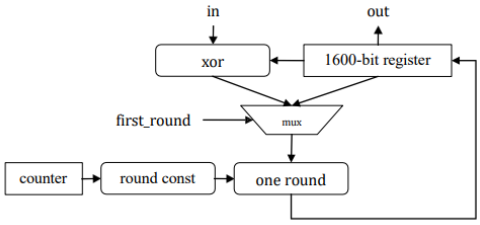


The architecture of the padding module is illustrated in the figure below. The width of the user input is farless than 576 bit. So the padding module uses a buffer to assemble the user input. If the buffer grows full, the padding module notices the permutation module its output is valid. Then the permutation module begins calculation, the buffer cleared, the padding module waiting for nput simultaneously.

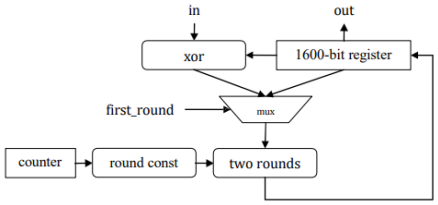


Architecture of the permutation module:

The permutation module of the low throughput core is composed of a combinational logic block computing around, a counter selecting the round constant, and a register storing the output.



In the high through put core, two round sared one per clock cycle.



The round constant module is implemented by combinational logic, saving resource than blockRAM, because most bits of the round constant is zero.

3. IMPLEMENTATION

XILINXISE13.1:

Xilinx, Inc. is the world's largest supplier of programmable logic devices, the inventor of the field programmable gate array (FPGA) and the first semiconductor company with a fabless manufacturing model.

Xilinx designs, develops and markets programmable logic products including integrated circuits(ICs), software design tools, predefined system functions delivered as intellectual property (IP) cores, design services, customer training, field engineering and technical support. Xilinx sells both FPGAs and CPLDs programmable logic devices for electronic equipment manufacturers in end markets such as communications, industrial, consumer, automotive and data processing.

Xilinx's FPGAs have even been used for the ALICE(ALargeIon Collider Experiment)at the CERN European laboratory on the French-Swissbordertomapanddisentanglethetrajectoriesofthousands of subatomic particles.

The Vertex-II Pro, Virtex-4, Virtex-5, and Virtex-6 FPGA families are particularly focused on system-on-chip (SOC) designers because they include up to two embedded IBM PowerPC cores. The ISE Design Suite is the central electronic design automation (EDA) product family sold by Xilinx. The ISE Design Suite features include design entry and synthesis supporting Verilog or VHDL, place-and- route (PAR), completed verification and debug using Chip Scope Protools, and creation of the bit files that are used to configurethechip.XST-XilinxSynthesisTechnologyperformsdevicespecificsynthesis for Cool Runner XPLA3/-II and XC9500/XL/XV families and generates an NGC file ready for the CPLD fitter.

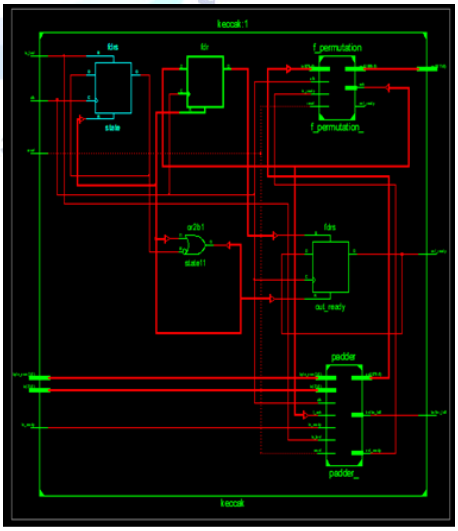
Verilog HDL

Verilog HDL is one of the two most common Hardware Description Languages (HDL) used by integratedcircuit(IC)designers. HDL's allows the design to be simulated earlierin the design cyclein order to correct errors or experiment with different architectures. Designs described in HDL are technology-independent, easy to design and debug, and are usually more readable than schematics, particularly for large circuits. Verilogisusedtodescribetheanydigitallogiccircuitisanint erconnectionofports.Themodeling techniques are

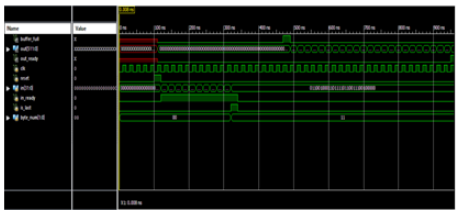
- Structural
- Behavioral
- Dataflow.

4. RESULT ANALYSIS

RTL SCHEMATIC



SHA3 RTL SCHEMATIC OUTPUT



Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Guido Bertoni, JoanDaemen, Michaël Peetersand GillesVan Assche,"The Keccak sponge function family: Specifications summary", http://keccak.noekeon.org/specs_summary.html

[2] "NIST Selects Winner of Secure Hash Algorithm (SHA-3) Competition", NIST .Oct .2012.\ http://www.nist.gov/itl/csd/sha-100212.cfm

[3] "SHA-3",Wikipedia,the freeen cyclopedia, http://en.wikipedia.org/wiki/SHA3 TheKeccak reference, version3.0,http://keccak.noekeon.org/Keccak-reference-3.0.pdf

[4] Keccak implementation overview, version3.2, http://keccak.noekeon.org/Keccak-implementation-3.2.pdf

[5] "Announcingrequestforcandidatealgorithmnominationsforane wcryptographichash algorithm (SHA-3) family", Federal Register Notices 72 (2007), no. 212, 62212–62220 http://csrc.nist.gov/groups/ST/hash/index.html

[6] K.Gaj,E.Homsirikamol,M.Rogawski,R.Shahid,andM.U.Sharif," Comprehensiveevaluation of high-speed and medium-speed implementations of Five SHA-3 Finalists using Xilinx and Altera FPGAs", 3rd SHA-3 candidate conference, Mar 2012.