DESIGN OF SHA3 ARCHITECTURE FOR SYSTEM SECURITY

PANDIRI PADMA
Assistant Professor
Department of ECE,UCE,
Osmania University
padmadandyala@gmail.com

DEEPIKA M.E
Student
ECE, Osmania University
Email-
mdeepika235@gmail.com

Prof. RAMESHWAR RAO
ECE, Osmania University
Email-
rameshwar_rao@hotmail.com

Abstract:

High performance Secure Hash Algorithm (SHA-3) software realization is investigated and proposed in this work. In Today's world every person relies on internet for various purposes. So there are chances to forge your data. We always need to take appropriate measures for all the way throughout this insecurity internet. There are various algorithms that ensure security. Cryptography is used to protect the personal or important data from the unauthorized people who try to access it. Cryptography is widely used by government and intelligence agencies around the world for the transmission of information. Many light weight cryptographic algorithms have been developed and SHA-3, a subset of the cryptographic primitive family Keccak is a cryptographic hash function for its many admirable qualities, including its elegant design and its ability to run well on many different computing devices. The clarity of Keccak's construction lends itself to easy analysis and it has higher performance in hardware implementations The design employs unrolling, pipelining and sub pipelining methods and by logically designing Theta, Rho, Pi and Chi and Iota steps of algorithm wecan see the change in the throughput. The performance is analyzed in terms of its architecture and throughput. A new Design of SHA3 which results in improved throughput is observed in casevi it is simulated, synthesized using Xilinx tool.

I. Introduction:

The main contribution of this work is directed towards the light weight design or architecture with high performance. First, a simplified round constant generator for SHA-3 hash is introduced, which it requires much smaller hardware re-sources compared to the conventional design. Second, a new2-stages sub pipelined transformation round is employed where the register is inserted after the Theta (θ) step in the hash permutation function. Third, several architectural designs that are suitable for both single and multi-message hashings are investigated and hence resulted in five different SHA-3 architectures (represented in Case I-V).

Among all the implementations the new architecture that has 3-stages sub pipelined and unrolling by factor 3, which is followed by 3-stages pipeline in between the adjacent rounds. The experimental results proved that Case VI (case V with unrolling factor of 3) offers higher throughputperformance and provides better efficiency compare to the other cases. Note that the aim of the Study is to introduce performance enhancement so hardware area is not in characteristics

II. The sha-3 function:

The Keccak hash function, designed by G. Bertoni , was announced by the NIST as the new Secure Hash Algorithm-3, i.e. SHA-3, in 2015. Generally, the Keccak algorithm is based on sponge construction, where the hash transformation is performed on an internal state that takes input of

arbitrary length, and produce an output of the desired length. Depending on the selection of hash variants, the sponge construction will be build using appropriate bitrate (r) and capacity (c), where these parameters also serves as the main factor that determines the overall security level.

In a brief, the SHA-3 hash function is composed of two phases which are shown in Figure 1. During the **absorbing phase**, the bitrate of the initialized state is XORed with the first part of the input. The new bitrate and together with the capacity of the initialized state matrix will form a new state that is used in f-permutation. The resulting state will serve as the new initial state for the next round and the process continues for 24 iteration rounds. Each round is divided into five separate steps, i.e. Theta (θ), Rho (ρ), Pi (π), Chi (χ) and Iota. The equations of each step are given below.

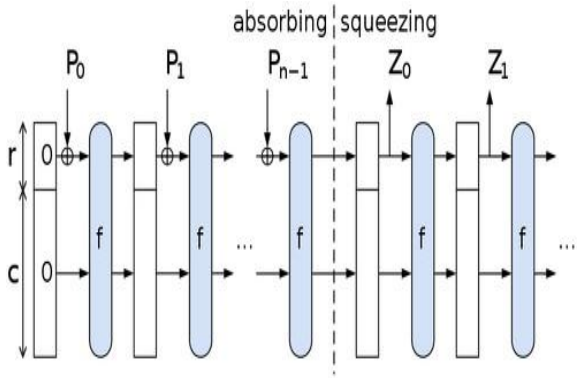


Fig 2: Sponge construction

III. Hardware architectures for hash functions:

Several research studies in the literature have presented various hardware implementations of SHA-3 hash function. The implementation's aim is directed either towards the lightweight design or architecture with high speed performance. Later various approaches like unrolling, pipelining and sub pipelining are used on the top of sha3 block for maximum throughput enhancement. This methodologies offer speed improvements in different manners .unrolling is particularly efficient in increasing the throughput for single long message whereas pipelining can increase combined throughput for multi messages. There are total five implementations of sha-3 are demonstrated as below. For instance, the straightforward approach is done by placement of pipeline registers in between the adjacent rounds within the unrolled hash architecture (Fig 3.1(a)).In other words, the hash function is implemented with unrolling of factor 2 (Fig 3.1(b)) and followed by 2-stages pipelining consequently, two different messages can be processed simultaneously in such implementation (fig 3.1 (c). Taking into the consideration of the combined throughput of all the available data stream, the overall hash function performance is in fact enhanced by a factor of two. Such achievement will be significantly beneficial in the case where the number of message blocks is relatively large another plausible means of pipelining is by insertion of pipeline registers inside the hash function round. To be exact, the sub-pipelining is applied in between the interval steps within the f-permutation block. In this methodology, the attainable throughput improvement is relative to the maximum reduction in the function's critical path. Thus, the challenge comes in determining the appropriate placement of register pipeline in order to achieve the minimum critical path in SHA3 function (Fig 3.2). The new sha3 architecture which enhances the throughput is shown in Fig 3.3 The different architectures are shown below.

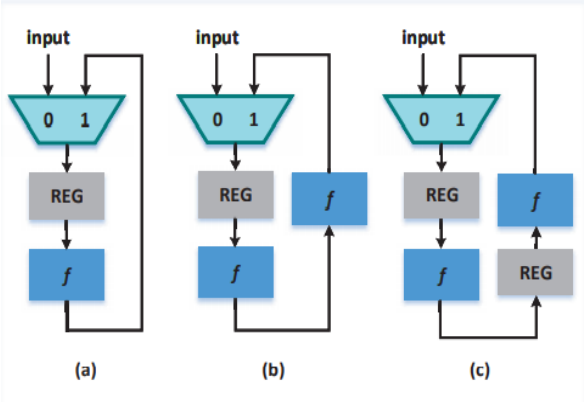


Fig 3.1 Architecture of SHA-3 Hash Function (a) Basic iterative (b) Unrolling (k=2) (c) Unrolling(k=2) with pipelining (n=2).

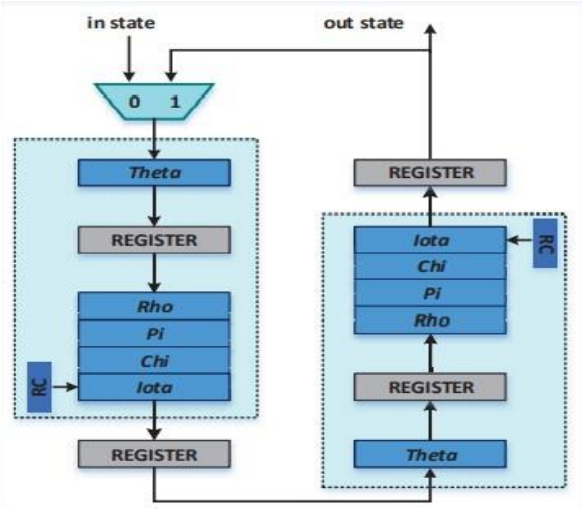


Fig 3.2: SHA-3 hash implementation in Case V.

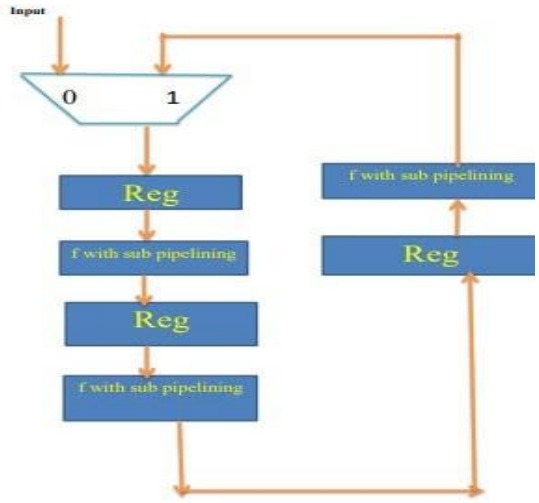
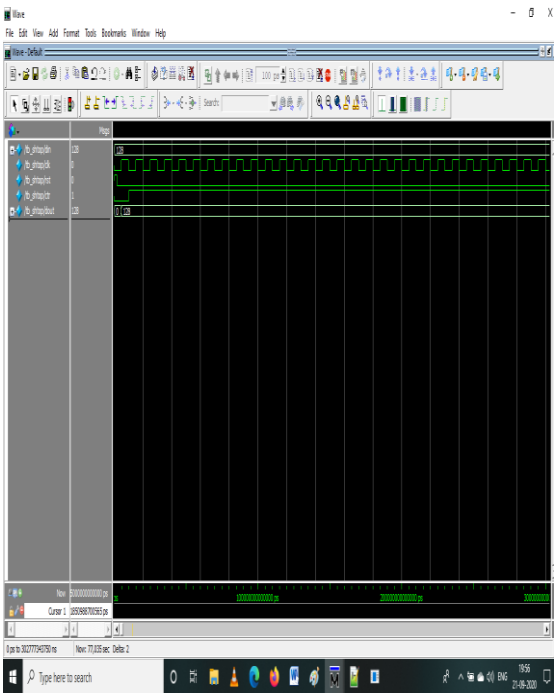


Fig 3.3: The new sha3 hash architecture

The SHA-3 hardware architectures (for Cases I, II, III, IV V and VI) were implemented using Verilog language, simulated by ModelSim and synthesized with Xilinx ISE Design Suite. The targeted FPGA platform was XC6VLX75T from Xilinx Virtex-6 family.



The simulations results of sha3 algorithm are shown in the following figure.

The FPGA implementation results are investigated in terms of the achievable frequency (maximum), area, throughput and efficiency and these metrics are tabulated in Table II. It is observed that the new sha3 implementation has high throughput In general, the throughput for multi-message hashing is computed as follows.

$$\text{Throughput} = \frac{\#blocksize \times F_{max}}{\#clockcycle} \times \#N_{msg}$$

Sha-3 hash architectures	Area	Delay(ns)	Fmax(Mhz)	No of clock cycles	Throughput
Case-1	2095	1.854	539.37	24	12,944.88
Case-2	2205	1.618	618.167	24	14,835.88
Case-3	3317	3.794	264.250	12	12,684
Case-4	3183	1.878	532.425	12	25,556.4
Case-5	4125	1.526	665.480	12	31,943.04
Case-5 with rolling factor 3	5431	1.321	756.71	8	54,483.12

Conclusion:

The hash function Keccak was presented we used simplified round

constant generator which is hardware cost effective for SHA-3 hash function and a new inner f-permutation sub pipelining approach was demonstrated. Incorporated with 2-stages sub pipelined and unrolling by factor 2, followed by 2- stages pipeline in between the adjacent rounds, a new SHA-3 hardware realization was presented in Case V and we also checked by increase the rolling factor to the existed caseV.we used Xilinx ISE 14.7 for synthesis and modelsim for simulation. Based on the experimental results the new sha-3 implementation with rolling factor 3 was proven high in throughput performance. And the whole design has a simple hardware structure and fast running speed and can be widely used in digital signatures and 3DES key generation systems.

References:

1. *B. Baldwin, A. Byrne, L. Lu, M. Hamilton, N. Hanley, M. O'Neill,and W. P. Marnane, "FPGA implementations of the round two SHA-3candidates," in 2010 International Conference on Field Programmable Logic and Applications, Aug 2010.*
2. *H. Mestiri, F. Kahri, M. Bedoui, B. Bouallegue, and M. Machhout, "High throughput pipelined hardware implementation of the KECCAK hash function," in 2016 International Symposium on Signal, Image, Video and Communications (ISIVC), Nov 2016,*
3. *George S. Athanasiou, George-Paris Makkas, Georgios Theodoridis "High throughput pipelined FPGA implementation of new sha3 cryptographic hash algorithm" In 2014 IEEE.*
4. *Ming Ming Wong, Jawad Haj-Yahya, Suman Sau and Anupam Chattopadhyay School of ComputerScience and Engineering (SCSE)," A New High Throughput and Area Efficient SHA-3 Implementation" in 2018 IEEE International symoposium on circuits and systems.*