ECE 646

Initial Project Specification:

1. Team:
   Sanjay Deshpande.

2. Title:
   Analysis and pipelined implementation of selected parallelizable CAESAR candidates.

3. Introduction and motivation. Placement of the problem in the broader research area. Why is this project worth working on? Why is it original? Why is it practical?

   Introduction:

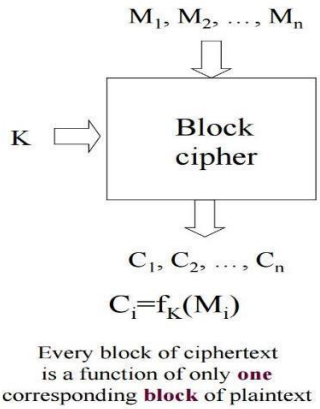

Fig 1: Block Diagram

Various block-cipher modes for authenticated encryption are inherently sequential, some to satisfy stricter notions of security, some others to achieve lightweight implementations. We call an encryption operation parallelizable if the processing of the i-th input block does not depend on the output of processing the j-th block, for any i not equal to j. As a slightly weaker kind of this feature, we call an AE scheme pipeline able if the encryption (and likewise the decryption) can be decomposed into operations f ◦ g, such that the first operation g(Mi) can be already performed for the i-th block before the encryption of the previous blocks have finished. Note that we regard parallelizable encryption and decryption separately.

Motivation: The Main Aim of the project is to select two authenticated ciphers which have clearly parallelizable Encryption and Decryption and with lowest possible maximum clock frequency and with relatively small values of w and then modifying the design to increase the w up to 256 and introducing two stages of pipelining within the main round of the cipher core data path and also modifying the cipher core controller accordingly and introducing minimal changes to the GMU preprocessor and postprocessor.

From the available sources of the selected ciphers modify the block diagrams and ASM charts of those ciphers for pipelining also modify the RTL codes and verifying the design using the universal test bench and the universal script for generating test vectors. After that generating optimized results and targeting the FPGAs Virtex 6, Virtex 7 and Zynq then verify your codes and the obtained maximum clock frequencies using timing simulation.

This is original because pipelining is one of the most common methods to increase the throughput. So we are using that technique on the candidate which is low in frequency and throughput and improve its characteristics. This project can be taken as reference for the Candidates which are parallelizable and which doesn't give good results with normal RTL and HLS implementation.

4. Design Entry Method : VHDL
   CAD tools Used to specify, Synthesize, Implement and verify the design is Xilinx ISE 14.7.
   FPGA Tools: Xilinx ISE and Xilinx Vivado
   Optimization Tools: ATHENa

5. Additional Libraries:
   No additional Libraries required for now.

6. Detailed Assumptions
   Presently the focus is on Improving the Throughput. Still have to decide the number by which these factors are improved. Still working on it.

7. Circuit Interface

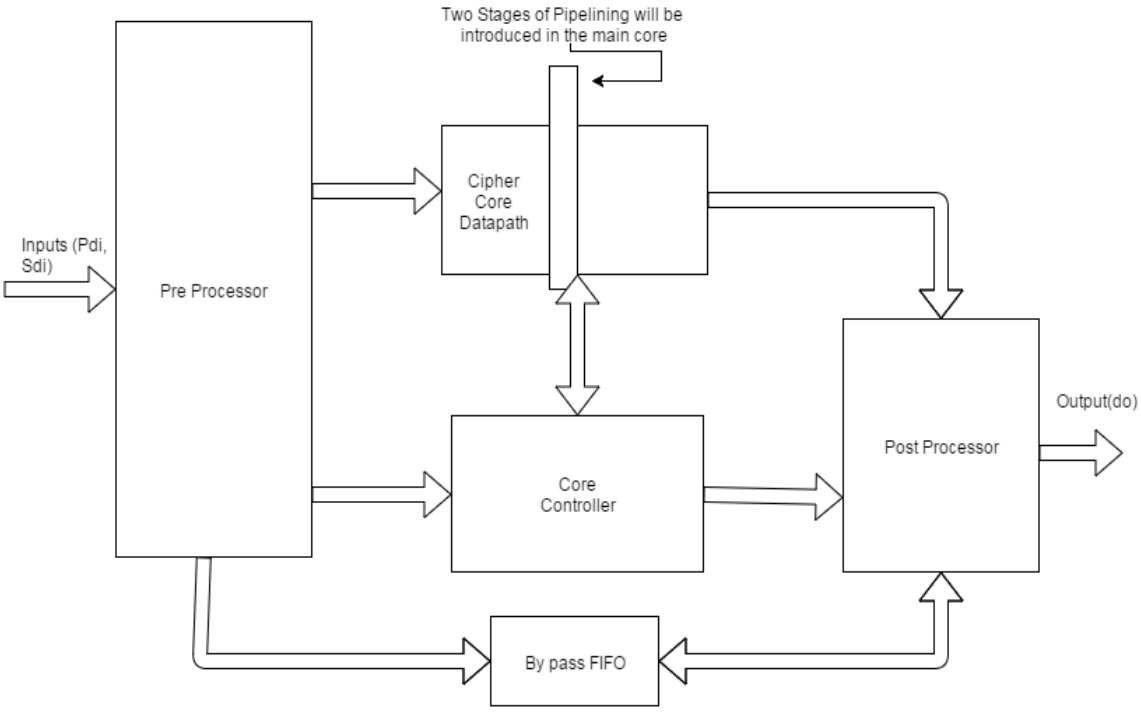

Fig 1: Top Level Interface

M-Messsage
W- Width of Message
C Cipher text

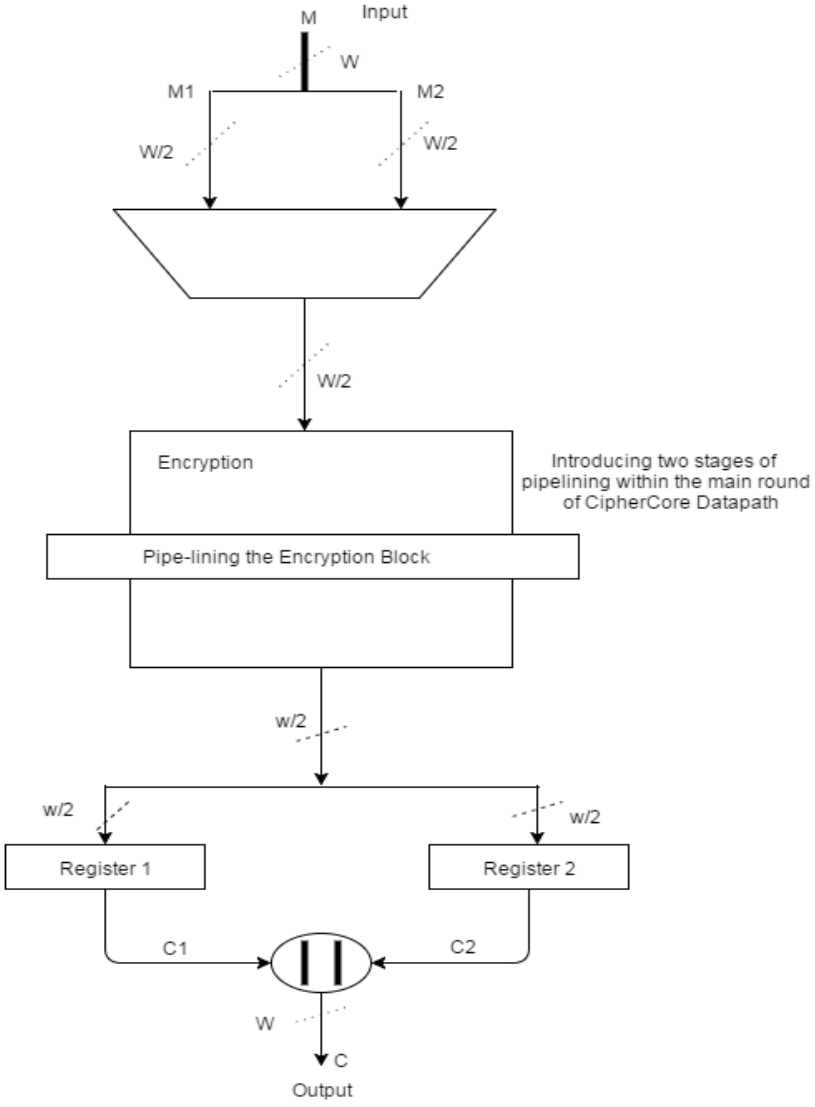

Fig 2: Sample Circuit

8. References to detailed descriptions of the implemented functions:

S no.	Candidate	Key Size	Nonce Size	Tag Size	Basic primitive
1	AES-COPA	128	128	128	AES
2	AES-OTR	256	96	128	AES
3	AEZ	128	96	128	AES-4
4	Deoxys ≠	128|256	64	128	Deoxys-BC, AES
5	Deoxys =	128|256	64	128	Deoxys-BC, AES
6	ELmD	128	64	128	AES
7	Joltik ≠	64|80|96|128	32|24|48|32	64	Joltik-BC, AES
8	Joltik =	64|80|96|128	32|24|48|32	64	Joltik-BC, AES
9	OCB	128	128	64|96|128	XEX
10	POET	128	128	128	ECB
11	SCREAM	128	96	128	SCREAM, SPN
12	SHELL	128	64|80	128	AES
13	SILC	128	64|96	64	AES*(AES-8|AES-12)

Table 1: Key Size, Nonce Size, Tag Size and Primitive

S no.	Candidate	Word Size-w	Block Size-b	#Rounds	#Cycles/Block RTL	Maximum Clock Frequency RTL(MHz)
1	SCREAM	32	128	10	11	101.968
2	AES-COPA	32	128	10	11	150.35
3	POET	32	128	10	11	176.680
4	OCB	32	128	10	11	221.80
5	Deoxys	32	128	14	29	264.201
6	Joltik	32	128	32	65	348.797

Table 2: Word Size, Block Size, Rounds, Cycles/Block RTL and Max Clock Frequency and Ranking According to the Lowest to Highest Frequency

9. List of literature following the IEEE Citation Style Guide

[1] F. Abed, C. Forler, and S. Lucks, “General Overview of the Authenticated Schemes for the First Round of the CAESAR Competition,” Cryptology ePrint Archive: Report 2014/792.

[2] CAESAR submissions, second-round candidates.
Available: http://competitions.cr.yp.to/caesar-submissions.html

[3] E. Homsirikamol, W. Diehl, A. Ferozpuri, F. Farahmand, M.U. Sharif, and K. Gaj, "GMU Hardware API for Authenticated Ciphers," Cryptology ePrint Archive: Report 2015/669.

[4] Cryptographic Engineering Research Group (CERG) at GMU. (2015,Jul.) GMU Hardware API
Available: https://cryptography.gmu.edu/athena/index.php?id=download

[5] E. Homsirikamol, W. Diehl, A. Ferozpuri, F. Farahmand, M.U. Sharif, and K. Gaj, “C vs. VHDL: Benchmarking CAESAR Candidates Using High-Level Synthesis and Register-Transfer Level Methodologies,” presented at Directions in Authenticated Ciphers, DIAC 2015, Singapore, Sep. 28-29, 2015.

[6] Cryptographic Engineering Research Group (CERG) at GMU. (2015, Jul.) GMU ATHENa Database of Results. Online Available at,
:https://cryptography.gmu.edu/athenadb/fpga_auth_cipher/rankings_view

[7] E. Homsirikamol, W. Diehl, A. Ferozpuri, F. Farahmand, M.U. Sharif, and K. Gaj, “C vs. VHDL: Benchmarking CAESAR Candidates Using High-Level Synthesis and Register-Transfer Level Methodologies,”presented at Directions in Authenticated Ciphers, DIAC 2015, Singapore, Sep. 28-29, 2015.

[8] Cryptographic Engineering Research Group (CERG) at GMU. (2015, Jul.) GMU ATHENa Database of Results. [Online].
Available:https://cryptography.gmu.edu/athenadb/fpga_auth_cipher/rankings_view

[10] K. Gaj and P. Chodowiec, "FPGA and ASIC Implementations of AES," Chapter 10 in C.K. Koc (Ed.), Cryptographic Engineering, pp. 235-320, Springer, Dec. 2008.

[11] C. Arnould, “Towards Developing ASIC and FPGA Architectures of High-Throughput CAESAR Candidates,” Master’s thesis, ETH Zurich, March 2015.

[12] K. Gaj, J.-P. Kaps, V. Amirineni, M. Rogawski, E. Homsirikamol, and B. Y. Brewster, “ATHENa – automated tool for hardware evaluation: Toward fair and comprehensive benchmarking of cryptographic hardware using FPGAs,” in 20th International Conference on Field Programmable Logic and Applications - FPL 2010. IEEE, 2010, pp. 414–421.

[13] Xilinx Vivado Design Suite User Guide: Hierarchical Design, April 2015.
Available: http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_1/ug905-vivado-hierarchical-design.pdf

10. Time Schedule including intermediate goals to be achieved by the dates of progress reports:
Will be providing you with detailed description after the selection of Ciphers.

Week	Work
Oct. 26-27	Choosing Two Authenticated Ciphers for implementation of Pipelining
Nov. 9-10	Getting familiar with the Block Diagram of Datapath and Controller, Modifying and Verifying the Block Diagrams.
Nov. 23-24	Conversion of newly pipelined Block Diagram in to HDL, Test Bench and Verification
Dec.7-8	Generating Optimized results targeting Virtex 6, Virtex 7 and Zynq, Obtaining maximum clock frequencies. Writing Final Report and preparing final version of presentation and Adding the results on the ATHENa Database.

11. Tentative table of contents of Final report

S No.	Content
1	Abstract.
2	Introduction to the Authenticated Ciphers that are being used in this project and application of pipelining.
3	Implementation of Pipelining on the Ciphers.
4	Result and comparison of the implementation with pipelining and without pipelining.
5	Conclusion.
6	Future Work.
7	References.