Grenze International Journal of Engineering and Technology, Jan Issue

Implementation of Mellin Transform based Cryptography on FPGA

K.R.K.Sastry
Gayatri Vidya Parishad College of Engineering (Autonomous), Visakhapatnam, India
sasatry_krk@gvpce.ac.in

***Abstract*—Mellin transform due to its scale invariant property has gained attention to various real time applications such as signal and image processing including speech processing, pattern recognition and speech recognition, Mobile, Network, Information, Security, e-mail and Firewall etc. Today, all these applications demand a reconfigurable hardware platform for the implementation.**
In this paper we discuss the Mellin Transform based encryption and Inverse Mellin transform based decryption of for the given data using a polynomial function.

***Index Terms*— Cipher text, Mellin transformation, FPGA.**

I. INTRODUCTION

Cryptography is the practice and study of techniques for secure communication over a insecure medium (wired, wireless) in the presence of third parties (adversaries). Encryption and Decryption process has two aspects: The algorithm and the key. The key is used for encryption and decryption that makes the process of cryptography secure. The Mellin transform is closely related to the Laplace and Fourier transforms and has applications in areas, including: digital data structures, probabilistic algorithms, asymptotics of Gamma-related functions coefficients of Dirichlet series,asymptotic estimation of integral forms, asymptotic analysis of algorithms, communication theory. [1]
The Mellin transform and its inverse are Linear Transformations, related to the two-sided Laplace transform. The Mellin transform has scale invariance property (the magnitude of Mellin transformation of scaled version of signals or images remains same), analogous to the Fourier transform's shift invariance property. This transform is used for the analysis of linear time-invariant systems. This transform is considered as a transformation from the time-domain in which inputs and outputs are functions of time. The transform has significant applications in probability theory, Markov chains, renewal theory, time series, vision and image processing. In particular, a so-called Fourier-Mellin transform can be used for pattern recognition for its invariance to shift, scale, and rotation [2,3,4]
Let f(t)denotes a complex valued function of the real, positive variable t. The Mellin transform for f(t) will be denoted by M[f;s] and defined by

$$M[\ f;\ s\ ] = M[s] = \int_o^\infty f(t)\ t^{s-1}\ dt \qquad (1)$$

In common with other integral transforms, the Mellin transform possesses a series of simple translational properties which greatly facilitate the evaluation of transforms of more involved functions.
The following properties of Mellin transform are considered in our paper:

Grenze ID: 01.GIJET.7.1.519

A. Transform of f(t)] for t>0 is $\int_o^\infty f(t)\, t^s - 1\, dt$
B. Transform of $Af_1(t) + Bf_2(t)$ is $AF_1(s) + BF_2(s)$
C. Scaling, s[f(at)], for a>0 is $a^{-s}F(s)$
D. $s[f(t^a)]$, for a=real ≠0, is $|a|^{-1}F(a^{-1}s)$
E. Multiplication by t^a, $s[t^a f(a)]$, for a>0, is F(s+a) [5]

Cipher depends on choosing a known key only by the sender and the receiver which defines how a particular message would be. The way of sending the key in advance to the receiver by off line communication or other means, for message decipher is not in the purview of cryptography.

Mellin transform is used for the plain text encryption and Inverse Mellin Transform is used for the decryption of the cipher text. This paper discusses Implementation of Mellin Transform based Cryptography with plain text with maximum length of 8 bytes during the design as the divider size of the Xilinx ISE IP core 3.0 is limited to 32-bits divisor. The value of "s" is constant and the rage is 3~5 in this paper.

This paper is organized as follows: Section II introduces a system overview of the implementation. Section III deals with the encryption and Section IV details the decryption engine. Simulation results are shown in section V. Section VI and VII deal with the conclusion and the future scope.

II. System Overview

The top level block diagram is shown in Figure 1 and function of each block is explained below

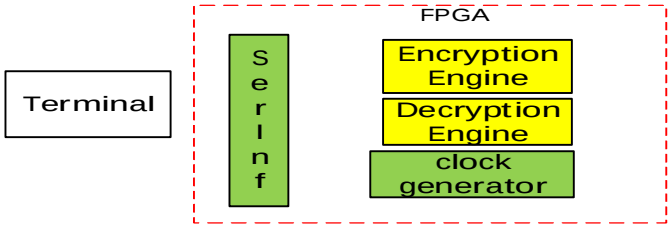

Figure 1 Top level Block diagram

A. Terminal: HyperTerminal is used to send and receive the ASCII plain text at 115 Kbps, 8 bit data, no parity.
B. SerInf: communicates with the Encryption Engine for receiving the encrypted message, and sending the plain text. Also transmits the decrypted message from the decryption engine to the Terminal.
C. Encryption Engine: converts the plain text message (data message length 7 bytes max) according to the Mellin transformation of a polynomial function that contains several constants for each monomial, sends the encrypted message (primary key), and uses a binary to BCD conversion logic module for sending the secondary key along with the delimiter in ASCII format.
D. Decryption Engine: It receives the encrypted message, the secondary key and applies the Inverse Mellin transformation of polynomial function coefficients for decrypting the message. Finite state machine is used for the same.
E. Clock generator: implements necessary logic for generating 115 kbps baud rate clock from 50MHz core clock. [6]

Method for the Mellin encryption of plain text is shown in the following steps

A. Polynomial function
B. $L(x)=e^{-x}\sum_{i=1}^{N} G_i x^i = \sum_{i=1}^{N} e^{-x} G_i x^i$ (1)
   a. G_i are the constants, equal to each character of the Palin text.
C. Using Melli Transform of each member
   a. $L(x)^* = \sum_n G_n(s+n-1)! = \sum_n G_n'$ (2)
   b. Where $G_n' = G_n(s+n-1)!$ (3)
D. Each G_n' is divided by 26, the remainder values (private key) and the quotient (Secondary key) are transmitted.

Plain text values are taken as A=1, B=2, C=3, Z=26.

The Decryption process follows the inverse flow. The process of encryption and decryption is detailed in paper [7].

III. Encryption Engine

The plain text message is read from a terminal and stored as G_n' with a maximum message character length as 8 bytes (for s=3 and s=4) and 7 bytes (for s=5). The encrypted message (Private Key) is transmitted continuously but the secondary key is transmitted as BCD, with a comma character (any other value mutually agreeable to both the sender and the receiver) appended at the end of each value and a CR, LF at the end of the key. The secondary key and the value of "s" (could be different for each message) may be sent on a different channel.
The Encryption engine block diagram is shown in Figure 2. The data flow among the sub-blocks is explained below.

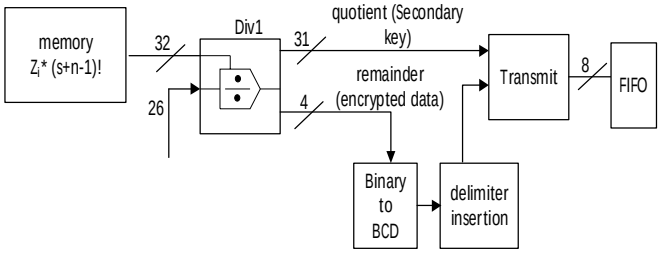

Figure 2 Encryption engine block diagram

The Memory block is a 8 x 32 memory (depth is equal to the maximum message data length, 8 bytes) and stores the values (s+n-1) ! at the time of the initialization. Div1 is an integer divider Xilinx IP core, takes each of the entry of the memory G_n(s+n-1)! and divides it with 26. The quotient of the divider is the encrypted message (private key) and the remainder is the secondary key.
A Binary to BCD converter logic is used to convert the secondary key in to BCD data. Delimiter insertion inserts a comma character (,) between each of the secondary key data and inserts a CR, LF at the last data in the Delimiter block. The Transmit block continuously sends the encrypted data and later sends the secondary key with delimiters in to a 50byte depth FIFO.

IV. Decryption Engine

Maximum value of the remainder data r_i, is 25. The data is stored as 0 in the memory when the value is 26 and corresponding Secondary message is added by 1. After completely receiving r_i, the state machine receives the secondary key. The Decryption engine block diagram is detailed in Figure 3. The data flow among the sub-blocks is explained below.

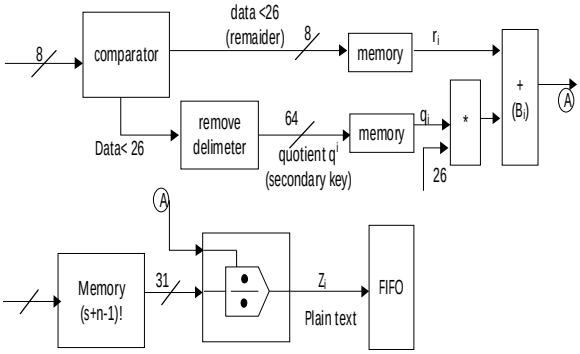

Figure 3 Decryption engine block diagram

The incoming data fed to a comparator block, the remainder is valid if the value <26, and stored in a 8 x1 bytes memory. When the data value ≥ 26, quotient of length 8bytes maximum is assumed transmitted. The delimiter block checks for a comma character and if not found, calculates the intermediate values of $q_i = q_i *10 +$data. The quotients are stored in 8bytes memory. B_i is calculated as $B_i = r_i + q_i *26$, for i<memory length. An integer divider Xilinx IP core is used with B_i as the dividend, **(s+n-1)!** as the divisor. The quotient Z_i, is the decrypted plain text message and is fed to a 50byte depth FIFO.

V. SIMULATION RESULTS

The design is developed in Verilog HDL and it is simulated and synthesized Xilinx 14.3 ISE. Table 1 shows the four test patterns with expected encrypted data and the secondary key of the encryption engine. The simulation results for encryption engine are shown in Figure 4, Figure.5 and decryption engine in Figure 6, Figure 7. Table 2 and Table 3 show the device utilization summary of Encryption and decryption engine respectively.

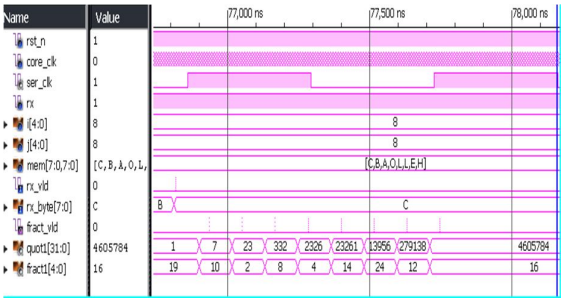

Figure 4 Primary, Secondary key of Encryption Engine

Figure 5 Encryption engine Serial output data

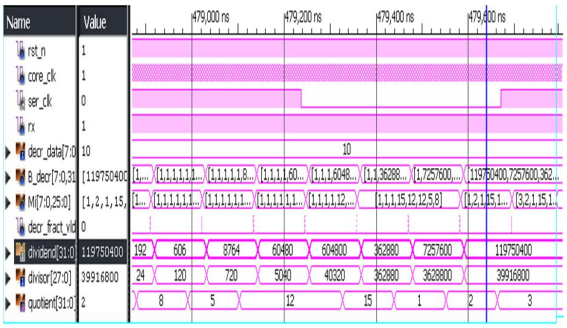

Figure 6 Primary Key of Decryption engine

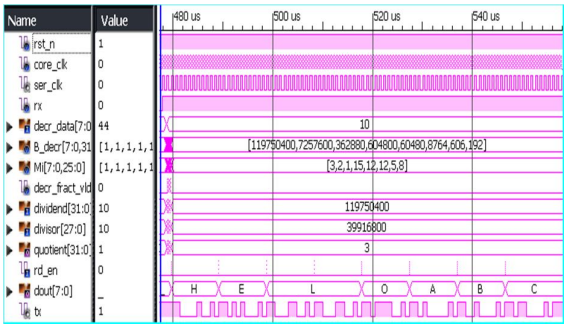

Figure 7.Decryption Engine Serial output data

TABLE I. TEST PATTERN DATA

's' value	Plain data	Encrypted data	Secondary Key
3	HELLOABC	22,16,10,8,18,20,22,18	1,4,55,332,2907,1550,27913,418707
4	HELLOABC	10,2,8,4,4,24,12,16	7,23,332,2326,23261,13956, 279138,4605784
5	HELLOABC	24,12,4,2,22,62	36,138,2326,18609,209353,139569,3070523
4	RANDOMDT	16,16,18,10 14,0,24,20	16,4,387,775,23261,181440,558276,30705230

TABLE II. DEVICE UTILIZATION TABLE - ENCRYPTION ENGINE

Logic Utilization	Used	Available
No. of slice Flip flops	1260	9312
No. of 4 input LUTs	1039	9312
No. of occupied Slices	1242	4656
No. of BRAMs	1	20
No. .of MULT 18x 18SIOs	2	20
No. of bonded IOBs	5	232

Timing Summary:

Minimum Period 10.34 ns

Maximum operating frequency 96.72 MHz

TABLE III. DEVICE UTILIZATION TABLE - DECRYPTION ENGINE

Logic Utilization	Used	Available
No. of slice Flip flops	3714	9312
No. of 4 input LUTs	1833	9312
No. of occupied Slices	2246	4656
No. of BRAMB16s	1	20
No. of MULT 18x 18SIOs	4	20
No. of bonded IOBs	6	232

Timing Summary:

Minimum Period 11.96 ns

Maximum operating frequency 83.63MHz

In this paper, we proposed the implementation of Mellin transform based Cryptography in detail. Encryption engine, Decryption engine with serial interface is successfully developed in Verilog HDL. The Simulation results for encryption and decryption targeted on Xilinx XC3S500E FPGA are discussed. The maximum operating Frequency of 96.72 MHz with device occupancy of 1242 slice registers is achieved for encryption and maximum operating Frequency of 83.63 MHz with device occupancy of 2246 slice registers is realized for

decryption. Value of ‘s’ is varied from 3,4and 5 for a given set of data for providing the complexity for eves dropper.

FUTURE SCOPE

The latency due to serial interface could be optimized with a high speed interface. Error correction mechanism could be added to the proposed work.

REFERENCES

[1] “The Mellin Transform”,The Transforms and Applications Handbook: Second Edition. CRC Press LLC, 2000 Ed.Bertrand et.al

[2] “Mellin’s Transform and Application to Some Time Series Models”, ISRN Applied Mathematics, Vol. 2014, Oct.2007, S. S. Appadoo et.al

[3] “A Fast Mellin and Scale Transform”, EURASIP Journal on Advances in Signal Processing, Vol. 2007, Oct 2007, Antonio De Sena, Davide Rocchesso

[4] “Implementation of Mellin Transform using FPGA based Embedded System Platform”, 2nd International Conference on Power, Control and Embedded Systems, Kshitij et.al

[5] “Asymptotic Nature Of Mellin Transform In Electrical Field”, International Journal of Engineering Research & Technology (IJERT), Vol. 2, April-2013, Dr. N. A. Patil et.al

[6] “Implementation of Sumudu transform based cryptography on reconfigurable hardware “, Journal of Critical reviews, vol 7, issue 19, 2020, K.R.K.Sastry.

[7] “A cryptographic scheme of Mellin transform”, Computer Science - Cryptography and Security, Jan '**14,** Yeray Cachon Santana.

BIOGRAPHY

K.R.K.Sastry received his B.E in 1986 in E.C.E Engineering, Andhra University, Visakhapatnam, India. He completed his M.Tech in 1989 in Optoelectronics and Optical communications, IIT New Delhi, India. He has industrial experience in Telecomm, VLSI and authored many IP cores. His main interests are in VLSI based designs. He is working as Associate Professor, in the Department of Electronics and communication. Engineering, Gayatri Vidya Parishad College of Engineering (Autonomous), Visakhapatnam, India.