Volume 1, Issue 3, July 2014 ISSN: 2348-8565 (Online)

IJMERT

International Journal of Modern Engineering and Research Technology

Website: http://www.ijmert.org Email: editor.ijmert@gmail.com

A New Improved Modulo ($2^n + 3$) Multiplication for DEA

Abhishek Singh Thakur
Research Scholar
Shri Ram Institute of Technology,
Jabalpur (M.P.) [INDIA]
Email: abhishek3012@gmail.com

Prof. Ravi Mohan
Head of the Department
Department of Electronics & Communication Engg,
Shri Ram Institute of Technology,
Jabalpur, (M.P.) [INDIA]
Email: ravimohan7677@yahoo.co.in

Abstract

Data Encryption Algorithm (DEA) is a cipher generator developed by Mr. James Masse from ETH Zurich and Mr. Xuejia La and it was first described in 1991. As a cipher generator, it is symmetric. The method was considered to a replacement for the AES Encryption Standard (AES). DEA is a little revision of a previous cipher, Proposed Encryption Standard, DEA at beginning called Improved PES. International Data Encryption Algorithm is popular and very famous cryptography algorithms from date since when characteristic of DEA is compatible for hardware implementation. This work presents an optimised hardware structure for modulo (2n + 3) multiplier, which is the most time and area consuming operation in DEA. The proposed modulo multiplier provides less time and area cost than older and existing designs. The proposed design allows DEA to be implemented on hardware along with high performance and low cost. Simulation results obtained as platform chosen Vertex FPGA system developed by Xilinx it shows that the new design has 39.71 MHz maximum speed for data encryption and also for decryption with eight pipeline stages for every round.

I. Introduction

The three major arithmetic need of DEA are modulo addition, XOR and modulo multiplication. Modulo addition add up two inputs of n-bit length, and perform mod of the result by $2n$. Modulo multiplication perform multiplication of two inputs of n-bit length, and then take mods the result by ($2n$ +3). also, an input value of zero can considered as $2n$. so, the input length for modulo multiplication is n +3 if the input value is zero. The encryption and decryption process for DEA requires eight rounds with the same structure and one last final output transformation. Encryption and decryption both runs on the same process, but have different sub-keys.

A single round needs four modulo multipliers, out of which three are on the critical path. By Reducing the complexity of modulo multiplier circuit significantly improves the performance of the complete DEA chip when one go for implementing DEA in hardware.

DEA works on 64-bit block and use a 128-bit key, and also consists of a series of eight identical transformations also called *round*, final produce a output with final transformation called *half-round*. The method for encryption and decryption are same. DEA uses much of its security by interleaving operations of different group which includes modular addition and modular multiplication, and also bitwise exclusive OR (XOR. Figure 1 shown below shows the method which is been used in proposed research work for encryption, as can seen it has 4 inputs of 16 bit, 4 output of 16 bit and also 6 keys are of 16 bit, this

K1,K2.....K6 is been derive from original 128 bit key.

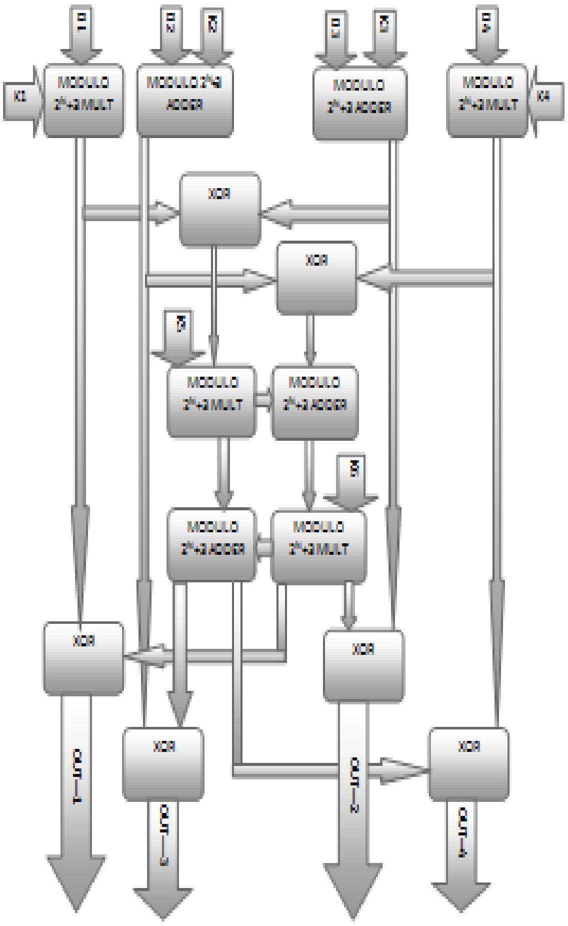

Figure 1: DEA Encryption with proposed modulo multiplier

II. PROPOSED MODULO MULTIPLIER

The diminished-1 representation of numbers was proposed by Leibowitz[3], as a convenient and efficient form for modulo 2^n+3operations on binary numbers. If we let *dim*(A) be the diminished-1 representation of *A*, then

$dim(A) = (A – 1)Mod\ (2^n+3)$............. (1)

The advantage of this representation is that zero is uniquely identified by MSB=1, for which case all arithmetic operations are inhibited.

We obtain the following relationships[3] for arithmetic operations within the representation:

$\dim(A+B) = dim(A)Å dim(B)$

$= \dim(A) + dim(B) + 1\ Mod\ (2^n+3)$....... (2)

$dim(A\text{-}B) = dim(A)Å[-dim(B)]$

$= \dim(A)\ \overline{\dim(B)}\ +1\ Mod\ \overline{\dim(B)}\ (2^n+3)$....... (3)

$\dim(A_1) \oplus ...... \oplus dim(A_n) = dim(A_1) + ... dim(A_n)+(n\text{-}1)Mod(2^n+3)$ ....(4)

Where ⊕ represents addition in the diminished -1 representation.

Since $2^n \approx (-1)Mod(2^n+3)$ the residue operation is accomplished by

$A\ Mod\ (2^n + 3) = A\ Mod\ 2^n\text{-}\ A\ div\ 2^n$ ……. (5)

Our algorithm assumes that neither the multiplier or multiplicand is zero; i.e. zero detection has been completed prior to using this algorithm.

$\dim(2A) = dim(A + A) =$

$2\dim(A) +1\ Mod\ (2^n+ 3)$................(6)

Since 2^kA can be treated as adding *A* to itself 2^k-1times, then

$Dim(2^k A) = d(A + A + ... + A) =$

$2k\dim(A)\ 2^k\text{-}\ 1\ Mod\ (2^n + 3)$............(7)

and eqn. (7) can be rearranged as

$2^k\dim(A) = \dim\ (2^k A) – 2^k+1\ Mod\ (2+ 3)$....(8)

Representing the *n* bits of the diminished-1 form as $dim(A)=\{a_{n-1}a_{n-2}......a_{n-k}\}$ then from eqn. (7), we will have

$\dim\ 2^k(\ A) = \{a_{n-(k+1)}a_{n-(k+2)......}a_0\} + \{1.....1\} – \{a_{n-1}a_{n-2..........}a_{n-k}\}$..............(9)

where we use k least significant zeros in the first number and k ones in the second number on the right hand side of eqn. (9). The first number is the binary representation of 2^k*dim* (*A*) Mod (2^n), and the third number is 2^k*dim*(*A*) div(2^n) The combination of the

last two numbers yields{} $\overline{a_{n-1}}\ \overline{a_{n-2}}\,\overline{a_{n-3}}$ the on $\overline{a_{n-1}}\ \overline{a_{n-2}}\,\overline{a_{n-3}}$ e's compliment of $\{a_{n-1}\, a_{n-2}\, a_{n-3}\}$ hence

dim(2^k *A*) =

$\{an\ -\ {}_4a_n\ -\ {}_5.....a_0\}$ $\overline{a_{n-1}}\ \overline{a_{n-2}}\,\overline{a_{n-3}}$ .... $\overline{a_{n-1}}\ \overline{a_{n-2}}\,\overline{a_{n-3}}$ (10)

or in other words, multiplication by 2^k is accomplished by a cyclic shift of *k* bits to the left with the shifted bits being complemented. This rule was previously derived in[3] by induction. The above argument provides a direct proof of this rule.

Similar to the derivation of eqn. (7), the general multiplication of *A*, a non-zero number, by *B*, a non-zero number, can be treated as adding *A* to itself *B*-1 times. Therefore, The efficient implementation of multi-operand diminished-1 addition is of great importance to the speed of the modulo (2^n+3) multiplier.

In the published literature, the addition of diminished-1 numbers is always implemented by self-contained diminished-1 adders, which require an *n*-bit binary addition plus a carry correction. Since a carry correction is equivalent to an *n*-bit half adder, which has a critical path delay similar to an *n*-bit binary full adder, the self-contained *n*-bit diminished-1 adder is considerably slower than an *n*-bit binary adder. If we look at the multiplication as a multi-operand adder tree, however, then we can invoke transformations which effectively remove the extra delay associated with the diminished-1 adder structure. To start with let us use an example to demonstrate the existing techniques of modulo (2^n+3) multiplication.

III. Observed Results

The table shown below shows the result of proposed design. The logical propagation delay time observed is 25.531 ns and so on behalf of that one could say proposed work max speed for the design is 39.71 Mhz and Area utilisation is 405 slices only. Proposed work results is 17% more area efficient and around 190% more faster as compare to existing Modulo multiplier design of ref[1].

Table 1: Results of Proposed Design

Platform Used: **Vertex4 XC4VLX80-12FF1148**			
Tool Used: **Xilinx 12.2**			
	Used	**Avail-able**	**Utili-zation**
Number of Slices	**405**	**35840**	**1%**
Number of 4 input LUTs	**707**	**71680**	**0%**
Number of bonded IOBs	**48**	**368**	**6%**

IV. Conclusion

This work presents an efficient and optimised modulo multiplier structure with some improvements in the partial product matrix using diminished-1, modulo adder for modulo used with CSA, and also provides zero-case handler. As comparing this proposed structure with other existing methods shows that this structure not only achieves better time delay performance but also the good production with less area and time delay among the base paper work methods compared.

The simulation after designing DEA chip with the proposed modulo multiplier structure was also tested on Xilinx EDA. Simulation observation indicate that the proposed DEA with one round implemented its eight pipeline stages can achieve frequency of maximum 39.71 MHz, and the no, of slice on platform

Vertex4 FPGA are 405. The clock rate of the simulation module can be further improved in the near future by full custom designing instead of semi custom designing.

REFERENCES:

[1] M. Bahrami and B. Sadeghiyan, "Efficient modulo 2n + 1 multiplication schemes for DEA," in Proceedings of IEEE International Symposiums on Circuits and Systems, 2000, pp. 653-656.

[2] A. Curiger, H. Bonnenberg, R. Zimmermann, N. Felber, H. Kaeslin, and W. Fichtner, "VINCI: VLSI implementation of the new secret-key block cipher DEA," in Proceedings of IEEE Custom Integrated Circuits Conference, 1993, pp. 15.5.1-15.5.4.

[3] L. M. Leibowitz: "A simplified binary arithmetic for the Fermat number transform" IEEE Trans. Acoust. Speech, Signal Processing. vol. ASSP-24, pp. 356-359, 1976.

[4] A. V. Curiger, H. Bonnenberg, and H. Kaeslin, "Regular VLSI architecture for multiplication modulo (2n + 1)," IEEE Journal of Solid-State Circuits, Vol. 26, 1991, pp. 990-994.

[5] D. D. Gajski, Principles of Digital Design, Prentice Hall, 1997.

[6] F. A. Jullien, "Implementation of multiplication, modulo a prime number, with applications to number theoretic transforms," IEEE Transactions on Computers, Vol. C-29, 1980, pp. 899-905.

[7] Y. Ma, "A simplified architecture for modulo 2n + 1 multiplication," IEEE Transactions on computers, Vol. 47, 1998, pp. 333-337.

[8] P. E. Madrid, B. Millar, and E. E. Swartzlander Jr., "Modified booth algorithm for high radix multiplication," in Proceedings of IEEE Computer Design: VLSI in Computer and Processors, 1992, pp. 118-121.

[9] L. Sousa, "A universal architecture for designing efficient modulo 2n + 1 multipliers," IEEE Transactions on Circuits and Systems, Vol. 52, 2005, pp. 1166-1178.

[10] R. Zimmermann, "Efficient VLSI implementation of modulo (2n ± 1) addition and multiplication," in Proceedings of the 14th IEEE Symposium on Computer Architecture, 1999, pp. 158-167.

[11] H. Ahmadifar and G. Jaberipur, Improved modulo-(2*n*±3) multipliers, School of Computer Sciences, Institute for research in Fundamental Science (IPM), P.O. Box: 19395-5746, Tehran, Iran, -1-4799-0565-2/13, 2013 IEEE.

* * * * *