ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

Implementation of Low-Complexity Multiplier using distributed arithmetic algorithm

J.Bamela Mary[1], K.Ramamoorthy[2]

PG Scholar (M.E., VLSI Design), PSNA college of engineering and technology, Dindigul, Tamilnadu, India[1]

Associate Professor, Dept. of ECE, PSNA college of engineering and technology, Dindigul, Tamilnadu, India[2]

***Abstract*—. For efficient hardware implementation many designers designs several multiplier structure based on different techniques. But these designs are achieving only 30% of power reduction and 28% of area reduction. In this paper we propose a low complexity and low latency multiplier in order to reduce the requirement of power and area. The proposed work is fully based on the distributed arithmetic algorithm (DAA) which provides the better performance than the existing designs. The proposed design will be coded in verilogHDL and synthesized in Xilinx ISE9.2i. From the synthesized result we will prove the modified structure that requires less area and less power than the existing ones. Finally the proposed design will be implemented on FPGA spartan3E hardware.**

Index Terms—All-one polynomial, finite field, systolic design.

I. INTRODUCTION

Finite field multipliers over GF(2^m) have wide applications in elliptic curve cryptography (ECC) and error control coding systems . Polynomial basis multipliers are popularly used because they are relatively simple to design, and offer scalability for the fields of higher orders. Efficient hardware design for polynomial-based multiplication is therefore important for real-time application.All-one polynomial (AOP) is one of the classes of polynomials considered suitable to be used as irreducible polynomial for efficient implementation of finite field multiplication. Multipliers for the AOP-based binary fields are simple and regular, and therefore, a number of works have been explored on its efficient realization. Irreducible AOPs are not abundant. They are very often not preferred in cryptosystems for security reasons, and one has to make careful choice of the field order to use irreducible AOPs for cryptographic applications. The AOP-based multipliers can be used for the nearly AOP (NAOP) which could be used for efficient realization of ECC systems. AOP-based fields could also be used for efficient implementation of Reed-Solomon encoders.Besides, the AOP-based architectures can be used as a kernel circuit for field exponentiation, inversion, and division architectures. Systolic design is a preferred type of specialized hardware solution due to its high-level of pipeline ability, local connectivity and many other advantageous features. In a bit-parallel AOP-based systolic multiplier has been suggested by Lee *et al*. In a recent paper a low-complexity bit-parallel systolic Montgomery multiplier has been suggested. Very recently an efficient digit-serial systolic Montgomery multiplier for AOP-based binary extension field is presented. The systolic structures for field multiplication have two major issues.

First, the registers in the systolic structures usually consume large area and power. Second, the systolic structures usually have a latency of nearly n cycles, which is very often undesired for real-time applications. Therefore, in this paper, we have presented a novel register- sharing technique to reduce the register requirement in the systolic structure. The proposed algorithm not only facilitates sharing of registers by the neighboring PEs to reduce the register complexity but also helps reducing the latency. Cut-set retiming allows to introduce certain number of delays on all the edges in one direction of any cut-set of a signal flow-graph (SFG) by removing equal number of delays on all the edges in the reverse direction of the same cut-set . When all the edges are in a single direction, one can introduce any desired number of delays on all the edges of any cut-set of an SFG. Therefore, this technique is highly useful for pipelining digital circuits to reduce the critical path. In this paper, we have proposed a novel cut-set retiming approach to reduce the clock-period. The proposed structure is found to involve significantly less area-time-power complexity compared with the existing designs.

II.RELATED WORK

In fact, real-time signals may also be processed in this manner if the associated block-processing delay is acceptable. Another potentially important application for backward filtering is the implementation of Mallat two-channel iterated filter banks based on power-complementary Butterworth filters (wavelets). The zero-phase case is often used to implement frequency-selective infinite-impulse response (IIR) filters corresponding to the squared-magnitude of the classical Butterworth, Chebyshev, and elliptic designs. However, other interesting and potentially important applications exist for non causal IIR filters that are not zero-phase. Examples include equalizers for non minimum- phase systems, non causal speech models, half-sample interpolators, and 90-degree phase shifters such as Hilbert transformers and differentiators. On the other hand, many fast algorithms in the context of digital filtering have been obtained based on particular matrix structures. Many approaches to block digital filters (BDFs) design exist. Some approaches compel the BDF to be time-invariant so that conventional filter synthesis techniques can be used. The best known and most widely used approach is Overlap-save. In some other approaches, no such constraint on the BDF is imposed so that the BDF can be time variant.

III.PROPOSED SYSTEM ARCHITECTURE

Let A,B and C are the extended polynomials and these are represented as:

$$A=\sum_{j=0}^{m} a_j.\propto^{i},$$

$$B=\sum_{j=0}^{m} b_j.\propto^{i} ,$$

$$C=\sum_{j=0}^{m} c_j.\propto^{i},$$

If is the product of elements and , then we can have C=A.B mod f(α)

$$\sum_{i=0}^{m} Xi$$

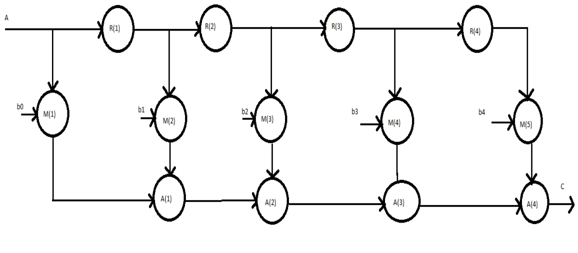
Where Xi is given by

$X_i=b_i.A^i$

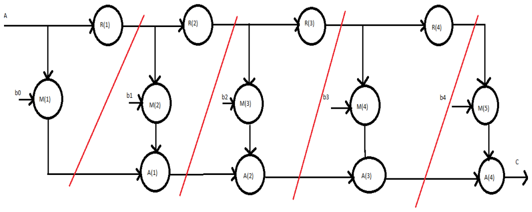
$$A^i=a_{m-i}\alpha^m+a_{m-i-1}\alpha^{m-1}+\ldots a_{m-i-2}\alpha+a_{m-i-1}$$

such that Ai+1 can be obtained from Ai recursively as

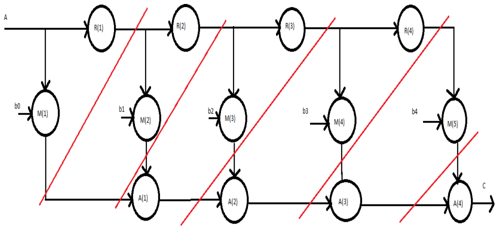
$$A^{i+1}=\alpha.A^{i} \text{ mod } f(\alpha)$$



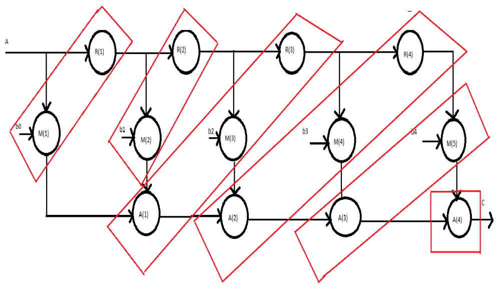
Fig(3.1) General Signal Flow Graph



Fig(3.2) Cutset Retiming Approach I

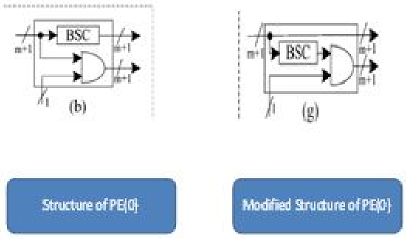


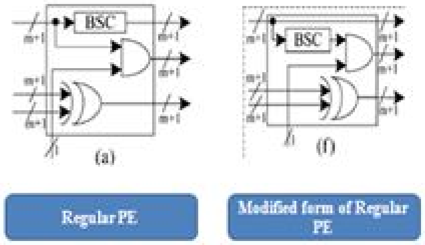
Fig(3.3) Cutset Retiming Approach II



Fig(3.4) Formation of Processing Elements

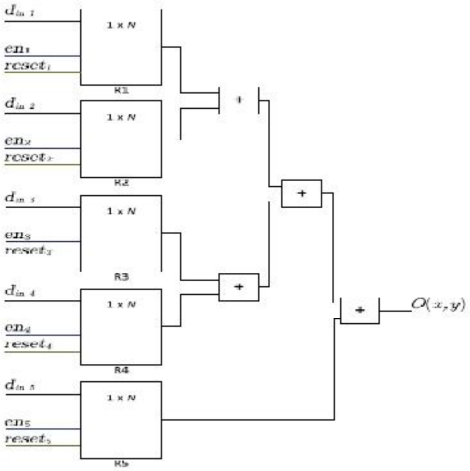
Architecture of PE(0):

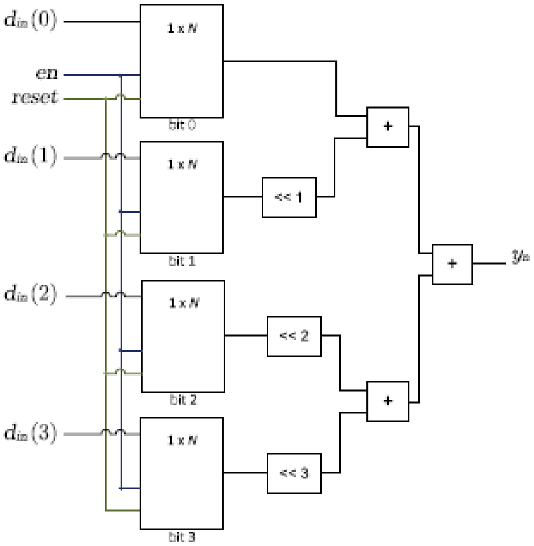




*Architecture for 1 *N convolution and 1-bit input:*

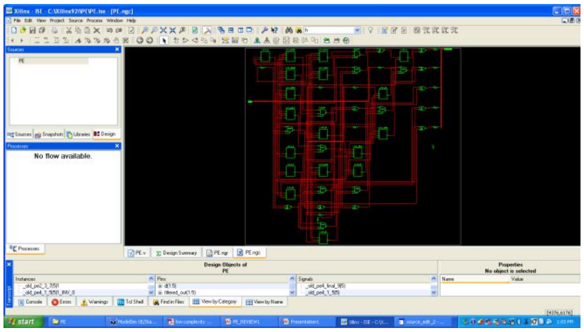
Most 2-D convolution implementations rely on the multiplying units embedded in modern FPGAs to carry out all the multiplications in parallel and to achieve great performance. However, as the kernel size increases, the number of embedded multipliers needed grows exponentially. This fact can constrain the kernel size or force to use a bigger FPGA device, which, in its turn, can yield a very high cost per operation ratio. On the other hand, a lot of work has been done on the design of multiplier less filters, mostly in the one-dimensional domain, and some authors have implemented 2-D convolution by replacing multiplications with shifting and adding operations or transforming the computation into the logarithmic domain.



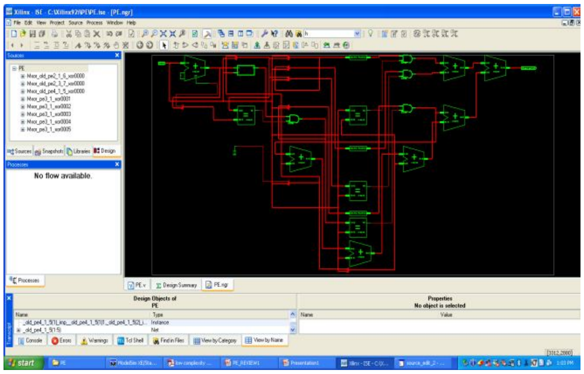


IV.SIMULATION RESULTS

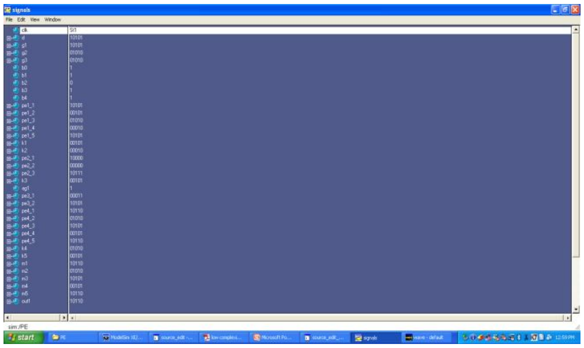
Cross Sectional Top View Of IC



Cross Sectional Front View of IC

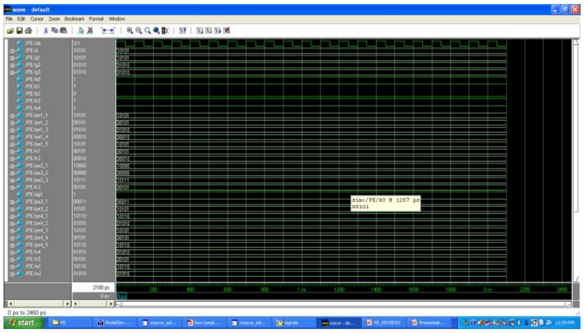


Simulation Screen Shot1

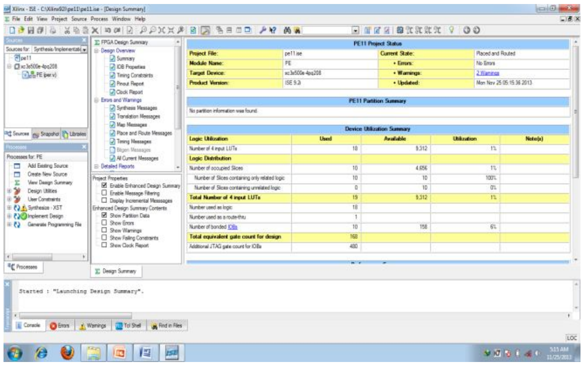


```
signals
File Edit View Window
clk    St1
d      10101
g1     10101
g2     01010
g3     01010
b0     1
b1     1
b2     0
b3     1
b4     1
pe1_1  10101
pe1_2  00101
pe1_3  01010
pe1_4  00010
pe1_5  10101
k1     00101
k2     00010
pe2_1  10000
pe2_2  00000
pe2_3  10111
k3     00101
ag1    1
pe3_1  00011
pe3_2  10101
pe4_1  10110
pe4_2  01010
pe4_3  10101
pe4_4  00101
pe4_5  10110
k4     01010
k5     00101
m1     10110
m2     01010
m3     10101
m4     00101
m5     10110
out1   10110
sim:/PE
```

Simulation Screen Shot2



Gate Count



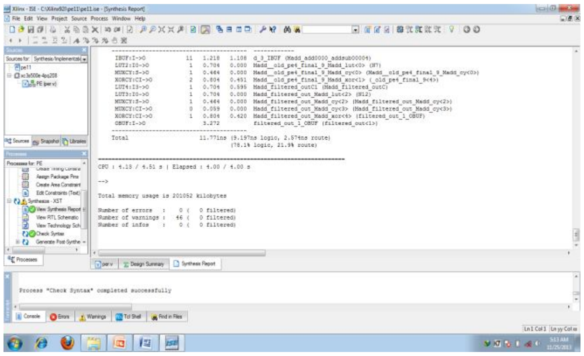
PE11 Project Status			
Project File:	pe11.ise	Current State:	Placed and Routed
Module Name:	PE	• Errors:	No Errors
Target Device:	xc3s500e-4pq208	• Warnings:	2 Warnings
Product Version:	ISE 9.2i	• Updated:	Mon Nov 25 05:15:36 2013

PE11 Partition Summary

No partition information was found.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of 4 input LUTs	18	9,312	1%	
Logic Distribution				
Number of occupied Slices	10	4,656	1%	
Number of Slices containing only related logic	10	10	100%	
Number of Slices containing unrelated logic	0	10	0%	
Total Number of 4 input LUTs	19	9,312	1%	
Number used as logic	18			
Number used as a route-thru	1			
Number of bonded IOBs	10	158	6%	
Total equivalent gate count for design	168			
Additional JTAG gate count for IOBs	480			

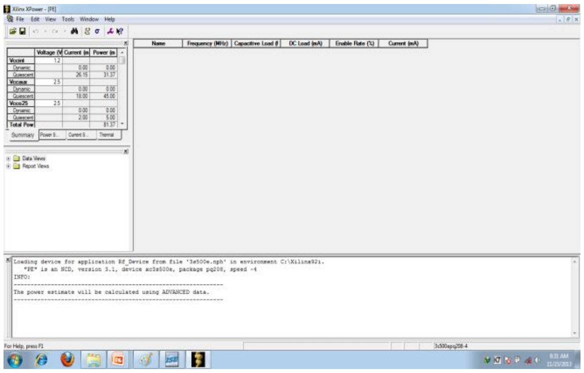
Latency Analysis



```
    ----------------------------------------  ------------
     IBUF:I->O            11   1.218   1.108  d_3_IBUF (Madd_add0000_addsub00004)
     LUT2:I0->O            1   0.704   0.000  Madd__old_pe4_final_9_Madd_lut<0> (N7)
     MUXCY:S->O            1   0.464   0.000  Madd__old_pe4_final_9_Madd_cy<0> (Madd__old_pe4_final_9_Madd_cy<0>)
     XORCY:CI->O           2   0.804   0.451  Madd__old_pe4_final_9_Madd_xor<1> (_old_pe4_final_9<4>)
     LUT4:I3->O            1   0.704   0.595  Madd_filtered_outC1 (Madd_filtered_outC)
     LUT3:I0->O            1   0.704   0.000  Madd_filtered_out_Madd_lut<2> (N12)
     MUXCY:S->O            1   0.464   0.000  Madd_filtered_out_Madd_cy<2> (Madd_filtered_out_Madd_cy<2>)
     MUXCY:CI->O           0   0.059   0.000  Madd_filtered_out_Madd_cy<3> (Madd_filtered_out_Madd_cy<3>)
     XORCY:CI->O           1   0.804   0.420  Madd_filtered_out_Madd_xor<4> (filtered_out_1_OBUF)
     OBUF:I->O                 3.272          filtered_out_1_OBUF (filtered_out<1>)
    ----------------------------------------
    Total                     11.771ns (9.197ns logic, 2.574ns route)
                                       (78.1% logic, 21.9% route)

=========================================================================
CPU : 4.18 / 4.51 s | Elapsed : 4.00 / 4.00 s

-->

Total memory usage is 201052 kilobytes

Number of errors   :    0 (   0 filtered)
Number of warnings :   46 (   0 filtered)
Number of infos    :    0 (   0 filtered)
```

```
Process "Check Syntax" completed successfully
```

Power Analysis



	Voltage (V	Current (m	Power (m
Vccint	1.2		
Dynamic		0.00	0.00
Quiescent		26.15	31.37
Vccaux	2.5		
Dynamic		0.00	0.00
Quiescent		18.00	45.00
Vcco25	2.5		
Dynamic		0.00	0.00
Quiescent		2.00	5.00
Total Pow			81.37

```
Loading device for application Rf_Device from file '3s500e.nph' in environment C:\Xilinx92i.
   "PE" is an NCD, version 3.1, device xc3s500e, package pq208, speed -4
INFO:
--------------------------------------------------------------
The power estimate will be calculated using ADVANCED data.
--------------------------------------------------------------
```

TABLE1: Performance Evaluation

Performance Evaluation Parameter	Estimated Values
Power Consumption	81mW
Latency	11.771ns
Gate Count	650

V.CONCLUSION

Efficient systolic design for the multiplication over GF(2^m) based on irreducible AOP is obtained in my existing system. By using cut-set retiming technique the critical path is reduced to one XOR gate delay and by sharing of registers for the input-operands in the PEs, the low-latency bit-parallel systolic multiplier have been derived. For self checking I have simulated in modelsim. For evaluating the performance parameter I have used Xilinx ISE 9.2i. In my existing system the total power consumption is 81mW, latency is 11.771ns and required number of gates are around 650. Moreover the existing design will be reconstructed for reduce the latency, power requirement and gate count by using distributed arithmetic algorithm.

REFERENCES

[1] M. Ciet, J. J. Quisquater, and F. Sica, "A secure family of composite finite fields suitable for fast implementation of elliptic curve cryptography," in *Proc. Int. Conf. Cryptol. India*, 2001, pp. 108–116.

[2] H. Fan and M. A. Hasan, "Relationship between Montgomery and shifted polynomial basis multiplication algorithms," *IEEE Trans. Computers*, vol. 55, no. 9, pp. 1202–1206, Sep. 2006.

[3] C.-L.Wang and J-L. Lin, "Systolic array implementation of multipliers for finite fields " *IEEE Trans. Circuits Syst.*, vol. 38, no. 7, pp. 796–800, Jul. 1991.

[4] B. Sunar and C. K. Koc, "Mastrovito multiplier for all trinomials," *IEEE Trans. Comput.*, vol. 48, no. 5, pp. 522–527, May 1999.

[5] C. H. Kim, C.-P. Hong, and S. Kwon, "A digit-serial multiplier for finite field *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 4, pp. 476–483, 2005.

[6] C. Paar, "Low complexity parallel multipliers for Galois fields based on special types of primitive polynomials," in *Proc. IEEE Int. Symp. Inform. Theory*, 1994, p. 98.

[7] H. Wu, "Bit-parallel polynomial basis multiplier for new classes of finite fields," *IEEE Trans. Computers*, vol. 57, no. 8, pp. 1023–1031, Aug. 2008.

[8] S. Fenn, M.G. Parker,M. Benaissa, and D. Taylor, "Bit-serial multiplication in using all-one polynomials," *IEE Proc. Com. Digit. Tech.*, vol. 144, no. 6, pp. 391–393, 1997.

[9] K.-Y. Chang, D. Hong, and H.-S. Cho, "Low complexity bit-parallel multiplier for ____ _ defined by all-one polynomials using redundant representation," *IEEE Trans. Computers*, vol. 54, no. 12, pp. 1628–1629, Dec. 2005.

[10] H.-S. Kim and S.-W. Lee, "LFSR multipliers over defined by all-one polynomial," *Integr., VLSI J.*, vol. 40, no. 4, pp. 571–578,2007.

[11] P. K. Meher, Y. Ha, and C.-Y. Lee, "An optimized design of serial-par- allel finite field multiplier for based on all-one polynomials,"in *Proc. ASP-DAC*, 2009, pp. 210–215.

[12] M. Sandoval, M. F. Uribe, and C. Kitsos, "Bit-serial and digit-serial montgomery multipliers using linear feedback shift regis- ters," *IET Comput. Digit. Tech.*, vol. 5, no. 2, pp. 86–94, 2011.

[13] C.-Y. Lee, E.-H. Lu, and J.-Y. Lee, "Bit-parallel systolic multipliers for ields defined by all-one and equally spaced polynomials,"*IEEE Trans. Computers*, vol. 50, no. 6, pp. 385–393, May 2001.

[14] Y.-R. Ting, E.-H. Lu, and Y.-C. Lu, "Ringed bit-parallel systolic mul- tipliers over a class of fields ," *Integr., VLSI J.*, vol. 38, no. 4,pp. 571–578, 2005.

[15] C.-Y. Lee, J.-S. Horng, I.-C. Jou, and E.-H. Lu, "Low-complexity bit-parallel systolic montgomery multipliers for special classes of," *IEEE Trans. Computers*, vol. 54, no. 9, pp. 1061–1070,.