Novel Pipelined Scalable Systolic Multiplier Based on Irreducible All-One Polynomials

S. Srinivas, E. John Alex, Prasad Janga

Abstract—* A planned productive structure issued for the systolic execution of authoritative based limited field duplication over G F(2 m) in light of final 56-bit AOP is proposed in this work. We extricated a recursive increase calculation and utilized it to plan an intermittent and confined piece level reliance outline (DG) for systolic registering. The intermittent piece level DG is changed into a very smal grained DG, and the pipe coating is utilized for snappier mapping into a parallel systolic design. It doesn't require any overall correspondences for measured decline, in contrast to most current developments. The suggested bit-parallel systolic structure is similar to the parallel systolic structure, however the quantity of registers is altogether lower.

Keywords: Pipelining, Elliptic curve cryptography (ECC), error-control-coding, very large scale integration (VLSI).

I. INTRODUCTION

Limited field number juggling has as of late found broad applications. Models incorporate cryptography, hypothesis of coding, and math of PCs. Augmentation is the most confused and tedious of the limited field number-crunching tasks. It is as of now utilized in a huge number of utilizations, for example, the Reed-Solomon coder's VLSI design[4],[5]. It is therefore urgently necessary to have a tiny, elite limited field multiplier. Such a multiplier can likewise be utilized as a development square to structure numerous enormous frameworks using arithmetic of finite fields. Bit-Parallel multiplications is used in coding and cryptography for error-control over finite fields and classified in to polynomial multipliers, normal baseline multipliers, and dual baseline multipliers, depending on the base used. Since, for example, the size intricacy of an average piece many researchers have considered special classes of finite fields to reduce the complexity of the size [10]. The intricacy of parallel MOM is essentially decreased when worked by ordinary base unchangeable AOP with 2m2-2 m XOR1 and 2m2-m AND gates [9]. By representing field components on a created a framework based on AOPs and ESPs for in GF(2m)[6]. Their design is modular and less complex than the MOM. Using ordinary foundation and polynomial foundation, Two forms of bit parallel multipliers are exhibited such as HWBM Massey-Omura (M MOM) multipliers [4].

Revised Manuscript Received on September 10, 2019.

Mr. S.Srinivas, PG Scholar, Department of ECE, CMR Institute of Technology, Hyderabad, Telangana, India.
(Email: sreenivas.sandela@gmail.com)
Dr. E.John Alex, Professor, Department of ECE, CMR Institute of Technology, Hyderabad, Telanagana, India.
(Email: johnalexvlsi@gmail.com)
Dr.Prasad.Janga, Professor, Department of ECE, CMR Institute of Technology, Hyderabad, Telanagana, India.
(Email: prasadjanga85@gmail.com)

Compared to MOM, the size and time complexities were considerably reduced[5]. In cryptography and coding theory, significant GF(2 m) such as short key exchange. Advanced standard of encryption (AES)[3] and codes of Reed Solomon[4]. GF(2 m) II's arithmetic operations incorporate expansion, increase, exponentiation, division, and multiplicative reversal. Among all the abovementioned, the plan of the multiplier with the unchangeable each of the one expansion and duplication are the fundamental polynomial segments proposed by[9],[10] can diminish the unpredictability of GF(2 m) to perform the others. Effectively implementing room and time.

In latest years, due to their implementation in error-control coding, finite fields have got a lot of attention[1],[2]. They were also used in the processing of digital signals, the generation of pseudo random numbers, and a strong decoding algorithm are significant considerations for high-speed design and low-complexity decoders formany error-control code. Therefore, the multipliers of the matrix basis are used widely. For multiplication of the areas different algorithms and architectures are suggested produced by trinomials and pentanomials basics of polynomial [8] -[13], mainly because of easier computation.

All one polynomials (AOPs) structure a one of a kind class that can be utilized in contrast with trinomials and penanomial-based multipliers for simpler and more effective application. As a result, the AOP-based element representation anticipated future application in effective equipment usage of elliptic bend crypto frameworks. Finding the AOP bases for limited fields is likewise not troublesome. It is realized that there are 108 possible AOP bases for m < 2000, [2]. The literature [15]–[28], [30]–[32] also suggested efficient field multiplication architectures and form power-sum calculation (A+B2) for field produced by AOPs.

The first G F(2m) multiplier produced by AOP was suggested by Ltoh and Tsujii[15]. Bit-parallel structures used in low-inactivity execution, however because of their wide basic computing time.

Systolic designs constitute architectural paradigm to execute of VLSI and FPGA owing structural modularity, together with important potential for high-throughput through pipe lining, parallel processing, or both[33]. There are a few piece sequential structures dependent on unchangeable AOP for standard multi-plier premise over G

F(2 m). For all-one polynomials and trinomials, the writers provided a conversion technique to introduce low-complexity Montgomery multipliers. A novel cut-set retiming on the basis of irreducible AOP[1] is recently suggested for an area-time-efficient systolic multiplier framework. Chen et al[26] proposed an AOP augmentation framework by extendeding portrayal of a poly-nomial premise. In this paper, we present another particular decrease calculation and a powerful recursive plan dependent on final AOP for authoritative premise increase over GF(2m). Furthermore, we get from the proposed calculation an intermittent and confined piece level reliance chart (DG) to a contingent upon the constraints and necessities of the applications, the equipment intricacy and the throughput pace of the collapsed systolic exhibit can be scaled.

Pipelining:

Pipe lining is one way to improve a processor's general processing efficiency. This architectural strategy enables several directions to be executed simultaneously. Pipe lining is straightforward to the software engineer; parallelism at the guidance stage is misused by covering the guidance execution process.

It is like a mechanical production system where representatives execute a specific occupation and move the halfway completed thing to the following specialist. It talks about different techniques for estimating their presentation. Guidance pipe covering and number-crunching funnel lining, alongside techniques for augmenting of pipeline is talked about. Booking table and inertness thoughts are talked about in conjunction with a technique of monitoring static and dynamic pipeline planning.

Pipeline Structure

Design of pipeline breaks down in to linear method of different phases. A phase conducts a specific work and creates a middle of the road result. It contains an info hook, otherwise called a register and a processing circuit follows. A specified stage's processing circuit is linked to the next stage's input latch (see Figure 3.1). Each input latch is linked to a clock signal through whole pipeline, the final outcome is generated, finishing one phase per clock pulse. The clock beat period ought to be adequately enormous to give adequate time to a sign to navigate the slowest arrange, called the bottleneck (for example the phase that requires the longest measure of time to finish). Moreover, a lock ought to have adequate time to store its info signals. On the off chance that the time of the clock, P, is communicated as P = tb + tl, at that point tb ought to be more noteworthy than the bottleneck stage's greatest deferral, and tl ought to be adequate to store information in a lock.

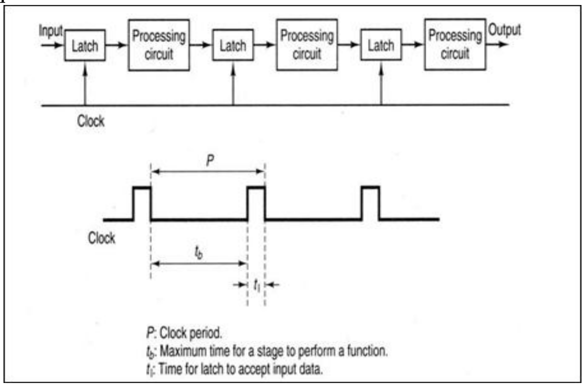


Figure 3.1 Pipeline structure.

Pipeline Performance Measures

The capacity to overlap sequential phases of various input functions results in general theory of completion moment

Tpipe = m*P + (n-1)*P, (3.1)

Where n is task input, m is amount of pipeline phases and P is clock period. The term m*P is the time required to get through first input task, and the term (n-1)*P is time required for rest of the task. In other words, it will only produce production as quickly as its slowest point after the pipeline is loaded. Even with this restriction, the pipeline will significantly outperform non-pipeline methods that require completion of each assignment before the execution sequence of another assignment starts. To be more precise, a pipeline processor can generate performance about m times quicker than non-pipeline processor (n is big). On the other side, the above sequential method needs in a non-pipeline The processor is equivalent to the amount of pipeline phases to finish m velocity.

T seq n   τi

i1

Tseq can be rewritten as

Tseq = n*m* .

By ignoring small time needed for latch storage (i.e., t1 = 0),Tseq = n * m * P.

Now, speed up (S) may be represented as:

S = Tseq / Tpipe = n*m / (m+n -1).

S approaches m when n needs to be omitted. In other words, the ideal is that a pipe-lined processor generates output about m times speeder with respect to non-pipe-lined processor for very n value. The speed-up reduces when n is low ; in reality, the pipeline has the minimum speed of 1. Two other variables, besides speedup, are often used to determine a pipeline's performance ; are efficiency and throughput. The effectiveness E is described as:

E = S/m = [n*m / (m+n -1)] / m = n / (m+n -1).

A pipeline's throughput H, known as bandwidth, is described by amount of assigned input per unit of moment it can process. H is described as if the pipeline has m phases

$H = n / T_{pipe} = n / [m*P + (n-1)*P] = E / P = S / (mP)$.When n , Maximum estimation of undertaking for every clock cycle. Quantity of steps in pipeline frequently relies upon the presentation cost trade off. By achieving performance cost ratio, the ideal selection for such a amount can be determined. PCR has been described by Larson as:

PCR maximum throughput

pipeline cost

To demonstrate, suppose that in order to process an input assignment, a non-pipe lined processor needs a tseq completion time. A clock period of P = (tseq / m) + tl is required for a pipeline m process assignment. Also called the pipeline frequency is the highest throughput 1/P. Based up on the speed of successive assignments entering the pipeline, the real throughput may be less than 1/P. The price of the pipeline cp expressed by complete price of logic gates and latches in phases. The PCR equation is given as PCR = 1/{[(tseq / m) + tl](cg+ mcl)}.This value can be used as an ideal selection for the amount of phases since the value m0 maximizes PCR.

Types of Pipeline

Instruction and arithmetic are two divisions of pipeline and by construction it is static or dynamic. One procedure (such as addition or multiplication) can be performed at a moment by a static pipeline. Only after draining the pipeline can the operation of a static pipeline be altered. Consider, for instance, a static pipeline capable of adding and multiplying.

The pipeline must be drained and set for the fresh procedure every time the pipeline moves from multiplication to addition operation. When activities shift frequently, the efficiency of static pipelines is significantly degraded, more than one operation at a moment can be carried out by a dynamic pipeline. The information must go through a certain series of phases in order to execute a specific procedure on an input information. For instance, Figure 3.2 displays a vibrant three-stage pipeline multiplying distinct information. The input information must go through phases 1, 2, and 3 for multiplication and 1 and 3 for addition. First phase of the addition conducted by input information D1 at phase 1, and final phase of the multiplication process is conducted on a distinct input information D2 at phase 3 at the same moment. Initiation time input in D1 and D2 pipeline should be such that they do not simultaneously reach phase 3; otherwise, a collision occurs. Regultion of mechanism is more complicated in dynamic pipeline than in static pipelines.

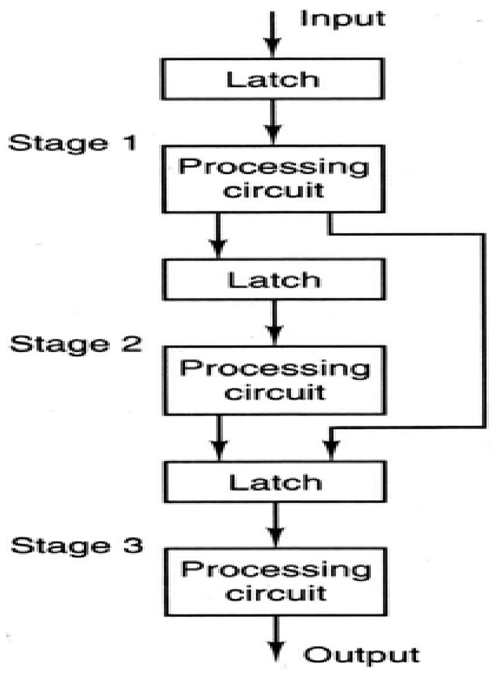


Figure 3.2 Dynamic pipeline (Three stage)

II. MATHEMATICAL FORMULATION

Let G F(2 m) provided by

$Q(z)= z^m+ q_{m-1}.z^{m-1}+. . .+ q_2.z^2+ q_1.z +1$ (1)

Where { qi is equivalent to 1 −1}G F(2). The basis of the polynomial { 1, α, α2,. To depict the field components, use αm−1 },

$m-1$ $m-1$

$A = \underset{i=0}{} a_i.\alpha^i$ and $B = \underset{i=0}{} b_i.\alpha^i$ (2)

where, a_i and $b_i \in \{0,1\}$, for$i =_m$

$C = A \cdot B \bmod Q(z)$ (3)

Recurrent multiplication relationship, (3) can be extended to a form.

$$\boxed{C = b_0 \cdot A + \underset{i=1}{\overset{m-1}{}} b_i \cdot \alpha^{i-1} \cdot P \bmod Q(z)}$$ (4)

where $P=A\alpha$, is a polynomial of degree m, and given by

$$\boxed{P = {}_m \ p_i \cdot \alpha^i}$$ (5)

$$\boxed{\alpha^m = q_{m-1} \cdot \alpha^{m-1} + \cdot \cdot \cdot + q_2 \cdot \alpha^2 + q_1 \cdot \alpha + 1}$$ (7)

When polynomial $Q(z)$ is an AOP, (7) can be written as From (8) it can be found further that

$\alpha^{m+1}=1.$ (9)

$P\alpha \equiv P_1= p_m+ p_0 \cdot \alpha+ p_1 \cdot \alpha^2+ \cdot \cdot + p_{m-1} \cdot \alpha^m$ (10)

$P_{i+1}=L(P_i)=L^i(P_1),$ $0 \leq i \leq m-2.$ (11)

If L is the cyclic-left-shift operation and the product word in (4) can then be expressed equivalently as

$$\boxed{C = b_0 \cdot A + \underset{i=1}{\overset{m-1}{}} b_i \cdot P_{i-1} \bmod Q(z)}$$ (12)

$C = Y \bmod Q(z)$ (13a)

$$\boxed{Y = \underset{i=0}{\overset{m-1}{}} Y_i}$$ (13b)

And

$Y_i= b_i \cdot P_{i-1}$ for $1 \leq i \leq m-1$ and (13c)

$Y0= b0 \cdot P-1,$ assuming $P-1=(0 \ \&A).$

$$\boxed{Y = y_0+ y_1.\alpha+ y_2.\alpha^2+ \cdot \cdot +y_{m-1} \cdot \alpha^{m-1}+ y_m.\alpha^m}$$ (14)

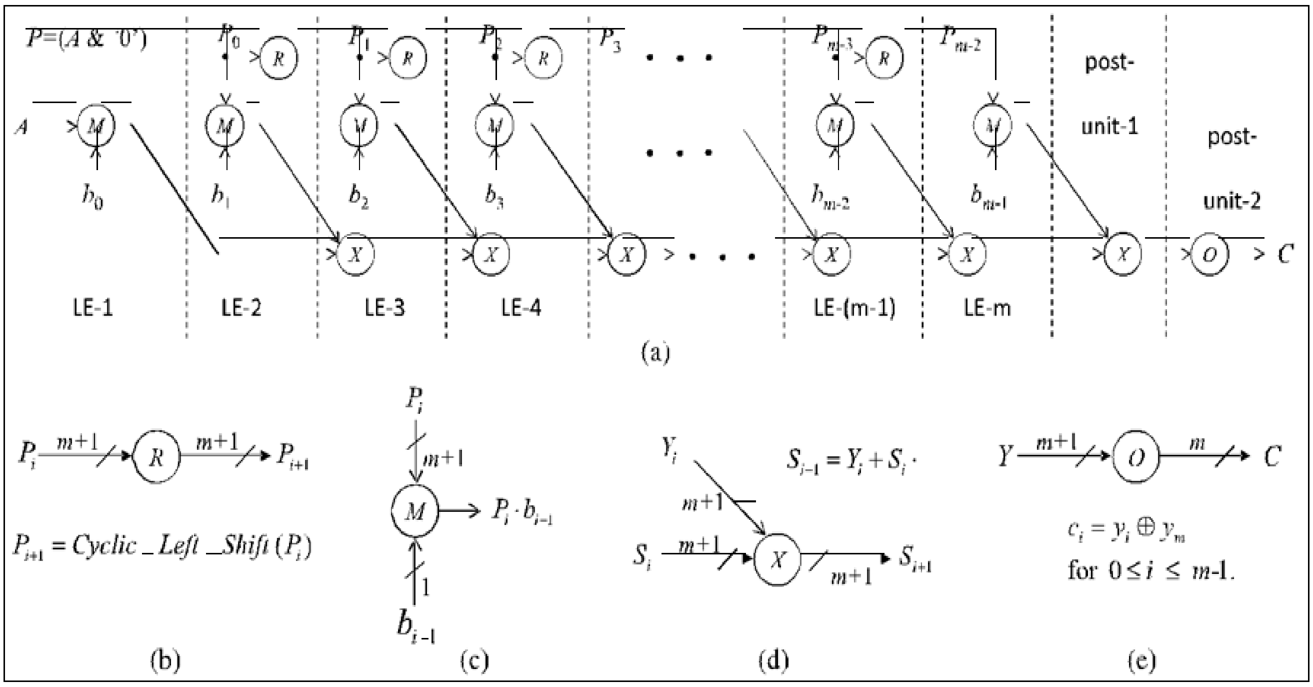


Fig. 1. The reliance graph (DG) of finite field multiplication recursive formulation over G F(2 m) based on irreducible AOP. (A) The Chief Executive.
(B) Functional node R decrease description. (C) Bit-multiplication node M functional description. (D) Addition node functional description X.
(E) Output Reduction Node O functional description.

- STEP-2: For i= 1 tom− 1 • do left-shift work in degree Pi−2α(m + 1) polynomial to decrease its degree by oneto obtaining the degree m Pi−1 oper and.
- Multiply bi bit-level with Pi−1 to get Yiac according to (13c).
- Add Yi to the FFA material to get a partial degree m (13b) outcome.

STEP-3: To achieve the required product value, perform Y modular decrease.

III. DERIVATION OF MULTIPLIERS FOR G F(2 M) BASED ON IRREDUCIBLE MULTIPLIERS

The systolic execution of duplication G F(2 m), second stage tasks directed recursively (11&13), every repeat comprises of three stages, for example secluded decrease duplication exercises (13c or 13d) trailed augmentations of (13b). Such portrayed chart of appeared in Fig. 1, comprising particular decrease hubs option hubs' X' and comparable measure duplication hubs ' M.' An extra piece increase hub is required forSTEP-1 pre-handling and a yield decline hub' O' is required for STEP-3 post-preparing.The decrease node feature is shown in Fig. 1(b). It carries out decrease (11). Bit-multiplication node function, and yield decrease nodes are shown in Figs. 1(a-e). What's more, task of a touch of operand B with a diminished type of operand (αi•P), every one of Addition hubs plays out an expansion activity. 1(a) and a couple of post-processing units in m logic units (LU). Based on this re-timing, mapping this DG to a direct systolic cluster made out of m handling parts (PE) together with two post-preparing cells is straightforward. The subsequent straight systolic cluster (not appeared in measurements) can give exceptionally high-throughput of single word of thing per cycle, where the process duration is just one deferral of XOR. But there are two significant drawbacks to such a linear systolic model.

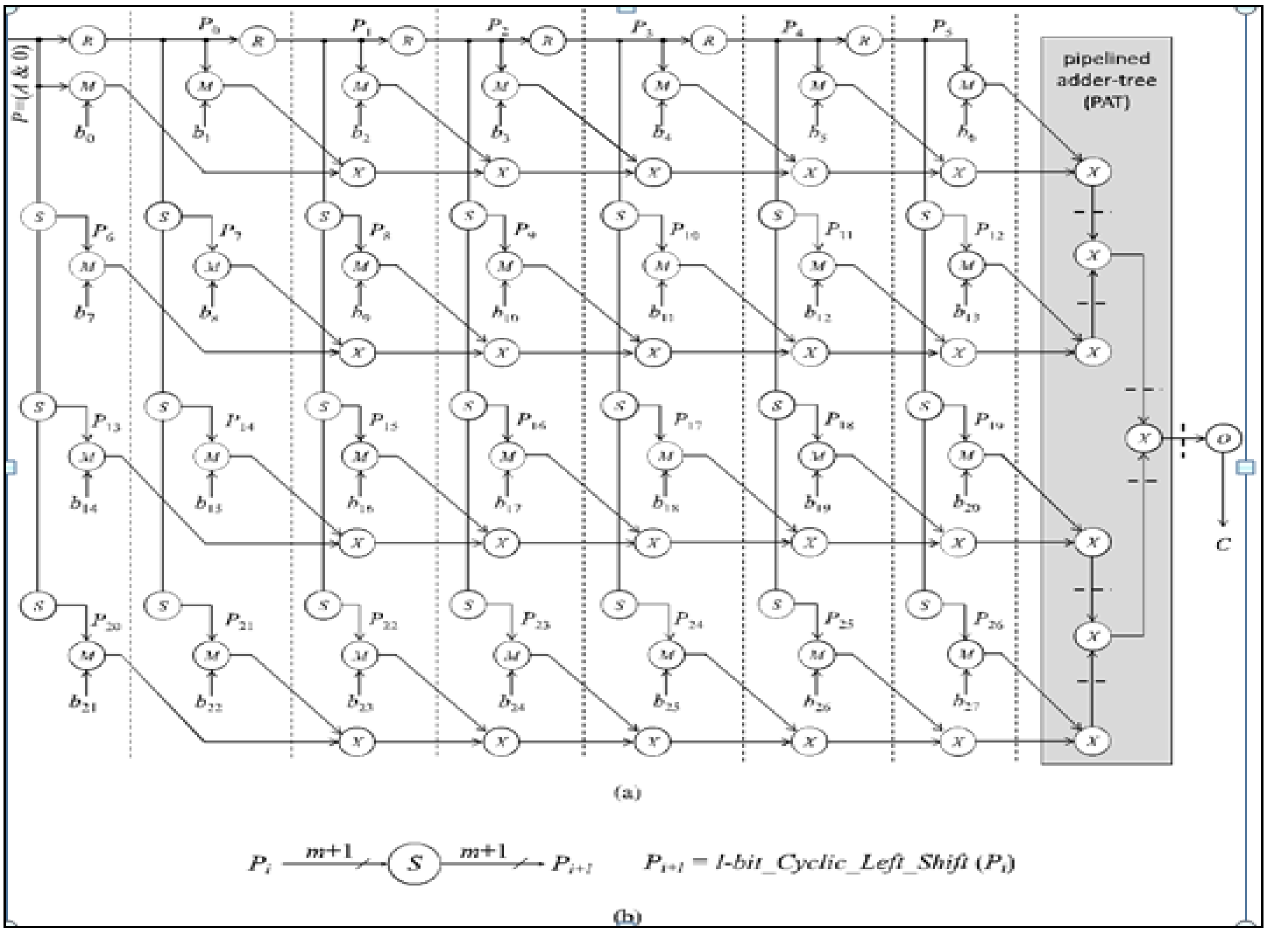
1.First, the structure's latency is almost m (field order).

2. Second, the structure's register complexity is greater than complexity of the combination circuit, as all PE will AND and XOR gate for executing the conbined logic.

We derive here a parallel multiplier structure for G F (2 m) to prevent these issues. In addition, we have suggested a time-multiplexed framework for hardware-efficient realization.

A. Parallel Systolic Structure

As opposed to mapping the DG in Fig. 1 In a one-dimensional direct systolic cluster, recommend apportioning and revamping the LUs with l LUs in each column. The estimation of l is identified with the idleness of the increase and ought to be controlled by the necessity of the application.Therefore, the quantity of columns in the cluster


pipelined adder-tree (PAT)
(a)
$P_i \xrightarrow{m+1} S \xrightarrow{m+1} P_{i+1}$ P_{i+1} = l-bit_Cyclic_Left_Shift (P_i)
(b)

56 Bit Extension Structure

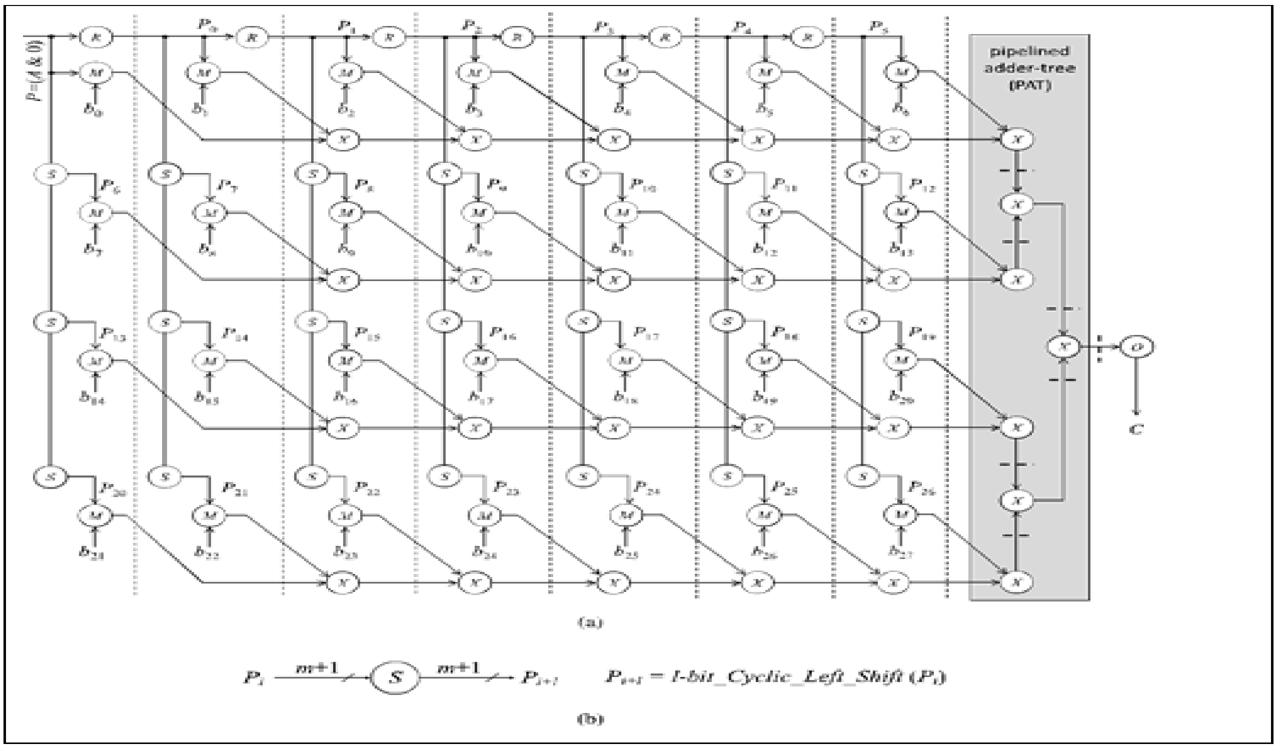
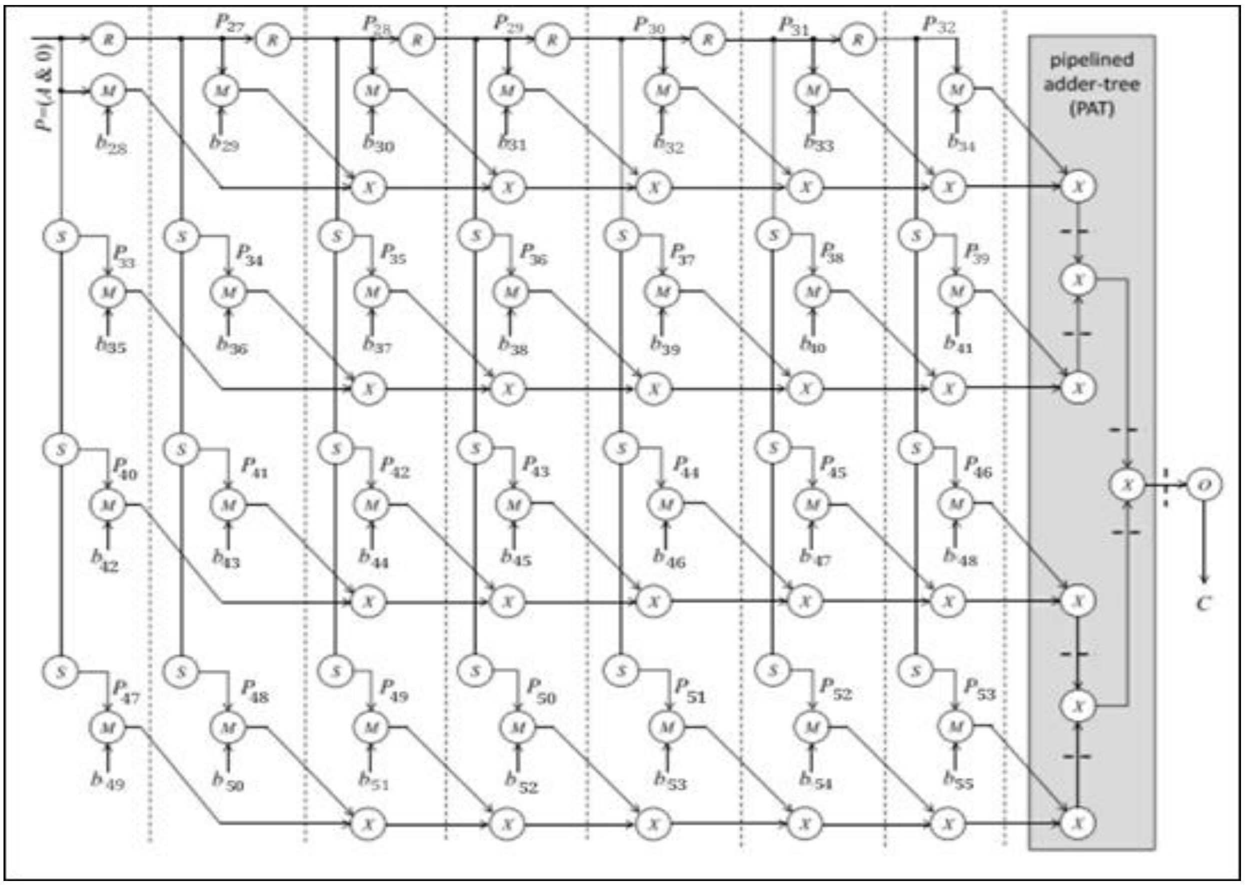

pipelined adder-tree (PAT)
(a)
$P_i \xrightarrow{m+1} S \xrightarrow{m+1} P_{i+1}$ P_{i+1} = l-bit_Cyclic_Left_Shift (P_i)
(b)

Fig. 2. The regularized dependence graph (DG) for the finite field multiplication over *G F*(2^m) for irreducible AOP. (a) The DG. (b) Function description of multi-reduction node *S*



is $s=m/l$ m is given as

$$m = ls - r, \quad (16)$$

If m, not integer multiples of l, last row may have fewer LUs. Integer r falls between [0,l] range. The multiplicands added with r-bit zeros for r > 0 instances.

Figure shows the building of a two-dimensional m=28 DG. 2 For instance, it comprises of four rows, with five periodic and two pre-processing LUs in each row. The cut-sets shown in Fig. 3, where each column's LU features are integrated in the corresponding PE. A pipe lined-viper tree utilized to execute post-preparing rationale pursued a yield decrease cell (ORC).

In each cycle, each PE takes 4 parts of input B. Figs shows the features of distinct kinds of PEs. Three (b), three (c), and three (d).

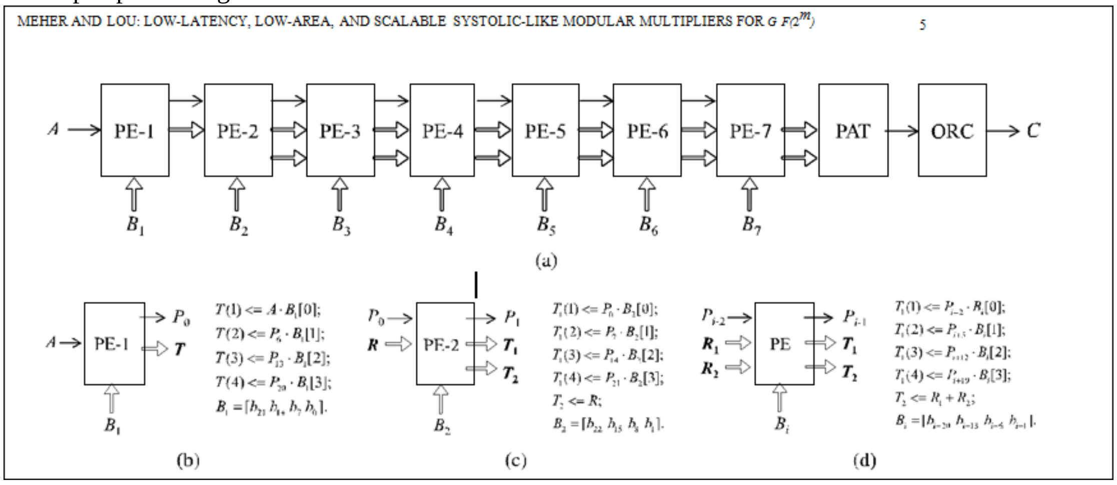

MEHER AND LOU: LOW-LATENCY, LOW-AREA, AND SCALABLE SYSTOLIC-LIKE MODULAR MULTIPLIERS FOR G F(2^m)
5
A
PE-1
PE-2
PE-3
PE-4
PE-5
PE-6
PE-7
PAT
ORC
C
B1 B2 B3 B4 B5 B6 B7
(a)
T(1) <= A · B1[0]; T(2) <= P6 · B1[1]; T(3) <= P13 · B1[2]; T(4) <= P20 · B1[3]; B1 = [b21 b14 b7 b0].
T1(1) <= P0 · B2[0]; T1(2) <= P7 · B2[1]; T1(3) <= P14 · B2[2]; T1(4) <= P21 · B2[3]; T2 <= R; B2 = [b22 b15 b8 b1].
Ti(1) <= Pi-2 · Bi[0]; Ti(2) <= Pi+5 · Bi[1]; Ti(3) <= Pi+12 · Bi[2]; Ti(4) <= Pi+19 · Bi[3]; T2 <= R1 + R2; Bi = [bi+20 bi+13 bi+6 bi-1].
(b)
(c)
(d)

Fig. 3. The proposed parallel systolic-like array for the finite field multiplication over $GF(2^m)$ based on irreducible AOP. (a) The linear array structure. (b) Function of PE-1. (c) Function of PE-2. (d) Function of the *i*th regular PEs (PE-3 to PE-7).

Each of the PE-1and PE-2 comprises of 4 AND cells for the 4 M multiplication nodes feature and normal PEs: PE-3toPE-7, comprises 4 AND cells and XOR cells for conducting node features.

Principle points of interest of the parallel systolic engineering over other systolic models is by picking diverse l and s esteems while keeping up the process duration and throughput, we can have modifiable latencies as far as cycles. When all is said in done, the recommended engineering's inertness in Fig. 2.

$$L = l + \log_2 s + 1, \qquad (17)$$

Where l = PE's latency, log2s is PAT latency and constant 1 is the ORC latency. According to (16), for a specified value of m, there could be a few other l and s alternatives for apportioning the DG.The inertness of the duplication is 9 cycles and process duration and throughput are equivalent to that of Fig. 3.

$$L = \quad + \log_2 \bar{s} + 1. \qquad (18)$$

It produces the required product phrase C according to (15) by reducing the Y from m to (m−1).

B. Structure of Time-Multiplexed Systolic

Calculating the 2-D DG rows of LUs in Fig to have a hardware-efficient execution. The most direct and equipment productive route plan the equipment one column of Fig. 2. Alluded structure, 1 speaks to time-multiplexed equipment of column. Fig viper tree. 2 A finite field accumulator (FFA) could be used. Figure shows the TM-1 multiplier structure for G F(2 m) AOP. 4 For28 m=.

It is made up of seven PEs. Various PEs feature is shown in Figs. 4(b)–(e). PE-1 and PE-2 comprise of just one AND cell each required for every ordinary PEs .All other periodic PEs consist of one decrease cell exceptPE-7 (the last PE). A multiple-reduction cell (MRC) is used to enforce the node S feature. It decreases the info operand by request to produce progressive information.The PE-7 yield provided for FFA aggregator to consecutively execute the pipeline-snake tree include. The FFA register is introduced to zero and four progressive PE-7 yields are included in cycles with the register content (13b).

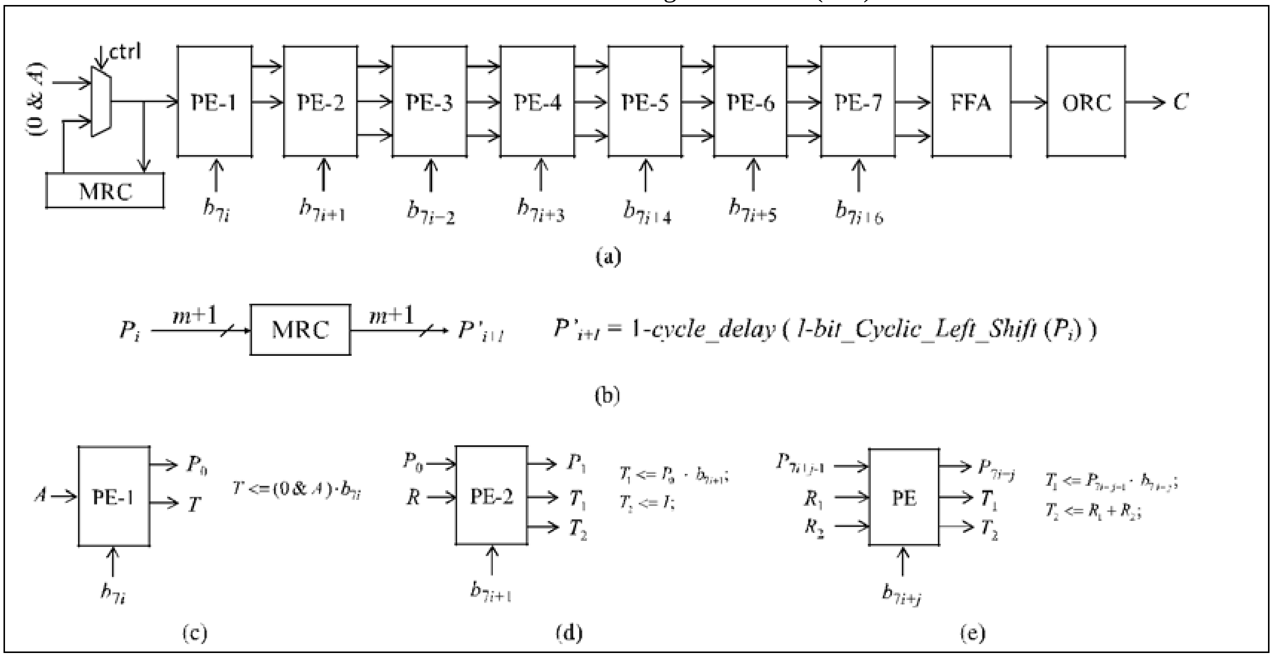


Fig. 4. The proposed time-multiplexed systolic-like array (TM-1) for the finite field multiplication over *G F(2^m^)* based on irreducible AOP. (a) The linear array structure. (b) Function of MRC. (c) Function of PE-1. (d) Function of PE-2. (e) Function of the *i*th regular PEs (PE-3 to PE-7).

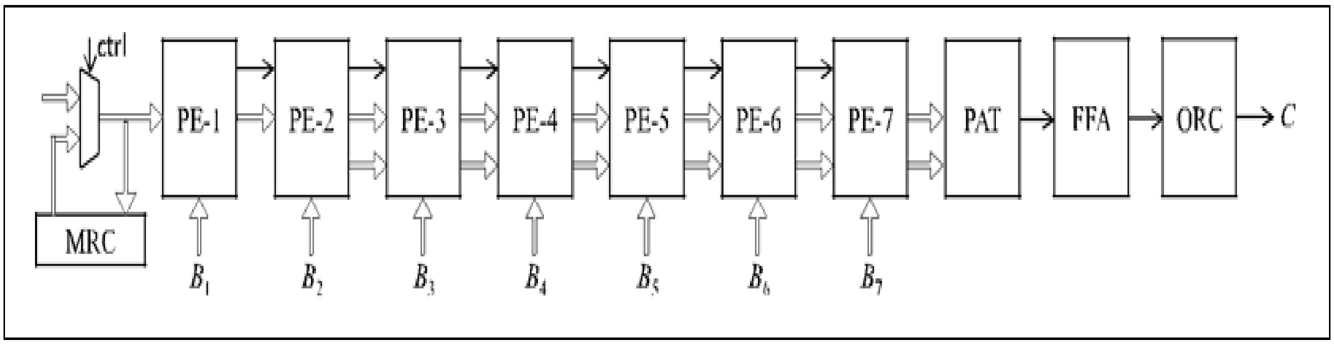


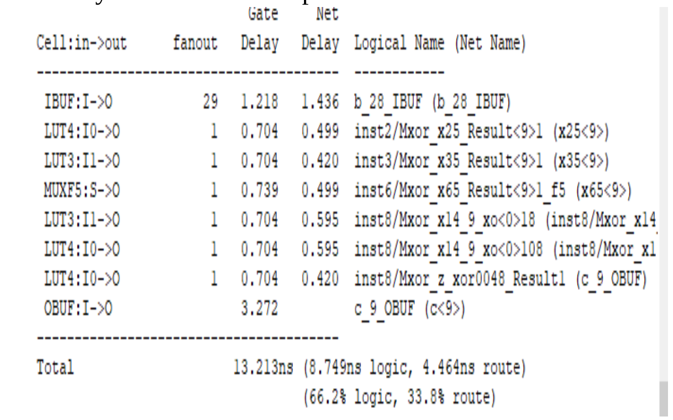
Fig. 5. The generalized time-multiplexed systolic-like array (TM-*n*) for the finite field multiplication over *G F(2^m^)* based on irreducible AOP

The information operand An is affixed with a zero, and (m+1) bit word P−1 is consequently produced sustained by PE-1 multiplexer, initial 7 bits of operand B nourished to duplicate operand (0&A)=P−1 by seven bits of B, added to FFA (introduced to zero) content. During the I th cycle, the result of the diminished type of P7i−1 is sustained to PE-1 for the augmentation of that by the exhibit with the I th, trailed amassing of the incomplete outcome FFA for following cycles i=1,2,and 3.

Inactivity of 12 cycles (principal halfway result PEs, 3 cycles following three sequential incomplete ORC). To deteriment of a gentle ascent in dormancy, the TM-1 structure is equipment proficient, yet has nearly parallel system. Fig's TM-1 structure. 4, incorporates very nearly multiple times less equipment and offers multiple times less exhibition with a 2-cycle ascend in inertness in respect to the Figure structure. 3.

In addition, the structute of TM-1 can be modified as TM-n structure by designing the hardware for n rows to trade off complexity of the hardware.

TM-n design offers n times more throughput and reduced latency compared to TM-1. On the off chance that we utilize a TM-2 system, for example, to execute the DG (fig.2), complete idleness diminished to 11 cycles, with 7 cycles, initial 2 halfway outcomes, 1 cycle for the following 2 incomplete outcomes and 3 cycles (PAT, FFA and ORC).For greater order areas with big s values, the decrease of latency could be more important.

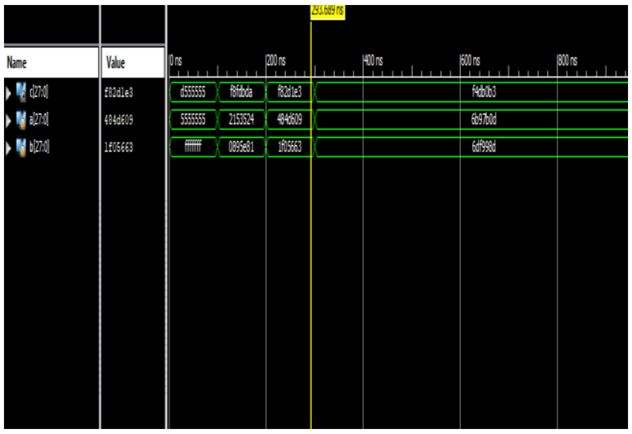


```
                                 Gate     Net
Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
----------------------------------------  ------------
 IBUF:I->O            29   1.218   1.436  b_28_IBUF (b_28_IBUF)
 LUT4:I0->O            1   0.704   0.499  inst2/Mxor_x25_Result<9>1 (x25<9>)
 LUT3:I1->O            1   0.704   0.420  inst3/Mxor_x35_Result<9>1 (x35<9>)
 MUXF5:S->O            1   0.739   0.499  inst6/Mxor_x65_Result<9>1_f5 (x65<9>)
 LUT3:I1->O            1   0.704   0.595  inst8/Mxor_x14_9_xo<0>18 (inst8/Mxor_x14
 LUT4:I0->O            1   0.704   0.595  inst8/Mxor_x14_9_xo<0>108 (inst8/Mxor_x1
 LUT4:I0->O            1   0.704   0.420  inst8/Mxor_z_xor0048_Result1 (c_9_OBUF)
 OBUF:I->O                 3.272          c_9_OBUF (c<9>)
----------------------------------------
Total                     13.213ns (8.749ns logic, 4.464ns route)
                                   (66.2% logic, 33.8% route)
```

IV. CONCLUSIONS

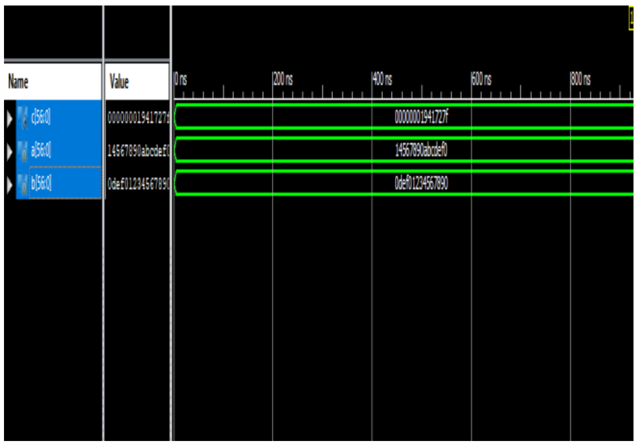
A powerful recursive detailing is recommended for the systeolic usage of sanctioned based limited field augmentation over G F(2 m) in view of 56 bit final AOP where cyclic-left-shift operations achieve the required modular degree reduction. Appropriate DGs are obtained from recursive formulation and from this are engineered systolic multipliers. In this we formulated the importance of Pipe lining for reliable speed operations in the irreducible AOP in the systolic processing element structures.

V. SIMULATION RESULTS

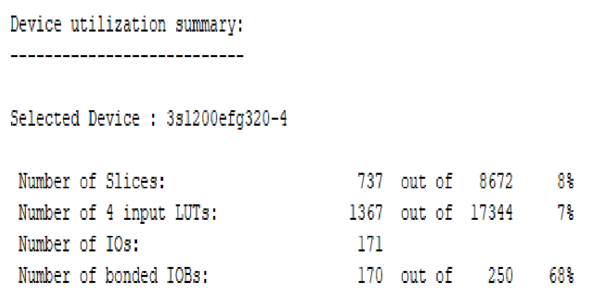


Result of SYSTOLIC STRUCTURE OF 28 BIT

Result of 56 Bit Structure



VI. AREA- AND TIME-COMPLEXITY



```
Device utilization summary:
---------------------------

Selected Device : 3s1200efg320-4

 Number of Slices:                  737  out of   8672     8%
 Number of 4 input LUTs:           1367  out of  17344     7%
 Number of IOs:                     171
 Number of bonded IOBs:             170  out of    250    68%
```

VII. REFERENCES

1. J. Xie, P. Meher, and J. He, “Low-complexity multiplier for *G F*(2^m) based on all-one polynomials,” *IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst.*, vol. 21, no. 1, pp. 168–173, Jan. 2013.
2. J. Menezes, I. F. Blake, S. Gao, S. A. V. R. C. Mullin, and Yaghoobian, Eds., *Applications of Finite Fields*. Boston, MA, USA: Kluwer, 1993.
3. R. Lidl and H. Niederreiter, Eds., *Introduction to Finite Fields TheirApplication*. New York, NY, USA: Cambridge Univ. Press, 1986.

4. [Online]. Available: http://www.csrc.nist.gov/publications
5. S. K. Jain, L. Song, and K. K. Parhi, “Efficient semisystolic architectures for finite-field arithmetic,” *IEEE Trans. Very Large Scale Integr. (VLSI)Syst.*, vol. 6, no. 1, pp. 101–113, Mar. 1998.
6. L. Song and K. K. Parhi, “Low-energy digit-serial/parallel finite field multipliers,” *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 19, no. 2, pp. 149–166, 1998.
7. P. K. Meher, “Systolic formulation for low-complexity serial-parallel implementation of unified finite field multiplication over $G F(2^m)$,” in *Proc. 18th IEEE Int. Conf. Appl.-Specific Syst., Archit. Process. (ASAP)*,Jul. 2007, pp. 134–139.
8. F. Rodriguez-Henriguez and C. K. Koc, “Parallel multipliers based on special irreducible pentanomials,” *IEEE Trans. Comput.*, vol. 52, no. 12, pp. 1535–1542, Dec. 2003.
9. A. Reyhani-Masoleh and M. A. Hasan, “Low complexity bit parallel architectures for polynomial basis multiplication over $G F(2^m)$,” *IEEETrans. Comput.*, vol. 53, no. 8, pp. 945–959, Aug. 2004.
10. W. Tang, H. Wu, and M. Ahmadi, “VLSI implementation of bit-parallel word-serial multiplier in $G F(2^{233})$,” in *Proc. 3rd Int. IEEE-NEWCASConf.*, Jun. 2005, pp. 399–402.
11. H. Wu, “Low complexity bit-parallel multiplier for a class of finite fields,” in *Proc. Int. Conf. Commun., Circuits Syst.*, vol. 4. Jun. 2006, pp. 2565–2568.
12. P. K. Meher, “High-throughput hardware-efficient digit-serial architec-ture for field multiplication over $G F(2^m)$,” in *Proc. 6th Int. Conf. Inf.Commun. Signal Process. (ICICS)*, Dec. 2007, pp. 1–5.
13. P. K. Meher, “Systolic and super-systolic multipliers for finite field $F(2^m)$based on irreducible trinomials,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 5, pp. 1031–1040, May 2008.
14. R. Katti and J. Brennan, “Low complexity multiplication in a finite field