Efficient Bit-Parallel Systolic Polynomial Basis Multiplier over GF(2^8) based on Irreducible Polynomials

S. E. Mathe and L. Boppana

Department of Electronics and Communication Engineering, National Institute of Technology, Warangal – 506004, Telangana, India; ellison@nitw.ac.in

Abstract

Objectives: Multiplication in Galois fields is used in many applications, especially in cryptography. Several algorithms and architectures are proposed in the literature to obtain efficient multiplication operations in Galois fields. Methods/Statistical Analysis: In this paper, based on a modified interleaved modular multiplication algorithm, a bit-parallel systolic multiplier based on generic irreducible polynomials over Galois Field (GF(2^8)) is proposed. Theoretical hardware and speed complexity analysis is performed and the proposed multiplier is compared with other systolic multipliers available in the literature for irreducible polynomials. Findings: The proposed systolic multiplier achieves 21.23% reduction in hardware complexity when compared with the best multiplier among existing multipliers for m = 8. Applications/Improvements: The Field-Programmable Gate Array (FPGA) implementation results for the Advanced Encryption Standard (AES) and Two fish algorithms, incorporating the proposed multiplier and some existing designs, are also presented which indicates that the proposed multiplier achieves low area and low power consumption when compared with other systolic multipliers available in the literature.

Keywords: Bit-Parallel, Cryptography, Field Programmable Gate Arrays, Galois Field, Polynomial Basis, Systolic

1. Introduction

Cryptography is the art of efficient data hiding and transmission over any unreliable medium to avoid unauthorized access. Modern cryptography came into existence with the advent of computers and mainly deals with the development of encryption and decryption algorithms which are very hard to break. Many cryptographic algorithms utilize the multiplication operation in Galois fields and it is the fundamental block for all the arithmetic operations in a GF. The multiplication in a GF can be performed over three basis representations, namely Normal Basis (NB), Polynomial Basis (PB) and Dual Basis (DB), where each basis has its advantages.

Several types of architectures have been reported in the literature for PB multiplication, namely bit-serial, digit-serial, bit-parallel, systolic, semi-systolic, sequential and pipelined. Every architecture type has its own merits and demerits. Systolic architectures are realized using replicas of a fundamental building block to achieve high speed multiplications with the cost of more hardware requirements and hence consume more power.
Therefore, systolic multipliers for Galois fields having less
hardware and less power consumptions are required for
cryptographic applications in portable devices.

Various systolic multipliers are proposed in the
literature for PB multiplication for irreducible poly-
nomials over GF (2^m). The first systolic PB multiplier
was suggested in 1984\(^4\) which perform product-sum com-
putation using a two-dimensional parallel-in parallel-out
structure. In\(^1\) reconsidered an existing algorithm and
proposed a parallel-in parallel-out systolic array which
achieves cascadel, fault tolerant designs with possible
wafer-scale integration while providing comparable hard-
ware complexity, throughput and latency. In\(^2\) proposed
a multiplication algorithm and realized a parallel systolic
architecture using it in\(^4\), in 1998, proposed a pipelined
semi-systolic architecture using the Least Significant Bit
(LSB) first algorithm. In\(^4\) proposed a Montgomery mul-
tiplication algorithm.In\(^6\) proposed a time-independent
Montgomery multiplication algorithm and realized a
systolic array Montgomery multiplier from the proposed
algorithm in 2006. In 2007, a two-dimensional systolic
array multiplier using a Linear Feedback Shift Register
(LFSR) is proposed by\(^16\). In\(^17\), in 2008, presented a time-
dependent and time-independent algorithm and two
bit-parallel systolic multipliers are realized from these
algorithms. In\(^18\) proposed a bit-parallel systolic archi-
tecture based on multiplexers derived from a multiplication
algorithm which uses modified Booth’s recoding along
with the cut-set retiming technique. In\(^18\) proposed a
Montgomery multiplication element realized from an
optimized Montgomery multiplication algorithm. In\(^19\)
proposed a bit-parallel systolic array derived from an LSB-
first algorithm in 2009.In\(^1\), in 2014, proposed a cellular
semi-systolic multiplier from an optimized algorithm.

In this paper, a modified interleaving multiplication
algorithm is introduced to overcome the limitation of
high hardware complexity. The modified algorithm per-
forms multiplication with interleaved modular reduction
of two arbitrary elements for a generic, field defining
irreducible polynomial over GF (2\(^m\)). Subsequently, a
bit-parallel systolic PB architecture is realized for multi-
plcation over GF (2\(^m\)). The proposed systolic multiplier
achieves less hardware complexity and comparable speed
when compared to other systolic multipliers available in
the literature. The multiplier achieves a latency of 8 clock
cycles and throughput of 1 with an initial delay.

The organization of this paper is as follows: Literature
survey is presented in section 2, section

3 presents the proposed systolic multiplier, section 4
gives the results and discussion, followed by conclu-
sions in Section 5.

2. Proposed Bit-Parallel Systolic
PB Multiplier over GF(2\(^8\))

2.1 Conventional PB Representation

The conventional PB representation and multiplication
of the field elements over GF (2\(^m\)) are presented in this
section\(^2\). Let the polynomials P(x) \& Q(x) be any two
elements in the field, T(x) be the field defining irreducible
polynomial and R(x) be the resultant polynomial. Then,

\[
R(x) = (P(x) \times Q(x)) \mod T(x)
\]

(1)

The product of P(x) and Q(x), each of degree at most
m-1, results in an intermediate polynomial given by

\[
S(x) = P(x) \times Q(x) = (p_0 + p_1 x + \ldots + p_{m-1}x^{m-1}) \times (q_0 + q_1 x + \ldots + q_{m-1}x^{m-1})
\]

\[
= s_0 + s_1 x + \ldots + s_{2m-2}x^{2m-2}
\]

(2)

The polynomial S(x) of degree at most 2m-2 is modular
reduced by degree m irreducible polynomial T(x) to
obtain R(x) of degree at most m-1 which is the final result
of the multiplication operation.

\[
R(x) = S(x) \mod T(x) = (s_0 + s_1 x + \ldots + s_{2m-2}x^{2m-2}) \times (t_0 + t_1 x + \ldots + t_{m-1}x^{m-1} + x^m)
\]

\[
r_0 + r_1 x + \ldots + r_{m-1}x^{m-1}
\]

(3)

2.2 Proposed Interleaving Multiplication
Algorithm

Algorithm1: Proposed multiplication algorithm with inter-
leaved modular reduction

1: INITIALIZATION: \( u = 0, \text{cnttr} = 0 \)
2: FOR \( \text{cnttr} = 0 \) TO 7
3: IF \( (q_0 = = 1) \)
4: \( u = u \oplus p \) \hspace{1cm} \( \oplus \) : Logical XOR operation
5: END IF
6: \( p = p << 1 \) \hspace{1cm} \( \ll \) : bit-wise left shift operation
7: IF \( (p_0 = = 1) \)
8: \( p = p \oplus t \)
9: END IF
10: \( q = q >> 1 \) \hspace{1cm} \( \gg \) : bit-wise right shift operation
11: END FOR
Many multiplication methods are available in the literature to perform efficient multiplication operations over polynomial basis. In this paper, a conventional interleaving multiplication algorithm is modified in an efficient manner to obtain a modified interleaving multiplication algorithm which is given in Algorithm1. The steps 3–10 are performed recursively (for \( i = 0, 1, \ldots , 7 \)) to obtain the multiplication operation over \( \text{GF}(2^8) \). Steps 3-5 perform the polynomial multiplication and Steps 7-9 perform the modular reduction.

### 2.3 Proposed Bit-Parallel Systolic Multiplier Architecture

A Based on Algorithm1, a bit-parallel PB systolic multiplier for irreducible polynomials over \( \text{GF}(2^8) \) is proposed in this section. The proposed systolic design is shown in Figure 1(a) consisting of 8 PEs.

The 7 regular PEs (i.e. PE[0] to PE[6]) perform the polynomial multiplication and modular reduction operations concurrently whereas PE[7] (the 8th PE) performs only the polynomial multiplication operation. The functions of the two types of PEs are shown in Fig. 1(b)-(c). PE[7] and the regular PEs are decomposed into 8 V-cells and 16 V-cells, respectively, to derive a more simple, scalable architecture. The systolic multiplier realized using the V-cells is shown in Figure 2.

The first set of 64 V-cells performs the polynomial multiplication operation corresponding to steps 3-5 and the second set of 56 V-cells performs the modular reduction operation corresponding to steps 7-9. The inputs to each cell are \( u_{i,j}, p_{i,j}, q_i \) and \( p_{i,j}, t_i, p_{6,j} \) respectively; where \( i \) denote the index of the coefficient of the polynomial under consideration and \( j \) denotes iteration count. It may be noted that the \( p_{i,j} \) in the modular reduction block is shifted by one bit to the right according to step 6. Figure 3 shows the circuit detail and function of a V-cell.

Figure 1. Proposed systolic multiplier realized using PEs. (a) The systolic multiplier. (b) Circuit detail and function of the regular PE (PE[0] to PE[6]). (c) Circuit detail and function of PE[7].

Figure 2. Proposed bit-parallel systolic multiplier design using V-cells for irreducible polynomials over \( \text{GF}(28) \).

Figure 3. Single carrier FDMA system with companding.

The V-cell consists of an Exclusive-OR (XOR) gate and a 2:1 Multiplexer (MUX). Hence, the gate count of the total structure is 120 XOR gates, 120 MUX gates and 64 latches. The critical path obtained is \( T_X + T_{X'}, \) where \( T_X \) and \( T_{X'} \) are the delays of the XOR gate and the 2:1 MUX respectively. The total clock cycles required by the multiplier to give the first output is 8 and thereafter gives...
Efficient Bit-Parallel Systolic Polynomial Basis Multiplier over GF(2\(^8\)) based on Irreducible Polynomials

3. Results and Discussion

In this section, the hardware and speed complexity analysis of the proposed bit-parallel systolic multiplier is presented and comparison with other systolic multipliers (for irreducible polynomials) available in the literature is performed. The comparisons of the Field Programmable Gate Array (FPGA) implementations of AES and Twofish algorithms is performed incorporating the proposed design and the best of existing designs available in the literature.

### 3.1 Comparison of Hardware and Speed Complexities

In order to estimate the hardware complexity, traditional CMOS logic is used wherein the transistor counts are six transistors for 2-input XOR gate, 2-input AND gate, 1-bit 2:1 MUX, sixteen transistors for 1-bit 4:1 MUX, eight transistors for a 1-bit latch. Real-time circuits from STMicroelectronics are considered to estimate the speed wherein the typical propagation delays of gates used in the various designs are: XOR gate \( t_{pd} = 12\) ns (M74HC08), AND gate \( t_{pd} = 7\) ns (M74HC08), 2:1 MUX \( t_{pd} = 11\) ns (M74HC257), 4:1 MUX \( t_{pd} = 16\) ns (M74HC153). The 3-input XOR gate can be realized by two 2-input XOR gates. Hence, the propagation delay is computed as \( t_{pd} = 24\) ns and the transistor count is twelve. Table 1 compares the hardware complexity, total clock cycles and critical path of the proposed systolic multiplier with other systolic multipliers available in the literature for \( m = 8\).

From Table 1, it can be observed that the proposed multiplier achieves the least hardware complexity when compared with other systolic designs. Specifically, it achieves 61.88%, 65.34%, 61.64%, 36.46%, 36.46%, 45.54%, 61.88%, 61.88%, 57.79%, 61.88%, 52.90%, 38.07%, 47.41%, 36.46%, 38.07%, 42.18%, 64.22%, 21.23%, 67.02%, and 45.29% reduction in hardware complexity for \( m = 8\) when compared with existing multipliers, respectively. The proposed systolic multiplier achieves comparable speed when compared to other systolic multipliers.

### 3.2 FPGA Implementation of AES

The AES algorithm, incorporating the proposed systolic multiplier and some existing systolic multipliers, is modeled in Verilog for the \( GF(f(x)) = x^8 + x^4 + x^3 + x + 1\) (an irreducible polynomial used in the AES algorithm). The simulation and synthesis of these models are carried out for every clock cycle. Therefore, the throughput is 1 with an initial delay of 8 clock cycles.

#### Table 1. Comparison of total transistor count, number of clock cycles, Critical Path (CP) delay, total delay and % reduction in hardware of the proposed systolic multiplier with existing multipliers over GF(2\(^8\))

<table>
<thead>
<tr>
<th>Design</th>
<th>#Transistors</th>
<th>#Clock cycles</th>
<th>CP (ns)</th>
<th>Total Delay (ns)</th>
<th>% Reduction in Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>[7]</td>
<td>5120</td>
<td>24</td>
<td>19</td>
<td>456</td>
<td>61.88</td>
</tr>
<tr>
<td>[12]</td>
<td>5088</td>
<td>15</td>
<td>19</td>
<td>285</td>
<td>61.64</td>
</tr>
<tr>
<td>[13]</td>
<td>3072</td>
<td>9</td>
<td>19</td>
<td>171</td>
<td>36.46</td>
</tr>
<tr>
<td>[14]</td>
<td>3584</td>
<td>16</td>
<td>19</td>
<td>304</td>
<td>45.54</td>
</tr>
<tr>
<td>[15]</td>
<td>4144</td>
<td>9</td>
<td>31</td>
<td>279</td>
<td>52.90</td>
</tr>
<tr>
<td>[16]</td>
<td>3072</td>
<td>8</td>
<td>19</td>
<td>152</td>
<td>36.46</td>
</tr>
<tr>
<td>[17](a)</td>
<td>3152</td>
<td>24</td>
<td>19</td>
<td>456</td>
<td>38.07</td>
</tr>
<tr>
<td>[17](b)</td>
<td>3376</td>
<td>32</td>
<td>19</td>
<td>608</td>
<td>42.18</td>
</tr>
<tr>
<td>[18]</td>
<td>5456</td>
<td>12</td>
<td>40</td>
<td>480</td>
<td>64.22</td>
</tr>
<tr>
<td>[19]</td>
<td>2478</td>
<td>16</td>
<td>19</td>
<td>304</td>
<td>21.23</td>
</tr>
<tr>
<td>[20]</td>
<td>5918</td>
<td>24</td>
<td>19</td>
<td>456</td>
<td>67.02</td>
</tr>
<tr>
<td>[21]</td>
<td>3568</td>
<td>5</td>
<td>19</td>
<td>95</td>
<td>45.29</td>
</tr>
<tr>
<td>Fig. 2</td>
<td>1952</td>
<td>8</td>
<td>23</td>
<td>184</td>
<td>-</td>
</tr>
</tbody>
</table>

**Fig. 1952**

---

AND gate \( t_{pd} = 7\) ns (M74HC08), 2:1 MUX \( t_{pd} = 11\) ns (M74HC257), 4:1 MUX \( t_{pd} = 16\) ns (M74HC153). The 3-input XOR gate can be realized by two 2-input XOR gates. Hence, the propagation delay is computed as \( t_{pd} = 24\) ns and the transistor count is twelve. Table 1 compares the hardware complexity, total clock cycles and critical path of the proposed systolic multiplier with other systolic multipliers available in the literature for \( m = 8\).
out using Xilinx Vivado 2014.2 to verify the functionality of the multipliers. The implementation of the synthesized netlist is carried out on a Xilinx Virtex-7 (XC7V2000TFLG1925-2) FPGA prototype board. The area, total delay and power consumption are tabulated in Table 2.

### Table 2. Comparison of the FPGA implementation results of AES incorporating the proposed systolic multiplier and the best of existing multipliers

<table>
<thead>
<tr>
<th>Design</th>
<th>Total delay (ns)</th>
<th>Power (W)</th>
<th>Area (#LUTs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[13]</td>
<td>66.175</td>
<td>5.749</td>
<td>452871</td>
</tr>
<tr>
<td>[16]</td>
<td>58.159</td>
<td>6.177</td>
<td>575882</td>
</tr>
<tr>
<td>[19]</td>
<td>119.856</td>
<td>7.583</td>
<td>374933</td>
</tr>
<tr>
<td>Fig. 2</td>
<td>70.557</td>
<td>2.976</td>
<td>279055</td>
</tr>
</tbody>
</table>

From Table 2, it can be observed that the proposed systolic multiplier, when compared to the best of existing multipliers\cite{13,16,19}, requires 48.24%, 51.82% & 60.74% less power and requires 38.38%, 51.54% & 25.57% less area, respectively. The proposed systolic multiplier has comparable speed to the existing multipliers\cite{13,16,19}.

### 3.3 FPGA Implementation of Twofish

The Twofish algorithm, incorporating the proposed systolic multiplier and the best of existing systolic multipliers\cite{13,16,19}, is modeled in Verilog for the $GF(x^8+x^7+x^5+x^3+1$ (an irreducible polynomial used in the Twofish algorithm). The simulation and synthesis of these models are carried as in the above sub-section. The hardware complexity, total delay and power consumption are tabulated in Table 3.

### Table 3. Comparison of the FPGA implementation results of Twofish incorporating the proposed systolic multiplier and the best of existing multipliers

<table>
<thead>
<tr>
<th>Design</th>
<th>Total delay (ns)</th>
<th>Power (W)</th>
<th>Area (#LUTs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[16]</td>
<td>73.311</td>
<td>7.286</td>
<td>616823</td>
</tr>
<tr>
<td>[19]</td>
<td>134.758</td>
<td>8.651</td>
<td>435979</td>
</tr>
<tr>
<td>Fig. 2</td>
<td>85.117</td>
<td>3.589</td>
<td>328553</td>
</tr>
</tbody>
</table>

From Table 3, it can be observed that the proposed systolic multiplier, when compared to the best of existing multipliers\cite{13,16,19}, requires 46.43%, 50.74% & 58.51% less power and requires 36.42%, 46.74% & 24.64% less hardware, respectively. The proposed systolic multiplier has comparable speed to the existing multipliers\cite{13,16,19}.

### 4. Conclusions

In this paper, a modified interleaving multiplication algorithm is proposed which performs multiplication of any two arbitrary field elements over $GF(2^n)$ for any generic irreducible polynomial. Based on this method, a bit-parallel systolic PB multiplier is proposed. Hardware and speed complexities are estimated for the proposed systolic multiplier and existing systolic multipliers available in the literature. It may be concluded from the estimation results that the proposed systolic multiplier achieves low hardware complexity and has comparable speed. The AES and Twofish algorithms are implemented on a FPGA incorporating the proposed systolic multiplier and the best of existing multipliers. It can be concluded from the implementation results that the proposed systolic multiplier achieves low power consumption and low area when compared to the best of existing multipliers.

### 5. References

7. Yeh CS, Reed IS, Truong TK. Systolic multipliers for finite fields GF (2^n). Institute of Electrical and Electronics
Efficient Bit-Parallel Systolic Polynomial Basis Multiplier over GF(2^8) based on Irreducible Polynomials