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.

*Author for correspondence*
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 polynomials over GF \((2^m)\). The first systolic PB multiplier was suggested in 1984\(^2\) which perform product-sum computation using a two-dimensional parallel-in parallel-out structure. In\(^{14}\) reconsidered an existing algorithm and proposed a parallel-in parallel-out systolic array which achieves cascadable, fault tolerant designs with possible wafer-scale integration while providing comparable hardware complexity, throughput and latency. In\(^{12}\) proposed a multiplication algorithm and realized a parallel systolic architecture using it in\(^{14}\), in 1998, proposed a pipelined semi-systolic architecture using the Least Significant Bit (LSB) first algorithm. In\(^{16}\) proposed a Montgomery multiplication algorithm. In\(^{12}\) 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 architecture based on multiplexers derived from a multiplication algorithm which uses modified Booth’s recoding along with the cut-set retiming technique. In\(^{12}\) 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\(^{11}\), 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 performs 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 multiplication 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 conclusions 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\(^{22}\). 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

\[
S(x) = P(x) \times Q(x) = (p_0 + p_1x + ... + p_{m-1}x^{m-1}) \times (q_0 + q_1x + ... + q_{m-1}x^{m-1}) = s_0 + s_1x + ... + s_{2m-2}x^{2m-2}
\]

(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.

\[
R(x) = S(x) \mod T(x) = (s_0 + s_1x + ... + s_{2m-2}x^{2m-2}) \times (t_0 + t_1x + ... + t_{m-1}x^{m-1} + x^m) = r_0 + r_1x + ... + r_{m-1}x^{m-1}
\]

(3)

2.2 Proposed Interleaving Multiplication Algorithm

Algorithm \#1: Proposed multiplication algorithm with interleaved modular reduction

1: INITIALIZATION: \(u = 0\), \(cntr = 0\)
2: FOR \(cntr = 0\) TO 7
3: IF \((q_0 = 1)\)
4: \(u = \oplus p\) \(\oplus\) : Logical XOR operation
5: END IF
6: \(p = p << 1\) \(<<\) : bit-wise left shift operation
7: IF \((p_0 = 1)\)
8: \(p = p \oplus t\)
9: END IF
10: \(q = q >> 1\) \(>>\) : 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 Algorithm 1. The steps 3-10 are performed recursively (for \( i = 0, 1, \ldots, 7 \)) to obtain the multiplication operation over 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 Algorithm 1, a bit-parallel PB systolic multiplier for irreducible polynomials over GF(2^8) is proposed in this section. The proposed systolic design is shown in Figure 1(a) consisting of 8 PEs.

![Figure 1](image1)

**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].

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.

![Figure 2](image2)

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

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 \( u_{i,j}, p_{i,j}, q_{i}, t_{i}, p_{6,j}, t_{j} \) 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 3](image3)

**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 \( T_{X} + T_{M} \) where \( T_{X} \) and \( T_{M} \) 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 $t_{pd} = 12\text{ns}$ (M74HC86), AND gate $t_{pd} = 7\text{ns}$ (M74HC08), 2:1 MUX $t_{pd} = 11\text{ns}$ (M74HC257), 4:1 MUX $t_{pd} = 16\text{ns}$ (M74HC153). The 3-input XOR gate can be realized by two 2-input XOR gates. Hence, the propagation delay is computed as $t_{pd} = 24\text{ns}$ 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 $m = 8$.

| Design | #Transistors | #Clock cycles | CP (ns) | Total Delay (ns) | % Reduction in Hardware |
|--------|--------------|---------------|---------|------------------|-------------------------|
| [7]    | 5120         | 24            | 19      | 456              | 61.88%                  |
| [11]   | 5632         | 24            | 31      | 744              | 65.34%                  |
| [12]   | 5088         | 15            | 19      | 285              | 61.64%                  |
| [13]   | 3072         | 9             | 19      | 171              | 36.46%                  |
| [14]   | 3584         | 16            | 19      | 304              | 45.54%                  |
| [15]   | 4144         | 9             | 31      | 279              | 52.90%                  |
| [16]   | 3072         | 8             | 19      | 152              | 36.46%                  |
| [17]a  | 3152         | 24            | 19      | 456              | 38.07%                  |
| [17]b  | 3376         | 32            | 19      | 608              | 42.18%                  |
| [18]   | 5456         | 12            | 40      | 480              | 64.22%                  |
| [19]   | 2478         | 16            | 19      | 304              | 21.23%                  |
| [20]   | 5918         | 24            | 19      | 456              | 67.02%                  |
| [21]   | 3568         | 5             | 19      | 95               | 45.29%                  |
| Fig. 2 | 1952         | 8             | 23      | 184              | -                       |

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%, 45.54%, 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[11,23], 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 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

| Design | Total delay (ns) | Power (W) | Area (#LUTs) |
|--------|-----------------|-----------|--------------|
| [13]   | 66.175          | 5.749     | 452871       |
| [16]   | 58.159          | 6.177     | 575882       |
| [19]   | 119.856         | 7.583     | 374933       |
| Fig. 2 | 70.557          | 2.976     | 279055       |

From Table 2, it can be observed that the proposed systolic multiplier, when compared to the best of existing multipliers\([13,16,19]\), 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\([13,16,19]\).

3.3 FPGA Implementation of Twofish

The Twofish algorithm, incorporating the proposed systolic multiplier and the best of existing systolic multipliers\([13,16,19]\), is modeled in Verilog for the GF\(x^8+x^6+x^5+x^3+1\) (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

| Design | Total delay (ns) | Power (W) | Area (#LUTs) |
|--------|-----------------|-----------|--------------|
| [13]   | 81.471          | 6.699     | 516712       |
| [16]   | 73.311          | 7.286     | 616823       |
| [19]   | 134.758         | 8.651     | 435979       |
| Fig. 2 | 85.117          | 3.589     | 328553       |

From Table 3, it can be observed that the proposed systolic multiplier, when compared to the best of existing multipliers\([13,16,19]\), 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\([13,16,19]\).

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

1. Schneier B. Foundations. In: applied cryptography. 2nd edition. John Wiley and Sons Inc.: UK; 1996. p. 1–4.
2. Data Encryption Standard (DES). NIST - Federal Information Processing Standard Publication (FIPS PUB). 1977 Jan; 46(3):1–22.
3. AES. NIST - Federal Information Processing Standard Publication (FIPS PUB). 2001 Nov 26; 197:5–47.
4. Wang M, Blake IF. Bit serial multiplication in finite fields. SIAM Journal on Discrete Mathematics. 1990; 3(1):140–8.
5. Song L, Parhi KK. Low-energy digit-serial/parallel finite field multipliers. Journal of Very Large Scale Integration (VLSI) Signal Processing Systems for Signal, Image and Video Technology. 1998; 19(2):149–66.
6. Wu H. Low complexity bit-parallel finite field arithmetic using polynomial basis. Proceedings In International Workshop on Cryptographic Hardware and Embedded Systems, Germany; 1999. p. 280–91.
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

Engineers (IEEE) Transactions on Computers. 1984; 100(4):357–60.
8. Song L, Parhi KK. Optimum primitive polynomials for low-area low-power finite field semi-systolic multipliers. Proceedings In Signal Processing Systems SIPS 97-Design and Implementation, UK; 1997. p. 375–84.
9. Ho H. Design and implementation of a PB multiplier architecture over GF(2^8). Journal of Signal Processing Systems. 2014; 75(3):203–8.
10. J. H. Guo, C. L. Wang. Digit-serial systolic multiplier for finite fields GF(2^m). Institute of Electrical and Electronics Engineers (IEEE) Proceedings-Computers and Digital Techniques. 1998; 145(2). 143–148.
11. Wang CL, Lin JL. Systolic array implementation of multipliers for finite fields GF(2^m). Institute of Electrical and Electronics Engineers (IEEE) Transactions on Circuits and Systems. 1991; 38(7):796–800.
12. Wu CW, Chang MK. Bit-level systolic arrays for finite-field multiplications. Journal of Very Large Scale Integration (VLSI) Signal Processing Systems for Signal, Image and Video Technology. 1995; 10(1):85–92.
13. Jain SK, Song L, Parhi KK. Efficient semisystolic architectures for finite-field arithmetic. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Very Large Scale Integration (VLSI) Systems. 1998; 6(1):101–13.
14. Koc CK, Acar T. Montgomery multiplication in GF (2^m). Designs, Codes and Cryptography. 1998; 14(1):57–69.
15. Chiou CW, Lee CY, Deng AW, Lin JM. Efficient Very Large Scale Integration (VLSI) implementation for montgomery multiplication in GF (2^n). Tamkang Journal of Science and Engineering. 2006; 9(4):365–72.
16. Chiou CW, Lee CY, Lin JM. Finite field polynomial multiplier with LFSR. Tamkang Journal of Science and Engineering. 2007; 10(3):253–64.
17. Lee CY. Low-complexity bit-parallel systolic multipliers over GF (2^m). Integration, the Very Large Scale Integration (VLSI) Journal. 2008; 41(1):106–12.
18. Lee CY. Multiplexer-based bit-parallel systolic multipliers over GF (2^m). Computers and Electrical Engineering. 2008; 34(5):392–405.
19. Fournaris AP, Koufopavlou O. Versatile multiplier architectures in GF (2^m) fields using the montgomery multiplication algorithm. INTEGRATION, the Very Large Scale Integration (VLSI) Journal. 2008; 41(3):371–84.
20. Kwon S, Kim CH, Hong CP. More efficient systolic arrays for multiplication in GF (2^n) using LSB first algorithm with irreducible polynomials and trinomials. Computers and Electrical Engineering. 2009; 35(1):159–67.
21. Kim KW, Jeon JC. PB multiplier using cellular systolic architecture. Institution of Electronics and Telecommunication Engineers (IETE) Journal of Research. 2014; 60(2):194–9.
22. Erdem SS, Y anik T, Koc CK. PB multiplication over GF (2^n). Acta Applicandae Mathematicae. 2006; 93(1):33–55.
23. Henriquez FR, Perez AD, Saqib NA, Koc CK. Binary Finite Field Arithmetic. In: Cryptographic Algorithms on Reconfigurable Hardware. 1st edition, Springer: USA; 2007. p. 139–88.