Comments on "Low-Latency Digit-Serial Systolic Double Basis Multiplier over $GF(2^m)$ Using Subquadratic Toeplitz Matrix-Vector Product Approach"

Arash Reyhani-Masoleh

Abstract—The digit-serial systolic double basis multiplier architecture proposed in the above paper does not generate the correct multiplication results as it requires more latches to process digits of inputs in appropriate clock cycles. In this comment, we present the corrected architecture and obtain its time and area complexities. More importantly, we show that the claims made by the authors regarding having significantly lower time and area complexities than its counterpart are not valid.

Index Terms—Finite field, digit-serial, multiplier, systolic

1 Introduction

Let A, B, and $C = AB \bmod F(x)$ be elements in the finite field $GF(2^m)$ constructed by an irreducible trinomial $F(x) = x^m + x^n + 1, n \leq m/2$. In [1], B and C are represented in dual basis, whereas A is represented in the polynomial basis. In fact, $k^2d - m$ zeros are padded after the most significant bits of the field element A to represent it in the polynomial basis as $A = \sum_{i=0}^{k^2-1} A_i x^{id}$, where $A_i = a_{id} + a_{id+1}x + \cdots + a_{id+d-1}x^{d-1}$ contains d bits of A and $k = \lceil \sqrt{\frac{m}{d}} \rceil$. Then, it is shown in [1] that the coordinates of $C = AB \bmod F(x)$ can be obtained from

$$\begin{aligned} C &= (B(A_0 + A_1x^d + \cdots + A_{k-1}x^{(k-1)d}) \\ &\quad + Bx^{dk}(A_k + A_{k+1}x^d + \cdots + A_{2k-1}x^{(k-1)d}) \\ &\quad + \cdots + Bx^{dk(k-1)}(A_{k(k-1)} + \cdots + A_{k^2-1}x^{(k-1)d}) \\ &\quad) \bmod F(x) = (C_0 + C_1 + \cdots + C_{k-1}) \bmod F(x) \end{aligned} \tag{1}$$

which is computed by Algorithm 2 of [1]. The digit-serial systolic double basis multiplier over $GF(2^m)$ proposed in [1] is shown in Fig. 1a. In this figure, the register B initializes with the coordinates of the field element B, the $R3$ module uses to update the register B with $Bx^{dk} \bmod F(x)$ in each clock cycle. The $R2$ module performs $C_i \bmod (x^m + 1)$. The processing element (PE) of this architecture (Fig. 1b) implements the subquadaratic Toeplitz matrix-vector product (TMVP) approach proposed in [2]. In Fig. 1b, the R_1 module performs the computation of $B_{in}x^d \bmod F(x)$ which appears in the output of the m-bit latch L after each cycle.

2 Corrected Architecture

If one removes all latches in Fig. 1b from all k PEs, then Fig. 1a correctly computes (1) in k clock cycles. This scheme has a long propagation delay. To reduce the propagation delay, the authors of [1] have added latches at the outputs of PEs as shown in Fig. 1b. However, they missed to add latches in the A_i,$0 \leq i \leq k-1$, input of all k PEs. Specifically, we propose the following to correct the architecture of Fig. 1a proposed in [1].

• *The author is with the Department of Electrical and Computer Engineering, Western University, ON, Canada. E-mail: areyhani@uwo.ca.*

Manuscript received 17 May 2014; revised 17 Jan. 2015; accepted 19 Jan. 2015; date of current version 13 Mar. 2015.
Recommended for acceptance by F. Rodríguez-Henríquez.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2015.2401024

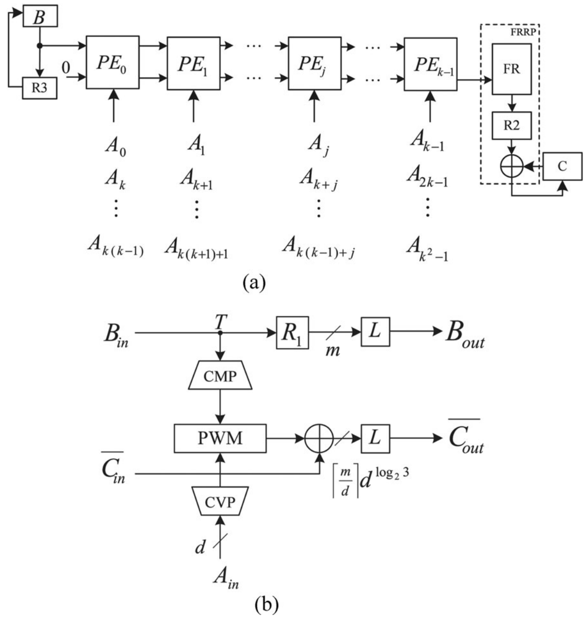

Fig. 1. (a) The original digit-serial systolic multiplier architecture, (b) its processing element [1].

Proposition 1. *To obtain the corrected architecture and hence to compute (1) correctly, one needs to add i, $1 \leq i \leq k-1$, d-bit latches between A_i and the input of PE_i in the architecture of Fig. 1a.*

Proof. At the beginning of computation, the input of PE_0 in Fig. 1a is B. Then, the output of the m-bit latches of PE_0 generates BA_0 after the first clock cycle. To reach A_1 to the PE_1 after one clock cycle, i.e., at the same time the output of PE_0 generates BA_0, a d-bit latch should be inserted between A_1 and the input of PE_1. Thus, the output of PE_1 generates $B(A_0 + A_1x^d)$ after two clock cycles. Similarly, to add BA_2x^{2d} to $B(A_0 + A_1x^d)$ and appear it at the output of PE_2, one needs to make two clock cycles delay for A_2. To implement two clock cycles delay, one needs to insert two blocks of d-bit latches between A_2 and the input of PE_2. Then, A_2 reaches PE_2 at the time $B(A_0 + A_1x^d)$ available at the input of PE_2 and hence the output of PE_2 generates $B(A_0 + A_1x^d + A_2x^{2d})$ after three clock cycles. Therefore, by adding $k-1$ blocks of d-bit latches at the input A_{k-1} of PE_{k-1}, the output of PE_{k-1} generates $B(A_0 + A_1x^d + A_2x^{2d} + \cdots + A_{k-1}x^{(k-1)d})$ after k clock cycles. Since after the first clock cycle the input of PE_0 becomes $Bx^{dk} \bmod F(x)$, one can verify that the output of PE_{k-1} becomes $Bx^{dk}(A_k + A_{k+1}x^d + A_{k+2}x^{2d} + \cdots + A_{2k-1}x^{(k-1)d}) \bmod F(x)$ after $k+1$ clock cycles. Similarly after $k+i$ clock cycles for $1 \leq i \leq k-1$, the output of PE_{k-1} computes $Bx^{idk}(A_{ik} + A_{ik+1}x^d + A_{ik+2}x^{2d} + \cdots + A_{ik-1}x^{(k-1)d}) \bmod F(x)$. Therefore, the corrected architecture computes (1) after $2k$ clock cycles. □

To illustrate the computations for the components of all PEs in each clock cycle, the readers is referred to [1, Table 3] for an example over $GF(2^{36})$.

2.1 Complexity Analysis

The number of d-bit latches added to the original architecture of Fig. 1a is $1 + 2 + \cdots + k - 1 = k(k-1)/2$. Thus, the total number of 1-bit latches becomes $0.5k(k-1)d + (k+2)m + kS_3$ using the complexities presented in [1]. Also, no multiplexers (MUXs) are reported for the multiplier architecture of Fig. 1. It is noted that m number of 2:1 MUXs are required to initialize the register B with the coordinates of the field element B. Moreover, the time delay of an AND gate (T_A) is missing in the propagation delay. As a result,

TABLE 1
Number of Gates, Total GE and Latency Comparison between the Digit-Serial Multipliers of [1] and [3] over $GF(2^{409})$

d	2		4		8		16		32		64	
Digit-size	$d=2$	$kd=30$	$d=4$	$kd=44$	$d=8$	$kd=64$	$d=16$	$kd=96$	$d=32$	$kd=128$	$d=64$	$kd=192$
# AND [3]	820	12,600	1,648	19,360	3,328	28,672	6,656	46,080	13,312	65,536	28,672	110,592
# AND [1]		9,225		10,197		11,232		12,636		12,636		15,309
# XOR [3]	824	12,660	1,656	19,448	3,344	28,800	6,688	46,272	13,376	65,792	28,800	110,976
# XOR [1]		16,256		24,191		32,057		40,813		45,443		59,727
# Latch [3]	1,647	1,771	1,661	1,893	1,689	1,985	1,713	2,209	1,761	2,433	1,985	2,881
# Latch [1]		16,388		15,734		15,546		16,148		15,282		17,546
Total GE [3]	8,849	47,711	11,601	70,195	17,182	100,884	28,120	158,428	49,996	222,628	100,884	370,996
Total GE [1]		105,498		120,131		136,452		157,976		163,989		204,388
Latency [3]	410	30	206	22	104	16	52	12	26	8	14	6
Latency [1]		30		22		16		12		8		6

its propagation delay becomes $T_A + (2 + \log_2 d)T_X$. Other complexities of the proposed architecture in [1] remain the same.

3 Comparison and Conclusion

In [1], the authors claim that "If the selected digit size is d bits, the proposed digit-serial multiplier for both polynomials, i.e., trinomials and AESPs, requires the latency of $2\lceil\sqrt{\frac{m}{d}}\rceil$, while traditional ones take at least $O(\lceil\frac{m}{d}\rceil)$ clock cycles." In this section, we show that the assumption of having the digit size of d in the proposed architecture is not correct and hence this claim is not valid. Let us review the definition of the digit size. Parhi mentioned in [4] that "the number of bits processed in each clock cycle in the digit-serial systems is referred to as the *digit size*". Similarly, it is defined in [5] that "the number of coefficients that are processed in parallel is defined to be the digit size d". Based on this definition, one can easily see that the number of bits/coefficients that are processed in each clock cycle in Fig. 1a and the corrected architecture is kd as the digits of A_i, $A_{i+1}, \ldots, A_{i+k-1}$are sampled in parallel in each clock cycle for $0 \le i \le k(k-1)$. Therefore, the digit size of the proposed architecture in [1] is kd; not d as claimed in [1]. To have a fair comparison, we obtain the latency of any traditional multiplier using the digit size of kd as follows.

Proposition 2. *Let d and $O(\lceil\frac{m}{d}\rceil)$ be the digit size and the latency of a traditional digit-serial multiplier, respectively. Then, if the digit size is increased to kd, $k = \lceil\sqrt{\frac{m}{d}}\rceil$, its latency will be decreased to $O(\lceil\sqrt{\frac{m}{d}}\rceil)$.*

Proof. One can easily prove it by substituting $k = \lceil\sqrt{\frac{m}{d}}\rceil$ in $O(\lceil\frac{m}{kd}\rceil)$. □

Therefore, one can easily see that the proposed architecture in [1] has the same latency as the one compared with, i.e., [3]. Also, the propagation delay of [1], i.e., $T_A + (2 + \log_2 d)T_X$, is not lower than the one of [3], i.e., $T_A + T_{MUX} + \log_2 dT_X$. To have a complete comparison table (see Table 1 of this comment), we have applied the formulations provided in [1, Table 5] (except for number of latches in [1] which is obtained from this comment) for the values of d presented in [1, Table 6]. In Table 1 of this comment, two values are reported (one for digit size d and another one for digit size kd) for the scheme presented in [3]. In this table, total gate equivalent (GE) of these schemes are estimated based on the used cell areas in the 65 nm CMOS technology of STMicroelectronics, i.e., a 2-input NAND gate area = 2.08 nm^2, a 2-input AND gate area = 2.6 nm^2, a 2-input XOR gate area = 4.16 nm^2, a D flip flop area = 7.8 nm^2. Looking at the original comparisons (comparing values between [3] for digit sizes of d with [1] for digit sizes of kd), one can see that the area complexity of [1] is higher than that of [3].

Comparing two schemes with the same low values of digit sizes kd, one can easily see from Table 1 that the proposed architecture in [1] does not outperform the counterpart in terms of area complexity for low digit sizes and hence it is not suitable for the resource constrained environments despite what has been claimed in the Conclusion of [1]. As a result, the corrected architecture of [1] is only suitable for high-performance applications using high digit sizes.

Acknowledgments

The author would like to thank associate editor and the reviewers for their valuable comments. This work has been supported in part by NSERC Discovery grant awarded to the author. Arash Reyhani-Masoleh is the corresponding author.

References

[1] J.-S. Pan, R. Azarderakhsh, M. Mozaffari Kermani, C.-Y. Lee, W.-Y. Lee, C. Chiou, and J.-M. Lin, "Low-latency digit-serial systolic double basis multiplier over $GF(2^m)$ using subquadratic Toeplitz matrix-vector product approach," *IEEE Trans. Comput.*, vol. 63, no. 5, pp. 1169–1181, May 2014.
[2] H. Fan and M. A. Hasan, "A new approach to subquadratic space complexity parallel multipliers for extended binary fields," *IEEE Trans. Comput.*, vol. 56, no. 2, pp. 224–233, Feb. 2007.
[3] S. Talapatra, H. Rahaman, and S. Saha, "Unified digit serial systolic Montgomery multiplication architecture for special classes of polynomials over $GF(2^m)$," in *Proc. 13th Euromicro Conf. Digital Syst. Des.: Archit., Methods Tools*, Sep. 2010, pp. 427–432.
[4] K. Parhi, "A systematic approach for design of digit-serial signal processing architectures," *IEEE Trans. Circuits Syst.*, vol. 38, no. 4, pp. 358–375, Apr. 1991.
[5] S. Kumar, T. Wollinger, and C. Paar, "Optimum digit serial $GF(2^m)$ multipliers for curve-based cryptography," *IEEE Trans. Comput.*, vol. 55, no. 10, pp. 1306–1311, Oct. 2006.