Low-Delay 4, 5 and 6-Term Karatsuba Formulae in $\mathbb{F}_2[x]$ Using Overlap-free Splitting

Haining Fan
fhn@tsinghua.edu.cn

Abstract

The overlap-free splitting method, i.e., even-odd splitting and its generalization, can reduce the XOR delay of a Karatsuba multiplier. We use this method to derive Karatsuba formulae with one less XOR delay in each recursive iteration. These formulae need more multiplication operations, and are trade-offs between space and time.

We also show that "finding common subexpressions" performs better than "the refined identity" in 4-term formula: we reduce the number of XOR gates given by Cenk, Hasan and Negre in *IEEE T. Computers* in 2014.

Index Terms

Karatsuba algorithm, polynomial multiplication, even-odd splitting, overlap-free splitting

I. INTRODUCTION

Even-odd splitting of polynomials and its generalization were first used in Karatsuba algorithms in 2007 [1]. These splitting methods eliminate overlaps in the reconstruction step, and reduce XOR gate delays of subquadratic Karatsuba multipliers in $\mathbb{F}_2[x]$ by about 33% and 25% for $n = 2^t$ and $n = 3^t$ $(t > 1)$, respectively. On the other hand, many efforts have been made to reduce the multiplicative complexity $M(n)$ of a Karatsuba formula, and these improvements reduce space complexities of Karatsuba multipliers.

In this work, we focus on optimising the time complexity, and give 4-term and 5-term Karatsuba formulae with one less XOR gate delay in each recursive iteration. We first replace the original splitting method by the above overlap-free splitting method in splitting steps of the two existing low-$M(n)$ formulae, and then reduce XOR delays by increasing $M(n)$ slightly.

We also give an improvement on the 4-term formula presented by Cenk, Hasan and Negre in [2]. Their formula combines the overlap-free splitting method and "the refined identity" together to reduce both the XOR space complexity $\mathcal{S}^{\oplus}(n)$ and the XOR time complexity $\mathcal{D}^{\oplus}(n)$. The idea behind "the refined identity" is presented by Zhou and Michalik [3] (for the case $n = 2^i$) and Bernstein [4]. The space and time complexities of formula in [2] are as follows:

$$\mathcal{S}^{\oplus}(n) = 9\mathcal{S}^{\oplus}(n/4) + 10n - 17 = \frac{47}{8}n^{\log_4 9} - 8n + \frac{17}{8} \quad and \quad \mathcal{D}^{\oplus}(n) = 4\log_4 n\, T_X = 2\log_2 n\, T_X,$$

where "T_X" is the delay of one 2-input XOR gate.

We optimise $\mathcal{S}^{\oplus}(n)$ by marking common subexpressions explicitly. While the method "finding common subexpressions" performs worse for the case $n = 3^i$, see [2], it wins for $n = 4^i$. The new formula needs 1 less addition in each recursive iteration, and thus improves the above space complexity bound to:

$$\mathcal{S}^{\oplus}(n) = 9\mathcal{S}^{\oplus}(n/4) + 10n - 18 = \frac{46}{8}n^{\log_4 9} - 8n + \frac{18}{8} \quad and \quad \mathcal{D}^{\oplus}(n) = 4\log_4 n\, T_X = 2\log_2 n\, T_X.$$

XOR complexities of formulae in this work are listed in the following table.

TABLE I

COMPARISONS OF COMPLEXITIES

n	Algorithm	#Multiplication	#XOR	XOR Gate Delay (T_X)
4^i	[2]	9	$\frac{47}{8}n^{\log_4 9} - 8n + \frac{17}{8}$	$4\log_4 n = 2.00\log_2 n$
	Eq. (2)	9	$\frac{46}{8}n^{\log_4 9} - 8n + \frac{18}{8}$	$4\log_4 n = 2.00\log_2 n$
	Eq. (3)	10		$3\log_4 n = 1.50\log_2 n$
	Schoolbook	16		$1.00\log_2 n$
5^i	Eq. (4)	13		$5\log_5 n \approx 2.15\log_2 n$
	Eq. (7)	15		$4\log_5 n \approx 1.72\log_2 n$
6^i	Eq. (8)	17		$5\log_6 n \approx 1.93\log_2 n$
	Eq. (9)	21		$4\log_6 n \approx 1.55\log_2 n$

II. Improve $\mathcal{S}^{\oplus}(n)$ of the 4-Term Karatsuba Formula

Let $A = a_3x^3 + a_2x^2 + a_1x + a_0$, $B = b_3x^3 + b_2x^2 + b_1x + b_0$ and $C = AB = \sum_{i=0}^{6} c_ix^i$. Cenk, Hasan and Negre combine the overlap-free splitting method and the refined identity, and present a formula with low XOR delay [2, Section 3.3]. For the initial step $n = 4$, the numbers of XOR gates needed in the reconstruction step are listed in the following table:

TABLE II
RECONSTRUCTION STEP FOR $n = 4$

Computations	Degree	#XOR
$R_0 = P_0 + xP_1 + x^2P_2 + x^3P_3$	$Deg(R_0) = 3$	0
$R_1 = (1+x)R_0$	$Deg(R_1) = 4$	3
$R_2 = R_1 + xP_{01} + x^3P_{23}$	$Deg(R_2) = 4$	2
$R_3 = P_{02} + xP_{13}$	$Deg(R_3) = 1$	0
$R_4 = (1+x)R_3$	$Deg(R_4) = 2$	1
$R_5 = R_4 + xP_{0123}$	$Deg(R_5) = 2$	1
$R_6 = (1+x^2)R_2$	$Deg(R_6) = 6$	3
$C = R_6 + x^2R_5$	x^2R_5 has 3 bits	3
Total		13

In this table, product terms $P_i = a_ib_i$, $P_{01} = (a_0+a_1)(b_0+b_1), P_{02} = (a_0+a_2)(b_0+b_2), P_{13} = (a_1+a_3)(b_1+b_3), P_{23} = (a_2+a_3)(b_2+b_3)$ and $P_{0123} = (a_0+a_1+a_2+a_3)(b_0+b_1+b_2+b_3)$ are elements in $\mathbb{F}_2$. There are $2*5 = 10$ XOR gates in P_{01}, P_{02}, P_{13}, P_{23} and P_{0123}. So we have $\mathcal{S}^{\oplus}(4) = 10 + 13 = 23$.

The total number of XOR gates for $n = 4^i$ is given in Table 3 and Eq. (8) of [2]:

$$\mathcal{S}^{\oplus}(1) = 0, \qquad \mathcal{S}^{\oplus}(4) = 10 + 13 = 23,$$
$$\mathcal{S}^{\oplus}(n) = 9\mathcal{S}^{\oplus}(n/4) + 10n - 17 = \tfrac{47}{8}n^{\log_4 9} - 8n + \tfrac{17}{8}.$$

We now optimise $\mathcal{S}^{\oplus}(n)$ by finding common subexpressions. Given a k-term Karatsuba formula using the original Karatsuba splitting method, it is easy to transform it to a formula using the overlap-free splitting method: combining coefficients of x^i and x^{i+k} together for $0 \le i \le k-2$. Take $k = 4$ as an example, we transform the following 9-multiplication formula

$$\begin{aligned} C \;=\; & P_0 + x(P_{01} + P_0 + P_1) + x^2(P_0 + P_1 + P_2 + P_{02}) + \\ & x^3(P_{0123} + P_{13} + P_{02} + P_{23} + P_2 + P_3 + P_{01} + P_0 + P_1) + \\ & x^4(P_{13} + P_1 + P_2 + P_3) + x^5(P_{23} + P_2 + P_3) + x^6P_3 \end{aligned}$$

to

$$\begin{aligned} C \;=\; & x^0[P_0 + x^4(P_{13} + P_1 + P_2 + P_3)] + x^2[(P_0 + P_1 + P_2 + P_{02}) + x^4P_3] \qquad (1) \\ & x[(P_{01} + P_0 + P_1) + x^4(P_{23} + P_2 + P_3)] + x^3(P_{0123} + P_{13} + P_{02} + P_{23} + P_2 + P_3 + P_{01} + P_0 + P_1). \end{aligned}$$

This is a rewrite of the overlap-free formula in [2, Section 3.3]. Please note that coefficients of x^0, x, x^2 and x^3 are summations of product terms P_*, and they are polynomials in $x^k = x^4$.

In order to count the number of XOR gates in this formula, we mark common subexpressions in different colors, denote the 3 shift-adds $((\cdots) + x^4(\cdots))$ by $\oplus$, and label the 12 actual "+"s in subscripts:

$$\begin{aligned} C \;=\; & [P_0 \oplus x^4(P_{13} +_1 P_1 +_2 P_2 +_3 P_3)] + x^2[(P_0 +_4 P_1 +_5 P_2 +_6 P_{02}) \oplus x^4P_3] + \qquad (2) \\ & x[(P_{01} +_7 P_0 +_4 P_1) \oplus x^4(P_{23} +_{11} P_2 +_3 P_3)] + \\ & x^3(P_{0123} +_8 P_{13} +_9 P_{02} +_{10} P_{23} +_{11} P_2 +_3 P_3 +_{12} P_{01} +_7 P_0 +_4 P_1). \end{aligned}$$

There are $2*5*\frac{n}{4}$ XORs in products $P_{01}, P_{02}, P_{13}, P_{23}$ and P_{0123}. These products are polynomials in x^4 with the same degree $2*(\frac{n}{4}-1)=\frac{n}{2}-2$. So the 3 shift-add $\oplus$ operations need $3*(\frac{n}{2}-2)$ XOR gates, and the 12 actual $+_i$ operations $12*(\frac{n}{2}-1)$ XOR gates. Therefore, we have $10*\frac{n}{4}+3*(\frac{n}{2}-2)+12*(\frac{n}{2}-1)=10n-18$ and

$$\mathcal{S}^{\oplus}(1)=0, \quad \mathcal{S}^{\oplus}(4)=10+12=22, \qquad \text{Note}: \mathcal{S}^{\oplus}(4)=23 \text{ in [2]}.$$
$$\mathcal{S}^{\oplus}(n)=9\mathcal{S}^{\oplus}(n/4)+10n-18=\tfrac{46}{8}n^{\log_4 9}-8n+\tfrac{18}{8}.$$

This improves the bound $\mathcal{S}^{\oplus}(n)=9\mathcal{S}^{\oplus}(n/4)+10n-17=\frac{47}{8}n^{\log_4 9}-8n+\frac{17}{8}$ presented in [2].

The XOR gate delay of coefficient of x^3 in (2) is $4T_X$ because we can compute it using

$$P_{0123}+_8[P_{13}+_9 P_{02}]+_{10}[P_{23}+_{11}(P_2+_3 P_3)]+_{12}[P_{01}+_7(P_0+_4 P_1)],$$

where P_{0123} and three $[\cdots]$ need $2T_X$ each. Therefore, the XOR gate delay of Eq. (2) is $\mathcal{D}^{\oplus}(n)=4\log_4 n\, T_X$.

The other advantage of this new overlap-free formula (2) is that products $P_{01}, P_{02}, P_{13}, P_{23}$ and P_{0123} each have $2*(\frac{n}{4}-1)+1=\frac{n}{2}-1$ bits because their degrees are all $2*(\frac{n}{4}-1)=\frac{n}{2}-2$. But R_0 in [2, Table 3], which also uses the overlap-free splitting and has the same $\mathcal{D}^{\oplus}(n)=4\log_4 n\, T_X$, has $4*(\frac{n}{2}-1)=2n-4$ bits. We need to manipulate this long polynomial in the following step of R_1, R_2 and R_6.

We note that Find and Peralta adopt the original splitting method, and obtain the bound $\mathcal{S}^{\oplus}(1)=0$ and $\mathcal{S}^{\oplus}(n)=9\mathcal{S}^{\oplus}(n/4)+\frac{34}{4}n-12$ [5]. This is an improvement to Bernstein's bound $\mathcal{S}^{\oplus}(n)=9\mathcal{S}^{\oplus}(n/4)+\frac{34}{4}n-11$ [4, p. 327] or [2, Eq. (5)]. The XOR gate delays of these formulae are $\mathcal{D}^{\oplus}(n)=5\log_4 n\, T_X$ because of the overlap.

III. 4-term Karatsuba Formula with 10-multiplication and 3-T_X

In order to reduce the XOR delay in (1), we eliminate P_{0123} using the following identity

$$P_{0123}=P_{01}+P_{02}+P_{03}+P_{12}+P_{13}+P_{23}.$$

This identity introduces 2 new multiplications P_{12} and P_{03}. So we have the following 10-multiplication formula:

$$\begin{aligned} C &= x^0[P_0+x^4(P_{13}+P_1+P_2+P_3)]+x^2[(P_0+P_1+P_2+P_{02})+x^4P_3] \qquad (3)\\ &\quad x[(P_{01}+P_0+P_1)+x^4(P_{23}+P_2+P_3)]+x^3(P_{12}+P_{03}+[P_2+P_3]+[P_0+P_1])\\ &= x^0\{[(P_0+x^4P_1)+x^4(P_2+P_3)]+x^4P_{13}\}+x^2\{[(P_0+P_1)+(P_2+x^4P_3)]+P_{02}\}\\ &\quad x\{[P_{01}+(P_0+P_1)]+x^4[P_{23}+(P_2+P_3)]\}+x^3\{P_{12}+P_{03}+(P_2+P_3)+(P_0+P_1)\}.\end{aligned}$$

The XOR gate delays of coefficients of x^0, x, x^2 and x^3 in "$\{\}$" are all $3T_X$. So the final XOR gate delay is $\mathcal{D}^{\oplus}(n)=3\log_4 n\, T_X=1.5\log_2 n\, T_X$.

There are $2*6*\frac{n}{4}$ XORs in products $P_{01}, P_{02}, P_{03}, P_{12}, P_{13}$ and P_{23}. These products are polynomials in x^4 with the same degree $2*(\frac{n}{4}-1)=\frac{n}{2}-2$. So the 3 shift-add operations need $3*(\frac{n}{2}-2)$ XOR gates, and the 11 addition operations $11*(\frac{n}{2}-1)$ XOR gates. Therefore, we have $12*\frac{n}{4}+3*(\frac{n}{2}-2)+11*(\frac{n}{2}-1)=10n-17$ and

$$\mathcal{S}^{\oplus}(1)=0, \quad \mathcal{S}^{\oplus}(4)=23,$$
$$\mathcal{S}^{\oplus}(n)=10\mathcal{S}^{\oplus}(n/4)+10n-17.$$

IV. 5-term Karatsuba Formula with 15-multiplication and 4-T_X

Let $A = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$, $B = b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$ and $C = AB = \sum_{i=0}^{8} c_i x^i$. We transform the following 13-multiplication formula in [6] using the overlap-free splitting method. This formula is based on the CRT moduli polynomials $(x-\infty)^3$, x^3, $(x+1)^1$, x^2+x+1.

$$\begin{aligned} C &= P_0 + x(P_0 + P_1 + P_{01}) + x^2(P_0 + P_1 + P_2 + P_{02}) + \\ &\quad x^3(P_0 + P_4 + P_3 + P_2 + P_{24} + P_{01234} + P_{023} + P_{0134}) + \\ &\quad x^4(P_0 + P_1 + P_{01} + P_4 + P_3 + P_{34} + P_{01234} + P_{023} + P_{124}) + \\ &\quad x^5(P_0 + P_1 + P_2 + P_{02} + P_4 + P_{01234} + P_{124} + P_{0134}) + \\ &\quad x^6(P_4 + P_3 + P_2 + P_{24}) + x^7(P_4 + P_3 + P_{34}) + x^8 P_4. \end{aligned}$$

The resulting low-delay formula is

$$\begin{aligned} C &= x^0[P_0 + x^5(P_0 + P_1 + P_2 + P_{02} + P_4 + P_{01234} + P_{124} + P_{0134})] + \\ &\quad x^1[P_0 + P_1 + P_{01} + x^5(P_4 + P_3 + P_2 + P_{24})] + \\ &\quad x^2[P_0 + P_1 + P_2 + P_{02} + x^5(P_4 + P_3 + P_{34})] + \\ &\quad x^3[P_0 + P_4 + P_3 + P_2 + P_{24} + P_{01234} + P_{023} + P_{0134} + x^5 P_4] + \\ &\quad x^4[P_0 + P_1 + P_{01} + P_4 + P_3 + P_{34} + P_{01234} + P_{023} + P_{124}]. \end{aligned} \tag{4}$$

The XOR gate delays of coefficients of x^0, x^3 and x^4 are all $5T_X$. In order to reduce it to $4T_X$, we use the identity $P_{01234} = P_{0134} + P_{023} + P_{124} + P_{03} + P_{14} + P_2$ to eliminate P_{01234} at the cost of introducing two new products P_{03} and P_{14}, and obtain the following expression of coefficient of x^4

$$c_4 = P_0 + P_1 + P_2 + P_3 + P_4 + P_{01} + P_{34} + P_{03} + P_{14} + P_{0134}.$$

But the XOR delay of this formula is still $5T_X$. In order to reduce it to $4T_X$, We use the identity $P_{03} = P_0 + P_3 + a_0 * b_3 + a_3 * b_0$, which introduces two new products $a_0 * b_3$ and $a_3 * b_0$, and get the following $4T_X$ formula

$$c_4 = P_1 + P_2 + P_4 + P_{01} + P_{34} + a_0 * b_3 + a_3 * b_0 + P_{14} + P_{0134}.$$

For other coefficients, we have

$$\begin{aligned} & [P_0 + P_4 + P_3 + P_2 + P_{24} + P_{01234} + P_{023} + P_{0134} + x^5 P_4] \\ = &\quad P_0 + P_4 + P_3 + P_{24} + P_{124} + P_{03} + P_{14} + x^5 P_4 \\ = &\quad P_4 + P_{24} + P_{124} + a_0 * b_3 + a_3 * b_0 + P_{14} + x^5 P_4 \end{aligned}$$

and

$$\begin{aligned} & [P_0 + x^5(P_0 + P_1 + P_2 + P_{02} + P_4 + P_{01234} + P_{124} + P_{0134})] \\ = &\quad P_0 + x^5(P_0 + P_1 + P_2 + P_{02} + P_4 + P_{023} + P_{03} + P_{14} + P_2) \\ = &\quad P_0 + x^5(P_1 + P_3 + P_4 + P_{02} + P_{023} + a_0 * b_3 + a_3 * b_0 + P_{14}). \end{aligned}$$

The final $4T_X$ formula is

$$\begin{aligned} C = \; & x^0[P_0 + x^5(P_1 + P_3 + P_4 + P_{02} + P_{023} + a_0 * b_3 + a_3 * b_0 + P_{14})] + \\ & x^1[P_0 + P_1 + P_{01} + x^5(P_4 + P_3 + P_2 + P_{24})] + \\ & x^2[P_0 + P_1 + P_2 + P_{02} + x^5(P_4 + P_3 + P_{34})] + \\ & x^3[P_4 + P_{24} + P_{124} + a_0 * b_3 + a_3 * b_0 + P_{14} + x^5 P_4] + \\ & x^4[P_1 + P_2 + P_4 + P_{01} + P_{34} + a_0 * b_3 + a_3 * b_0 + P_{14} + P_{0134}]. \end{aligned}$$

We mark common subexpressions in the above formula as follows:

$$\begin{aligned} C = \; & x^0[P_0 + x^5(P_1 + (P_3 + P_4) + P_{02} + P_{023} + [a_0 * b_3 + a_3 * b_0 + P_{14}])] + \\ & x^1[\{P_0 + P_1 + x^5(P_3 + P_4)\} + x^5(P_2 + P_{24}) + P_{01}] + \quad (5) \\ & x^2[\{P_0 + P_1 + x^5(P_3 + P_4)\} + x^5 P_{34} + P_2 + P_{02}] + \quad (6) \\ & x^3[P_4 + P_{24} + P_{124} + [a_0 * b_3 + a_3 * b_0 + P_{14}] + x^5 P_4] + \\ & x^4[P_1 + P_2 + P_4 + P_{01} + P_{34} + [a_0 * b_3 + a_3 * b_0 + P_{14}] + P_{0134}]. \end{aligned}$$

There are $2*8*\frac{n}{5}$ XORs in products $P_{01}, P_{02}, P_{14}, P_{24}, P_{34}, P_{023}, P_{124}$ and P_{0134}. These products are polynomials in x^5 with the same degree $2*(\frac{n}{5}-1) = \frac{2n}{5} - 2$. We compute two "$\{\cdots\}$"s in Eq. (5) and (6) once, and save 1 shift-add. Shift-adds $x^5(P_2 + P_{24})$ in Eq. (5) and $x^5 P_{34}$ in Eq. (6) now become a normal addition. In summary, the 3 shift-add operations need $3*(\frac{2n}{5}-2)$ XOR gates, and the $30-1-2-4 = 23$ addition operations $23*(\frac{2n}{5}-1)$ XOR gates. Therefore, we have $16*\frac{n}{5} + 3*(\frac{2n}{5}-2) + 23*(\frac{2n}{5}-1) = \frac{68n}{5} - 29$ and

$$\mathcal{S}^{\oplus}(1) = 0, \qquad \mathcal{S}^{\oplus}(5) = 16 + 23 = 39,$$
$$\mathcal{S}^{\oplus}(n) = 15\mathcal{S}^{\oplus}(n/5) + \tfrac{68n}{5} - 29.$$

Another method to find a formula with $4T_X$ was presented in [8], i.e., for all $0 \le i < j \le 4$, we replace $a_i * b_j + a_j * b_i$ in the schoolbook formula by the identity $a_i * b_j + a_j * b_i = P_{ij} + P_i + P_j$. And obtain the following 15-multiplication $4T_X$ formula:

$$\begin{aligned} C = \; & x^0 * [P_0 + x^5 * (P_{14} + P_{23} + P_1 + P_2 + P_3 + P_4)] \\ + \; & x^1 * [\{P_0 + P_1 + x^5 * (P_3 + P_4)\} + P_{01} + x^5 * (P_{24} + P_2)] \\ + \; & x^2 * [\{P_0 + P_1 + x^5 * (P_3 + P_4)\} + P_{02} + P_2 + x^5 * P_{34}] \\ + \; & x^3 * [(P_{03} + P_{12} + P_0 + P_1 + P_2 + P_3) + x^5 * P_4] \\ + \; & x^4 * (P_{04} + P_{13} + P_0 + P_1 + P_2 + P_3 + P_4). \quad (7) \end{aligned}$$

There are $2*10*\frac{n}{5}$ XORs in products $P_{01}, P_{02}, P_{03}, P_{04}, P_{12}, P_{13}, P_{14}, P_{23}, P_{24}$ and P_{34}. These products are polynomials in x^5 with the same degree $2*(\frac{n}{5}-1) = \frac{2n}{5} - 2$. We compute two "$\{\cdots\}$"s once, and save 1 shift-add.

In summary, the 3 shift-add operations need $3*(\frac{2n}{5}-2)$ XOR gates, and the $26-3-3=20$ addition operations $20*(\frac{2n}{5}-1)$ XOR gates. Therefore, we have $20*\frac{n}{5}+3*(\frac{2n}{5}-2)+20*(\frac{2n}{5}-1)=\frac{66n}{5}-26$ and

$$\begin{array}{l}\mathcal{S}^{\oplus}(1)=0, \quad \mathcal{S}^{\oplus}(5)=20+20=40,\\ \mathcal{S}^{\oplus}(n)=15\mathcal{S}^{\oplus}(n/5)+\frac{66n}{5}-26.\end{array}$$

The number of XOR gates in this formula is less than that in Eq. (5) for $n>5$.

V. 6-TERM KARATSUBA FORMULAE

We transform the following 17-multiplication formula presented by Montgomery in [7]

$$\begin{aligned} C &= P_0+x*(P_{01}+P_0+P_1)+x^2*(P_{012}+P_{12}+P_{01})\\ &+ x^3*(P_{0235}+P_{025}+P_{345}+P_{23}+P_{12}+P_{34}+P_{45}+P_1+P_4)\\ &+ x^4*(P_{0235}+P_{025}+P_{345}+P_{23}+P_{14}+P_{0134}+P_{01}+P_{45}+P_1)\\ &+ x^5*(P_{0235}+P_{025}+P_{035}+P_{14}+P_1+P_4+P_5+P_0)\\ &+ x^6*(P_{0235}+P_{035}+P_{012}+P_{23}+P_{14}+P_{1245}+P_{01}+P_{45}+P_4)\\ &+ x^7*(P_{0235}+P_{035}+P_{012}+P_{23}+P_{12}+P_{34}+P_{01}+P_1+P_4)\\ &+ x^8*(P_{345}+P_{34}+P_{45})+x^9*(P_{45}+P_4+P_5)+x^{10}*P_5, \end{aligned}$$

and obtain the following $5T_X$ formula

$$\begin{aligned} C &= x^0*[P_0+x^6*(P_{0235}+P_{035}+P_{012}+P_{23}+P_{14}+P_{1245}+P_{01}+P_{45}+P_4)]\\ &+ x^1*[(P_{01}+P_0+P_1)+x^6*(P_{0235}+P_{035}+P_{012}+P_{23}+P_{12}+P_{34}+P_{01}+P_1+P_4)]\\ &+ x^2*[(P_{012}+P_{12}+P_{01})+x^6*(P_{345}+P_{34}+P_{45})]\\ &+ x^3*[(P_{0235}+P_{025}+P_{345}+P_{23}+P_{12}+P_{34}+P_{45}+P_1+P_4)+x^6*(P_{45}+P_4+P_5)]\\ &+ x^4*[(P_{0235}+P_{025}+P_{345}+P_{23}+P_{14}+P_{0134}+P_{01}+P_{45}+P_1)+x^6*P_5]\\ &+ x^5*(P_{0235}+P_{025}+P_{035}+P_{14}+P_1+P_4+P_5+P_0). \end{aligned} \quad (8)$$

In order to find a formula with $4T_X$, we consider the method in [8], i.e., for all $0\le i<j\le 5$, we replace $a_i*b_j+a_j*b_i$ in the schoolbook formula by the identity $a_i*b_j+a_j*b_i=P_{ij}+P_i+P_j$. And obtain the following 21-multiplication $4T_X$ formula:

$$\begin{aligned} C &= x^0*[P_0+x^6*(P_{15}+P_{24}+P_1+P_2+P_3+P_4+P_5)]\\ &+ x^1*[(P_{01}+P_0+P_1)+x^6*(P_{25}+P_{34}+P_2+P_3+P_4+P_5)]\\ &+ x^2*[(P_{02}+P_0+P_1+P_2)+x^6*(P_{35}+P_3+P_4+P_5)]\\ &+ x^3*[(P_{03}+P_{12}+P_0+P_1+P_2+P_3)+x^6*(P_{45}+P_4+P_5)]\\ &+ x^4*[(P_{04}+P_{13}+P_0+P_1+P_2+P_3+P_4)+x^6*P_5]\\ &+ x^5*[(P_{05}+P_{14}+P_{23}+P_0+P_1+P_2+P_3+P_4+P_5)]. \end{aligned} \quad (9)$$

REFERENCES

[1] H. Fan and J. Sun and M. Gu and K. Lam, “Overlap-free Karatsuba-Ofman Polynomial Multiplication Algorithms,” *IET Information Security*, vol. 4, no. 1, pp. 8-14, 2010. https://eprint.iacr.org/2007/393.

[2] M. Cenk, M. A. Hasan, and C. Negre, “Efficient subquadratic space complexity binary polynomial multipliers based on block recombination,” *IEEE Transactions on Computers*, vol. 63, no. 9, pp. 2273-2287, Sep. 2014.

[3] G. Zhou and H. Michalik, “Comments on ‘A New Architecture for a Parallel Finite Field Multiplier with Low Complexity Based on Composite Field’,” *IEEE Transactions on Computers*, vol. 59, no. 7, pp. 1007-1008, 2010.

[4] D. J. Bernstein, “Batch Binary Edwards,” *CRYPTO 2009*, LNCS 5677 pp. 317-336, 2009.

[5] M. G. Find and R. Peralta, “Better circuits for binary polynomial multiplication,” *IEEE Transactions on Computers*, vol. 68, no. 4, pp. 624-630, 2019.

[6] M. Cenk and F. Özbudak, “Improved polynomial multiplication formulas over $\mathbb{F}_2$ using Chinese Remainder Theorem,” *IEEE Transactions on Computers*, vol. 58, no. 4, pp. 572-576, 2009.

[7] P. L. Montgomery, “Five, Six, and Seven-Term Karatsuba-Like Formulae,” *IEEE Transactions on Computers*, vol. 54, no. 3, pp. 362-369, Mar. 2005.

[8] A. Weimerskirch and C. Paar, “Generalizations of the Karatsuba algorithm for efficient implementations”, https://eprint.iacr.org/2006/224.