Low Space Complexity Multiplication over Binary Fields with Dickson Polynomial Representation
Anwar Hasan, Christophe Negre

To cite this version:
Anwar Hasan, Christophe Negre. Low Space Complexity Multiplication over Binary Fields with Dickson Polynomial Representation. IEEE Transactions on Computers, Institute of Electrical and Electronics Engineers, 2011, 60 (4), pp.602-607. hal-00813621

HAL Id: hal-00813621
https://hal.archives-ouvertes.fr/hal-00813621
Submitted on 16 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract

We study Dickson bases for binary field representation. Such a representation seems interesting when no optimal normal basis exists for the field. We express the product of two field elements as Toeplitz or Hankel matrix-vector products. This provides a parallel multiplier which is subquadratic in space and logarithmic in time. Using the matrix-vector formulation of the field multiplication, we also present sequential multiplier structures with linear space complexity.

Index Terms

Binary field, Dickson basis, Toeplitz matrix, multiplier, parallel, sequential.

1 INTRODUCTION

Finite field arithmetic is extensively used in cryptography. For public key cryptosystems, the size (i.e. the number of elements) of the field may be quite large, say $2^{2048}$. Finite field multiplication over such a large field requires a considerable amount of resources (time or space). For binary extension fields, used in many practical public key cryptosystems, field elements can be represented with respect to a normal basis, where squaring operations are almost free of cost. In order to reduce the cost of multiplication over the extension field, instead of using an arbitrary normal basis, it is desirable to use an optimal normal basis. The latter however does not exist for all extension fields, in which case one may use Dickson bases [2], [8] and develop an efficient field multiplier.

In this paper first we consider subquadratic space complexity bit parallel multipliers using the Dickson basis. To this end, using low weight Dickson polynomials, we formulate the problem of field multiplication as a product of a Toeplitz or Hankel matrix and a vector, and apply subquadratic space complexity
algorithm for the product [3], which gives us a subquadratic space complexity field multiplier. Using
the matrix-vector product formulation, we then develop sequential multipliers. For such multipliers, we
consider both bit-serial and bit-parallel output formats.

The article is organized as follows. In Section 2 we present some general results on Dickson poly-
nomials. Then in Section 3 we give a matrix-vector product approach for field multiplication using the
Dickson basis representation. We use low weight Dickson polynomials and present parallel multipliers
of subquadratic space complexity. In Section 4, we develop sequential multipliers that have linear space
complexity. We wind up this article with a brief conclusion in Section 5.

## 2 Dickson Polynomials

Dickson polynomials over finite fields were introduced by L.E. Dickson in [2]. These polynomials have
several applications and interesting properties, the main one being a permutation property over finite
fields. For a complete explanation on this the reader may refer to [7]. Our interest here concerns the use
of Dickson polynomial for finite field representation for efficient binary field multiplication. There are
two kinds of Dickson polynomials, and there are several ways to define and construct both of them. We
give here the definition of [7] of the first kind Dickson polynomials.

**Definition 1:** [Dickson Polynomial[7] page 9] Let \( R \) be a ring and \( a \in R \). The Dickson polynomial of the first
kind \( D_n(X,a) \) is defined by

\[
D_n(X,a) = \sum_{i=0}^{\lfloor n/2 \rfloor} \frac{n}{n-i} \binom{n-i}{n} (-a)^i X^{n-2i}. \tag{1}
\]

For \( n = 0 \), we set \( D_0(X,a) = 2 \) and for \( n = 1 \) we have \( D_1(X,a) = X \).

In this paper, we will consider only \( \beta_i = D_i(X,1) \) the Dickson polynomials in \( \mathbb{F}_2[X] \).

**Theorem 1:** Let \( P \) be an irreducible polynomial of degree \( n \) in \( \mathbb{F}_2[X] \). The system \( B = \{ \beta_1, \ldots, \beta_n \} \) forms a
basis of \( \mathbb{F}_{2^n} = \mathbb{F}_2[X]/(P) \) over \( \mathbb{F}_2 \).

**Proof:** For a detailed proof we refer to [6], we just give a brief explanation here. Using (1), we can
see that a \( \beta_i = X^i + \text{terms of lower degree} \). This implies that the conversion matrix from \( \{ \beta_1, \ldots, \beta_n \} \) to
\( \{X, \ldots, X^n\} \), is lower triangular with 1 on the diagonal. The conversion matrix is thus invertible and
since \( \{X, \ldots, X^n\} \) form a basis of \( \mathbb{F}_{2^n} \), then \( B = \{ \beta_1, \ldots, \beta_n \} \) is a basis of \( \mathbb{F}_{2^n} = \mathbb{F}_2[X]/(P) \).

The following theorem will be extensively used for the construction of subquadratic multipliers in the
Dickson basis.

**Theorem 2:** We denote \( \beta_i = D_i(X,1) \) the \( n \)-th Dickson polynomial in \( \mathbb{F}_2[X] \). Then for all \( i, j \geq 0 \) the following
equation holds

\[
\beta_i \beta_j = \beta_{i+j} + \beta_{|i-j|}. \tag{2}
\]
Proof: We will show it by induction on \(i\) and \(j\). We can easily check that equation (2) holds for \(i, j \leq 1\). We suppose that the equation is true for all \(i, j \leq n\) and we prove that the equation is true for \(i, j \leq n+1\). We first prove it for \(i = n + 1\) and \(j \leq n\). We have

\[
\beta_{n+1} \beta_j = (X \beta_n + \beta_{n-1}) \beta_j \\
= X(\beta_{n+j} + \beta_{n-1+j}) + (\beta_{n-1+j} + \beta_{n-1-j}),
\]

by induction hypothesis. Now we have

\[
\beta_{n+1} \beta_j = (X \beta_{n+j} + \beta_{n-j-1}) + (X \beta_{n-j} + \beta_{n-1-j}) \\
= \beta_{n+1+j} + \beta_{n+1-j}.
\]

For the other case \(i = n + 1\) and \(j = n + 1\), the product \(\beta_{n+1} \beta_{n+1}\) is obtained using similar tricks.

\[\square\]

3 Field Multiplication Using Low Weight Dickson Polynomials

In this section we consider multiplication of two elements of the binary field \(\mathbb{F}_2^n = \mathbb{F}_2[X]/(P)\) where the polynomial \(P\) is a low weight Dickson polynomial. In particular we consider two and three-term Dickson polynomials \(P\), i.e., Dickson binomials and trinomials. Like low weight conventional polynomials the use of low weight Dickson polynomials is expected to yield lower space complexity multipliers.

In Table 1 we give the degree \(n \in [160, 285]\) of field \(\mathbb{F}_2^n = \mathbb{F}_2[X]/(P)\) where \(P\) is a low weight Dickson polynomial. Specifically, since no irreducible Dickson binomials were available, we have looked for Almost Dickson binomials (ADB) with irreducible \(P\) satisfying \(P \times (X + 1) = \beta_{n+1} + 1\). We also give Dickson trinomials (DT) of the form \(P = \beta_n + \beta_k + 1\) with \(k \leq n/2\). For the purpose of comparison, we mention also whether for each degree an ONB of type I or II exists (marked as NI and NII).

Our main goal here is to express the product of two elements, represented in the Dickson basis, as a Toeplitz or Hankel matrix-vector products. Recall that an \(n \times n\) Toeplitz matrix \(T = [t_{i,j}]\) satisfies \(t_{i,j} = t_{i+1,j+1}\) and a Hankel matrix \(H = [h_{i,j}]\) satisfies \(h_{i,j} = h_{i+1,j-1}\). We will then use the subquadratic Toeplitz matrix-vector product of [3] to design a subsquadratic multiplier.

3.1 Irreducible Dickson binomials

In this subsection we focus on finite fields \(\mathbb{F}_2^n = \mathbb{F}_2[X]/(P)\) where \(P\) is a two-term polynomial of the form \(P = \beta_n + 1\) where \(\beta_n\) is the \(n\)-th Dickson polynomial.

The elements of \(\mathbb{F}_2^n\) are expressed in the Dickson basis \(\mathcal{B} = \{\beta_1, \ldots, \beta_n\}\). Now, our main goal is to show that the product of two elements \(A\) and \(B\) in \(\mathbb{F}_2^n\) can be computed as a matrix-vector product \(M_A \cdot B\) where \(M_A\) is a sum of a Toeplitz matrix and an essentially Hankel matrix.
Table 1
Irreducible Dickson binomials and trinomials

<table>
<thead>
<tr>
<th>n</th>
<th>ADB=Dickson Binomial</th>
<th>DT=Dickson Trinomial</th>
<th>NI=ONBI</th>
<th>NII=ONBII</th>
</tr>
</thead>
<tbody>
<tr>
<td>163</td>
<td>DT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>164</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>165</td>
<td>DT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>166</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>167</td>
<td>ADB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>168</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>169</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>DT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>171</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>173</td>
<td>ADB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>174</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>175</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>176</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>177</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>178</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>179</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>180</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>181</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>182</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>183</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>184</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>185</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>186</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>187</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>188</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>189</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>190</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>191</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>192</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>193</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>194</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If we multiply two elements $A$ and $B$ expressed in $B$ and if we use Theorem 2 we get the following

$$AB = \left( \sum_{i=1}^{n} a_i \beta_i \right) \times \left( \sum_{i=j}^{n} b_j \beta_j \right)$$

$$= \left( \sum_{i,j=1}^{n} a_i b_j \beta_{i+j} \right) + \left( \sum_{i,j=1}^{n} a_i b_j \beta_{|i-j|} \right)$$

Now we express each sum $S_1$ and $S_2$ as matrix-vector products. Let us begin with $S_1$. We remark that $S_1$ has a similar expression as product of two polynomials of the same degree. In other words, $S_1$ can
be computed as $Z_A \cdot B$ where

\[
Z_A = \begin{bmatrix}
0 & 0 & \cdots & 0 & 0 \\
0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
a_{n-1} & \cdots & a_1 & 0 \\
a_n & \cdots & a_2 & a_1 \\
0 & a_n & \cdots & a_3 & a_2 \\
\vdots & \vdots & \ddots & \ddots & \ddots \\
0 & 0 & \cdots & 0 & a_n
\end{bmatrix} \leftarrow \beta_1
\]

(4)

We reduce the matrix $Z_A$ modulo $P = \beta_n + 1$ to get non-zero coefficients only on rows corresponding to $\beta_1, \ldots, \beta_n$. We use the fact that $\beta_{n+i}$ for $i \geq 0$ satisfies $\beta_{n+i} = \beta_i \beta_n + \beta_{n-i}$. This equation is a simple consequence of (2) and that $\beta_n = 1 \mod P$. This implies that the rows corresponding to $\beta_{n+i}$ are reduced into two rows one corresponding to $\beta_i$ and the other to $\beta_{n-i}$. After performing this reduction and removing zero rows we get

\[
S_1 = Z_A \cdot B = \begin{bmatrix}
a_n & a_{n-1} & \cdots & a_2 & a_1 \\
a_1 & a_n & \cdots & a_3 & a_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
a_{n-1} & \cdots & a_1 & a_n
\end{bmatrix} \begin{bmatrix}
b_1 \\
\vdots \\
\vdots \\
b_n
\end{bmatrix}
\]

\[
S_1,1 + S_1,2
\]

Finally, we get an expression of $S_1$ as matrix-vector product where the matrix is a sum of a Toeplitz and an essentially Hankel matrix.

Now we do the same for $S_2$. We split $S_2$ into two sums

\[
S_2 = \left( \sum_{k=1}^{n} \sum_{j=1}^{n-k} a_{j+k} b_j \beta_k \right) + \left( \sum_{k=1}^{n} \sum_{j=k}^{n} a_{j-k} b_j \beta_k \right).
\]

(5)
We express $S_{2,1}$ and $S_{2,2}$ as matrix-vector products

$$S_{2,1} = \begin{bmatrix}
  a_2 & a_3 & \cdots & a_{n-1} & a_n & 0 \\
  a_3 & a_4 & \cdots & a_n & 0 & 0 \\
  \vdots & \vdots & & \vdots & & \vdots \\
  a_n & 0 & \cdots & 0 & 0 & 0
\end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix},$$

(6)

$$S_{2,2} = \begin{bmatrix}
  0 & a_1 & a_2 & \cdots & a_{n-1} \\
  0 & 0 & a_1 & \cdots & a_{n-2} \\
  \vdots & \vdots & \vdots & & \vdots \\
  0 & 0 & \cdots & a_1 & 0 \\
  0 & \cdots & \cdots & 0 & \cdots & 0
\end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix},$$

(7)

So now we have each of $S_1$ and $S_2$ in the required form. We finally write $S_{1,1} + S_{2,2} = T_A \cdot B$ where $T_A$ is a Toeplitz matrix and $S_{1,2} + S_{2,1} = H_A \cdot B$ where $H_A$ is an Hankel matrix. We obtain

$$A \times B = T_A \cdot B + H_A \cdot B$$

(8)

as stated at the beginning of the current subsection.

### 3.2 Dickson trinomials

Now we assume that the field $\mathbb{F}_{2^n}$ is defined by a three-term irreducible Dickson trinomial $P$

$$P = 1 + \beta_k + \beta_n, \text{ with } k \leq n/2.$$  

The elements in $\mathbb{F}_{2^n} = \mathbb{F}_2[X]/(P)$ are expressed in the Dickson basis $B = \{\beta_1, \ldots, \beta_n\}$. Our aim is to express the product of two elements $A$ and $B$ of $\mathbb{F}_{2^n}$ as Toeplitz or Hankel matrix-vector product. We use again the expression of the product $C = A \times B = S_1 + S_2$ given in equation (3). Similar to the previous subsection, here we express $S_1$ and $S_2$ as matrix-vector product separately. Specifically

1) The sum $S_1$ is expressed as $Z_A \cdot B$ where $Z_A$ is given in (4)
2) For $S_2$ we use the expression of (5) and we put this expression in a matrix-vector product form.

$$S_2 = \begin{bmatrix}
  a_2 & a_3 & \cdots & a_n & 0 \\
  a_3 & a_4 & \cdots & 0 & 0 \\
  \vdots & \vdots & \vdots & \vdots & \vdots \\
  a_n & 0 & \cdots & 0 & 0 \\
  0 & 0 & \cdots & 0 & 0
\end{bmatrix} + \begin{bmatrix}
  0 & a_1 & \cdots & a_{n-1} \\
  0 & 0 & \cdots & a_{n-2} \\
  \vdots & \vdots & & \vdots \\
  0 & 0 & \cdots & a_1 \\
  0 & \cdots & \cdots & 0 & \cdots & 0
\end{bmatrix} \cdot B.$$  

(9)
Now we replace $S_1$ and $S_2$ by their corresponding expressions given above in $A \times B = S_1 + S_2$. We get

$$
\begin{pmatrix}
0 & 0 & \cdots & 0 \\
a_1 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
a_{n-1} & \cdots & \cdots & 0 \\
a_n & \cdots & \cdots & a_2 \\
0 & a_n & \cdots & a_2 \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \cdots & a_n \\
\end{pmatrix} +
\begin{pmatrix}
0 & a_1 & \cdots & a_{n-1} \\
0 & 0 & \cdots & a_{n-2} \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \cdots & a_1 \\
0 & \cdots & \cdots & 0 \\
0 & \cdots & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \cdots & 0 \\
\end{pmatrix}
\cdot B
$$

In (10) the addition of two $2n \times n$ Toeplitz matrices results in one single $2n \times n$ Toeplitz matrix. The latter can be horizontally split in the middle to obtain two $n \times n$ Toeplitz matrices, say $T_{up}$ and $T_{down}$, which can be then multiplied separately with vector $(b_1, \ldots, b_n)$ with a total cost of two $n \times n$ Toeplitz matrix-vector products.

The other $2n \times n$ Hankel matrix in (10) has all zero in the lower $n$ rows, contributing nothing to the cost of the matrix-vector multiplication. Thus, the total computational cost of (10) is no more than three $n \times n$ Toeplitz or Hankel matrix-vector products.

The reduction:

The resulting expression of $C$ in (10) is an unreduced form of $A \times B$, since it has non zero coefficients $c_i$ on rows $i = n+1, \ldots, 2n$. These coefficients are obtained by multiplying $T_{down}$ with vector $(b_1, b_2, \ldots, b_n)$, and must be reduced modulo $P = \beta_n + \beta_k + 1$, to get an expression of $C$ in $B$. We have

$$
\beta_i = \beta_n \beta_{i-n} + \beta_{2n-i} = (\beta_k + 1)\beta_{i-n} + \beta_{2n-i} = \beta_{i-n+k} + \beta_{|i-n-k|} + \beta_{i-n} + \beta_{2n-i}.
$$

We reduce the expressing of $C = \sum_{i=1}^{2n} c_i \beta_i$ by replacing each $\beta_i$ for $i > n$ by the expression given above. Since we assume $k < \frac{n}{2}$ this process must be done two times to get a reduced expression of $C$. A circuit can be designed to perform this process which requires $6n - k$ XOR gates and is performed in time $3T_X$ (see [6] for details).

### 3.3 Parallel multiplier

We can design multiplier using the expression of the multiplication in $F_{2^n}$ as a Toeplitz or Hankel matrix-vector product (TMVP). Specifically we use the Toeplitz or Hankel matrix-vector multiplier presented
in [3] to perform these products. In Table 2, we recall the complexity of the TMVP multiplier established by Fan and Hasan [3].

Table 2
Asymptotic complexity of TMVP multiplier

<table>
<thead>
<tr>
<th></th>
<th>2-way split method</th>
<th>3-way split method</th>
</tr>
</thead>
<tbody>
<tr>
<td># AND</td>
<td>$n \log_2(3)$</td>
<td>$n \log_3(6)$</td>
</tr>
<tr>
<td># XOR</td>
<td>$5.5n \log_2(3) - 6n + 0.5$</td>
<td>$\frac{24}{5}n \log_3(6) - 5n + \frac{1}{5}$</td>
</tr>
<tr>
<td>Delay</td>
<td>$T_A + 2 \log_2(n)T_X$</td>
<td>$T_A + 3 \log_3(n)T_X$</td>
</tr>
</tbody>
</table>

In the case of Dickson binomials, to compute the matrix-vector products of (8) we need two TMVP multipliers in parallel. Each of them can use 2-way or 3-way split approach of [3]. We also need additional 2n XOR gates to compute the coefficient of $T_A$ and add the result of the two matrix-vector products.

In the case Dickson trinomials, as specified in Subsection 3.2, three TMVPs are done in parallel using 2-way or 3-way split approach of [3]. We also need to perform a reduction using the circuit depicted in [6]. We obtain the complexities of Table 3 below where the second left most column indicates $b$-way splits with the value of $b$ being either 2 or 3.

Table 3
Comparison of Subquadratic Space Complexity Parallel Multipliers

<table>
<thead>
<tr>
<th>$b$</th>
<th># AND</th>
<th># XOR</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB</td>
<td>$2n \log_2(3)$</td>
<td>$11n \log_2(3) - 11n$</td>
<td>$(21 \log_2(n) + 1)T_X + T_A$</td>
</tr>
<tr>
<td></td>
<td>$3n \log_3(6)$</td>
<td>$48/5n \log_3(6)$</td>
<td>$(3 \log_3(n) + 1)T_X + T_A$</td>
</tr>
<tr>
<td></td>
<td>$7/3 \cdot 2n \log_2(3)$</td>
<td>$\frac{38}{5}n \log_2(3)$</td>
<td>$(2 \log_2(n) + 6)T_X + T_A$</td>
</tr>
<tr>
<td></td>
<td>$3n \log_3(6)$</td>
<td>$48/5n \log_3(6)$</td>
<td>$(3 \log_3(n) + 5)T_X + T_A$</td>
</tr>
<tr>
<td>DT</td>
<td>$2n \log_2(3) + n$</td>
<td>$5.5n \log_2(3) - 4n - 0.5$</td>
<td>$(21 \log_2(n) + 1)T_X + T_A$</td>
</tr>
<tr>
<td></td>
<td>$3n \log_3(6) + n$</td>
<td>$24/5n \log_3(6) - 3n - 4/5$</td>
<td>$(3 \log_3(n) + 1)T_X + T_A$</td>
</tr>
<tr>
<td>ONBI</td>
<td>$n \log_2(3)$</td>
<td>$6n \log_2(3) - 3n + (8n + 2) \log_2(2n + 1)$</td>
<td>$(3 \log_2(n) + 1)T_X + T_A$</td>
</tr>
<tr>
<td>[4]</td>
<td>$n \log_3(6)$</td>
<td>$\frac{72}{15}n \log_3(6) - \frac{1}{5}n - 1 + (8n + 2) \log_2(2n + 1)$</td>
<td>$(4 \log_3(n) + 1)T_X + T_A$</td>
</tr>
</tbody>
</table>

The row of Table 3 labelled by † (resp. ‡) refers to the method of [5] combined to the polynomial multiplication of [9] (resp. [11]).
In a recent paper Mullin et al. [8] pointed out that there were some links between the Dickson basis and the normal basis. In practice, a Dickson basis is interesting when no optimal normal basis exists for the considered field.

This is the case for NIST recommended binary fields $\mathbb{F}_{2^{163}}$ and $\mathbb{F}_{2^{283}}$. Using Table 1, we can remark that NIST fields can be constructed with Dickson trinomials, and thus we obtain a subquadratic multiplier in each of these cases.

4 **Sequential Multipliers**

In this section, we present sequential multipliers. Each of these multipliers takes $O(n)$ clock cycles but has a space complexity of $O(n)$.

4.1 **Using irreducible Dickson binomials**

4.1.1 **Multiplier with bit serial output**

In the sequel, we denote the entry at $(i, j)$ of the Toeplitz and the Hankel matrices of (8) as $T_{i,j}$ and $H_{i,j}$, respectively. We also denote the rows of the Toeplitz matrix as $T_{1,*}, T_{2,*}, \ldots, T_{n,*}$ and those of the essentially Hankel matrix as $H_{1,*}, H_{2,*}, \ldots, H_{n,*}$. Thus we can write

\[
A \times B = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} T_{1,*} + H_{1,*} \\ T_{2,*} + H_{2,*} \\ \vdots \\ T_{n,*} + H_{n,*} \end{bmatrix} \cdot B. \tag{11}
\]

We remark that

a) $T_{n,*}$ consists of the coordinates of input $A$ that are rotated left one position, i.e., $T_{n,*} = \begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & a_n \end{bmatrix}$. On the other hand, $H_{n,*}$ is the all zero row vector and $H_{n-1,*} = \begin{bmatrix} 0 & a_{n-1} & a_{n-2} & \cdots & a_2 & a_1 \end{bmatrix}$.

b) Given $T_{i,*}$ and $H_{i,n-1,*}$, we can express $T_{i-1,*}$ for $1 < i \leq n$ as $T_{i-1,*} = \begin{bmatrix} T_{i,2} & T_{i,3} & \cdots & T_{i,n} & T_{i,1} + H_{i,n-1} \end{bmatrix}$.

Furthermore, given the row $H_{i,*}$ and the entry $T_{i+1,1}$ we can express $H_{i-1,*}$ for $1 < i \leq n - 1$ as follows

\[
H_{i-1,*} = \begin{bmatrix} T_{i+1,1} & H_{i,1} & \cdots & H_{i,n-2} & H_{i,n-1} \end{bmatrix}.
\]

The following diagram (Figure 1) corresponds to a sequential structure to realize the multiplication $C = A \times B$ in accordance with (11). In the initial clock cycle, the left side register (LR) in the diagram is loaded with $T_{n,*}$ and the right side register (RR) with $H_{n,*}$. In this cycle, rows $T_{n,*}$ and $H_{n,*}$ are added and an inner product is performed to yield $c_n = (T_{n,*} + H_{n,*}) \cdot B$. Also, in this cycle the output of MUX is $a_1$ (and in other cycles the MUX output is the second right most bit of RR). In the next cycle, RR is loaded with $H_{n-1,*}$ and LR is shifted left to generate $T_{n-1,*}$ eventually yielding $c_{n-1}$.

For each of the following $n - 2$ clock cycles, LR is shifted left, RR is shifted right, their contents are bit-wise added and an inner product is performed to produce one coordinate of $C$. The space and time complexity of the architecture of Fig. 1 is given in Table 4.
Figure 1. Sequential multiplier with bit serial out using Dickson binomials

\[ c_0, c_2, \ldots, c_n \]

binary tree of \( n-1 \) XOR gates

\[ b_0, b_1, \ldots, b_{n-1} \]

\[ a_1, a_2, \ldots, a_n \]

\[ b_i \]

4.1.2 Sequential multiplier with bit parallel output

Referring to (8) we denote the columns of a Toeplitz matrix as \( T_{*,i} \) for \( 1 \leq i \leq n \) and those of an essentially Hankel matrix as \( H_{*,i} \) for \( 1 \leq i \leq n \). Thus we can write \( A \times B = \left[ T_{*,1} + H_{*,1} \quad T_{*,2} + H_{*,2} \quad \cdots \quad T_{*,n} + H_{*,n} \right] \).

\[ C = \sum_{i=1}^{n} b_i (T_{*,i} + H_{*,i}). \quad (12) \]

We remark that

a) \( T_{*,1} = \left[ a_n \quad a_1 \quad \cdots \quad a_{n-1} \right]^t \) and \( H_{*,1} = \left[ a_2 \quad a_3 \quad \cdots \quad a_{n-1} \quad 0 \quad 0 \right]^t \)

b) Given the column \( T_{*,i} \) and the entry \( H_{1,i-1} \), we can express \( T_{*,i+1} \) as \( T_{*,i+1} = \left[ T_{n,i} + H_{1,i-1}, T_{1,i}, \cdots, T_{n-1,i} \right]^t, 1 \leq i \leq n \), where \( H_{1,0} \) is assumed to be \( a_1 \).

Additionally, given column \( H_{*,i} \) and entry \( T_{n,i} \), we can express \( H_{*,i+1} \) as \( H_{*,i+1} = \left[ H_{2,i}, H_{3,i}, \cdots, H_{n-1,i}, T_{n,i}, 0 \right]^t, 1 \leq i \leq n \).

In the following diagram (Fig. 2), the column vectors \( T_{*,1} \) and \( H_{*,1} \) are initially loaded into the top register (TR) and the bottom register (BR) respectively. The one-bit feedback cell F is initialized with \( H_{1,0} = a_1 \). If TR is shifted downward and BR upward with the feedback connections as shown in the diagram, the new contents of TR and BR will be \( T_{*,2} \) and \( H_{*,2} \) respectively. Note that BR is an \( n-1 \) bits long shift register, since \( H_{n,i} = 0 \) for \( 1 \leq i \leq n \). With additional shifts on TR and BR, the remaining columns of the Toeplitz and the essentially Hankel matrices are generated.

Each corresponding pair of columns (i.e., \( T_{*,i} \) and \( H_{*,i} \)) are added and the resulting columns are multiplied with \( b_i \) (in the diagram these are shown using an array of XOR and AND gates).

The weighted columns are accumulated in accordance with (12) to produce the desired output \( C \) in a total of \( n \) clock cycles. The delay and the space complexity of this architecture are given in Table 4.
4.2 Using irreducible Dickson trinomials

From (10) of Subsection 3.2 the coefficients $c_1, c_2, \ldots, c_n$ are given by

$$
\begin{bmatrix}
0 & a_1 & \cdots & a_{n-1} \\
a_1 & 0 & \cdots & a_{n-2} \\
\vdots & \ddots & \ddots & \vdots \\
a_{n-2} & a_{n-3} & \cdots & a_1 \\
a_{n-1} & a_{n-2} & \cdots & 0
\end{bmatrix}
+ \begin{bmatrix}
a_2 & a_3 & \cdots & a_n & 0 \\
a_3 & a_4 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
a_n & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & 0 & 0
\end{bmatrix}
\cdot B, \tag{13}
$$

and

$$
\begin{bmatrix}
c_{n+1} \\
c_{n+2} \\
\vdots \\
c_{2n}
\end{bmatrix}
= \begin{bmatrix}
a_n & a_{n-1} & \cdots & a_1 \\
0 & a_n & \cdots & a_2 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & a_n
\end{bmatrix}
\cdot \begin{bmatrix}
b_1 \\
b_2 \\
\vdots \\
b_n
\end{bmatrix}. \tag{14}
$$

Note that $c_{n+1}, c_{n+2}, \ldots, c_{2n}$ can be reduced as explained at the end of Subsection 3.2.

Below we will first present a hardware structure to generate $c_1, c_2, \ldots, c_n$ in accordance with (13). Then we will discuss how to use part of the above hardware to generate $c_{n+1}, c_{n+2}, \ldots, c_{2n}$. In practice, one can first generate $c_{n+1}, c_{n+2}, \ldots, c_{2n}$. While these $n$ bits are reduced, one can generate $c_1, c_2, \ldots, c_n$. This overlap of operations will effectively eliminate/hide the extra time for reduction of $c_{n+1}, \ldots, c_{2n}$.

4.2.1 Sequential multiplier with bit serial output

We denote the rows of the Toeplitz and the Hankel matrices of (13) as $T_{i,*}$ and $H_{i,*}$ respectively. For $1 \leq i \leq n$, we can then write the following

$$
T_{i+1,*} = \begin{bmatrix}
H_{i-1,1} & T_{i,1} & \cdots & T_{i,n-2} & T_{i,n-1}
\end{bmatrix},
$$
$$
H_{i+1,*} = \begin{bmatrix}
H_{i,2} & H_{i,3} & \cdots & H_{i,n} & 0
\end{bmatrix}. \tag{15}
$$
where $H_{0,1}$ is assumed to be $a_1$ and
\[
T_{1,*} = \begin{bmatrix}
  0 & a_1 & a_2 & \cdots & a_{n-2} & a_{n-1} \\
  a_2 & a_3 & a_4 & \cdots & a_n & 0
\end{bmatrix},
\]
\[
H_{1,*} = \begin{bmatrix}
  a_2 & a_3 & a_4 & \cdots & a_n & 0
\end{bmatrix}.
\] (16)

In Fig. 3 below, registers RR and LR are initialized with $T_{1,*}$ and $H_{1,*}$. The feedback cell $F$ is initialized with $a_1 = H_{0,1}$. Then with the application of a shift to these registers together, the second rows of the Toeplitz and the Hankel matrices of (12) are formed in RR and LR respectively. This happens due to the fact that the shift and the feedback connection as shown in Fig. 3 essentially realize (15). The remaining rows of the two matrices are formed pair by pair with successive shifts.

Note that LR is $n - 1$ bits long, since the right most bit of each row of the Hankel matrix is zero. The upper part of Fig. 3 is similar to that of Fig. 1 and is to add the corresponding rows of the Toeplitz and the Hankel matrices, followed by inner product operations to yield $c_1, c_2, \ldots, c_n$.

To generate $c_{n+1}, c_{n+2}, \ldots, c_{2n}$ using the structure in Fig. 3, we initialize RR with $[a_n, a_{n-1}, \ldots, a_1]$, which is the first row of the upper triangular Toeplitz matrix of (14). Register LR and cell F are initialized with all zeros. Then with successive shifts, RR will contain the remaining rows of the Toeplitz matrix and LR will have all zeros. This will result in $c_{n+1}, c_{n+2}, \ldots, c_2$ at the output of Fig. 3.

The time and the space complexities of the structure of Fig. 3 are given in Table 4. These exclude the cost associated with the reduction of $c_{n+1}, c_{n+2}, \ldots, c_{2n}$.

---

**Figure 3.** Bit serial output sequential multiplier for Dickson trinomials

![Diagram of the multiplier](image)

**4.2.2 Bit parallel output**

For (13), let $T_{*,i}$ and $H_{*,i}$ be the $i$-th columns of the Toeplitz and the Hankel matrices, respectively. Then (13) can be re-written as
\[
\begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix}^t = \sum_{i=1}^{n} b_i (T_{*,i} + H_{*,i}).
\]
In order to generate \( T_{*,i} \) and \( H_{*,i} \) we note that in (13) \( T_{*,i} = T_{i,*} \) and \( H_{*,i} = H_{i,*} \). In other words, the \( i \)-th column is the same as the \( i \)-th row for each of the matrices. Thus columns can be generated using the same system of feedback registers as shown in Fig. 3 earlier.

To obtain \( c_1, c_2, \cdots, c_n \) in bit parallel fashion the inner product unit of Fig. 3 can be replaced with an array unit of weighting AND gates and accumulators (as used in Fig. 2). The complete diagram is shown in Fig. 4, and its space and time complexities are given in Table 4.

**Figure 4. Bit parallel output sequential multiplier for Dickson trinomials**

The coefficients \( c_{n+1}, c_{n+2}, \cdots, c_{2n} \) from (14) will be computed with the same hardware of Fig. 2. Specifically, in Fig. 4 RR is initialized with the first column of the above matrix. The accumulators LR and F are all initialized to zero. Then in \( n \) clock cycles with weighting input as \( b_n, b_{n-1}, \ldots, b_1 \) the accumulators will have \( c_{2n}, c_{2n-1}, \ldots, c_{n+1} \).

### 4.3 Complexity and comparison

In Table 4 we put the resulting complexities of the different sequential multipliers based on the Dickson basis representation. For the purpose of comparison, we also give the complexity of the method of [4] using ONB of type I and II. We remark that when no ONB is available, a Dickson binomial seems to be the best choice since Dickson trinomials require an increased number of clock cycles.

### 5 Conclusion

In this paper we have presented new parallel multipliers based on Dickson basis representation of binary fields. The multiplier for an irreducible Dickson binomial has a complexity similar to the subquadratic multiplier for ONB II of [4]. For an irreducible Dickson trinomial, the multiplier has a slightly more space complexity, but can still be used for fields with degree of several hundreds (for example those used in today’s elliptic curve cryptographic systems).
Table 4

Complexity of sequential multipliers

<table>
<thead>
<tr>
<th>Archi.</th>
<th>#AND</th>
<th>#XOR</th>
<th>#FF</th>
<th>#MUX</th>
<th>#CC</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Fig 1</td>
<td>n</td>
<td>2n</td>
<td>2n+1</td>
<td>1</td>
<td>n</td>
<td>$T_A + (1 + \lceil \log_2(n) \rceil)T_X$</td>
</tr>
<tr>
<td>DB Fig 2</td>
<td>n</td>
<td>2n</td>
<td>2n+1</td>
<td>1</td>
<td>n</td>
<td>$T_A + 2T_X$</td>
</tr>
<tr>
<td>DT Fig 3</td>
<td>n</td>
<td>2n-2</td>
<td>2n</td>
<td>0</td>
<td>2n</td>
<td>$T_A + (1 + \lceil \log_2(n) \rceil)T_X$</td>
</tr>
<tr>
<td>DT Fig 4</td>
<td>n</td>
<td>2n-1</td>
<td>3n-1</td>
<td>0</td>
<td>2n</td>
<td>$T_A + T_X$</td>
</tr>
<tr>
<td>DG[1]</td>
<td>2n</td>
<td>4n-3</td>
<td>3n</td>
<td>0</td>
<td>n</td>
<td>$2T_X + T_A$</td>
</tr>
<tr>
<td>ONBI[10]</td>
<td>n</td>
<td>$\frac{3n}{2}$</td>
<td>2n</td>
<td>0</td>
<td>n</td>
<td>$T_A + 2T_X$</td>
</tr>
<tr>
<td>ONBI[12]</td>
<td>n</td>
<td>$\frac{3n-1}{2}$</td>
<td>3n</td>
<td>0</td>
<td>n</td>
<td>$T_A + 2T_X$</td>
</tr>
</tbody>
</table>

DB=Dickson Binomial, DT=Dickson Trin., DG=General Dickson, CC=Clock Cycle.

In this paper, we have also presented sequential multipliers using the above mentioned Dickson representation. The sequential multipliers have a space complexity of $O(n)$. We have considered both bit-serial and bit-parallel output formats for the sequential multipliers. Compared to the sequential multipliers with bit-parallel output format presented in [1] and [8], the sequential multipliers presented here with the same output format reduce the number of XOR and AND gates by a factor of two or more, while keeping the number of flip-flops and clock cycles about the same.

ACKNOWLEDGMENT

A preliminary version of this work was presented at the WAIFI 2008 conference [6]. This work was supported in part by an NSERC research grant awarded to Dr. Hasan.

REFERENCES


