社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－ＳＬＤＭ－128（５）
2007／1／17

楕円曲線暗号向け$GF(2^m)$上のDigit-Serial乗算器の設計

奈良 竜太† 小原 俊逸† 清水 一範†† 戸川 望† 池永 剛††
柳澤 政生† 後藤 敏†† 大附 辰夫†

† 早稲田大学理工学部コンピュータ・ネットワーク工学科
〒 169-8555 東京都新宿区大久保 3-4-1
Tel: 03-3209-3211(5716), Fax: 03-3204-4875
†† 早稲田大学大学院情報生産システム研究科
〒 808-0135 福岡県北九州市若松区ひびきの 2-7
Tel: 093-692-5017, Fax:093-692-5021
E-mail: †nara@yanagi.comm.waseda.ac.jp

あらまし $GF(2^m)$におけるdigit-serial乗算器とは，bit-serial乗算を拡張し，複数のビットを同時に処理することで1サイクルあたりの処理量を増やした手法である．本稿では，MSB(most significant bit)乗算器をベースに，digit-serial乗算器の一つであるMSD(most significant digit)乗算器を提案する．本手法はMSB乗算器をdigitサイズDだけ直列に接続することでMSD乗算器を実装できるため，従来の手法より設計が容易になり，面積を小さくできる．さらに1回の乗算に必要なクロックサイクル数を抑えることができる．提案手法による乗算器を用いた楕円曲線暗号処理回路をROHM0.35μmテクノロジで実装した結果，$GF(2^{163})$における楕円暗号処理を50MHz動作時に約0.115msで処理することができた．

キーワード $GF(2^m)$，digit-serial乗算器，MSB (most significant bit)乗算器， MSD (most significant digit)乗算器，楕円曲線暗号，公開鍵暗号

$GF(2^m)$ Digit-Serial Multiplier for Elliptic Curve Cryptosystem

Ryuta NARA†, Shunitsu KOHARA†, Kazunori SHIMIZU††, Nozomu TOGAWA†, Takeshi IKENAGA††, Masao YANAGISAWA†, Satoshi GOTO††, and Tatsuo OHTSUKI†

† Dept. of Computer Science, Waseda University
3-4-1 Okubo, Shinjuku, Tokyo 169-8555, Japan
Tel: +81-3-3209-3211(5716), Fax: +81-3-3204-4875
†† Grad. School of IPS, Waseda University
2-7 Hibikino, Wakamatsu, Kitakyushu, Fukuoka 808-0135, Japan
Tel: +81-93-692-5017, Fax: +81-93-692-5021
E-mail: †nara@yanagi.comm.waseda.ac.jp

Abstract Digit serial multiplier for $GF(2^m)$ is an architecture that increases throughput at one cycle by extending multiplicand bits of a bit serial multiplier. In this paper, we propose an MSD(most significant digit) multiplier, which is one of the digit serial multiplier, based on an MSB(most significant bit) multiplier. By connecting D(digit size) pieces of MSB multipliers in series, our implementation is simpler, lower area and less clock-cycles than traditional methods. Implementing elliptic curve cryptosystem (ECC) using the proposal multiplier with ROHM 0.35μm technology, we achieved operation times of 0.115ms for EC scalar multiplication in $GF(2^{163})$ at 50HMz.

Key words $GF(2^m)$, digit-serial multiplier, most significant bit (MSB) multiplier , most significant digit (MSD) multiplier, elliptic curve cryptosystem, public key cryptosystem

1. まえがき

楕円曲線暗号[4],[7]とは公開鍵暗号の一種で，現在最も普及している公開鍵暗号方式であるRSA暗号[8]よりも高い安全性を有している．公開鍵暗号は安全性は鍵長に依存するが，楕円曲線暗号の場合，鍵長160ビットでの安全性がRSA暗号の鍵長1024ビットの安全性と同等とされている．そのため楕円曲線暗号はRSA暗号よりも扱うデータ量が減り，高速化，低面積化などが期待できるため，RSA暗号では実装できないリソースの限られた用途に適しているとされている．

楕円曲線暗号は有限体の演算で構成されており，主に素体$GF(p)$と2の拡大体$GF(2^m)$がある．$GF(2^m)$は加算がXORで桁上がりがなく，乗算はシフトと加算で演算ができるため，$GF(p)$よりも面積や実行時間などを小さくできる．そのためハードウェアで実装する際には$GF(2^m)$を選択する場合が多い．楕円曲線暗号の演算量はほとんどが有限体の計算であり，特に乗算は演算量やハードウェアの大部分を占めている．従って楕円曲線暗号の高速化や低面積化には乗算器の改善が有効である．$GF(2^m)$での乗算器の実装方法には暗号回路用に限らず様々な手法が提案されている．例えば，bit-serial型やmや有限体の法$f(z)$などのパラメータを任意に変えられるプロセッサ型[9]などがある．前者は乗算の基本的なアルゴリズムに沿っており，回路が単純で容易に実装できるが，1サイクル事に処理できる量がmビットで決まっているため処理速度が遅く，また拡張性が低い．後者は汎用性が高く，動作周波数が高いため高速な処理が可能であるが，1サイクルあたりの処理量が小さく，高速処理を行うには動作周波数を大きくしなくてはならず，消費電力の面で不利である．

digit-serial乗算はbit-serial乗算を拡張し，複数のビットを同時に処理することで1サイクルあたり処理量を増やしたアーキテクチャである．digitサイズDを変えることにより処理量と面積のトレードオフを選択することができるため，bit-serial型よりも拡張性が高い．また，プロセッサ型と比べ1サイクルあたりの処理量が大きいため，低い動作周波数でも同等の処理性能が維持できる．そのため低消費電力動作に有効である[10]．また，乗算部を複数並列にすることにより，遅延を減らすことで高速化する手法[5]が提案されている．しかし，乗算部とRedution部が分かれており，どちらもDに依存するためbit-serial型よりも回路が複雑になる．また，1回の乗算に必要なクロックサイクル数が大きくなる．

そこで本稿ではbit-serialアーキテクチャの一つであるmost significant bit first (MSB)乗算器をベースにしたmost significant digit first (MSD)乗算器を提案する．MSB乗算器をD個を直列に接続することでMSD乗算器を実装している．MSB乗算器のように1ビットシフトするごとにReduction演算を行ない，その出力を次のMSB乗算器に伝えることをDだけ繰り返すことで処理を行う．そのためデータ長はDに依存せず常にmビットにすることができ，面積の増加率を抑えることができる．

本稿は以下のように構成される．2章ではdigit-serial乗算の概要を説明する．3章ではMSB乗算器をベースにしたMSD乗算器を提案し，面積と遅延の見積もる．4章では論理合成結果を示す．5章では提案する乗算器を載せた楕円曲線暗号処理回路をROHM0.35μmプロセスにてチップ実装したので，その結果について示す．

Input: $A = \sum_{i=0}^{m-1} a_i z^i, a_i \in GF(2)$
Input: $B = \sum_{i=0}^{m-1} b_i z^i, b_i \in GF(2)$
Output: $S = A \cdot B \bmod f(z)$
$f(z) = z^m + \sum_{i=0}^{m-1} c_i z^i,\ c_i \in GF(2)$
1: $S \leftarrow 0$
2: for $i = m-1$ to 0 do
  2.1: $S \leftarrow A \cdot B_i + S$
  2.2: $S \leftarrow S \cdot z \bmod f(z)$
3: Return(S)

図1 MSBアルゴリズム．

Input: $A = \sum_{i=0}^{m-1} a_i z^i, a_i \in GF(2)$
Input: $B = \sum_{i=0}^{d-1} B_i z^{Di}$, where B_i is as in (1)
Output: $S = A \cdot B \bmod f(z)$
$f(z) = z^m + \sum_{i=0}^{m-1} c_i z^i$,where $c_i \in GF(2)$
1: $S \leftarrow 0$
2: for $i = \lceil m/D \rceil - 1$ to 0 do
  2.1: $S \leftarrow A \cdot B_i + S$
  2.2: $S \leftarrow S \cdot z^D \bmod f(z)$
3: Return(S)

図2 MSDアルゴリズム．

2. digit-serial乗算

GF(2^m)の要素AとBとし，乗算$S = A \cdot B \bmod f(z)$を演算する方法は様々に提案されている．基本的な手法としてはmビット×1ビットずつ処理するbit-serialアルゴリズム[2]がある．bit-serialアルゴリズムの一つであるBの最上位ビットから処理するMSB乗算アルゴリズムを図1に示す．

一方，digit-serialアルゴリズム[3]は複数のビット同時に処理する．同時に処理するデータ量Dをdigitサイズと定義する．BをDで分割したとき，digitの数は$d = \lceil m/D \rceil$となる．このとき$B = \sum_{i=0}^{d-1} B_i z^{Di}$となり，$B_i$は式(1)となる．

$$B_i = \sum_{j=0}^{D-1} b_{Di+j} z^j, 0 \leq i \leq d-1, b_i \in GF(2) \qquad (1)$$

従って，digit-serialアルゴリズムにおける最上位Digitから処理するMSDアルゴリズムは図2のように表すことができる．

3. MSB乗算器をベースとしたMSD乗算器

本章では，MSB乗算器をベースにしたMSD乗算器を提案する．MSB乗算器をD個を直列に接続することでMSD乗算器を実現する．MSB乗算器のように1ビットシフトするごとにReduction演算を行ない，その出力を次のMSB乗算器に伝えることをDだけ繰り返すことで処理を行う．そのためデータ長はDに依存せず常にmビットにすることができ，面積の

Input: $A = \sum_{i=0}^{m-1} a_i z^i, a_i \in GF(2)$
Input: $B = \sum_{i=0}^{d-1} B_i z^{Di}$, where B_i is as in (1)
Output: $S = A \cdot B \bmod f(z)$
$f(z) = z^m + \sum_{i=0}^{m-1} c_i z^i$,where $c_i \in GF(2)$

1: $S \leftarrow 0$
2: for $i = \lceil m/D \rceil - 1$ to 0 do
  2.1: for $j = D - 1$ to 0 do
    2.1.1: $S \leftarrow A \cdot (B_i)_j + S$
    2.1.2: if($i == 0$ and $j == m\%D$)
      Return(S)
    2.1.3: $S \leftarrow S \cdot z \bmod f(z)$

図 3 提案 MSD-first アルゴリズム.

増加率を抑えることができる．従来の手法[5],[10]では，乗算部と Redution 部が分かれており，どちらも D に依存するため提案手法よりも回路が複雑になる．また，乗算部と Reduction 部をレジスタで分離しているため 1 回の乗算に必要なクロックサイクル数が提案手法より多い．図 3 に提案 digit-serial 乗算アルゴリズムを示す．2.1 の for ループ部が MSB-first 型乗算器である．乗算，Reduction 演算を digit サイズ D ごと処理するのではなく 1 ビットごとに処理している．m=163，D=4 の時のブロック図を図 4 に示す．

クロックサイクル数と出力位置は digit サイズ D から計算する．各パラメータの求め方は以下の通りである．

- 1 回の乗算に必要なクロックサイクル数 $= \lceil m/D \rceil$
- 出力 $= m\%D$

m%n : m を D で除算したときの余り

$D \in \{0, \ldots m-1\}$ であるが，そのすべてを評価する必要はなく，サイクル数が同じなら D が小さい値を用いればよい．例えば表 1 に示すように，m=163, D=21,22,23 の時，サイクル数は同じである．この場合 D=21 が最もハードウェア量が少なくなるため D=21 のみを選べばよい．

3.1 面積と遅延の見積もり

提案手法の面積と遅延の見積もりについて述べる．乗算に必要な回路は AND ゲート部，乗算と Reduction 演算を行う XOR ゲート部，途中結果を保存する Accumulator 部で構成されている．AND ゲート部は m ビットの A と D ビットの B_i との AND を取るのでゲート数は mD である．同様に加算を行う XOR ゲートも mD となる．Reduction 演算を行う XOR ゲートは $f(x)$ の項数 k に依存し，kD となる．従って XOR ゲートは $(m+k)D$ になる．Accumulator 部は m ビットのレジスタである．表 2 に面積の見積もり値を示す．文献[5],[10]の手法は乗算部の出力が $m+D$ となるためレジスタのサイズが大きくなる．

表 1 m=163, D=21,22,23 時のパラメータ.

m	D	商	余り	サイクル数
163	21	7	16	8
163	22	7	9	8
163	23	7	2	8

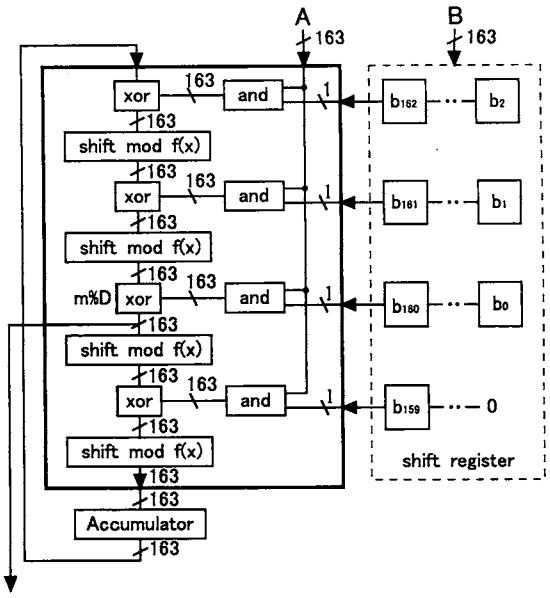

図 4 提案 MSD-first 乗算器のブロック図 ($GF(2^{163}), D = 3$).

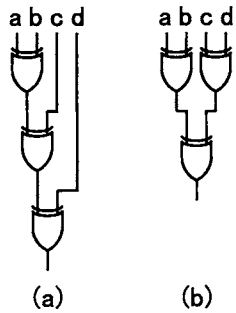

図 5 XOR ゲートの構造 (a) アレイ型 (b)2 分木型.

遅延の見積もりは D に対する出力 $S_i (i \in \{0, 1, \ldots, m-1\})$ を計算するのに必要な項 $\sum_{i=0}^{m-1} \sum_{j=0}^{m-1} a_i b_j$ の最大数から求める．$a_i b_j$ は AND で計算するが，依存関係はないので AND ゲート 1 つ分の遅延で済む．XOR ゲートは MSB 乗算器を並べただけでは遅延は digit サイズ D に比例して増加してしまうが，図 5 に示すように XOR のアレイ構造を 2 分木構造に再構成し，遅延を対数に抑えることができる．そのため XOR ゲートの遅延は出力までに加算される項数の対数で求めることができる．表 3 にサイクル数と遅延の見積もりを示す．k は法 $f(z)$ の項数である．NIST 推奨パラメータ[1]の場合，$k = 2, 4$ である．

提案手法は次のような優位点を持つ．

- MSB 乗算器を D 個直列につなぐだけなので設計が容易
- クロックサイクル数が $\lceil m/D \rceil$ で済む．

4. 合成結果

NIST 推奨パラメータ[1]に準拠した m=163,$f(z) = z^{163} + z^7 + z^6 + z^3 + 1$ における提案乗算器を実装した．STARC90nm

表 2　面積の見積もり.

	# XOR	# AND	# FF
提案手法	$(m+k)D$	mD	m
文献[10]	$(m+k+1)D+(k+1)(D-1)$	$(m+k)D+(k+1)(D-1)$	$2m+D+k$
文献[5]	$(m+r-2)D+(r-1)(D-1)$	$(m+r-1)D+(r-1)(D-1)$	$3m+D-2$

表 3　サイクル数と遅延の見積もり.

	サイクル数	遅延
提案手法	$\lceil m/D \rceil$	$1\Delta_{AND}+\lceil \log_2(r(D-1)) \rceil \Delta_{XOR}$
SAM[10]	$\lceil m/D \rceil+1$	$1\Delta_{AND}+\lceil \log_2(D+1) \rceil \Delta_{XOR}$
DAM[5]	$\lceil m/D \rceil+1$	$1\Delta_{AND}+\lceil \log_2(\lceil D/2 \rceil+1) \rceil \Delta_{XOR}$

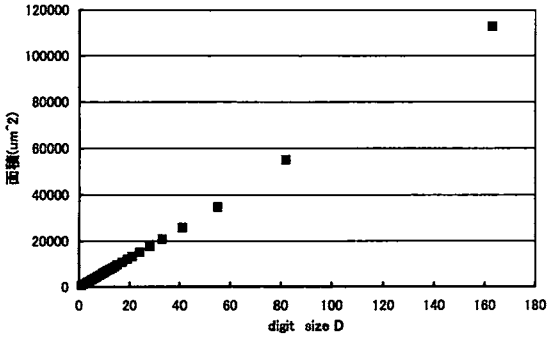

図 6　digit サイズ D と面積の関係.

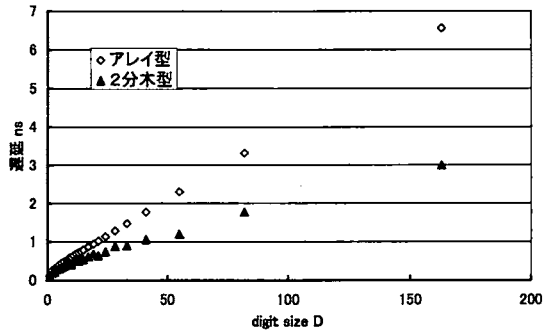

図 7　digit サイズ D と遅延の関係.

$z^7+z^6+z^3+1$ における提案乗算器を実装した．STARC90nm プロセス用ライブラリを使用し，DesignComplier W-2004.12-SP2 で合成した．D に対する面積のグラフを図 6，D に対する遅延のグラフを図 7 にそれぞれ示す．面積は見積もり通り digit サイズ D に対して線形に増加している．遅延は参考としてアレイ型にした場合を載せた．アレイ型は線形に増加するのに対し，2 分木にすると遅延の増加率が小さくなることが分かる．図 8 は一回の乗算に必要な処理時間をプロットしたグラフである．サイクル数×遅延で計算している．D が大きいほど処理速度は向上するが，面積が大きくなるため D は実用的な範囲があり，合成結果から $D=3,\cdots,21$ であることが分かった．

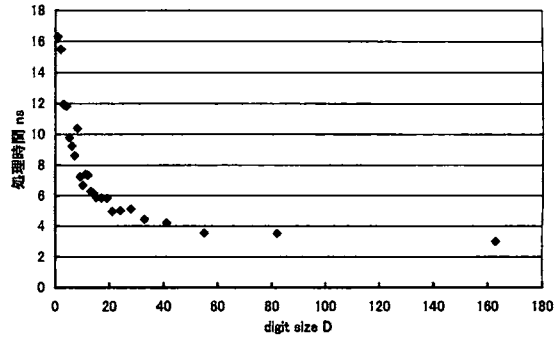

図 8　digit サイズ D と処理速度の関係.

表 4　配置配線後の性能.

Field	Platform	ゲート数	最大周波数	処理時間
$GF(2^{163})$	0.35μm	45k	50MHz	0.115ms

5. 実　　装

提案手法の乗算アーキテクチャを用いた楕円曲線暗号処理回路を ROHM0.35μm プロセス用ライブラリを用いて実装した．ライブラリ作成は Synopsys Milkyway X-2005.09，配置配線には Synopsys Astro X-2005.09，DRC/LVS には Mentor Calibre をそれぞれ使用した．NIST 推奨パラメータ [1] に基づいた $GF(2^{163})$ 上の楕円曲線を用いた．スカラー乗算 $(Q=kP)$ アルゴリズムは [6] を用いたが，秘密鍵による処理部のみ実装した．アフィン座標から射影座標，射影座標からアフィン座標への変換回路は実装していない．乗算器は $GF(2^{163})$，digit サイズ $D=48$ で設計した．ブロック図を図 9 に示す．配置配線後の様子を図 10 に示す．表 4 に配置配線後の性能を示す．ゲート数は約 45k，クリティカルパスは 20ns となった．論理合成時は 10ns であったが，クロックツリー合成後には 20ns となった．座標変換を除いたスカラー乗算 $(Q=kP)$ を約 0.115ms で処理できた．

6. む　す　び

本稿では，楕円曲線暗号に用いる $GF(2^m)$ 上での MSB 乗算器をベースとした MSD 乗算器について提案した．MSB 乗算器を直列に接続することで MSD 乗算器を構成できるため設計が容易であり，従来の手法と比較して，面積が小さく，1 回の乗算に必要なクロックサイクル数が小さいことを示した．また ROHM0.35μm でチップ試作を行い，提案手法の有効性を示した．今後は，チップのテストと消費電力の見積もりを行う予定である．

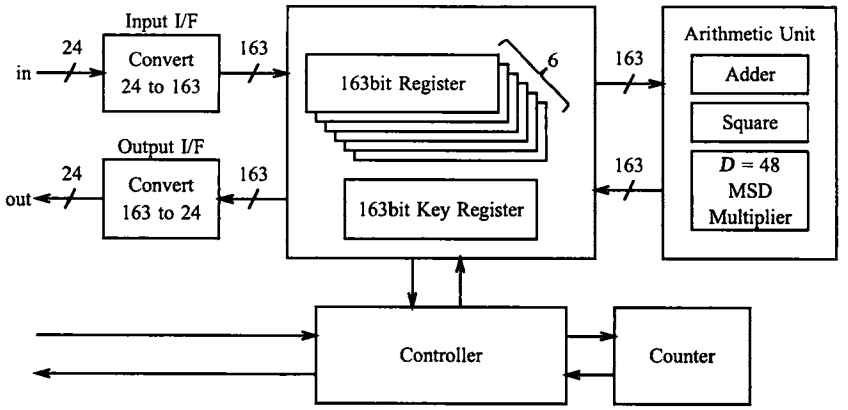

図 9　楕円曲線暗号処理回路のブロック図.

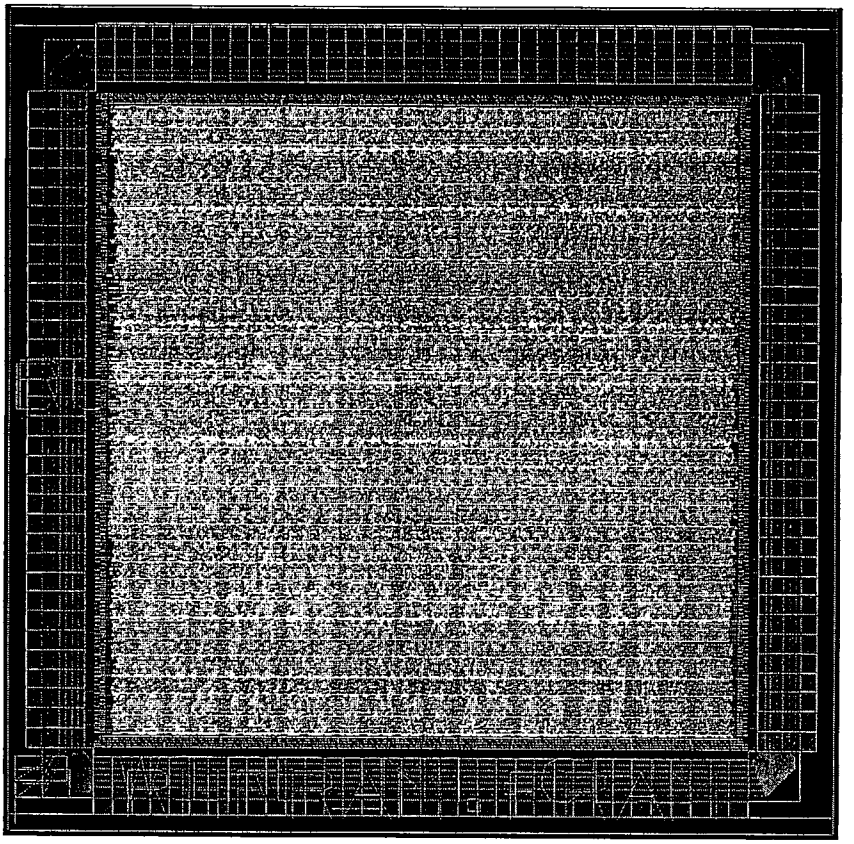

図 10　配置配線後の様子.

謝　　辞

本チップ試作は東京大学大規模集積システム設計教育研究センターを通し ローム (株) および凸版印刷 (株) の協力で行われたものである。

文　　献

[1] *IEEE Standard Specifications for Public-Key Cryptography*, IEEE Std. 1363-2000.

[2] T. Beth and D. Gollmann, "Algorithm engineering for public key algorithms." *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 458–465, 1989.

[3] R.I. Hartley and K.K. Parhi, *Digit-Serial Computation*, Kluwer Academic Publishers, 1995.

[4] N. Koblitz, "Elliptic Curve Cryptosystems," *Math. Computation*, vol. 48, pp. 203–209, 1987.

[5] S. Kumar, T. Wollinger and C. Paar, "Optimum Digit Serial $GF(2^m)$ Multipliers for Curve-Based Cryptography," *IEEE TRANSACTIONS ON COMPUTERS*, vol. 55, no. 10, pp. 1306–1311, Oct. 2006.

[6] J. Lopez and R. Dahab, "Fast multiplication on elliptic curves over $GF(2^m)$ without precomputation," *Cryptographic Hardware and Embedded Systems – CHES'99, Springer-Verlag, Lecture Notes in Computer Science 1717*, pp. 316–327, August, 1999.

[7] V. Miller, "Uses of Elliptic Curves in Cryptography," *Advances in Cryptology, Proc. CRYPTO '85*, H.C. Williams, ed., pp. 417–426, 1986.

[8] R.L. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Comm. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[9] A. Satoh and K. Takano, "A Scalable Dual-Field Elliptic Curve Cryptographic Processor," *IEEE TRANSACTIONS ON COMPUTERS*, vol. 52, no. 4, pp. 449–460, April, 2003.

[10] L. Song and K.K. Parhi, "Low Energy Digit-Serial/Parallel Finite Field Multipliers," *J. VLSI Signal Processing*, vol. 19, no. 2, pp. 149–166, June. 1998.