Bit-serial multiplication in $GF(2^m)$ using irreducible all-one polynomials

S.T.J. Fenn
M.G. Parker
M. Benaïssa
D. Taylor

Indexing terms: Bit-serial multipliers, Reed–Solomon codes, VLSI, Cryptography

Abstract: Two architectures for carrying out bit-serial multiplication in $GF(2^m)$ are presented where the defining irreducible polynomial for the field is an all-one polynomial. The multipliers presented have low hardware requirements, regular structures and are therefore suitable for VLSI implementation.

1 Introduction

Finite fields of the form $GF(2^m)$ have found applications to cryptography and error correcting codes such as Reed–Solomon (RS) codes [1]. RS codes are used in a variety of technologies such as CDs [2], the Hubble space telescope [3] and channel coding for compressed video services [4]. RS codes operate over finite fields and correct, not individual bits, but symbols where each symbol is an element of $GF(2^m)$ and hence is represented by m bits. Because RS codes operate over finite fields there is a need for fast and hardware efficient arithmetic operators for $GF(2^m)$ if RS encoders and decoders are to be efficiently implemented in hardware.

Of the arithmetic operations required in the implementation of RS codes, finite-field multiplication is the most frequently studied [5–12]. This is because addition is trivial to implement in hardware and because operations such as inversion and division can be decomposed into repeated multiplications. The two most well-known bit-serial architectures are the Berlekamp multiplier (BM) [5] and the Massey–Omura multiplier (MOM) [6]. Of these, the BM has lower hardware requirements and an easy-to-derive structure based on the defining irreducible polynomial for the field $f(x)$. The BM is also particularly suited to applications where constant multiplication is required such as RS encoders. The only disadvantage of the BM is that it operates over two bases, the dual basis and the polynomial basis [5]. The MOM operates over just one basis, the normal basis, which has the advantage that the square of a field element can be generated by a cyclic shift of the basis coefficients. However, the MOM requires more hardware than the BM, cannot efficiently carry out constant multiplication and, for a given normal basis, it is not obvious what the defining boolean function for the multiplier is [9].

In this brief contribution we consider bit-serial multiplication in $GF(2^m)$ for which $f(x) = x^m + x^{m-1} + ... + x + 1$, that is, where $f(x)$ is an all-one polynomial (AOP). A number of hardware efficient bit-parallel architectures have been presented for those fields for which $f(x)$ is an AOP [10–12], however we are not aware of any bit-serial architectures being described. In this contribution two bit-serial multipliers are presented for which $f(x)$ is an AOP. One of these operates over an extended basis of $(m + 1)$ coefficients while the other operates over a basis comprising m elements. These approaches allowed us to derive regular and hardware-efficient structures which are also appropriate for carrying out constant multiplication. In one of these two cases, squaring can also be carried out by a reordering of basis coefficients and so the proposed multiplier demonstrates the advantages of both the BM and MOM without any of the disadvantages.

2 Mathematical background

Let $f(x) = x^m + x^{m-1} + ... + x + 1$ be an irreducible AOP over $GF(2)$ and let α be a root of $f(x)$. ($f(x)$ can only be chosen as an irreducible AOP for $GF(2^m)$ if 2 is an element of order m modulo $(m + 1)$, where $(m + 1)$ is prime [10].) Then $\{1, \alpha, ..., \alpha^{m-1}\}$ forms the polynomial basis for $GF(2^m)$ and any field element $A \in GF(2^m)$ can be represented as $A = \Sigma_{i=0}^{m-1} A_i\alpha^i$ where $A_i \in GF(2)$. If we now let $\{1, \alpha, ..., \alpha^{m-1}, \alpha^m\}$ be an 'extended polynomial basis' the field element A can also be represented as $A = \Sigma_{i=0}^{m} a_i\alpha^i$ where $a_i \in GF(2)$. These A_i and a_i values are related [10] by the equation

$$A_i = a_i + a_m \quad (i = 0, 1, \ldots, m - 1) \qquad (1)$$

Over the extended polynomial basis, $A^2 = a_0 + \mathrm{a}_{m/2+1}\alpha + a_1\alpha^2 + ... + a_{m/2-1}\alpha^{m-2} + a_m\alpha^{m-1} + a_{m/2}\alpha^m$. Hence the extended basis representation of A^2 can be obtained from that of A with only a reordering of the basis coefficients.

Now let $a, b, c \in GF(2^m)$ such that $a = bc$ and represent these elements in the extended polynomial basis as $a = \Sigma_{i=0}^{m} a_i\alpha^i$, $b = \Sigma_{i=0}^{m} b_i\alpha^i$ and $c = \Sigma_{i=0}^{m} c_i\alpha^i$. Then the fol-

IEE Proceedings online no. 19971586

Paper first received 6th November 1996 and in revised form 14th August 1997

S.T.J. Fenn, M. Benaissa, and D. Taylor are with the Department of Electronic & Electrical Engineering, The University of Huddersfield, Queensgate, Huddersfield, HD1 3DH, UK

M.G. Parker is with the Department of Electronic & Electrical Engineering, The University of Bradford, Bradford, BD7 1DP, UK

lowing relationship holds:

$$\begin{bmatrix} a_m \\ a_{m-1} \\ \vdots \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} b_0 & b_1 & \cdots & b_m \\ b_m & b_0 & \cdots & b_{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_2 & b_3 & \cdots & b_1 \\ b_1 & b_2 & \cdots & b_0 \end{bmatrix} \begin{bmatrix} c_m \\ c_{m-1} \\ \vdots \\ c_1 \\ c_0 \end{bmatrix} \qquad (2)$$

Eqn. 2 was implicitly used to derive the bit-parallel multiplier in [10] and is now used in the design of two bit-serial multipliers.

3 Bit-serial multiplication

3.1 Bit-serial AOP multipliers

Consider eqn. 2 and Fig. 1. If the registers in Fig. 1 are initialised by $y_i = c_i$ and $x_i = b_i$ for $(i = 0, 1, ..., m)$ the first product bit a_m will immediately be available on the output line. The remaining product bits a_i $(i = m - 1, m - 2, ..., 1, 0)$ are obtained by clocking the upper shift register a further m times.

The multiplier presented here has a very similar structure to the bit-serial BM [5] but requires two extra register elements and an extra AND gate. A BM will require at least as many XOR gates as the proposed architecture, if not more, depending on the complexity of the irreducible polynomial for the field. Also, because it operates over words of length $(m + 1)$ bits rather than m bits, the proposed multiplier requires an extra clock cycle to yield a solution as compared to the BM. However, for large values of m these disadvantages are offset by the greater degree of regularity displayed by the proposed architecture.

Furthermore, like the BM, the proposed structure can efficiently carry out constant multiplication. This is because the c_i values do not have to be shifted once loaded into the circuit and so the multiplier can be hardwired to carry out many constant multiplications. These multipliers are therefore highly suited to applications where constant multiplication is required, such as in RS encoders and syndrome calculators.

3.2 Bit-serial modified AOP multipliers

We now show how to modify the structure to form an m-bit-serial multiplier. Let $b_m = c_m = 0$ in eqn. 2. Then

$$\begin{bmatrix} a_m \\ a_{m-1} \\ \vdots \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & \cdots & b_{m-1} & 0 \\ b_0 & b_1 & \cdots & b_{m-2} & b_{m-1} \\ 0 & b_0 & \cdots & b_{m-3} & b_{m-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_3 & b_4 & \cdots & b_0 & b_1 \\ b_2 & b_3 & \cdots & 0 & b_0 \end{bmatrix} \begin{bmatrix} c_{m-1} \\ c_{m-2} \\ \vdots \\ c_1 \\ c_0 \end{bmatrix} \qquad (3)$$

Eqn. 3 can be used to derive a structure in which both inputs contain m bits but the output contains $(m + 1)$ bits. However, we require both the inputs and the outputs to be represented by only m bits. To achieve this, note from eqn. 1 that $A_i = a_i + a_m$ $(i = 0, 1, ..., m - 1)$ and from eqn. 3 $a_m = \Sigma_{i=1}^{m-1} b_i c_{m-i}$. Hence this value of a_m can be generated and added to the a_i values to form A_i $(i = 0, 1, ..., m - 1)$. A circuit implementing this multiplier for $GF(2^4)$ is shown in Fig. 2. After four clock cycles the registers hold the values $x_i = b_i$ and $y_i = a_i$ $(i = 0, 1, ..., m - 1)$, $x_4 = 0$ and $y_4 = a_4 = b_1c_3 + b_2c_2 + b_3c_1$. The first product bit A_3 will be immediately available on the output line and the remaining product bits A_k $(k = 2, 1, 0)$ are obtained by clocking the upper shift register a further three times.

The MAOPM requires an extra $(m - 2)$ AND gates and an extra $(m - 2)$ XOR gates compared to the AOPM, or roughly double the number of combinational gates. However, the MAOPM operates on only m bits as opposed to the $(m + 1)$ of the AOPM and therefore requires one less clock cycle to generate a result. A summary of the characteristics of the multipli-

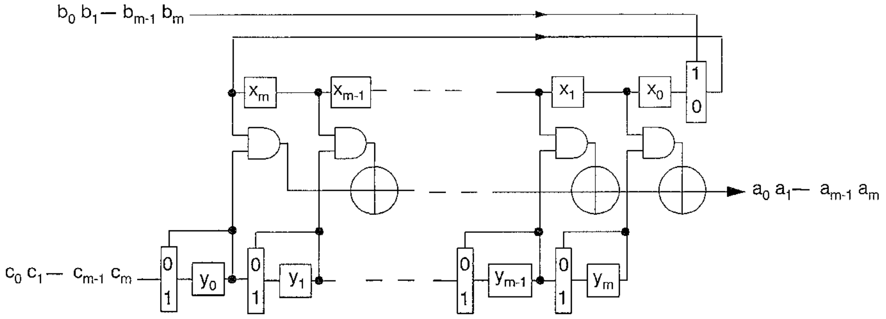

Fig. 1 *Bit serial AOPM for GF(2^m)*

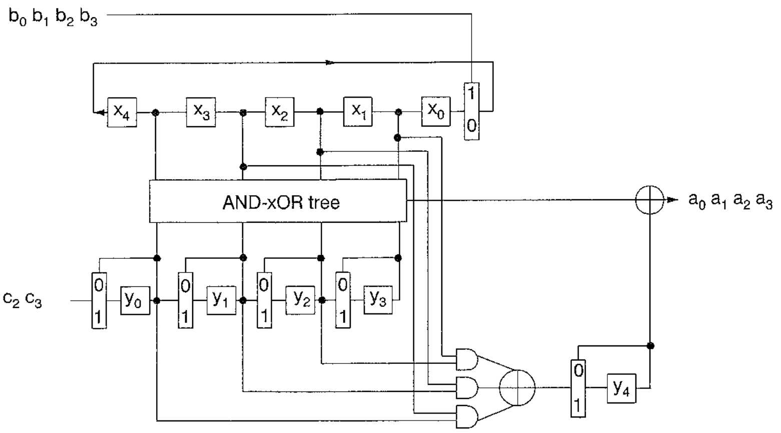

Fig. 2 *Bit serial MAOPM for GF(2^m)*

ers described here and those of the BM and the MOM are presented in Table 1. All four multipliers have approximately the same number of registers but the AOPM and BM require approximately half the number of XOR and AND gates as compared to the MOM and MAOPM. The main advantage offered by the AOPM and the MAOPM over the other two multipliers is that the defining Massey–Omura function and dual basis do not have to be generated. This results in the multipliers presented having simple, more regular architectures.

Table 1: Comparison of characteristic of multipliers

	Multiplier			
Characteristic	BM	MOM	AOPM	MAOPM
Registers	$2m$	$2m$	$2m+2$	$2m+2$
AND gates	m	$\geq 2m-1$	$m+1$	$2m-1$
XOR gates	$\geq m$	$\geq 2m-2$	m	$2m-2$
Clock cycles to solution	m	m	$m+1$	m

It is worth noting that eqn. 3 can also be used to derive a bit-parallel multiplier. This multiplier comprises m identical modules consisting of $(m-1)$ AND gates and $(m-2)$ XOR gates and one further module comprising m AND gates and $(m-1)$ XOR gates. In addition, a further m XOR gates are required to add a_m to a_i $(i = 0, 1, ..., m-1)$. This multiplier therefore has exactly the same hardware requirements as the bit-parallel multiplier presented in [12]. The delay of these two multipliers due to gate delays is also the same.

4 Conclusions

Bit-serial multiplication in the finite field $GF(2^m)$ has been considered. It has been shown that when the defining irreducible polynomial for the field is an all-one polynomial, hardware efficient and regular architectures can be derived. One of these multipliers operates over an extended basis of $(m+1)$ bits and has a particularly simple architecture. This basis also has the advantage that squaring can be carried out with only a reordering of basis coefficients. The disadvantage of this architecture compared with traditional bit-serial architectures is that, because it operates over an extended basis, it takes an extra clock cycle to yield a result. The second architecture presented has higher hardware requirements but only operates over a basis of m elements. Both multipliers can be hardwired to carry out constant multiplication and are expected to find applications in RS codecs.

5 References

1 LIDL, R., and NIDERREITER, H.: 'An introduction to finite fields and their applications' (CUP, Cambridge, 1986)

2 HOEVE, H., TIMMERMANS, J., and VRIES, L.B.: 'Error correction and concealment in the compact disc system', *Philips Tech. Rev.*, 1982, **40**, (6), pp. 166–172

3 WHITAKER, S.R., CAMERON, K., MAKI, G., CANARIS, J., and QWSLEY, P.: 'VLSI Reed–Solomon processor for the Hubble Space Telescope' *in* 'VLSI signal processing' (IEEE Press, 1991), Chap. 35

4 DRAY, G.N.: 'DVB channel coding standards for broadcasting compressed video services', *Electron. Commun. Eng. J.*, 1997, **9**, (1), pp. 11–20

5 BERLEKAMP, E.R.: 'Bit-serial Reed–Solomon encoders', *IEEE Trans.*, 1982, **IT-28**, (6), pp. 869–874

6 WANG, C.C., TRUONG, T.K., SHAO, H.M., DEUTSCH, L.J., OMURA, J.K., and REED, I.S.: 'VLSI architectures for computing multiplications and inverses in $GF(2^m)$', *IEEE Trans.*, 1985, **C-34**, (8), pp. 709–716

7 FENN, S.T.J., BENAISSA, M., and TAYLOR, D.: '$GF(2^m)$ multiplication and division over the dual basis', *IEEE Trans. Comp.*, 1996, **C-45**, (3), pp. 319–327

8 MASTROVITO, E.D.: 'VLSI architectures for computations in Galois fields'. PhD Thesis, Linkoping University, Sweden, 1991

9 HSU, I.S., TRUONG, T.K., DEUTSCH, L.J., and REED, I.S.: 'A comparison of VLSI architectures of finite field multipliers using dual, normal or standard bases', *IEEE Trans.*, 1988, **C-37**, (6), pp. 735–737

10 ITOH, T., and TSUJII, S.: 'Structure of parallel multipliers for a class of fields $GF(2^m)$', *Inf. Comput.*, 1989, **83**, pp. 21–409

11 HASAN, M.A., WANG, M.Z., and BHARGAVA, V.K.: 'Modular construction of low complexity parallel multipliers for a class of finite fields $GF(2^m)$', *IEEE Trans.*, 1992, **C-41**, (8), pp. 962–971

12 HASAN, M.A., WANG, M.Z., and BHARGAVA, V.K.: 'A modified Massey–Omura parallel multiplier for a class of finite fields', *IEEE Trans.*, 1993, **C-42**, (10), pp. 1278–1280